



Monitoreo Avanzado de OpenMediaVault con Grafana

Daniel Infantes Lara

Administración de Sistemas Informáticos en Red

IES Medina Azahara

20-06-2024



Esta obra está sujeta a una licencia de Reconocimiento - No Comercial - Sin Obra Derivada 3.0 [España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

ÍNDICE

1. Introducción
 - Descripción del problema
 - Justificación
 - Análisis del estado del arte
 - Marco legal o normativo
 2. Planificación del proyecto
 - Temporalización
 - Objetivos del proyecto
 - Análisis de requisitos
 - Análisis de las soluciones
 - Recursos necesarios para la solución elegida
 - Presupuesto
 3. Implementación
 - Descripción de la implementación técnica
 - Documentación generada por la implementación
 4. Pruebas y Validación
 - Estrategia de pruebas
 - Resultados de las pruebas realizadas
 - Procedimientos de resolución de incidencias
 5. Conclusiones
 - Resumen de los logros alcanzados
 - Reflexión sobre los desafíos enfrentados y soluciones aplicadas
 - Implicaciones prácticas y posibles mejoras futuras
 6. Referencias
 7. Anexos
-

Introducción

Descripción del problema

El almacenamiento en red es esencial para la gestión de datos en empresas y usuarios avanzados. OpenMediaVault (OMV) es una plataforma popular para configurar servidores NAS debido a su facilidad de uso y capacidad de integración con diversas herramientas. Sin embargo, OMV presenta limitaciones en el monitoreo detallado de archivos y el uso del almacenamiento, lo cual puede dificultar la administración eficiente del sistema y la planificación de la capacidad de almacenamiento. En la UCO se presentaba un problema en el que la transferencia de archivos era muy lenta y se enviaban los archivos mediante correo, este tenía un límite de espacio para los archivos, teniendo que así enviarlo por partes.

Justificación

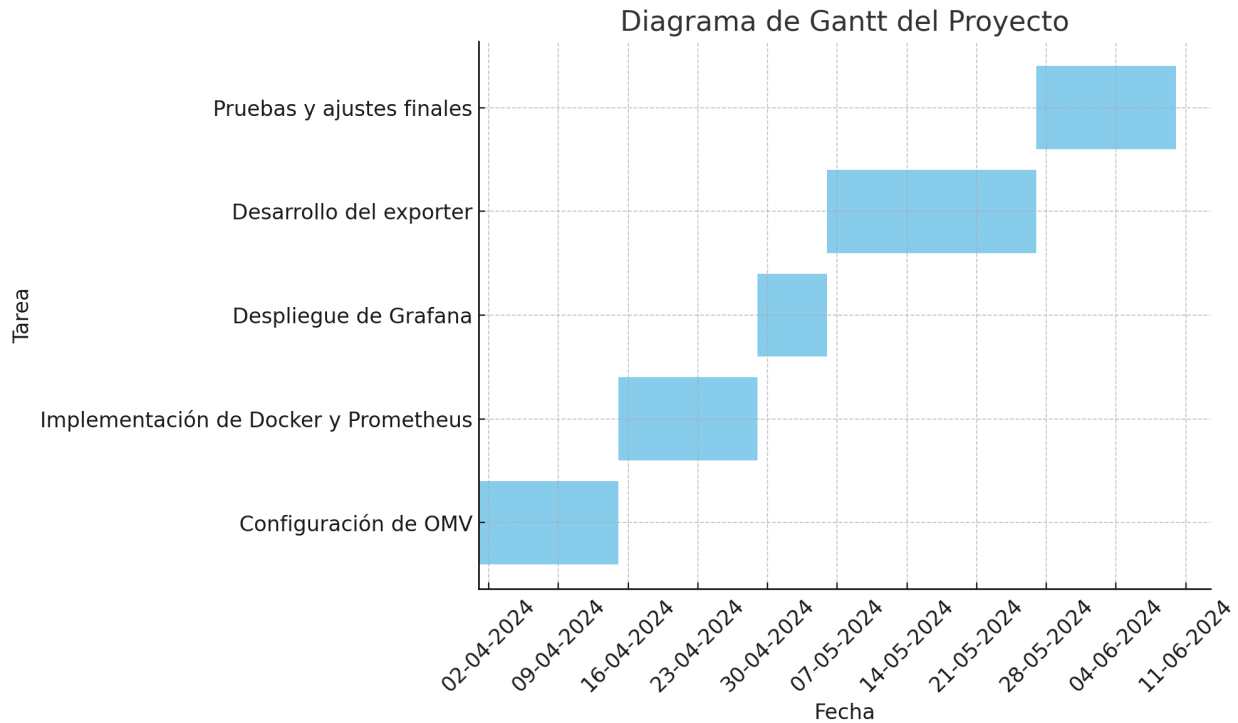
Para superar estas limitaciones, se propone integrar Grafana, Prometheus y Node Exporter en OMV. Esta integración proporcionará una visualización y monitoreo avanzados del sistema, mejorando significativamente la capacidad de diagnóstico y mantenimiento preventivo. Con esta solución, los administradores de sistemas podrán tener una visión clara y detallada del rendimiento y uso del servidor NAS, permitiendo una mejor gestión y planificación de los recursos. La solución para eso sería un servicio en la nube para poder transferir archivos de manera rápida, segura y eficaz, usando OMV para arreglar eso.

