



TP FINAL

LABORATORIO DE REDES Y SISTEMAS OPERATIVOS



GRUPO:

- Boglione Martin
- Borzi Emiliano
- García Alan

Consignas

- Instalar servidor Navidrome linux
- Probar con algún cliente mobile y/o desktop que soporte Subsonic API
- Sistema de back ups
- Instalar monitoreo con Grafana (docker)

<https://www.navidrome.org/docs/installation/>

NAVIDROME

¿Qué es?

Navidrome es un servidor de música en streaming de código abierto que permite a los usuarios acceder a su biblioteca de música desde cualquier dispositivo conectado a internet.

¿Por que elegimos este proyecto?

Navidrome nos pareció una buena idea como proyecto debido a sus beneficios:

- **Flexibilidad:** Permite a los usuarios acceder a su musica desde cualquier dispositivo.
- **Personalización:** Permite a los usuarios personalizar su biblioteca de música y acceder a ella de manera flexible.
- **Código abierto:** Significa que los desarrolladores pueden modificar y personalizar el código para adaptarlo a sus necesidades específicas.

PASOS A SEGUIR PARA INSTALAR NAVIDROME

Empezar actualizando el sistema:

```
sudo apt update
sudo apt upgrade
sudo apt install vim ffmpeg
```

Crear un directorio

Cree un directorio para almacenar el ejecutable de Navidrome y un directorio de trabajo con los permisos adecuados.

Cambiar <user> y <group> por las credenciales del sistema:

```
sudo install -d -o <user> -g <group> /opt/navidrome
sudo install -d -o <user> -g <group> /var/lib/navidrome
```

Descargar Navidrome

Descargue la última versión desde la [página de versiones](#).

En este caso vamos a utilizar la versión **v0.57.0**

```
wget https://github.com/navidrome/navidrome/releases/download/v0.57.0/na
sudo tar -xvzf Navidrome.tar.gz -C /opt/navidrome/
sudo chmod +x /opt/navidrome/navidrome
sudo chown -R <user>:<group> /opt/navidrome
```

Crear un archivo de configuración

Primero crear una carpeta navidrome en **/etc**

```
sudo mkdir -p /etc/navidrome
```

En el directorio `/etc/navidrome` crear un nuevo archivo llamado `navidrome.toml`

Luego crear el archivo **navidrome.toml**

```
sudo nano navidrome.toml
```

Y agregarle:

```
MusicFolder = "/carpeta/carpeta/"
```

ej: /var/lib/navidrome/music

Crear la carpeta:

```
sudo mkdir -p /var/lib/navidrome/music
```

```
sudo chown -R <user>:<group> /var/lib/navidrome/music
```

Crear un archivo navidrome.service

En el directorio **/etc/systemd/system/** crear el archivo **navidrome.service**

```
sudo nano navidrome.service
```

Adentro, agregar las siguientes líneas y cambiar el **<user>** y **<group>** por el usuario actual.

Unit]

Description=Navidrome Music Server and Streamer compatible with Subsonic
After=remote-fs.target network.target
AssertPathExists=/var/lib/navidrome

Install]

WantedBy=multi-user.target

Service]

User=<user>

Group=<group>

Type=simple

ExecStart=/opt/navidrome/navidrome --configfile "/etc/navidrome/navidrome

WorkingDirectory=/var/lib/navidrome

```

TimeoutStopSec=20
KillMode=process
Restart=on-failure

# See https://www.freedesktop.org/software/systemd/man/systemd.exec.htm
DevicePolicy=closed
NoNewPrivileges=yes
PrivateTmp=yes
PrivateUsers=yes
ProtectControlGroups=yes
ProtectKernelModules=yes
ProtectKernelTunables=yes
RestrictAddressFamilies=AF_UNIX AF_INET AF_INET6
RestrictNamespaces=yes
RestrictRealtime=yes
SystemCallFilter=~@clock @debug @module @mount @obsolete @reboot @
ReadWritePaths=/var/lib/navidrome

# You can uncomment the following line if you're not using the jukebox This
# will prevent navidrome from accessing any real (physical) devices
#PrivateDevices=yes

# You can change the following line to `strict` instead of `full` if you don't
# want navidrome to be able to write anything on your filesystem outside of
# /var/lib/navidrome.
ProtectSystem=full

# You can uncomment the following line if you don't have any media in /home
# This will prevent navidrome from ever reading/writing anything there.
#ProtectHome=true

# You can customize some Navidrome config options by setting environment
#Environment=ND_BASEURL "/navidrome"

