

# Despliegue de un servidor de almacenamiento en la nube basado en contenedores

**Guía de despliegue**

Autor/es: Ismael Carrasco Cubero

Fecha: 02/05/2024

<b>1. Introducción.....</b>	<b>4</b>
<b>2. Estimación económica.....</b>	<b>5</b>
<b>3. Recursos Humanos.....</b>	<b>5</b>
<b>4. Planificación temporal .....</b>	<b>6</b>
<b>5. Viabilidad y objetivos del proyecto .....</b>	<b>8</b>
<b>6. Preparación del sistema base.....</b>	<b>10</b>
<b>6.1 Elección del sistema base. Principales ramas del Kernel Linux. ....</b>	<b>10</b>
<b>6.2 Preparación del sistema base. El hardware.....</b>	<b>12</b>
<b>6.3 Preparación del sistema base. Instalación de Fedora Linux 39 Server Edition .....</b>	<b>13</b>
<b>7.Consolas de Administración del servidor.....</b>	<b>21</b>
<b>7.1 La consola de administración Cockpit.....</b>	<b>21</b>
<b>7.1.1 Cockpit. Configuración de nombre de Host.....</b>	<b>25</b>
<b>7.1.2 Cockpit. Actualización del sistema.....</b>	<b>27</b>
<b>7.1.3 Cockpit. Configuración de IP estática del servidor .....</b>	<b>29</b>
<b>7.2 La consola de administración Webmin .....</b>	<b>31</b>
<b>7.2.1 Webmin. Instalación .....</b>	<b>31</b>
<b>7.2.2 Webmin. Configuración del servicio SSH.....</b>	<b>38</b>
<b>7.3 Consola remota SSH. Autorización del cliente con clave publica.....</b>	<b>40</b>
<b>8. El runtime de contenedores. RedHat Podman .....</b>	<b>43</b>
<b>8.1 Podman. Diferencias respecto a docker .....</b>	<b>44</b>
<b>8.2 Podman. Instalación.....</b>	<b>45</b>
<b>8.3 Podman. Creación de la red.....</b>	<b>47</b>
<b>9. Servicios del servidor. Descripción y despliegue.....</b>	<b>49</b>
<b>9.1 Stack Nextcloud.....</b>	<b>49</b>
<b>9.1.1 El Sistema Gestor de Bases de Datos PostgreSQL.....</b>	<b>50</b>
<b>9.1.2 PostgreSQL. Despliegue del pod y primer contenedor.....</b>	<b>51</b>
<b>9.1.3 Consola de administración del SGBD. PGAdmin .....</b>	<b>54</b>
<b>9.1.4 PGAdmin. Despliegue del contenedor y preparación de la base de datos .....</b>	<b>56</b>
<b>9.1.5 Contenedor Nextcloud. Despliegue y configuración básica .....</b>	<b>63</b>
<b>9.1.6 Stack Nextcloud. Despliegue automatizado de pods con Podman Kube y manifiesto Yaml.....</b>	<b>67</b>
<b>9.1.7 Stack Nextcloud. Anexo I: Creación de usuarios en la plataforma.....</b>	<b>69</b>
<b>9.1.8 Stack Nextcloud. Anexo II: Sincronización de archivos con cliente de escritorio.....</b>	<b>73</b>
<b>9.2 Stack LDAP.....</b>	<b>85</b>
<b>9.2.1 Stack LDAP. OpenLDAP .....</b>	<b>86</b>
<b>9.2.2 Stack LDAP. LDAP Account Manager (LAM).....</b>	<b>86</b>
<b>9.2.3 Stack LDAP. Despliegue y configuración básica .....</b>	<b>88</b>
<b>9.3 Stack Grafana.....</b>	<b>105</b>
<b>9.3.1 Stack Grafana. El panel de métricas Grafana.....</b>	<b>105</b>
<b>9.3.2 Stack Grafana. Prometheus .....</b>	<b>105</b>

<b>9.3.3 Stack Grafana. Node Exporter</b> .....	106
<b>9.3.4 Stack Grafana. Despliegue y configuración básica</b> .....	107
<b>9.4 Contenedor Caddy Server</b> .....	114
<b>9.4.1 Caddy server. El servidor multiuso</b> .....	114
<b>9.4.2 Caddy Server. Esquema lógico de la red</b> .....	115
<b>9.4.3 Caddy Server. Aprovisionamiento del Caddy file y despliegue del contenedor</b> .....	116
<b>9.4.4 Caddy Server. Anexo I: Configuración adicional de nextcloud para el proxy Inverso</b> .....	122
<b>10. Script de automatización de tareas. El asistente de despliegue en bash</b> .....	123
<b>10.1 Script de automatización de tareas. Despliegue inicial</b> .....	124
<b>10.2 Script de automatización de tareas. Introducción de datos en servicio de directorio</b> .....	128
<b>10.4 Script de automatización de tareas. Restauración de una instancia previa</b> .....	130
<b>11. Copias de seguridad y resiliencia</b> .....	132
<b>11.1 Copias de seguridad y resiliencia. El comando rsync</b> .....	132
<b>11.1.1 Comando rsync. Ejecución y automatización</b> .....	137
<b>11.2 Copias de seguridad y resiliencia. Puesta a prueba en casos de desastres o migración</b> .....	140
<b>12. Biliografia</b> .....	144

# 1. Introducción

Este proyecto final del ciclo superior ASIR, versa sobre el despliegue de una instancia de nextcloud; una suite de productividad gratuita y OpenSource que incluye gran cantidad de funcionalidades útiles para cualquier entorno, bien sea particular o empresarial.

Su eje central es la aplicación de almacenamiento de archivos, que permite la conexión con el servidor de la instancia desde cualquier red que sea accesible, permitiendo además la sincronización de los archivos entre el servidor y los dispositivos conectados a través de aplicaciones cliente (con versiones para los principales sistemas operativos del mercado), siendo muy útil como almacén centralizado de archivos y documentos.

Permite la edición colaborativa de documentos mediante diversas aplicaciones instalables en el servidor, como collabora online u onlyoffice; así como otra gran cantidad de funcionalidades como streaming multimedia de los medios almacenados, y gestión de usuarios con acceso al servicio mediante servidor LDAP.

No obstante, Nextcloud no es más que la excusa para desarrollar otra idea central, que es la que se ha de valorar como idea principal en este proyecto: El despliegue de un servidor con múltiples servicios, usando runtimes de contenedores.

El paradigma de los contenedores no es nuevo, existe desde aproximadamente el año 2000 con FreeBSDjail, una tecnología que permitía subdividir el sistema en distintas jails o “jaulas” que proporcionaban aislamiento de procesos.

Actualmente un contenedor es una porción del sistema operativo Linux aislada del resto, que es capaz de ejecutar procesos que quedan teóricamente aislados de todo el resto de procesos. Se consigue de esta forma crear una suerte de pequeñas “pseudomáquinas virtuales” que son capaces de comportarse como un host independiente de otros contenedores y del propio sistema host (no obstante, no son máquinas virtuales al uso, pues entre otras cosas utilizan el kernel del host).

## 2. Estimación económica

Los costes asociados a este proyecto son muy bajos, gracias a su naturaleza OpenSource. En la siguiente tabla se detalla los distintos recursos necesarios, con su coste asociado, de darse el caso.

Recurso	Coste
HP EliteDesk 800 G2	106€
SO Fedora Linux 39 Server Edition	Gratis (Licencia GPL V3)
Instancia de Nextcloud	Gratis (Licencia GPL V3)
SGBD PostgreSQL	Gratis (Licencia PostgreSQL)
Consola de administración Cockpit	Gratis (Incluido en Fedora)
Consola de administración PGAdmin	Gratis (Licencia PostgreSQL)
Proxy Inverso CADDY	Gratis (Licencia Apache 2.0)
Firewall	Gratis (Incluido en Fedora)
Servicio de directorio OpenLDAP	Gratis (Licencia Pública OpenLDAP)
LDAP Account Manager	Gratis (Licencia GPL V3)
Registro DNS "Enhanced" en NoIP	18,69€/Año

El coste total aproximado del proyecto es de tan solo 124,69€

## 3. Recursos Humanos

Este proyecto como tal no implica la creación de ninguna empresa, pero un esbozo de que departamentos de una hipotética empresa se encargarían de cada parte del proceso podría ser la siguiente:

Forma en la que participa el departamento	Departamento
Despliegue de los contenedores, configuración del sistema operativo y su conectividad de red	Departamento de Sistemas
Creación de la base de datos de la instancia	Departamento de Sistemas
Configuración del proxy y el firewall	Departamento de Ciberseguridad
Gestión de los usuarios y los grupos de la instancia	Departamento de Recursos Humanos
Monitorización y control del servidor	Departamento de Sistemas

## 4. Planificación temporal

Resulta muy complicado establecer una distribución temporal, para la consecución de metas en el proyecto. Por experiencia personal, cuando se trata de computación, ciertas tareas que pueden parecer complejas, pueden ser resueltas en tiempo récord, y otras cuya complejidad parece en principio muy baja, pueden complicarse “ad eternum” por problemas imprevistos.

No obstante, realizare un diagrama de Gantt con una estimación del tiempo de consecución de las tareas (basadas en los objetivos del proyecto).

En el queda representada una estimación aproximada del tiempo que tomara alcanzar cada uno de los hitos del proyecto, repartido en 5 semanas con segmentos que cubren un área proporcional al tiempo estimado para cada parte, siendo los verdes hitos individuales, y rojos conjuntos relacionados de los mismos. No obstante, dicho diagrama no incluye fechas de inicio o de finalización, porque simple y llanamente, me es imposible estimar fechas concretas.

Quede claro, no obstante, que los tiempos pueden variar enormemente respecto a lo representado a continuación.

Hitos	Mayo. Semana 1	Mayo. Semana 2	Mayo. Semana 3	Mayo. Semana 4	Junio. Semana 1
Instalación y configuración del sistema base. Fedora Linux 39	█				
Configurar la red del servidor	█				
Obtención y gestión de los nombres de dominio en NoIP	█				
Instalación y configuración de las consolas de administración	█	█			
Preparaciones previas en el runtime de contenedores	█	█			
Configuración del Firewall		█			
Despliegue del Proxy inverso, contenedor Caddy		█			
Despliegue del Pod "Stack Nextcloud"			█		
Despliegue del primer contenedor. PostgreSQL			█		
Despliegue del contenedor PgAdmin			█		
Despliegue del contenedor Nextcloud. Configuración básica			█		
Despliegue del Pod "Stack LDAP"			█		
Despliegue del contenedor del servicio de directorio OpenLDAP			█		
Despliegue del contenedor LAM (LDAP Account Manager)			█		
Despliegue el Pod "Stack de monitorización"			█		
Política de copias de seguridad. Definición y creación de script para el respaldo de la información				█	
Automatización del Despliegue. Creación del script bash				█	

## 5. Viabilidad y objetivos del proyecto

Pasemos a enumerar punto por punto de forma rápida y simplificada, que es lo que se pretende conseguir con este proyecto, y la viabilidad general del mismo. De forma general, siempre que no se especifique lo contrario, los servicios implicados serán ejecutados mediante runtime de contenedores Podman.