Marco legal o normativo

En el marco del proyecto de monitoreo avanzado de OpenMediaVault con Grafana, Prometheus y Node Exporter, es crucial asegurar el cumplimiento con la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales. Esta ley establece las directrices para la protección de datos personales en España y garantiza los derechos digitales de los ciudadanos.

Planificación del proyecto

Temporalización



El proyecto se planifica para completarse en un plazo de dos meses, distribuyendo el tiempo de la siguiente manera:

1. Configuración de OMV: 2 semanas (1 de abril al 14 de abril)
2. Implementación de Docker y Prometheus: 2 semanas (15 de abril al 28 de abril)
3. Despliegue de Grafana: 1 semana (29 de abril al 5 de mayo)
4. Desarrollo del exporter: 3 semanas (6 de mayo al 26 de mayo)
5. Pruebas y ajustes finales: 2 semanas (27 de mayo al 9 de junio)

Objetivos del proyecto

- Proveer una solución de monitoreo en tiempo real para OpenMediaVault.
- Visualizar el uso del espacio de almacenamiento y la actividad de archivos y carpetas.
- Facilitar la administración y diagnóstico del servidor NAS.

- Implementar un exportador personalizado para reportar información específica sobre archivos y carpetas.

Análisis de requisitos

Requisitos funcionales:

- Monitoreo en tiempo real del sistema.
- 1 Gb de RAM
- 4 GB Almacenamiento (para el sistema)
- Espacio de almacenamiento en la nube lo necesario dependiendo del uso
- Visualización del uso de espacio de almacenamiento.

Requisitos no funcionales:

- Alta disponibilidad del sistema de monitoreo.
- Escalabilidad para soportar un mayor número de métricas.
- Facilidad de uso e interpretación de los datos.

Requisitos de seguridad:

- Protección de los datos de monitoreo.
- Acceso seguro a los dashboards de Grafana.
- Cumplimiento con normativas de gestión de datos.

Análisis de las soluciones

Se evaluaron varias herramientas de monitoreo, como Zabbix y Grafana. La combinación de Grafana y Prometheus fue elegida debido a su flexibilidad, facilidad de integración y capacidad avanzada de visualización. Grafana permite crear dashboards personalizados y Prometheus ofrece un sistema robusto de recolección de métricas , alertas y permiten vincularse entre ellos muy fácilmente.

Recursos necesarios para la solución elegida

Hardware:

- Servidor NAS con CPU de 2 núcleos, 4GB de RAM y 1TB de almacenamiento mínimo.

Software:

- OpenMediaVault
- Prometheus
- Node Exporter
- Grafana

Presupuesto

- Hardware: 500 € - 1000 €

- Software: Gratuito (OpenMediaVault, Docker, Prometheus, Grafana)
 - Tiempo del técnico: Aproximadamente 80 horas de trabajo (a estimar en función del coste horario del técnico).
-

Implementación

Descripción de la implementación técnica

1. Instalación y configuración de OpenMediaVault (OMV)

- Descargar e instalar OMV en un servidor dedicado.
Para instalar OpenMediaVault , si lo queremos hacer en un servidor dedicado o en una máquina virtual , con instalar la ISO y ejecutarla , si es en una raspberry tendrías que instalarlo con un pendrive de arranque
- Configurar las opciones básicas y avanzadas del servidor NAS.
La configuración inicial es elegir el idioma y luego configurar el nombre (este nombre es solo para identificarla en red)

[[! Configurar la red

Por favor, introduzca el nombre de la máquina.

El nombre de máquina es una sola palabra que identifica el sistema en la red. Consulte al administrador de red si no sabe qué nombre debería tener. Si está configurando una red doméstica puede inventarse este nombre.

Nombre de la máquina:

openmediavault

<Retroceder>

<Continuar>

El siguiente paso será configurar el dominio (.com, .es) para que pueda ser buscado en la Internet.