```

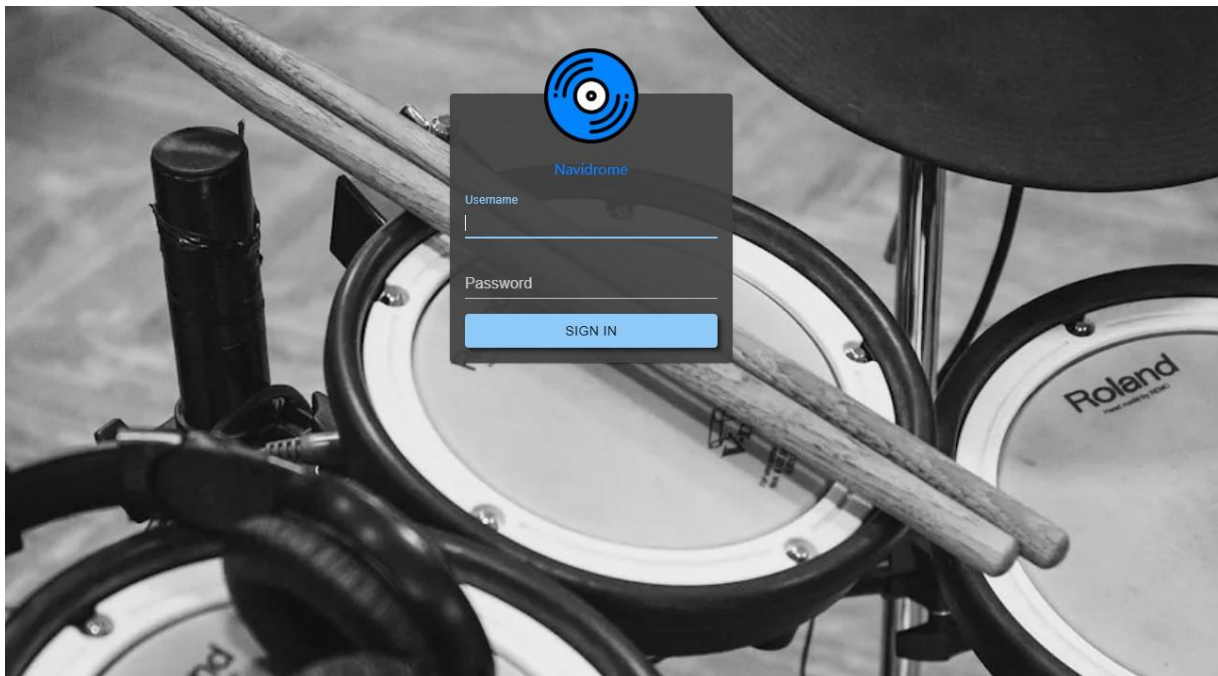
Iniciar navidrome.service

Recargar el daemon-reload, iniciar el servicio recién creado y verificar que se haya iniciado correctamente

```
sudo systemctl daemon-reload  
sudo systemctl start navidrome.service  
sudo systemctl status navidrome.service
```

Una vez corriendo, en el navegador web buscar: <http://localhost:4533>

Después crear un usuario con contraseña.



Para frenarlo

```
sudo systemctl stop navidrome.service
```

Para reiniciar

```
sudo systemctl restart navidrome.service
```

Iniciar Navidrome al inicio

```
sudo systemctl enable navidrome.service
```

Creación de otros usuarios en web de Navidrome

Con la cuenta Admin logueada, en configuración buscar el apartado "users".
Añadir las credenciales del usuario nuevo y aceptar.

Subsonic API

La API de Subsonic viene habilitada por defecto en Navidrome.

Navidrome implementa la API Subsonic como una forma de ser compatible con múltiples aplicaciones de música ya existentes (como Substreamer, DSub, Ultrasonic, Navidrome UI, etc).

Esto permite que puedas:

- Usar apps móviles que se conectan vía Subsonic
- Controlar la reproducción y navegación remota
- Hacer integraciones con clientes compatibles con Subsonic

Cliente mobile que soporta Subsonic API

En Android se puede descargar la app **Ultrasonic** de la PlayStore. Esta app es un cliente de código abierto para servidores compatibles con la API de Subsonic.

