UNIVERSIDAD ESTATAL DE MILAGRO

REPÚBLICA DEL ECUADOR

VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO

PROYECTO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE: MAESTRÍA

MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN

TÍTULO DEL PROYECTO:

IMPLEMENTACIÓN DE UNA PLATAFORMA DE MONITOREO PARA LA GESTIÓN Y CONTROL DE LOS DISPOSITIVOS CRÍTICOS DEL CENTRO DE CONTROL DE RED (NOC) DEL PROVEEDOR DE SERVICIOS DE INTERNET IN.PLANET S.A., DE LA CIUDAD DE MILAGRO.

ROBERTO OMAR ANDRADE PAREDES
TUTOR

PIÑA CAMPOVERDE LEONARDO WASHINGTON
AUTOR

MILAGRO, SEPTIEMBRE, 2022

UNEMI
UNIVERSIDAD ESTATAL DE MILAGRO

VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO

Milagro, 4 de octubre, 2022

CERTIFICACIÓN DE ACEPTACIÓN DEL TUTOR

En calidad de Tutor del Proyecto de Investigación, nombrado por el Comité Académico del Programa de Maestría en Tecnologías de la Información de la Universidad Estatal de Milagro.

CERTIFICO

Que he analizado el Proyecto de Investigación con el tema **IMPLEMENTACIÓN DE UNA PLATAFORMA DE MONITOREO PARA LA GESTIÓN Y CONTROL DE LOS DISPOSITIVOS CRÍTICOS DEL CENTRO DE CONTROL DE RED (NOC) DEL PROVEEDOR DE SERVICIOS DE INTERNET IN.PLANET S.A., DE LA CIUDAD DE MILAGRO**, elaborado por **LEONARDO WASHINGTON PIÑA CAMPOVERDE** el mismo que reúne las condiciones y requisitos previos para ser defendido ante el tribunal examinador, para optar por el título de **MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN**.

Firmado electrónicamente por:
ROBERTO OMAR ANDRADE PAREDES

_____

ROBERTO OMAR ANDRADE PAREDES

C.I: 1715509475

DECLARACIÓN DE AUTORÍA DE LA INVESTIGACIÓN

El autor de esta investigación declara ante el Comité Académico del Programa de Maestría en Tecnologías de la Información de la Universidad Estatal de Milagro, que el trabajo presentado es de mi propia autoría, no contiene material escrito por otra persona, salvo el que está referenciado debidamente en el texto; parte del presente documento o en su totalidad no ha sido aceptado para el otorgamiento de cualquier otro Título de una institución nacional o extranjera.

Milagro a los 9 días del mes de marzo de 2023.

_____

LEONARDO WASHINGTON PIÑA CAMPOVERDE

C.I: 0929137586

UNEMI
UNIVERSIDAD ESTATAL DE MILAGRO

VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO

DIRECCIÓN DE POSGRADO

CERTIFICACIÓN DE LA DEFENSA

El TRIBUNAL CALIFICADOR previo a la obtención del título de **MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN**, presentado por **ING. PIÑA CAMPOVERDE LEONARDO WASHINGTON**, otorga al presente proyecto de investigación denominado "IMPLEMENTACIÓN DE UNA PLATAFORMA DE MONITOREO PARA LA GESTIÓN Y CONTROL DE LOS DISPOSITIVOS CRÍTICOS DEL CENTRO DE CONTROL DE RED (NOC) DEL PROVEEDOR DE SERVICIOS DE INTERNET IN.PLANET S.A., DE LA CIUDAD DE MILAGRO.", las siguientes calificaciones:

TRABAJO DE TITULACION	59.00
DEFENSA ORAL	40.00
PROMEDIO	**99.00**
EQUIVALENTE	**Excelente**

Mgti. BRAVO DUARTE FREDDY LENIN
PRESIDENTE/A DEL TRIBUNAL

HERRERA TAPIA JORGE SERGIO
VOCAL

M.A.E. VINUEZA MORALES MARIUXI GEOVANNA
SECRETARIO/A DEL TRIBUNAL

CESIÓN DE DERECHOS DE AUTOR

Doctor

PhD. Fabricio Guevara Viejó
Rector de la Universidad Estatal de Milagro

Presente.

Mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor del Trabajo realizado como requisito previo a la obtención de mi Titulo de Curato Nivel, cuyo tema fue IMPLEMENTACIÓN DE UNA PLATAFORMA DE MONITOREO PARA LA GESTIÓN Y CONTROL DE LOS DISPOSITIVOS CRÍTICOS DEL CENTRO DE CONTROL DE RED (NOC) DEL PROVEEDOR DE SERVICIOS DE INTERNET IN.PLANET S.A., DE LA CIUDAD DE MILAGRO y que corresponde al Vicerrectorado de Investigación y Postgrado.

Milagro a los 9 días del mes de marzo de 2023.

Firmado electrónicamente por:
LEONARDO WASHINGTON PINA CAMPOVERDE

_____

LEONARDO WASHINGTON PIÑA CAMPOVERDE

C.I: 0929137586

AGRADECIMIENTO

Agradezco a mis padres, Olmedo Piña y Germania Campoverde por darme la vida, y la oportunidad de poder realizar mis estudios de cuarto nivel. A mi hermano Darío Piña, por su apoyo incondicional, sobre todo en el proceso de redacción. Agradezco a mi tutor, Master Roberto Andrade Paredes por su esfuerzo y dedicación en guiarme para culminar con éxito el presente TFM, a cada docente que compartieron sus conocimientos en las diferentes cátedras que impartieron a lo largo del tiempo que duró la maestría. Un agradecimiento especial a Jerry Palomeque, gracias por toda la retroalimentación en la fase de pruebas e implementación del producto final, sin su gran ayuda no lo hubiese logrado a tiempo.

DEDICATORIA

Quiero dedicar el presente trabajo a mis padres; Olmedo Piña y Germania Campoverde, porque cada logro conseguido ha sido posible gracias a su amor y sacrificio, éste logro es de ustedes y de nadie más. Gracias y mil veces gracias. A mi hermano, Darío Piña, un elemento clave en la redacción de este trabajo, quien me brindo tardes inolvidables de ajedrez. A todos los amigos y familiares, que en su momento me dieron sus palabras de aliento para no dar el brazo a torcer. Finalmente, a Cristian Fernando Ortiz Paucar; amigo lo hemos logrado, espero que al otro lado del silencio también lo estes celebrando.

RESÚMEN

El presente trabajo de fin de maestría, tiene como objetivo implementar una plataforma de monitoreo haciendo uso de herramientas Open Source que permitirá al departamento del Network Operations Center, realizar un monitoreo eficiente a los equipos del core de la ISP. Im.Planet. S. A.

El desarrollo de la plataforma de monitoreo cumplió con los requerimientos solicitados por el NOC. La plataforma de monitoreo contó con la integración de las herramientas Open Source, LibreNMS y Grafana, las cuales permiten llevar un registro minucioso mediante la creación de dashboard y alertas; ya sean vía Telegram o correo electrónico, así como guardar el registro histórico de consumo del uso de ancho de banda. El trabajo de fin de maestría contó con la Metodología PPDIOO, la cual permitió implementar la plataforma de monitoreo de forma categórica y estructurada, a su vez, ayudó a cumplir con los objetivos trazados, como también a realizar las optimizaciones correspondientes (implementación de Prometheus) que ayudan a mejorar el monitoreo de core.

Palabras clave:

Plataforma Monitoreo, Core, LibreNMS, Grafana, Prometheus, Dashboard, Open Source.

ABSTRACT

The objective of this master's thesis is to implement a monitoring platform using Open-Source tools that will allow the Network Operations Center department to efficiently monitor the ISP's core equipment. InPlanet. S. A.

The development of the monitoring platform met the requirements requested by the NOC. The monitoring platform had the integration of Open-Source tools, LibreNMS and Grafana, which allow to keep a detailed record by creating dashboards and alerts, these are by Telegram or email, as well as saving the historical record of consumption of bandwidth usage. The end of master's degree work had the PPDIOO Methodology, which allowed implementing the monitoring platform in a categorical and structured way, in turn, helped to meet the objectives outlined, as well as to make the corresponding optimizations (Implementation of Prometheus) that help to improve core monitoring.

Keywords:

Monitoring Platform, Core, LibreNMS, Grafana, Prometheus, Dashboard, Open Source.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

INTRODUCCIÓN

La función principal de los ISP es la provisión de servicios de Internet bajo un esquema de demanda que debe cumplir niveles de acuerdo de servicios (SLA) predefinidos con sus clientes. El no cumplimiento de estos niveles de SLA puede generar una deserción por parte de los clientes o inclusive el pago de multas.

Para mantener los niveles de servicios, una práctica común de los ISP es a adopción de un NOC (Network Operation Center). Las ventajas principales de implementar un NOC son:

- Monitoreo de red activo
- Agilidad para la reacción en la resolución de problemas
- Planificación de acciones futuras en base de indicadores de rendimiento

Las complicaciones de implementar un nivel de monitoreo eficiente en el NOC durante periodos 24/7/365, es que requiere de la implementación de herramientas de paga que representan costos muy elevados. Este es el caso del El NOC de la ISP In.Planet S.A. que no cuenta con las adecuadas herramientas de monitoreo que le permita identificar detalladamente el estado de su core y servidores críticos.

Consecuentemente la importancia de este trabajo es identificar herramientas OpenSource que cumplan estas funciones y que no representen elevados costos para la empresa. Al existir una gran variedad de herramientas de monitoreo open, que ha llevado a saturar la selección de herramientas específica para funciones que se desea implementar. En este trabajo se pretende identificar las herramientas más versátiles que ayuden a implementar una plataforma que ayude al Network Operations Center (NOC) de In.Planet S.A, para monitorizar el core y servidores críticos. Posteriormente se comparará y seleccionará las herramientas que mejor se adecuen para el proceso de monitoreo de la red en el hardware que soportará las herramientas.

Finalmente, este trabajo tiene la finalidad de implementar una plataforma de monitoreo para la gestión y control de los dispositivos críticos del Network Operations Center del proveedor de servicios de internet In.Planet S.A., ciudad de Milagro. La implementación de una plataforma para realizar el monitoreo de la red, le va a permitir al NOC tener un mejor control de los eventos que ocurran en los dispositivos del core permitiendo el registro de dichos eventos, esto facilitará el monitoreo y la intervención, lo cual resulta una acción estratégica para evitar posibles problemas.

1. CAPÍTULO 1

1.1. Planteamiento del problema:

En la actualidad, el NOC de In.Planet S.A cuenta con varios servidores los cuales sirven para realizar monitoreo o administración de otros sistemas y clientes. Por tanto, existen distintos tipos de servidores que cumplen con la función de monitoreo, muchos analizan el tráfico de la red, estos reciben as solicitudes, respuestas a clientes y de los servidores, mientras tanto otros solicitan o responden a sus propios datos. Es así que, la plataforma de monitoreo puede verificar el tráfico que presente la red, de igual forma las solicitudes y respuestas con otros servidores, y clientes, sin interrumpir las operaciones.

Las herramientas que se pueden seleccionar para el monitoreo se centran en dos aspectos: herramientas de paga y Open Source. Por un lado, las herramientas de paga representan un gran costo para la empresa, en cambio las herramientas de código abierto no representan gastos adicionales a la empresa ya que son gratuito, y pueden cubrir las mismas funciones que uno de paga. (Vargas Maquilon, J. A., & Maruri Uriña, E. S. (2021) Las diferentes herramientas de monitoreo como Cacti, ACS, Zabbix, Nagios, y demás, son software Open Source, los cuales permiten realizar el trabajo de monitoreo gratuitamente, sin generar costes o algún plan mensual, obteniendo en singular los mismos resultados que una herramienta de pago.

Sin embargo, al existir una gran gama de herramientas de monitoreo, ha llevado a saturar la selección de herramientas específica para funciones que se desea implementar, por ello se pretende en este trabajo identificar las herramientas más versátiles que ayuden a implementar una plataforma de monitoreo que proporcione al Network Operations Center (NOC) de In.Planet S.A., con la finalidad de agilizar la identificación de posibles futuros problemas que afecten al cumplimiento de sus niveles de servicio.

1.2. Objetivos:

Este trabajo de titulación de fin de maestría tiene como finalidad, implementar una plataforma Open Source para el proceso de monitoreo del Network Operations Center de la ISP In.Planet S.A, misma que debe adecuarse a las necesidades que susciten dentro del departamento.

1.2.1. Objetivo general:

Analizar e implementar una plataforma Open Source para agilizar y mejorar los procesos de monitoreo del Network Operations Center de la ISP In.Planet S.A, para la prevención de posibles problemas en el core y servidores críticos.

1.2.2. Objetivo específico:

- Identificar herramientas OpenSource disponibles para contrarrestar las necesidades del Network Operations Center (NOC) de In.Planet S.A.
- Evaluar y seleccionar las herramientas de monitoreo OpenSource que posean un mayor rendimiento con el dashboard (Grafana) de visualización de datos.
- Instalar y configurar, las herramientas de monitoreo en conjunto con el dashboard de visualización de datos, que ayudarán a mejorar la eficiencia del Network Operations Center (NOC) en In.Planet S.A.

1.3. Alcance:

El presente trabajo de titulación, tiene la finalidad de implementar de una plataforma de monitoreo en el departamento del NOC de la ISP In.Planet S.A., el cual es un ISP que tiene presencia en varias ciudades del país, dentro del cual están distribuidos los diferentes Data Centers y nodos que ayudan a bridar el servicio de internet a sus abondos, sin embargo el presente trabajo se centrará exclusivamente en el core de la empresa, el cual es la parte fundamental de la empresa ya de dicho core es donde se alimentan los demás nodos y Data Centers de le empresa. El NOC necesita la implementación de un sistema con la capacidad monitorear, que muestre y alerte el estado en el que se encuentran los equipos, dicha plataforma permitirá saber el consumo de ancho de banda correspondientes a interfaces principales de equipos del core.

1.4. Estado del arte:

En el presente apartado se realizará la compilación teórica perteneciente al estado del arte, el cual cumple una función esencial para la implementación de la propuesta correspondiente al trabajo de titulación. Se mostrará las diversas plataformas de código abierto disponible, mismas que serán seleccionas según su versatilidad en la implementación metodológica.

1.4.1. Qué es la gestión de redes:

Red Hat. (2019) La gestión de las redes comprende el proceso mediante el cual se aprueba el control de la conectividad y las correspondiente a las configuraciones de diferentes dispositivos; como sistemas que contienen dos categorías, como lo son las redes ocultas y las superficiales.

Está dos particularidades que convergen en la gestión de redes, como lo son las redes subyacentes y superpuestas, implican el entendimiento particular del proceso de GR. Red Hat. (2019) La gestión correspondiente a las redes ocultas, implican la coordinación de los distintos dispositivos que corresponden a las estructuras físicas, como los hubs, los servidores, puertas de enlace, los módems, etc. Lo que se refiere como gestión de las redes superficiales, se debe entender en singular a la concepción de conexiones digitales y de administración de los respectivos accesos a los usuarios finales, así mismo de aplicaciones y los dispositivos existentes, virtual como nodos.

1.4.2. Protocolo SNMP.

Se refiere como "Protocolo Simple de Administración de Redes" a la aplicación establecida por la "Junta de arquitectura de Internet" en RFC1157, para realizar el cambio equivalente de la información, administración, entre distintos dispositivos de la red. Quispe Ccuno, J. R. (2019) este protocolo, es por defecto, una herramienta de capa de aplicación del modelo TCP/IP. Permite compartir de forma bidireccional la información de distintos dispositivos de red. El protocolo SNMP, podemos decir que es el estándar dentro de la red de computadoras. Las funciones por defecto del protocolo SNMP, es monitorizar el rendimiento de los equipos, permite recuperar fallas y concede la configuración de los equipos.

Es importante agregar que la mayor parte de los componentes de red de nivel profesional, poseen un agente SNMP adjunto. Mismos que deben estar habilitados y a su vez, configurados para el proceso de comunicación con el "NMS"

1.4.3. Plataformas de gestión (NMS):

Las plataformas de gestión, se encarga de la supervisión del funcionamiento de administración de la red. Según, Oré Alvaro, C. (2019) Las plataformas de gestión, ejecutan los software de monitorización, de una cantidad significativa que forma parte del CPU y

memoria necesarios para la gestionar la red. Es recomendable tener uno o varios software NMS deben los cuales deben existir en la red a administrar.

En otras palabras, un sistema para gestionar la red, corresponde a una aplicación o conjunto de aplicaciones destinada a los administradores de red para monitorear los dispositivos autónomos de una red, en correlación a la administración de una determina red. Consulting Informático (2021), Un Software de monitoreo de red, puede usarse en la monitorización de los componentes del software y hardware en una red. Básicamente, se encarga de registrar los datos de los puntos remotos de una determinada red para así llevar a cabo informes centrales a un administrador determinado del sistema.

1.4.3.1. Zabbix:

Es una herramienta de sistema para el monitore de red, de código libre, que permite monitorear de diversos parámetros de una red, el estado e integridad de servidores, máquinas virtuales, aplicaciones, servicios, bases de datos, sitios web, la nube, etc. (Zabbix, 2018) Usa un sistema de notificación de característica flexible el cual, permite que los operadores realizar configuraciones de alertas basadas en correo electrónico, en el caso de un evento adverso. Ello concede una alerta inmediata de los acontecimientos del servidor. También brinda admirables funciones con correspondencia a informes, y la visualización de datos de la información almacenada.

1.4.3.1.1. Características que posee la plataforma NMS, Zabbix:

Entre las características más relevantes de Zabbix podemos mencionar las siguientes:

Plataforma de código libre.

Recolección de datos:

- Comprobaciones de operatividad y rendimiento.
- Soporte para protocolo SNMP.
- Recopila datos requeridos a intervalos personalizados.
- Lo pueden realizar desde el servidor o haciendo uso de agentes.

Notificaciones:

- Las notificaciones pueden ser personalizadas.
- La cantidad notificaciones se pueden ser significativas y útiles utilizando variables macro.

Graficación en tiempo real:

- Los dispositivos monitoreados se grafican de manera inmediatamente usando la función de gráficos incorporados.

Capacidades de monitoreo web:

- puede seguir una ruta de clics del cursor que se puede simular en un sitio web y verificar la operatividad y lapso de respuesta.

Distintas opciones para visualización:

- Crea gráficos de manera personalizada los cuales pueden llegar a combinarse con varios elementos en una sola vista.
- Esquema de red.
- Informes.
- Vista estandarizada de los dispositivos supervisados.

Almacenamiento de datos históricos:

- Datos almacenados en una base de datos.
- Historial se puede configurar.
- Los procedimientos de limpieza sen encuentran incorporado.

Facilidad de configuración:

- Se pueden agregar distintos dispositivos para monitorizar.
- Los dispositivos se seleccionan para su seguimiento, una vez incorporados en la base de datos.
- Se pueden usar plantillas a los dispositivos monitoreados.

Detección de redes:

- Autodescubrimiento de dispositivos de red.
- Registro por defecto; agente.
- Hallazgos de sistemas de archivos, de interfaces de red y SNMP - OID.

Interfaz de programación de aplicaciones de Zabbix:

- La Interfaz de programación de aplicaciones de Zabbix provee una interfaz programable a Zabbix para operaciones masivas, unificación de software de terceros y otros fines.

Permisos:

- Autenticación de usuario segura.
- Ciertos usuarios pueden estar limitados a ciertas vistas.

Agente con sus funciones y aplicabilidad sencilla:

- Desplegado en objetivos de seguimiento.
- Se puede efectuar tanto en Linux como en Windows

1.4.3.2. LibreNMS:

LibreNMS que cumple la función de monitoreo de red de detección automática y bajo mantenimiento que acepta una variedad de dispositivos, al igual que plataformas y sistemas operativos como: Windows, NetApp, Cisco, Linux, HP, FreeBSD, Brocade, Dell, entre otros. LibreNMS está disponible de forma gratuita y recibe sus correspondientes actualizaciones, y funciones de forma periódica (dos veces al año). Cantos San Emeterio, J. (2021) LibreNMS muestra su fortaleza en las respuestas aceleradas de su software, el cual debe su eficiencia a una interfaz de programación de aplicaciones open source. Esta herramienta de monitorización en linea, no es inusitado para las más herramientas de monitoreo exintendentes de pago o gratuitas, pero lo que permite que LibreNMS sea particularmente útil es la proporción de la automatización de sus actualizaciones, enfocado en el rendimiento de la red a un sistema de alerta.

1.4.3.2.1. Características que posee la plataforma NMS, LibreNMS:

Entre las características más relevantes de LibreNMS podemos mencionar las siguientes:

Descubrimiento automático:

- Descubre de manera automática toda nuestra red, utilizando, LLDP, CDP, ARP, LLDP, SNMP, OSPF, BGP y FDP.

Personalización de alertas:

- Estructura dúctil de alertas, incluyendo la notificación mediante correo electrónico o por Internet Relay Chat, entre otros.

Acceso API:

- Una Internet Relay Chat completa para administrar, graficar y rescatar datos de instalación.

Sistema de cobro:

- Permite generar facturas de la velocidad de transferencia utilizado, para puertos en nuestra red, según el uso o transferencia.

Actualizaciones automáticas:

- Podemos tener el producto actualizado en todo momento, incluyendo corrección de errores, nuevas funcionalidades y mucho más.

Escalamiento distribuido:

- Escalamiento horizontal para crear con nuestra red.

Aplicaciones para iPhone y para Android:

- Las aplicaciones nativas para estos sistemas operativos de dispositivos móviles están disponibles, lo que proporciona una funcionalidad básica.

1.4.3.3. Cacti:

Es una herramienta open source que brinda permisibilidad para monitorizar la red. (Salas, G., Stteeven, J., & Roa Piñeros, C. A., 2020) Es un sistema de monitoreo mediante el cual podemos controlar, casi a tiempo real, los diversos dispositivos que abarcan los servicios de nuestra red; routers, CPU, servidores, cargas, temperaturas, tráfico de las interfaces, y otras. (Salas, G., Stteeven, J., & Roa Piñeros, C. A., 2020) Es un software que permite controlar de manera consecuente el estado de nuestra red. Esta herramienta posee un recolector de datos prometedor, un sistema eficiente de elaboración de plantillas y gráficos, también cuenta con una interfaz abastecida que permite la administración de usuarios. La aplicación está concebida en PHP, y emplea el software MySql que ayudad a almacenar la información de los gráficos y datos recogidos.

1.4.3.3.1. Características que posee la plataforma NMS, Cacti:

Entre las características más relevantes de LibreNMS podemos mencionar las siguientes:

Actualizaciones automáticas:

- Automatización de dispositivos.

Configuración de datos:

- Elementos gráficos ilimitados.
- Soporte de relleno automático para gráficos.
- Manipulación de datos gráficos.
- Plantillas de gráficos.
- Agregación de gráficos.
- Vistas de árbol, lista y vista previa de datos de gráficos.

Configuración de datos:

- Fuentes de datos flexibles.
- Recopilación de datos en un período de tiempo no estándar.
- Scripts de recopilación de datos personalizados.
- Recopilación de datos remota.

Gestión y seguridad basadas en usuarios y grupos de usuarios.

Detección de redes.

Soporte SNMP incorporado.

1.4.3.4. Nagios:

(Rueda Ortega, P., 2020) Nagios es un software gratuito enfocado en la monitorización de redes el cual, es ampliamente usado, que permite la monitorización tanto del hardware como del software que se especifican en la red que se esté visualizando, alertando posible problema. (Rueda Ortega, P., 2020) Permite comprobar el equipo que se está monitorizando por él, responda las solicitudes para evaluar su estado y en el caso de comportamiento no sea el esperado, la herramienta notificará al administrador, mediante correo electrónico, entre otros medios. Esta notificación permite su configuración, identificando cinco estados para los equipos y servicios, entre ellos se encuentran; "Up" "Down" "Warning" "Flapping" y "Critical".

1.4.3.4.1. Características que posee la plataforma NMS, Nagios:

Entre las características más relevantes de Nagios podemos mencionar las siguientes:

- Monitorear servicios de red (SMTP, HTTP, HTTPS, POP3, NTP, DNS, ICMP, SNMP, FTP, etc.).
- Supervise los recursos de hardware (carga de la CPU, uso del disco, procesos del sistema) en diferentes sistemas operativos.
- Monitoreo de escritorio remoto a través de túnel encriptado SSL o SSH.
- Diseño de complemento simple que permite a los usuarios crear sus propias pruebas de servicio según sus necesidades utilizando su lenguaje de programación preferido (Bash, C++, Perl, Ruby, Python, PHP), C#).
- Capacidad para definir la jerarquía de la red para distinguir las máquinas excluidas de las máquinas inaccesibles.
- Notificar a los contactos cuando haya problemas con el servicio o servidor y cuando estén resueltos.
- Rotar automáticamente los archivos de registro.
- Admite la implementación de hosts de pantalla de respaldo.
- Visualización en tiempo real del estado de la red a través de una interfaz web, con la capacidad de generar informes y gráficos de la actividad de los sistemas controlados, así como visualizar la lista de mensajes enviados, historial de fallas, archivo de registro.

1.4.3.5. Prometheus:

Prometheus es una herramienta open source que permite el monitoreo de la red en determinados lapsos de tiempo, se basa en la recopilación de métricas de los trabajos instrumentados, directa o a su vez por vía push Gateway, también permite el almacenamiento de los ejemplares, y poner en marcha las reglas definidas en estas, y de esta manera generar nuevas alertas. (Hidalgo de Benito, C., 2021) Este software es funge como un sistema para monitorizar crear alerta. Prometheus también puede comprenderse como una base de series de tiempo, esta permite almacenar datos de diferentes variables, la cuales se ordenan de manera cronológicas. Esta herramienta es parte del Cloud Native Computing Foundation, encargada que se encarga de ordenar los proyectos open source.

1.4.3.5.1. Características que posee la plataforma NMS, Prometheus:

Entre las características más relevantes de Prometheus podemos mencionar las siguientes:

- Modelo de datos multivariante con datos de series temporales identificados por nombres de métricas y pares clave/valor.
- PromQL, un lenguaje de consulta flexible que aprovecha esta multidimensionalidad.
- No depende del almacenamiento distribuido; los nodos de servidor individuales son autónomos.
- La agregación de series temporales se realiza mediante el modelo de extracción a través del protocolo HTTP.
- La transmisión de series temporales es manejada por un puerto intermedio.
- Destinos descubiertos por descubrimiento de servicios o configuración estática.
- Varios modos para el manejo de gráficos y cuadros de mando.

1.4.3.6. Grafana:

El software de código abierto de Grafana le permite consultar, visualizar, alertar y explorar sus métricas, registros y seguimientos dondequiera que estén almacenados. Grafana OSS le proporciona recursos para transformar los datos de su base de datos de series

temporales en gráficos y diversas visualizaciones. (Hidalgo de Benito, C., 2021) La herramienta permite observar de manera gráfica las métricas, y se debe tener en cuenta que los esquemas o gráficas son completamente personalizables. Dispone de ítems dinámicos, los cuales conceden la facilidad de crear y guardarlos como plantillas para usarlos posteriormente. También es permisible al momento de indagar los datos con solicitudes de información de consultas temporales y comparar los datos de distintas franjas de tiempo. Permite la visualización de exploraciones gracias a sus etiquetas, se pueden programar advertencias que se evalúen continuamente y notifiquen distintos sistemas.

1.4.3.6.1. Características que posee la plataforma NMS, Grafana:

Entre las características más relevantes de Grafana podemos mencionar las siguientes:

- Conéctese a todas las fuentes de datos posibles como: Graphite, Prometheus, Influx DB, ElasticSearch, MySQL, PostgreSQL, etc.
- Explorar, analizar y monitorear datos durante un período de tiempo.
- Rastrea el comportamiento del usuario y de la aplicación.
- Tiene un modelo híbrido entre nube privada y pública.
- Grafana dispone de paneles con diferentes opciones de visualización.
- El tablero de juego contiene varios tableros individuales diferentes en una cuadrícula. admite todos los análisis de nuestra aplicación.

Puede establecer alertas:

- Podemos buscar, visualizar, configurar alertas y analizar datos fácilmente usando métricas.

Proporciona una plataforma de complementos que facilita a todos los usuarios de Grafana la creación de complementos de alta calidad.

2. CAPÍTULO 2

El capítulo 2 comprende la estructura de la metodología implementada para llevar a cabo el desarrollo de la plataforma de monitoreo a implementar en el NOC.

2.1. Metodología:

El trabajo de fin de maestría se encuentra estructurado bajo la estructura metodológica: Preparar, Planificar, Diseñar, Implementar, Operar, Optimizar (PPDIOO). Estas seis fases se realizaron de forma secuencial. En la ilustración 1 se puede observar el esquema ordenado de la metodología antes mencionada, además de las especificaciones de que comprenden cada una de las seis fases.

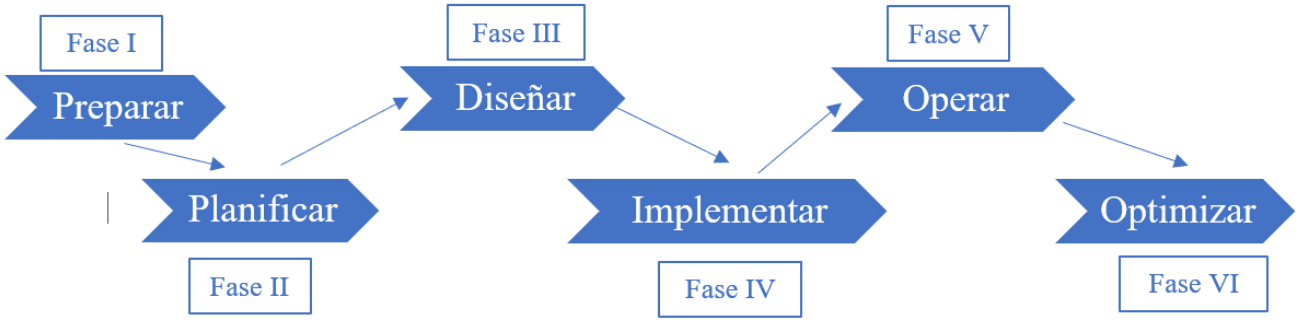


Ilustración 1: Fases de la metodología.

Fase I: Preparar.

En primera fase se efectuó el levantamiento de información de la infraestructura del NOC y se evidenciaron los componentes y servicios a ser supervisados mediante la plataforma de monitoreo.

Fase II: Planificar.

Dentro de esta fase se efectuó el levantamiento de información de las necesidades y requisitos del NOC para la implementación de la plataforma de monitoreo.

Fase III: Diseñar.

En esta fase se cumplió con la selección de la herramienta más conveniente para crear la infraestructura de red, mediante la comparación y evaluación de las herramientas de monitoreo Open Source más destacadas en la actualidad que se adecuen a las necesidades del NOC; además de tener en cuenta la opinión del personal administrativo de NOC sobre los resultados obtenidos. Así mismo, se realizará el diseño de la arquitectura lógica y física de la plataforma de monitoreo, basándose en los requisitos obtenidos durante la fase de planeación.

Fase IV: Implementar.

Dentro de esta fase se llevó a cabo los procedimientos de instalación, configuración y monitorización de los dispositivos que integran el sistema de monitoreo diseñado. De igual manera, se realizaron ensayos para comprobar el correcto funcionamiento de la solución implantada y validar el cumplimiento de los requisitos definidos por el personal administrativo del NOC y se creará el plan de continencia que permite recuperar la disponibilidad del sistema de monitoreo ante una falla.

Fase V: Operar.

En esta fase se procedió a poner en marcha el sistema de monitoreo, con el fin de verificar su rendimiento y así poder realizar futuras optimizaciones.

Fase VI: Optimizar.

Dentro de esta fase se realizó cambios en el sistema de monitoreo para mejorar el rendimiento de el mismo, según los requerimientos suscitados en la etapa de operacionalización de la plataforma de monitoreo.

3. CAPÍTULO 3

3.1. Propuesta de implementación

El presente apartado corresponde al capítulo, que abarca la propuesta de implementación del trabajo de fin de maestría. A partir de este apartado se procederá a explicar según las fases establecidas en la metodología, el proceso que conllevo el desarrollo e implementación la plataforma de monitoreo haciendo usos de software open source en la ISP. Im.Planet. S.A.

3.2. Fase 1: Preparación.

En primera fase se efectuó el levantamiento de información de la infraestructura del NOC y se evidenciaron los componentes y servicios a ser supervisados mediante la plataforma de monitoreo.

3.2.1. Centro de datos:

El centro de datos de In.Planet S.A., se encuentra segmentado en varios nodos, instalados en diferentes puntos de la ciudad de Milagro y los cantones aledaños, en el cual se ofrece cobertura, sin embargo en el presente trabajo de titulación nos centraremos en el DataCenter principal ubicado en la ciudad de Milagro, dirección; Malecón 312 y Federico Proaño en el edificio "HEY!", en dicho DataCenter se encuentran los equipos de *core* y críticos que el departamento de NOC requiere monitorizar.

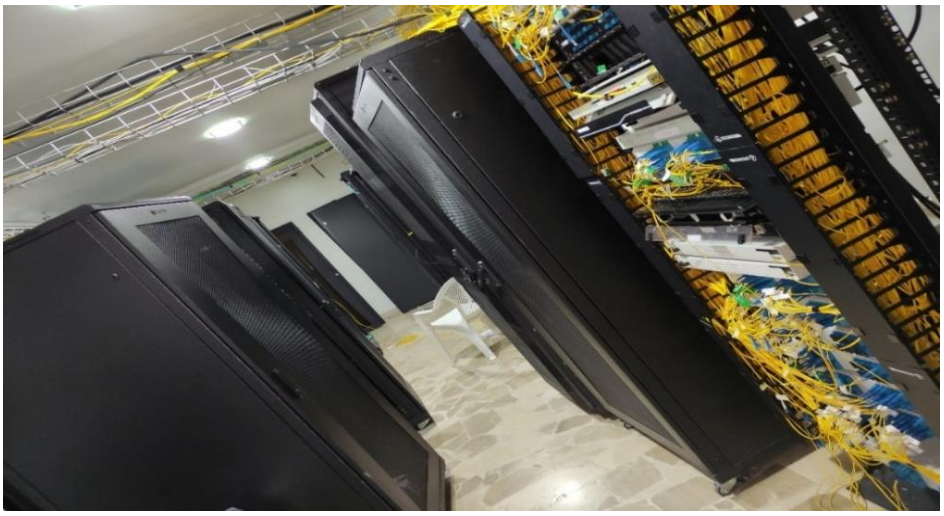


Ilustración 2: Centro de datos Milagro.

Ubicación	Edificio	Cantidad
Milagro	HEY!	1

Tabla 1: Centro de datos Milagro.

3.2.2. Enlace de comunicación:

La Central de datos de la ISP; In.Planet S. A., actualmente cuenta con solo un enlace de comunicación, este es sustancial para la comunicación entre la ISP y la Sede Central Milagro. En la siguiente tabla se detalla el *Enlace de Comunicación* existente. En la *tabla 2* se encuentran especificidades sobre el enlace de comunicación.

Enlace	Tipo	Capacidad
Telefónica Principal	Fibra óptica	40GB
Telefónica Backup	Fibra óptica	40 GB
Equinix	Fibra óptica	10GB

Tabla 2: Enlace de comunicación.

3.2.3. Equipos:

El departamento del NOC cuenta con dispositivos de red activos, recursos y sus respectivos servicios que son parte de la red de datos, los cuales requieren monitorización. Los equipos antes mencionados los podemos observar en mayor detalle en la tabla *3 y tabla 4.*

3.2.4. Dispositivos de la red:

En la tabla tres se puede observar los dispositivos activos que conforman el core, el cual está comprendido en tres vendor tecnológicos diferentes, estos equipos son los encargados de distribuir el servicio de internet a los demás nodos y suscriptores de In.Planet S.A.

Sede	Dispositivo	Marca	Cantidad
Milagro	Switch L3	Huawei	2
	Switch L3	Mikrotik	4
	Firewall	Fortinet	1
	Router	Mikrotik	9

Tabla 3: Dispositivos de red activos.

3.2.5. Recursos y servicios:

Dentro del equipamiento de red de la empresa, se cuenta con cierta cantidad de servidores, los cuales se encargan de alojar más servidores haciendo uso de tecnologías de virtualización, en otro caso se usa todo el hardware del servidor físico para ofrecer un servicio específico, a su vez un servicio crítico a monitorizar es el DNS, tal como se detalla en la tabla cuatro.

Tipo	Descripción	Marca	Cantidad
Recursos	Servidores	Supermicro	6
	Servidores	HP	1
	Servidores	Dell	2

Tipo	Descripción	Cantidad
Servicio	DNS	1

Tabla 4: Recursos y servicios de red.

Los contenidos que podemos encontrar en las *tablas 3 y 4* fueron recopilados mediante la realización de levantamiento de información haciendo uso de entrevistas al personal que comprende el departamento del NOC de la ISP Im.Planet. S. A.

3.2.6. Análisis económico:

Para el presente trabajo se usaron servidores que ya existían en el datacenter de In.Planet S.A. Sin embargo se realizó una cotización hacia el proveedor de servidores de la empresa para que, de esta manera tener un indicador económico de cuanto le cuesta o le hubiese costado a In.Planet S.A. Implementar ésta solución, que si bien es cierto usa software opensource y sin costo de licencia, si requiere equipos físicos para operar.

El proveedor nos hizo llegar la cotización, la misma que se encuentra en la sección de anexos.

Descripción	Cantidad	Precio unitario	Impuestos	Importe
Servidor				
HPE ProLiant DL180 Gen10 - Processor: Intel 4208 (2.1GHz/8-core/85W) - Memory: HPE 16GB 1Rx4 PC4-2933Y-R Smart Kit. - Network Controller: Embedded 2-Port 1GbE. - Storage Controller: P816i-a and Smart Storage Battery. - PCIe Slots: 3 PCIe: 1x8 FHFL, 1x8 FHHL, 1x8 FHHL. - Power Supply: 1x 500W Hot Plug; RPS ready. - Management: HPE iLO5, Infosight - Form Factor: 2U Rack.	1.00 Unidades	2,469.41	IVA 12%	$ 2,469.41

Warranty: 3-year parts, 3-year labor, 3-year onsite support with next business day response.				
Garantía Extendida, HPE 3 AÑOS TC Ess DL180 Gen10 SVC	1.00 Unidades	702.35	IVA 12%	$ 702.35
Memoria, HPE 16GB 2Rx8 PC4-2933Y-R Smart Kit	1.00 Unidades	297.65	IVA 12%	$ 297.65
Disco, HPE 1.2TB SAS 10K SFF SC DS HDD	2.00 Unidades	287.06	IVA 12%	$ 574.12
Fuente de Poder,HP 500W FS Plat Ht Plg Pwr Supply Kit	1.00 Unidades	154.12	IVA 12%	$ 154.12
			Subtotal	$ 4,197.65
			IVA 12%	$ 503.72
			Total	$ 4,701.37

Adicional al equipamiento, se usó tiempo del SysAdmin para la implementación de la solución, la cual en tiempo hombre fue:

Tiempo implementado	**Descripción de la actividad realizada**	**Encargado**
4 horas	Implementación LibreNMS y su configuración, así como la instalación y configuración de Influxdb	Leonardo Piña
3 horas	instalación e integración de Grafana, así como la creación de los dashboards.	Leonardo Piña
2 horas	implementación de prometheus y creación del dashboard	Leonardo Piña
4 horas	capacitación al personal del NOC en el uso de las herramientas	Leonardo Piña
Cantidad total de horas:		**13 horas**

Tabla 5: Tiempo del SysAdmin para la implementación de la solución.

En las *ilustraciones 3 hasta la 9*, se pueden observar los recursos mencionados en la *tabla 3;* Switch L3, Switch L3, Firewall, Router y en la *tabla 4, servidores*, exceptuando el apartado de servicios.

Ilustración 3: Switch L3 Huawei.

Ilustración 4: Switch L3 Mikrotik.

Ilustración 5: Firewall Fortinet.

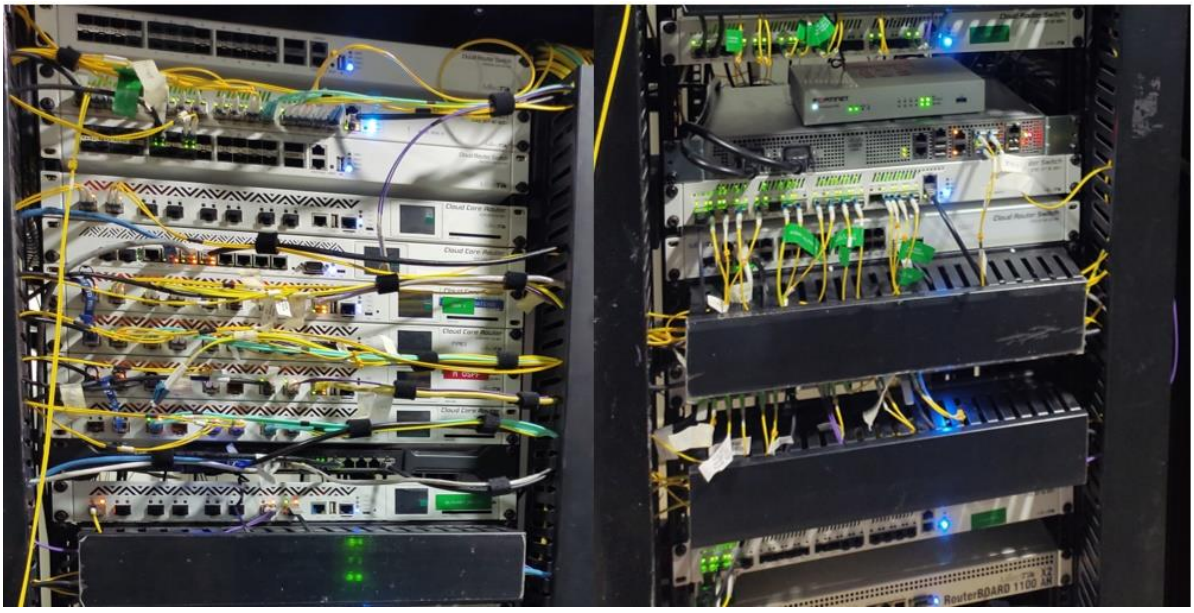


Ilustración 6: Routers Mikrootik.

Ilustración 7: Servidores Supermicro.

Ilustración 8: Servidores Dell.

Ilustración 9: Servidores HP.

3.3. Fase 2: Planificar:

En esta fase se efectuó el levantamiento de información de la infraestructura del NOC y se evidenciaron los componentes y servicios a ser supervisados mediante la plataforma de monitoreo.

3.3.1 Necesidades para cada equipo:

Como resultado del levantamiento de información en el departamento del NOC, obtuvimos los datos que se requieren monitorizar de los equipos; Switch, Firewall, Router, Servidores y DNS, a continuación, se puede observar de forma detallada las características que se solicitaron monitorizar.

Switch:

- **Generales:**
  - Nombre del dispositivo.
  - Sistema operativo.
  - Contacto del dispositivo.
  - Uptime.

- **CPU:**
  - Uso del procesador.

- **Memoria:**
  - Memoria total.
  - Memoria usada.
  - Memoria libre.

- **Uso de disco:**
  - Almacenamiento total.
  - Almacenamiento en uso.
  - Almacenamiento disponible.
- **Temperatura:**

    - Temperatura del procesador.
    - Temperatura de los módulos.

- **Ventiladores:**
    - Estado operativo.

- **Interfaces:**
    - Potencia interfaces SFP.
    - Consumo bits TX.
    - Consumo bits RX.
- **Histórico:**
    - Log de manipulación de equipo.

Firewall:

- **Generales:**
    - Nombre del dispositivo.
    - Sistema operativo.
    - Contacto del dispositivo.
    - Uptime.

- **CPU:**
    - Uso del procesador.

- **Memoria:**
    - Memoria total.
    - Memoria usada.
    - Memoria libre.

- **Uso de disco:**

    - Almacenamiento total.
    - Almacenamiento en uso.
    - Almacenamiento disponible.
- **Temperatura:**
    - Temperatura del procesador.
    - Temperatura de los módulos.

- **Ventiladores:**
    - Estado operativo.

- **Interfaces:**
    - Potencia interfaces SFP.
    - Consumo bits TX.
    - Consumo bits RX.
- **Histórico:**
    - Log de manipulación de equipo.

Router:

- **Generales:**
    - Nombre del dispositivo.
    - Sistema operativo.
    - Contacto del dispositivo.
    - Uptime.

- **CPU:**
    - Uso del procesador.

- **Memoria:**
    - Memoria total.
    - Memoria en uso.

    - Memoria libre.

- **Uso de disco:**
    - Almacenamiento total.
    - Almacenamiento en uso.
    - Almacenamiento disponible.
- **Temperatura:**
    - Temperatura del procesador.
    - Temperatura de los módulos.

- **Ventiladores:**
    - Estado operativo.

- **Interfaces:**
    - Potencia interfaces SFP.
    - Consumo bits TX.
    - Consumo bits RX.
- **Histórico:**
    - Log de manipulación de equipo.

Servidores:

- **Generales:**
    - Nombre del dispositivo.
    - Sistema operativo.
    - Contacto del dispositivo.
    - Uptime.

- **CPU:**
    - Uso del procesador.

- **Memoria:**
    - Memoria total.
    - Memoria usada.
    - Memoria libre.

- **Uso de disco:**
    - Almacenamiento total.
    - Almacenamiento en uso.
    - Almacenamiento libre.

- **Interfaces:**
    - Consumo bits TX.
    - Consumo bits RX.

DNS.

- **Consultas:**
    - Consultas totales al DNS.
    - Cantidad de consultas por tipo.

3.3.2. Especificidades de requerimientos solicitados:

A continuación, se presentan de manera específica las especificidades requeridas para la implementación de la plataforma de monitoreo. En la tabla cinco reúne la información reunida, en el cual se detalla los requerimientos que el NOC requirió para el software NMS que se implementó.