También nos pedirá la contraseña de superusuario (root) y nos pedirá que la confirmamos.

Nos pedirán que réplica de Debian vamos a usar.

Al ser de España usaremos la de deb.debian.org

[[! Configurar el gestor de paquetes

Por favor, seleccione una réplica de Debian. Debería escoger una réplica en su país o región si no sabe qué réplica tiene mejor conexión de Internet hasta usted.

Normalmente, deb.debian.org es una buena elección.

Réplica de Debian:

Después de esto se empezará la instalación , y dentro de unos minutos lo tendremos listo , nos pedirá reiniciar , así que reiniciamos la máquina para que la configuración se aplique correctamente

Para poder acceder al monitor de OMV tendremos que poner la IP que se nos indique en nuestro navegador

```
openmediavault 7.0-32 (Sandworm) OMV tty1
Copyright (C) 2009-2024 by Volker Theile. All rights reserved.

To manage the system visit the openmediavault workbench:
enp0s3: 192.168.10.189

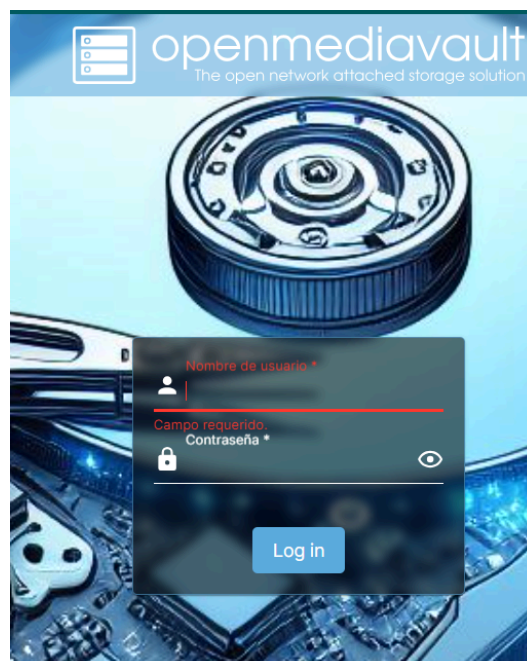
By default the workbench administrator account has the
username 'admin' and password 'openmediavault'.
It is recommended that you change the password for this account
within the workbench or using the 'omv-firstaid' CLI command.

For more information regarding this appliance, please visit the
web site: https://www.openmediavault.org

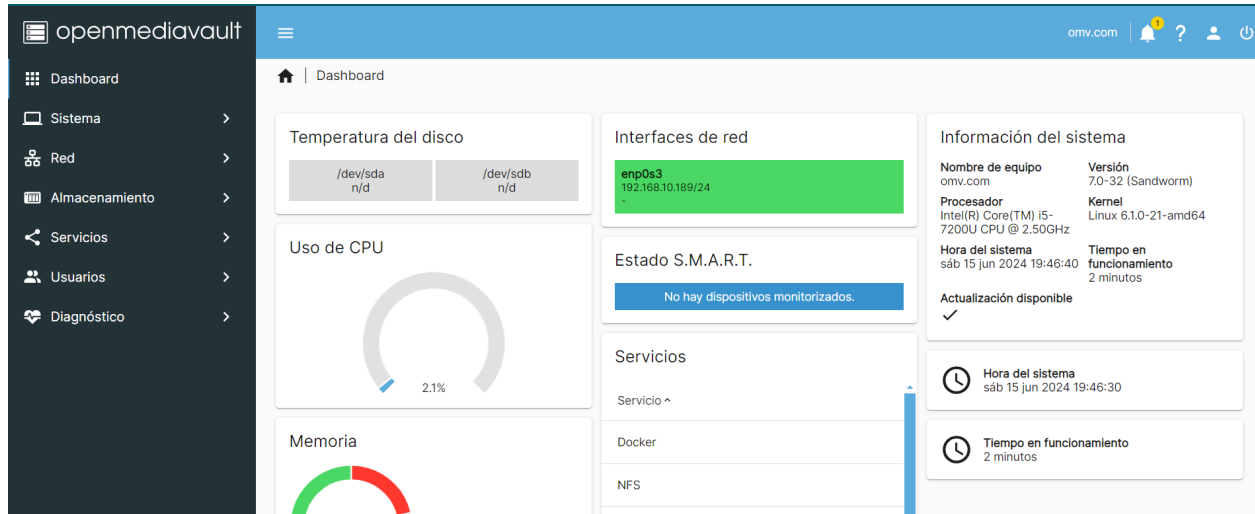
OMV login: _
```