1. Instalación y configuración de un servidor Fedora Linux 39 Server edition, que alojara todos los componentes del proyecto
2. Proporcionar a dicho servidor, conectividad a Internet con el exterior para que pueda aceptar y resolver peticiones desde cualquier lugar
3. Asociar los servicios del mencionado servidor a diferentes nombres de dominio mediante un registro DNS con Wildcard, que permita acceder a los distintos componentes desde cualquier parte
4. Dotar al servidor de Herramientas de administración remota, tanto de consola (ssh) como gráficas (webmin y Cockpit). Dichas herramientas de administración, son de los pocos componentes del servidor, que serán ejecutados mediante paquetería RPM standard Linux, propia de los repositorios del proyecto Fedora.
5. Desplegar una instancia de la suite de productividad Nextcloud en el servidor
6. Dotar a la instancia de Nextcloud de un servidor de Base de datos PostgreSQL para el almacenamiento de la información interna
7. Desplegar una consola de administración PgAdmin para dicha instancia de PostgreSQL
8. Desplegar un servidor LDAP, con la funcionalidad básica necesaria para permitir el login a la instancia de Nextcloud usando su base de datos de directorio, eliminando así la necesidad de gestionar individualmente las cuentas y grupos de Nextcloud
9. Desplegar una consola de administración del directorio LDAP, para facilitar su gestión.
10. Desplegar y configurar un Firewall, para la gestión de la seguridad de red, de forma que solo se permitan conexiones por los puertos estrictamente necesarios.
11. Desplegar alguna consola de monitorización del servidor, para que podamos comprobar en tiempo real el estado general del sistema, y visualizar posibles alertas y problemas. (Probablemente Grafana, pero aún por decidir)
12. Desplegar un Proxy Inverso, que nos permita redirigir peticiones basadas en nombre de dominio a los diversos componentes que conformaran el sistema.
13. Interconectar todos los componentes mencionados, de forma que cada componente por separado, y el conjunto al completo, funcionen a la perfección.
14. Establecer mecanismos y políticas de copias de seguridad para la resiliencia de todo el sistema.
15. Por último, se pretende crear un script en bash, que permita el despliegue inicial o el redespliegue de todo el sistema, de forma automatizada.

Todos los objetivos propuestos para este proyecto son factibles técnica y económicamente. En la cuestión técnica me siento perfectamente capacitado para alcanzar un nivel de cumplimiento cercano al 100% o incluso completo.

Por otra parte, el grado de viabilidad económica es perfectamente asumible tanto por mí de forma individual, como por parte de una empresa. Todas las tecnologías que se van a utilizar son proyectos de software libre, con distribuciones completamente gratuitas. Dispongo además de un pequeño servidor de pruebas con potencia más que suficiente para poder ejecutar todo el stack. En un entorno empresarial con necesidades de potencia superiores, probablemente habría

de realizarse una inversión adicional, pero doy por hecho, que esa hipotética empresa, ya dispondría de hardware propio para desplegar las soluciones que aquí se proponen, quedando solo la necesidad de añadir a su infraestructura el software, el cual como se ha mencionado es completamente Libre y gratuito en su totalidad, por lo que este proyecto no solo resulta interesante y útil para gran variedad de circunstancias, sino también económico de desarrollar.

## 6. Preparación del sistema base

Como cualquier sistema informático, nuestro servidor necesita antes de cualquier consideración, hardware y un sistema operativo que haga de intermediario entre dicho hardware y el software del proyecto.

La elección de dicho sistema operativo no ha estado exenta de dificultades. Windows estaba completamente descartado desde el principio, por ser un sistema cerrado y poco flexible a la hora de gestionar y moldear, además una licencia de la versión Servidor del sistema operativo de Microsoft requiere de una inversión económica considerable. Además, los runtimes de contenedores como el que se pretende utilizar, si bien tienen distribuciones instalables en Windows, son en realidad máquinas virtuales de Linux que corren en su interior dichos Runtimes.

Esto nos deja con alguna distribución Linux como la opción más razonable, gracias a la gran flexibilidad, robustez, estabilidad y carácter libre y gratuito de buena parte de las distros Linux existentes.

Elegido el Kernel, ¿Que distribución usar?

Linux tiene en la actualidad 4 “ramas” principales, con diferencias sutiles pero importantes a la hora de elegir la distro en función del uso que se busque dar al sistema.

A continuación, se describe brevemente el hardware que ejecutara el proyecto, así como el SO elegido para tal cometido.

### **6.1 Elección del sistema base. Principales ramas del Kernel Linux.**

Podemos distinguir las siguientes ramas dentro del ecosistema del pingüino, pasamos a describirlas brevemente:

- **Rama Debian:** Una de las más veteranas dentro del ecosistema. Basada en paquetería de tipo deb, sus mayores exponentes son Debian en sí mismo y Ubuntu, tal vez más popular, aunque la distribución original. Para un servidor, Debian es la elección más obvia, sin embargo, su relativa complejidad (más por una cuestión de cambiar ciertas nomenclaturas y archivos de configuración) y una selección de software muy conservadora que prima ante todo la estabilidad del software me hicieron descartarla rápidamente.
- **Rama Arch Linux:** Probablemente uno de los mayores exponentes del “hazlo tú mismo” y de las versiones Bleeding Edge de software. Arch Linux (Y sus derivadas) son distribuciones de liberación continua o “Rolling Release”. Si bien esto tiene ciertas ventajas para ciertos usos profesionales o particulares, está completamente desaconsejado su uso como servidor, pues tienen una tendencia natural a la inestabilidad y las “roturas” de software durante las actualizaciones.
- **Rama SuSE Linux Enterprise:** La rama SuSE Linux Enterprise es una de las más valoradas por el mundo corporativo y de grandes centros de datos, no sin motivo. Potente, flexible, estable e inusualmente fácil de administrar gracias a su herramienta de configuración del sistema Yast, Open SuSE Leap (Variante comunitaria de la versión empresarial) es una elección obvia para el despliegue de un servidor. Sin embargo, al igual que Debian, su aproximación algo conservadora, con paquetería en los repositorios primando la estabilidad por encima de la novedad, hace que la descarte.

- **Rama RedHat Enterprise Linux;** En competencia directa con SuSE, tenemos al que es probablemente el sistema operativo orientado al mundo empresarial más utilizado del mundo, RedHat enterprise. Dentro de dicho ecosistema podemos encontrar actualmente 3 Proyectos bien diferenciados, RedHat enterprise Linux y sus clones (Rocky Linux, Alma Linux y Oracle Linux), CentOS Stream (Versión de liberación continua de la paquetería de Redhat) y Fedora Linux.

Esta es mi elección final para sistema operativo.

Fedora es un sistema operativo comunitario, completamente Libre y gratuito y sin ningún tipo de restricción de uso.

Conocido muchas veces como “*El campo de pruebas de RedHat*” es actualmente la distribución Linux que abandera el progreso tecnológico del ecosistema del pingüino, sentando habitualmente cátedra sobre componentes que primero se despliegan en Fedora, y a posteriori el resto de distribuciones adoptan. Systemd, wayland, Pipewire y un largo etc, son solo algunos ejemplos de componentes habituales hoy en día en la mayoría de sistemas Linux, que vieron su nacimiento en el proyecto Fedora.

Se trata de un sistema operativo, con un ritmo de actualización rápida, con frecuentes actualizaciones.

Si bien a priori esto no parece ser la mejor idea para un servidor, los repositorios de Fedora son minuciosamente revisados antes de lanzar las actualizaciones, por lo que es un sistema operativo más estable de lo que a priori puede parecer. Además, puesto que es una distribución siempre a la vanguardia tecnológica, nos aseguramos de que nuestro sistema basado en contenedores (Una tecnología que recordemos, elimina los problemas de versiones y dependencias) este siempre a la última de la tecnología.

## 6.2 Preparación del sistema base. El hardware

Pero... ¿Dónde ejecutaremos todo el software? El autor de este proyecto dispone en su casa actualmente de un pequeño servidor en el que lleva tiempo realizando diversos experimentos por entretenimiento y ganas de aprender. En parte, dichos experimentos son el origen de la inspiración este proyecto final de ASIR.



**El servidor HP EliteDesk 800 G2**

La máquina elegida es una HP EliteDesk 800 G2. Se trata de una pequeña máquina de escritorio dedicada en su día al uso ofimático, aunque posee una potencia de proceso considerable para un uso de servidor como el que pretendemos desarrollar.

Es de muy reducido tamaño, extremadamente silencioso, y con una generación de calor mínima, lo cual lo hace muy apropiado para estar en funcionamiento ininterrumpido 24h/365d.

Pasamos a detallar brevemente sus principales características de hardware:

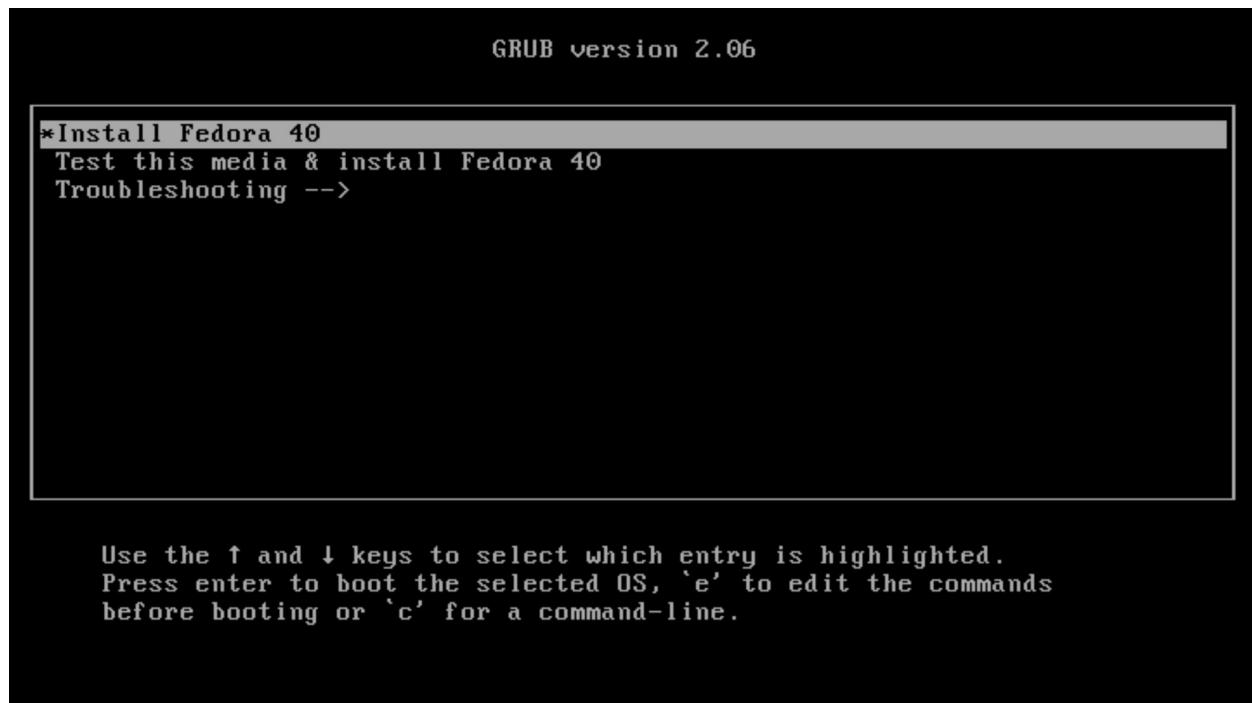
- CPU: Intel Core i5 6500T 2.5 GHZ. 4 núcleos Físicos / 4 Núcleos Lógicos
- RAM: 8GB DDR4 3200Mhz. Single channel en un único modulo
- Almacenamiento interno: Unidad de estado sólido SATA III, Sandisk 480GB
- Puertos: 6 Puertos USB 3.0, 1 puerto USB 3.0 tipo C
- Dispositivo de red: NIC Integrada Intel, Gigabit Ethernet

Si bien el hardware puede parecer modesto a primera vista, el uso de un sistema operativo Linux en modo servidor “headless” (sin entorno gráfico), junto al SSD interno, hacen que esta pequeña maquina sea capaz de manejar un número considerable de contenedores y otros servicios con total soltura. Es sobradamente capaz para entornos domésticos o empresariales de pequeño tamaño.