Una vez instalada se puede conectar a un nuevo servidor Navidrome, en este caso el que está corriendo en el localhost y utilizar las credenciales del usuario.

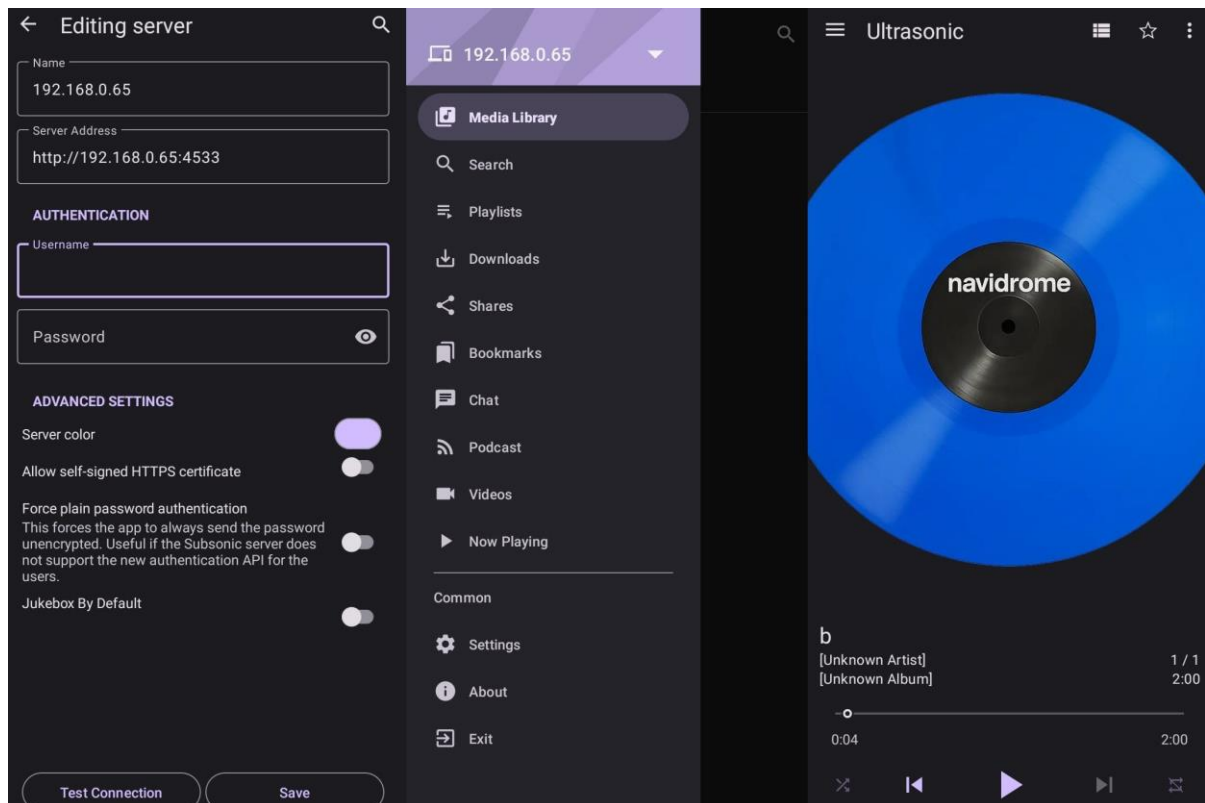
Nombre: Navidrome

Dirección del servidor: https://>IP^:4533

Autenticación:

Nombre del usuario: <user>

Contraseña: <password>



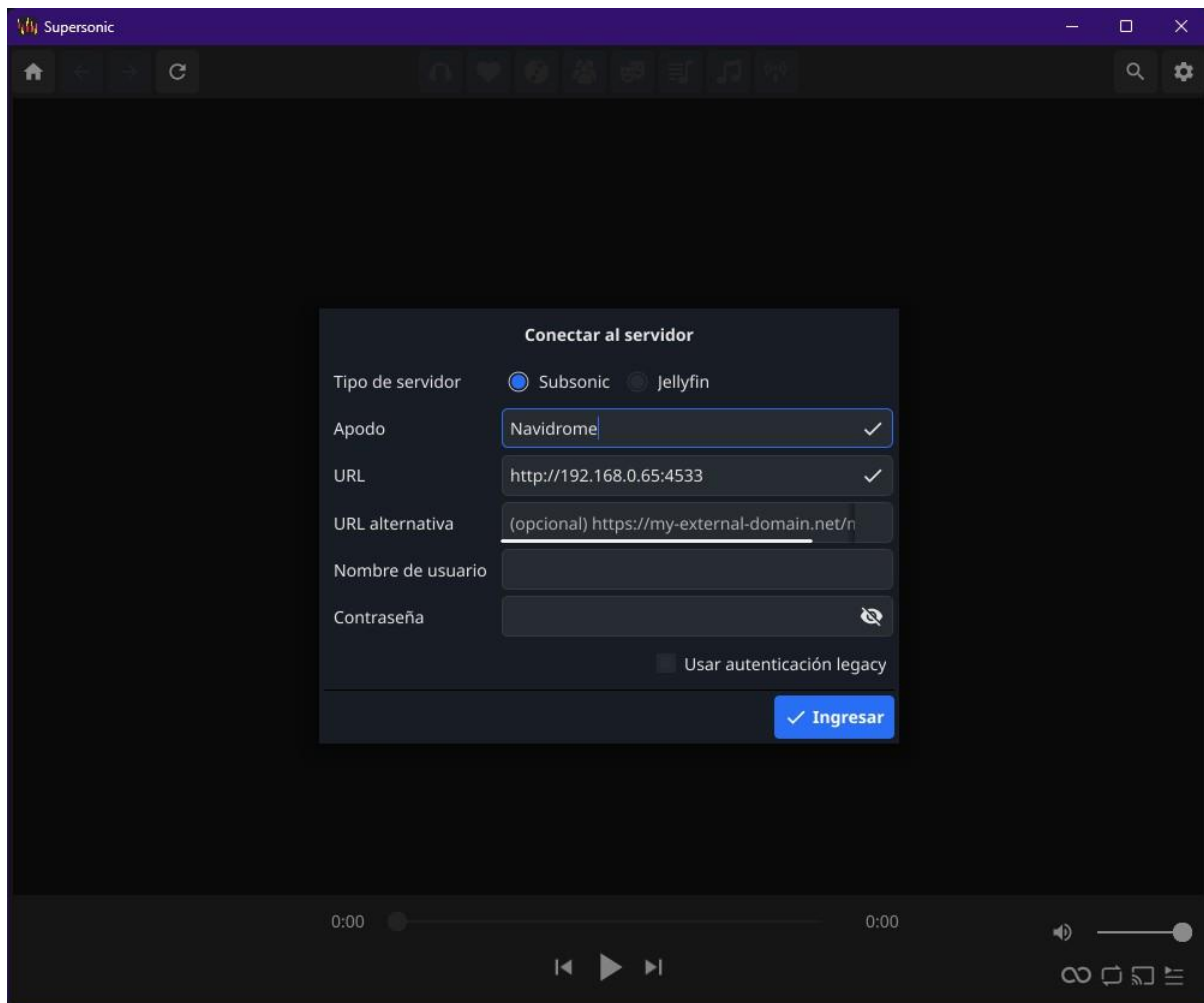
Cliente desktop que soporta Subsonic API