(La IP esta puesta como fija , pero si no esta configurado , aparecerá en cada reinicio , una diferente)

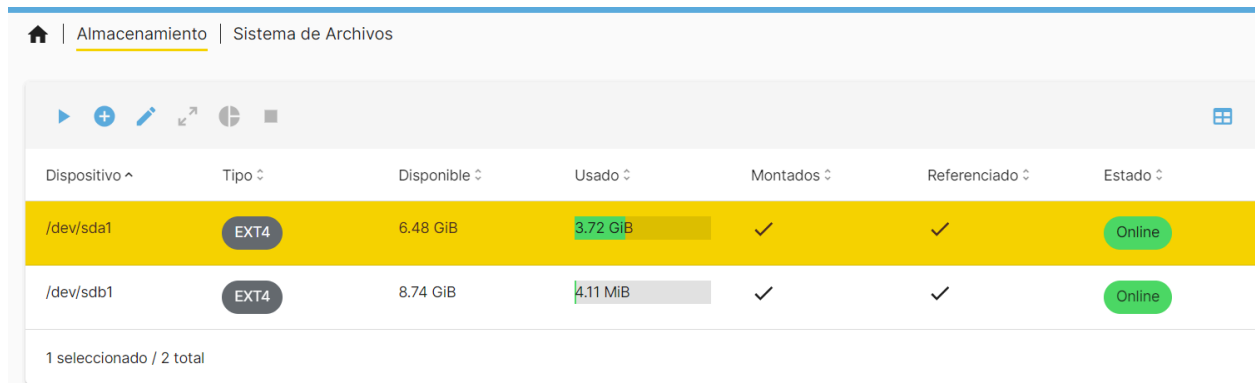
En el login estan las contraseñas por defecto que eso se puede cambiar o quitar en cualquier momento desde la página , de usuario es admin y de contraseña es openmediavault



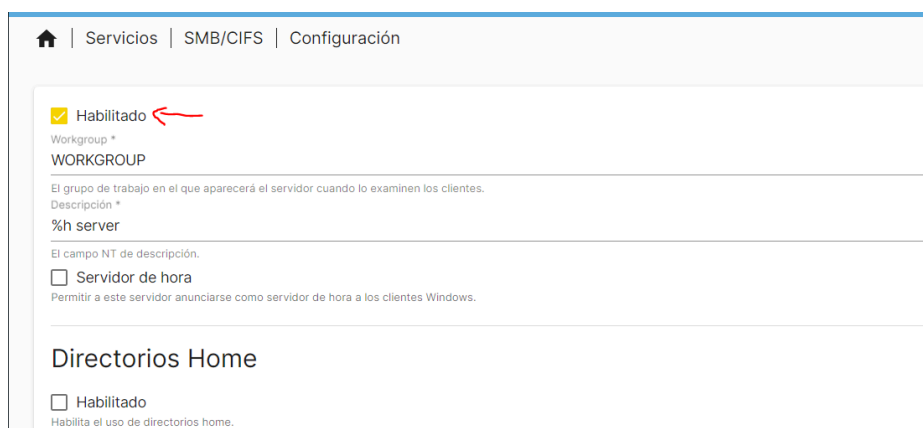
Y entraremos en el panel de control



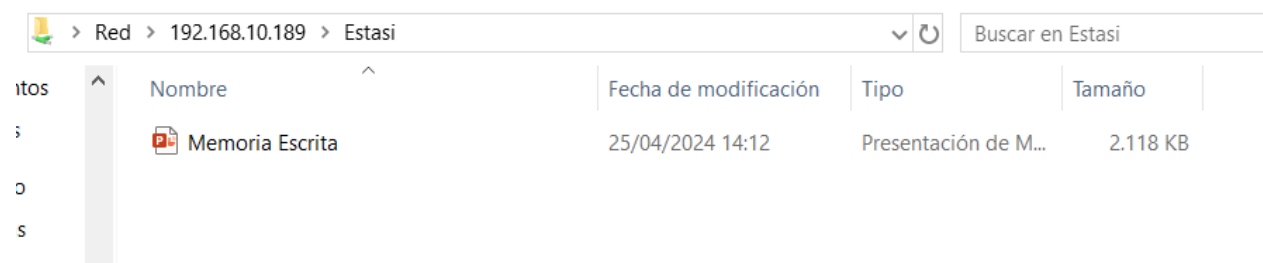
Después nos vamos a storage > sistema de archivos y creamos un almacenamiento con otro disco duro insertado en la máquina (desde virtualbox es solo crear un disco duro virtual y ya)