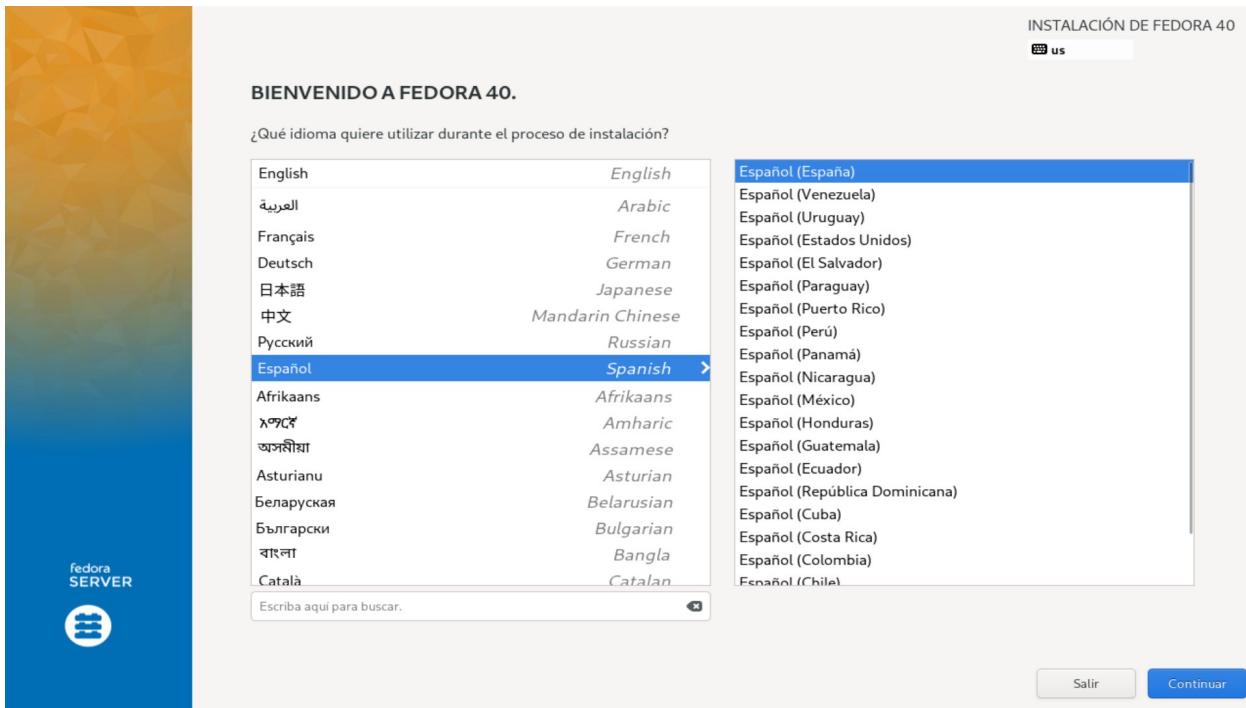
## 6.3 Preparación del sistema base. Instalación de Fedora Linux 39 Server Edition

Una vez hemos escogido el sistema Operativo en el que desplegar nuestro sistema, toca instalarlo. Procedemos pues a detallar los pasos para la instalación de la versión Server de dicha distribución.

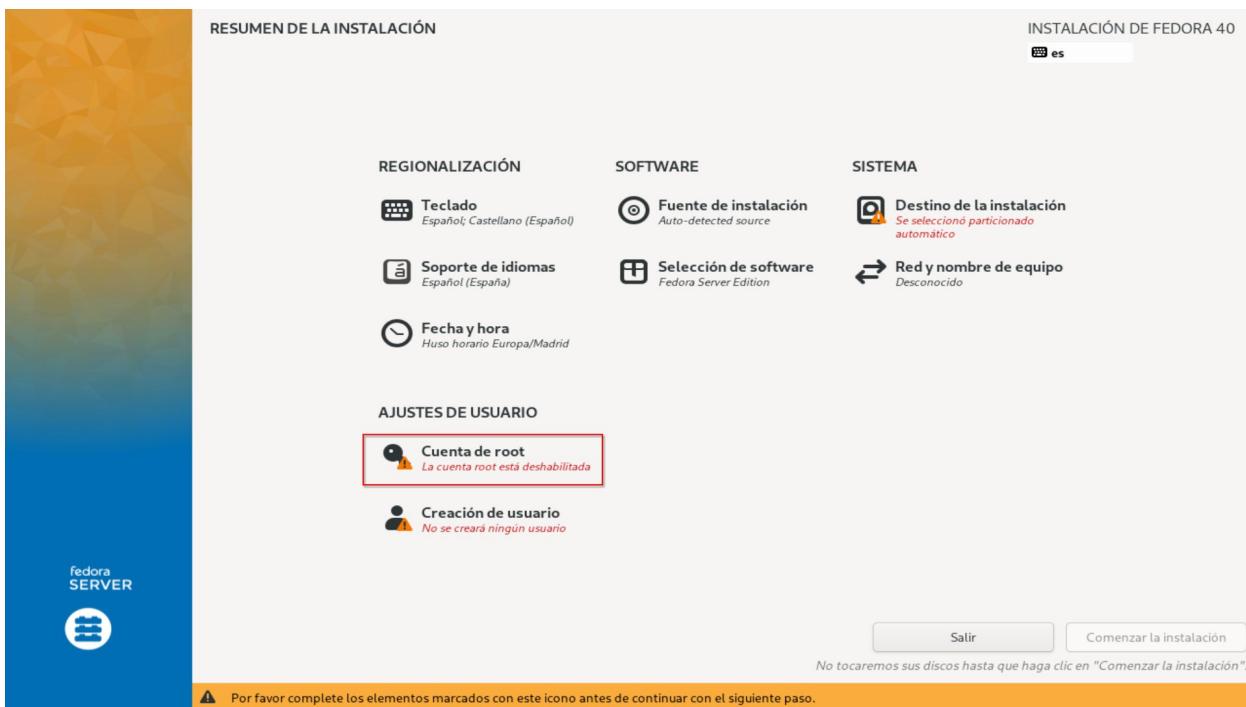
NOTA IMPORTANTE: En mi hardware, dicho sistema operativo ya está instalado y listo para comenzar el despliegue, pues es un SO que uso habitualmente para mis pruebas y experimentos. Las capturas a continuación son una simulación de la instalación del sistema operativo sobre una máquina virtual. Sin embargo, el proceso de instalación aquí mostrado, es idéntico a una maquina física, y solo difiere en que la versión aquí mostrada es Fedora 40 (la más actual) en lugar de la 39, que es la actualmente implementada en el servidor.



Tras haber descargado la Imagen ISO de la web oficial del Proyecto Fedora: [Fedora Linux Server Edition](#) debemos como resulta habitual “quemar” dicha imagen ISO en algún medio extraíble para crear el dispositivo de arranque. Una vez creado dicho medio y con la pertinente configuración de nuestra UEFI/BIOS para arrancar medios extraíbles, se nos muestra el selector GRUB del medio de instalación. Tenemos la opción de instalar directamente, o bien comprobar la unidad en busca de errores y a continuación ejecutar el instalador. Ambas opciones son perfectamente validas.



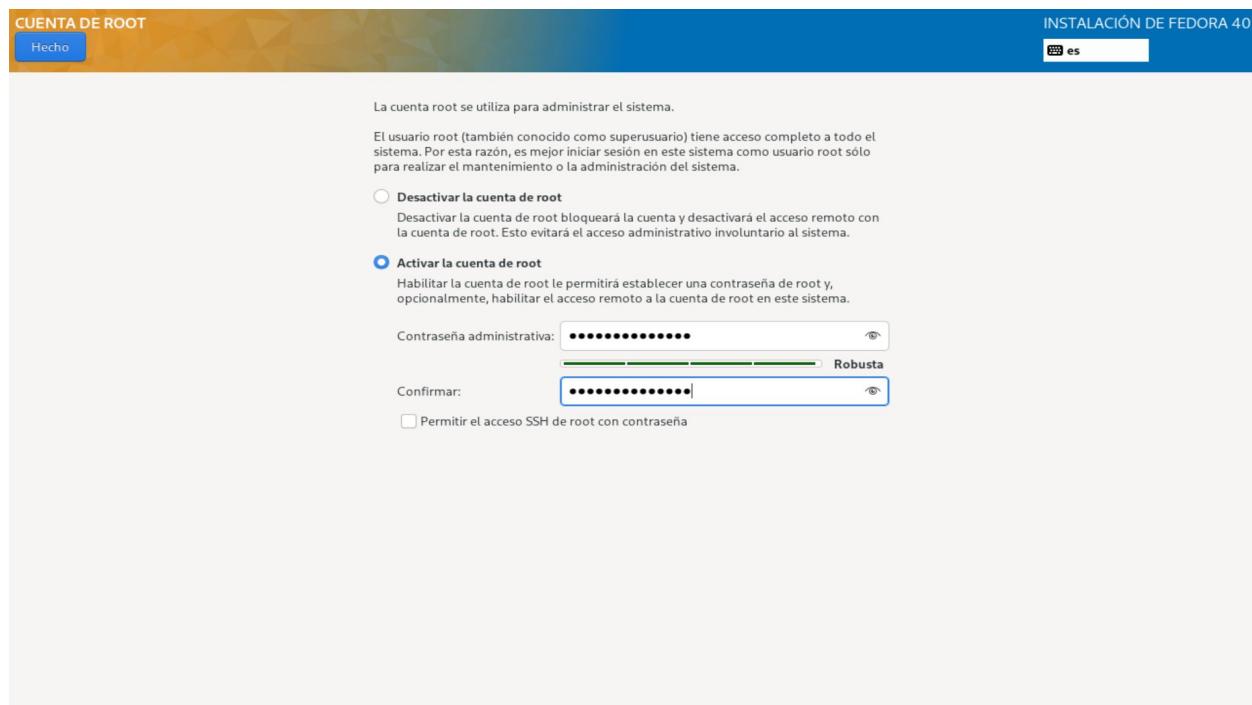
Una vez cargados todos los componentes del SO, el instalador Anaconda de Fedora nos da la bienvenida. En esta primera pantalla debemos seleccionar el idioma del sistema y la distribución de teclado. Una vez seleccionada, pulsamos en Continuar para pasar al siguiente paso.



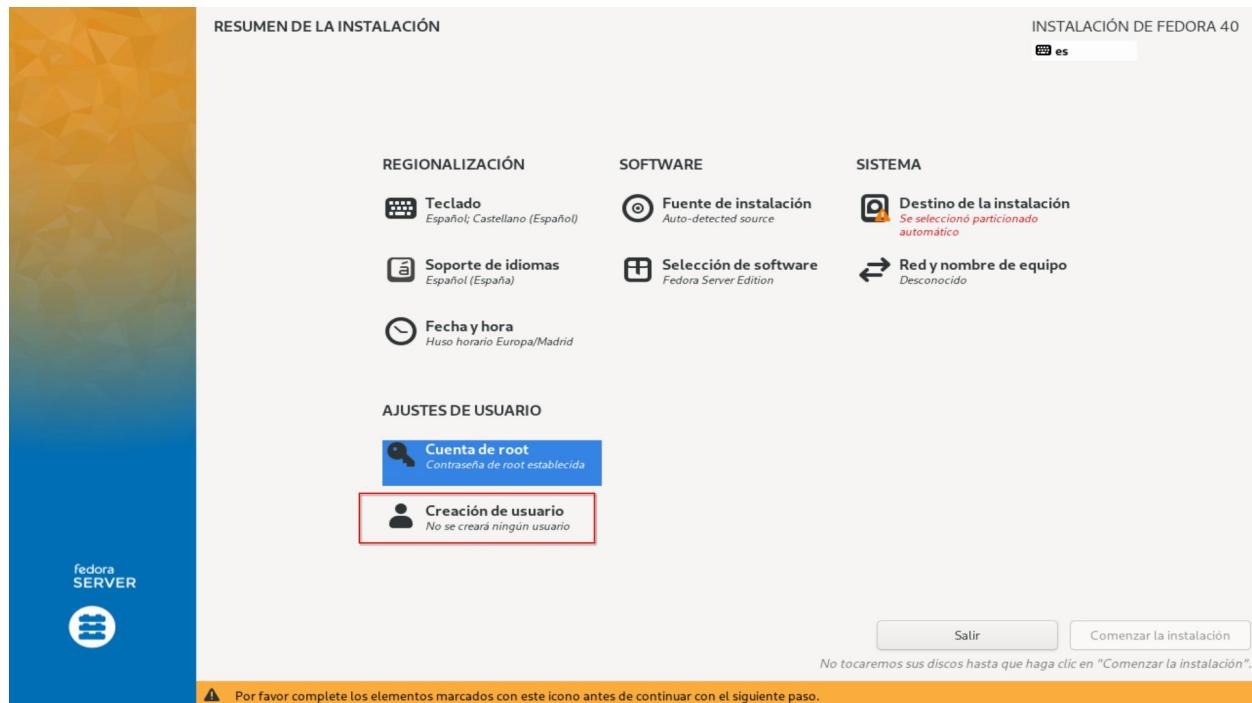
El instalador Anaconda es increíblemente sencillo de utilizar. Consta de un panel principal en el que se nos muestran los diferentes aspectos a configurar para el futuro sistema instalado.