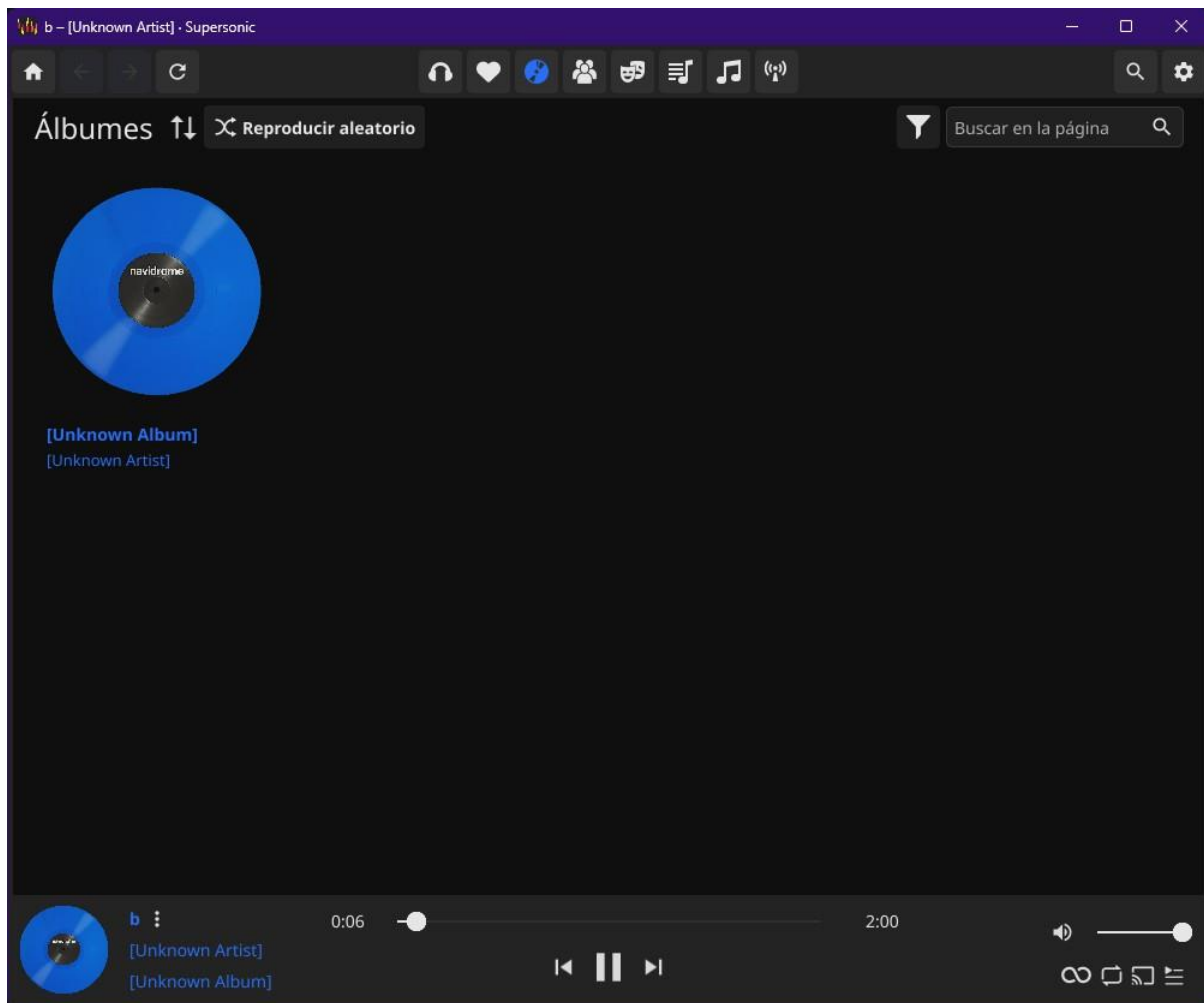
Un cliente desktop puede ser el proyecto Supersonic, un cliente de escritorio multiplataforma liviano para servidores de música Subsonic y Jellyfin.

Página para descargar e instalar Supersonic:

<https://github.com/dweymouth/supersonic>.

Poner las credenciales y tildar "Subsonic".





Problema posible:

Cuando usás una máquina virtual, la forma en que la VM se conecta a la red depende del modo de red que le configures.

Si la IP del server no tiene IP local del mismo rango que otros dispositivos (`10.x.x.x` o `192.168.x.x`), entonces no va a poder hablar con otros dispositivos de tu red local (LAN).

Solución: Configurar el adaptador de red correctamente

Cambiar la VM para que esté en modo puente (Bridged Adapter).

Esto hace que **tu VM aparezca como otro dispositivo real en tu red local.**

Así tu celular o tu PC pueden acceder a Navidrome usando esa IP.

- Apagá la VM.
- En VirtualBox, seleccioná tu VM * **Configuración**.
- **Red**.
- En “**Conectado a:**”, elegí **Adaptador Puente** (*Bridged Adapter*).
- En “**Nombre:**”, elegí tu adaptador de red física (Ethernet o Wi-Fi, el que estés usando para salir a Internet).

Música

Descargar musica: <https://pixabay.com/music/>

Como mostrar la música en Navidrome:

Agregar la música dentro del directorio puesto en `MusicFolder = "/carpeta/carpeta/"` .

La música debe estar por carpetas. Esquema: Artista * Album * Tema.mp3

Modificar los tags:

El nombre del artista, álbum y canción depende de los tags que tenga el archivo mp3.