Una vez esté el volumen de almacenamiento montado nos dirigimos a servicios > SMB/CIFS desde ahí entramos en ajustes y lo habilitamos para que puedan acceder todos los usuarios o los especificados



Y después desde un ordenador con un sistema operativo de Windows , presionamos WIN+R y ponemos la IP de OMV , y podremos tener una carpeta compartida en la que podemos almacenar los archivos que queramos hasta quedarnos sin espacio



(para acceder tienes que poner \\(tu_ip)/)

2. Instalación y configuración de Prometheus y Node Exporter

Prometheus es un sistema de monitorización y de alerta de código abierto que se desarrolló en el año 2012 . Prometheus recopila datos y métricas de servicios y hosts mediante solicitudes HTTP y los almacena en una base de datos de series de tiempo

He decidido que la misma máquina virtual de OMV ponga todos los servicios para mayor facilidad , estos no se sobreponen ni hay que cambiar puertos.

Para poder instalar Prometheus he accedido mediante SSH a la máquina virtual para poder instalarlo :

```
wget
https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0
.freebsd-amd64.tar.gz
```

Esto lo que hace es descargar el tar.gz de prometheus para poder instalarlo

```
tar xvfz prometheus-2.52.0.freebsd-amd64.tar.gz
```

Extraemos el contenido del tar.gz

```
cd prometheus-2.52.0.freebsd-amd64
```

Y nos dirigimos a la carpeta que acabamos de crear

```
sudo useradd --no-create-home --shell /bin/false prometheus
```

Creamos un usuario para usar prometheus

```
sudo mkdir /var/lib/prometheus
```

```
sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus
```

```
sudo chown prometheus:prometheus /usr/local/bin/prometheus /usr/local/bin/promtool
```

Tenemos que darle unos permisos para:

- 1.- Crear la carpeta /var/lib/prometheus
- 2.- Cambiar el propietario del directorio /etc/prometheus y /var/lib/prometheus a prometheus
- 3.- Cambiar el propietario de los archivos binarios prometheus y promtool a prometheus

Creamos un usuario para poder usar prometheus

```
sudo systemctl daemon-reload
```

```
sudo systemctl start prometheus
```

```
sudo systemctl enable prometheus
```

Y recargamos los servicios e iniciamos Prometheus

- Instalar y configurar Node Exporter para recolectar métricas del servidor.

Node Exporter es una herramienta que obtiene estadísticas de aplicaciones y las convierte en métricas que Prometheus puede utilizar .

```
wget  
https://github.com/prometheus/node\_exporter/releases/download/v1.8.1/node\_exporter-1.8.1.linux-amd64.tar.gz
```

Descargamos el node exporter

```
tar xvfz node_exporter-1.8.1.linux-amd64.tar.gz
```

Extraemos el archivo

```
cd node_exporter-1.8.1.linux-amd64
```

Movemos el archivo a bin

```
sudo mv node_exporter /usr/local/bin/
```

Creamos un Usuario para Node Exporter

```
sudo useradd --no-create-home --shell /bin/false node_exporter
```

Este paso es opcional podemos hacer que el servicio se ejecute de manera automática al iniciar la máquina

```
sudo nano /etc/systemd/system/node_exporter.service
```

Creamos ese archivo con este contenido

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

[Service]

User=node_exporter

Group=node_exporter

Type=simple

ExecStart=/usr/local/bin/node_exporter

Restart=always

RestartSec=3

[Install]

WantedBy=multi-user.target

```
sudo systemctl daemon-reload  
sudo systemctl start node_exporter
```

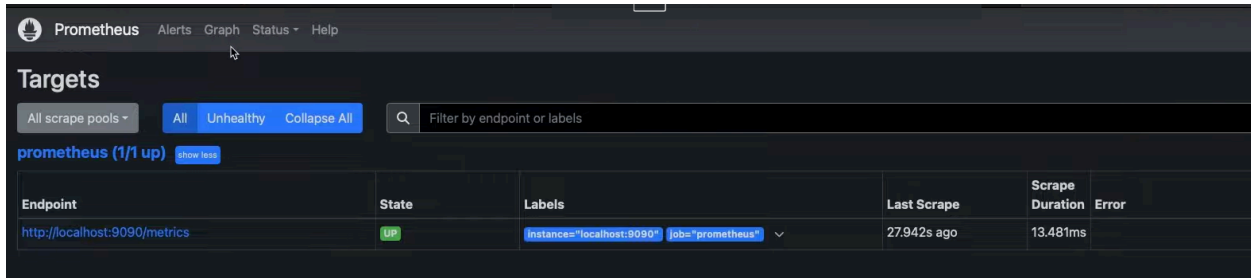
```
sudo systemctl enable node_exporter
```

Y reiniciamos el servicio e iniciamos el node_exporter

Comprobamos si el servicio funciona correctamente

Una vez instalados estos 2 complementos tenemos que irnos a `/etc/prometheus/prometheus.yaml`, en este archivo vamos a encontrar la configuración de prometheus y encontraremos un “trabajo” ya creado llamado prometheus.