Si algún apartado requiere de atención, aparecerá con una breve explicación en rojo. No importa el orden en el que decidamos configurar los apartados, solo debemos configurar aquellos que requieran de nuestra atención.

Comenzaremos con la configuración de la cuenta del todopoderoso usuario Root pulsando en su opción correspondiente.



Una vez en dicha pantalla de configuración, tenemos varias opciones a configurar, como desactivar completamente la cuenta root, activarla, asignar su contraseña en caso de activarla, así como permitir o restringir el acceso a dicha cuenta a través de la consola remota por ssh. En nuestro caso, habilitaremos el acceso root por ssh. Establecemos la contraseña deseada y pulsamos en Hecho, en la esquina superior izquierda.



De vuelta al panel principal de Anaconda, pasaremos a configurar la cuenta de usuario regular del sistema, pulsando en su correspondiente opción.

**CREAR USUARIO**

Hecho

INSTALACIÓN DE FEDORA 40  
es

Nombre completo	wizz
Nombre de usuario	wizz
<input checked="" type="checkbox"/> Añadir privilegios administrativos a esta cuenta de usuario (membresía al grupo wheel)	
<input checked="" type="checkbox"/> Se requiere una contraseña para usar esta cuenta	
Contraseña	••••••••••••••••
Confirmar la contraseña	••••••••••••••••
Robusta	
<a href="#">Avanzado...</a>	

En la pantalla que nos aparece, cumplimentamos los datos de usuario como su nombre completo, el nombre de usuario Unix y su contraseña, además de elegir si le otorgaremos permisos de root mediante sudo y permitimos el autologin o no. Una vez terminado, volvemos a pulsar en Hecho.

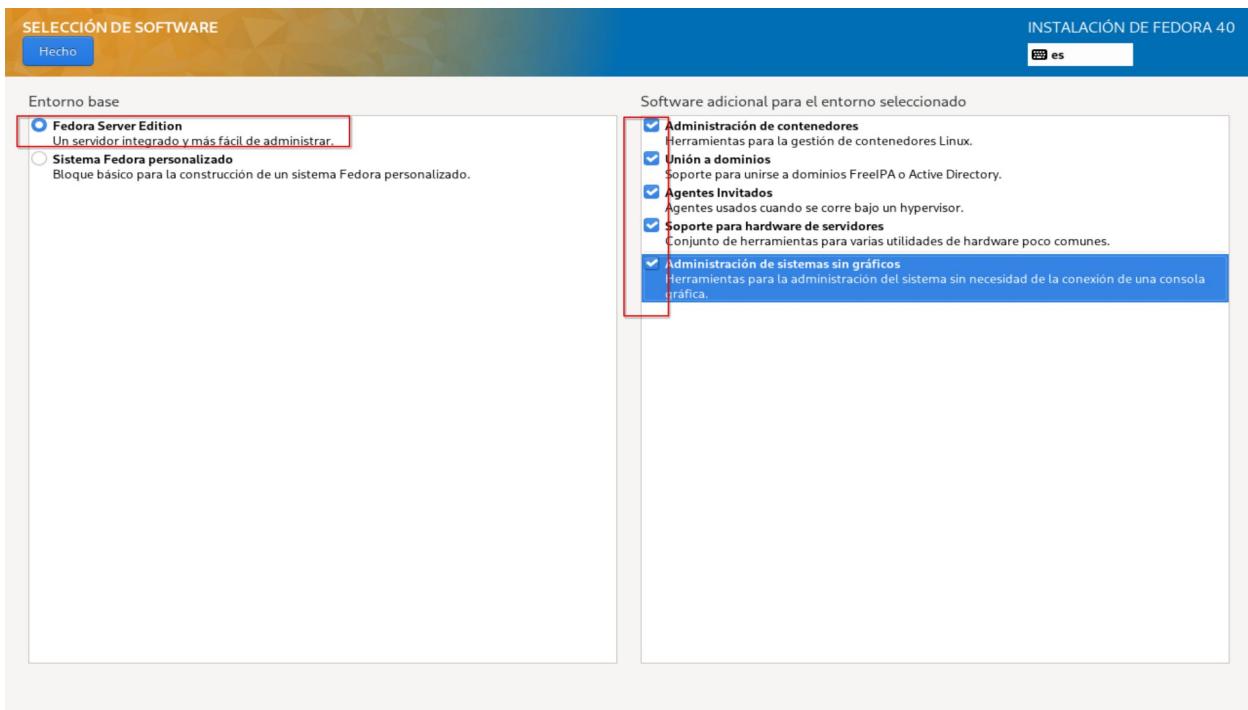
**RESUMEN DE LA INSTALACIÓN**

INSTALACIÓN DE FEDORA 40  
es

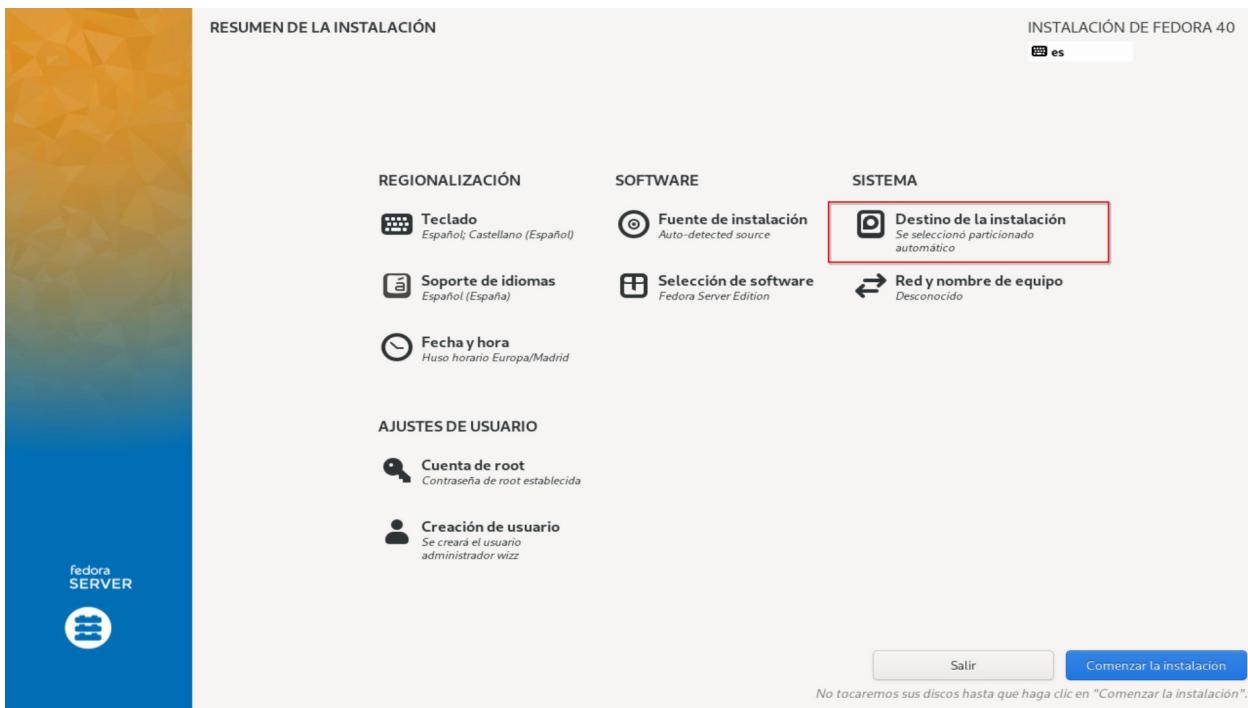
<b>REGIONALIZACIÓN</b>	<b>SOFTWARE</b>	<b>SISTEMA</b>
<ul style="list-style-type: none"> <li><b>Teclado</b> Español; Castellano (Español)</li> <li><b>Soporte de idiomas</b> Español (España)</li> <li><b>Fecha y hora</b> Huso horario Europa/Madrid</li> </ul>	<ul style="list-style-type: none"> <li><b>Fuente de instalación</b> Auto-detected source</li> <li><b>Selección de software</b> Fedora Server Edition</li> </ul>	<ul style="list-style-type: none"> <li><b>Destino de la instalación</b> Se seleccionó particionado automático</li> <li><b>Red y nombre de equipo</b> Desconocido</li> </ul>
<b>AJUSTES DE USUARIO</b>		
<ul style="list-style-type: none"> <li><b>Cuenta de root</b> Contraseña de root establecida</li> <li><b>Creación de usuario</b> Se creará el usuario administrador wizz</li> </ul>		
<input type="button" value="Salir"/> <input type="button" value="Comenzar la instalación"/> <p><i>No tocaremos sus discos hasta que haga clic en "Comenzar la instalación".</i></p>		

**⚠ Por favor complete los elementos marcados con este icono antes de continuar con el siguiente paso.**

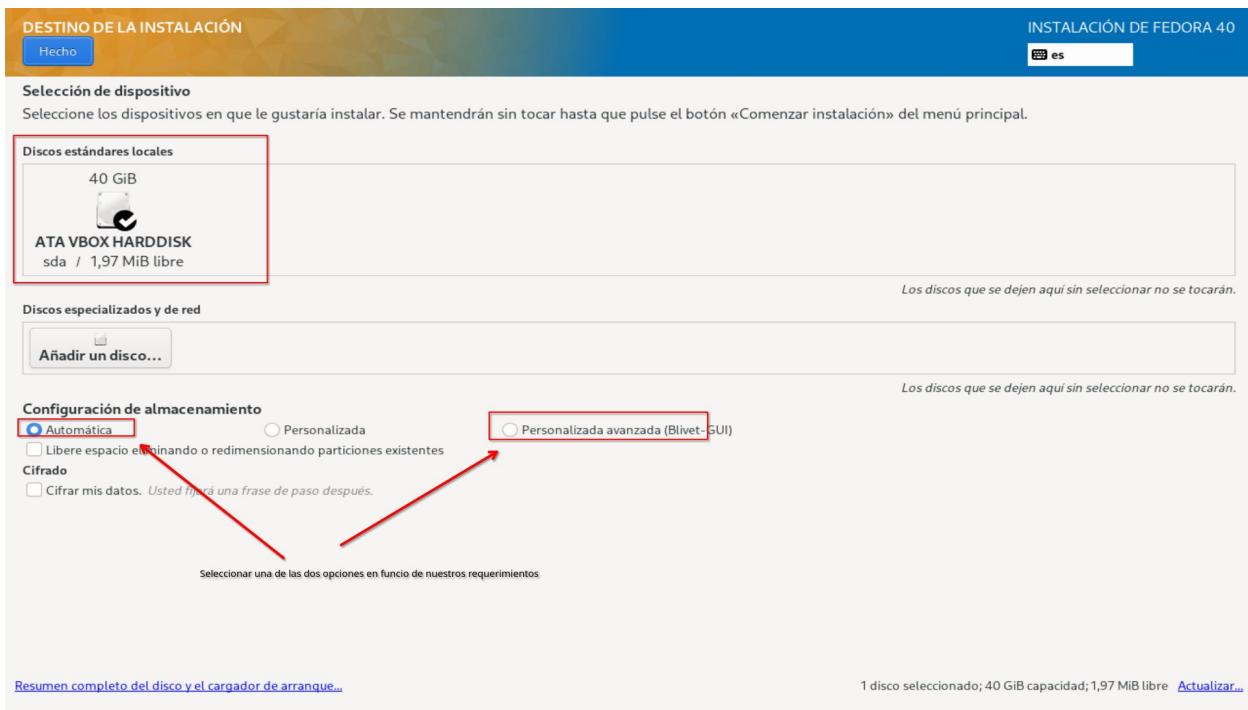
De vuelta al panel principal, pasemos a configurar la selección de software pulsando en su correspondiente opción.