Para modificarlos manualmente instalar:

```
sudo apt install easytag
```

Ejecutar comando easytag para correr el programa y modificar el archivo

Sistema de Back ups

En el directorio `/etc/navidrome/` , al archivo **navidrome.toml** agregar las siguientes líneas:

Backup]

Path = "/path/to/backup/folder"

Count > 7

Schedule = "0 0 * * *"

Path recomendado: /backup/navidrome

```
sudo mkdir -p /backup/navidrome
```

```
sudo chown <user>:<group> /backup/navidrome
```

 (no los crea sin permisos)

Desglose de "0 0 * * *" :

- 0 minuto: en el minuto 0
- 0 hora: en la hora 0 (medianoche)
- día del mes: todos los días del mes
- mes: todos los meses
- día de la semana: todos los días de la semana

¿Qué significa en otras palabras?

Ejecutar la tarea todos los días a la medianoche (00:00).

```
emi@Ubuntu:/backup/navidrome$ ls
navidrome_backup_2025.07.16_18.54.01.db
navidrome_backup_2025.07.16_18.55.06.db
navidrome_backup_2025.07.16_18.56.05.db
navidrome_backup_2025.07.16_18.59.45.db
```

Backup manual

Ejecutar:

```
sudo systemctl stop navidrome.service
```

```
sudo /opt/navidrome/navidrome backup create
```

Si tira el error "No existing database":

agregar en **navidrome.toml**:

```
DataFolder = "/var/lib/navidrome"
```

Ese es el directorio donde **Navidrome espera encontrar** `navidrome.db` .

El backup manual lo guarda en la misma carpeta que el .db original si se hace sin configurar el automático al no existir el path en .toml.

Otro error:

Si vuelve a aparecer "No existing database"

```
sudo /opt/navidrome/navidrome --configfile /etc/navidrome/navidrome.toml backup create
```

Para decirle explícitamente que use el archivo de configuración.

Restaurando un Backup

Cuando se restaura un backup, los datos existentes en la base de datos se borran y los datos del backup se copian en la base de datos.

Este comando se debe ejecutar mientras la aplicación de Navidrome no esté activa.

```
sudo systemctl stop navidrome.service
```

```
ls -lh /backup/navidrome
```

 → para ver lista de backups

```
sudo /opt/navidrome/navidrome backup restore -b /backup/navidrome/nombreadarchivo
```

Hay que agregar el **flag** del tipo de backup.

-b: para el path del backup de db

Crea 2 archivos: .db-shm y .db-wal.

.db-wal: Write-Ahead Log, almacena los cambios antes de escribir en la DB.

.db-shm: Shared Memory, ayuda con la coordinación entre procesos concurrentes.

Si existe un error:

```
sudo /opt/navidrome/navidrome --datafolder /var/lib/navidrome backup restore -b  
/backup/navidrome/nombrearchivo
```

 Hay que indicarle el `DataFolder` **explícito**.

Monitoreo con Grafana y Prometheus

Prometheus se integra con Grafana y recopila datos de exporters o servicios que exponen métricas en un formato especial vía HTTP (por ejemplo, /metrics).

Mientras que **Grafana** puede conectarse a Prometheus para mostrar dashboards gráficos con las métricas que Prometheus recolecta.

Prerrequisitos:

- Docker Engine ^> 1.13
- Docker Compose ^> 1.11

Instalar Docker Engine y Docker Compose:

Docker es una **plataforma de código abierto para desarrollar, desplegar y ejecutar aplicaciones en contenedores**. Los contenedores son unidades estandarizadas que empaquetan el código de la aplicación con sus dependencias, permitiendo que se ejecute de manera consistente en cualquier entorno. Docker simplifica la gestión y despliegue de aplicaciones, especialmente en arquitecturas de microservicios, donde cada servicio puede ejecutarse en su propio contenedor.

Un contenedor Docker es un **paquete de software estandarizado y ligero que incluye todo lo necesario para ejecutar una aplicación**: código, bibliotecas, herramientas del sistema y configuraciones.

Paso 1: Instalar paquetes necesarios

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install ca-certificates curl gnupg lsb-release
```

Paso 2: Agregar la clave GPG oficial de Docker

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian(o ubuntu)/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Paso 3: Configurar el repositorio oficial de Docker

```
echo \
"deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/debian(o ubuntu) \
$(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Paso 4: Instalar Docker Engine

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

Paso 5: Verificar instalación

```
sudo docker version
docker compose version
sudo docker run hello-world
```

Si ves un mensaje que dice "Hello from Docker!", todo está funcionando correctamente!