Requerimiento	Descripción	Prioridad
Monitoreo de equipos:	El software NMS, permitirá el monitoreo de cualquier dispositivo de red que soporte el protocolo SNMP y el cual disponga de una IP de administración dentro de la red.	Alta
Parámetros de monitoreo:	El software NMS deberá permitir el monitoreo de parámetros tales como CPU, memoria, red, disco, entre otros.	Alta
Gestión de host:	El software NMS debe permitir la administración (Eliminación y creación) de los hosts mediante un panel de administración.	Alta
Soporte SNMP:	El Software NMS deberá contar con soporte para el protocolo de red SNMP.	Alta
Alertas:	La herramienta NMS deberá permitir la configuración de alertas a través de las cuales notificará el estado de los dispositivos.	Alta
Niveles de Alertas:	La herramienta NMS deberá permitir varios tipos de alertas, sean estos de Warning, Informatión, Security entre otros.	Alta
Gráficos:	El software NMS debe permitir la generación de gráficos de todos los parámetros de monitoreo.	Alta
Notificaciones	El software NMS, deberá enviar notificaciones con respectos a las alertas configuradas, sea vía mensajería o correo electrónico.	Alta

Tabla 6: Especificidades de requerimientos solicitados.

3.3.3. Especificidades de requerimientos no funcionales:

Los requisitos no funcionales son restricciones de las funciones ofrecidas por el sistema.

Requerimiento	Descripción	Prioridad
Disponibilidad:	El software NMS deberá tener continua disponibilidad, las veces que el usuario intente ingresar.	Alta
Interfaz:	El software NMS deberá contar con una interfaz web para su uso y administración.	Alta
Seguridad:	El software NMS deberá permitir el acceso mediante un usuario y contraseñas definidos por el administrador.	Alta
Seguridad:	El software NMS deberá permitir el manejo de perfiles de usuario donde los de tipos Admin tienen acceso a todas las funciones Administrativas.	Alta
Implementación:	El software NMS así como sus complementos deberán se OpenSource o FreeSoftware	Alta

Tabla 7: Especificidades de requerimientos no funcionales

3.3.4. Disponibilidad de entorno:

En la tabla 7 se describe los requerimientos del servidor, entre ellos, la cantidad de RAM, capacidad de almacenamiento, y versión de sistema operativo a instalar.

Sistema Operativo	CPU	Disco Duro	RAM	Descripción
Ubuntu Server 22.04 LTS	3Ghz	80GB	8GB	Sistema NMS para monitoreo SNMP.
Ubuntu Server 22.04 LTS	3Ghz	80GB	4GB	Prometheus.
Ubuntu Server 22.04 LTS	3Ghz	80GB	8GB	Dashboard Grafana.

Tabla 8: Disponibilidad de entorno.

3.4. Fase 3. Diseñar:

En esta fase se cumplió con la selección de la herramienta más conveniente para crear la infraestructura de red, mediante la comparación y evaluación de las herramientas de monitoreo Open Source más destacadas en la actualidad que se adecuen a las necesidades del NOC; además de tener en cuenta la opinión del personal administrativo de NOC sobre los resultados obtenidos. Así mismo, se realizará el diseño de la arquitectura lógica y física de la plataforma de monitoreo, basándose en los requisitos obtenidos durante la fase de planeación.

3.4.1. Determinación de la herramienta de monitoreo:

Existe una gran gama de herramientas de monitoreo Open Source que cumple con la función de monitorización de infraestructura de red. Por este motivo se procedió a la realización de una tabla comparativa de las herramientas de monitoreo de mayor eficiencia, misma que se adecuen a las necesidades que se deben cubrir.

3.4.2. Comparación:

En este apartado se definió los parámetros que se tuvo en consideración para la selección de los softwares que se utilizarán en la plataforma de monitoreo. Las especificidades se podrán observar en la *"Tabla 8: Comparación de herramientas de monitoreo de red open source".*

- **Licencia:** Tipo de licencia.
- **Almacenamiento:** Sistema de base de datos.
- **Aplicación web:** Permitir el acceso web al sistema, desde cualquier dispositivo conectado a la red.
- **SNMP:** Debe soportar la recolección de datos vía el protocolo de red.
- **Gráficas:** Debe permitir generar graficas de la información recolectada.
- **Alertas:** Enviar mensajes vía mensajería instantáneo y correo electrónico.
- **Eventos:** Debe permitir visualizar comportamientos atípicos e informar de aquello.
- **Mapa de red:** Permite ver un esquema de la red de una manera gráfica.

3.4.3. Comparación de los softwares de monitoreo de red Open Source:

La tabla ocho, muestra que se realizó un check a cada una de las herramientas que cumplan con ciertos parámetros establecidos por el NOC, de igual manera de especificó el tipo de licencia a la cual se rigen y que tipo de BD utilizan para su funcionamiento.

Comparación de herramientas de monitoreo de red Open Source:

Nombre	Licencia	Almacenamiento	Aplicación web	SNMP	Graficas	Alertas	Eventos	Mapa de red
Cacti	GNU GPL	RRD, MySQL	✓	✓	✓	✓	✓	✓
Nagios	GPLv2	RRD, MySQL	✓	✓	✓	✓	✓	✓
Zabbix	GPLv2	RRD, MySQL, PostgreSQL	✓	✓	✓	✓	✓	✓
Grafana	Apache 2.0	La presente aplicación no presenta almacena-miento.	✓	**X**	✓	**X**	**X**	**X**
LibreNMS	GPLv3	RRD, MySQL	✓	✓	✓	✓	✓	✓
Prometheus	Apache 2.0	LevelDB	✓	**X**	**X**	**X**	**X**	**X**

Tabla 9: Comparación de herramientas de monitoreo de red Open Source.

A partir de la comparación de herramientas de monitoreo de red Open Source, pudimos identificar cuatro herramientas (Cacti, Nagios, Zabbix y Libre NMS) con los parámetros requeridos para la implementación de la plataforma de monitoreo. Una vez realizada selección de estas cuatro herramientas que cumplen con los requerimientos solicitados por el NOC, se procedió al siguiente paso que corresponde a la "instalación piloto", la cual ayudará a una nueva comparativa entre las cuatro herramientas ya antes mencionadas para que, de esta forma elegir a la herramienta más versátil de mejor rendimiento. La comparativa de las cuatro herramientas se las puede observar en la *Tabla 10*

3.4.4. Instalación piloto:

A partir de la preselección de los softwares de NMS (Network Management System), se procedió a realizar una prueba piloto para medir su eficiencia según las necedades del NOC. Para ello se utilizó el *hypervisor* Esxi 7 desarrollado por VMware Inc, el cual nos permite la virtualización de sistemas operativos teniendo un amplio abanico de sistemas operativos compatibles

3.4.5. Escala de equivalencias:

Carácter Cuantitativo	1	2	3	4	5
Carácter Cualitativo	Insuficiente	Regular	Bueno	Muy bueno	Excelente

Tabla 10: Escala equivalente.

En base la escala equivalente que se muestra en la *tabla 9*, se procedió a realizar la selección de la herramienta que se usó en la plataforma de monitoreo, dicha comparativa la pueden observar en la *tabla 10*. A continuación en la *ilustración 10* se puede ver instalado el software Cacti, de igual forma Nagios en la *ilustración 11*, Zabbix en la *ilustración 12 hasta la 17*, y por último tenemos a LibreNMS que va desde la ilustración 18 hasta la 23.

Cacti:

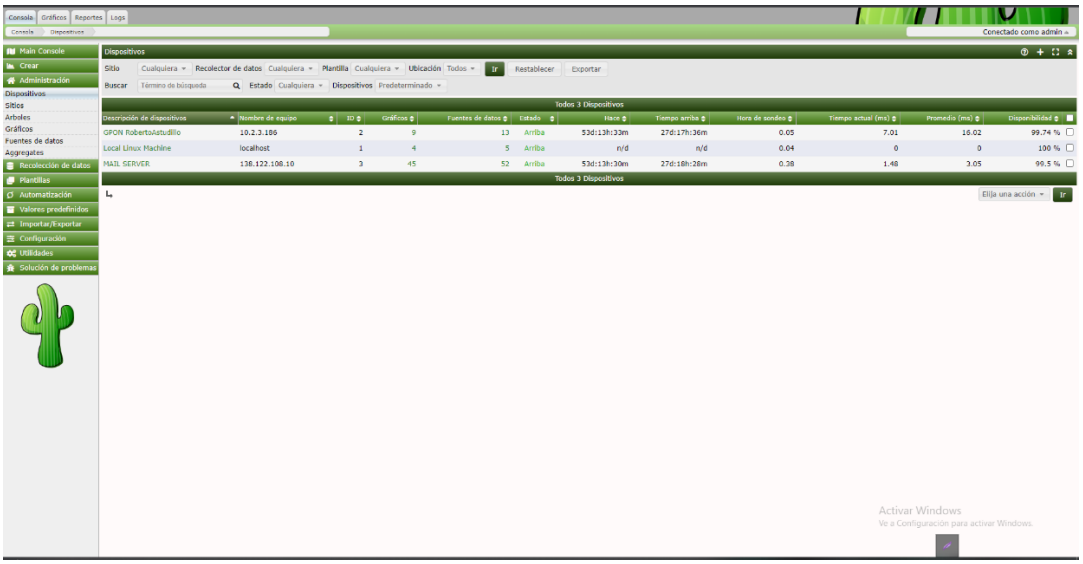


Ilustración 10: Sitio web inicial de Cacti.

Nagios:

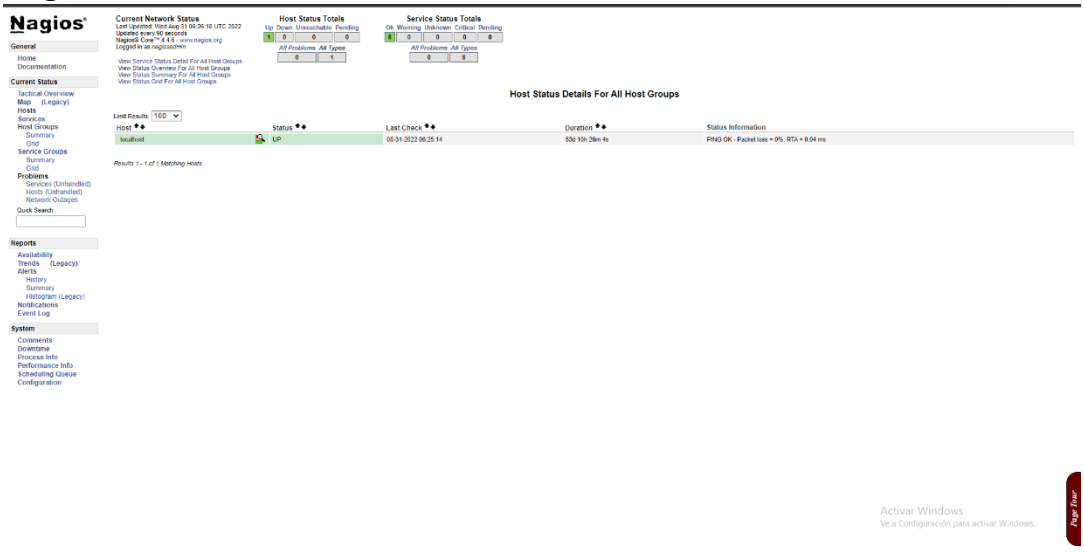


Ilustración 11: Sitio web inicial de Nagios.

Zabbix:

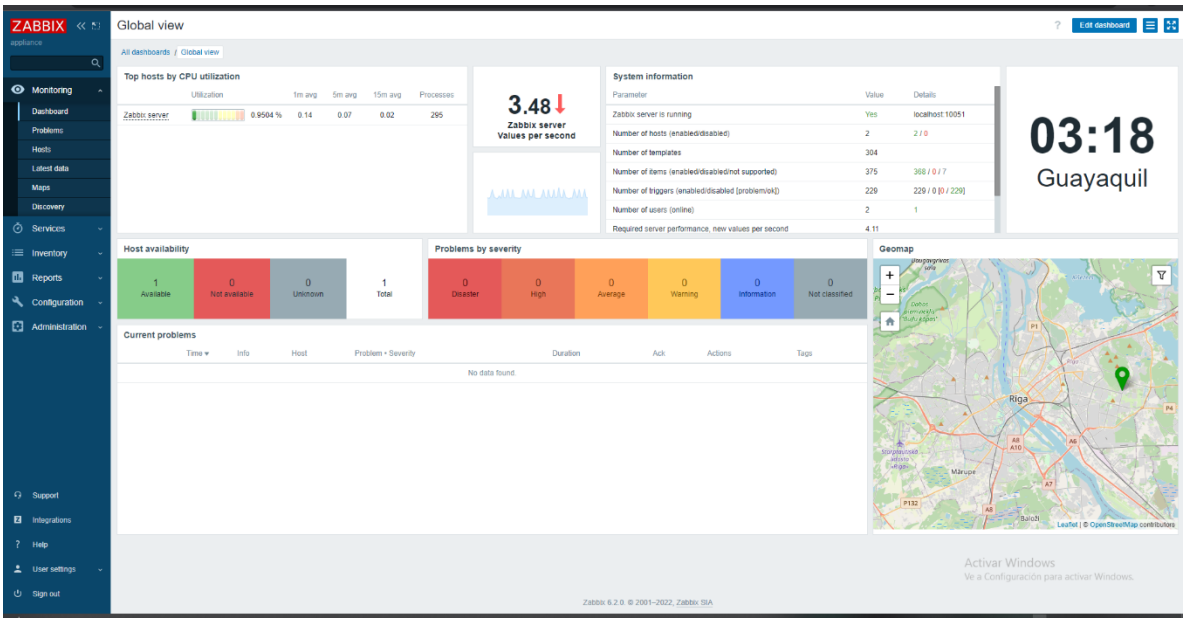


Ilustración 12: Sitio web inicial de Zabbix.

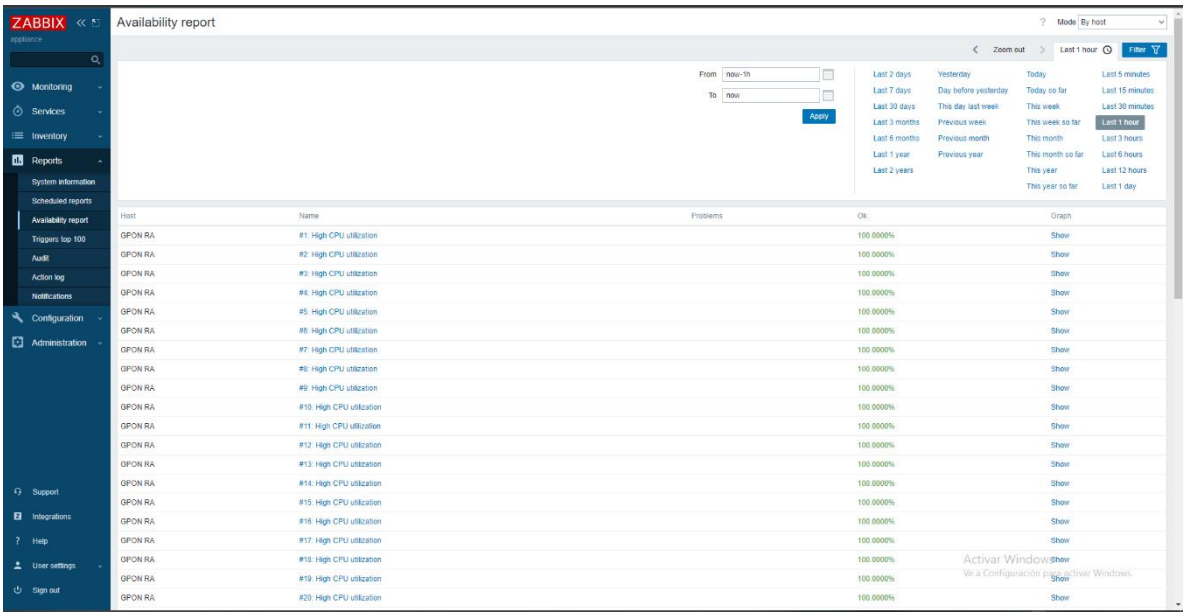


Ilustración 13: Reportes.

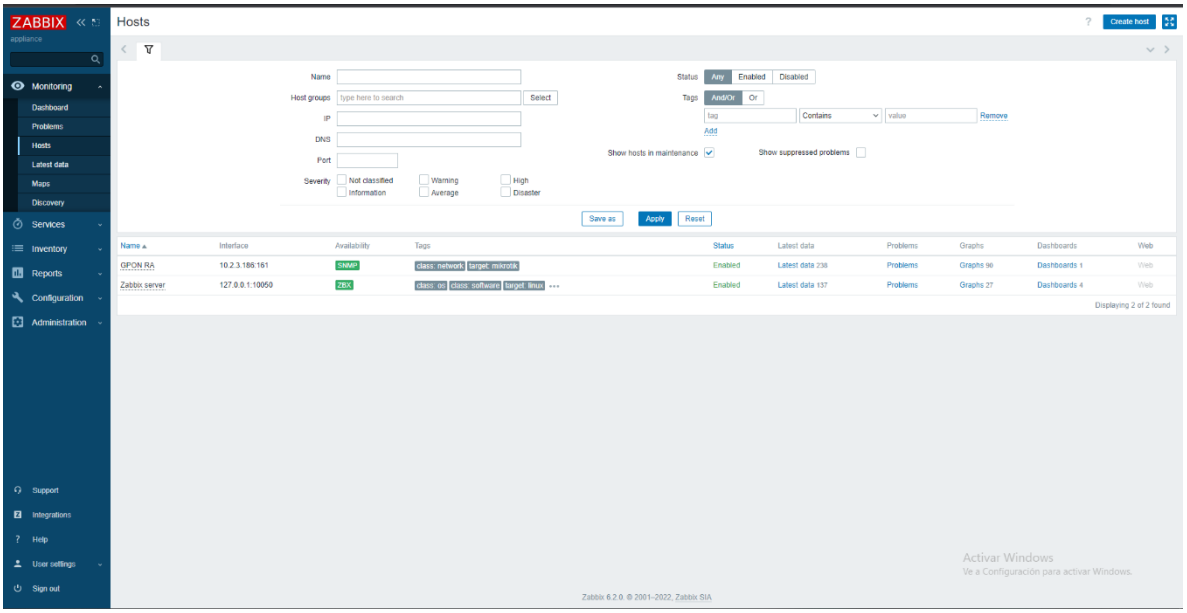


Ilustración 14: Pantalla de Hosts

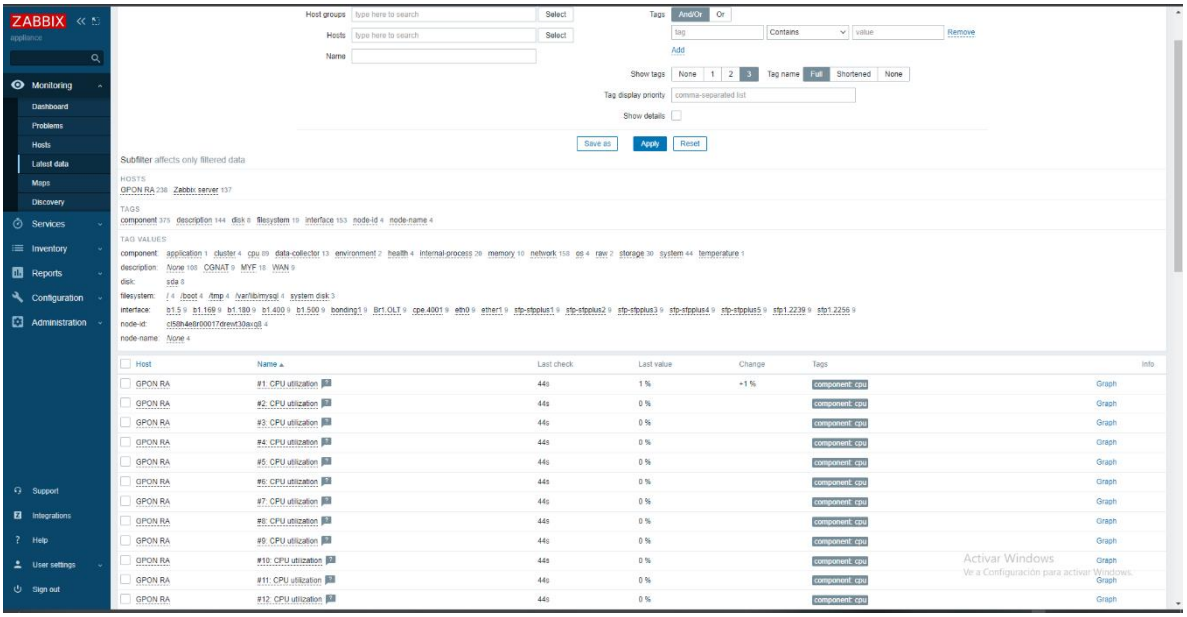


Ilustración 15: Información histórica.

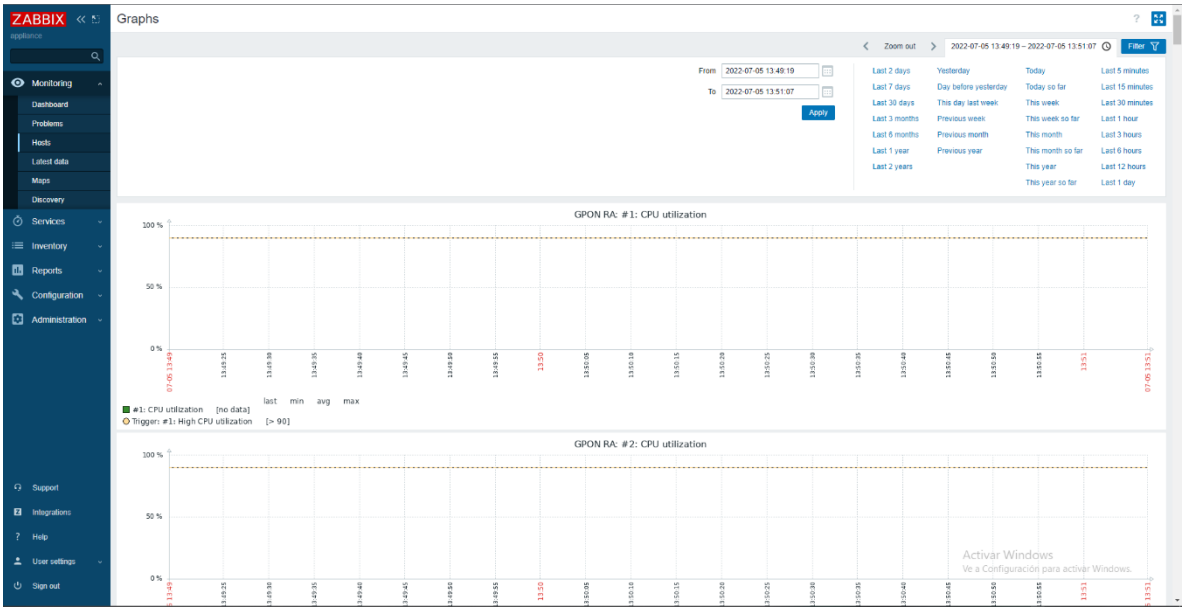


Ilustración 16: Gráficos de monitoreo CPU.

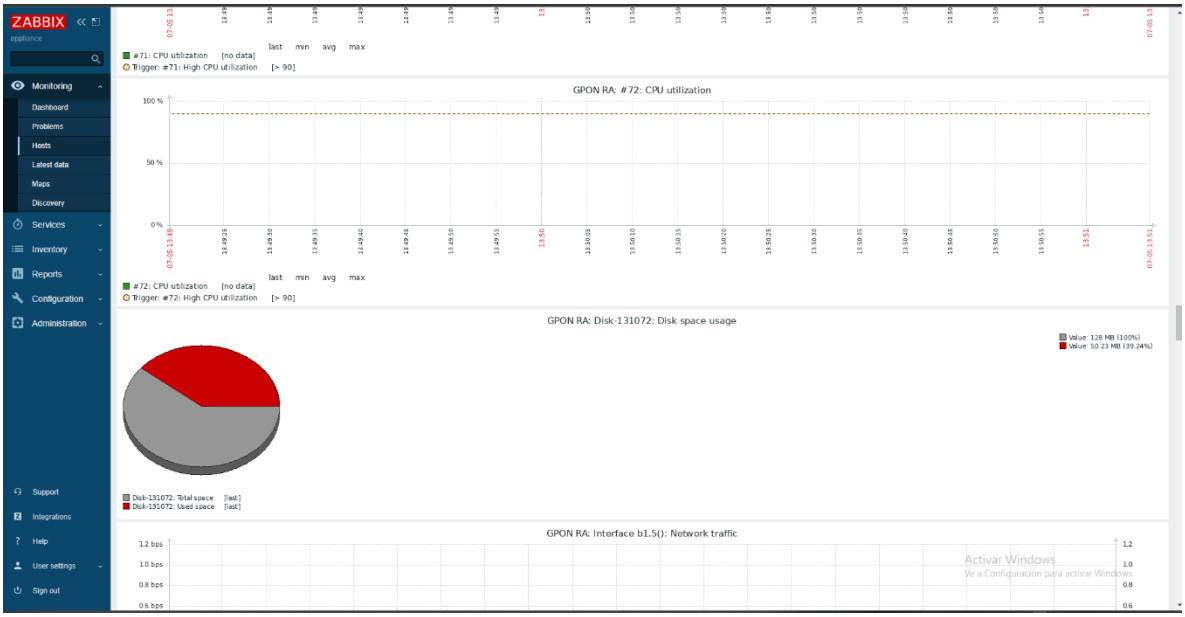


Ilustración 17: Gráficos de monitoreo del disco duro.

Libre NMS:

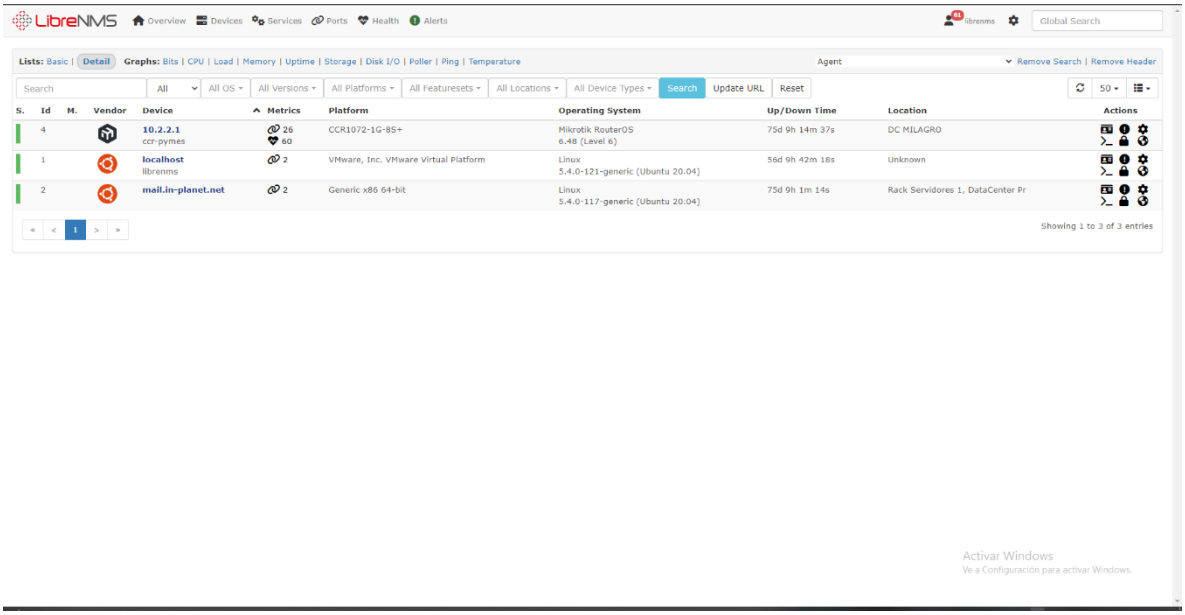


Ilustración 18: Pantalla del Hosts.

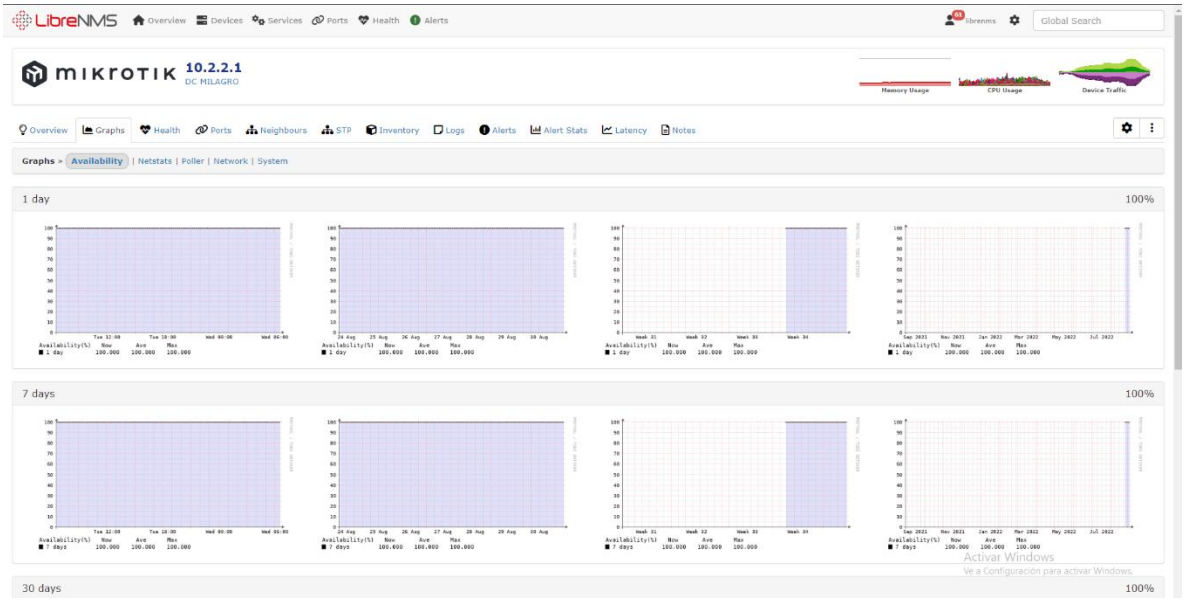


Ilustración 19: Gráficos generales.

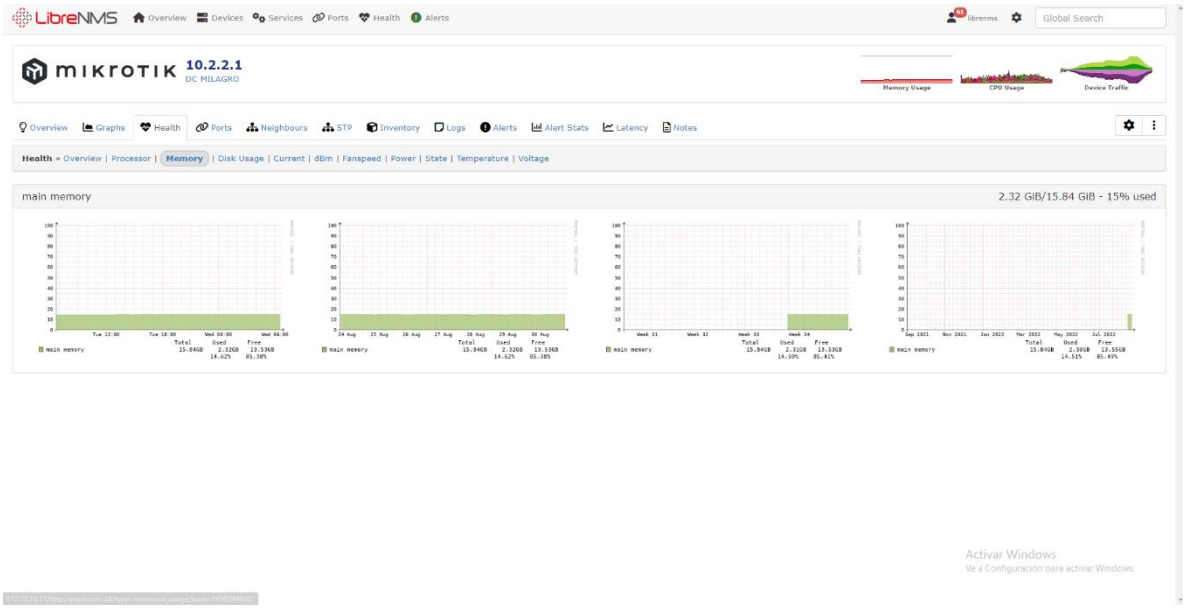


Ilustración 20: Gráficos de memoria.

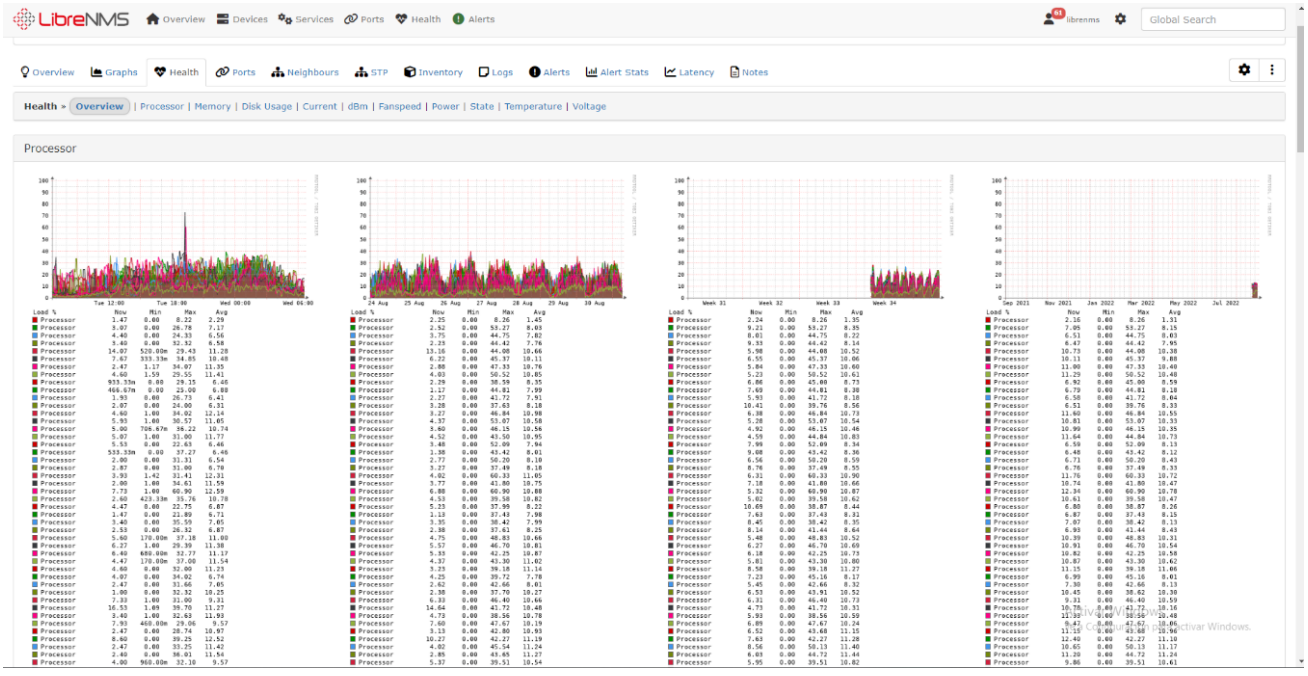


Ilustración 21: Gráficos de CPU

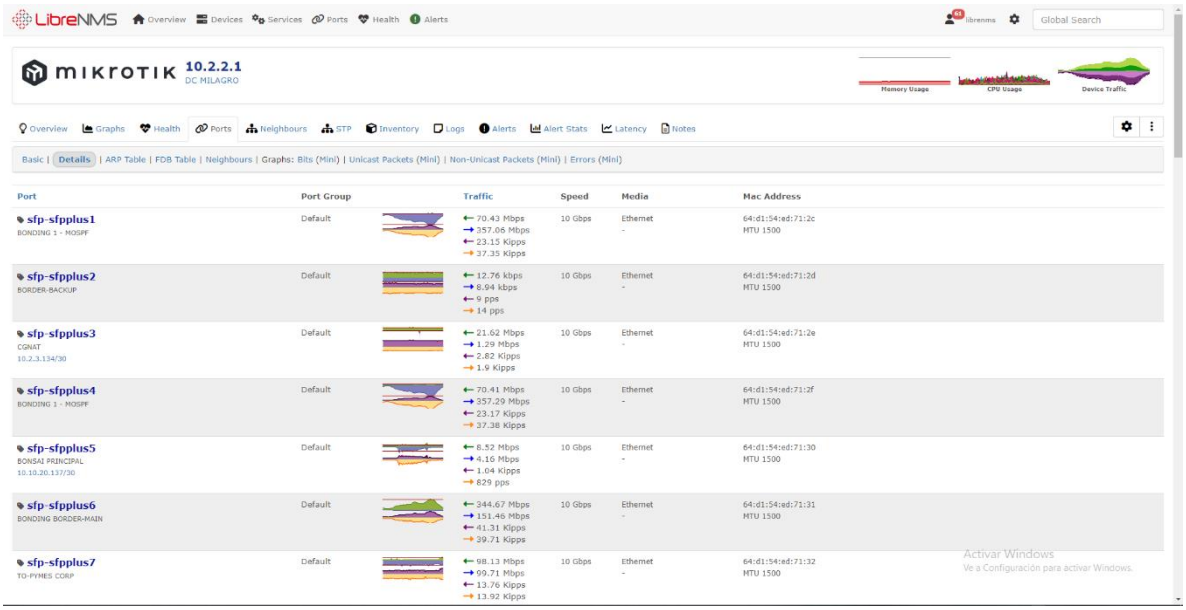


Ilustración 22: Gráfico de interfaces.

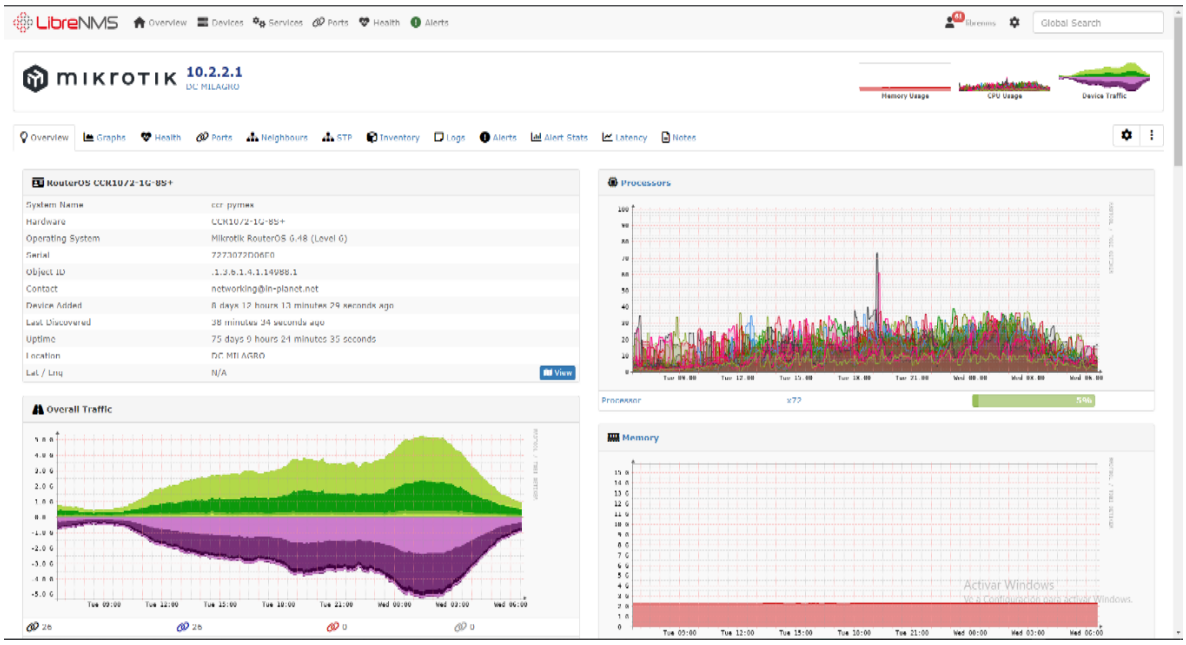


Ilustración 23: Dispositivo.

Una vez realizada la *instalación piloto*, se procedió a evaluar las cuatro herramientas (Cacti, Nagios, Zabbix, Libre NMS) que cumplen con los requerimientos solicitados; los cuales se pueden verificar en la *tabla 8.*

3.4.6. Evaluación de distintos rangos:

En la *tabla 8* se realizó una comparativa entre cuatro herramientas NMS, en el escalafón de uno a cinco, especificado en la *tabla 9*, en la cual 1 corresponde a la calificación más baja y cinco el puntaje más alto. La ponderación final tendrá un máximo de 45 puntos, ya que, se comparó siete tópicos principales indicados por el NOC, en consecuencia, al análisis realizado, se le otorgó la evaluación respectiva.

Parámetros	Cacti	Nagios	Zabbix	Libre NMS
Aplicación web	3	3	4	5
SNMP	5	5	5	5
Graficas	3	3	4	5
Plantillas	3	2	5	5
Alertas	3	2	5	5
Eventos	3	3	4	4
Curva de aprendizaje	3	2	4	5
Total	27	23	37	41

Tabla 11: Evaluación de distintos rangos.

En vista de la evaluación realizada sobre los distintos rasgos para definir cuál herramienta de las cuatro antes preseleccionadas, sería elegida para la implementación de la plataforma de monitoreo. Siendo que se elaboró siete ítems de evaluación y una puntuación máxima de 5, la evaluación tuvo un puntaje de 45 puntos.

Las herramientas que obtuvieron los puntajes más aproximados al establecido fueron, Zabbix con 37 puntos y LibreNMS 41 puntos, por ello en la tabla 11 se realizó la comparativa

para elegir de forma definitiva la herramienta que se usará para la implementación de la plataforma de monitoreo.

3.4.7. Aspectos relevantes en la evaluación de las herramientas de monitoreo:

En la tabla, se comparó a las dos herramientas mejor puntadas de la comparativa anterior, siendo Zabbix y LibreNMS las que mejor puntaje obtuvieron respecto a los requerimientos del NOC.

Aspectos	**Zabbix**	**LibreNMS**
Instalación y configuración inicial	Instalación y configuración de Zabbix es un poco compleja aun siguiendo los pasos de instalación, muchas veces da error con los paquetes de idiomas y se debe hacer configuraciones adicionales, dispone de una VM de appliance donde se puede poner en marcha con solo cambiar la IP, sin embargo, está limitada al espacio en disco configurado.	Instalación y configuración inicial es sencilla, siguiendo los pasos de la wiki oficial se puede dejar operativo el software LibreNMS sin necesidad de realizar troubleshooting adicional, dispone de una VM de appliance donde se puede poner en marcha con solo cambiar la IP, sin embargo, está limitada al espacio en disco configurado.
Configuración	La configuración de cualquier parámetro adicional fuera del setup de instalación se la realiza mediante la plataforma web del software	La configuración de cualquier parámetro adicional fuera del setup de instalación se la realiza mediante la plataforma web del software

Aplicación web	La plataforma web es bastante completa sin embargo para un usuario nuevo es complicado navegar entre las opciones, desde está plataforma se puede obtener gráficos, reportes, revisión de eventos, etc.	La plataforma web es bastante completa sumamente sencilla navegar entre sus opciones, desde está plataforma se puede obtener gráficos, reportes, revisión de eventos, etc.
SNMP	Compatibilidad completa con el protocolo	Compatibilidad completa con el protocolo
Graficas	Zabbix genera sus gráficas haciendo uso de rrd.	LibreNMS genera sus gráficas haciendo uso de rrd.
Plantillas	Zabbix dispone de varias plantillas para la generación de alertas, gráficos etc., que son compatibles con los diversos vendors	Librenms dispone de varias plantillas para la generación de alertas, gráficos etc., que son compatibles con los diversos vendors
Alertas y notificaciones	Zabbix dispone de varios niveles de alertas, las notificaciones pueden ser enviadas por correo o servicio de mensajería instantánea compatibles	LibreNMS tiene un número de alertas predeterminadas que pueden utilizarse fácilmente, las notificaciones pueden ser enviadas por correo o servicio de mensajería instantánea compatibles
Dashboard	Solo nos muestra un dasboard donde se encuentran todos los dispositivos.	Tiene la opción de crear múltiples dashboard y configurar a gusto del usuario que información desea visualizar.

Curva de aprendizaje	El aprendizaje tanto para el administrador y el usuario final es demasiado extensa debido, debido a las múltiples configuraciones posteriores a la instalación, citando un ejemplo para añadir un dispositivo se deben realizar varios pasos para poder añadir al monitoreo SNMP.	Aprendizaje relativamente sencillo, no requiere mucha configuración adicional, wiki sumamente explicativa, para añadir un disipativo a monitoreo solo hace falta la IP, el puerto SNMP y la comunidad SNMP.

Tabla 12: Aspectos relevantes en la evaluación de las herramientas de monitoreo.

Sobre la base de los requerimientos solicitados para la implementación de la plataforma de monitoreo, basándonos en los resultados de la evaluación realizada y en conjunto con el equipo del departamento NOC, se eligió el software Libre NMS, ya que, cumple de forma eficiente con todos los parámetros indicados para la monitorización de la infraestructura de red de datos del core de la ISP In.Planet. S.A. Milagro.