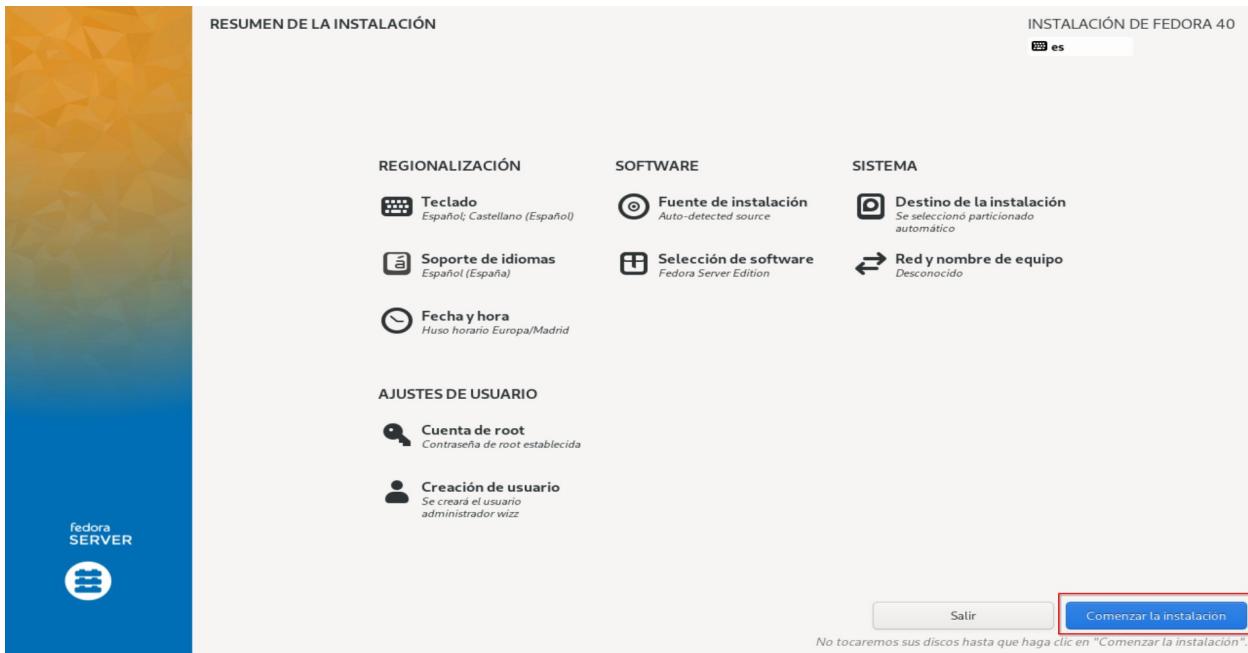
Podemos optar por una instalación personalizada o por una predeterminada en modo servidor. Puesto que vamos a hacer uso de la opción servidor, seleccionamos dicha opción, escogiendo en la columna de la derecha opciones adicionales que consideremos de utilidad para nuestro sistema. Pulsamos Hecho una vez que hayamos terminado.



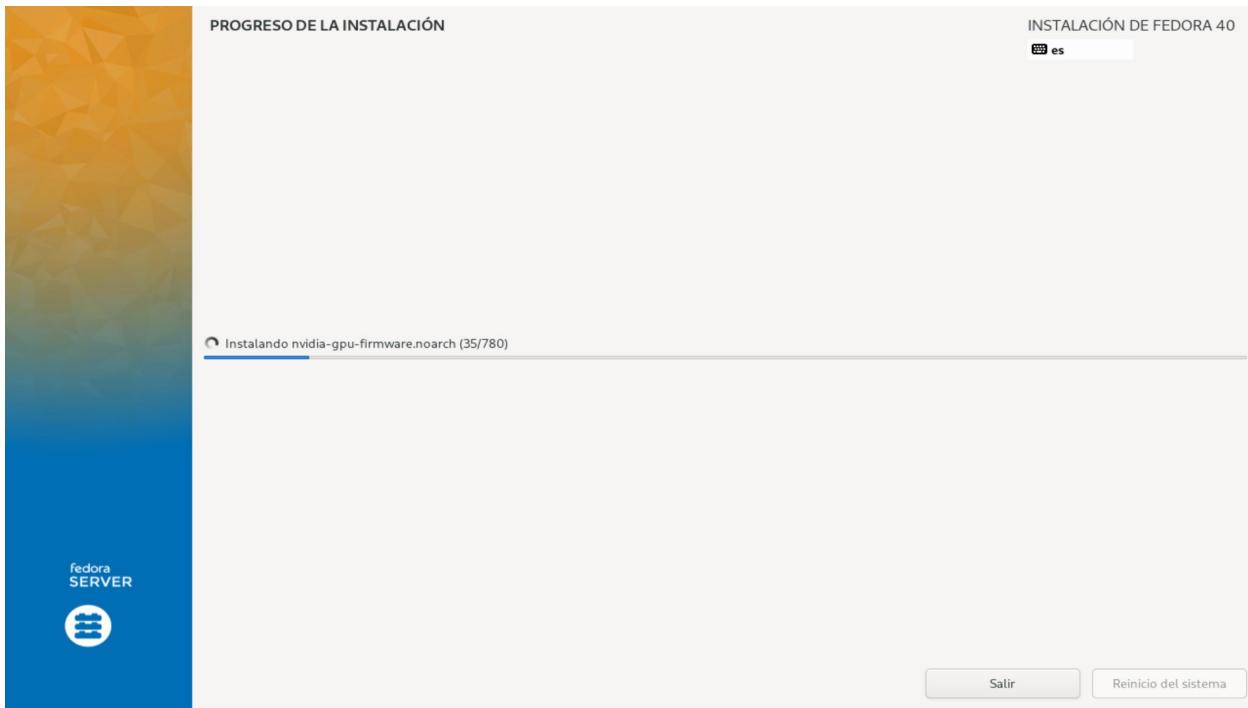
Por último, toca seleccionar el medio de almacenamiento interno en el que queremos instalar el sistema. Pulsamos sobre Destino de la instalación para proceder con dicha configuración.



En este panel se nos mostraran todos los discos (Físicos) presentes en el sistema, pudiendo seleccionar cual deseamos usar. Podremos tambien seleccionar el modo de instalación en el disco, con opciones Automática, Personalizada o Personalizada avanzada (Gestión minuciosa de las particiones). En nuestro caso, y puesto que el sistema va a albergar únicamente este sistema operativo, seleccionamos la opción automática y dejamos a Anaconda que se ocupe de todo. Una vez listo, pulsamos en Hecho.



¡Listo! Anaconda no nos requiere atención en ningún punto más para proceder. Podríamos Configurar ajustes de fecha, hora y región, pero estos han sido automáticamente detectados desde el principio, así como los parámetros del/los dispositivos de red del equipo, pero por el momento lo dejaremos en la configuración por defecto (DHCP). Cuando estemos preparados, pulsamos en Comenzar con la instalación y dejamos que Anaconda haga su trabajo.



Se nos mostrara el progreso de la instalación.



¡Listo! El instalador ha terminado y podemos reiniciar el sistema para iniciar en Fedora Server por primera vez.

```
Fedora Linux 40 (Server Edition)
Kernel 6.8.5-301.fc40.x86_64 on an x86_64 (tty3)

Web console: https://localhost:9090/ or https://192.168.68.244:9090/

localhost login: _
```

Una vez arrancado por primera vez nuestro sistema, se nos mostrara el clásico prompt de login Unix. Si nos fijamos bien, veremos que el sistema nos informa amablemente de que disponemos de una consola web, y que podemos entrar en ella a través de un navegador web con la dirección IP del servidor a través del puerto 9090. Esto es una sorpresa, puesto que, en anteriores versiones, dicha consola web, Cockpit, había que instalarla manualmente. Pasemos pues a echarle un vistazo, pues es una herramienta de administración del servidor, muy útil.

## **7.Consolas de Administración del servidor**

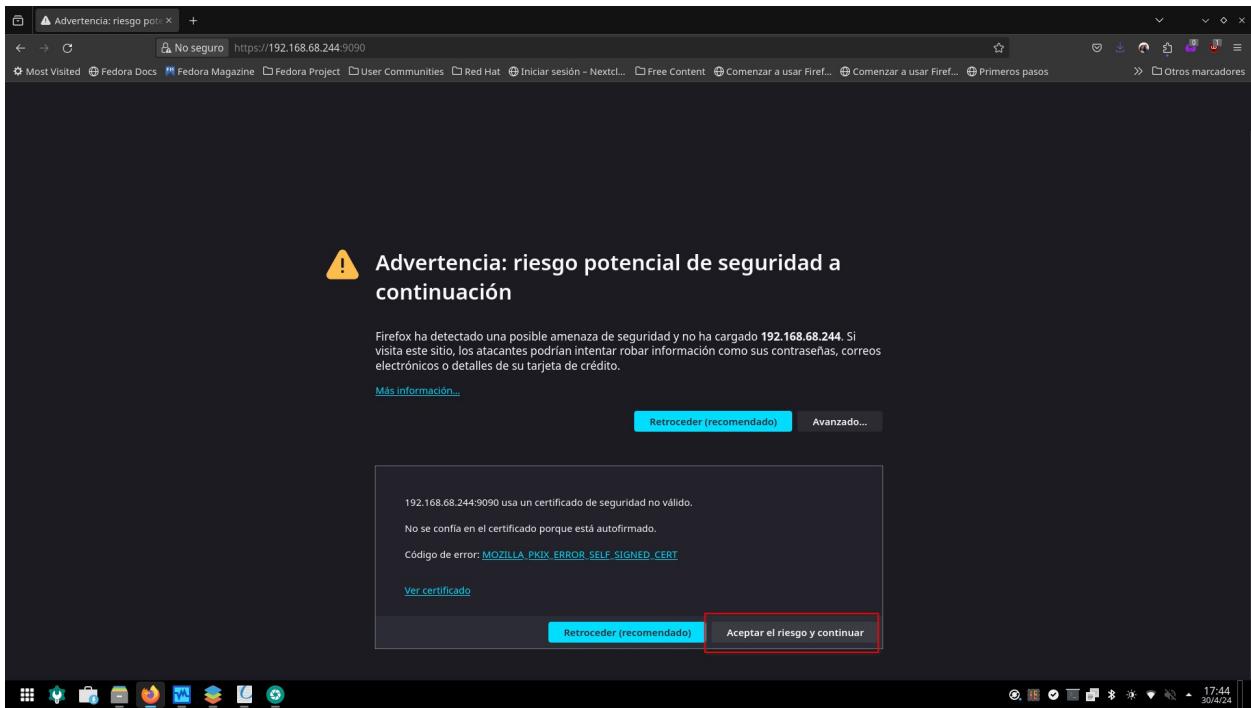
Todo servidor que se precie necesita herramientas para su administración más básica.

Si bien un administrador de sistemas experimentando puede gestionar un servidor Linux únicamente mediante consola con soluciones como ssh, lo cierto es que la tendencia natural de la administración de sistemas es utilizar consolas web de administración, pues facilitan dichas tareas y ahorran tiempo en dicha administración. No obstante, también configuraremos el servidor ssh, que servirá como ultima línea de defensa en caso de que nuestras consolas web fallen.