Clonar el repositorio

Esta es una solución de monitoreo para Docker hosts y containers con Prometheus y Grafana

Clonar el repositorio.

```
git clone https://github.com/Einsteinish/Docker-Compose-Prometheus-and-G
```

Ir a la carpeta:

```
cd Docker-Compose-Prometheus-and-Grafana
```

Create .env:

```
nano .env
```

Contenido del .env

```
ADMIN_USER=admin  
ADMIN_PASSWORD=admin
```

Modificar navidrome.toml:

```
Prometheus.Enabled = true  
Prometheus.MetricsPath = "/metrics"
```

Modificar algunos elementos del archivo **docker-compose.yml**:

```
services:  
  prometheus:  
    ports:  
      - "9090:9090"  
  grafana:  
    env_file:  
      - config  
    ports:  
      - "3000:3000"  
    enviroment:  
      GF_SECURITY_ADMIN_USER=admin  
      GF_SECURITY_ADMIN_PASSWORD=admin  
      GF_USERS_ALLOW_SIGN_UP=false  
  caddy:
```

```
ports:  
  - (sacar los que se repiten 3000 y 9000)
```

Editar

`prometheus/prometheus.yml` , añadir esto:

```
scrape_configs:  
  - job_name: 'navidrome'  
    metrics_path: /metrics  
static_configs:  
  - targets: ['<tuip>:4533']
```

Levantar servicios ejecutando:

```
sudo docker compose up -d
```

Para frenar de ser necesario:

```
sudo docker compose down
```

Containers:

- Prometheus (metrics database) `http://<host-ip>:9090`
- Prometheus-Pushgateway (push acceptor for ephemeral and batch jobs) `http://<host-ip>:9091`
- AlertManager (alerts management) `http://<host-ip>:9093`
- Grafana (visualize metrics) `http://<host-ip>:3000`
- NodeExporter (host metrics collector)
- cAdvisor (containers metrics collector)

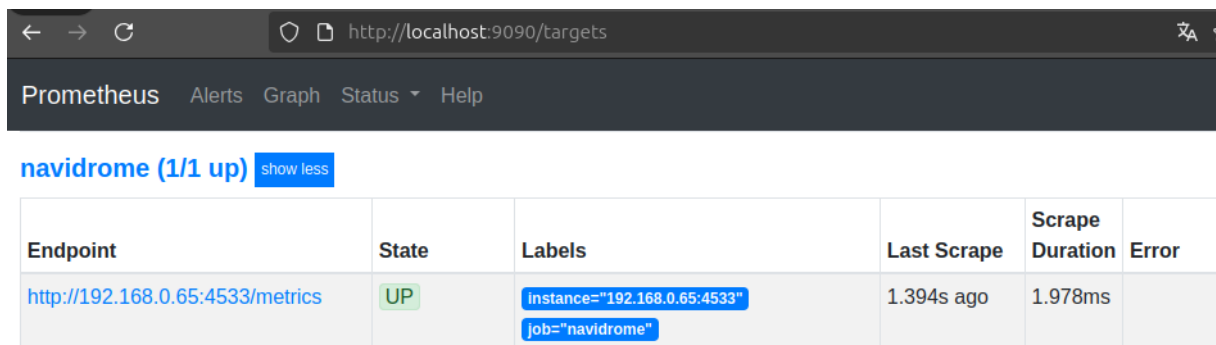
- Caddy (reverse proxy and basic auth provider for prometheus and alertmanager)

Verificar que Prometheus vea a Navidrome

Ir a:

`http://localhost:9090/targets`

`navidrome:4533` debería aparecer como activo.



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.0.65:4533/metrics	UP	instance="192.168.0.65:4533" job="navidrome"	1.394s ago	1.978ms	

Configurar Grafana (viene por defecto configurado)

1. Ingresa a Grafana `http://localhost:3000` (admin/admin por defecto).
2. Ve a **Connections** ≡ **Data Sources** ≡ **Add data source**.
3. Elige **Prometheus**.
4. URL: `http://prometheus:9090` (usando el nombre del servicio definido en Docker Compose).
5. Haz clic en **Save & Test**, debe conectar sin errores.