Nos dirigimos a la página principal de prometheus > Status > Targets y tendremos que ver algo similar



The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below this, the 'Targets' section is active. It shows 'All scrape pools' and a search bar. A table lists the targets. One target is shown: 'http://localhost:9090/metrics' with a state of 'UP'. The table has columns for Endpoint, State, Labels, Last Scrape, Scrape Duration, and Error.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	27.942s ago	13.481ms	

Esto significa que nuestro Prometheus y Node Exporter están funcionando correctamente .

3. Despliegue de Grafana

Grafana es un software de visualización de gráficos y análisis de código abierto. Grafana permite la visualización y el formato de datos métricos . Le permite consultar, visualizar, alertar y explorar sus métricas a partir del almacenamiento de base de datos de series temporales

```
sudo apt-get install -y apt-transport-https software-properties-common wget
```

Instalamos las dependencias

```
sudo mkdir -p /etc/apt/keyrings/
```

Creamos ese directorio

```
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
```

Actualizamos

```
sudo apt-get update
```

Instalamos Grafana

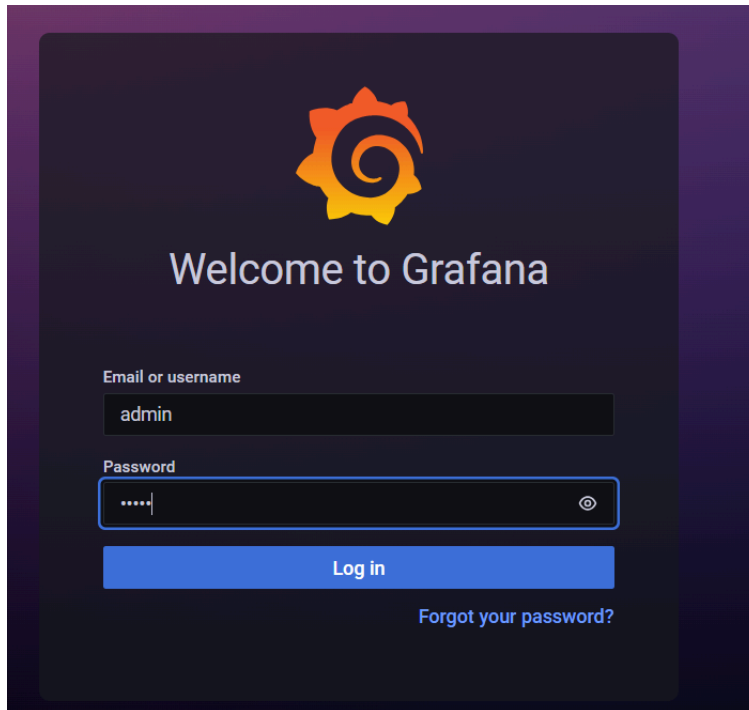
```
sudo apt-get install grafana
```

E iniciamos el servidor

```
sudo systemctl start grafana-server
```

```
sudo systemctl enable grafana-server
```

Para entrar a la página de Grafana , tienes que poner en tu navegador (tu_ip:3000) y aparecerá esto



(El usuario y la contraseña es admin)

Y ya tendríamos Grafana funcionando

- Configurar Grafana para conectar con Prometheus y crear dashboards personalizados.

Para conectar Grafana con Prometheus , una vez instalado el node exporter , prometheus y grafana , hay que enlazarlos mediante un archivo que se encuentra en /etc/prometheus/prometheus.yaml , que sería este archivo

```

# Configuración global
global:
| scrape_interval: 15s # Intervalo de raspado cada 15 segundos
  evaluation_interval: 15s # Evaluar las reglas cada 15 segundos

# Configuración de Alertmanager
alerting:
  alertmanagers:
    - static_configs:
      - targets: []

# Reglas
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# Configuración
scrape_configs:
  - job_name: daniel
    honor_timestamps: true
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    static_configs:
      - targets:
        - localhost:9090

```

Lo importante de este archivo es el apartado de configuración en el que:

- job_name: 'daniel': Nombre del trabajo de scraping.
- honor_labels: true: Prometheus respetará las etiquetas ya presentes en las métricas del exportador.
- scrape_interval: 15s: Intervalo de scraping específico para este trabajo.
- metrics_path: /metrics: Ruta de las métricas (por defecto es /metrics).
- static_configs: Configuración estática que define los endpoints a los que Prometheus debe hacer scraping. En este caso, localhost:9090

Documentación generada por la implementación

- Ficheros de configuración editados para OMV, Prometheus y Grafana.