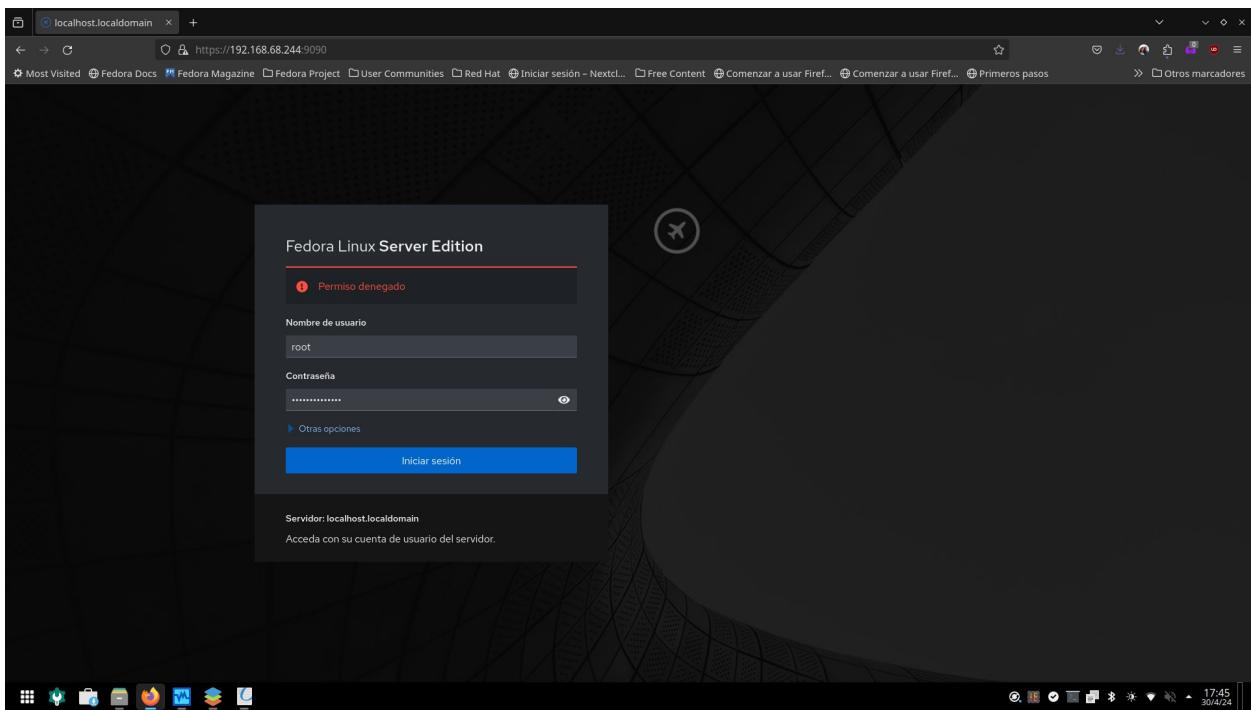
## 7.1 La consola de administración Cockpit

Puesto que la consola Cockpit ya está instalada por defecto, pasemos a comprobar el estado del servicio mediante `systemctl`.

Como vemos, está instalada, pero no está en ejecución, por lo que procedemos a arrancarla mediante **systemctl start**, y a continuación la habilitamos para el inicio automático con el sistema mediante **systemctl enable**. El sistema nos informara de que no tiene una configuración para enable, sin embargo, quedara activo el socket Cockpit, que hará que esta arranque por defecto en el próximo reinicio. Un último **systemctl status** nos muestra que Cockpit está en ejecución y listo para acceder a él.

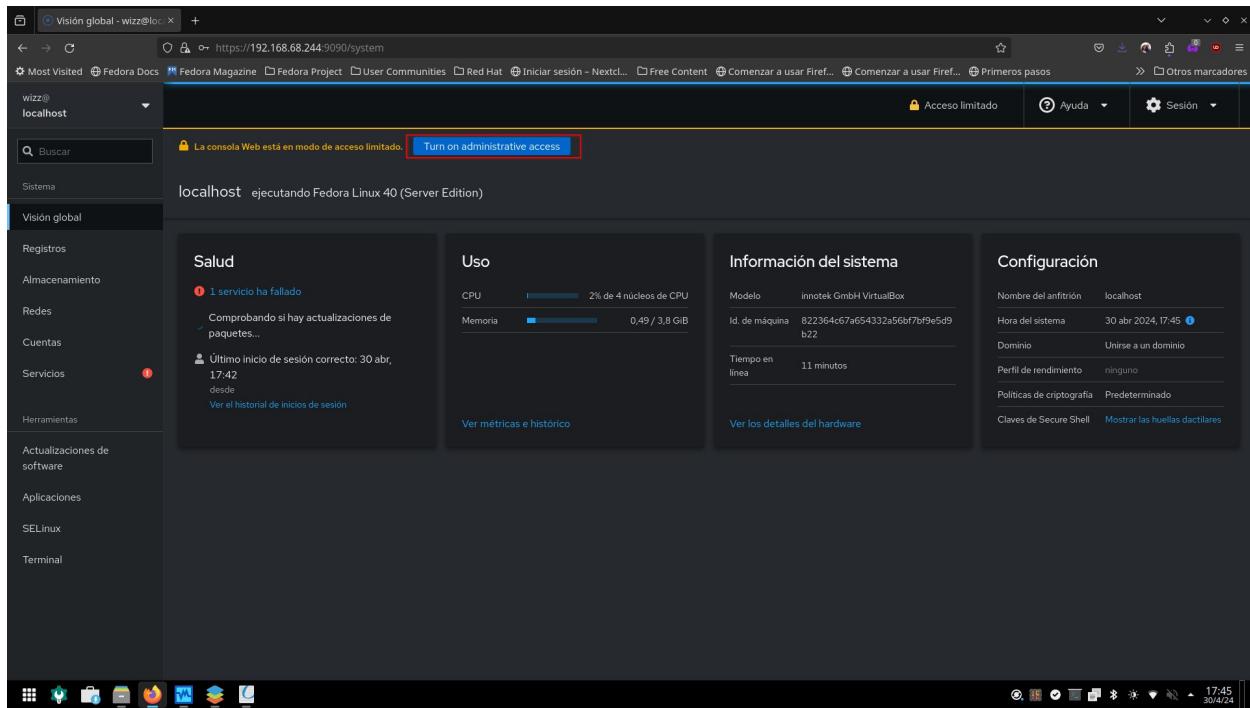


Si intentamos acceder a la misma desde una maquina cliente con acceso de red al servidor con la dirección IP al puerto 9090, recibiremos la clásica advertencia de seguridad de certificado auto firmado. Este problema se resolverá en el futuro, por el momento ignoramos el riesgo y accedemos a la misma.



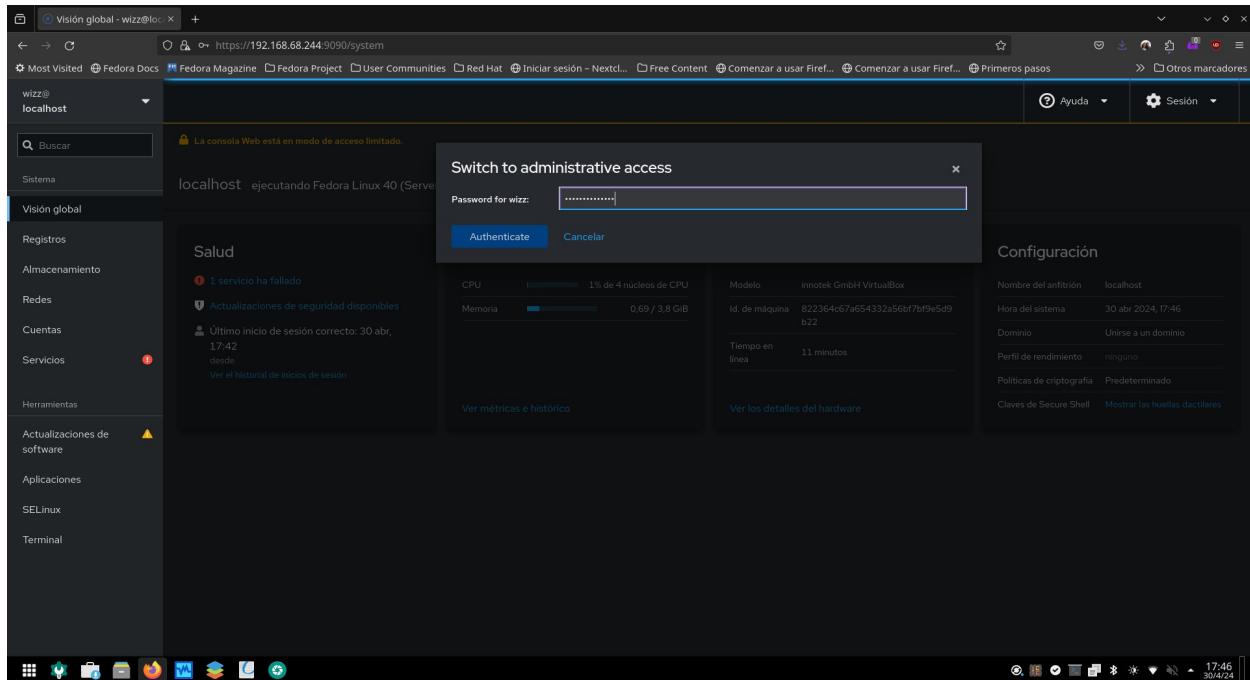
Estamos listos para loguearnos en cockpit, sin embargo, si intentamos acceder a la misma con el usuario Root del sistema, Cockpit denegara el acceso, pues por defecto y por seguridad, su configuración no permite el acceso con el usuario root.

No hay problema, nos loquearemos con el usuario regular que creamos durante la instalación.



Ya estamos logueados en cockpit. Se trata de una consola de administración web patrocinada por RedHat, que permite configurar y gestionar fácilmente diversos apartados del sistema operativo, y dispone de varios módulos instalables para ampliar su funcionalidad. No es tan potente y completa como webmin u otras opciones, pero lo compensa con una interfaz muy limpia y cuidada y un uso increíblemente fácil e intuitivo. Es una consola particularmente ideal para configurar aspectos básicos del sistema como las cuentas de usuarios, o los ajustes de red y firewall.

Para ganar acceso administrativo con el usuario regular, debemos pulsar sobre la opción resaltada en la captura superior, lo cual nos solicitará nuestra contraseña de administrador, invocando internamente un sudo.



Una vez introduzcamos la contraseña, obtendremos privilegios administrativos.