Creación de dashboard en Grafana

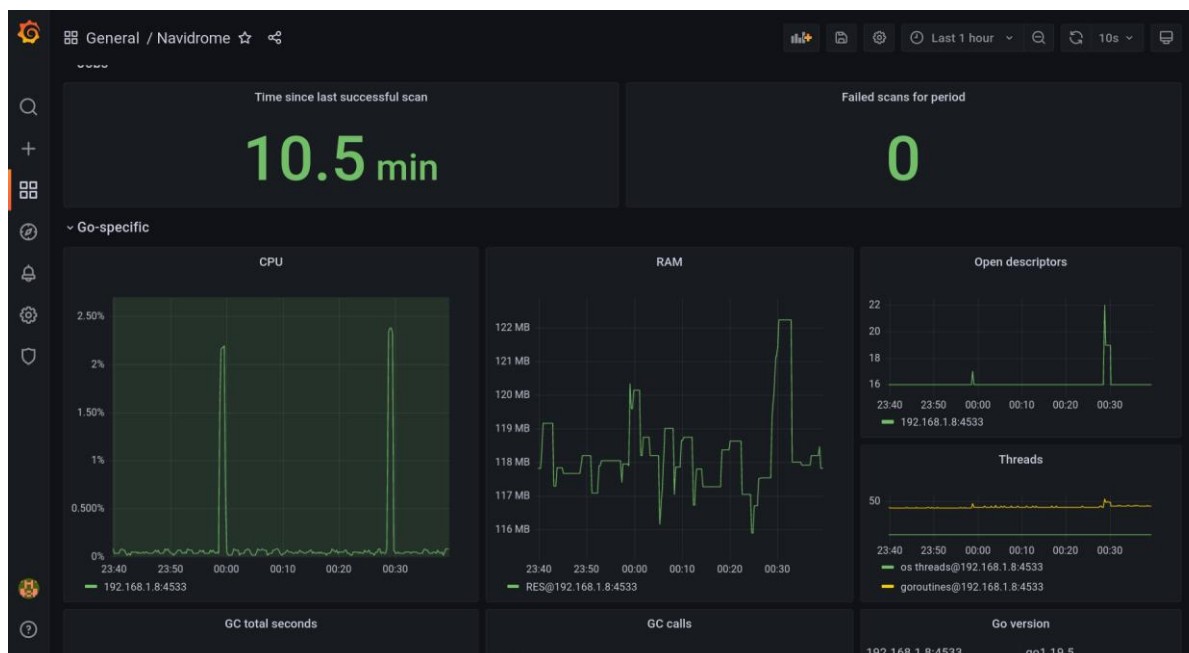
Ir a + "Create dashboard".

En New panel, "Add query".

Agregar lo deseado y volver al dashboard.

En Add panel, se pueden agregar más métricas.

En Dashboard Settings, configurar nombre, guardarlo, etc.



Algunas métricas de Navidrome:

Versión Navidrome: `navidrome_info`, Stat o Gauge.

Uso de CPU Navidrome (%): `rate(process_cpu_seconds_total[1m]) * 100`, Graph.

Memoria usada: `process_resident_memory_bytes`, Graph o Gauge con unidad "bytes".

Total de usuarios : `db_model_totals{model="user"}`, Stat.

Total de artistas: `db_model_totals{model="artist"}`, Stat.

Total de álbumes : `db_model_totals{model="album"}`, Stat.

Escaneos realizados: `media_scans`, Stat.

Último escaneo: `media_scan_last`, Stat o Gauge con unidad "date/time".

Latencia de streaming: `http_request_latency{endpoint="/rest/stream", quantile="0.9"}`, Línea o Gauge.

Cantidad de requests HTTP por endpoint (último minuto): `sum by (endpoint) (rate(http_request_count[1m]))`, Table o Bar Gauge.

Errores y problemas

En la instalación de Navidrome tuvimos un problema que no arrancaba por error de los directorios. Pero fue solucionado.

Tanto al momento de la creación y uso de las carpetas de backup/música, tuvimos problemas porque no tenían los permisos necesarios.

Al utilizar solo el backup manual no quedaban los archivos en la carpeta de backups porque no tenía la ruta.

Con los clientes subsonic no hubo casi problema alguno.

Lo que más trabajo nos llevó fue docker y su contenido, su instalación y configuración.

No podíamos encontrar la configuración para hacer que Prometheus vea a Navidrome.

Hasta que arreglamos que las métricas tengan un path. Luego no sabíamos identificar que métricas eran necesarias para Grafana.