3.4.8. Arquitectura:

Posteriormente a la selección de la herramienta de monitoreo Libre NMS se procedió a crear el diseño de la arquitectura, la cual comprende la siguiente estructura: está compuesta por dos servidores, LibreNMS y Grafana. El software LibreNMS cumplirá con la función de recibir información mediante el agente SNMP configurado en los dispositivos de red del core a monitorizar, mientras que Grafana se alimentará de los datos recolectados por LibreNMS para la creación de lo dashboards.

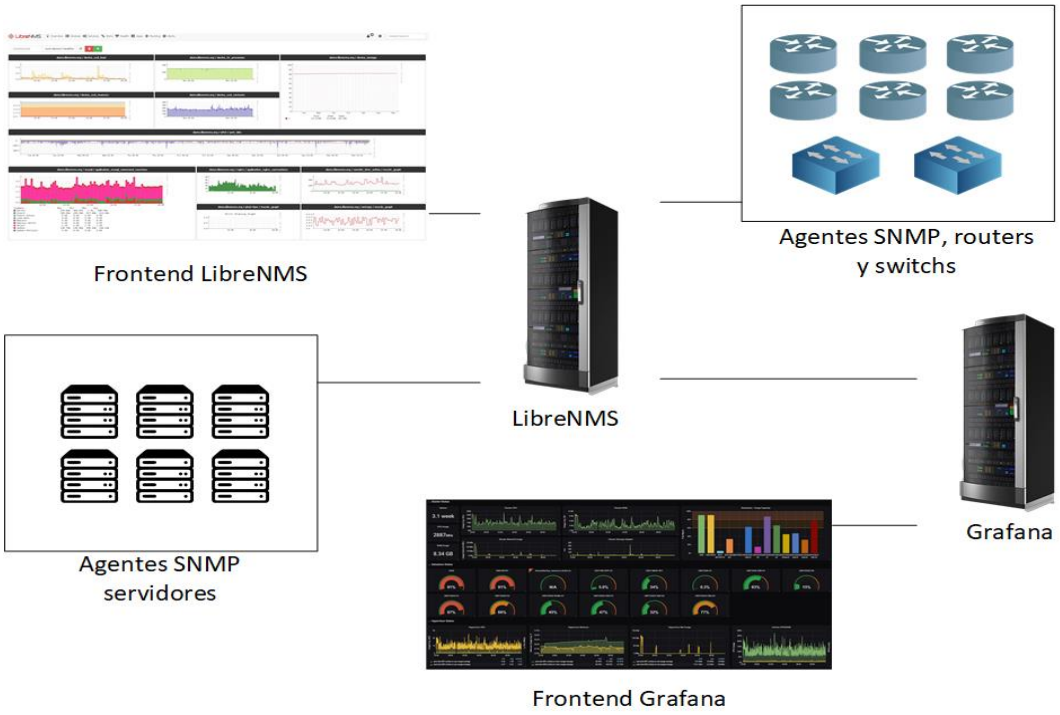


Ilustración 24: Arquitectura.

3.5. Fase IV: Implementar:

Dentro de esta fase se llevó a cabo los procedimientos de instalación, configuración y monitorización de los dispositivos que integran el sistema de monitoreo diseñado. De igual manera, se realizaron ensayos para comprobar el correcto funcionamiento de la solución implantada y validar el cumplimiento de los requisitos definidos por el personal administrativo del NOC y se creará el plan de continencia que permite recuperar la disponibilidad del sistema de monitoreo ante una falla.

3.5.1. Requisitos para instalación del software LibreNMS:

El software Open Source, LibreNMS, requiere ciertos requisitos mínimos como sistema operativo, base de datos, PHP y servidor web, para operar de forma correcta. *La tabla 12* especifica a profundidad las características de los requerimientos necesarios para implementar LibreNMS.

Software	Plataforma	Versión
Sistema Operativo	Ubuntu	20.04
	Ubuntu	22.04
	Centos	7
	Debian	10
	Debian	11
Base de datos	MySQL	5.7 o posterior
	MariaDB	10.0 o posterior
PHP	-	7.2 o posterior
Servidor web	Apache	1.3 o posterior
	Nginx	1.10 o posterior

Tabla 13: Requisitos de software.

Se debe tener en cuenta el prerrequisito para la instalación de LibreNMS, previamente en la terminal de nuestro servidor que contendrá al Software, ejecutaremos los siguientes comandos con los cuales completaron la instalación del software LibreNMS, el cual también nos permite recolectar la información necesaria mediante el uso de los agentes SNMP. A continuación, se puede observar los comandos utilizados en las terminales del servidor:

3.5.2. Instalación PHP y librerías necesarias:

```
root@librenms01:~# apt-get install wget php php-pear php-cgi php-common php-curl php-mbstring php-gd php-mysql php-bcmath php-imap php-json php-xm
l php-snmp php-fpm php-zip -y
```

Ilustración 25: Instalación PHP y librerías necesarias.

3.5.3. Configuración de zona horaria:

```
root@librenms01:~# nano /etc/php/8.1/fpm/php.ini
root@librenms01:~# nano /etc/php/8.1/cli/php.ini
```

Ilustración 26: Configuración de zona horaria.

Cambiamos date.timezone por nuestra zona horaria:

[Date]
date.timezone = America/Guayaquil

Reiniciamos el servicio FPM:

root@librenms01:~# systemctl restart php8.1-fpm

3.5.4. Instalación servidor web nginx:

Primero desinstalamos apache que se instaló con los primeros paquetes de PHP:

root@librenms01:~# apt-get remove apache2 -y

Instalamos nginx

root@librenms01:~# apt-get -y install nginx

Comprobamos que el servicio está activo:

root@librenms01:~# systemctl status nginx

Activamos el inicio al arraque del OS

root@librenms01:~# systemctl enable nginx

3.5.5. Instalación de MariaDB:

```
root@librenms01:~# apt-get remove apache2 -y
```

Ilustración 27:Instalación de MariaDB.

Modificamos el archivo de configuración:

```
root@librenms01:~# nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Ilustración 28: Modificamos el archivo de configuración.

Y añadiremos lo siguiente en el apartado [mysqld]:

```
# this is only for the mysqld standalone daemon
[mysqld]
innodb_file_per_table=1
sql-mode=""
lower_case_table_names=0
```

Ilustración 29: Agregamos comando en el apartado [mysqld].

Reiniciamos el servicio de la BD:

root@librenms01:~# systemctl restart mariadb

Creamos la BD, el usuario y le asignamos los permisos necesarios:

```
MariaDB [(none)]> CREATE DATABASE librenms CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
REATE USER 'liQuery OK, 1 row affected (0.001 sec)

MariaDB [(none)]> brenms'@CREATE USER 'librenms'@'localhost' IDENTIFIED BY ' Cl@v3S3gUr@';
ANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
FLUSH PRIVILEGES;
exitQuery OK, 0 rows affected (0.057 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
Query OK, 0 rows affected (0.045 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
```

Ilustración 30:Creamos la BD, el usuario y le asignamos los permisos necesarios.

Posteriormente a la incorporación de los comandos en la terminal del servidor que contendrá al software LibreNMS, procedemos a la instalación del software open source, la consta de los siguientes pasos:

3.5.6. Instalación de LibreNMS:

root@librenms01:~# apt-get install git -y

Creamos un usuario para LibreNMS:

```
root@librenms01:~# useradd -r -M -d /opt/librenms librenms
root@librenms01:~# getent passwd librenms
librenms:x:998:998::/opt/librenms:/bin/bash
root@librenms01:~# usermod -a -G librenms www-data
```

Ilustración 31: Crear ususario para LibreNMS.

Instalamos los paquetes necesarios para el funcionamiento de LibreNMS:

```
root@librenms01:~# apt-get install rrdtool whois fping imagemagick graphviz  mtr-tiny nmap python3 python3-pip python3-mysqldb snmp snmpd python3-me
mcache mtr-tiny composer acl unzip python3-pymysql python3-dotenv python3-redis python3-setuptools python3-systemd -y
```

Ilustración 32: Instalaciónde los paquetes necesarios para el funcionamiento de LibreNMS.

Clonamos el repositorio de git de LibreNMS que contiene la última versión estable:

```
root@librenms01:~# git clone https://github.com/librenms/librenms.git /root/librenms
```

Ilustración 33: Clonación del repositorio de git de LibreNMS.

Movemos la carpeta clonada de librenms al directorio /opt:

root@librenms01:~# mv /root/librenms/ /opt/

Copiamos la configuración del template por defecto de SNMP a un nuevo template:

```
root@librenms01:~# cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
```

Ilustración 34:Copiamos la configuración del template por defecto de SNMP a un nuevo template.

Modificamos la comunidad SNMP por defecto RANDOMSTRINGGOESHER por la que se vaya a utilizar:

```
# Change RANDOMSTRINGGOESHERE to your preferred SNMP community string
com2sec readonly  default         inet_snmp

group MyROGroup v2c        readonly
view all    included  .1                                80
access MyROGroup ""      any       noauth    exact  all    none   none

syslocation Servidores, DC Matriz, Milagro
syscontact Leonardo Pina <lpina@in-planet.net>

#OS Distribution Detection
extend distro /usr/bin/distro
```

Ilustración 35: Modificación del archive snmpd.conf.

Instalamos el plugin para que detecte el logo del vendor del dispositivo:

```
root@librenms01:~# curl -o /root/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5017  100  5017    0     0  12100      0 --:--:-- --:--:-- --:--:-- 12118
root@librenms01:~# chmod 775 /root/distro
root@librenms01:~# mv /root/distro  /usr/bin/distro
```

Ilustración 36: Instalación del plugin distro.

Reiniciamos el servicio SNMP:

root@librenms01:~# systemctl restart snmpd

Realizamos copia del archivo www.conf:

```
root@librenms01:~# cp /etc/php/8.1/fpm/pool.d/www.conf /etc/php/*/fpm/pool.d/librenms.conf
```

Ilustración 37: Copia del archivo de configuración de ejemplo al archivo de configuración para la web de LibreNMS.

Editamos el archivo librenms.conf:

```
root@librenms01:~# nano /etc/php/8.1/fpm/pool.d/librenms.conf
```

Ilustración 38: Comando para editar el archivo.

Cambiamos [www] por [librenms]

Cambiamos el usuario y el grupo por “librenms”:

user = librenms
group = librenms

Cambiamos el contenido del parametro listen:

```
;listen = /run/php/php8.1-fpm.sock
listen = /run/php-fpm-librenms.sock
```

Ilustración 39: Conectar línea original y sea añadió una nueva.

Ejecutamos el siguiente comando para configurar el cron:

```
root@librenms01:~# cp /opt/librenms/librenms.nonroot.cron /etc/cron.d/librenms
```

Ilustración 40:Copia del archivo de cron del directorio de instalación al directorio de cron del sistema operativo.

LibreNMS mantiene registros de eventos en el archivo /opt/librenms/logs. Al transcurrir el tiempo, estos tienden volverse muy grandes y pesados. Para realizar un guardado histórico de los registros antiguos, se usó logrotate:

```
root@librenms01:~# cp /opt/librenms/misc/librenms.logrotate /etc/logrotate.d/librenms
```

Ilustración 41: Copia del archivo de cron del directorio de instalación al directorio logrorate del sistema operativo.

Finalmente, corregimos todos los permisos:

```
root@librenms01:~# chown -R librenms:librenms /opt/librenms
root@librenms01:~# setfacl -d -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache/ /opt/librenms/storage/
root@librenms01:~# setfacl -R -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache/ /opt/librenms/storage/
```

Ilustración 42: Comandos para corregir permisos.

Ejecutar la instalación composer:

```
root@librenms01:~# su - librenms
librenms@librenms01:~$ ./scripts/composer_wrapper.php install --no-dev
librenms@librenms01:~$ exit
logout
root@librenms01:~#
```

Ilustración 43: Ejecución de comandos para instalación de composer y sus complementos.

Habilitamos el comando lnms:

```
root@librenms01:~# ln -s /opt/librenms/lnms /usr/bin/lnms
root@librenms01:~# cp /opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
```

Ilustración 44: Creación del link simbólico y copia del directorio de instalación al directorio de bash.

3.5.7. Configuramos nginx:

Creamos un nuevo archivo:

```
 GNU nano 6.2                                  /etc/nginx/conf.d/librenms.conf *
server {
 server_name nms.in-planet.net;
 root        /opt/librenms/html;
 index       index.php;
 listen 80;

 charset utf-8;
 gzip on;
 gzip_types text/css application/javascript text/javascript application/x-javascript image/svg+xml text/plain text/xsd text/xsl text/xml image/x-ic>
 location / {
  try_files $uri $uri/ /index.php?$query_string;
 }
 location /api/v0 {
  try_files $uri $uri/ /api_v0.php?$query_string;
 }
 location ~ [^/]\.php(/|$) {
  fastcgi_pass unix:/run/php-fpm-librenms.sock;
  fastcgi_split_path_info ^(.+\.php)(/.+)$;
  include fastcgi.conf;
 }
 location ~ /\.(?!well-known).* {
  deny all;
 }
}
```

Ilustración 45: Creación y configuración del archivo para el funcionamiento de la web de LibreNMS.

Comprobamos la sintaxis del archivo:

root@librenms01:~# nginx -t

Si todo está bien, reiniciamos nginx

```
root@librenms01:~# rm /etc/nginx/sites-enabled/default
root@librenms01:~# systemctl restart nginx
root@librenms01:~# systemctl restart php8.1-fpm
```

Ilustración 46: Eliminación del archivo por efecto y reinicio de servicios.

3.5.8. Instalación y configuración mediante el navegador web:

http://monitoreo.in-planet.net/install.php

Luego de instalar los pre requisitos y el backend de LibreNMS procedimos a configurar el frontend donde se comprobó los requisitos instalados, y se configuró la conexión a la base datos, así mismo se creó el usuario principal del software NMS. El proceso de instalación de la "parte gráfica" y las configuraciones correspondientes, lo pueden verificar desde la *ilustración 25 hasta la 64.*

3.5.9. Instalación de la parte gráfica del software LibreNMS:

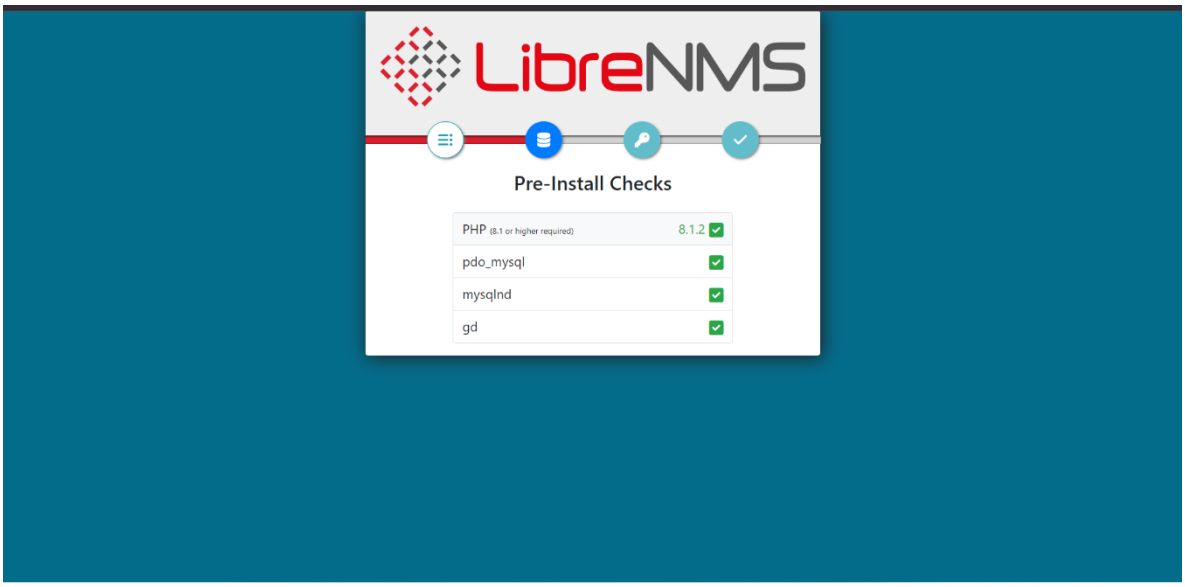


Ilustración 47: El Setup comprueba la versión de PHP y las librerías más importantes.

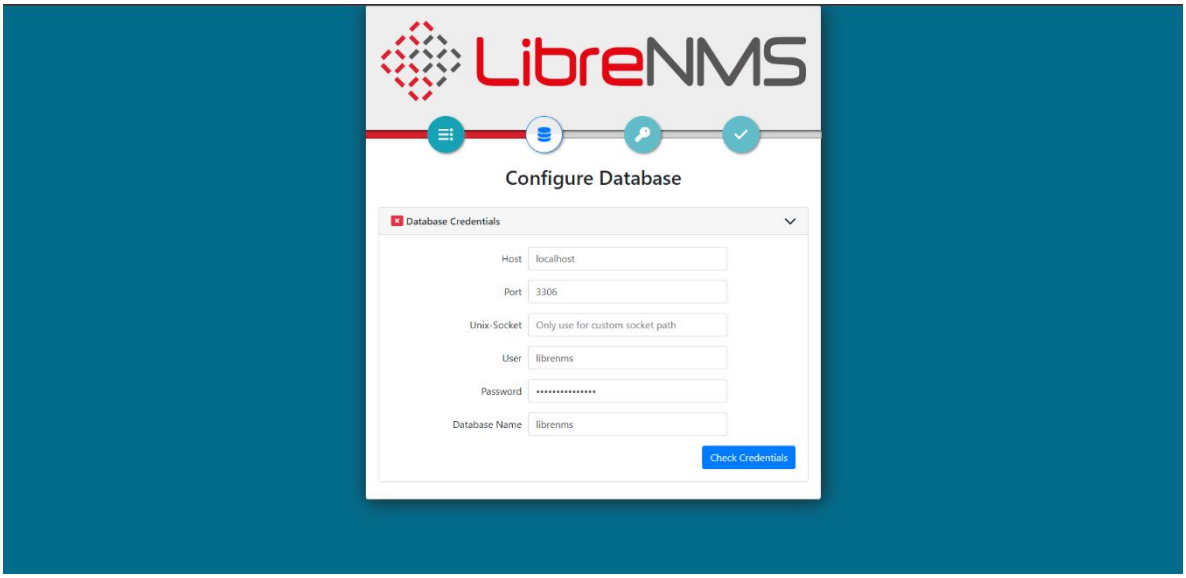


Ilustración 48: Se configura la BD, debemos indicar la clave que se configuro mediante terminal en MySQL.

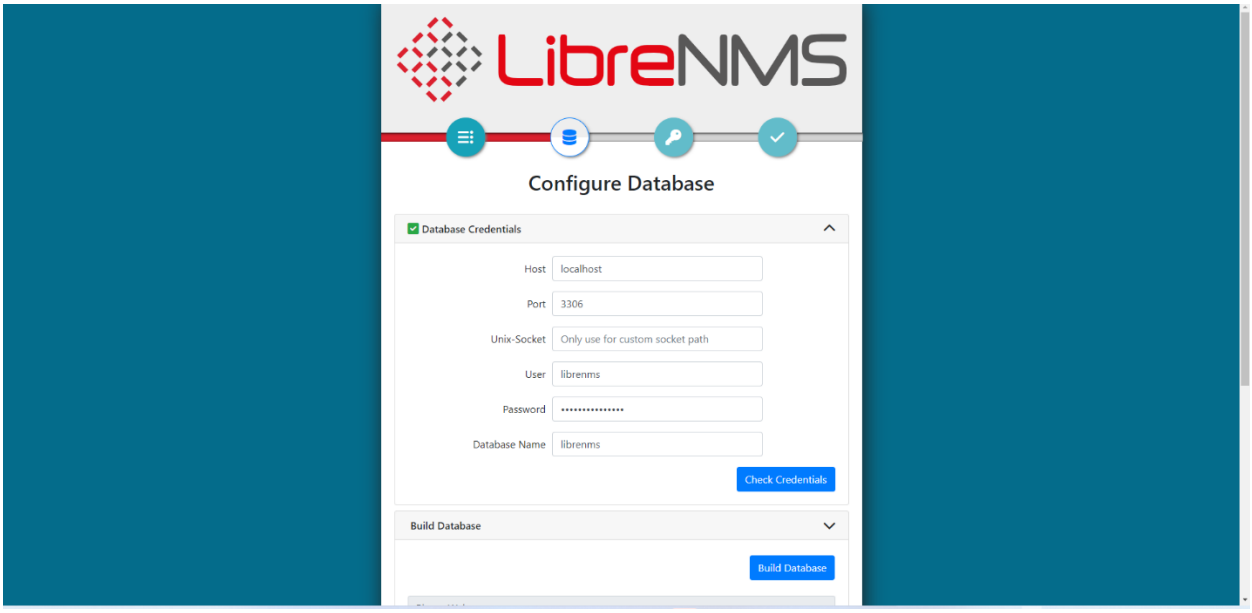


Ilustración 49: Presionamos en Check Credentials si la clave es correcta saldrá un visto color verde.

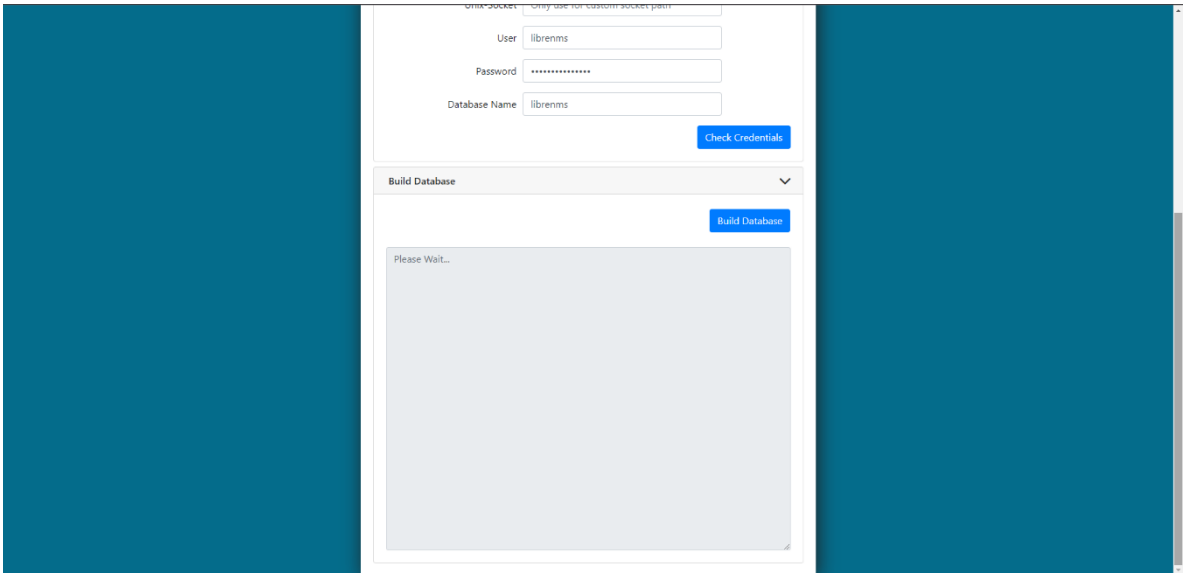


Ilustración 50: Presionamos en Build Database para crear las tablas y campos necesarios.

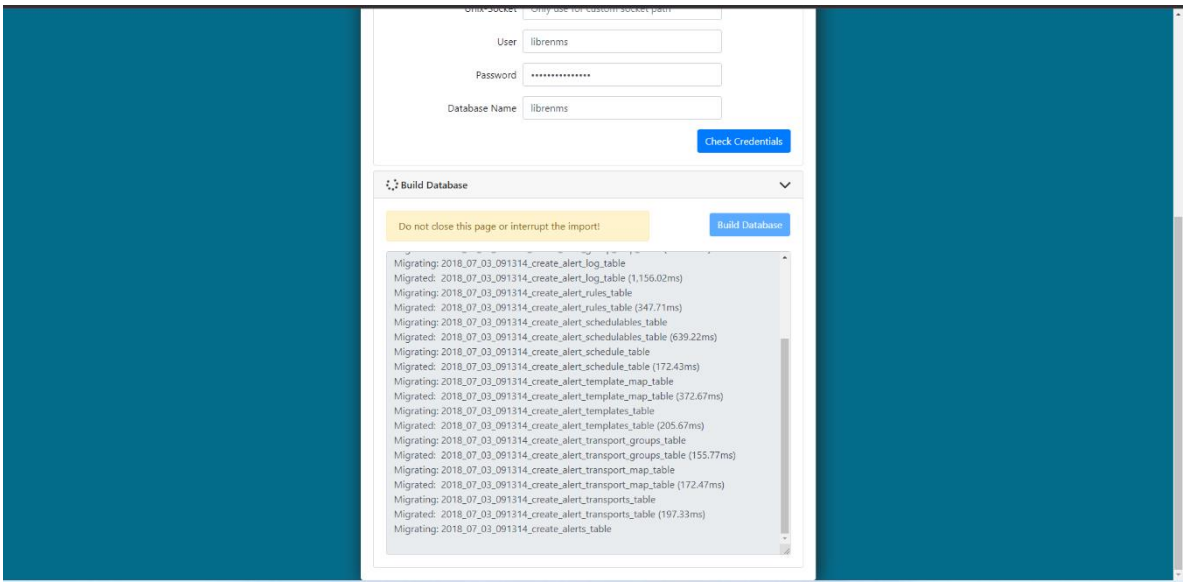


Ilustración 51: Se crean tablas y campos en la BD.

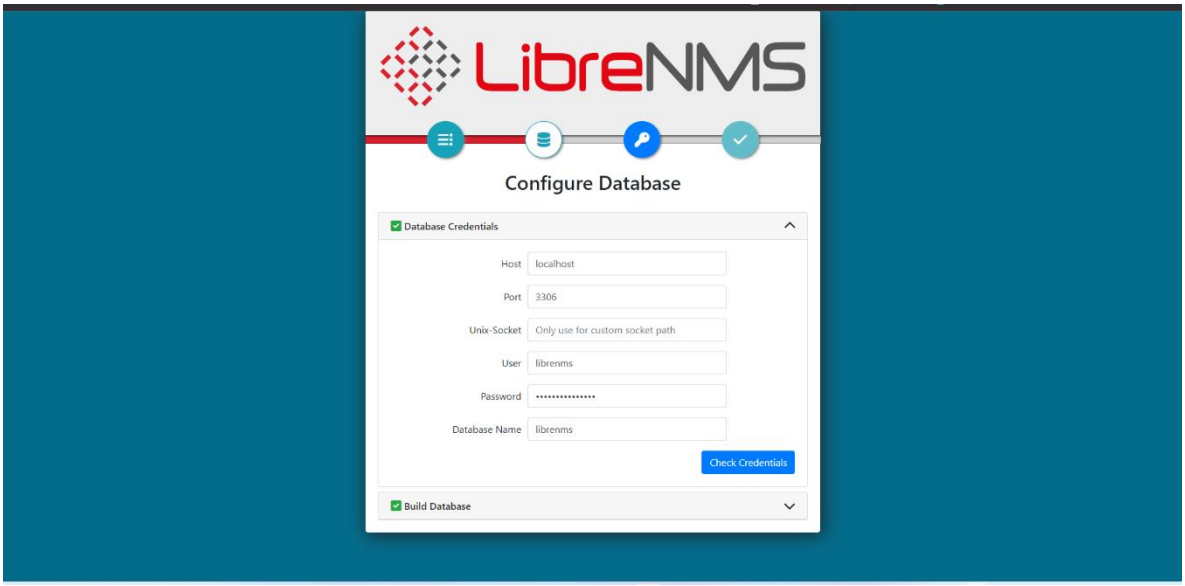


Ilustración 52: Cuando concluya la tarea saldrá un visto color verde

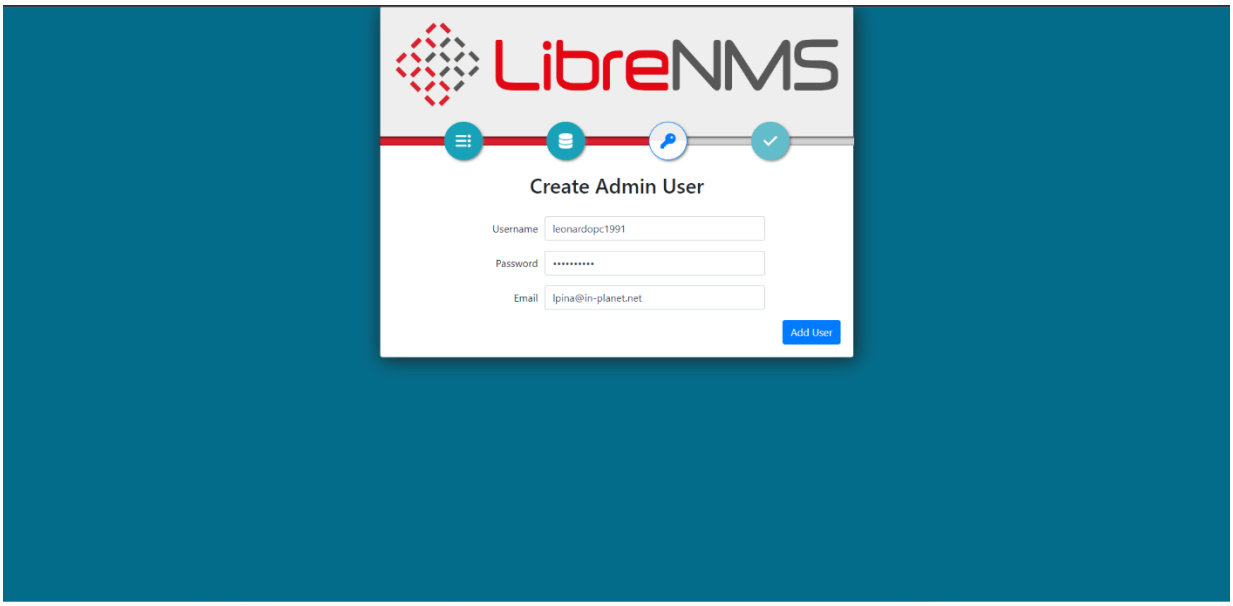


Ilustración 53: Se configura el usuario administrador, debemos ingresar el nombre de usuario clave y correo.

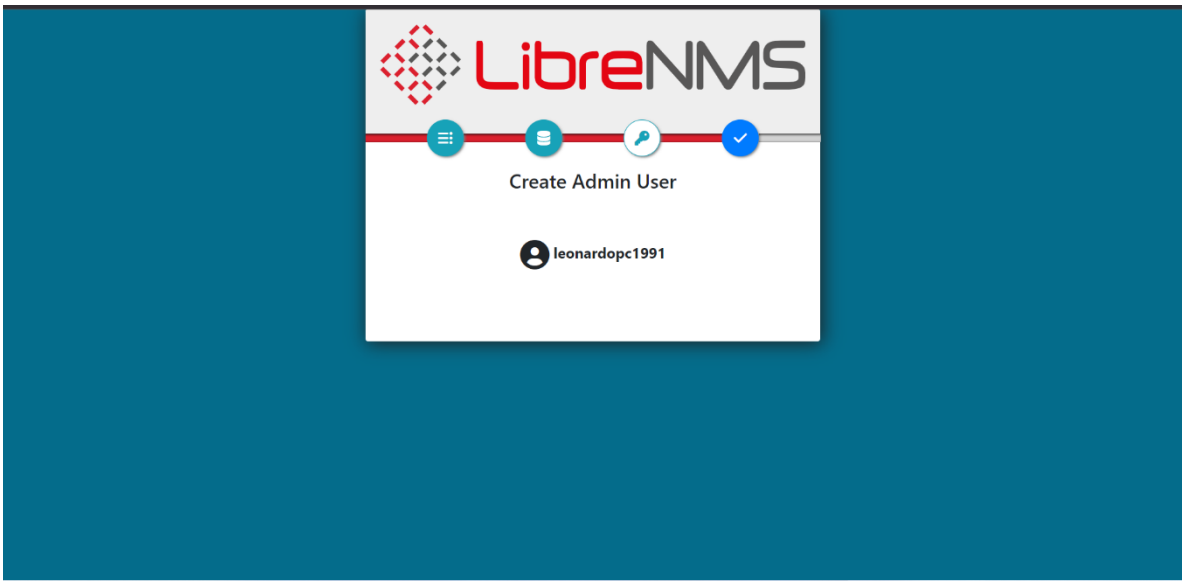


Ilustración 54: Usuario creado correctamente.

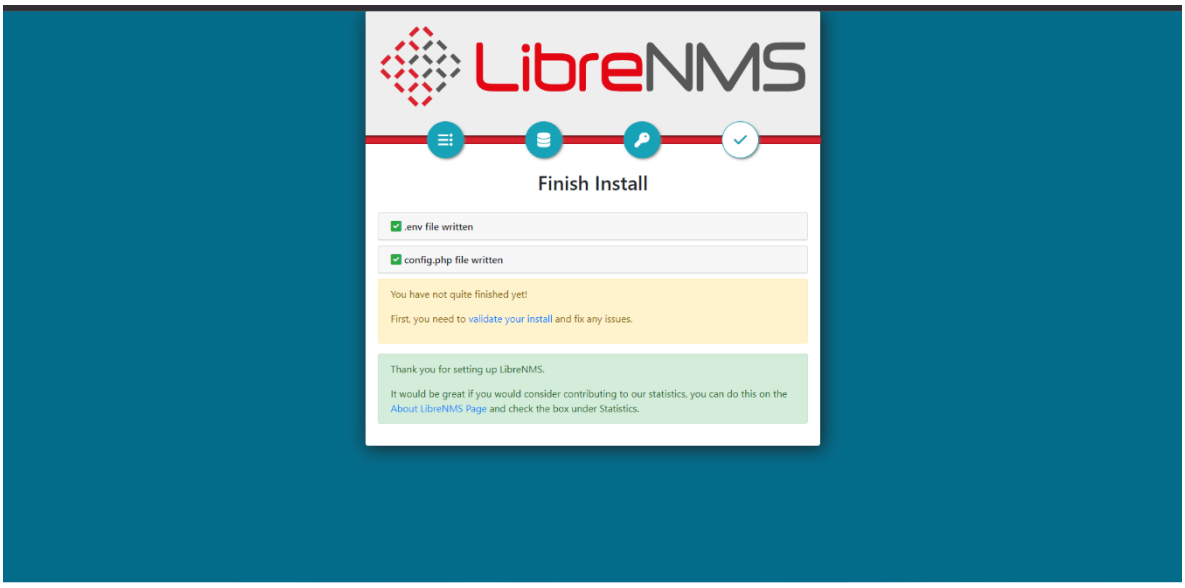


Ilustración 55: Una vez completado, presionaremos en validate your install.

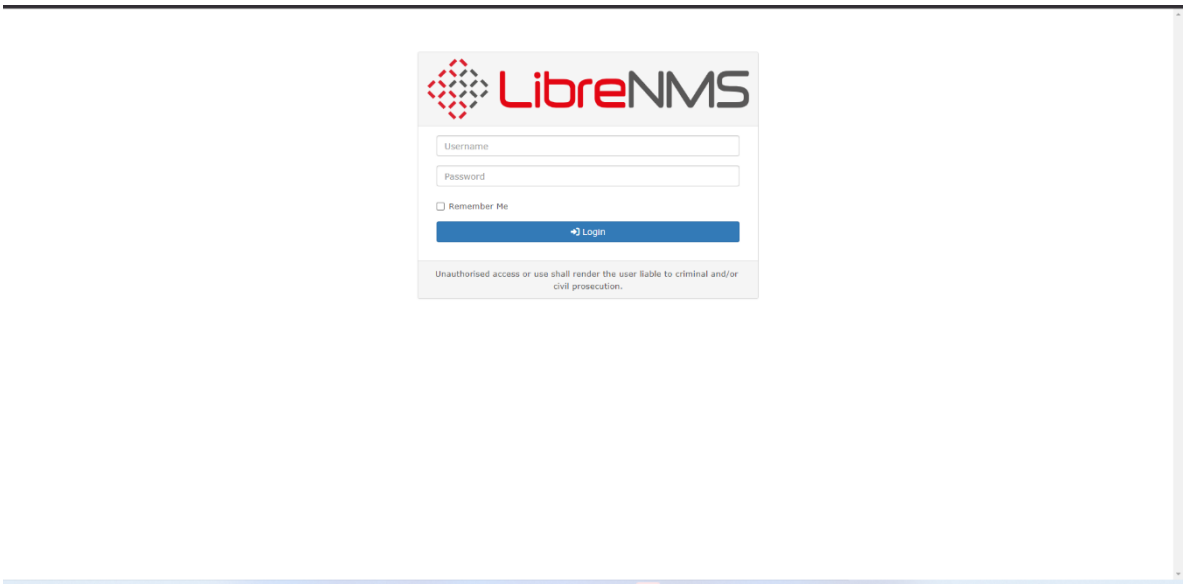


Ilustración 56: Nos llevara a la ventana de login donde ingresaremos los datos anteriormente configurados.

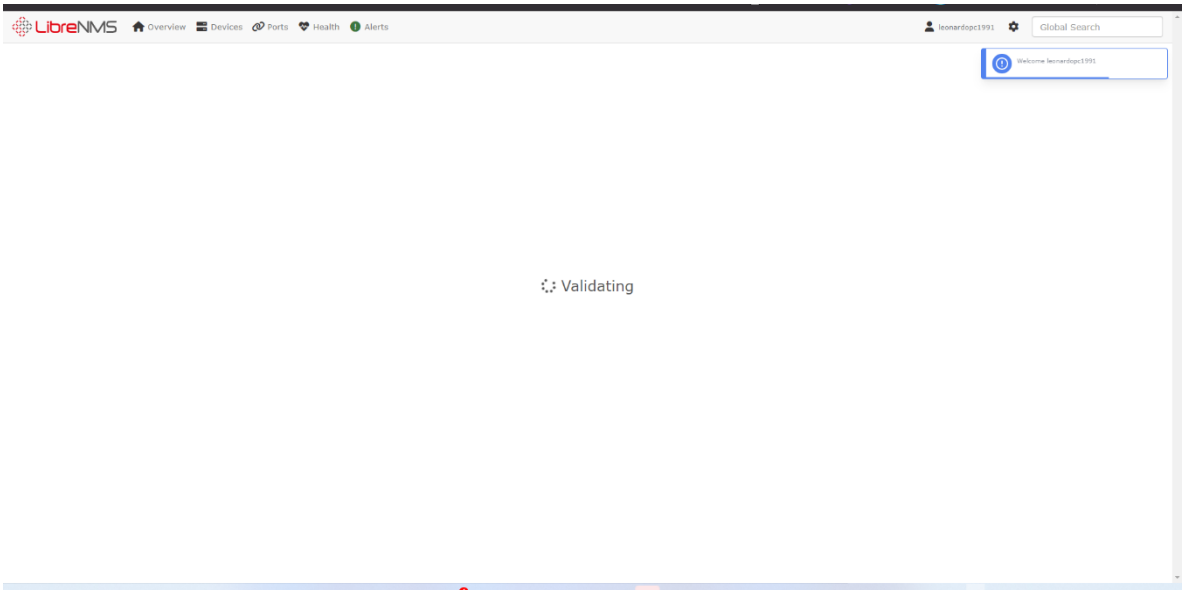


Ilustración 57: Comenzará con la validación de todas las configuración y librerías.

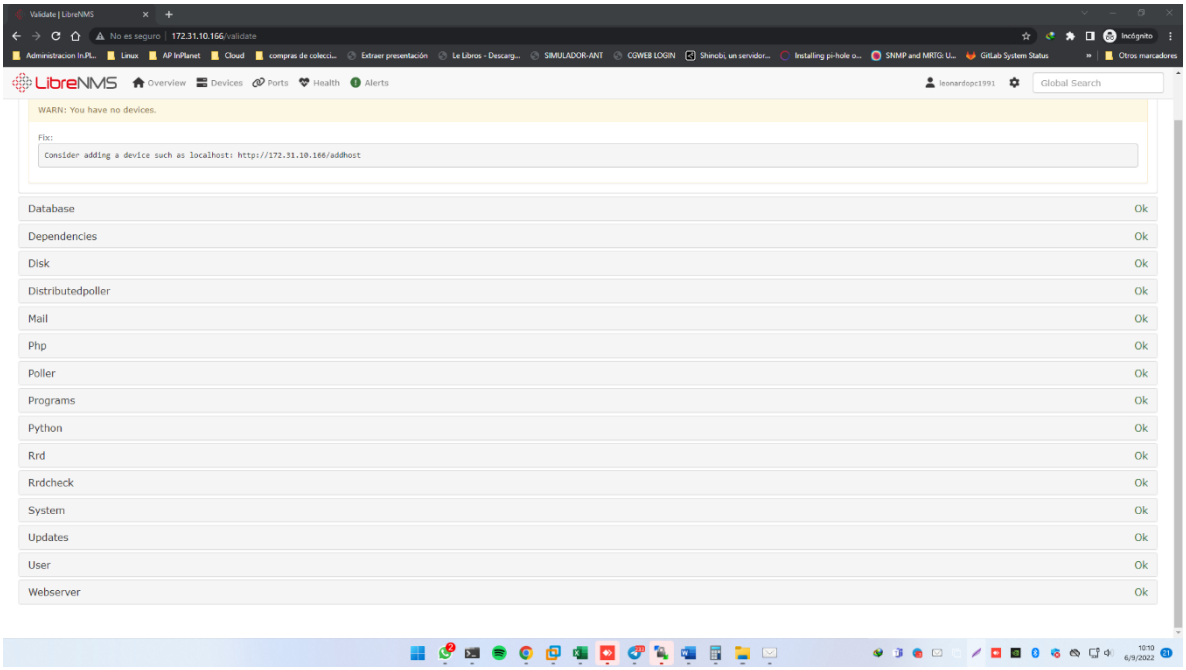


Ilustración 58: Debe salir todo en OK a excepto el addhost.