23

## Administración de sistemas informáticos en red

Ismael Carrasco Cubero



localhost ejecutando Fedora Linux 40 (Server Edition)

Salud

1 servicio ha fallado

Actualizaciones de seguridad disponibles

Último inicio de sesión correcto: 30 abr, 17:42 desde Ver el historial de inicios de sesión

Uso

CPU 0% de 4 núcleos de CPU

Memoria 0.71 / 3.8 GiB

Información del sistema

Modelo innotek GmbH VirtualBox

Id. de máquina 822964c67a654332a56bf7bf9e5d9b22

Tiempo en línea 12 minutos

Configuración

Nombre del anfitrión localhost editar

Hora del sistema 30 abr 2024, 17:47

Dominio Unirse a un dominio

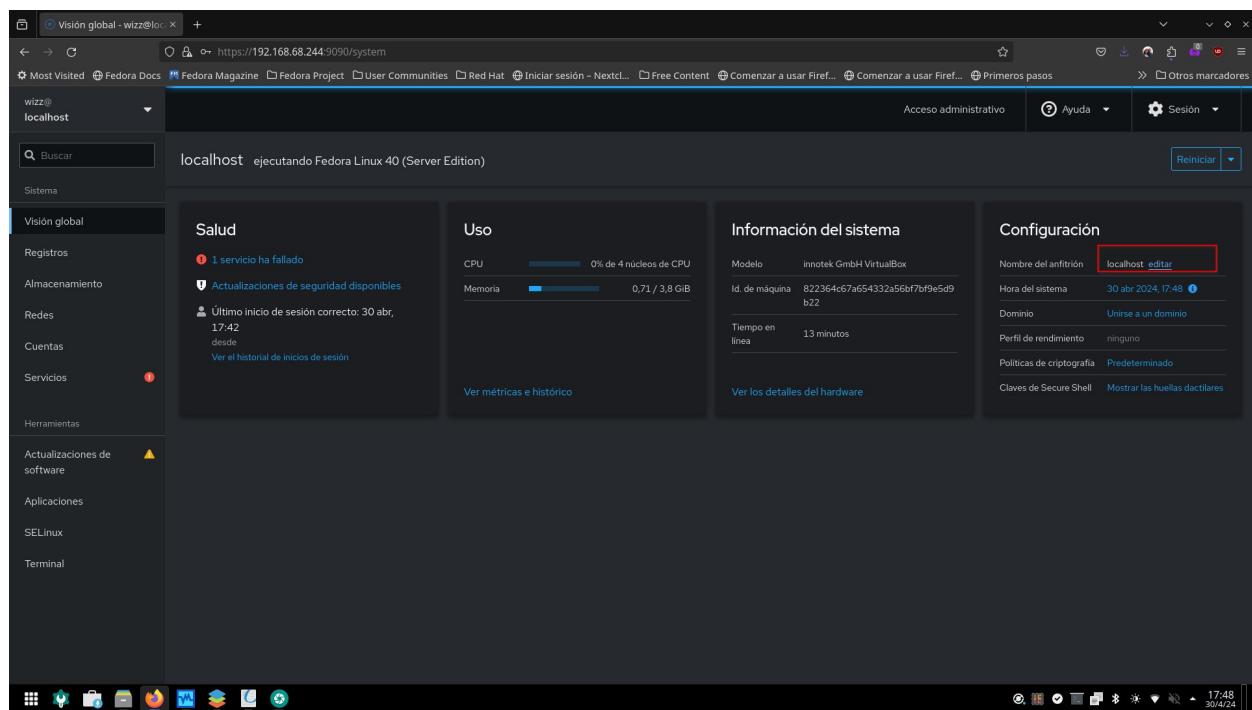
Perfil de rendimiento ninguno

Políticas de criptografía Predeterminado

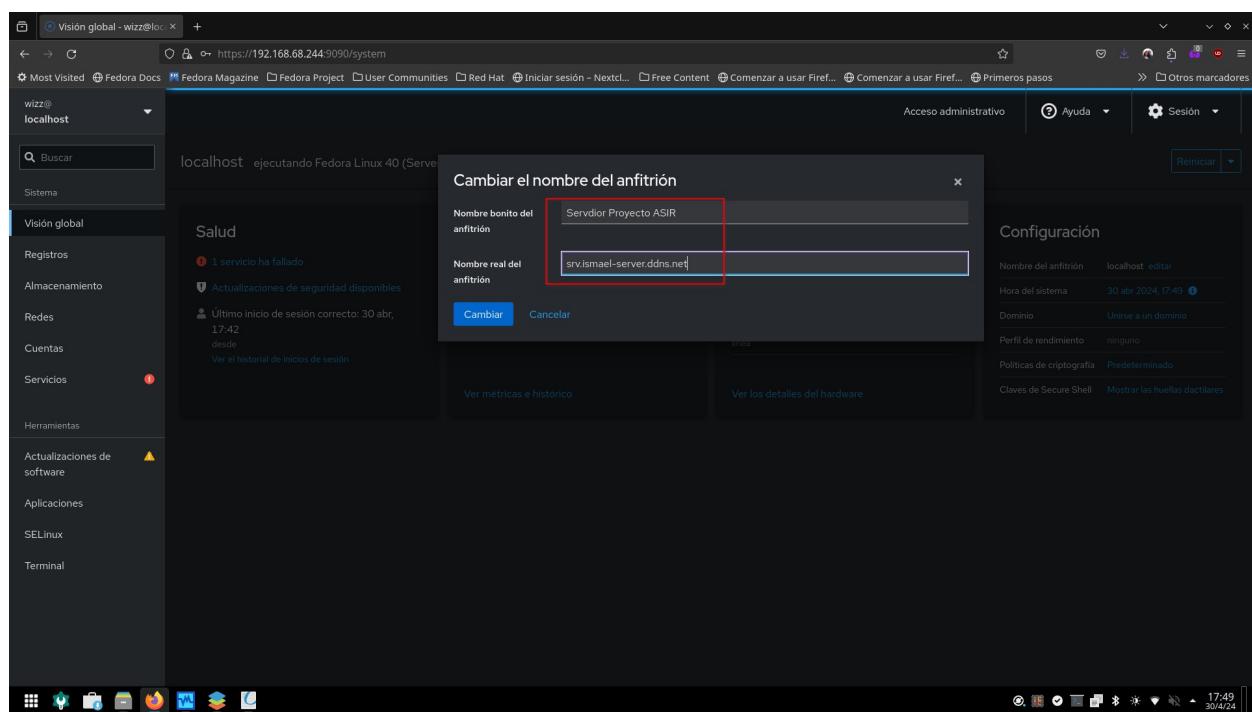
Claves de Secure Shell Mostrar las huellas dactilares

Para acceder a los diferentes apartados a gestionar, podemos seleccionar la opción deseada en el a columna izquierda, donde también pueden aparecer avisos, como el que se muestra en la captura sobre actualizaciones del sistema disponibles.

## 7.1.1 Cockpit. Configuración de nombre de Host



Comenzaremos la configuración del sistema seleccionado la opción resaltada, para darle un nombre identificativo a nuestro servidor.



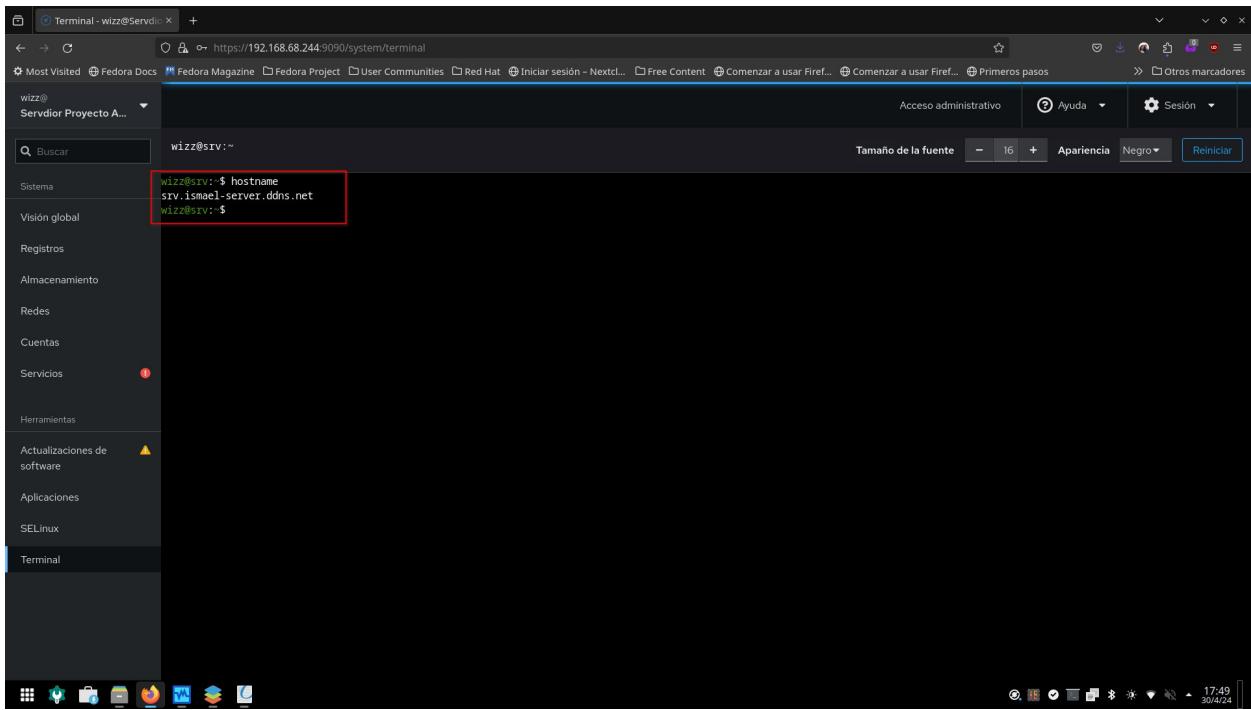
Damos un nombre descriptivo para el equipo y el FQDN que utilizaremos a partir de ahora para nuestro servidor. Dicho sea de paso, el nombre de dominio ismael-server.ddns.net, es el nombre de dominio con wildcard de NoIP que este proyecto utilizará durante toda su extensión.

25

Administración de sistemas informáticos en red

Ismael Carrasco Cubero





Tras realizar el cambio, podemos comprobar en la consola incorporada en Cockpit, que efectivamente el nombre del equipo ha cambiado a **srv.ismael-server.ddns.net**

## 7.1.2 Cockpit. Actualización del sistema

The screenshot shows the Cockpit web interface under the 'Actualizaciones de software' section. On the left sidebar, 'Actualizaciones de software' is selected. The main area shows a table of updates:

Nombre	Versi...	Seve...	Detalles
c-ares	1.28...	✓ 1	1.28.1 fixes a significant bug in 1.28.0.
curl, libcurl	8.6...	✓ 2	fix Usage of disabled protocol (CVE-2024-2004)
glibc, glibc-common, glibc-gconv-extra, glibc-langpack-es	2.39...	✓ 1	This update includes several bug fixes from the upstream glibc release branch, including the fix for a buffer overflow in iconv when converting to the ISO-2022-CN-EXT character set (CVE-2024-2961, RHBZ#2275855).
gnutls, gnutls-dane	3.8....	✓ 2	Rebase gnutls to version 3.8.5
grub2-common, grub2-pc, grub2-pc-modules, grub2-tools, ...	1.2....	✓ 2	Security fix for CVE-2023-4692
libipa_hbac, libsss_autofs, libsss_certmap, libsss_idmap, ...	2.9....	✓ 1	Fix CVE-2023-3758 <a href="https://bugzilla.redhat.com/show_bug.cgi?id=2275905">https://bugzilla.redhat.com/show_bug.cgi?id=2275905</a>
libjose	13.1...	✓ 1	Security fix for CVE-2023-50967
libnghttp2	1.59...	✓ 1	fix CONTINUATION frames DoS (CVE-2024-28182)
python3-idna	3.7....	✓ 1	Update to 3.7 (rhbz#2274439), security fix for CVE-2024-3651
unbound-libs	1.19....	✓ 1	Use the origin (DNAME) TTL for synthesized CNAMEs as per RFC 6672.
amd-gpu-firmware, amd-ucode-firmware, atheros-firmware, bremfmac-firmware, ...	202....	✗ 4	Update to upstream 20240410
b43-fwcutter	019....	⚙️	Update to the latest GIT
bind-libs, bind-license, bind-utils	32.9....	⚙️ 2	Update to 9.18.26
bluez, bluez-libs	5.75....	✗ 1	This update provides a new release of bluez which should fix several problems people encountered in 5.7.3, including connection issues, bluez potentially becoming stuck in a loop consuming 100% of CPU if you disable bluetooth with a device connected, and wheel scroll on some mice (especially Logitech ones) not working properly after a period of inactivity. If you're affected by any of the bugs, we recommend powering your system off entirely and then back on after installing the update – it's possible for bluetooth adapters to get in a problematic state.

Pasemos ahora a realizar una primera actualización del sistema, paso crucial en cualquier nueva instalación de un SO. El módulo de actualizaciones de software de Cockpit nos da información detallada acerca de las actualizaciones disponibles, incluyendo el número de paquetes a actualizar, y el tipo de actualizaciones disponibles para cada paquete. Podemos ver que tenemos numerosas actualizaciones, incluyendo múltiples actualizaciones importantes de seguridad. Procedemos pues a instalarlas todas pulsando en el botón correspondiente.