Con Grafana y OMV no he tocado ningún archivo de configuración, ya que este es todo desde el apartado de configuración gráfica, pero con Prometheus y node exporter sí he tenido que cambiar cosas manualmente


```
GNU nano 7.2 prometheus.yml
# Configuración global
global:
  scrape_interval: 15s # Intervalo de raspado cada 15 segundos
  evaluation_interval: 15s # Evaluar las reglas cada 15 segundos

# Configuración de Alertmanager
alerting:
  alertmanagers:
    - static_configs:
      - targets: []

# Reglas
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# Configuración de raspado
scrape_configs:
  - job_name: daniel
    honor_timestamps: true
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    static_configs:
      - targets:
        - localhost:9090
```

El archivo ya mencionado anteriormente “prometheus.yml” en él definimos la configuración para que el node exporter la recoja y la envíe a grafana mediante los targets

- Capturas de pantalla de la configuración y de los dashboards creados en Grafana.

Este dashboard es creado por un usuario pero he tenido que modificar unos valores para que me salgan los valores mostrados en esta captura



Pruebas y Validación

Estrategia de pruebas

- Pruebas unitarias: Asegurar que cada componente del sistema funciona de manera independiente.

```
root@OMV:/opt/prometheus# cd
root@OMV:~# sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-06-10 10:00:25 CEST; 2h 12min ago
     Docs: http://docs.grafana.org
   Main PID: 515 (grafana-server)
      Tasks: 9 (limit: 2517)
     Memory: 101.6M
        CPU: 14.322s
    CGroup: /system.slice/grafana-server.service
            └─515 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:default.paths.1

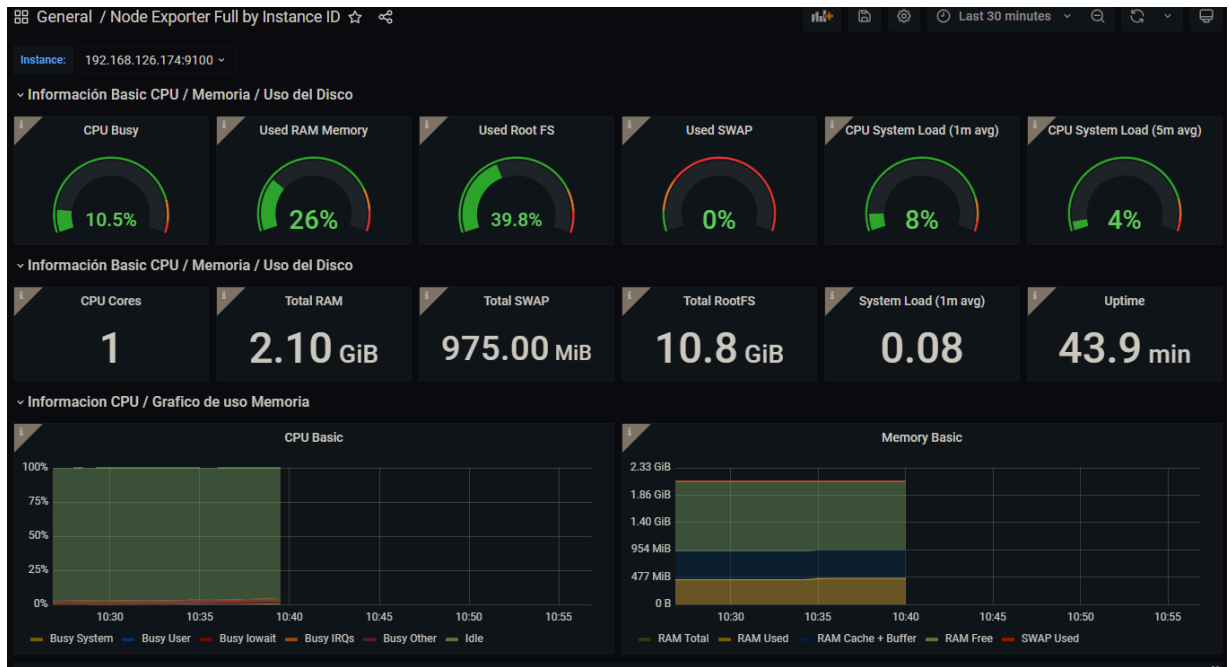
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="Path Provisioning" logger=settings path=/etc/grafana/provisioning
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="App mode production" logger=settings
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="Connecting to DB" logger=sqlstore dbtype=sqlite3
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="Starting DB migrations" logger=migrator
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="migrations completed" logger=migrator performed=0 skipped=346 duration=3.
jun 10 10:00:29 OMV grafana-server[515]: t=2024-06-10T10:00:29+0200 lvl=info msg="Starting plugin search" logger=plugins
jun 10 10:00:30 OMV grafana-server[515]: t=2024-06-10T10:00:30+0200 lvl=info msg="Registering plugin" logger=plugins id=input
jun 10 10:00:30 OMV grafana-server[515]: t=2024-06-10T10:00:30+0200 lvl=info msg="Live Push Gateway initialization" logger=live.push_http
jun 10 10:00:30 OMV grafana-server[515]: t=2024-06-10T10:00:30+0200 lvl=info msg="Writing PID file" logger=server path=/run/grafana/grafana-server.pid pid=5
jun 10 10:00:30 OMV grafana-server[515]: t=2024-06-10T10:00:30+0200 lvl=info msg="HTTP Server Listen" logger=http server_address=:13000 protocol=http sub
```

```
root@OMV:~# sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-06-10 10:00:25 CEST; 2h 29min ago
   Main PID: 517 (node_exporter)
      Tasks: 5 (limit: 2517)
     Memory: 29.1M
        CPU: 242ms
    CGroup: /system.slice/node_exporter.service
            └─517 /usr/local/bin/node_exporter

jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=thermal_zone
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=time
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=timex
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=udp_queues
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=uname
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=vmstat
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=xfs
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:115 level=info collector=zfs
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=node_exporter.go:199 level=info msg="Listening on" address=:9100
jun 10 10:00:27 OMV node_exporter[517]: ts=2024-06-10T08:00:27.393Z caller=tls_config.go:195 level=info msg="TLS is disabled." http2=false
```