```
root@librenms01:~# su - librenms
$ ./validate.php
===================================
Component | Version
--------- | -------
LibreNMS  | 22.8.0-49-g14799ecdd
DB Schema | 2022_07_19_081224_plugins_unique_index (244)
PHP       | 8.1.2
Python    | 3.10.4
Database  | MariaDB 10.6.7-MariaDB-2ubuntu1.1
RRDTool   | 1.7.2
SNMP      | 5.9.1
===================================

[OK]    Composer Version: 2.4.1
[OK]    Dependencies up-to-date.
[WARN]  You have no devices.
        [FIX]:
        Consider adding a device such as localhost: /addhost
[OK]    Database connection successful
[OK]    Database Schema is current
[OK]    SQL Server meets minimum requirements
[OK]    lower_case_table_names is enabled
[OK]    MySQL engine is optimal
[OK]
[OK]    Database schema correct
[OK]    MySQl and PHP time match
[OK]    Active pollers found
[OK]    Dispatcher Service not detected
[OK]    Locks are functional
[OK]    Python poller wrapper is polling
[OK]    Redis is unavailable
[OK]    rrd_dir is writable
[OK]    rrdtool version ok
$
```

Ilustración 59: Desde consola en el usuario de librenms podemos ejecutar la comprobación con ./validate.php

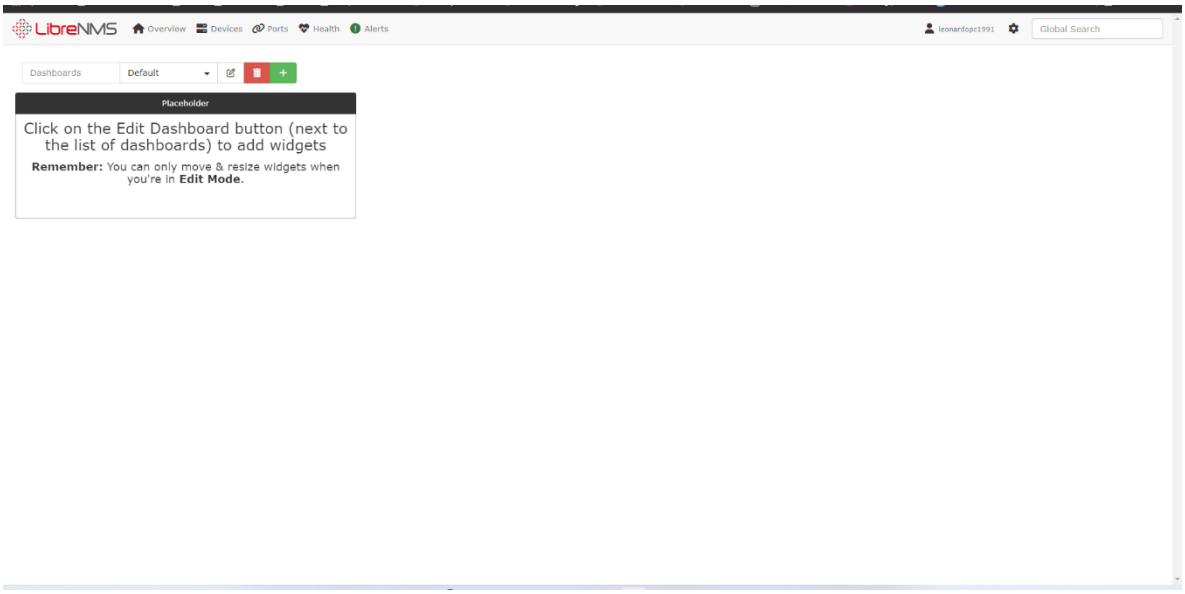


Ilustración 60: Luego de validar ya podremos ir a la ventana principal.

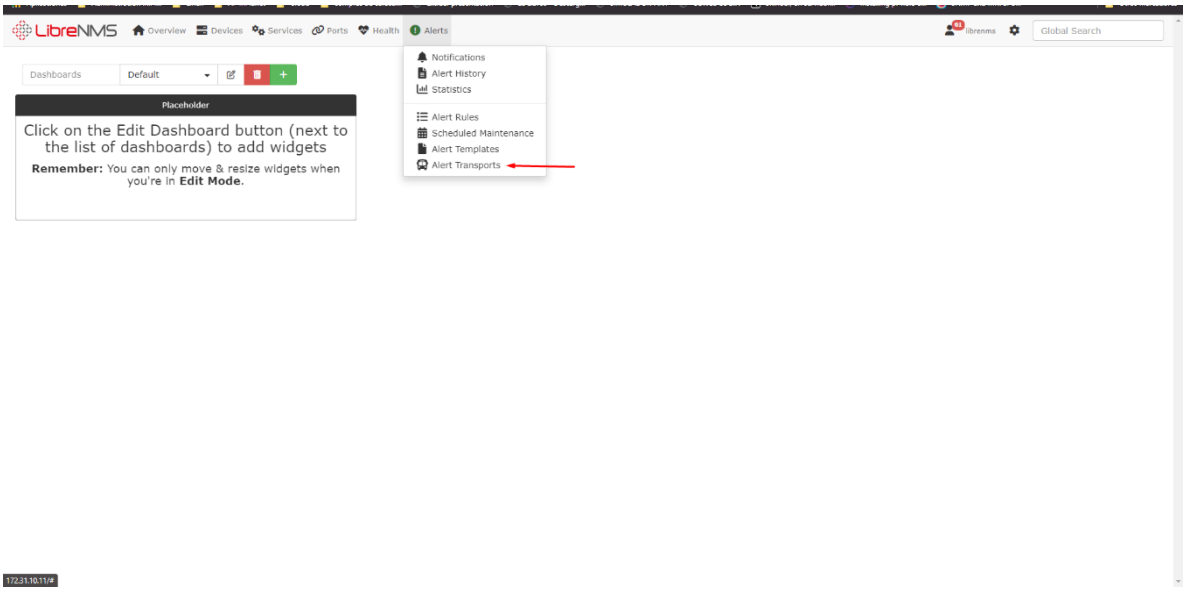


Ilustración 61: En el menú Alerts, seleccionamos Alert Transports.

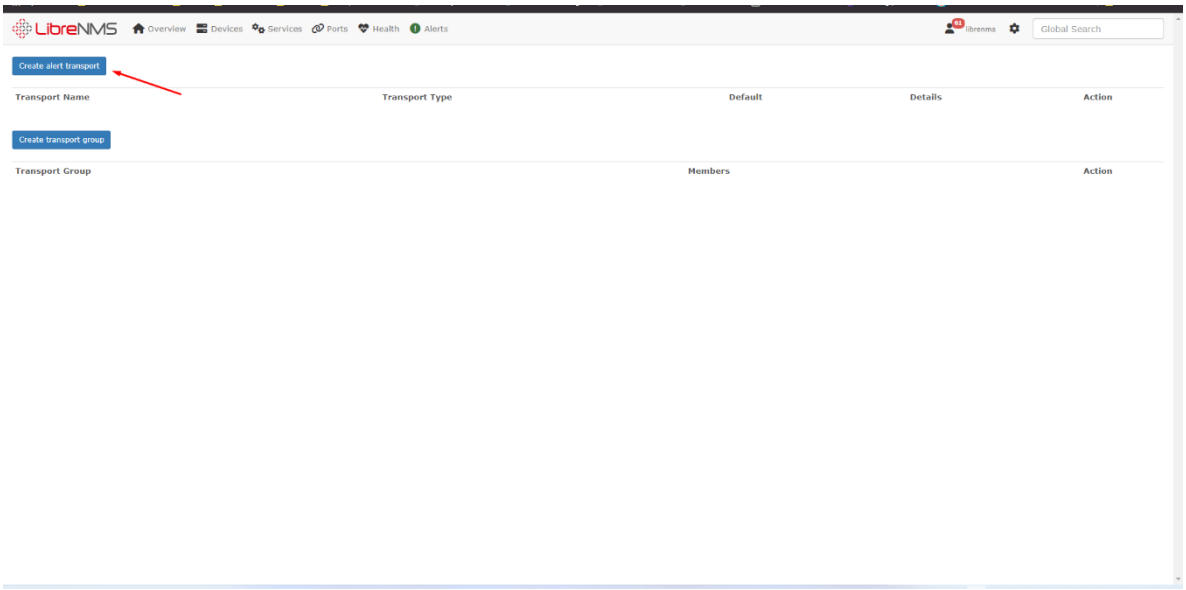


Ilustración 62: Seleccionaremos Create alert transport.

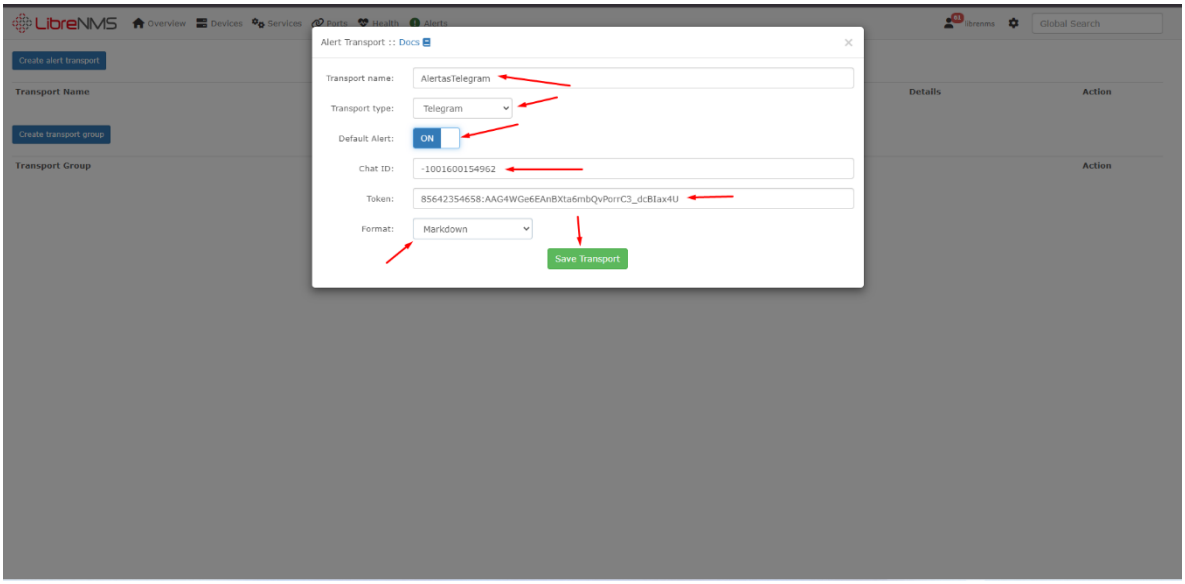


Ilustración 63: Luego crearemos un transport para telegram, donde indicaremos el Chat ID y el Token del grupo de telegram, el formato será Markdown.

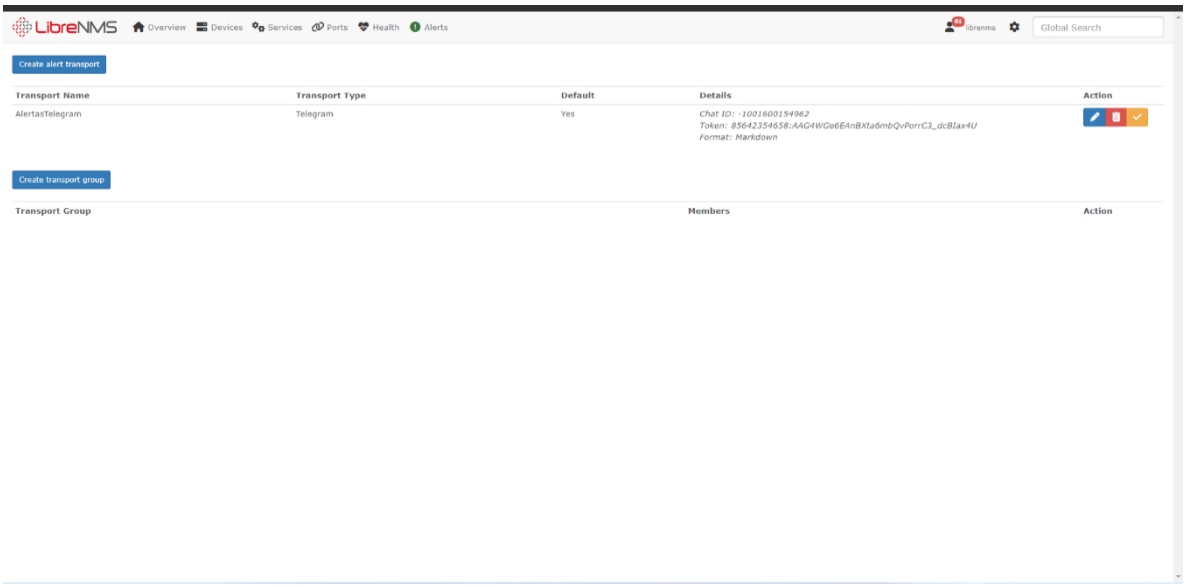


Ilustración 64: Podremos ver que fue creado el transport.

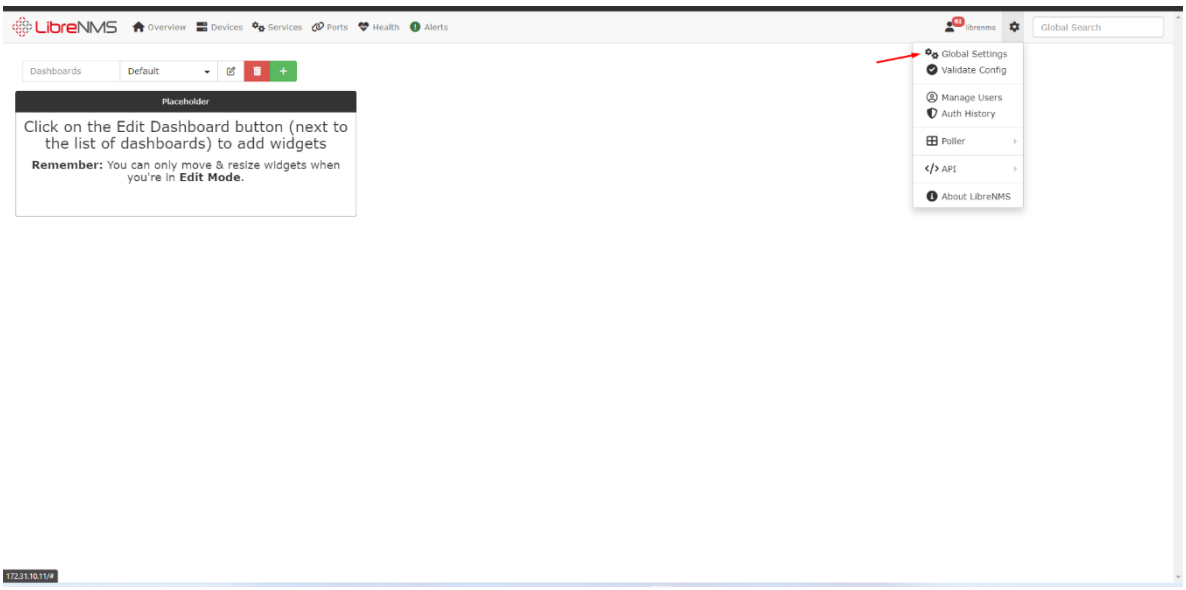


Ilustración 65: En el icono del engranaje seleccionamos Global Settings.

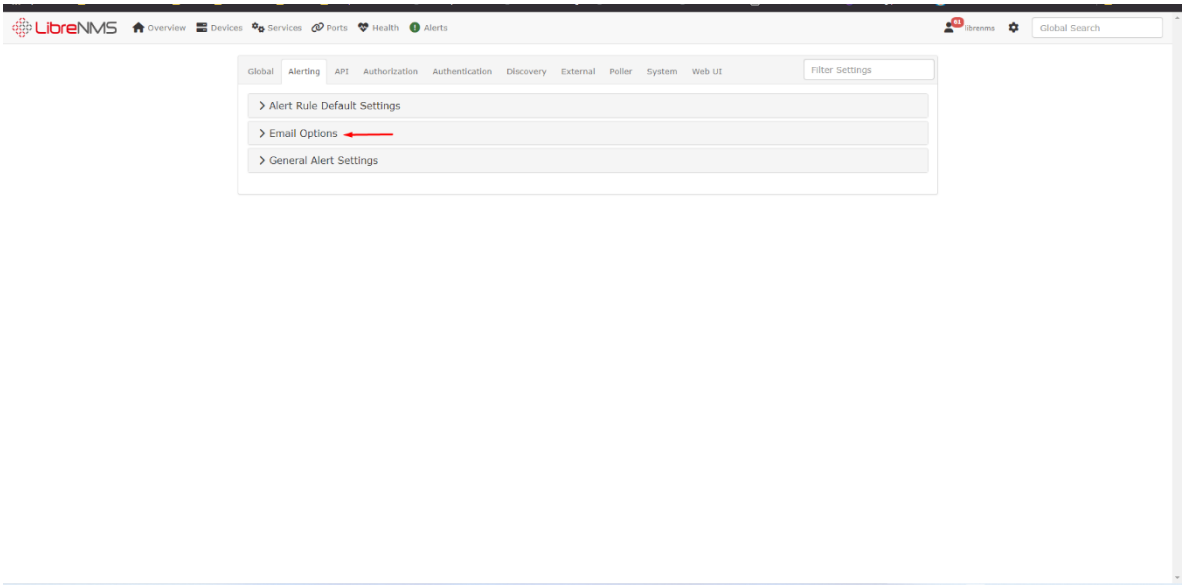


Ilustración 66: Nos ubicaremos en Alerting y luego en Email Options.

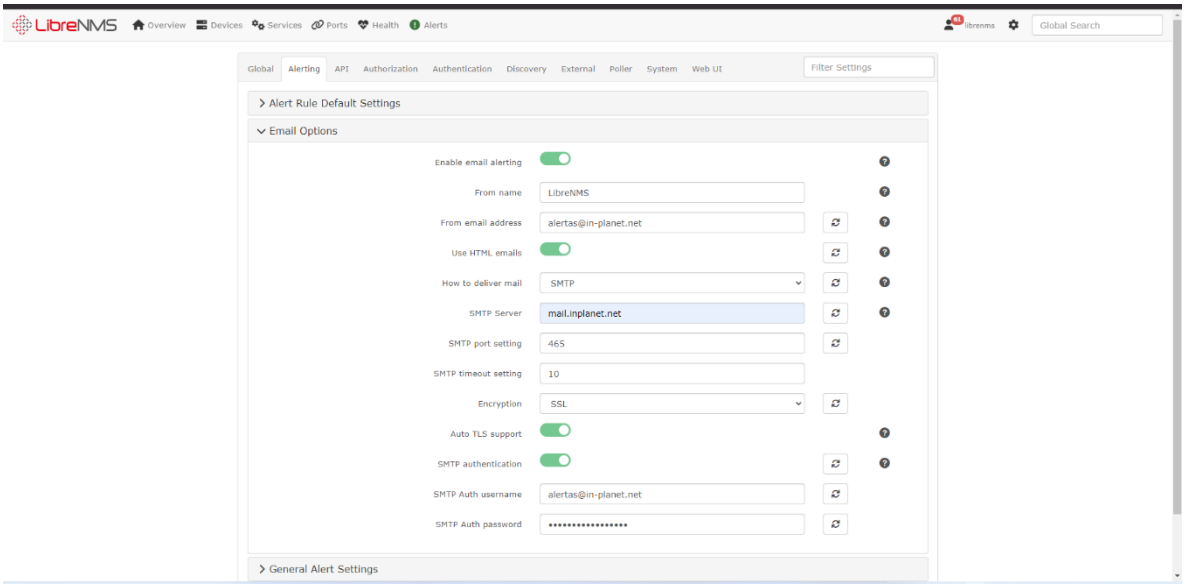


Ilustración 67: Configuraremos la cuenta de correo para las alertas con los datos proporcionados por el SysAdmin, guardamos.

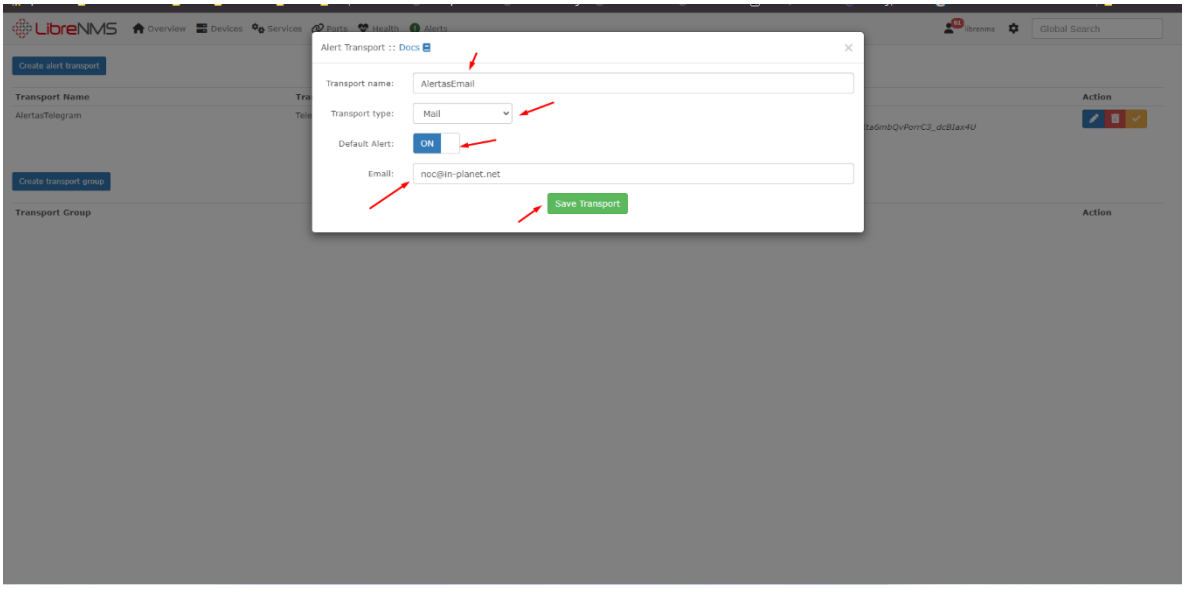


Ilustración 68: Creamos un nuevo Alert Transport, donde el type será Mail e indicaremos el mail al que se enviaran los correos.

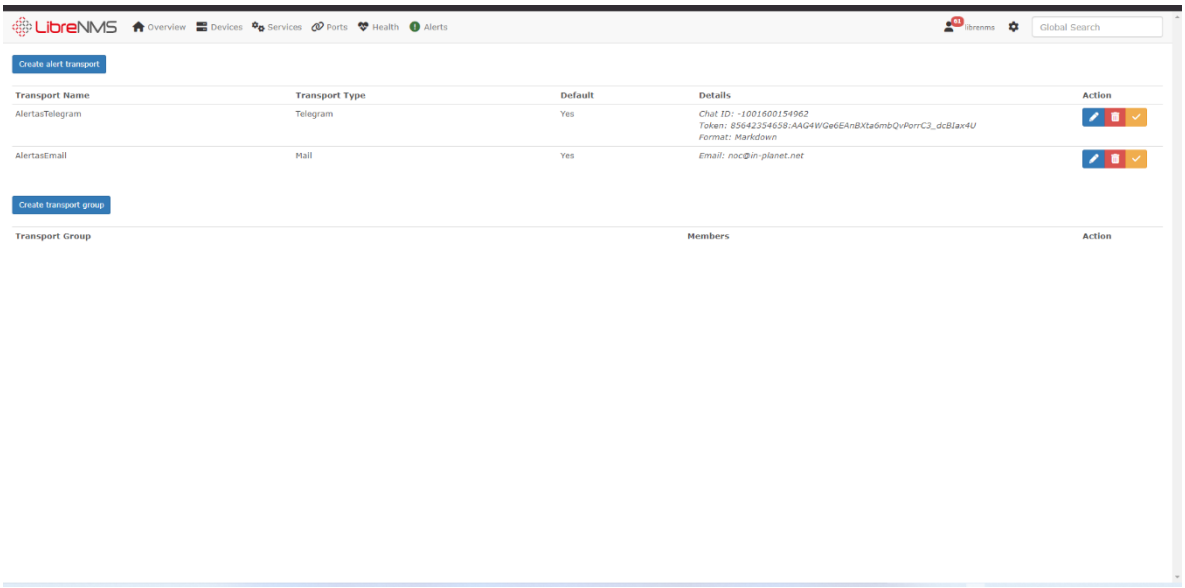


Ilustración 69:Podremos observar los transport creados.

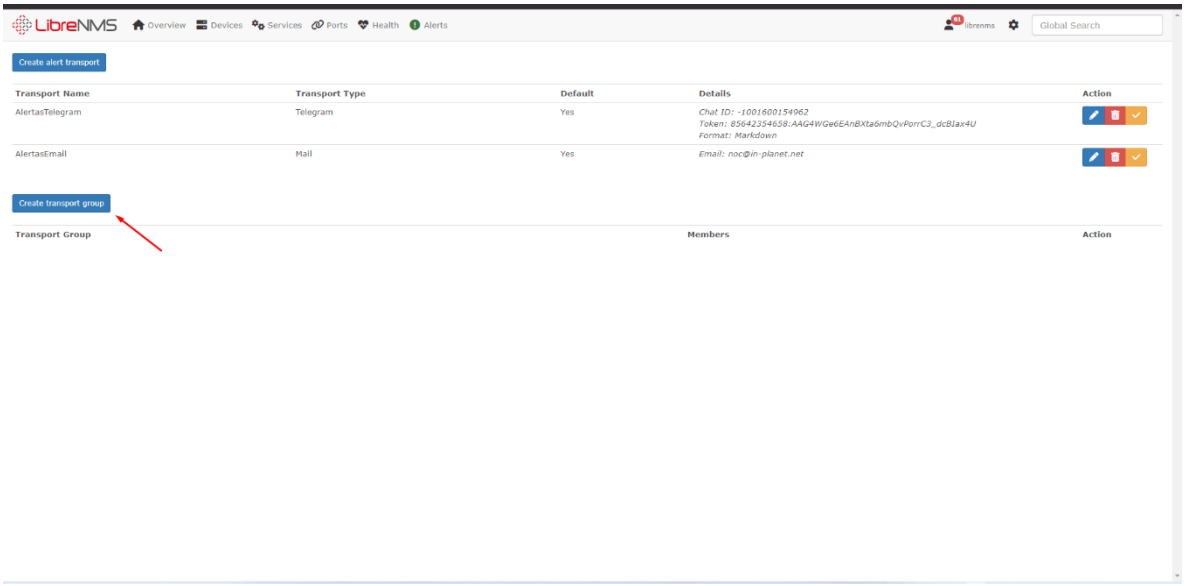


Ilustración 70: Crearemos un Transport group.

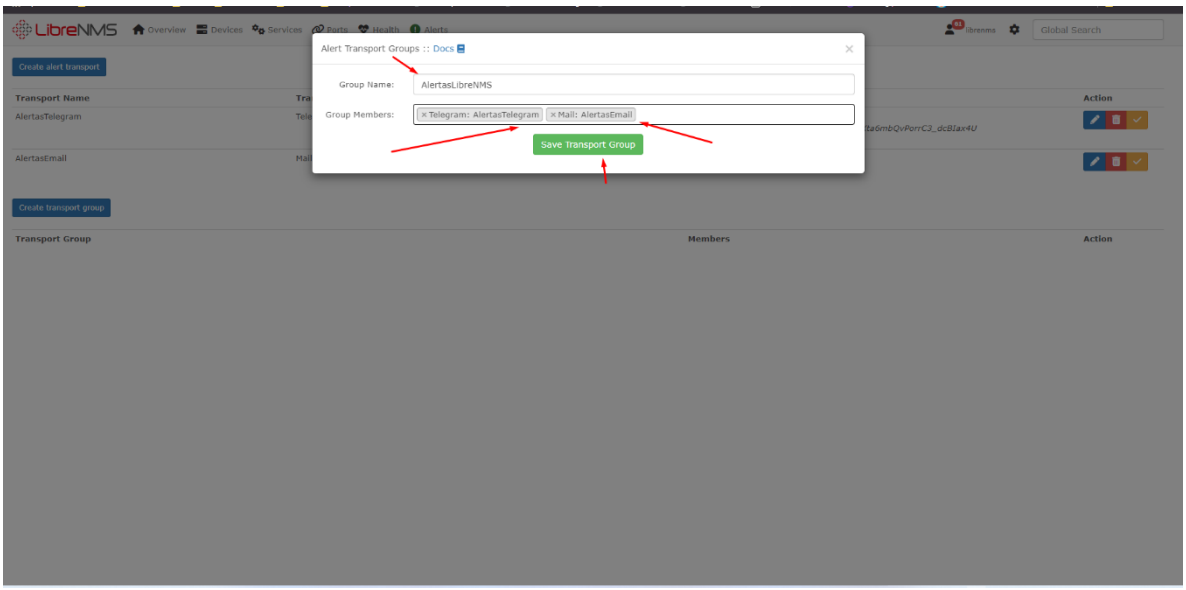


Ilustración 71: Indicaremos un nombre, y añadiremos los trnasports que acabamos de crear.

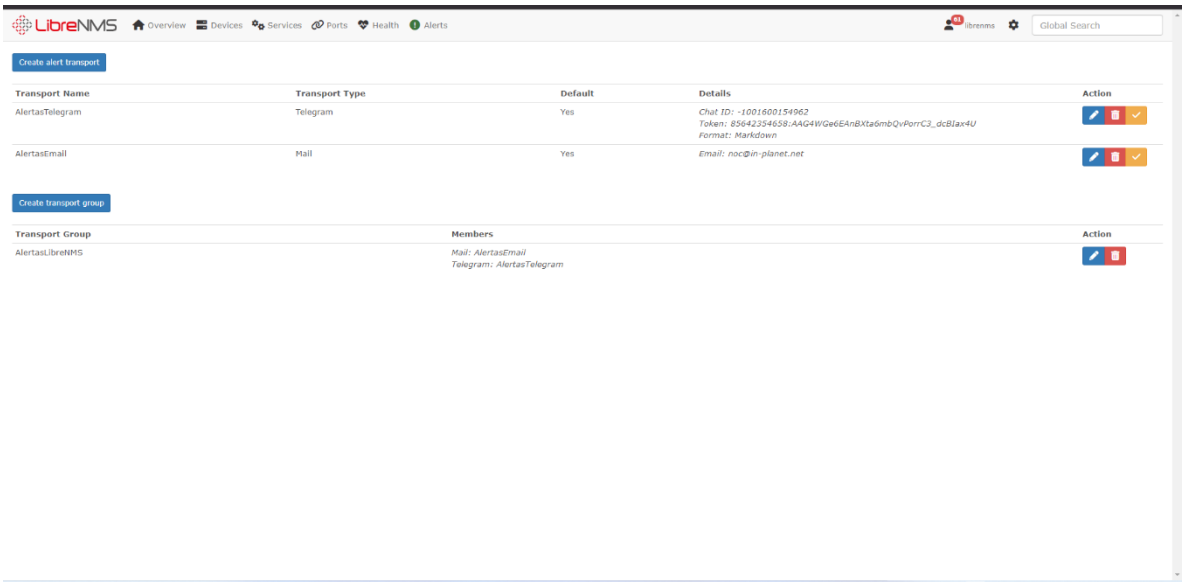


Ilustración 72: Podremos observar el transport grorp creado.

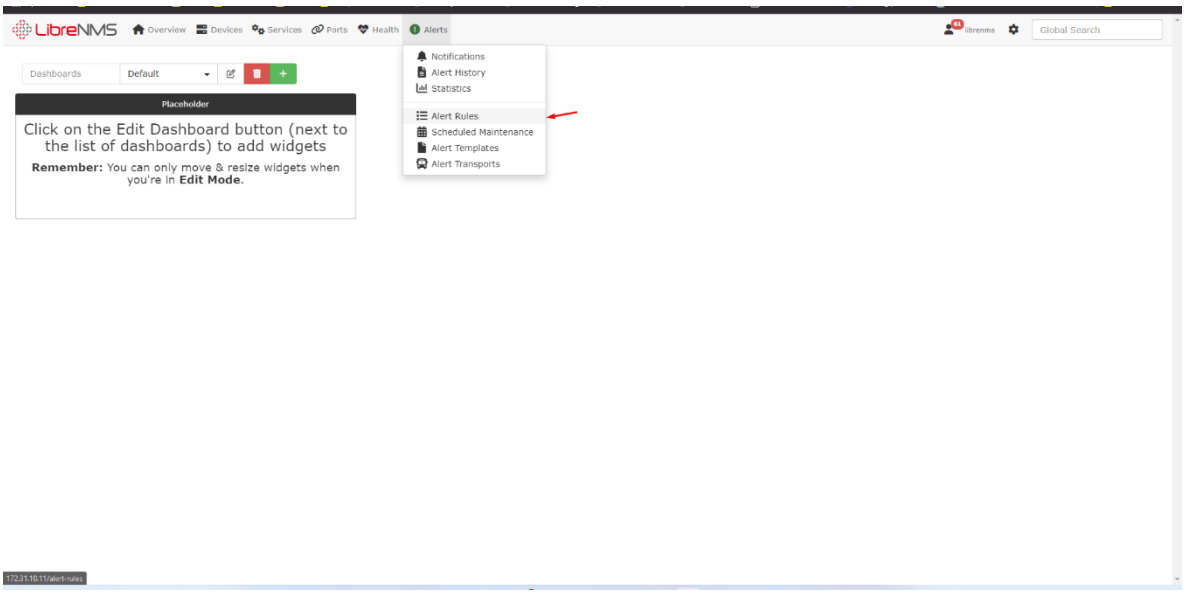


Ilustración 73: En el menú Alerts seleccionaremos, Alert Rules.

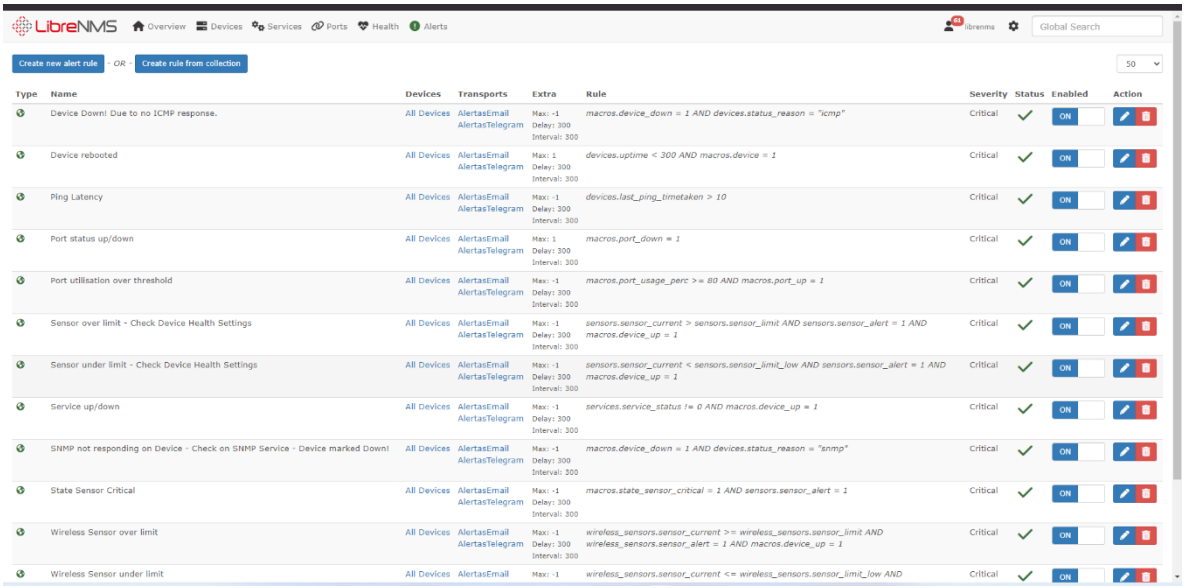


Ilustración 74: Veremos las reglas que vienen preconfiguradas en LibreNMS.

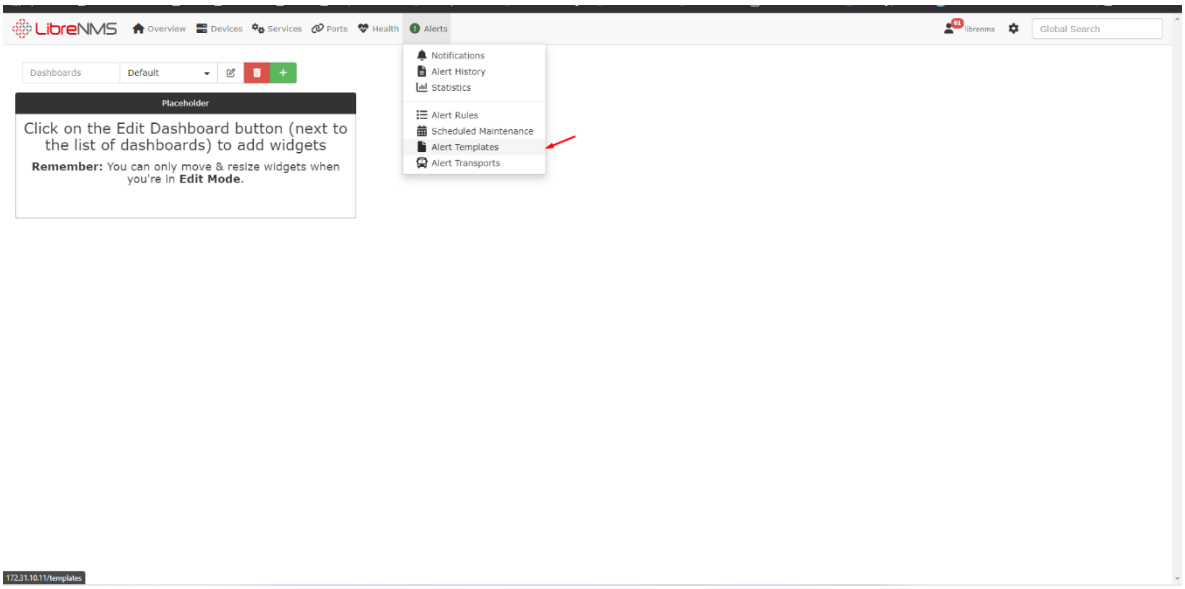


Ilustración 75: Ahora crearemos las plantillas de las alertas.

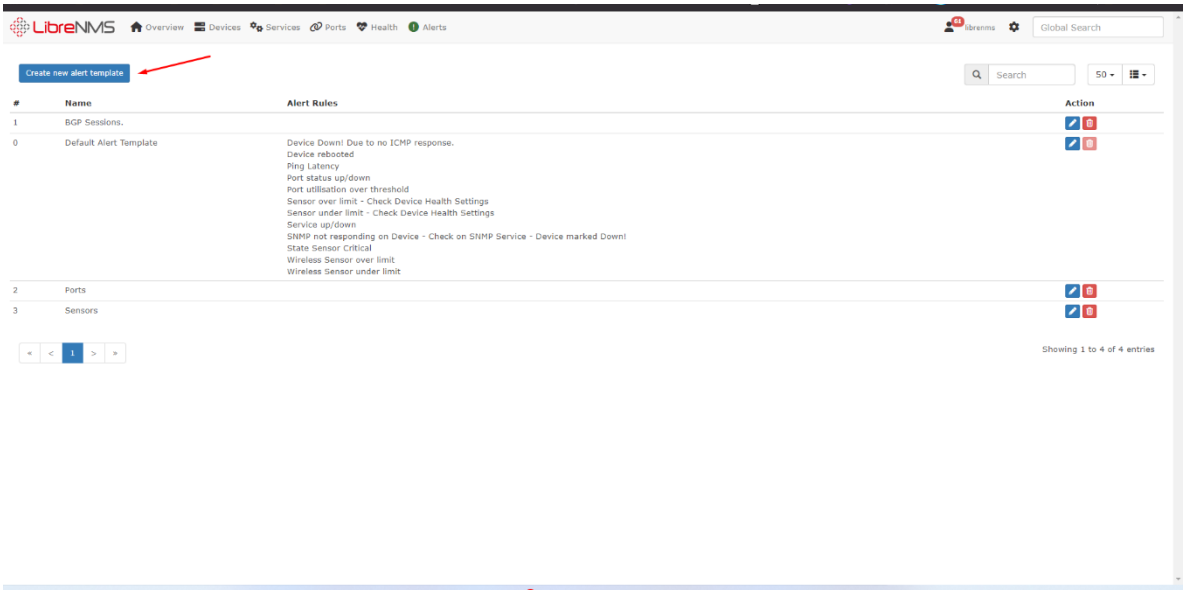


Ilustración 76: Creamos una nueva template.

Se creó la alerta para Interfaces Down, añadimos la regla Port status up/down y en el template pegaremos lo siguiente:

☠️ EQUIPO: {{ $alert->sysName }}

🔄 IP: {{ $alert->hostname }}

⌨️ VERSION FIRMWARE: {{ $alert->version}}

✋ GRAVEDAD: {{ $alert->severity }} @if ($alert->severity== warning) ⚠️ @endif @if ($alert->severity== critical) ❌ @endif @if ($alert->severity== ok) ❎ @endif

@if ($alert->state == 0)

⏳ TIEMPO TRANCURRIDO: {{ $alert->elapsed }} @endif

⏰ HORA DE ALERTA: {{ $alert->timestamp }}

✍️ DESCRIPCION DE ALERTA: @if ($alert->name) {{ $alert->name }} @else {{ $alert->rule }} @endif

@if ($alert->faults)

Ilustración 77: Primera parte del template.

PROBLEMAS CON:

PROBLEMAS CON:

@foreach ($alert->faults as $key => $value)

@if ($value['ifSpeed'] == 0)

- ❌ INTERFAZ: {{ $value['ifDescr'] }}
- ⭕ DESCRIPCION: {{ $value['ifAlias'] }}
- ⚠️ NEGOCIACION: @if ($value['ifSpeed'] == 0) ☹️ @endif Gbps

<img src="https://security.in-planet.net:777/graph.php?id={{ $value['port_id'] }}&type=port_bits&from=end-12h&height=300&width=1440">

@endif

@endforeach

@endif

Ilustración 78: Segunda parte del template.

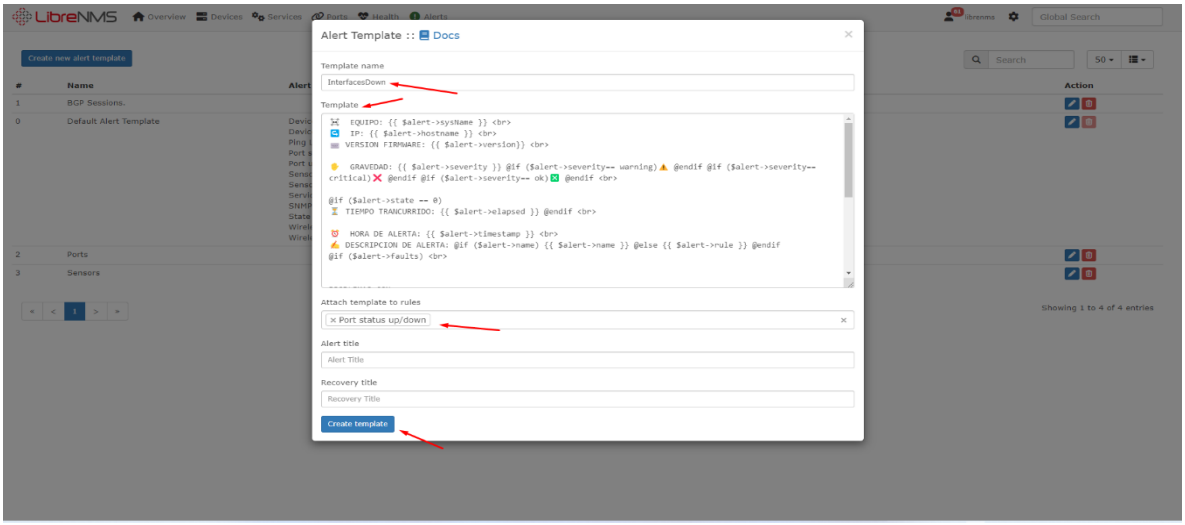


Ilustración 79: Crearemos la alerta para InterfacesDown, añadimos la regla Port status up/down y en el template pegaremos lo siguiente.

Crearemos la alerta para PingDown, en las reglas usaremos Device Down! Due to no ICMP responde, y pegaremos lo siguiente:

☠️ EQUIPO: {{ $alert->sysName }}

🔄 IP: {{ $alert->hostname }}

⌨️ VERSION FIRMWARE: {{ $alert->version}}

✋ GRAVEDAD: {{ $alert->severity }} @if ($alert->severity== warning) ⚠️ @endif @if ($alert->severity== critical) ❌ @endif @if ($alert->severity== ok) ❎ @endif

@if ($alert->state == 0)

⏳ TIEMPO TRANCURRIDO: {{ $alert->elapsed }} @endif

⏰ HORA DE ALERTA: {{ $alert->timestamp }}

✍️ DESCRIPCION DE ALERTA: @if ($alert->name) {{ $alert->name }} @else {{ $alert->rule }} @endif

Ilustración 80: Template para PingDown.

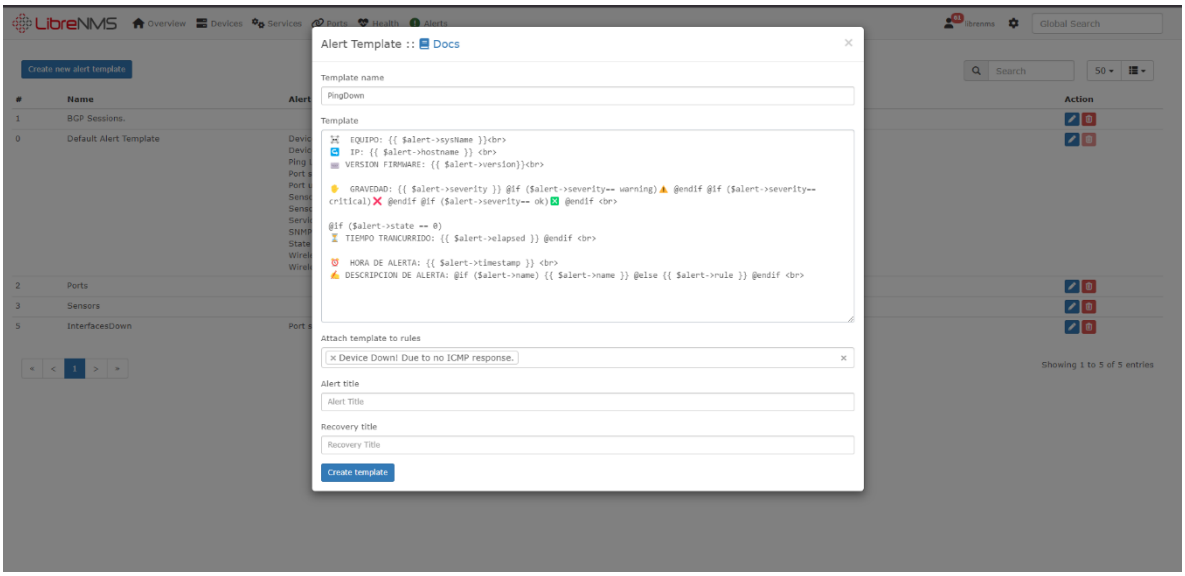


Ilustración 81: Crearemos la alerta para PingDown, en las reglas usaremos Device Down! Due to no ICMP responde, y pegaremos lo siguiente.

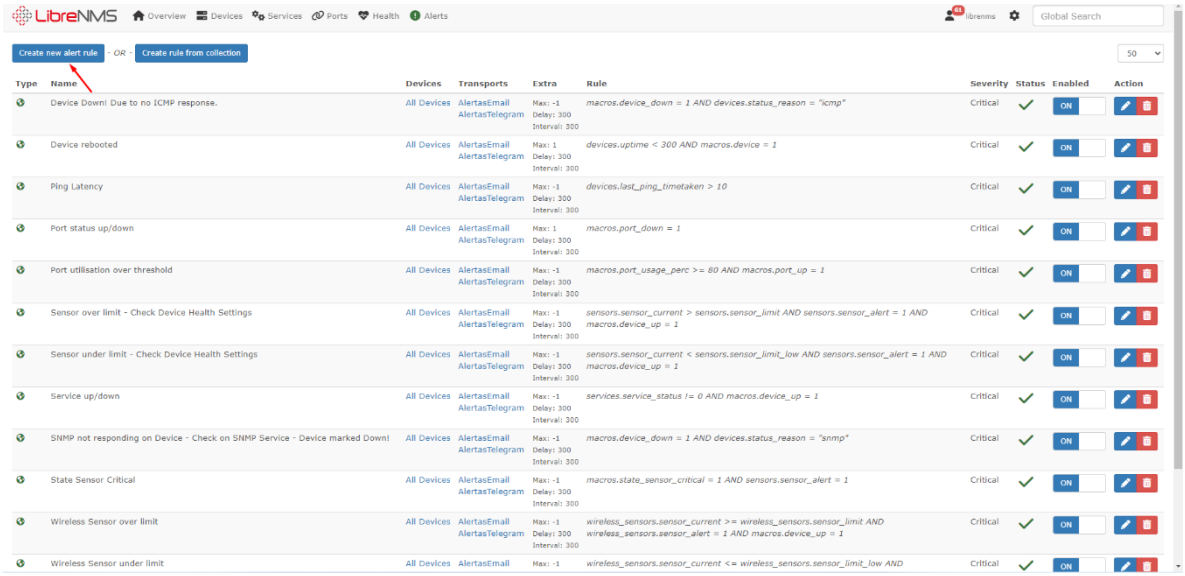


Ilustración 82: Iremos al menú Alert luego Alert rule y crearemos una nueva.

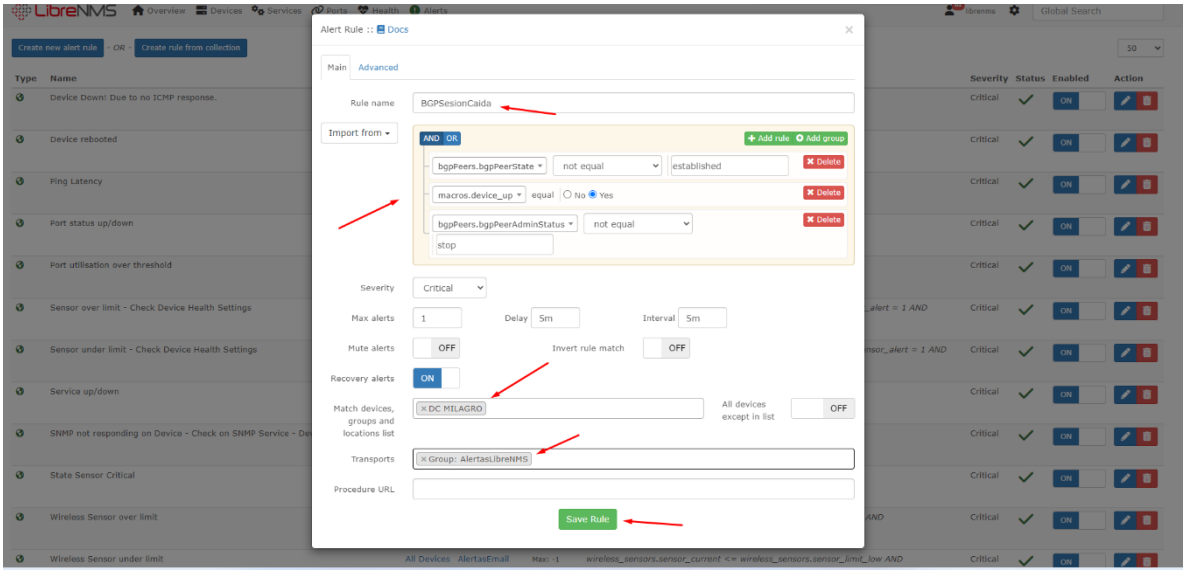


Ilustración 83: Añadiremos la regla BGPSesionCaida, añadiremos las reglas eindicaremos el Match de dispositivos y los transports.

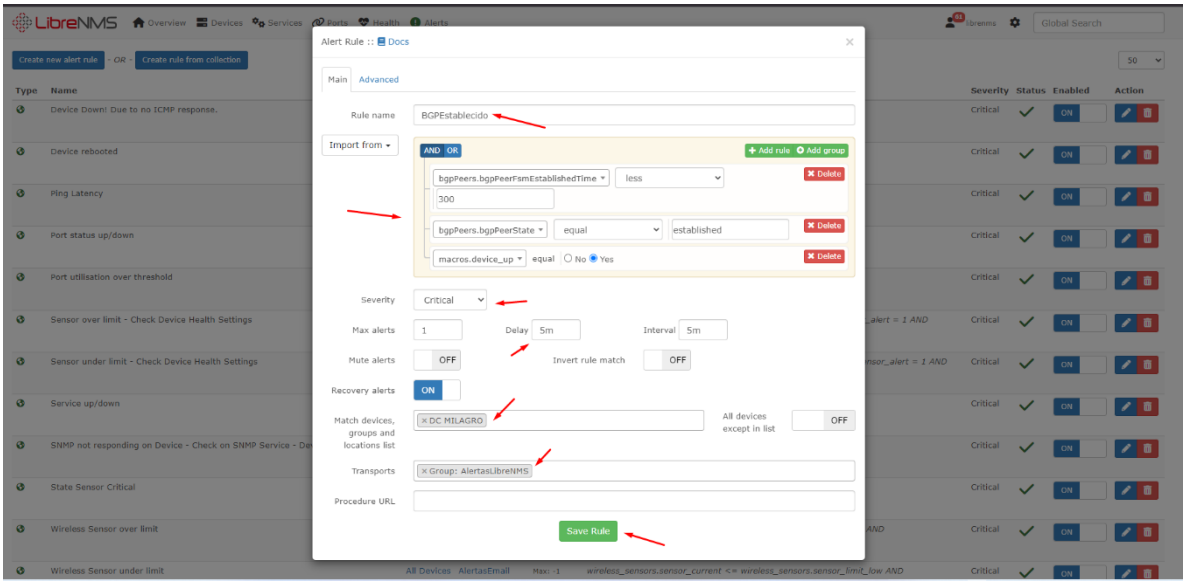


Ilustración 84: Realizaremos algo similar para BGPEstablecido.

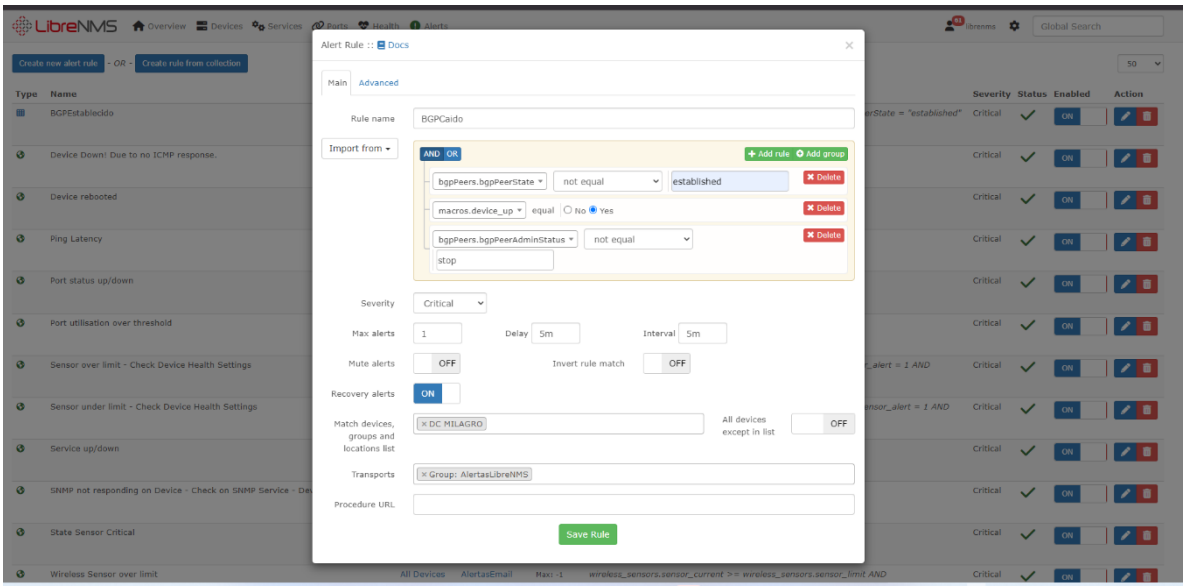


Ilustración 85: Realizaremos algo similar para BGPCaido.

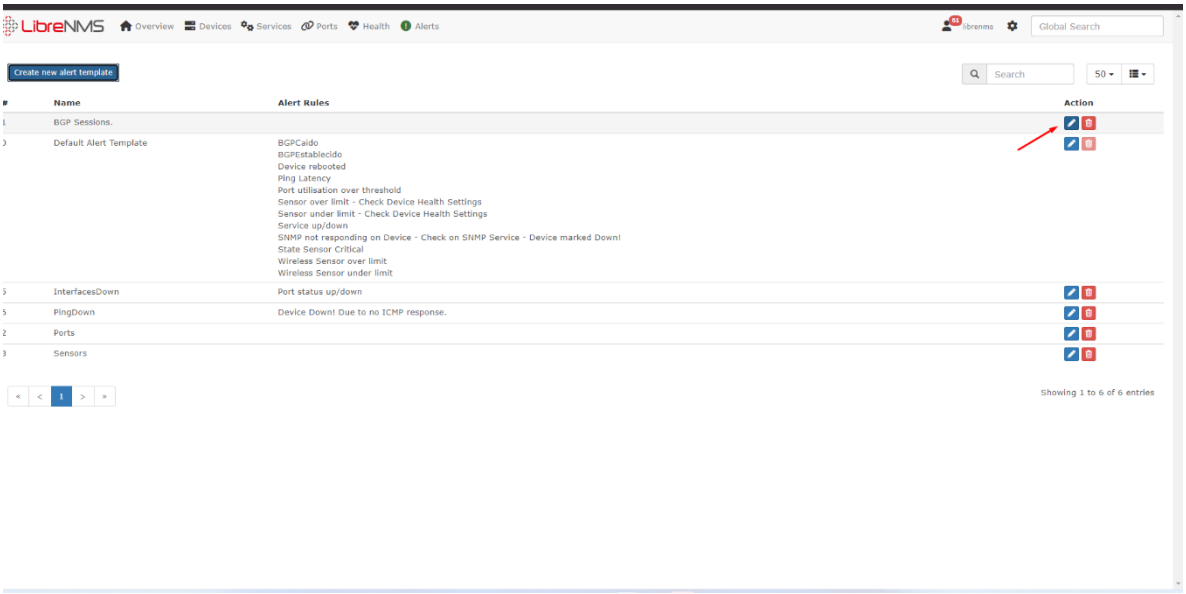


Ilustración 86: Modificaremos el Alert Templates de BGP.

Se modificó las reglas y añadiremos BGPCaido y BGPEstablecido, añadiremos lo siguiente al template:

☠️ EQUIPO: {{ $alert->sysName }}

🔗 IP: {{ $alert->hostname }}

⌨️ VERSION FIRMWARE: {{ $alert->version}}

✋ GRAVEDAD: {{ $alert->severity }} @if ($alert->severity== warning) ⚠️ @endif @if ($alert->severity== critical) ❌ @endif @if ($alert->severity== ok) ❎ @endif

@if ($alert->state == 0)

⏳ TIEMPO TRANCURRIDO: {{ $alert->elapsed }} @endif

⏰ HORA DE ALERTA: {{ $alert->timestamp }}

✍️ DESCRIPCION DE ALERTA: @if ($alert->name) {{ $alert->name }} @else {{ $alert->rule }} @endif

@if ($alert->faults)

Ilustración 87:primera parte del template para BGP.

```
☹️ FALLAS:<br>
@foreach ($alert->faults as $key => $value)<br>
 # Peer: {{ $value['astext'] }}<br>
 Peer IP: {{ $value['bgpPeerIdentifier'] }} <br>
 Peer AS: {{ $value['bgpPeerRemoteAs'] }} <br>
 Peer EstTime: {{ $value['bgpPeerFsmEstablishedTime'] }} <br>
 Peer State: {{ $value['bgpPeerState'] }} <br>
@endforeach <br>
@endif <br>
```

Ilustración 88: Segunda parte del template para BGP.

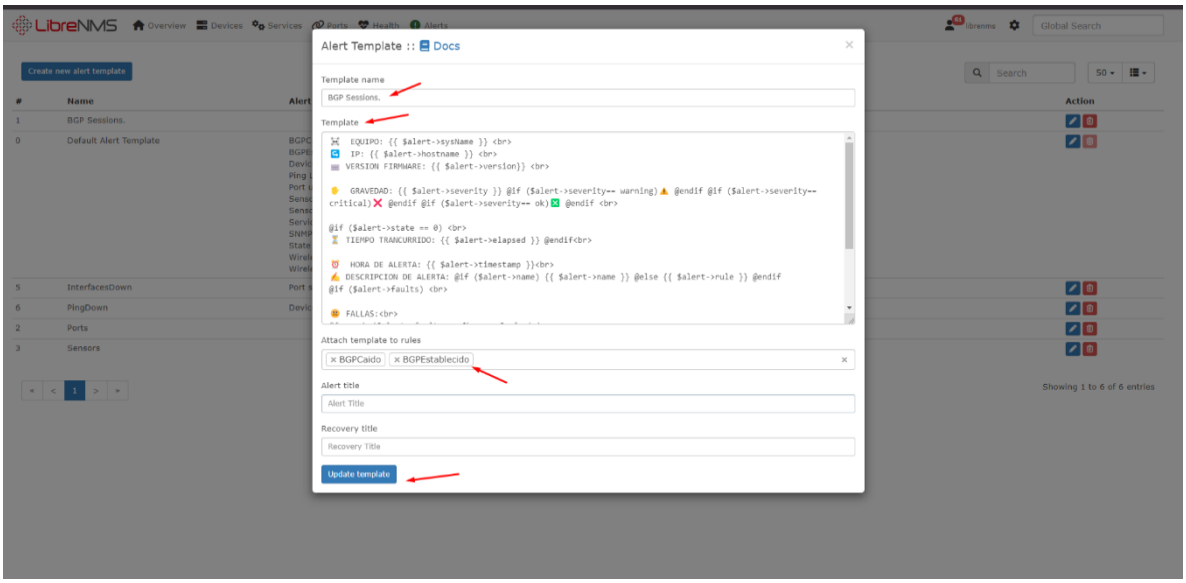


Ilustración 89:Modificaremos las reglas y añadiremos BGPCaido y BGPEstablecido, añadiremos lo siguiente al template.

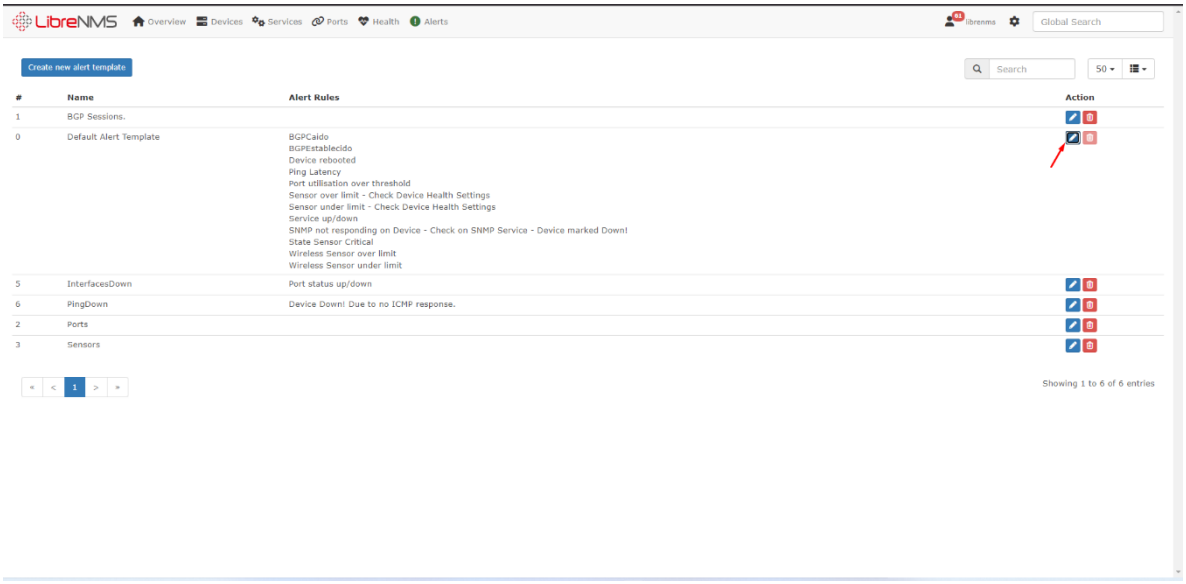


Ilustración 90: Modificaremos Default Alert Template.

En el template añadiremos:

```
EQUIPO: {{ $alert->sysName }}<br>
IP: {{ $alert->hostname }}<br>
VERSION FIRMWARE: {{ $alert->version}}<br>

GRAVEDAD: {{ $alert->severity }} @if ($alert->severity== warning) ⚠️
@endif @if ($alert->severity== critical) ❌ @endif @if ($alert->severity==
ok) ✅ @endif<br>

@if ($alert->state == 0)
⏳ TIEMPO TRANCURRIDO: {{ $alert->elapsed }} @endif<br>

⏰ HORA DE ALERTA: {{ $alert->timestamp }}<br>
✍️ DESCRIPCION DE ALERTA: @if ($alert->name) {{ $alert->name }}
@else {{ $alert->rule }} @endif<br>
@if ($alert->faults)

@foreach ($alert->faults as $key => $value)
```

Ilustración 91:Primera parte del template.

⭕ #{{ $key }} PROBLEMAS CON: {{ $value['sensor_descr'] }}

⏭️ TIPO DE SENSOR: {{$value['sensor_class']}}

⛔ Valor Actual: {{$value['sensor_current']}} @if ($value['sensor_class'] == fanspeed) RPM 🪽 @endif @if ($value['sensor_class'] == temperature) °C 🌞 @endif @if ($value['sensor_class'] == voltage) V ⚡ @endif

⏫ Valor Máximo: {{$value['sensor_limit']}} @if ($value['sensor_class'] == fanspeed) RPM @endif @if ($value['sensor_class'] == temperature) °C @endif @if ($value['sensor_class'] == voltage) V @endif

⏬ Valor Mínimo: {{$value['sensor_prev']}} @if ($value['sensor_class'] == fanspeed) RPM @endif @if ($value['sensor_class'] == temperature) °C @endif @if ($value['sensor_class'] == voltage) V @endif

<img src="https://security.in-planet.net:777/graph.php?id={{ $value['sensor_id']}}&type=sensor_{{ $value['sensor_class']}}&from=end-12h&width=459&height=213&lazy_w=55">

@endforeach

@endif

Ilustración 92: Segunda parte del template.

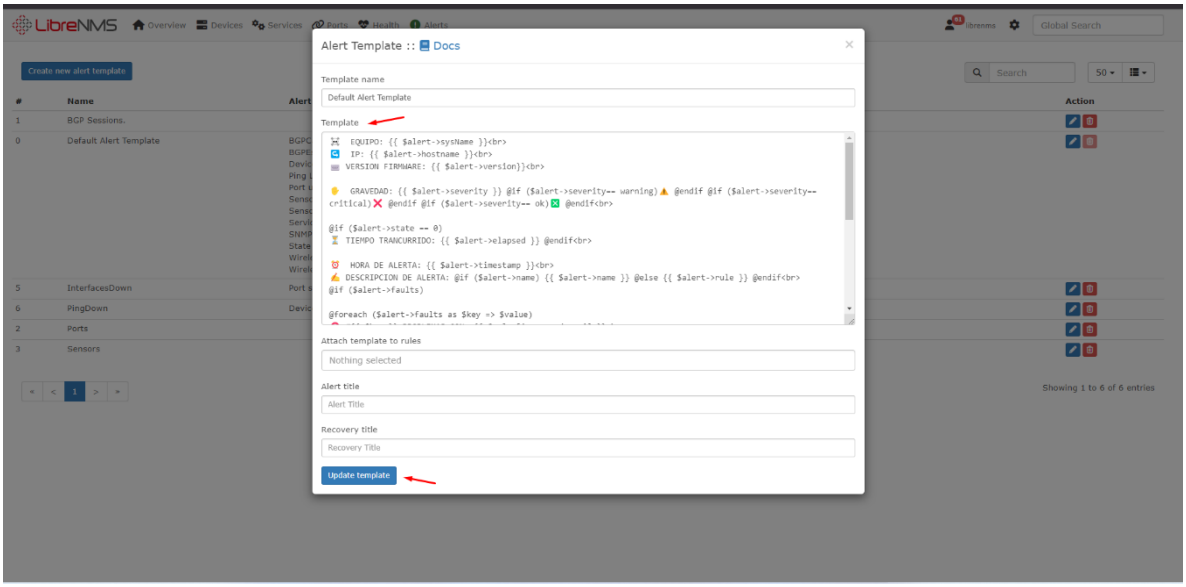


Ilustración 93: En el template añadiremos.

En la terminal de nuestro servidor LibreNMS ejecutamos los comandos para la instalación y configuración de la base de datos Influx la cual será consumida por Grafana para la configuración de los dashboards, la *ilustración 65* sirve como referente para verificar la correcta instalación de Influxdb. Continuación se puede observar los pasos y el comando utilizado para la instalación de Influxdb:

3.5.10. Instalación de Influxdb:

Ilustración 94: Configuración del repositorio y su respectiva llave.

Actualizaremos el repositorio:

root@librenms01:~# apt-get update -y

Instalaremos el paquete de Influx:

root@librenms01:~# apt-get install influxdb -y

Habilitaremos el incio con el OS:

root@librenms01:~# systemctl enable --now influxdb

Iniciaremos el servicio:

root@librenms01:~# systemctl start influxdb

Modificaremos el archivo de configuración influxdb.conf:

root@librenms01:~# nano /etc/influxdb/influxdb.conf

Añadiremos lo siguiente en el apartadp [http]:

auth-enabled = true

Reinicaremos el servicio:

root@librenms01:~# systemctl restart influxdb

Ingresaremos al motor de la BD de Influx:

root@librenms01:~# influx

Crearemos una BD:

> create database librenms

Nos ubicaremos en la BD:

> use librenms

Crearemos un usuario con todos los permisos:

```
> create user admin with password 'admin' with all privileges
```

Ilustración 95: Creación de usuario para la base de datos en InfluxDB.

Ilustración 96: Si la instalación fue correcta en el navegador escribiendo la ip y el puerto 8086 tendremos que ver lo siguiente.

A continuación, se instaló el software Grafana para la creación de los dashboard, para lo cual mediante en la terminal se ejecutaron los comandos que instalaban los paquetes

necesarios, luego se configuró el usuario administrador cambiando su clave, y por último se procedió con la integración de Grafana con LibreNMS, como podrán observar en las *ilustraciones 66 hasta la 83.*

3.5.11. Instalación de Grafana:

Actualizamos los repositorios:

root@grafana01:~# apt-get update -y

Instalamos Grafana:

```
root@grafana01:~# apt-get install apt-transport-https software-properties-common wget -y
root@grafana01:~# wget -q -O /usr/share/keyrings/grafana.key https://packages.grafana.com/gpg.key
root@grafana01:~# echo "deb [signed-by=/usr/share/keyrings/grafana.key] https://packages.grafana.com/oss/deb beta main" | tee -a /etc/apt/sources.li
st.d/grafana.list
root@grafana01:~# apt-get update -y
root@grafana01:~# apt-get -y  install grafana
```

Ilustración 97: Instalación de software previo, repositorio y llave, actualización de los repositorios e instalación de Grafana.

Iniciamos el servicio de grafana:

root@grafana01:~# systemctl start grafana-server.service

Configuramos para que arranque junto al OS:

root@grafana01:~# systemctl enable grafana-server.service

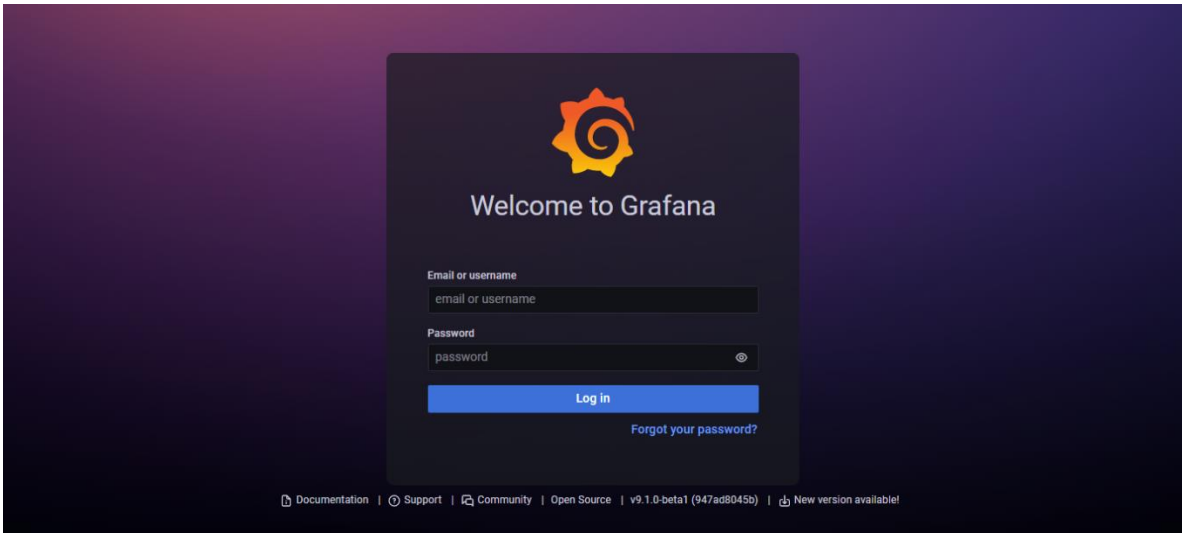


Ilustración 98: en el navegador escribiremos la ip de nuestro servidor seguido del puerto 3000.

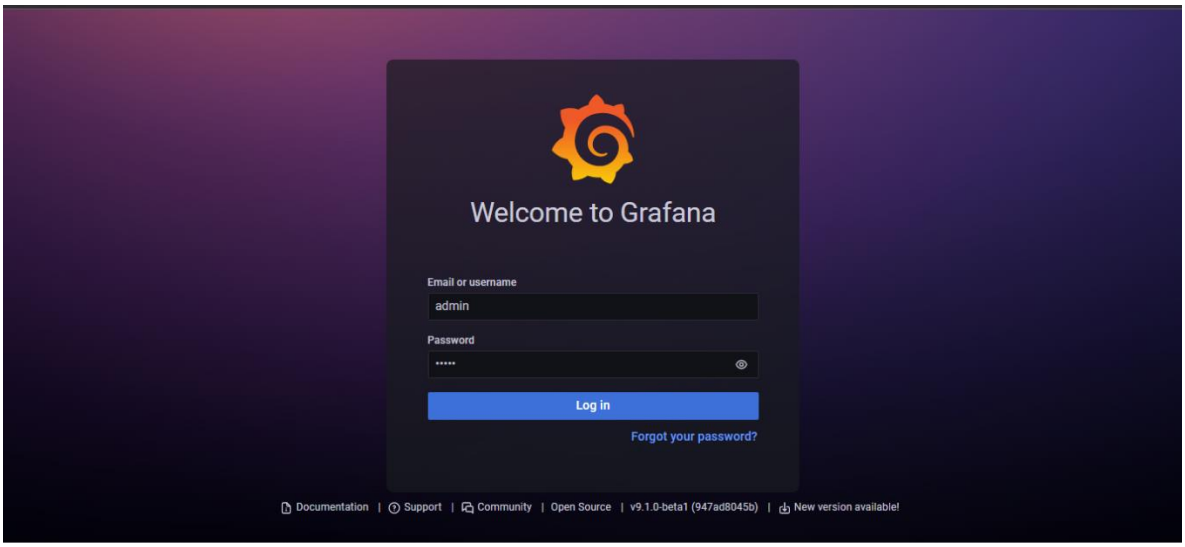


Ilustración 99: Las credenciales por defecto son admin/admin.

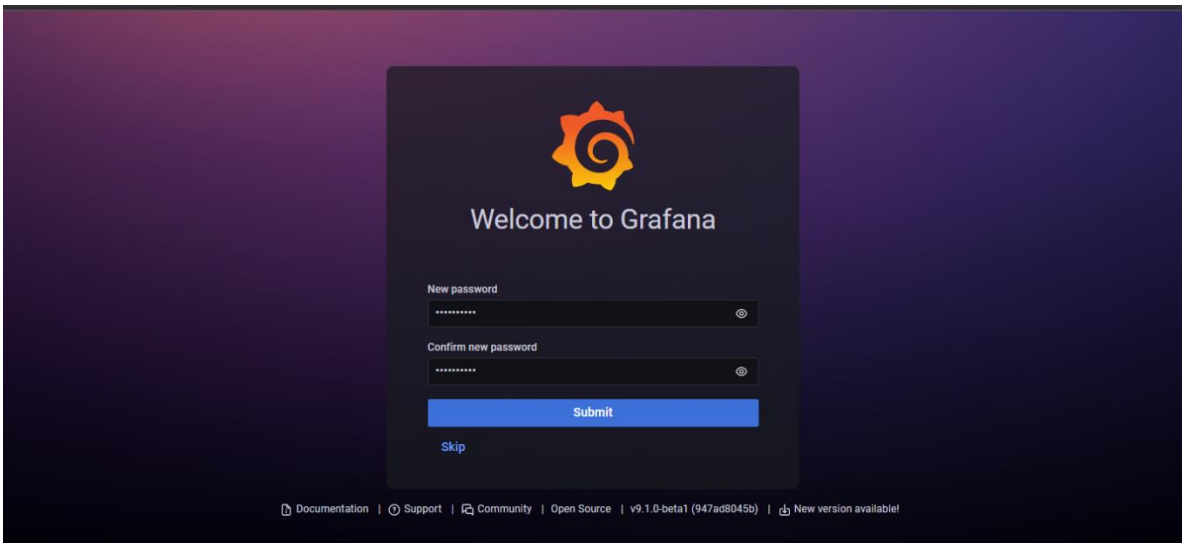


Ilustración 100: Nos pedirá una nueva contraseña.

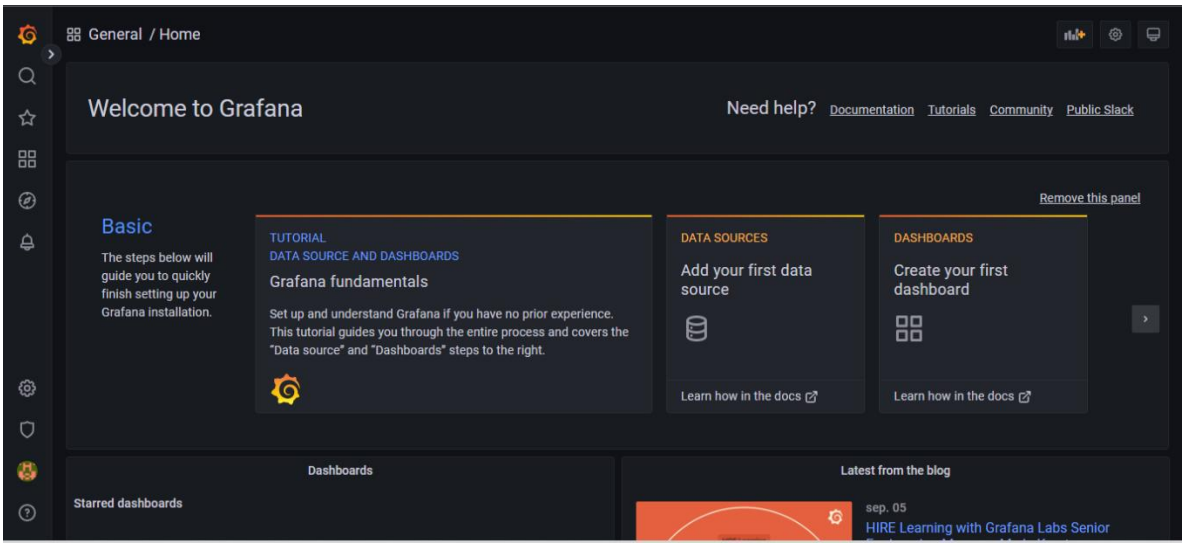


Ilustración 101: Nos cargará la pantalla principal.

3.5.12. Incorporación de Grafana:

```
// añadido influx
$config['influxdb']['enable'] = true;
$config['influxdb']['transport'] = 'http'; # Default, other options: https, udp
$config['influxdb']['host'] = '127.0.0.1';
$config['influxdb']['port'] = '8086';
$config['influxdb']['db'] = 'librenms';
$config['influxdb']['username'] = 'admin';
$config['influxdb']['password'] = 'admin';
$config['influxdb']['timeout'] = 0; # Optional
$config['influxdb']['verifySSL'] = false; # Optional
// fin influx
// This is the user LibreNMS will run as
//Please ensure this user is created and has the correct permissions to your install
$config['user'] = 'librenms';

### This should *only* be set if you want to *force* a particular hostname/port
### It will prevent the web interface being usable form any other hostname
$config['base_url']     = "/";

### Enable this to use rrdcached. Be sure rrd_dir is within the rrdcached dir
### and that your web server has permission to talk to rrdcached.