Resultados de las pruebas realizadas

- Pruebas de integración: Prometheus recolecta correctamente las métricas de



Node Exporter y del exporter personalizado; Grafana muestra los datos adecuadamente.

Procedimientos de resolución de incidencias

- Identificación de incidencias: Utilización de logs de Prometheus y Grafana para identificar problemas.
- Resolución de problemas:

Incidencias que he tenido a lo largo del proyecto

1.- OMV no me daba IP , la solución es simple solo poner en la terminal de OMV "omv-firstaid" , esto crea una interfaz en la que seleccionas la opción 1 y se configura la IP

2.- El el documento "prometheus.yaml" los errores de indentación son comunes una solución es instalar "yamllint" con "sudo apt-get install yamllint" y su utilización es yamllint /ruta/archivo este comprueba su indentación y otros errores , es básicamente un compilador de muchos archivos (html, yaml , dockercompose, etc)

3.- Cuando vas a mostrar los datos en el panel de Grafana , algunas veces aparece diciendo no data , eso es un fallo de prometheus y node exporter , se debe a que en el archivo de prometheus , los targets y o jobs estan mal configurados (eso se debe a cambio de IP)

Conclusiones

Resumen de los logros alcanzados

- Integración exitosa de Grafana y Prometheus con OpenMediaVault.
- Creación de dashboards personalizados para visualizar el uso del almacenamiento y la actividad de archivos y carpetas.

Reflexión sobre los desafíos enfrentados y soluciones aplicadas

- Desafíos: Configuración inicial de los componentes, fallos en la indentación de archivos en Prometheus, exportación incorrecta en node exporter ,problemas del servicio de prometheus no ejecutándose correctamente, y pruebas de integración.
- Soluciones: Documentación detallada, pruebas continuas y ajustes incrementales en la configuración y explicación detallada de los errores y cómo solucionarlo .

Implicaciones prácticas y posibles mejoras futuras

- Una mejora posible es que se pueda acceder desde cualquier dispositivo que tenga conexión a internet
 - O que sea accesible sin conexión / de manera offline
-

Referencias

- Instalación de Grafana: [Install Grafana on Debian or Ubuntu | Grafana documentation](#)
 - Instalación de Prometheus y Node Exporter: Prometheus documentation
 - Video Explicativo de la unión de estas aplicaciones [Video aquí](#)
 - [Panel de Grafana](#) usado para la exportación de datos
-

Anexos

- Manuales de usuario para OMV, Docker, Prometheus y Grafana.
- Diagramas de red mostrando la arquitectura del sistema.
- Configuraciones específicas utilizadas durante la implementación.