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket
^X Exit  ^R Read File  ^\ Replace  ^U Uncut Text  ^T To Spell  ^_ Go To Line  M-E Redo  M-6 Copy Text  M-W WhereIs Next
```

Ilustración 102: Editamos el archivo config.php dentro del directorio de librenms y añadimos las configuraciones necesarias para la conexión.

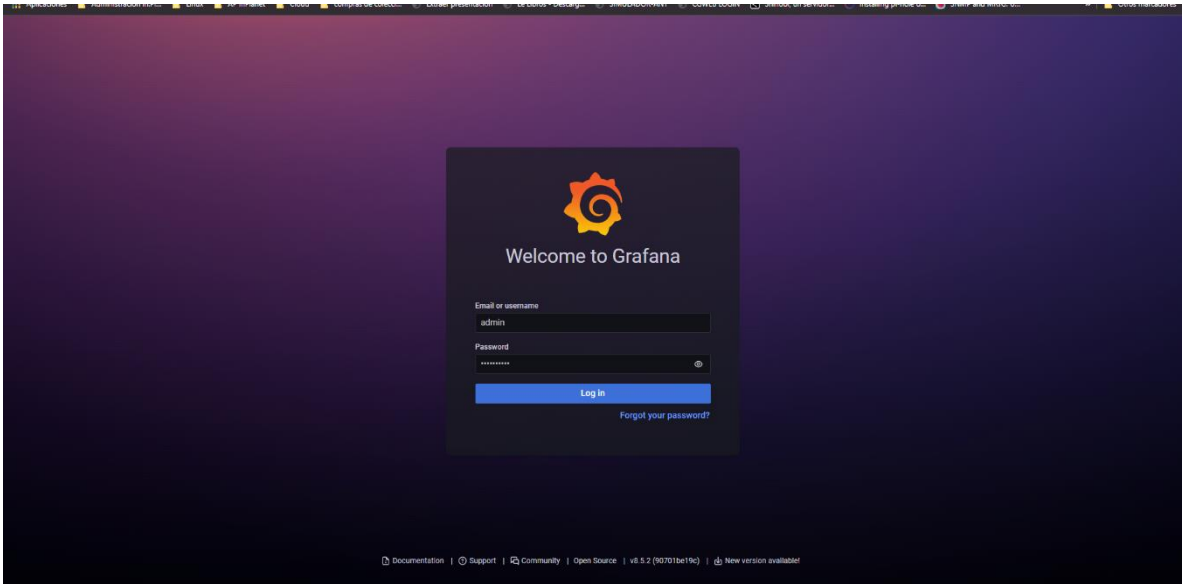


Ilustración 103: Iniciamos sesión en Grafana.

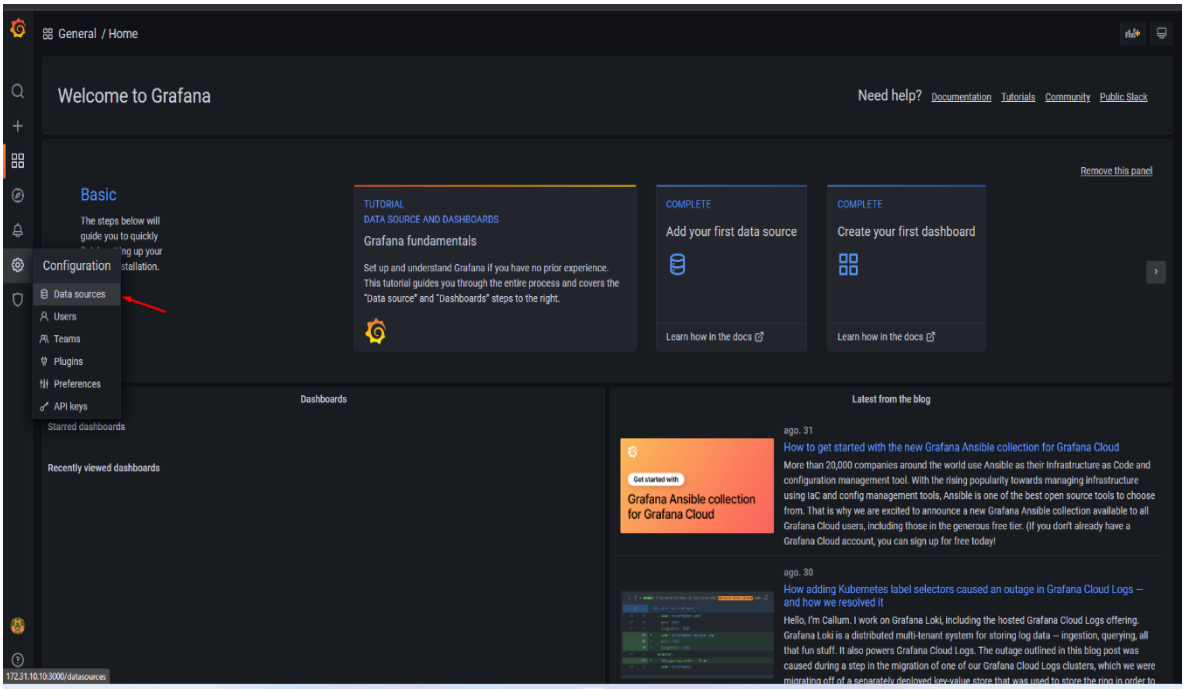


Ilustración 104: En la ventana principal iremos a configuración y luego a Data sources.

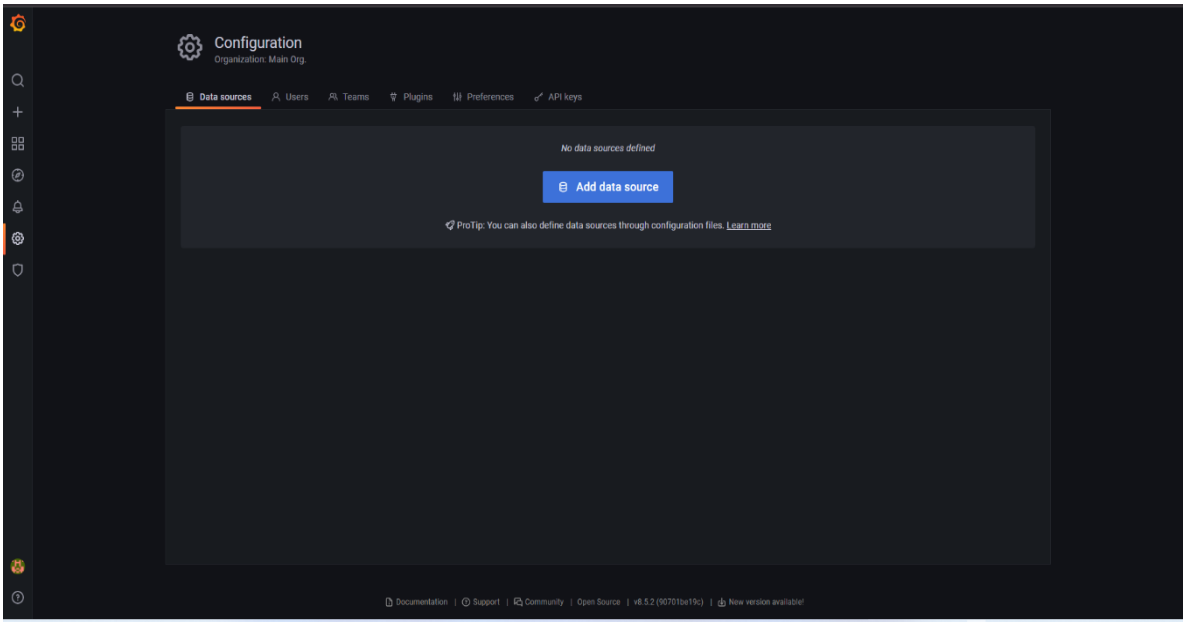


Ilustración 105: Daremos clic en Add data source.

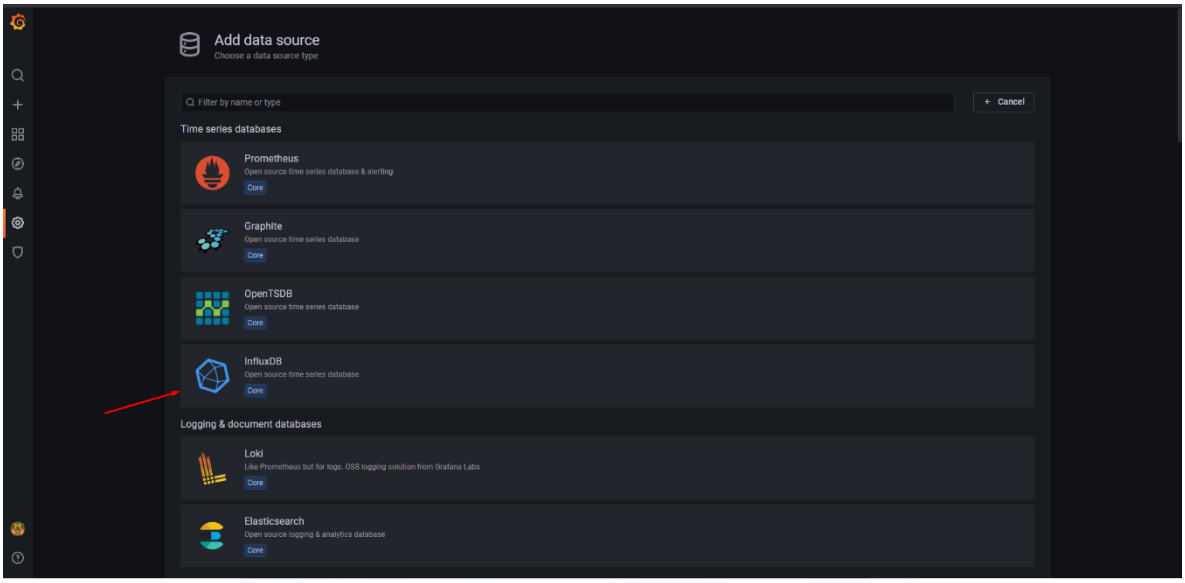


Ilustración 106: Seleccionaremos el tipo influxDB.

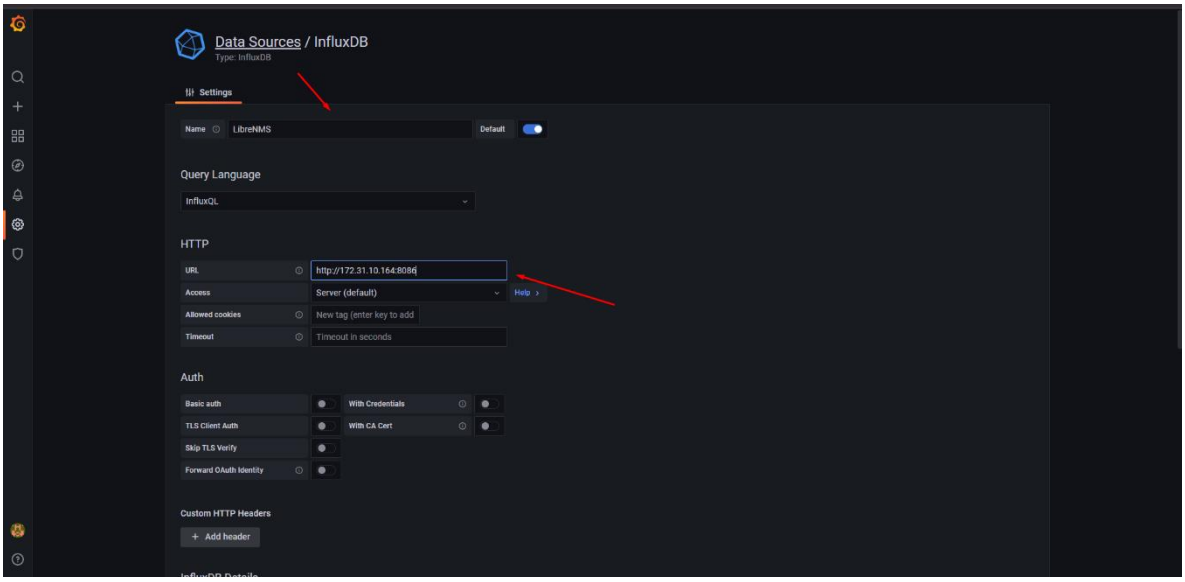


Ilustración 107: Configuramos un nombre al data source así como la URL de conexión.

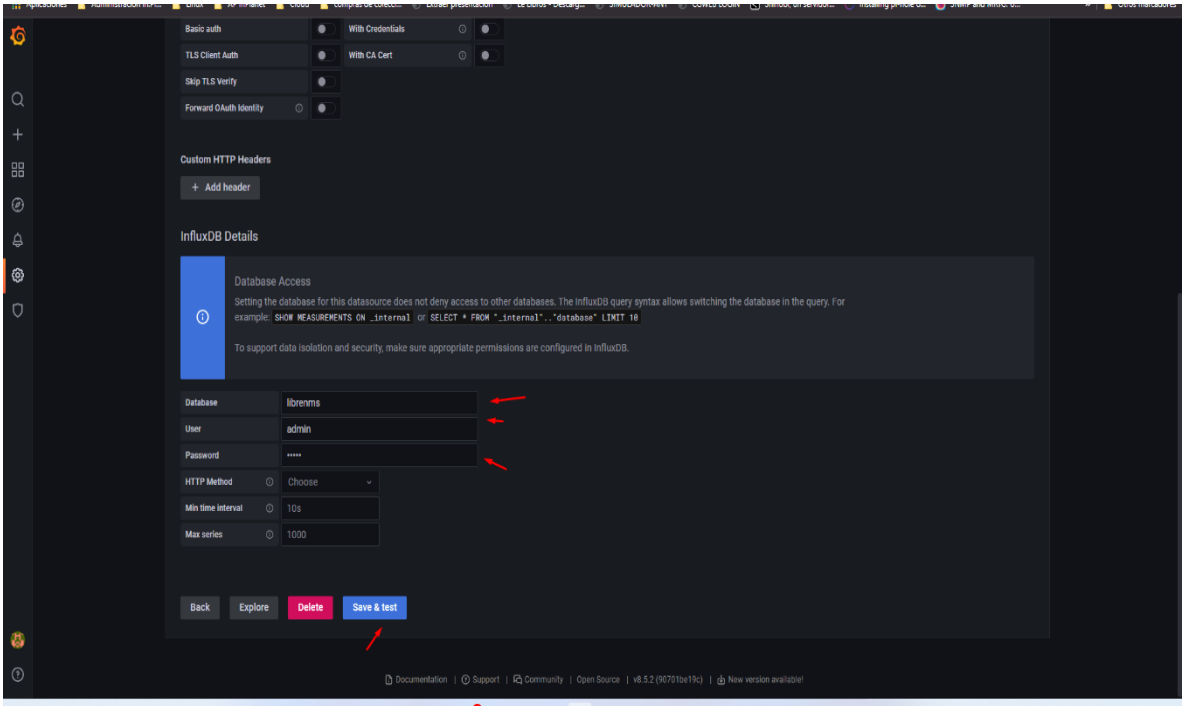


Ilustración 108: Configuraremos la BD el usuario y contraseña previamente creados.

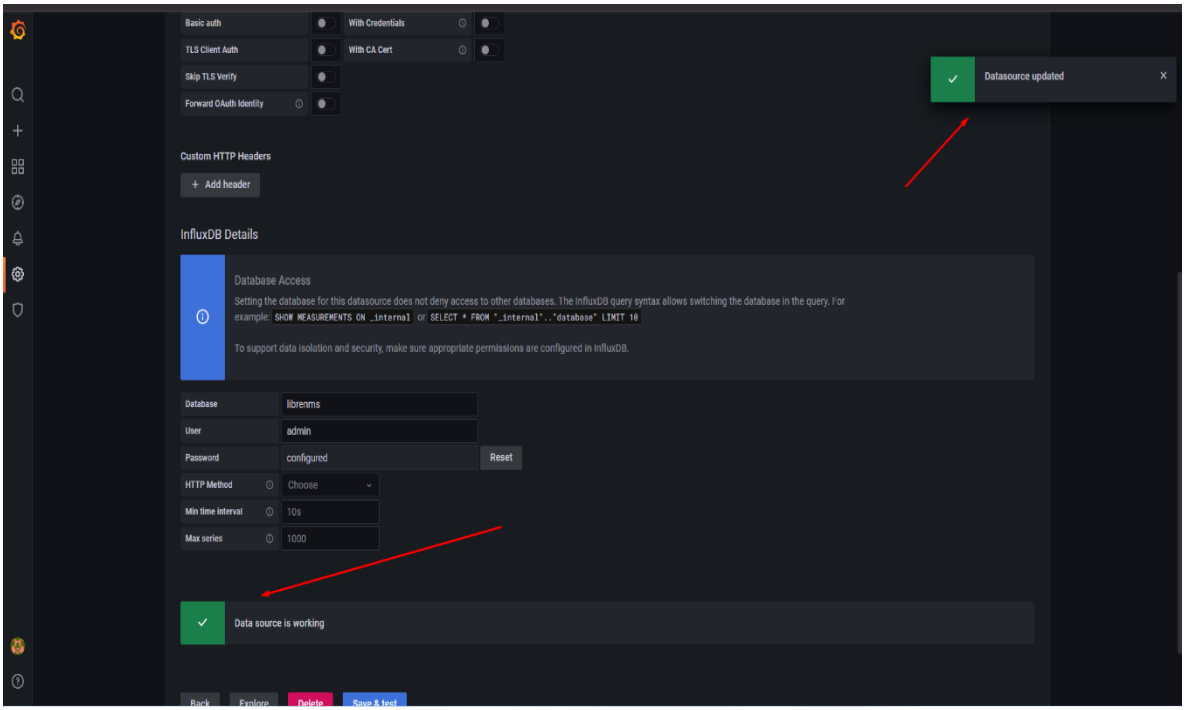


Ilustración 109: Dando click en save & test, podremos corroborar que la conexión sea exitosa, así como guardar los cambios.

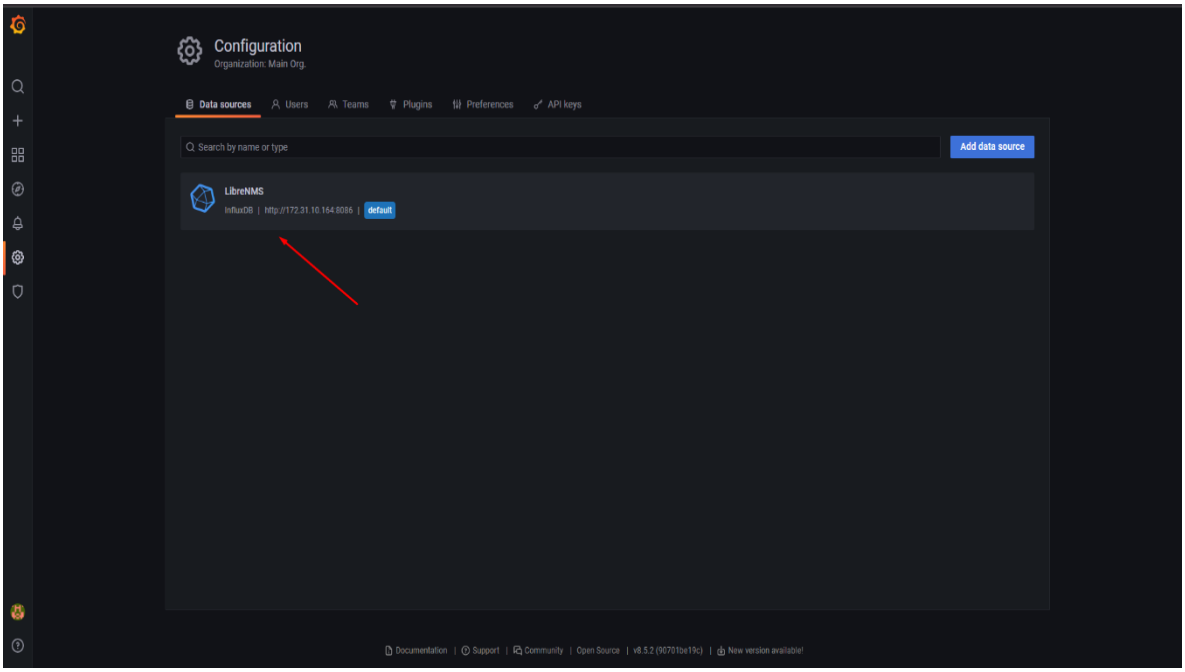


Ilustración 110:Veremos el data soruce creado.

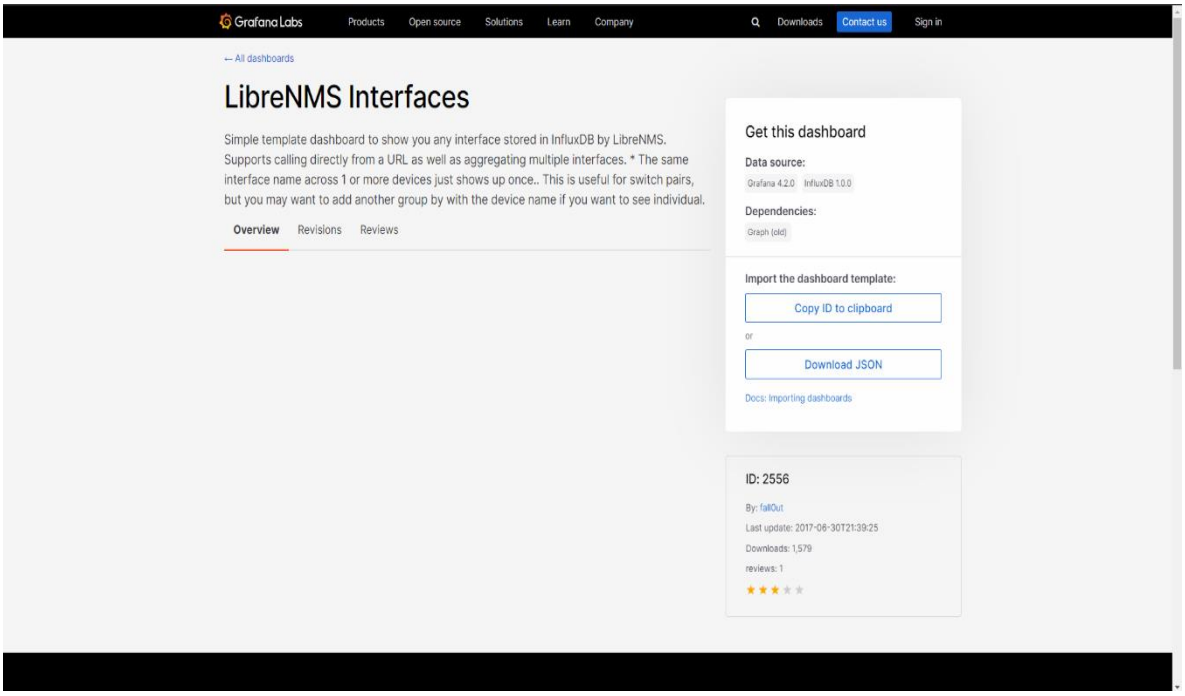


Ilustración 111: Usaremos un dashboard con ID 2556.

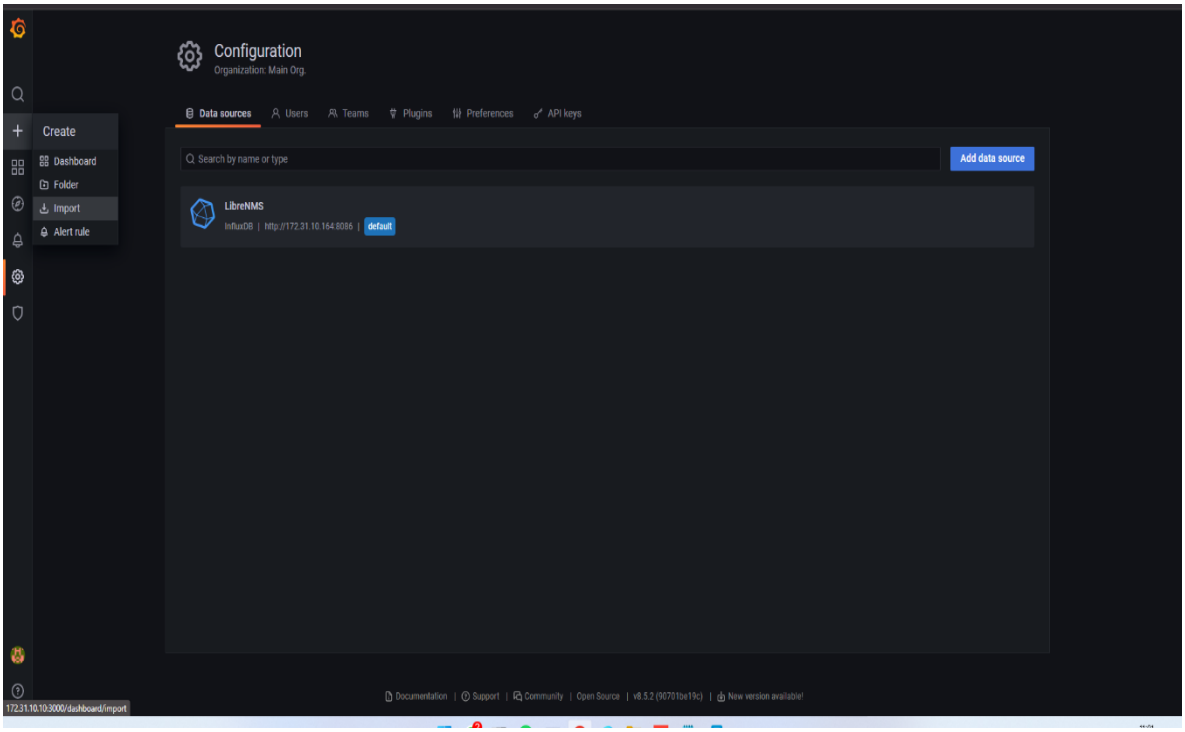


Ilustración 112: En el menu Create seleccionamos Import.

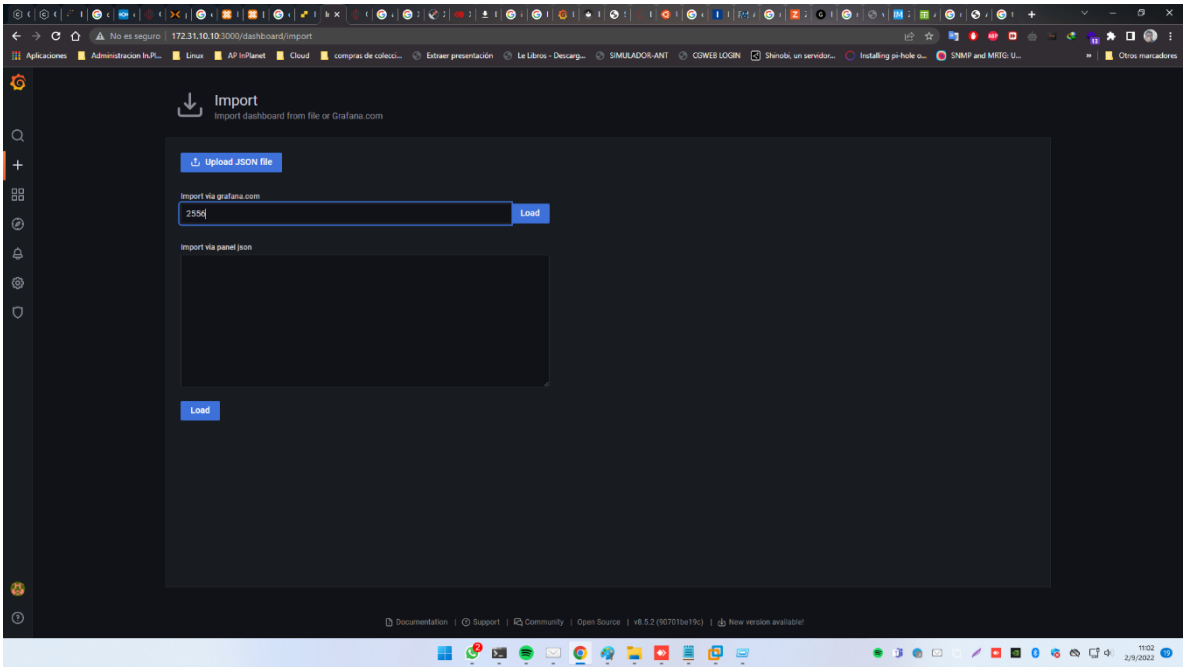


Ilustración 113: Ingresaremos el ID del dashboard.

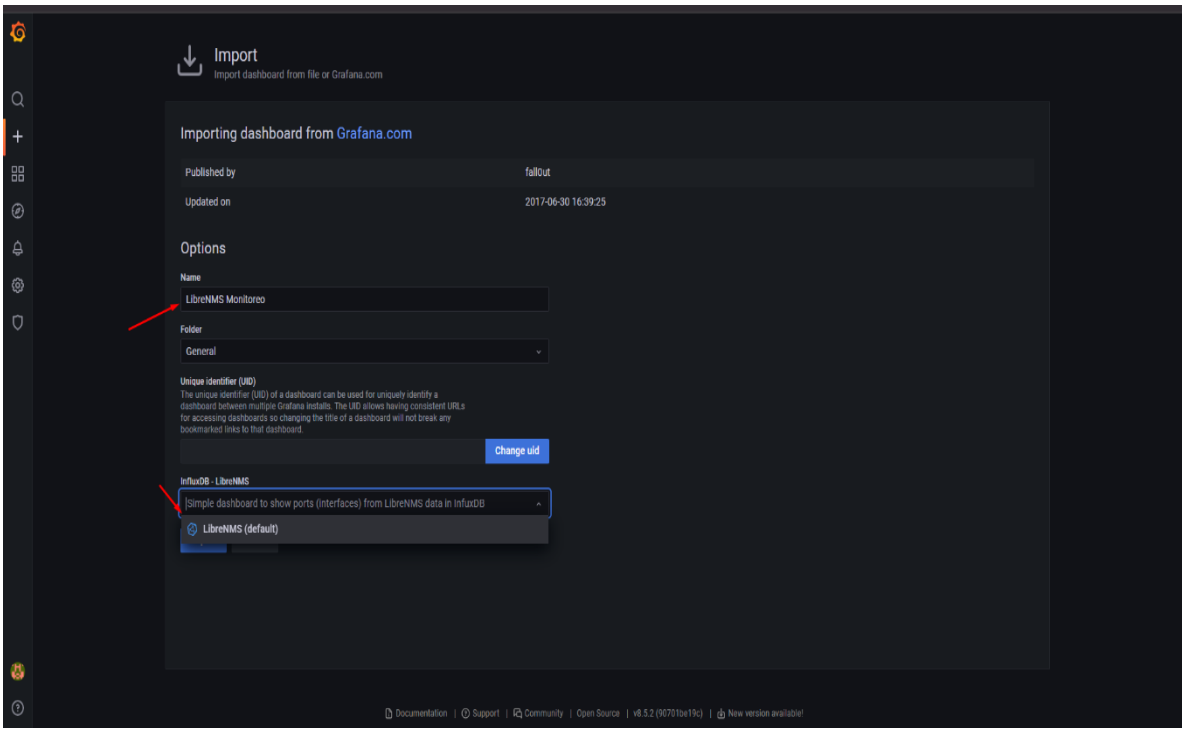


Ilustración 114: Indicaremos un nombre y el data source de influx.

Ilustración 115: Nos cargara los datos de consumo de red de los dispositvos añadidos.

Posterior a la integración de Grafana y LibreNMS, se creó un plan de contingencia para prever situaciones adversas que pongan en riesgo la información que maneja la plataforma de monitoreo a cargo del departamento NOC

3.5.13. Plan de contingencia:

Uno de las interrogantes que tuvo el departamento de NOC de In.Planet S.A. fue que sucede en caso de que el servidor de LibreNMS sufra alguna avería irreparable, o si en el peor de los casos el HDD o la VM son dañados o borrados accidentalmente. Para aquello se contempló la realización de un plan de contingencia o DRP, en el cual, mediante el uso de script de bash se logrará mantener un respaldo de la BD de LibreNMS así como de los archivo RRD. Para la realización de esta tarea se usaron los siguientes componentes:

- Lenguaje BASH
- Servidor de respaldos de In.Planet S.A.(OpenMediaVault)
- Editor Visual Studio Code
- Samba

Mediante el uso de las herramientas anteriormente descritas se logró obtener los respaldos necesarios para la restauración del sistema LibreNMS, lo cual se puede verificar en las ilustraciones *84 hasta la 82.*

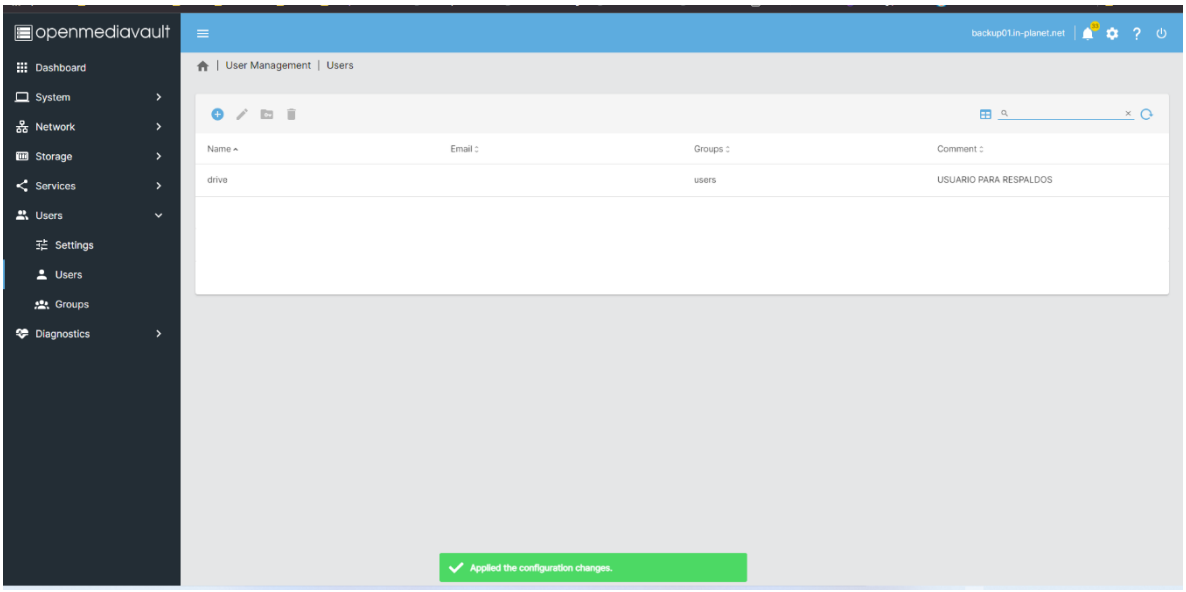


Ilustración 116: Se creó el usuario drive.

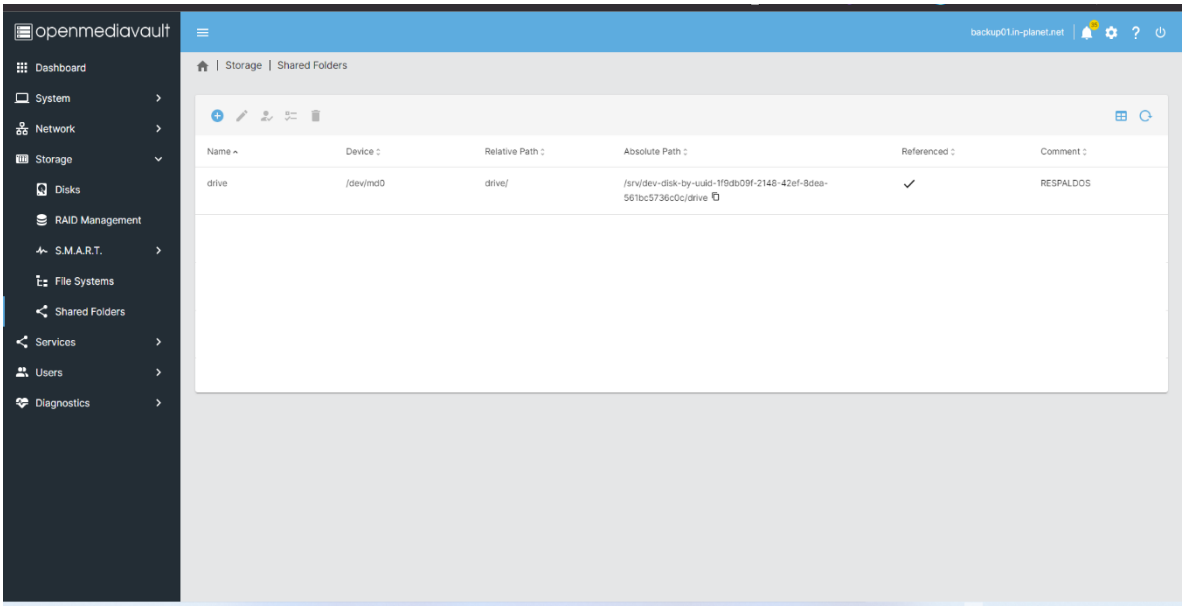


Ilustración 117: Se creó una carpeta compartida llamada drive donde se realizará el respaldo.

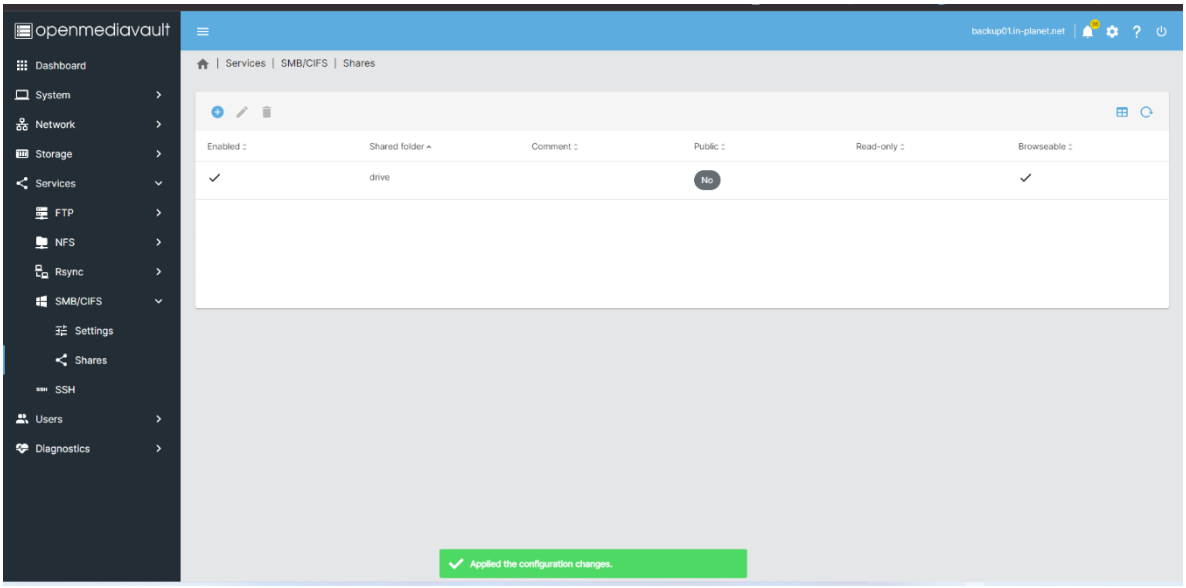


Ilustración 118: Se compartió el recurso usando SMB.

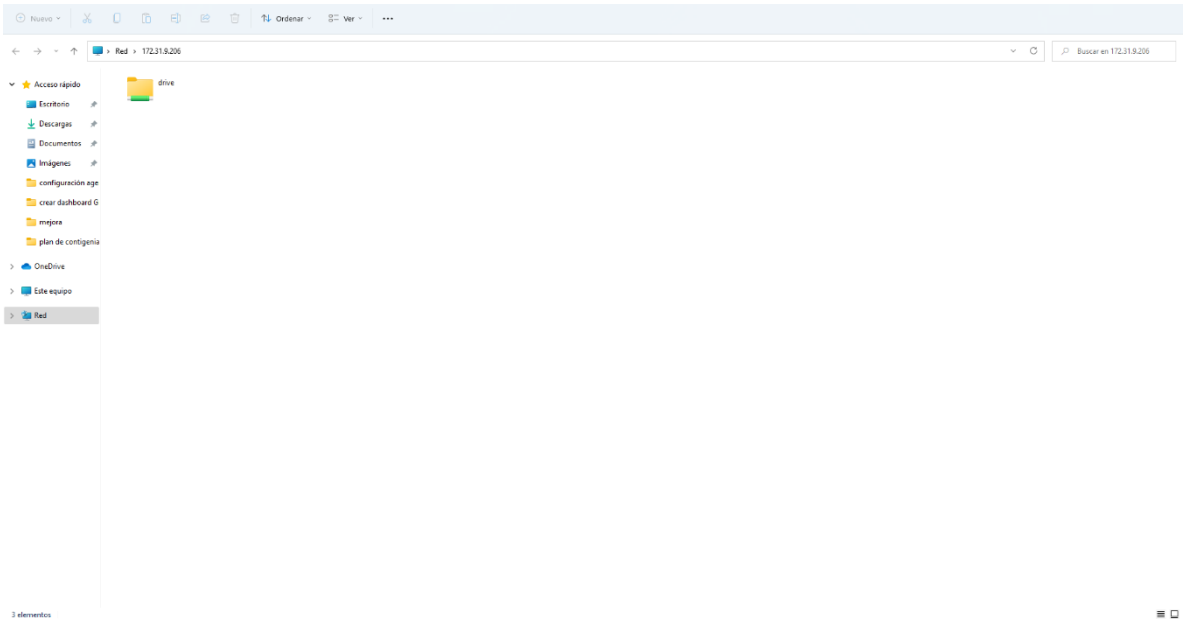


Ilustración 119: Se ingresó a la ruta del servidor y se pudo observar la carpeta recién creada.

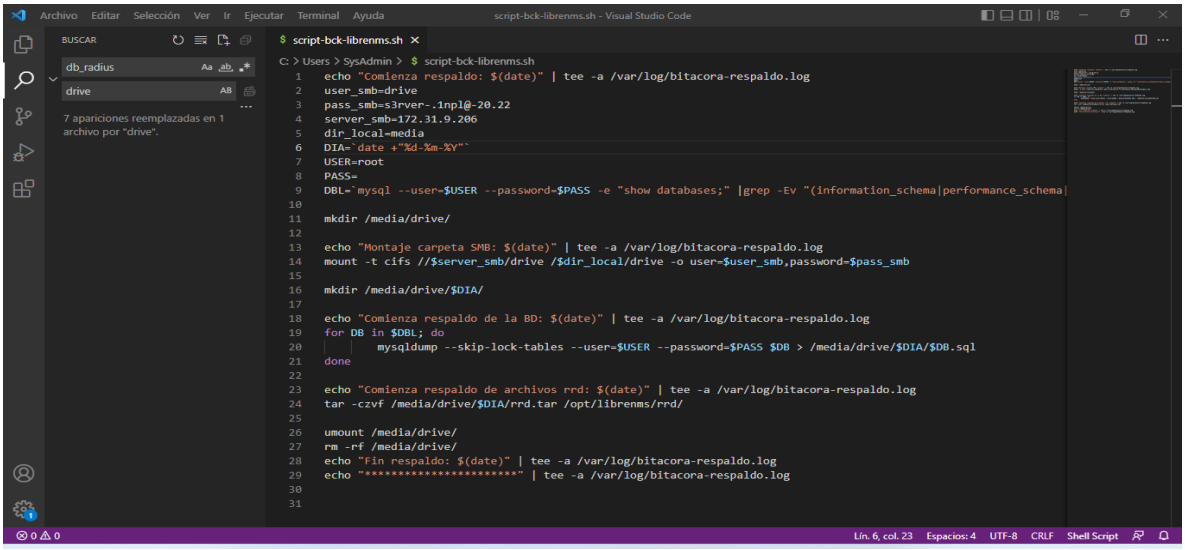


Ilustración 120: Scrip de bash programado usando Visual Studio Code.

En el servidor de Librenms se instaló el paquete compatibilidad CIFS para samba lo que permitirá montar la carpeta compartida desde el terminal en nuestro Librenms:

root@librenms01:~# apt install cifs-utils -y

Se creo un archivo de log para guardar una bitacora de la tarea:

root@librenms01:~# touch /var/log/bitacora-respaldo.log

Se le dio permisos al archivo de log para que pueda ser editado:

root@librenms01:~# chmod 777 /var/log/bitacora-respaldo.log

Se creo un archivo bash con extension .sh donde se pegará el código del script desarrollado para el respaldo de la BD y archivos RRD:

root@librenms01:~# nano script.sh

```
echo "Comienza respaldo: $(date)" | tee -a /var/log/bitacora-respaldo.log
user_smb=drive
pass_smb=s3rver-.1npl@-20.22
server_smb=172.31.9.206
dir_local=media
DIA=`date +"%d-%m-%Y"`
USER=root
PASS=
DBL=`mysql --user=$USER --password=$PASS -e "show databases;" |grep -Ev
"(information_schema|performance_schema|mysql|sys|phpmyadmin|Database)"`
```

```
mkdir /media/drive/

echo "Montaje carpeta SMB: $(date)" | tee -a /var/log/bitacora-respaldo.log
mount -t cifs //$server_smb/drive /$dir_local/drive -o
user=$user_smb,password=$pass_smb

mkdir /media/drive/$DIA/

echo "Comienza respaldo de la BD: $(date)" | tee -a /var/log/bitacora-respaldo.log
for DB in $DBL; do
      mysqldump --skip-lock-tables --user=$USER --password=$PASS $DB >
/media/drive/$DIA/$DB.sql
done

echo "Comienza respaldo de archivos rrd: $(date)" | tee -a /var/log/bitacora-
respaldo.log
tar -czvf /media/drive/$DIA/rrd.tar /opt/librenms/rrd/

umount /media/drive/
rm -rf /media/drive/
echo "Fin respaldo: $(date)" | tee -a /var/log/bitacora-respaldo.log
echo "**********" | tee -a /var/log/bitacora-respaldo.log
```

Se dio permisos al archivo de script:

root@librenms01:~# chmod 777 script.sh

Se realizo una prueba, para comprobar que el script se ejecute de manera correcta:

root@librenms01:~# ./script.sh

Ilustración 121: Se ejecuto el script observando que el respaldo a comenzado.

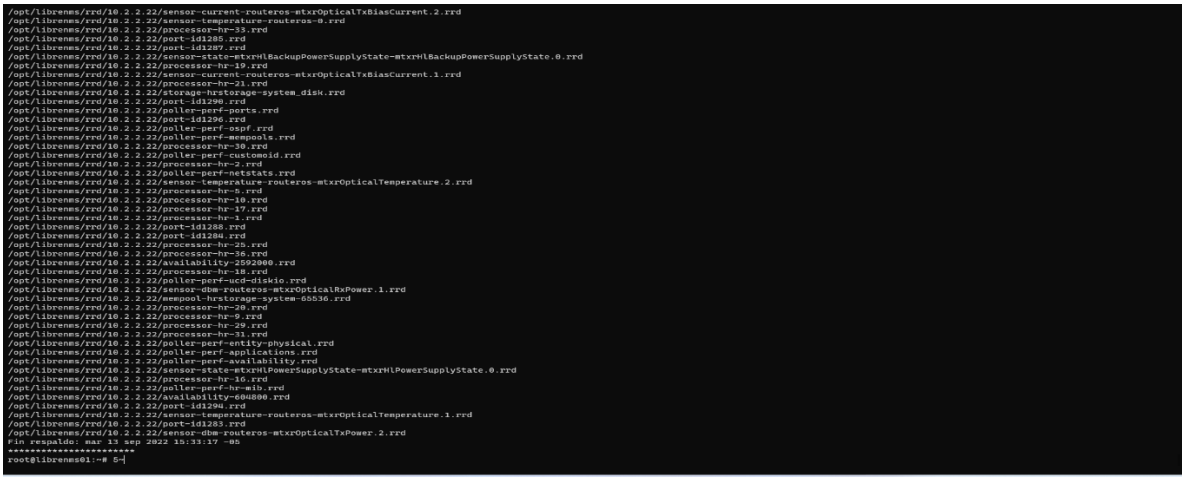


Ilustración 122: Transcurrido el tiempo podemos observar que el script llego a su fin.

Ilustración 123: Se realizo un cat para observar el log.}

Se creo un cron para que el respaldo se realice de manera automática todos los días a las 10 de la noche:

root@librenms01:~# crontab -e

00 22 * * * root sh /root/script.sh

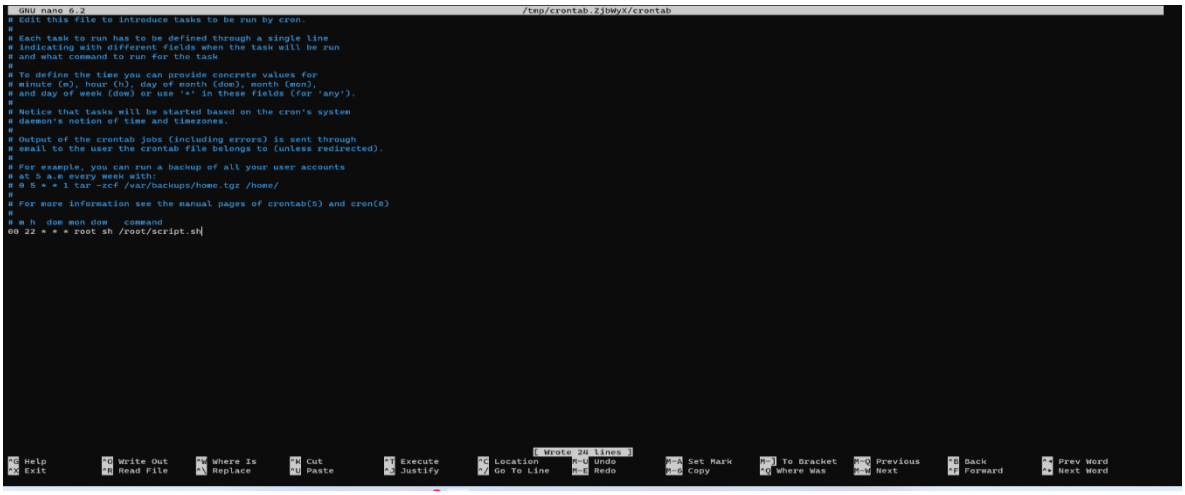


Ilustración 124: El archivo de crontab quedo de la siguiente manera.

3.6. Fase V: Operar:

En esta fase se procedió a poner en marcha el sistema de monitoreo, con el fin de verificar su rendimiento y así poder realizar futuras optimizaciones.

En esta fase se procedió a configurar el agente SNMP en los dispositivos del core a manipular para posteriormente añadirlos a LibreNMS, una vez se configuró los agentes y LibreNMS traía datos y la BD de influxdb se alimentaba con los datos se procedió a la creación de los dashboard con acompañamiento de un Ingeniero del área de networking para realizar la creación de los mismos según sus necesidades. En las *ilustraciones 93 hasta la 115* se encuentran evidencia del proceso.

3.6.1. Para los servidores configuraremos lo siguiente:

Instalaremos los paquetes necesarios para derivadas de Debian:

root@srv-ubuntu:~# apt-get install curl wget build-essential snmpd -y

Para derivadas de Red Hat Linux:

root@centos:~# yum -y install net-snmp net-snmp-utils

Descargaremos un plugin que le indicara a LibreNMS que logo usar en el dashboard según nuestra distribución Linux:

```
root@mail:~# curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp/distro
```

Ilustración 125: Comando para la instalación del plugin distro.

Le damos permiso de ejecución al archivo descargado:

root@mail:~# chmod +x /usr/bin/distro

Modificamos el archivo snmpd.conf:

```
  GNU nano 4.8                                                  /etc/snmp/snmpd.conf
com2sec readonly  default         inet_snmp

group MyROGroup v2c        readonly
view all    included  .1                               80
access MyROGroup ""      any       noauth    exact  all    none   none

syslocation Servidores, DC Matriz, Milagro, Ecuador
syscontact Leonardo Pina <lpina@in-planet.net>

#OS Distribution Detection
extend distro /usr/bin/distro
```

Ilustración 126: Modificación del archivo snmpd con la configuración necesario para su funcionamiento

Reiniciamos el servicio de snmp:

root@mail:~# systemctl restart snmpd

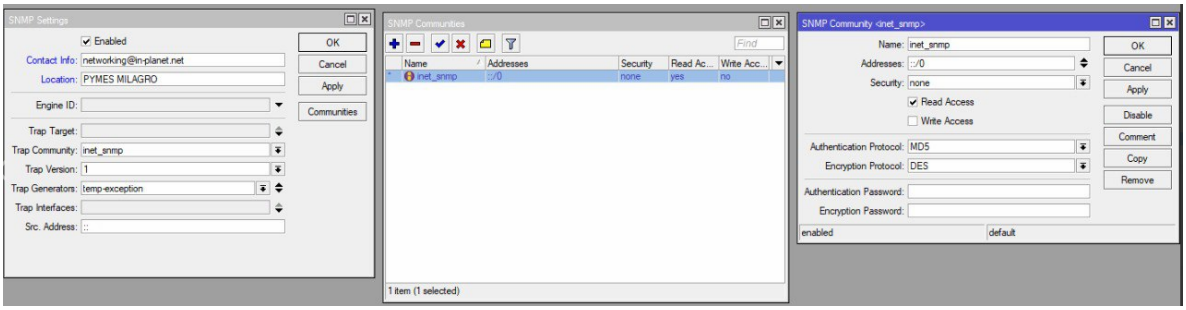


Ilustración 127: En Mikrotik añadimos el permiso a la IP, y configuramos la comunidad.

Ilustración 128: En Fortinet realizamos la configuración de la IP del LibreNMS y configuramos la comunidad.

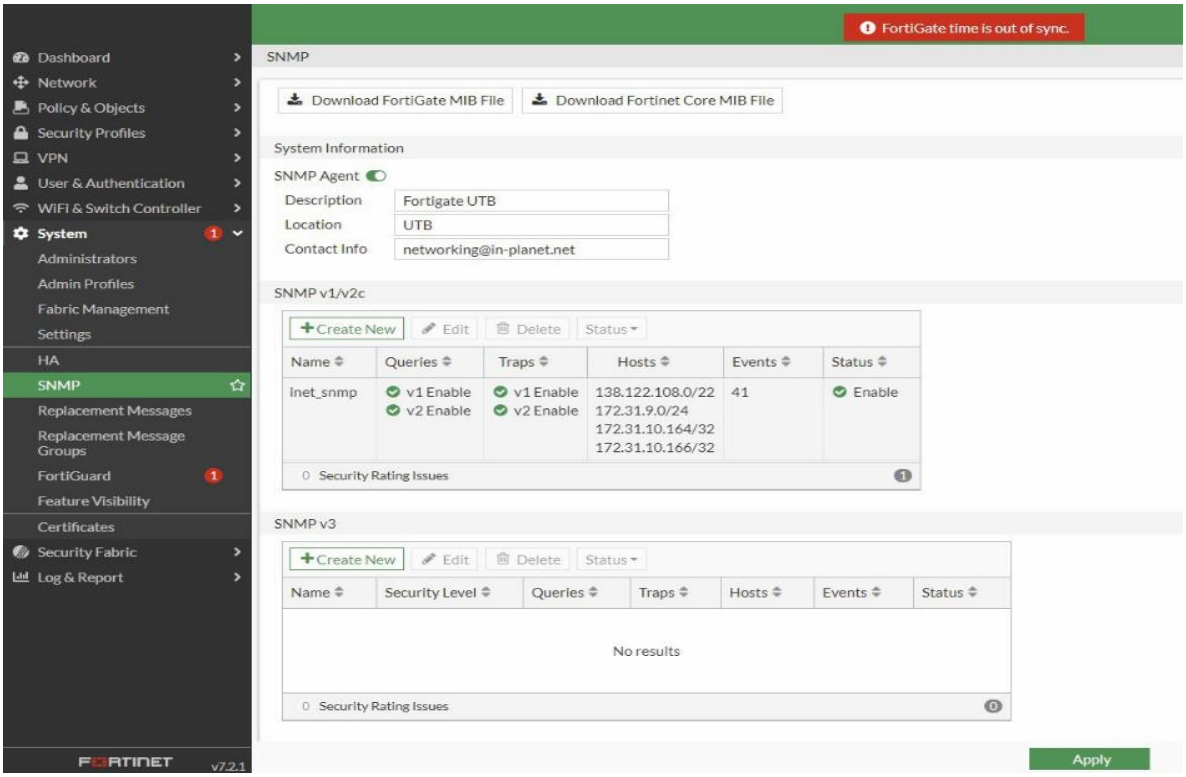


Ilustración 129: Fortinet, podemos ver la configuración SNMP realizada.

Ilustración 130: Podemos añadir desde la consola con el comando. /lnms device:add --v2c - "nombre_comunidad" "ip_dispositivo".

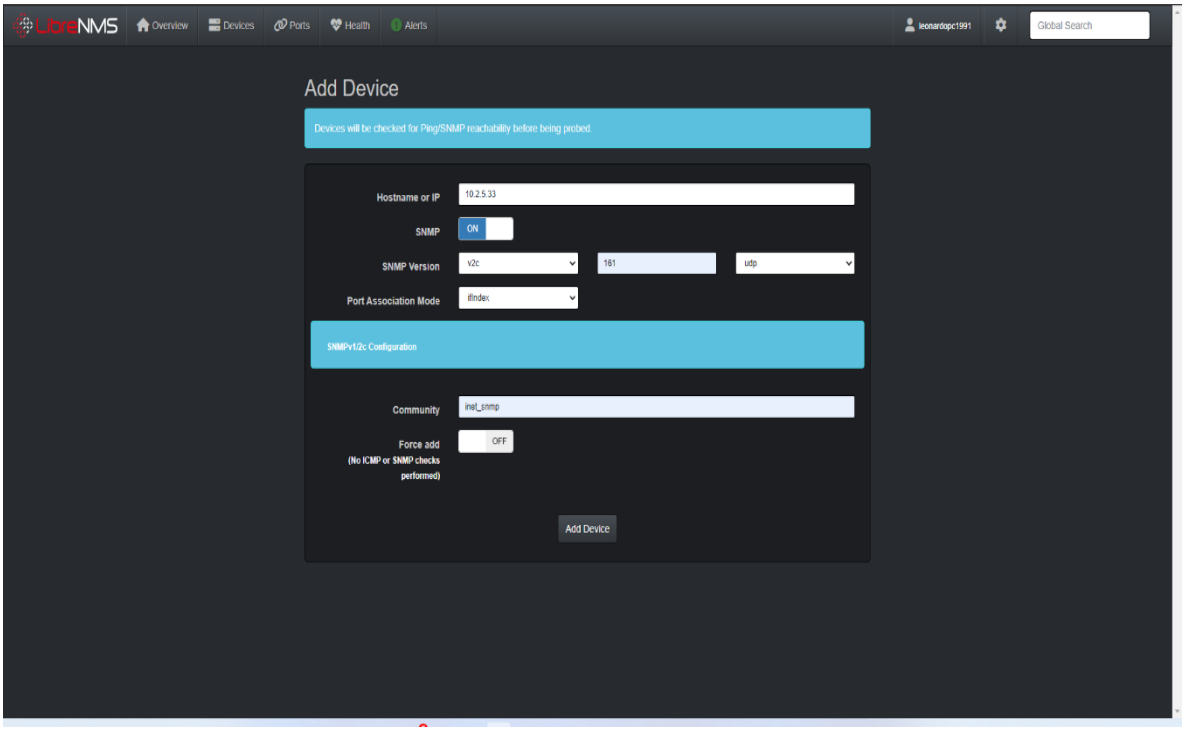


Ilustración 131: Desde la consola en el menú Devices luego Add Devices, y completemos los datos como IP, puerto y comunidad.

3.6.2. Creación del Dashboard LibreNMS:

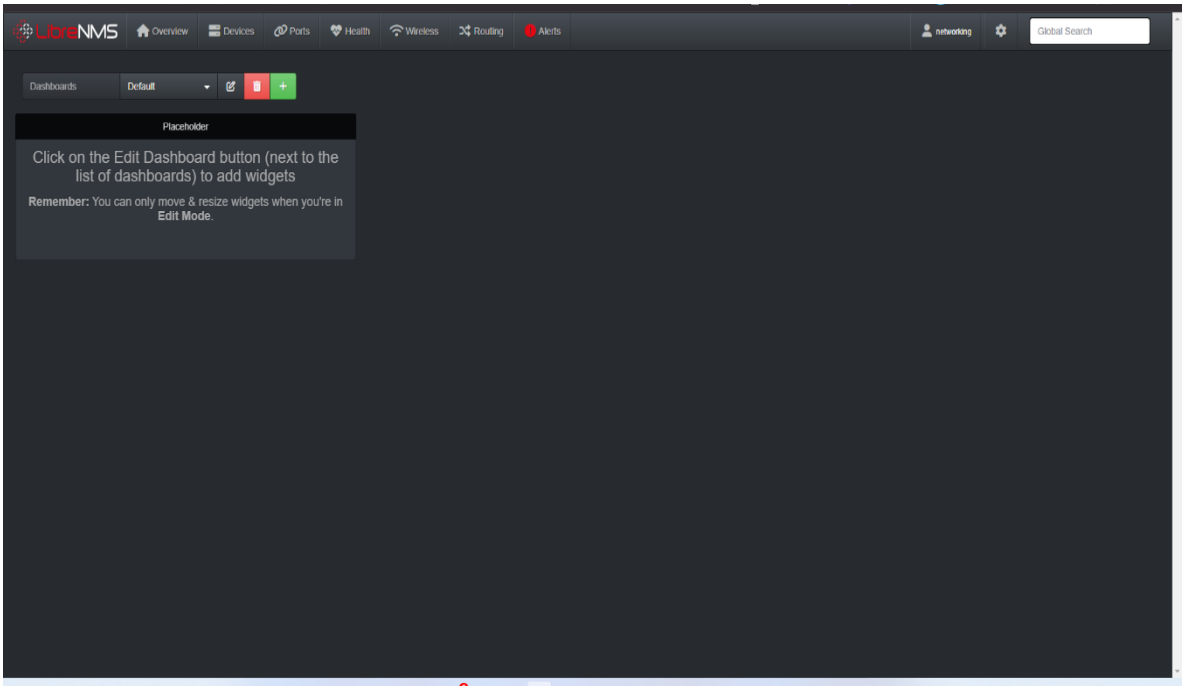


Ilustración 132: En la pantalla principal tendremos opciones para el dashboard.

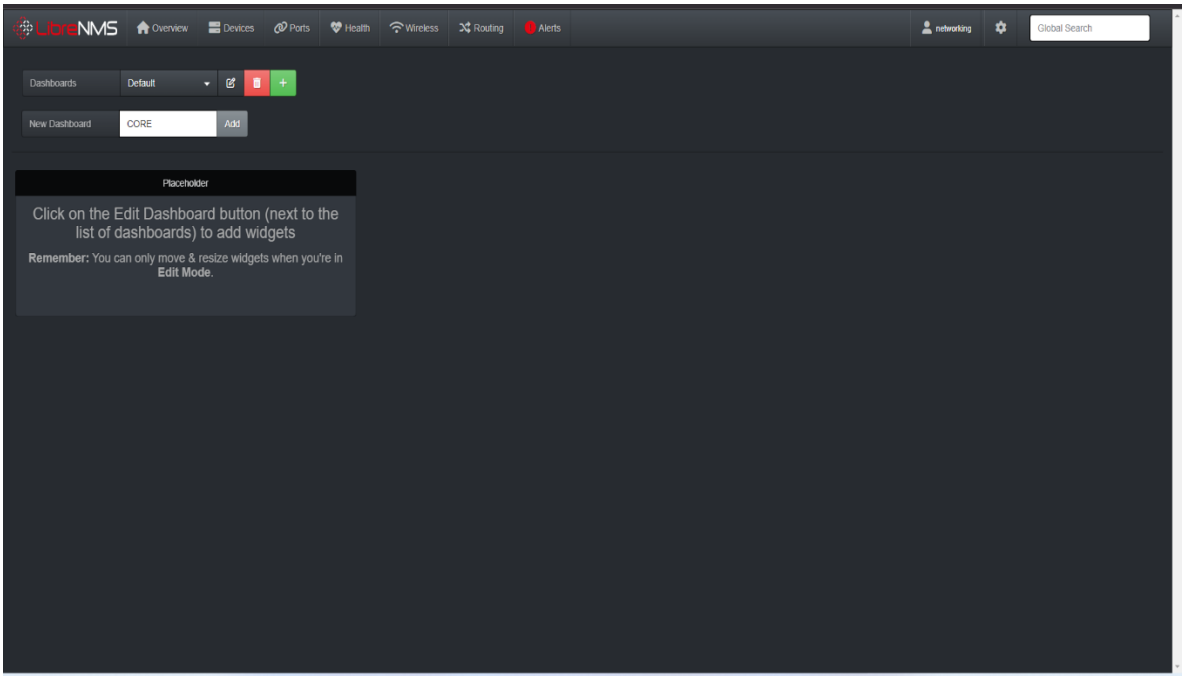


Ilustración 133: Crearemos uno nuevo llamado core.

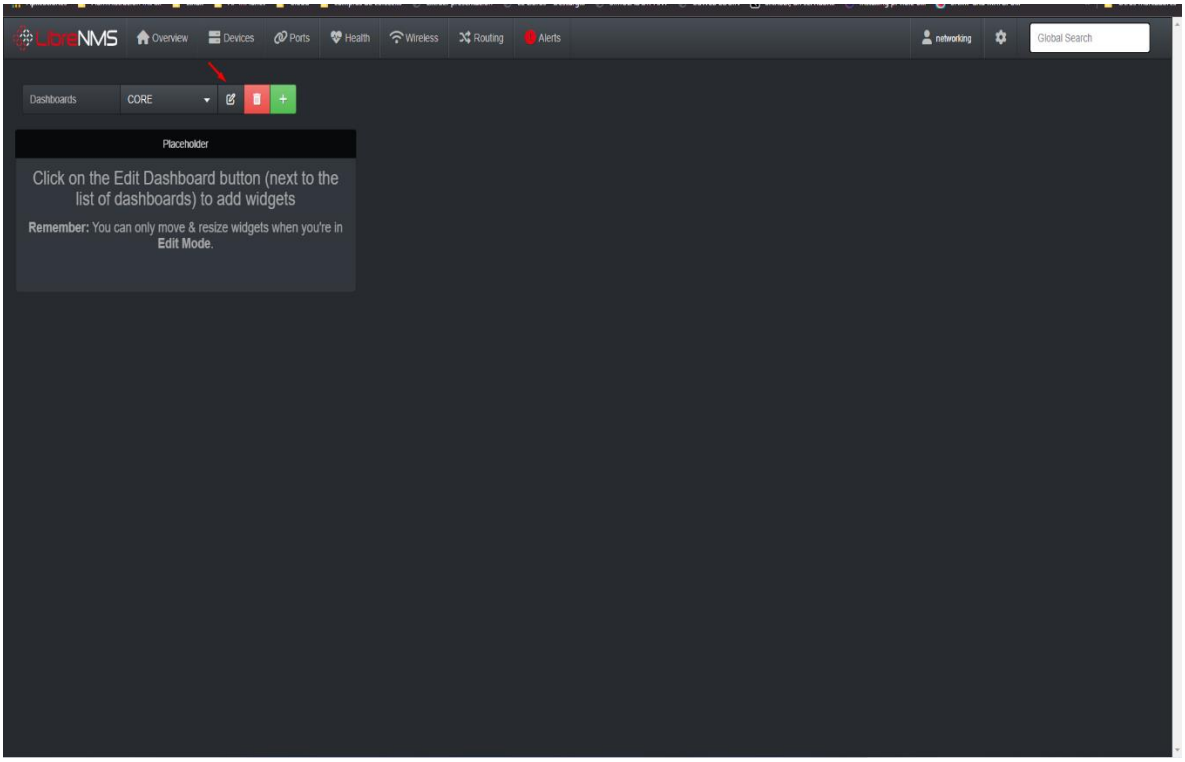


Ilustración 134: Una vez creado lo editaremos.

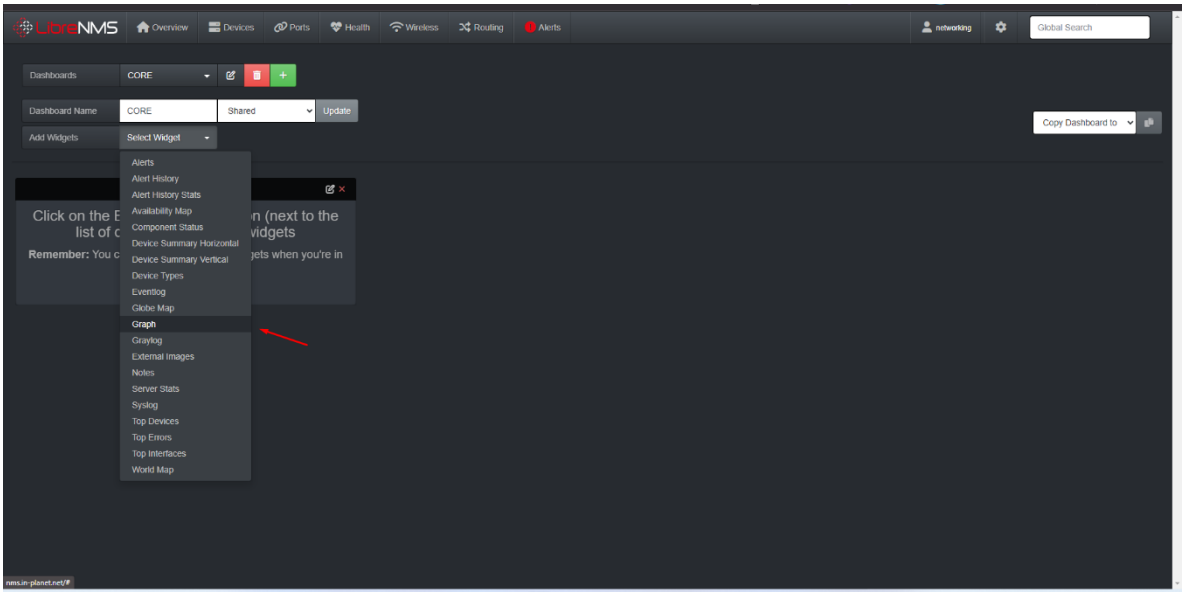


Ilustración 135: En add widget, elejiremos el que necesitemos.

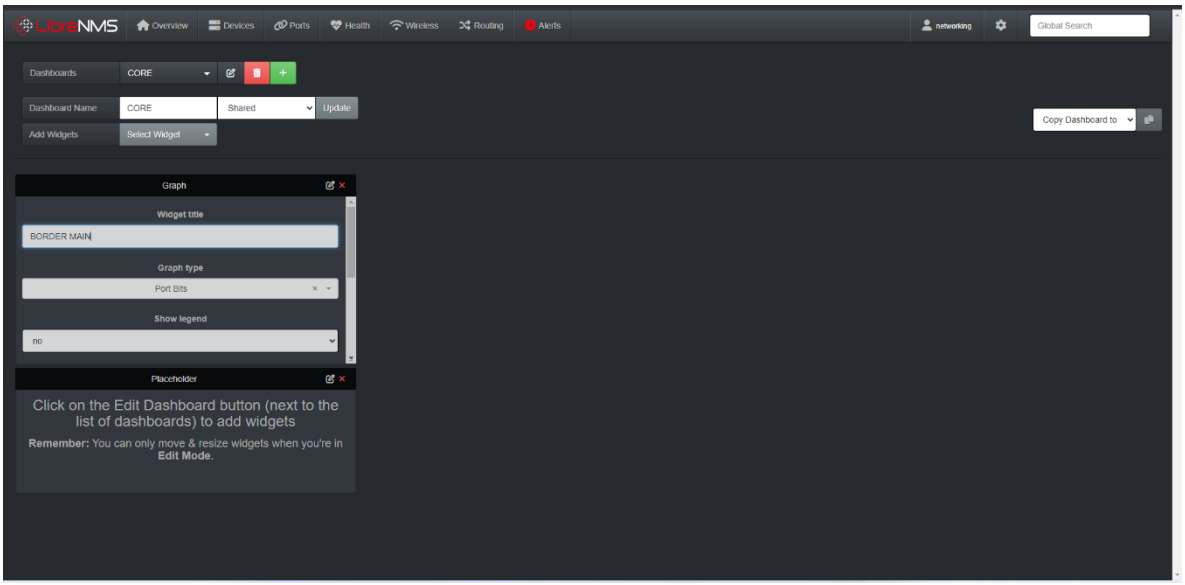


Ilustración 136: Indicaremos un nombre al widget.

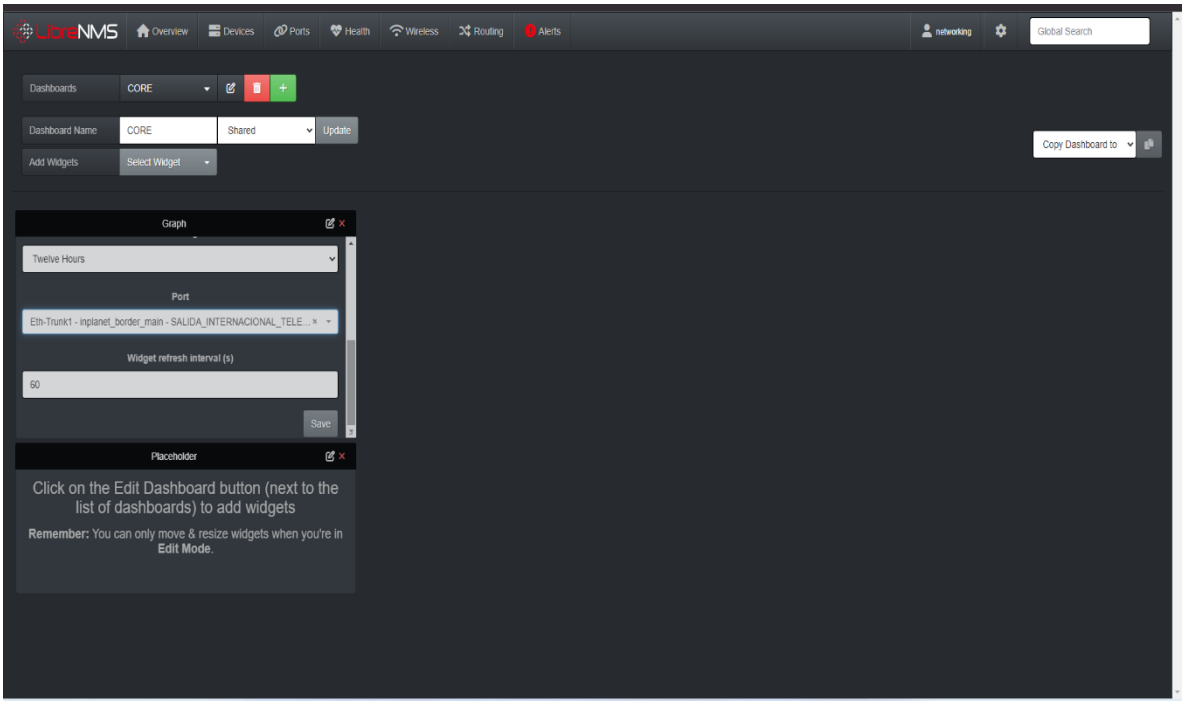


Ilustración 137: Elegiremos el puerto que deseemos monitorizar y daremos clic en save.

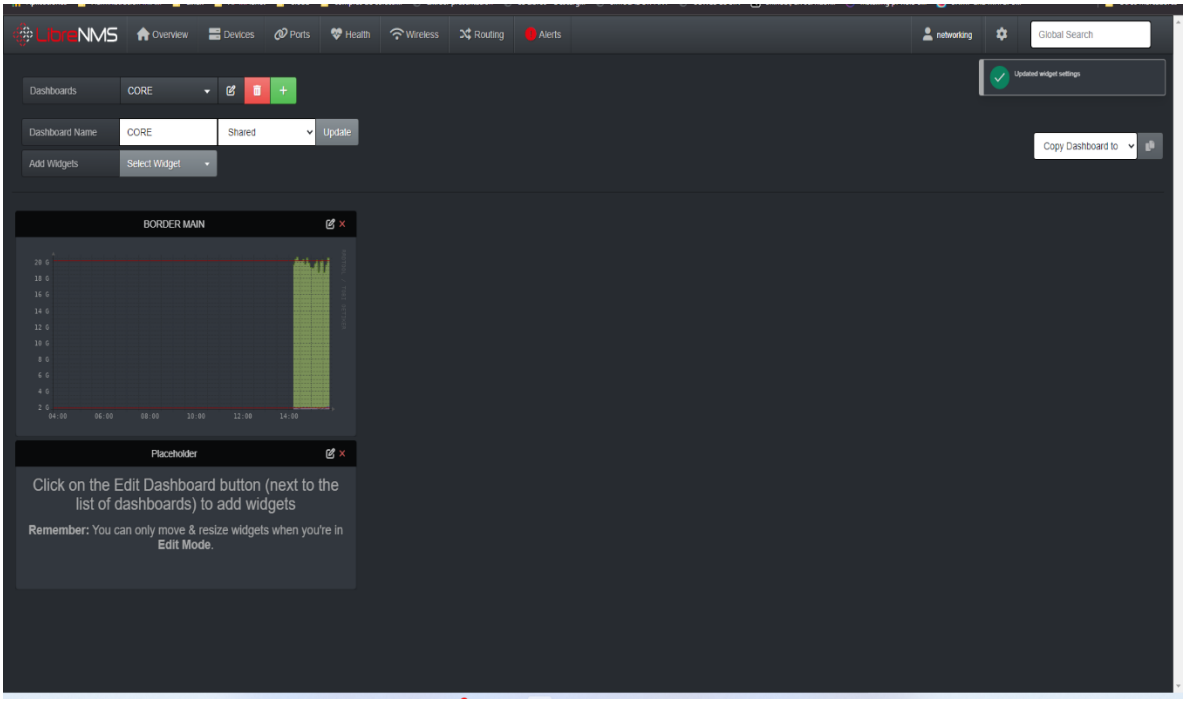


Ilustración 138: El widget empieza a graficar.

3.6.3. Creación del Dashboard Grafana:

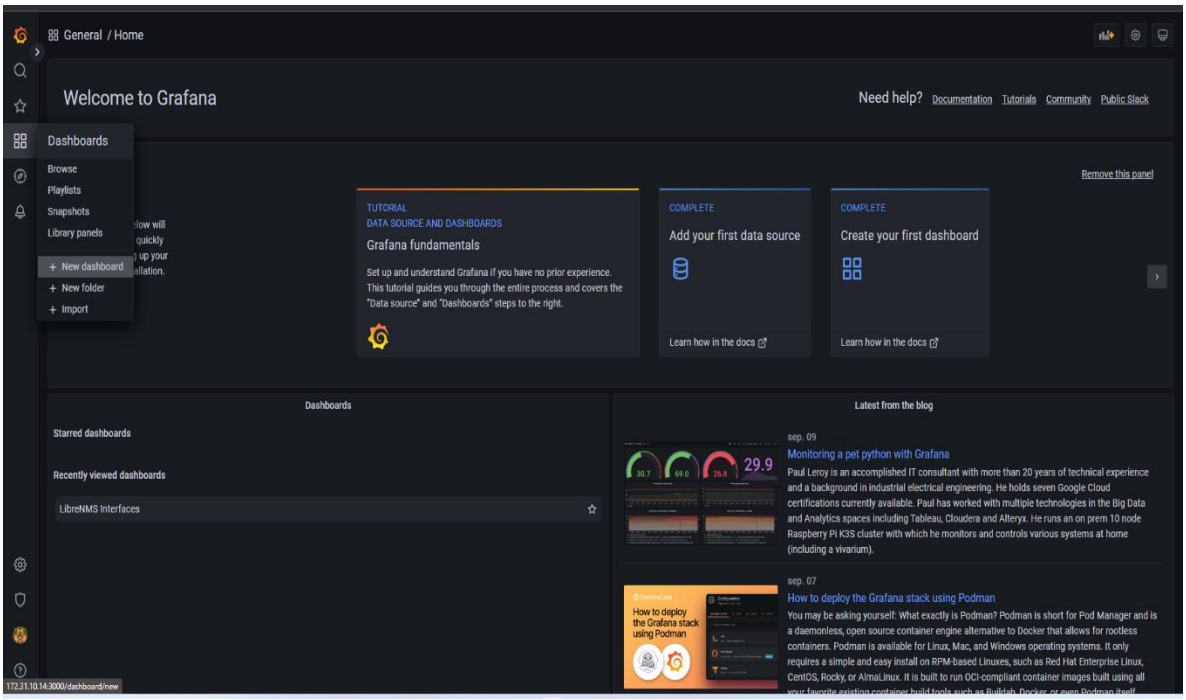


Ilustración 139: En el menú dashboard crearemos un nuevo dashboard.

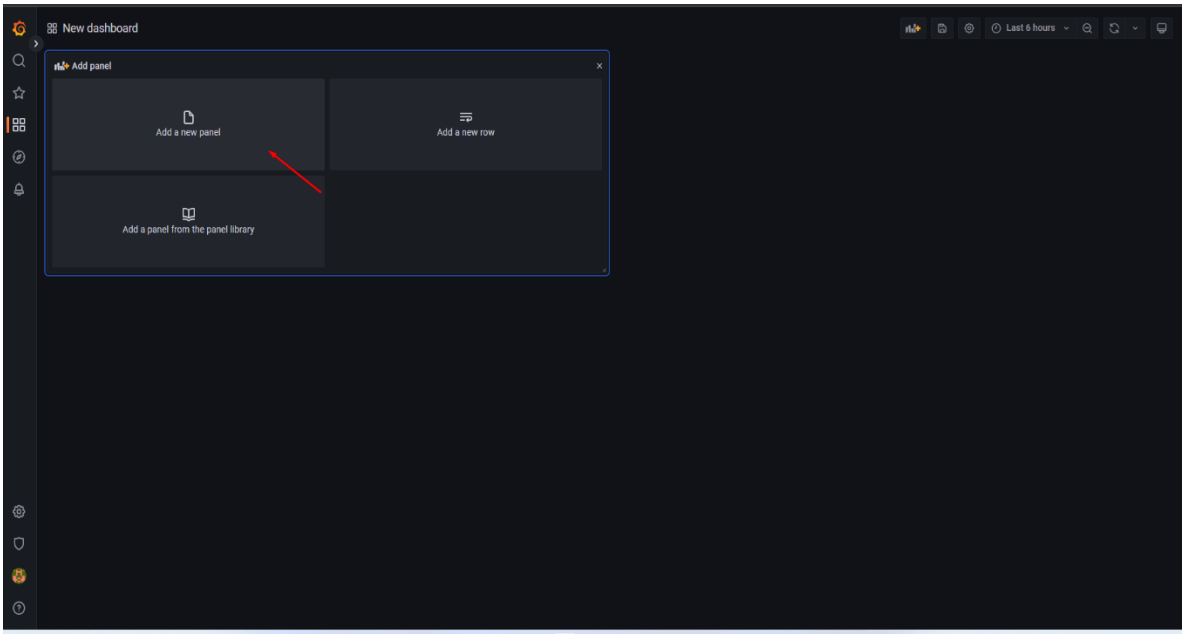


Ilustración 140: Seleccionamos Add a new panel.

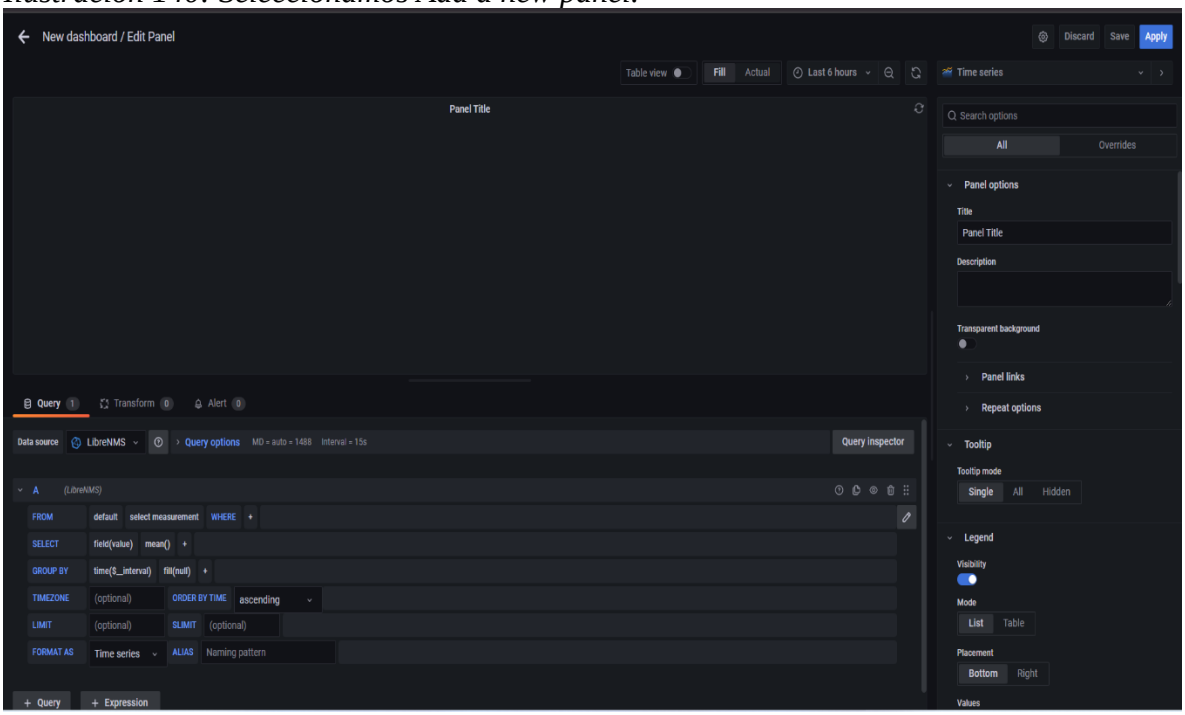


Ilustración 141: Nos mostrará el editor.

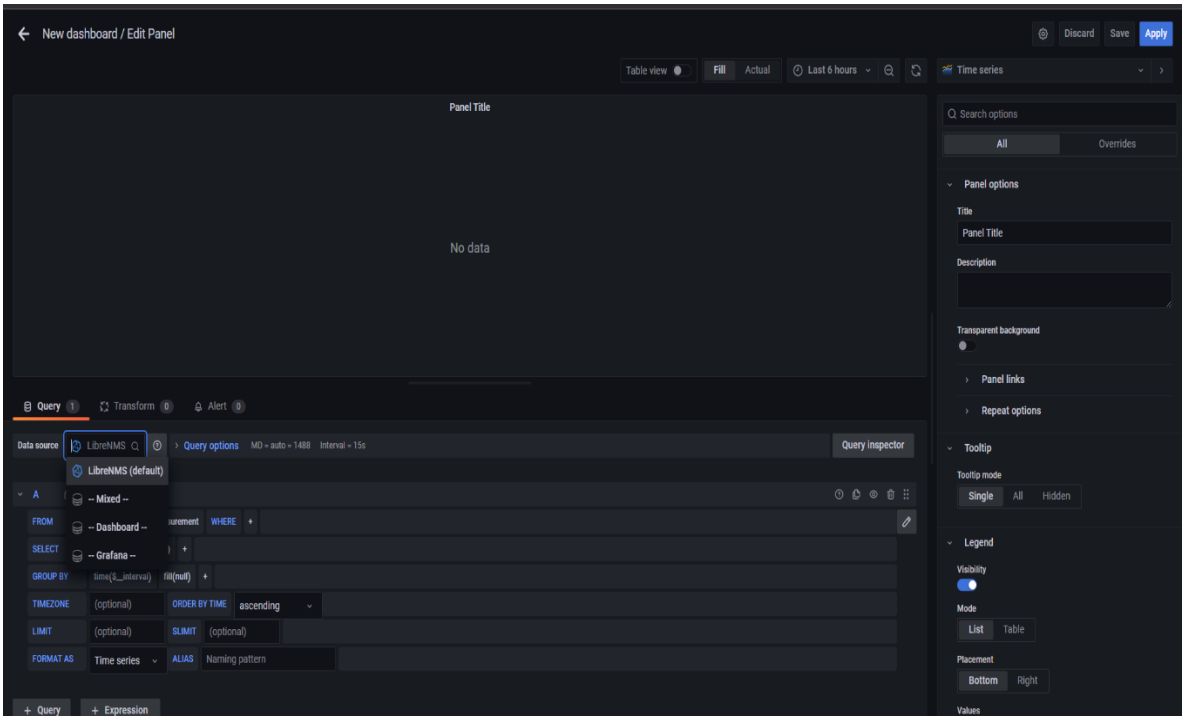


Ilustración 142: Seleccionaremos el data source anteriormente creado de influxdb.

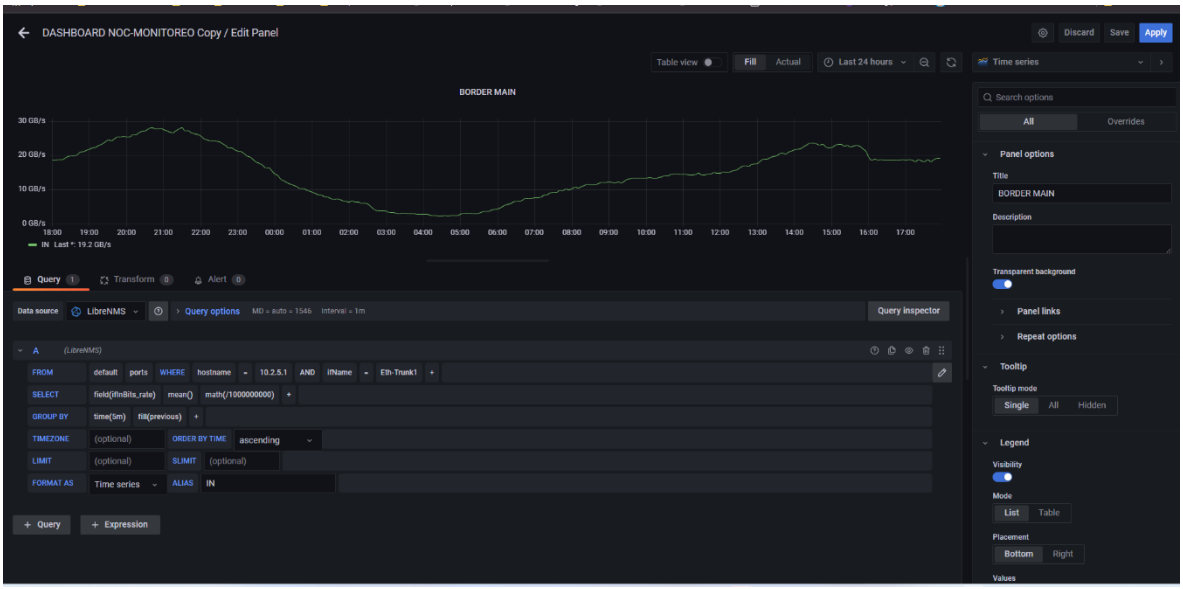


Ilustración 143: Configuraremos la consulta, según lo requerido.

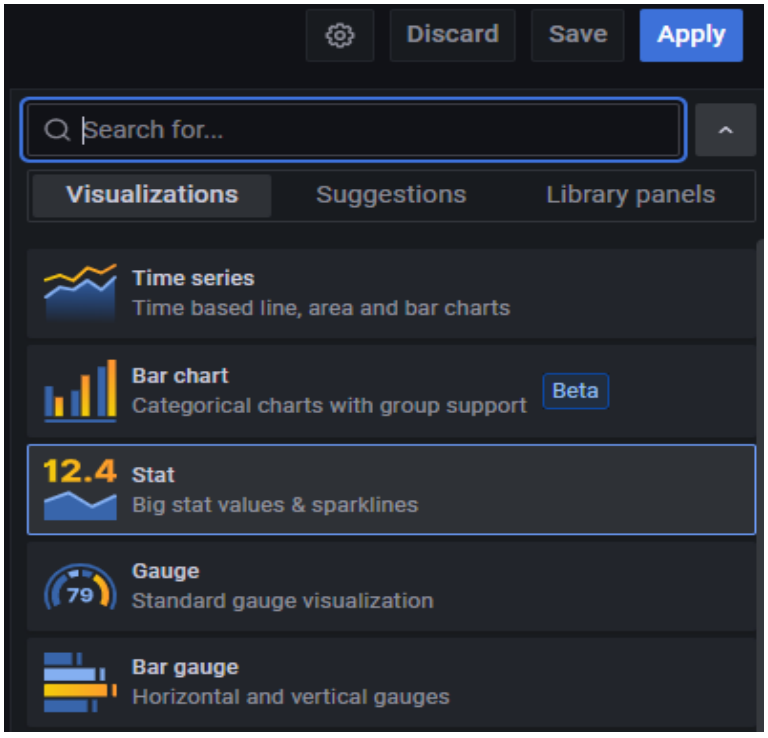


Ilustración 144: Dentro del tipo de visualización, seleccionamos time series.

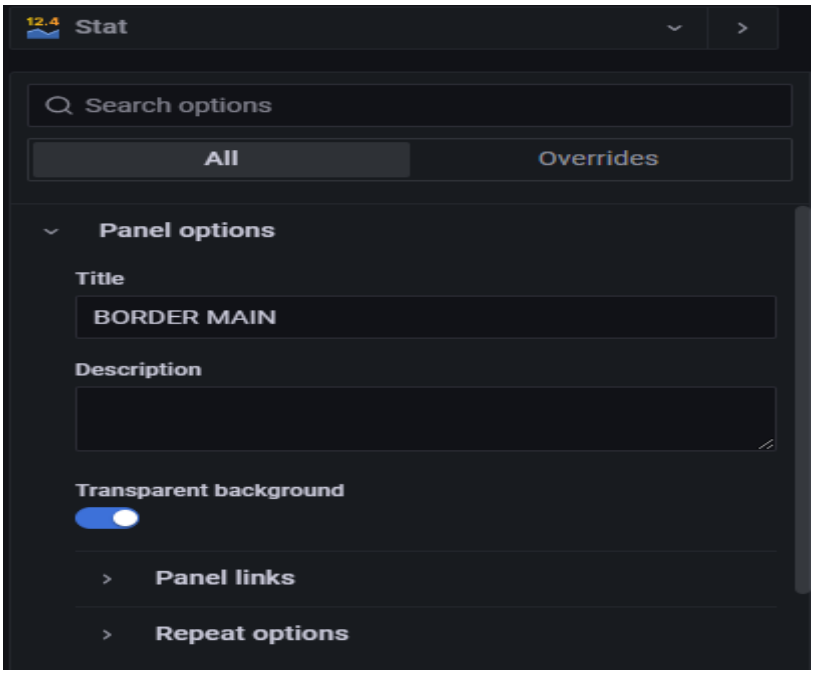


Ilustración 145: Configuramos un Title y selecionamos Transparent background.

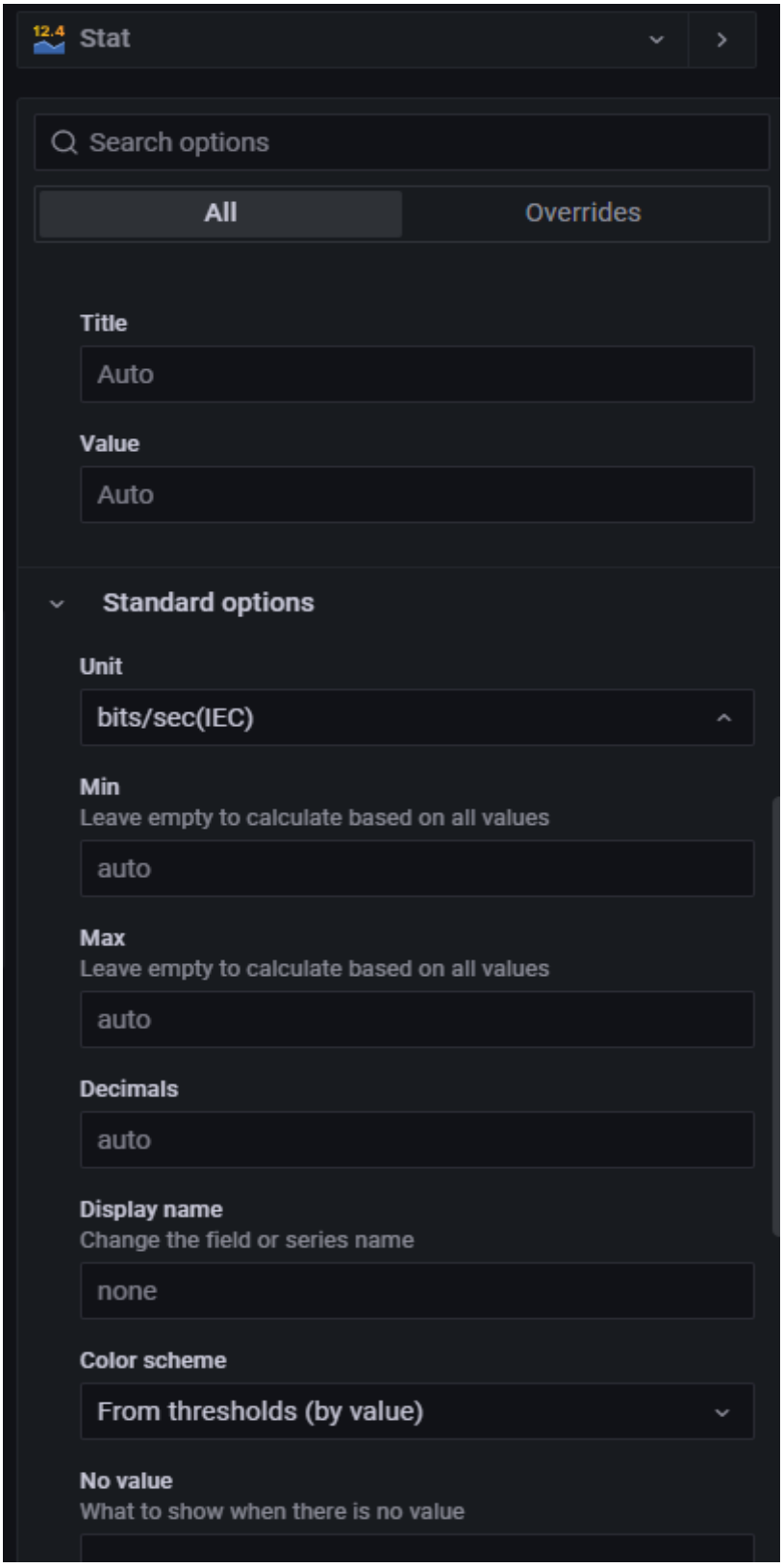


Ilustración 146: En Unit, dentro de Data & rate, seleccionamos bits/sec(IEC).

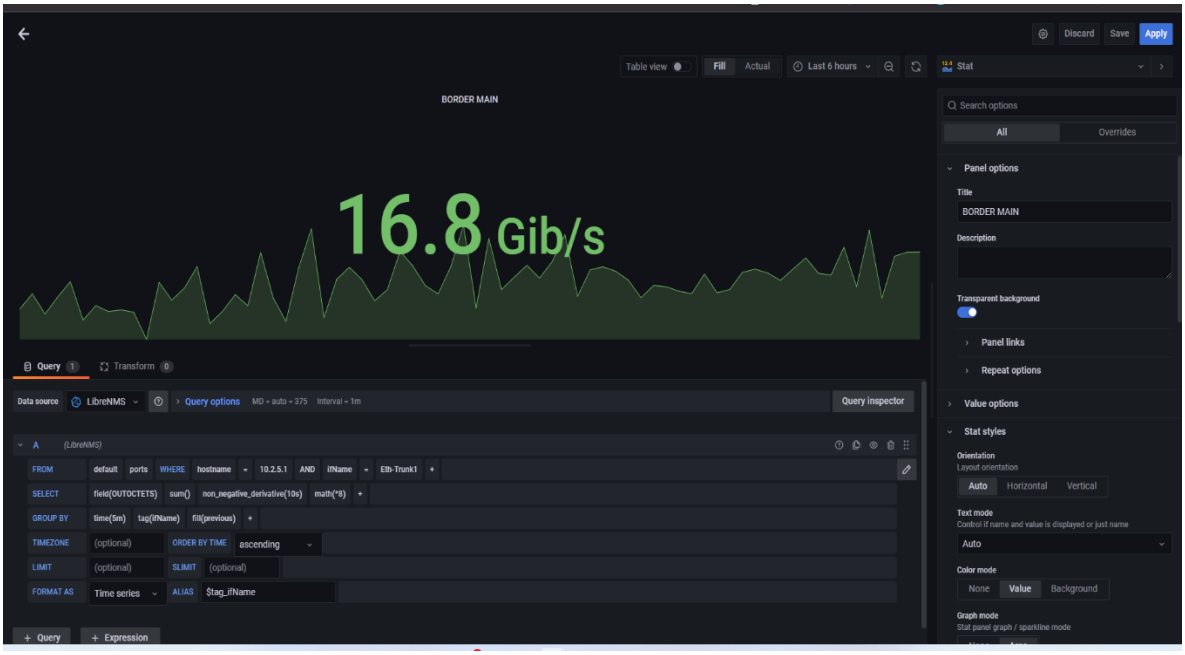


Ilustración 147: Obtendremos la siguiente gráfica.

Ilustración 148: En la pantalla principal del dashboard obtendremos el grafico, guardaremos el dashboard dando clic en el icono del disquete.

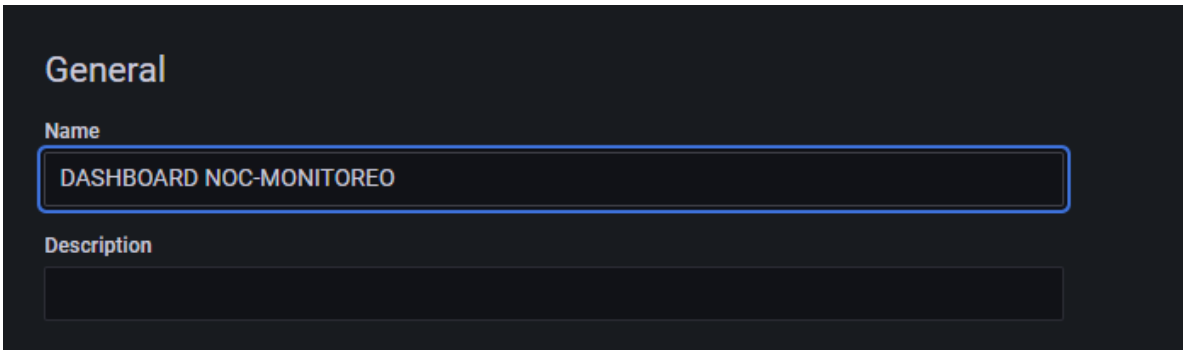


Ilustración 149: Daremos un nombre al dashboard.

3.6. Fase VI: Optimizar:

Dentro de esta fase se realizó cambios en el sistema de monitoreo para mejorar el rendimiento del mismo, según los requerimientos suscitados en la etapa de operacionalización de la plataforma de monitoreo.

En consecuencia a la implementación de las herramientas de monitoreo LibreNMS y el software de visualización de datos Grafana, pudimos observar que la plataforma de monitoreo funcionaba correctamente, sin embargo, para realizar el monitoreos de servicios específicos, estos dos software requieren la instalación de un nuevo software para monitorizar servicios específicos. Como alternativa escogimos Prometheus. A continuación, se indica los pasos y los comandos utilizados. También se puede verificar en las *ilustraciones 116 hasta la 124*, evidencia del proceso de instalación y configuración.

3.6.1. Instalación de Pometheus:

Actualizamos los repositorios:

root@prometheus01:~# apt update -y

Descargamos el paquete de instalación,lo descomprimimos e ingresaremos al directorio:

```
root@prometheus01:~# wget https://github.com/prometheus/prometheus/releases/download/v2.37.0-rc.0/prometheus-2.37.0-rc.0.linux-amd64.tar.gz
root@prometheus01:~# tar xvfz prometheus-2.37.0-rc.0.linux-amd64.tar.gz
root@prometheus01:~# cd prometheus-2.37.0-rc.0.linux-amd64/
root@prometheus01:~# mkdir -p /etc/prometheus
root@prometheus01:~# mkdir -p /var/lib/prometheus
```

Ilustración 150: Comandos de configuración y librerías.

Moveremos los binarios de prometheus y promtool:

root@prometheus01:~# mv /root/prometheus /root/promtool /usr/local/bin/

Crearemos el fichero yml de Prometheus y añadiremos lo siguiente:

```
  GNU nano 6.2                                    /etc/prometheus/prometheus.yml
global:
  scrape_interval:     15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
alerting:
  alertmanagers:
  - static_configs:
    - targets:
rule_files:
scrape_configs:
  - job_name: 'prometheus'

    static_configs:
    - targets: ['localhost:9090']
```

Ilustración 151: Archivo de configuración Prometheus .

Movemos el archivo al directorio /etc/prometheus:

root@prometheus01:~# mv prometheus.yml /etc/prometheus/prometheus.yml

Añadimos a los grupos y damos permisos a los directorios:

```
root@prometheus01:~# groupadd --system prometheus
root@prometheus01:~# useradd -s /sbin/nologin --system -g prometheus prometheus
root@prometheus01:~# chown -R prometheus:prometheus /etc/prometheus/  /var/lib/prometheus/
root@prometheus01:~# chmod -R 775 /etc/prometheus/ /var/lib/prometheus/
```

Ilustración 152: Comandos necesarios, para la creación de grupos y permisos.

Crearemos el archivo que iniciara el servicio de prometheus:

```
  GNU nano 6.2                                        /etc/systemd/system/prometheus.service
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Restart=always
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file=/etc/prometheus/prometheus.yml \
    --storage.tsdb.path=/var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries \
    --web.listen-address=0.0.0.0:9090

[Install]
WantedBy=multi-user.target
```

Ilustración 153: Archivo de configuración del servicio de Prometheus.

Iniciamos el servicio:

root@prometheus01:~# systemctl start prometheus

Configuramos para que inicie con el OS:

root@prometheus01:~# systemctl enable prometheus

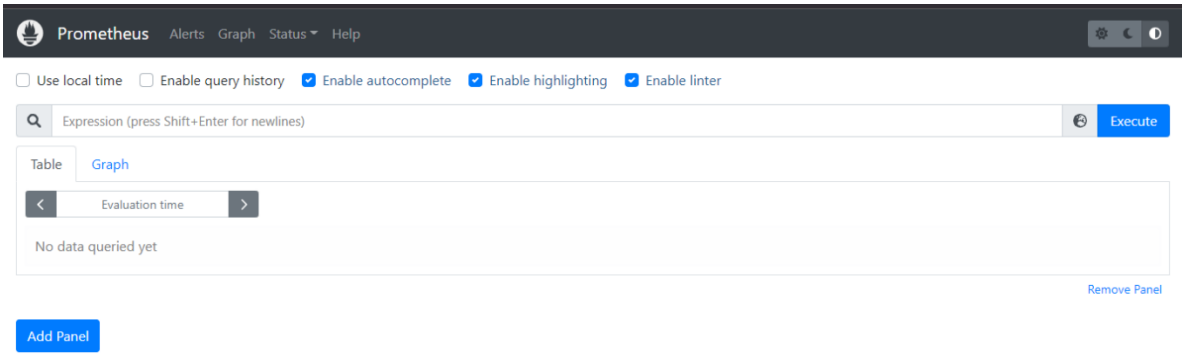
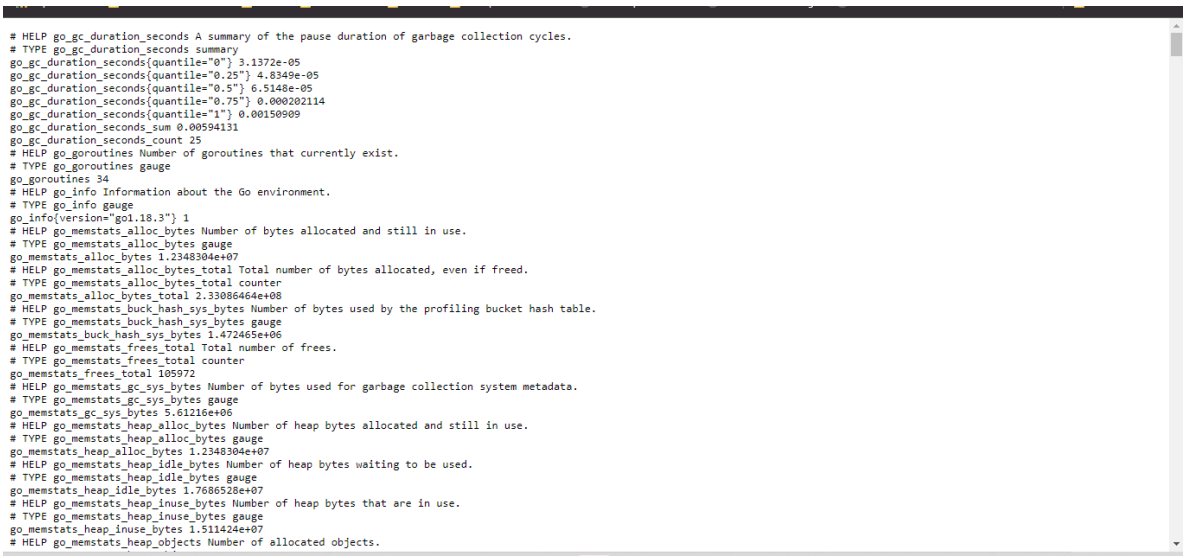


Ilustración 154: La instalación fue correcta en el navegador poniendo nuestra ip con el puerto 9090 deberá mostrar la siguiente ventana.



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.1372e-05
go_gc_duration_seconds{quantile="0.25"} 4.8349e-05
go_gc_duration_seconds{quantile="0.5"} 6.5148e-05
go_gc_duration_seconds{quantile="0.75"} 0.000202114
go_gc_duration_seconds{quantile="1"} 0.00150909
go_gc_duration_seconds_sum 0.00594131
go_gc_duration_seconds_count 25
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 34
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.18.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.2348304e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 2.33086464e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.472465e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 105972
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 5.61216e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.2348304e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 1.7686528e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 1.511424e+07
# HELP go_memstats_heap_objects Number of allocated objects.
```

Ilustración 155: En la misma ip con puerto y poniento metrics(ip:9090/metrics) nos mostrará lo siguiente.

3.6.2Configuración de Prometheus:

Descargamos el exportes en nuestro servidor DNS:

```
root@srv-dns-01:~# curl -s https://api.github.com/repos/prometheus-community/bind_exporter/releases/latest | grep browser_download_url | grep linux-amd64 |  cut -d '"' -f 4 | wget -qi -
```

Ilustración 156: comando para la descarga en el servidor DNS

Descomprimiremos el paquete descargado:

root@srv-dns-01:~# tar xvf bind_exporter*.tar.gz

Movemos los archivos descomprimidos al directorio /usr/local/bin

root@srv-dns-01:~# mv bind_exporter-*/bind_exporter /usr/local/bin

Comprobaremos la 128ersión instalada con el siguiente comando:

root@srv-dns-01:~# bind_exporter --version

Editaremos el archivo options de bind y añadiremos la opciòn de statistics-channels:

```
statistics-channels {
  inet 127.0.0.1 port 8053 allow { 127.0.0.1; };
```

Ilustración 157: Parámetros a añadir.

Creamos un nuevo grupo para prometheus:

root@srv-dns-01:~# groupadd --system prometheus

Añadiremos un usuario para prometheus y lo añadiremos al grupo anteriormente creado:

```
root@srv-dns-01:~# useradd -s /sbin/nologin --system -g prometheus prometheus
```

Ilustración 158: Comando para añadir al grupo al usuario Prometheus.

Crearemos el servicio para el exporter:

```
root@srv-dns-01:~# tee /etc/systemd/system/bind_exporter.service<<EOF
> [Unit]
> Description=Prometheus
> Documentation=https://github.com/digitalocean/bind_exporter
> Wants=network-online.target
> After=network-online.target
>
> [Service]
> Type=simple
multi-user.target
EOF> User=prometheus
> Group=prometheus
> ExecReload=/bin/kill -HUP \$MAINPID
> ExecStart=/usr/local/bin/bind_exporter \
>   --bind.pid-file=/var/run/named/named.pid \
>   --bind.timeout=20s \
>   --web.listen-address=0.0.0.0:9153 \
>   --web.telemetry-path=/metrics \
>   --bind.stats-url=http://localhost:8053/ \
>   --bind.stats-groups=server,view,tasks
>
> SyslogIdentifier=prometheus
> Restart=always
>
> [Install]
> WantedBy=multi-user.target
> EOF
```

Ilustración 159: Archivo de configuración del exporter de Prometheus.

Recargaremos los servicios:

root@srv-dns-01:~# systemctl daemon-reload

Reiniciaremos el servicio del exporter:

root@srv-dns-01:~# systemctl restart bind_exporter.service

Indicaremos que el servicio inicie junto al sistema:

root@srv-dns-01:~# systemctl enable bind_exporter.service

En el servidor de prometheus en al archivo prometheus.yml, añadiremos el job para el DNS:

```
  - job_name: dns-master
    static_configs:
      - targets: ['138.122.108.26:9153']
        labels:
          alias: dns-master
```

Ilustración 160: añadimos el job correspondiente para la extracción de datos del servidor DNS.

Reiniciaremos el servicio de prometheus:

root@prometheus01:~# systemctl restart prometheus

```
root@srv-dns-01:~# bind_exporter --version
bind_exporter, version 0.5.0 (branch: HEAD, revision: 792f7f381f95fbe5fb319b18cedb9ae6f6ad013b)
  build user:       root@711e62b7164b
  build date:       20211123-09:00:54
  go version:       go1.17.3
  platform:         linux/amd64
```

Ilustración 161: Nos mostrará la versión del exporter instalado.

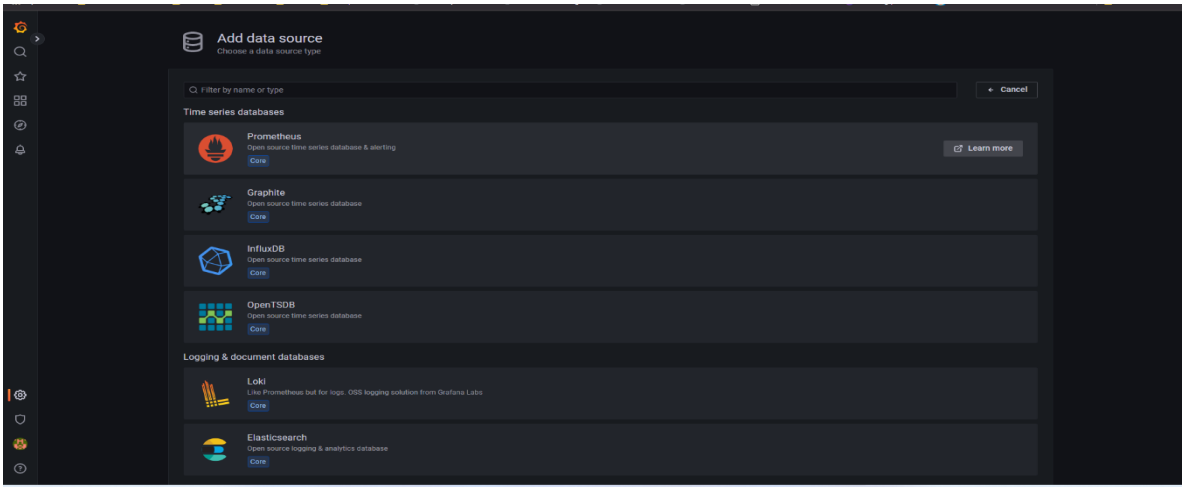


Ilustración 162: Dentro de Grafana añadiremos el datasource tipo Prometheus.

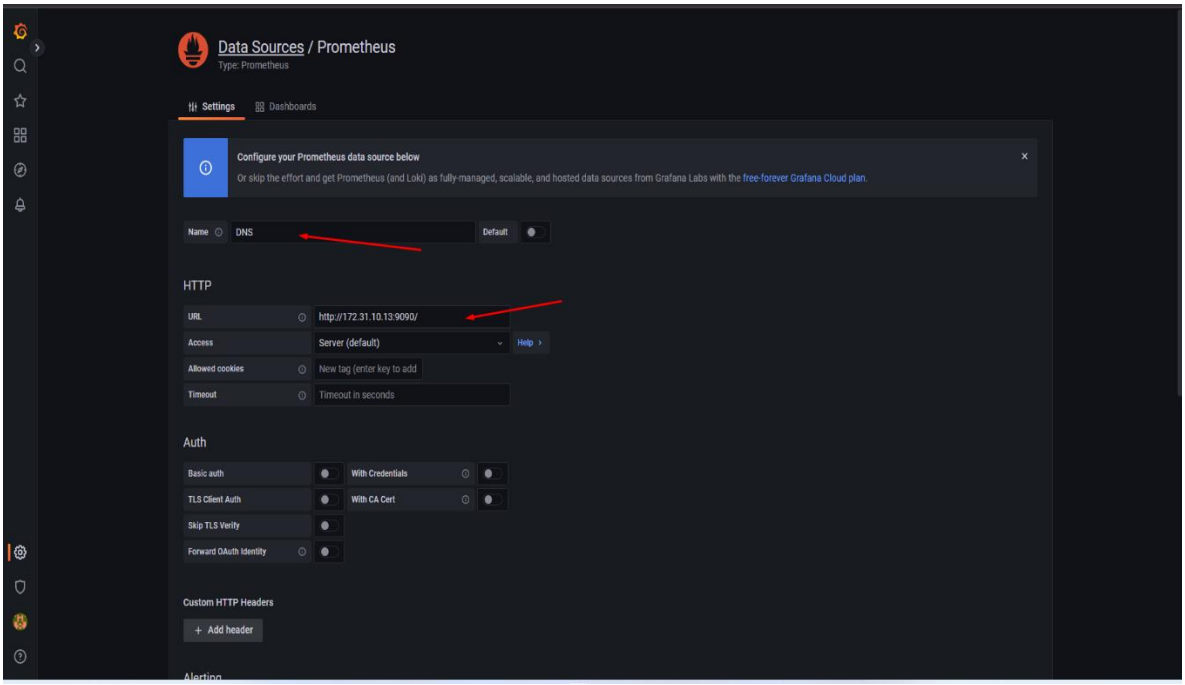


Ilustración 163: Le daremos un nombre y en la URL especificaremos ip:9090.

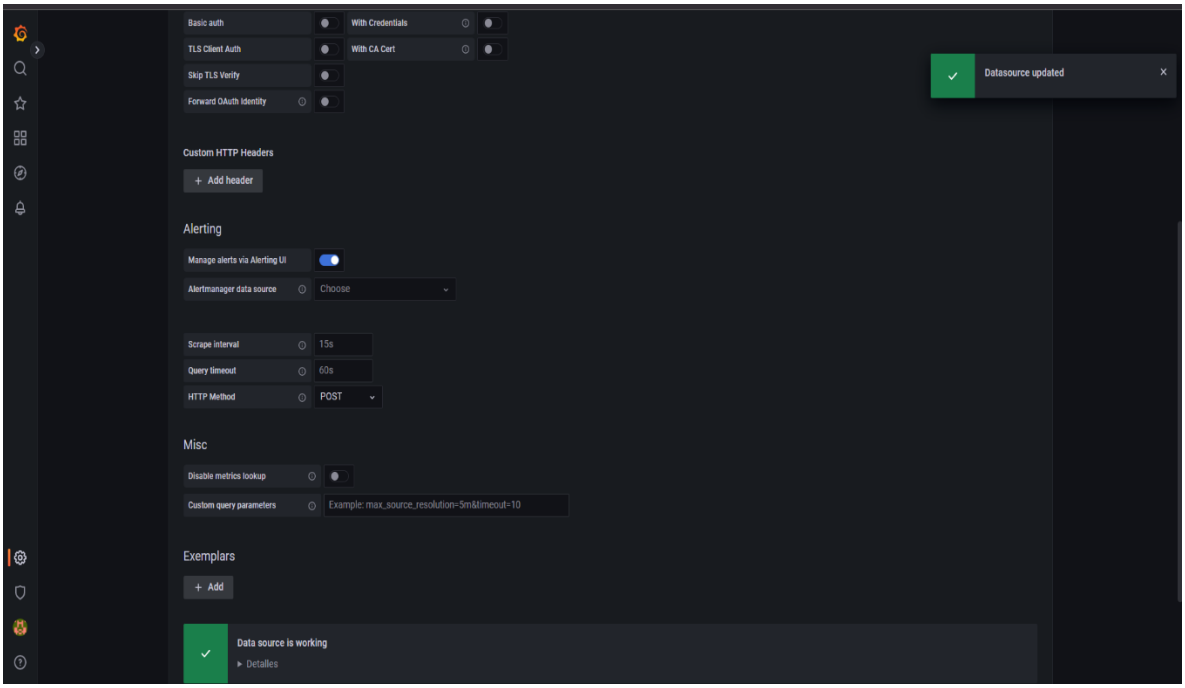


Ilustración 164: Damos clic en save & test y si todo es correcto nos mostrará pruebas exitosas.

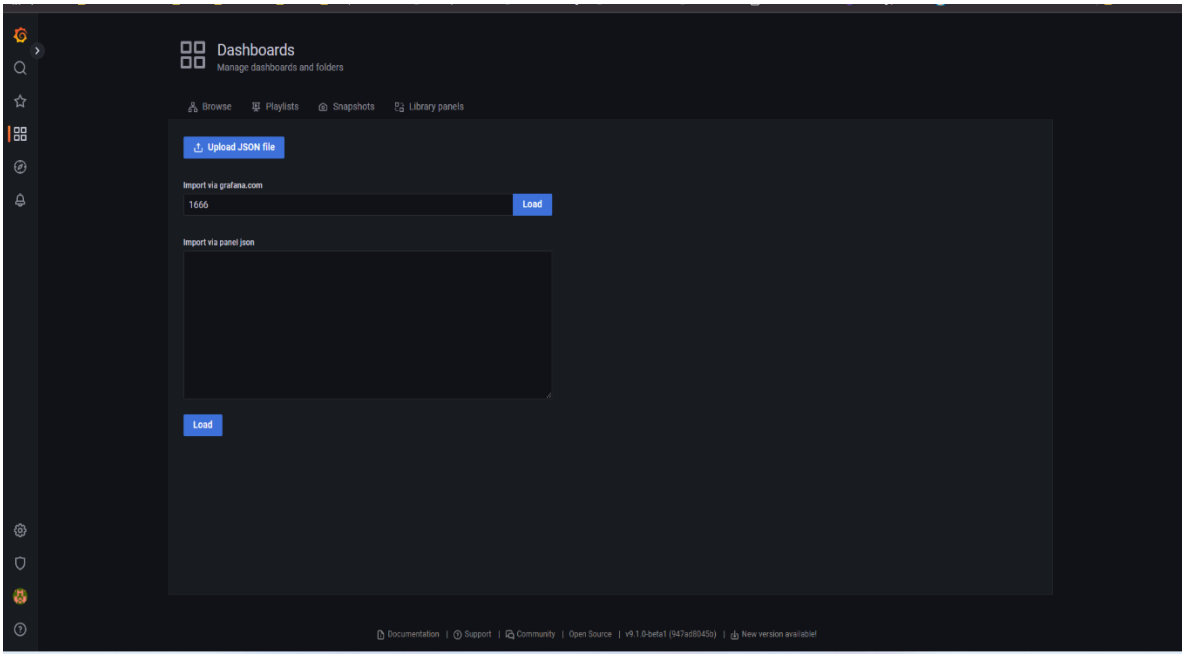


Ilustración 165: Para probar el datasource usaremos un dashboard ha creado para lo cual lo exportaremos usando el ID del dashboard es 1666.

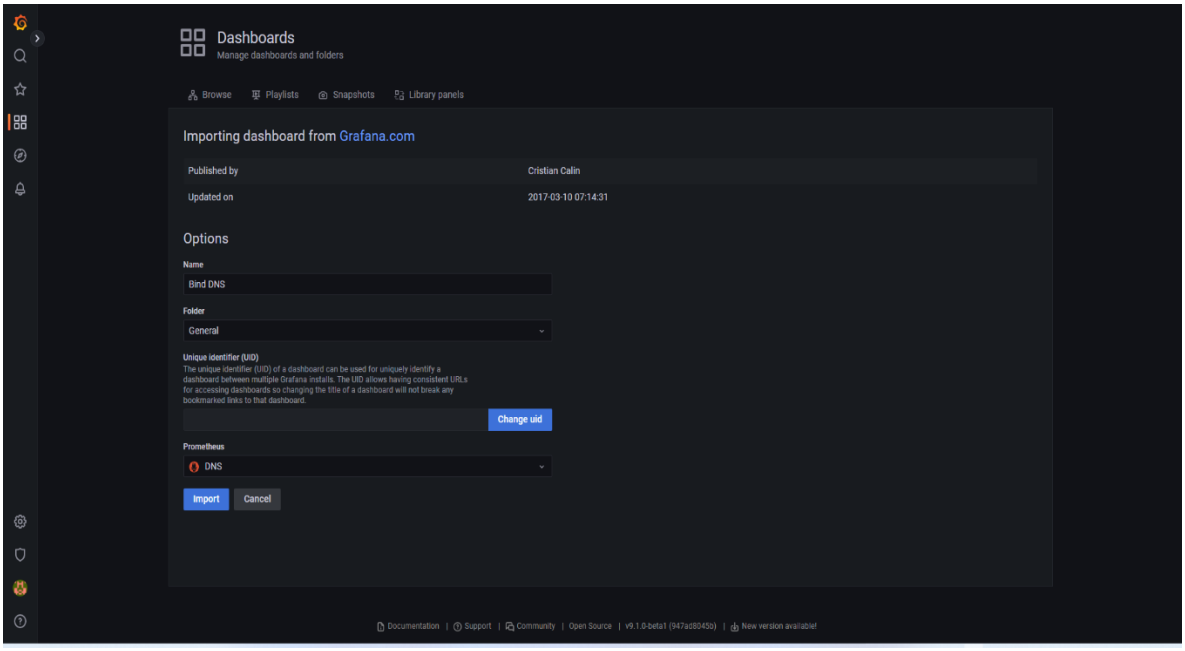


Ilustración 166: Le indicamos el datasource que usará.

Ilustración 167: Nos mostrará los datos que se encuentran en Prometheus.

3.6.3. Producto final:

Haciendo uso de las herramientas y los pasos descritos en el presente trabajo investigativo, se logró obtener los resultados esperados por parte del NOC, los cuales fueron los siguientes:

- Monitoreo de los dispositivos del core.
- Alertas vía correo electrónico.
- Alertas vía mensajería instantánea Telegram.
- Histórico de consumo de interfaces.
- Estado de los dispositivos del core.
- Integración con el dashboard Grafana para la generación de gráficos.

En las ilustraciones 225 hasta la 230 se pueden observar los resultados de implementación de la plataforma de monitoreo.

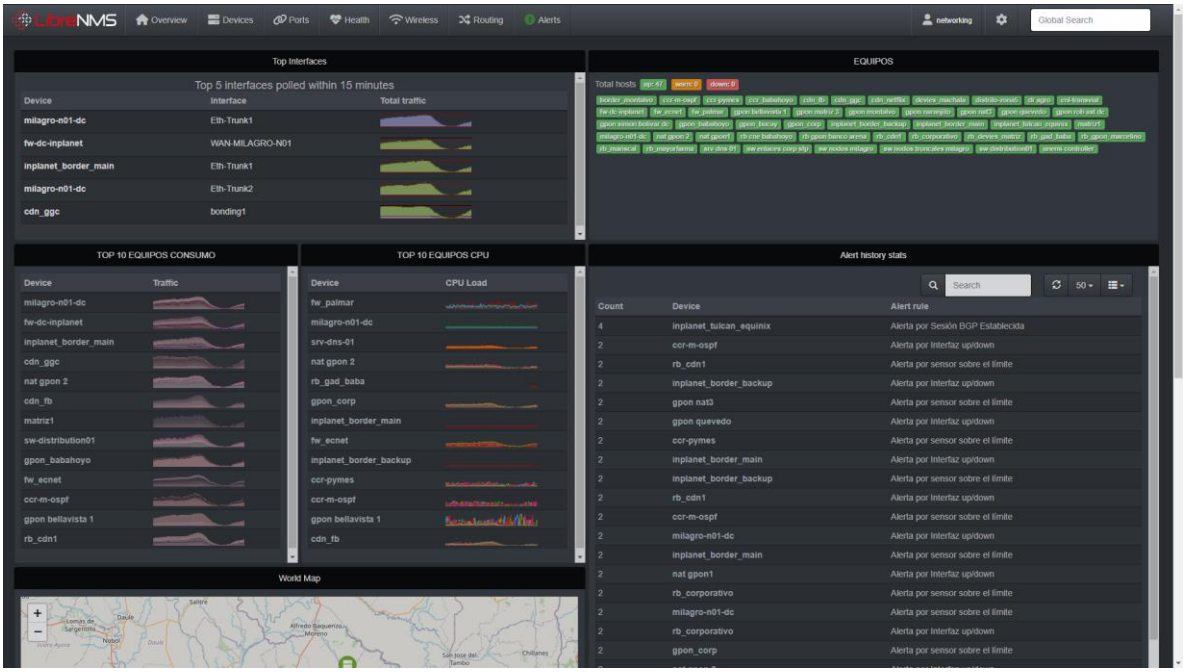


Ilustración 168: Dashboard LibreNMS donde muestra top de interfaces con mayor consumo de ancho de banda, equipos que consumen mayor ancho de banda en la red, estados de equipos, y alertas.

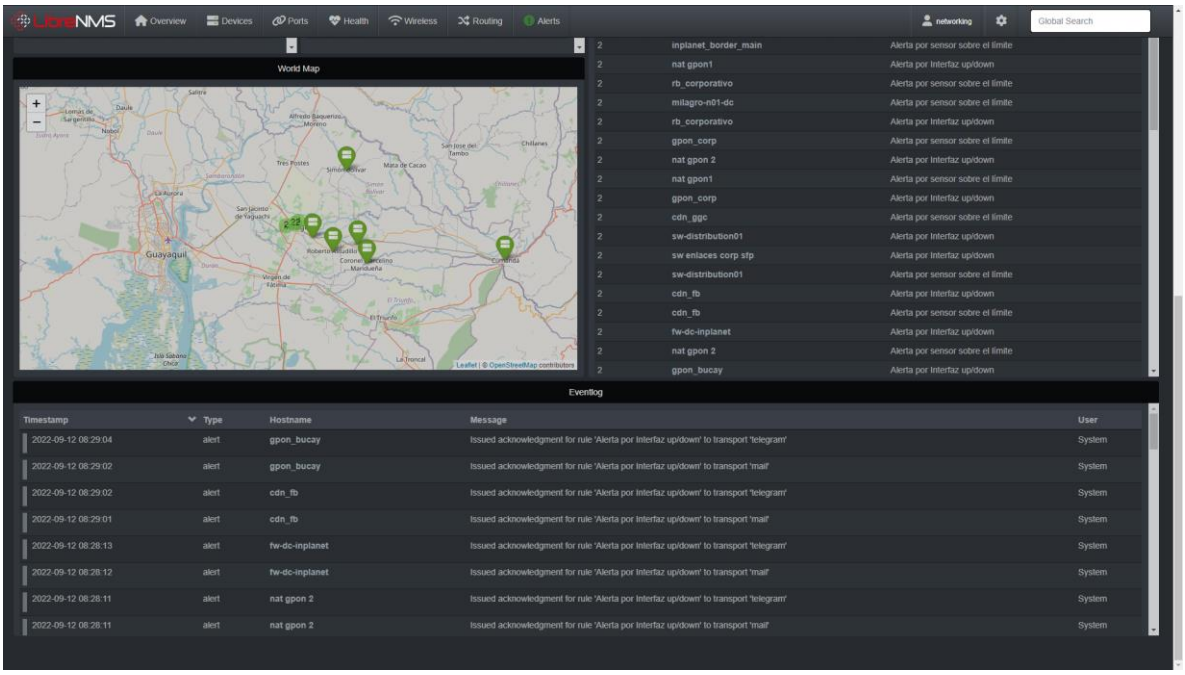


Ilustración 169: Dashboard LibreNMS donde se muestra la ubicaciòn geografica de los dispositivos, parte de las alertas y el eventlog.

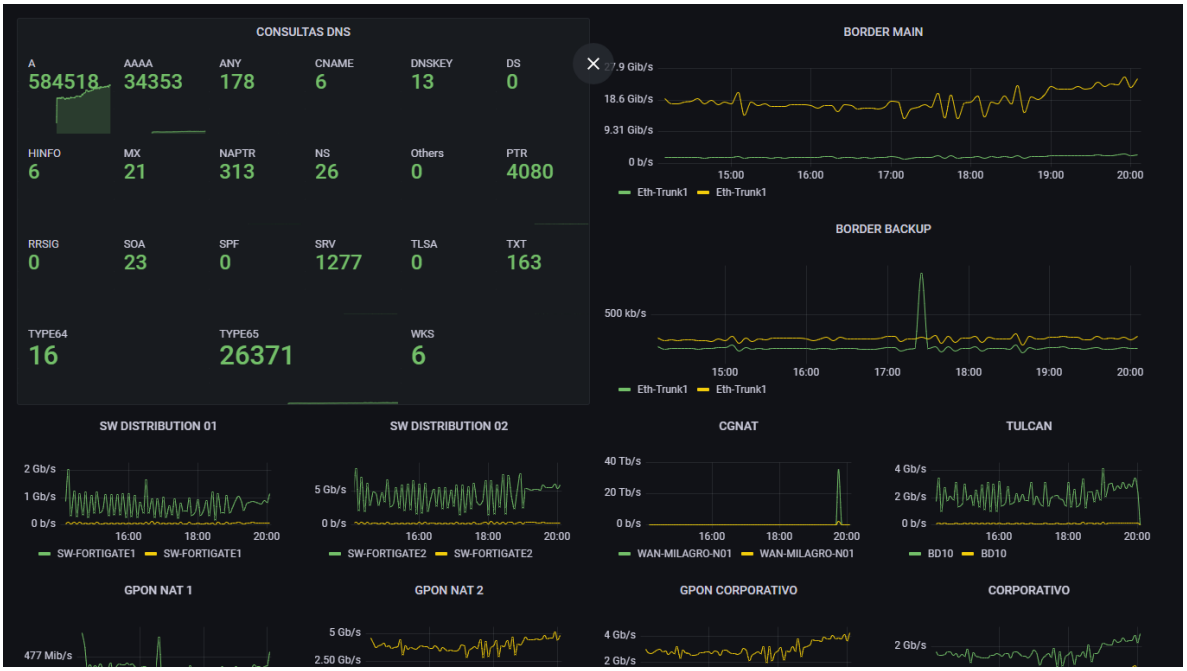


Ilustración 170: Dashboard Grafana donde muestra el consumo de las interfaces principales de los dispositivos del core así como la estadística de las consultas DNS.

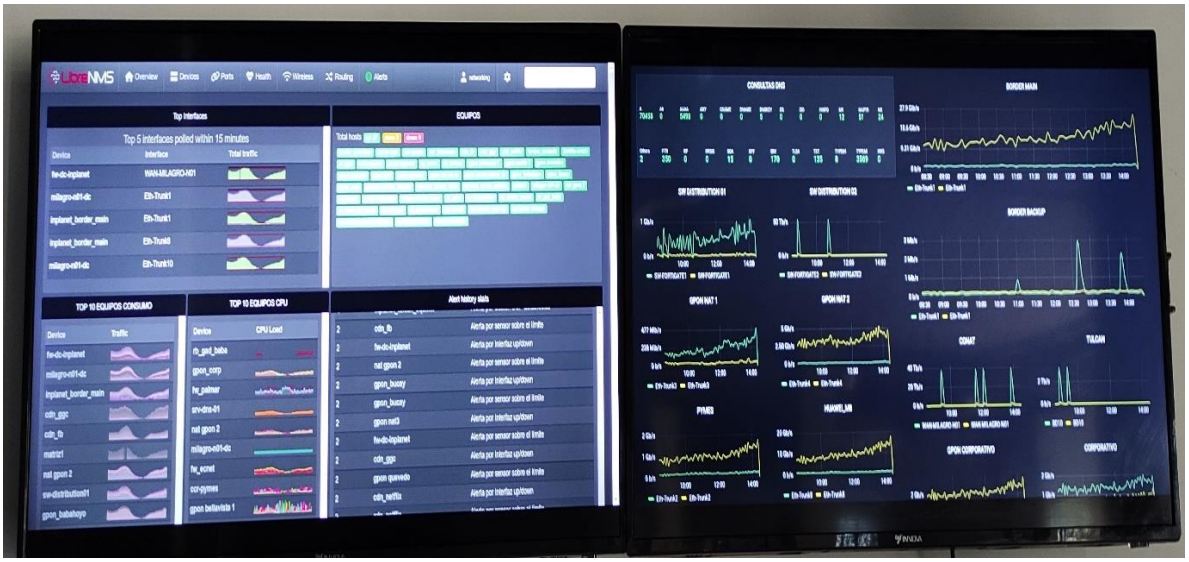


Ilustración 171: Dashboards proyectados en los monitores del NOC.

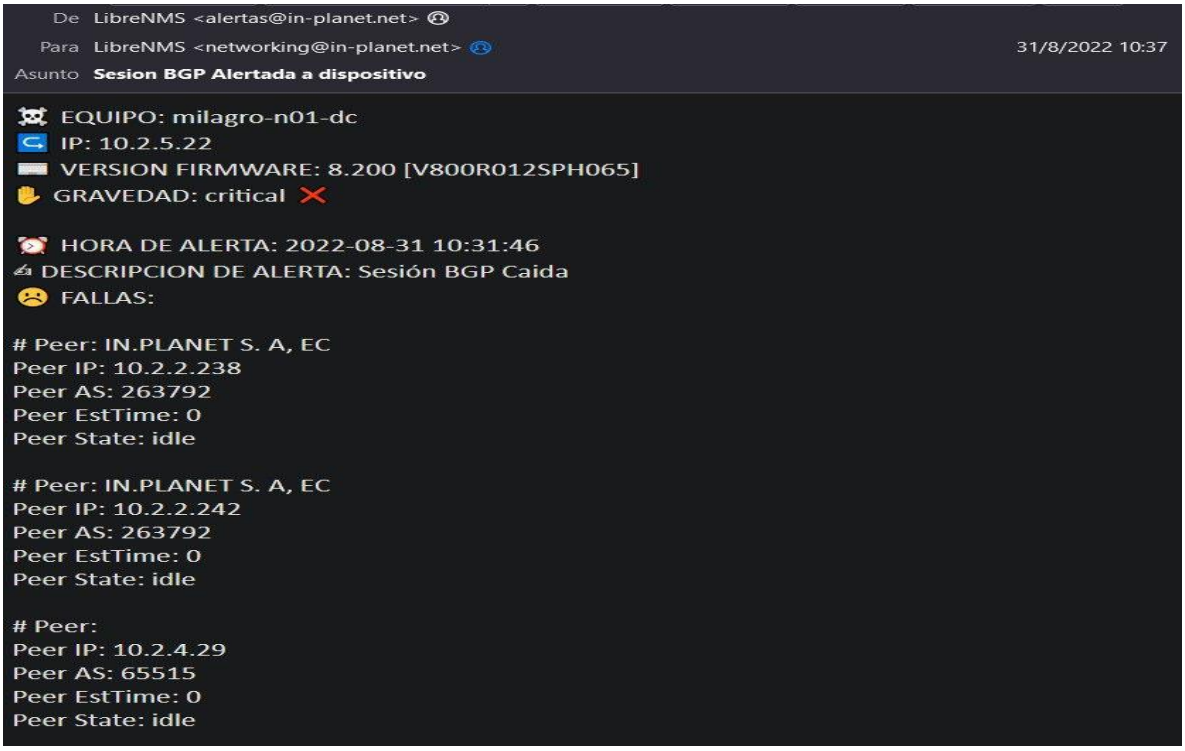


Ilustración 172: Alerta enviada desde el servidor LibreNMS hacia la cuenta de correo configurada a recibir las alertas.

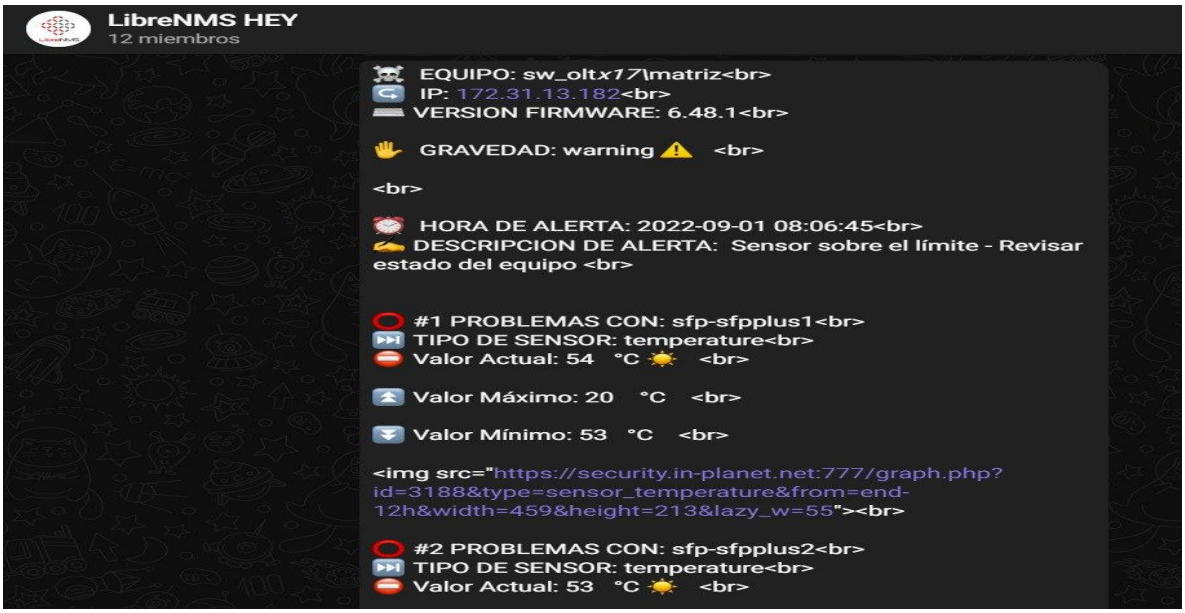


Ilustración 173: Alerta enviada desde el servidor LibreNMS hacia el grupo de Telegram configurado para recibir las alertas.

4. CONCLUSIÓN:

La identificación de las herramientas open source fueron de vital utilidad, ya que, representa un aporte significativo para las empresas que no cuentan con los recursos económicos necesario para adquirir otras herramientas que impliquen la cancelación por brindar los mismos servicios que un software gratuito.

Teniendo en cuenta la variedad de herramientas open source, resultó conveniente evaluar cada uno de los softwares, según los aspectos más relevantes que ayuden a solucionar de forma eficiente las necesidades por las cuales se encuentra transitando el departamento del NOC de la ISP. Im.Planet. S. A., y que se integren correctamente con el dashboard de visualización de datos, Grafana. La evidencia de ello podemos encontrar en la *tabla 8*; Comparación de herramientas de monitoreo de red Open Source, *tabla 9*; Escala equivalente, y la *tabla 10*; Evaluación de distintos rangos.

La instalación y la respectiva configuración de las herramientas de monitoreo seleccionadas, se realizaron sin mayor dificultad, además es importante mencionar que. la curva de aprendizaje de LibreNMS junto con Grafana para el uso correcto de la plataforma de monitoreo, no representa mayor dificultad.

Pudimos corroborar que, la implementación de la plataforma de monitoreo de red haciendo uso de herramienta OpenSource, para suplir las necesidades del Network Operations Center de la ISP, In. Planet. S. A., funciona con gran eficiencia. Cabe mencionar que, en primera instancia, la integración entre las herramientas de monitoreo seleccionadas para la creación de la plataforma de monitoreo de red, LibreNMS y Grafana, contribuyeron eficazmente lo solicitado por el departamento del NOC, Sin embargo, requirió una optimización la cual, fue cubierta mediante la instalación de una herramienta adicional (Prometheus) OpenSource, lo cual no significó ninguna intervención económica por parte de la ISP. Como lo podrán corroborar, en el producto final en las *ilustraciones 125*; Dashboard LibreNMS donde muestra top de interfaces con mayor consumo de ancho de banda, equipos

que consumen mayor ancho de banda en la red, estados de equipos, y alertas, *Ilustración 126;* Dashboard LibreNMS donde se muestra la ubicaciòn geografica de los dispositivos, parte de las alertas y el eventlog, *ilustración 127;* Dashboard Grafana donde muestra el consumo de las interfaces principales de los dispositivos del core así como la estadística de las consultas DNS, *ilustración 128*; Dashboards proyectados en los monitores del NOC, *ilustración 129*; Alerta enviada desde el servidor LibreNMS hacia la cuenta de correo configurada a recibir las alertas, y *la ilustración 130*; Alerta enviada desde el servidor LibreNMS hacia el grupo de Telegram configurado para recibir las alertas, se puede visualizar la plataforma de monitoreo en acción la cual cumple con todos los requerimientos solicitados.

BIBLIOGRAFÍA:

León León, J. F. (2019). implementación de un servidor zabbix en el consorcio educativo continental, para el analisis y monitoreo de equipos en la red (Doctoral dissertation).

Bustincio, Q., & Watson, J. (2018). Implementación de un sistema de monitoreo y control de red, para un canal de televisión, basado en herramientas Open Source y Software Libre, Lima-2017.

Camarena Escriba, A. E. (2019). Diseño del prototipo de un sistema de monitoreo de recursos y servicios de redes de comunicación usando CENTREON en la empresa DATCOM SAC.

Enciso Cochachi, H. G. (2021). Diseño e implementación de un sistema de monitoreo del centro de datos para la red del INICTEL-UNI utilizando software libre (Doctoral dissertation, Universidad Nacional Tecnológica de Lima Sur).

Cantos San Emeterio, J. (2021). Diseño e implementación de una plataforma de gestión mediante LibreNMS: monitorización y control de la red privada del laboratorio de Telemática (GIT–UNICAN).

González Sierra, D. X. (2016). Diseño e implementación de un servidor SNMP (a través de nagios), DHCP, DNS y Netflow sobre la arquitectura de Raspberry PI2 para el monitoreo y administración del ISP urbanet en el cantón Durán.

Guamialamá Narváez, D. F. (2008). Implementación de un sistema para monitoreo de servicios en servidores críticos bajo sistemas operativos open source (linux) y controlado mediante mensajería externa GSM (Bachelor's thesis, QUITO/EPN/2008).

Oré Alvaro, C. (2019). Implementación de un sistema de monitoreo para asegurar la continuidad de los servicios en un data center utilizando protocolo SNMP.

Quispe Ccuno, J. R. (2019). Implementación de un prototipo de monitoreo de dispositivos de comunicación y usuarios finales utilizando el protocolo SNMP basada en software libre para una empresa e-Commerce.

Rios Epalza, L. (2020). Implantación de un sistema de monitoreo para la infraestructura de red de datos de la UFPS sede Cúcuta y Campos Elíseos.

Soto Alvarez, S. (2021). Monitoreo de redes con software open source para el control de estado de servicios en una empresa de telecomunicaciones (Doctoral dissertation).

Vargas Maquilon, J. A., & Maruri Uriña, E. S. (2021). Implementación de un servidor de autoconfiguración y monitoreo para un ISP ubicado en el Cantón Balzar (Doctoral

dissertation, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones).

Vega Picon, G. E. (2018). Implementación de un sistema de monitoreo para el análisis de la disponibilidad, capacidad, calidad y latencia de enlaces corporativos de última milla.

Velasco Briones, C. A., & Cagua Ordoñez, G. S. (2017). Implementación de un sistema de monitoreo de redes utilizando herramientas open source y proveer servicios de directorio a través de active directory en la facultad de Filosofía y Ciencias de la educación de la universidad de Guayaquil.

Zambrano Burgos, M. A., Santisteban Avalos, E. I., Landio Rojas, R. F., & Flores Panaifo, J. M. (2019). Sistema de monitoreo de infraestructura para la gestión de recursos de TI en la empresa COGA.

Red Hat. (2019). ¿Qué es la gestión de las redes? https://www.redhat.com/es/topics/management/what-is-network-management#resumen

Consulting Informático. (2021). ¿Qué es un NMS – Network Management System? https://www.cic.es/que-es-un-nms-network-management-system/

Zabbix. (2018). What is Zabbix. Recuperado el marzo de 2018, de http://www.zabbix.com/features

Rueda Ortega, P. (2020). Gestión de red en entornos Cloud: demostrador sobre OpenStack y su integración con la plataforma Nagios.

Salas, G., Stteeven, J., & Roa Piñeros, C. A. (2020). Diseño de una herramienta de monitoreo y control de servidores utilizando como eje principal cacti. Aplicado a una pyme mediana. Universidad Cooperativa de Colombia, Facultad de Ingeniería, Bogotá DC. Recuperado de https://repository. ucc. edu. co/bitstream/20.500, 12494(16571), 3.

Agudelo, C. E. G., Florian-Gaviria, B., & Aristizábal, E. M. (2022). Estudio de plataformas de monitoreo para seleccionar la pila tecnológica base de un sistema de analíticas especializado para pruebas de software. Ingeniería y Competitividad, 24(1).

Influxdata. (2021). Influxdata. Telegraf. https://www.influxdata.com/time-seriesplatform/telegraf/

ANEXOS:

Entrevista realizada al NOC:

Preguntas realizadas en la entrevista al departamento del NOC:

ENTREVISTA

Nombre del entrevistado: Jerry Palomeque

Nombre del entrevistador: Leonardo Piña

PREGUNTAS

¿Qué software utilizan actualmente?

MRTG y dudo se uran actualmente para el monitoreo de la red, dicho software no cumple con las necesidades requeridas por parte del departamento ya que son herramientas muy limitadas, si bien cumplen su función no cubre todas las necesidades que el departamento requiere

MRTG solo permite ver el ancho de banda actualmente esta usando la interfaz de red, pero no dispone de mas sensores como CPU RAM Temperatura fuentes Poder entre otros, mientras que el software dudo si bien es cierto usa SNMP no hace información no cumple con las necesidades ya que, solo alerta si existe caída mas no si hay una falla en el sensor

¿Qué necesitan?

El departamento del NOC necesita tener un monitoreo completo que alerte si existe alguna falla en el sensor, así como tener un registro de log de los eventos sucitados en los sensores por ejemplo si se tiene un enlace entre el routes A y B y dicho enlace se cae pero los routes siguen activos al tener enlaces redundantes ese daño debería ser alertado ya sea vía mail o vía mensajería como telegram mail slack ICQ, discord entre otros, el sistema implementado debe tener dos instancias para el monitoreo del NOC, 1 instancia sera con el software NHS y la 2 sera con el dashboard, el NHS lo usara personal de networking y gerencia los usara personal de monitoreo

¿Qué podrían necesitar a futuro?

integración vía bot de mensajería para el monitoreo proactivo es decir enviar comandos que se ejecutaran en el servidor o revisiones del monitoreo.