The screenshot shows the Cockpit web interface during a system update. The progress bar at the top indicates 'Actualizando nss 3.99.0-1.fc40 (x86\_64)' is at 18%. Below the progress bar, there are two buttons: 'Cancelar' (Cancel) and 'Reiniciar al terminar' (Restart after finish). The main content area shows a list of packages being updated:

Actualizado	Paquete
Actualizado	util-linux-core 2.40-13.fc40 (x86_64)
Actualizado	util-linux 2.40-13.fc40 (x86_64)
Actualizado	unbound-libs 1.19.3-1.fc40 (x86_64)
Actualizado	tzdata 2024a-5.fc40 (noarch)
Actualizado	tiwlink-firmware 20240410-1.fc40 (noarch)
Actualizado	teamd 1.32-7.fc40 (x86_64)
Actualizado	sssd-tools 2.9.4-7.fc40 (x86_64)
Actualizado	sssd-proxy 2.9.4-7.fc40 (x86_64)
Actualizado	sssd-nfs-idmap 2.9.4-7.fc40 (x86_64)

Se nos muestra el progreso de la actualización, y tenemos la posibilidad de cancelar la misma.

wizz@ Servidor Proyecto A...

Buscar

Sistema

Visión global

Registros

Almacenamiento

Redes

Cuentas

Servicios

Herramientas

**Actualizaciones de software**

Aplicaciones

SELinux

Terminal

Estado

El sistema está actualizado  
Última comprobación: hace 5 minutos

Ajustes

Actualizaciones automáticas Sin configurar

Habilitar

**Historial de actualización**

30 abr 2024, 17:52 196 paquetes

amd-gpu-firmware	amd-ucode-firmware	atheros-firmware	b43-f
bind-lbs	bind-license	bind-utils	cirrus-audio-firmware 20240312-1.fc40
bluez-lbs	brcmfmac-firmware	c-ares	bluez
cockpit	cockpit-bridge	cockpit-networkmanager	cirrus-audio-firmware
cockpit-selinux	cockpit-storaged	cockpit-system	cockpit-packagekit
cryptsetup	cryptsetup-libs	cups-lbs	cockpit-ws
dbus-broker	default-fonts-core-sans	dnf	curl
dnf-plugins-core	dnsmasq-langpack	dracut	dnf-data
dracut-network	dracut-squash	ed	dracut-config-rescue
expat	fedora-release-common	fedora-release-identity-server	emacs-fsysten
firewalld	firewalld-filesystem	fwupd	fedora-release-server
glibc-common	glibc-gconv-extra	glibc-langpack-es	glibc
gnutls-dane	gobject-introspection	grub2-common	gnutls
grub2-pc-modules	grub2-tools	grub2-minimal	grub2-pc
initscripts-service	intel-audio-firmware	intel-gpu-firmware	hwdata
iwlwifi-dvm-firmware	iwlwifi-mvm-firmware	kernel	iwlegacy-firmware
kernel-modules	kernel-modules-core	langpacks-core-es	kernel-core
langpacks-fonts-es	libblkid	libblockdev	langpacks-es
libblockdev-fs	libblockdev-loop	libblockdev-lvm	libblockdev-crypto
libblockdev-nvme	libblockdev-part	libblockdev-swap	libblockdev-mdraid
libcap	libcurl	libdnf	libblockdev-utils
libfdisk	libgpg-error	libibus	libertas-firmware
libjose	libmount	libnetapi	libipa_hbac

¡Sistema actualizado!

## 7.1.3 Cockpit. Configuración de IP estática del servidor

The screenshot shows the Cockpit web interface for system management. The left sidebar is a navigation menu with items like 'Sistema', 'Visión global', 'Registros', 'Almacenamiento', 'Redes' (which is highlighted with a red box), 'Cuentas', 'Servicios', 'Herramientas', 'Actualizaciones de software', 'Aplicaciones', 'SELinux', and 'Terminal'. The main content area has two graphs at the top: 'Kbps Transmitiendo' and 'Kbps Recibiendo', both showing minimal activity over time. Below the graphs is a section titled 'Cortafuegos' (Firewall) with a status of 'Habilitado' (Enabled) and '1 zona activa' (1 active zone). There are buttons to 'Editar reglas y zonas' (Edit rules and zones) and 'Añadir VPN', 'Añadir agregación', 'Añadir equipo', 'Añadir puente', and 'Añadir VLAN'. A table follows, showing network interfaces: 'Nombre' (Name), 'Dirección IP' (IP Address), 'Enviendo' (Transmitting), and 'Recibiendo' (Receiving). The row for 'enp0s3' is selected and highlighted with a red box. At the bottom, there's a 'Registros de redes' (Network logs) section with a list of log entries from April 30, 2024, and a 'Ver todos los registros' (View all logs) button.

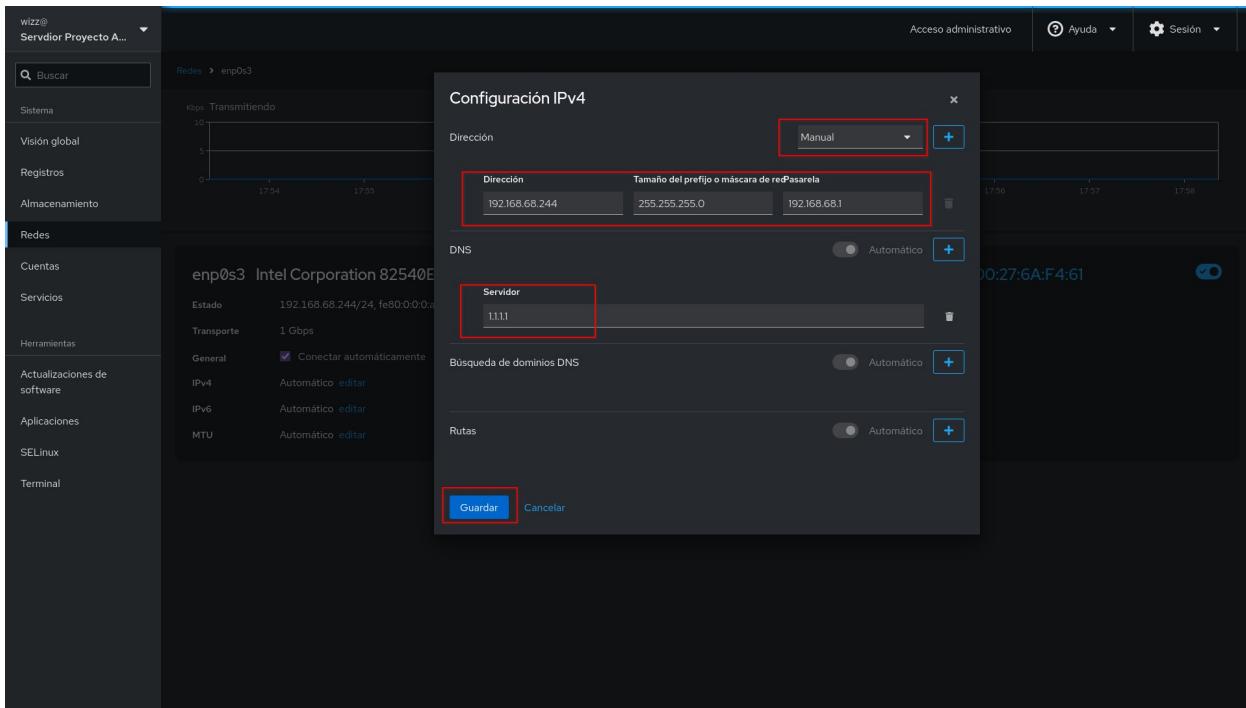
Procedemos a usar Cockpit para configurar los parámetros de red del servidor, pues todo servidor que se precie debe tener una dirección IP estática, y la configuración DHCP por defecto no nos sirve. Nos dirigiremos para ello al apartado de redes en la columna Izquierda.

Se nos mostrarán varios apartados a configurar, incluyendo los adaptadores presentes en el equipo, así como el firewall, gestión de VPN, bridges, VLANs etc.

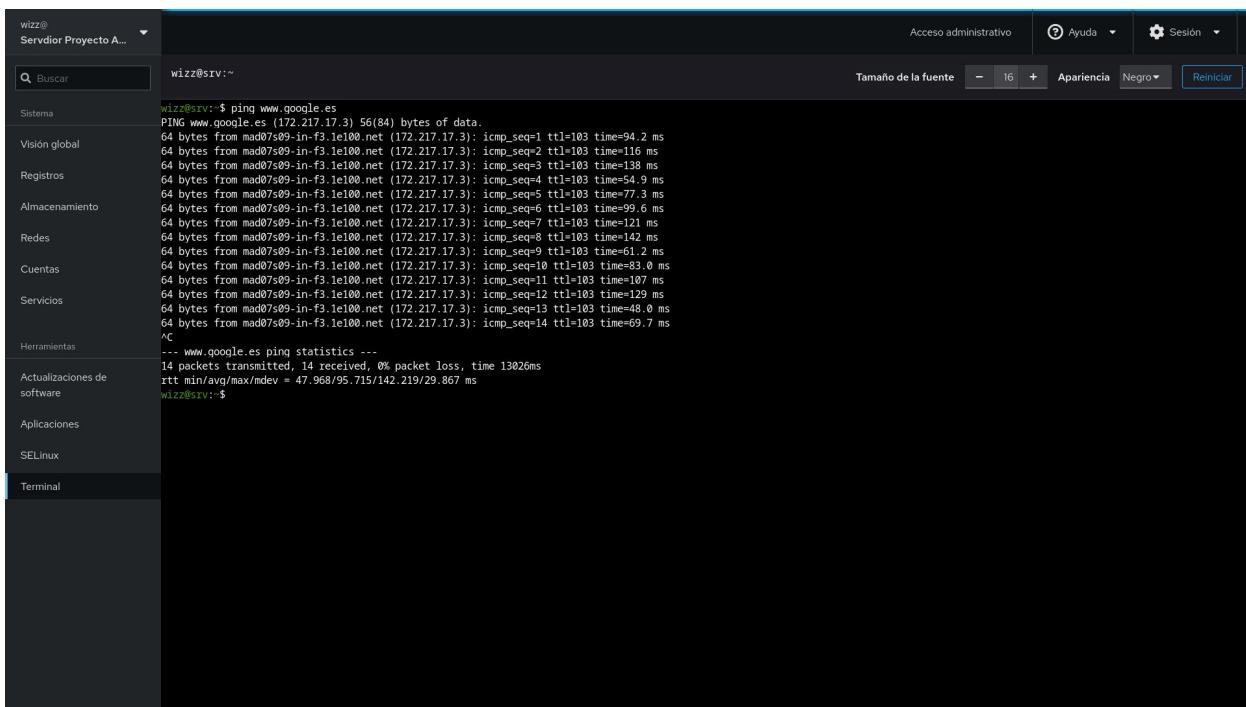
Lo que nos interesa es acceder al dispositivo de red del equipo pulsando en su nombre en la lista de adaptadores.

This screenshot shows the detailed configuration for the 'enp0s3' network adapter. The left sidebar is identical to the previous one. The main area now focuses on 'Redes > enp0s3'. It displays a summary of the adapter: 'enp0s3 Intel Corporation 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter) e1000 08:00:27:6A:F4:61'. Below this, it shows the 'Estado' (State) as '192.168.68.244/24, fe80::0:a00:27ff:fe6af:461/64', 'Transporte' (Transport) as '1 Gbps', and a checked checkbox for 'Conectar automáticamente' (Connect automatically). Under the 'General' tab, 'IPv4' is set to 'Automático' (Automatic) with an 'editar' (edit) button. The 'IPv6' and 'MTU' settings are also shown. The 'IPv4' setting is highlighted with a red box.

Una vez dentro de las opciones del adaptador, podemos ver gráficas sobre su estado, y podemos gestionarlo. Pulsaremos sobre editar en el apartado IPV4.



Como vemos, el proceso es increíblemente sencillo. Escogemos el tipo de configuración Manual en el desplegable e introducimos los parámetros típicos de red IPv4 como la IP, el prefijo de red, servidores DNS etc y pulsamos sobre Guardar. Los cambios quedarán guardados inmediatamente.



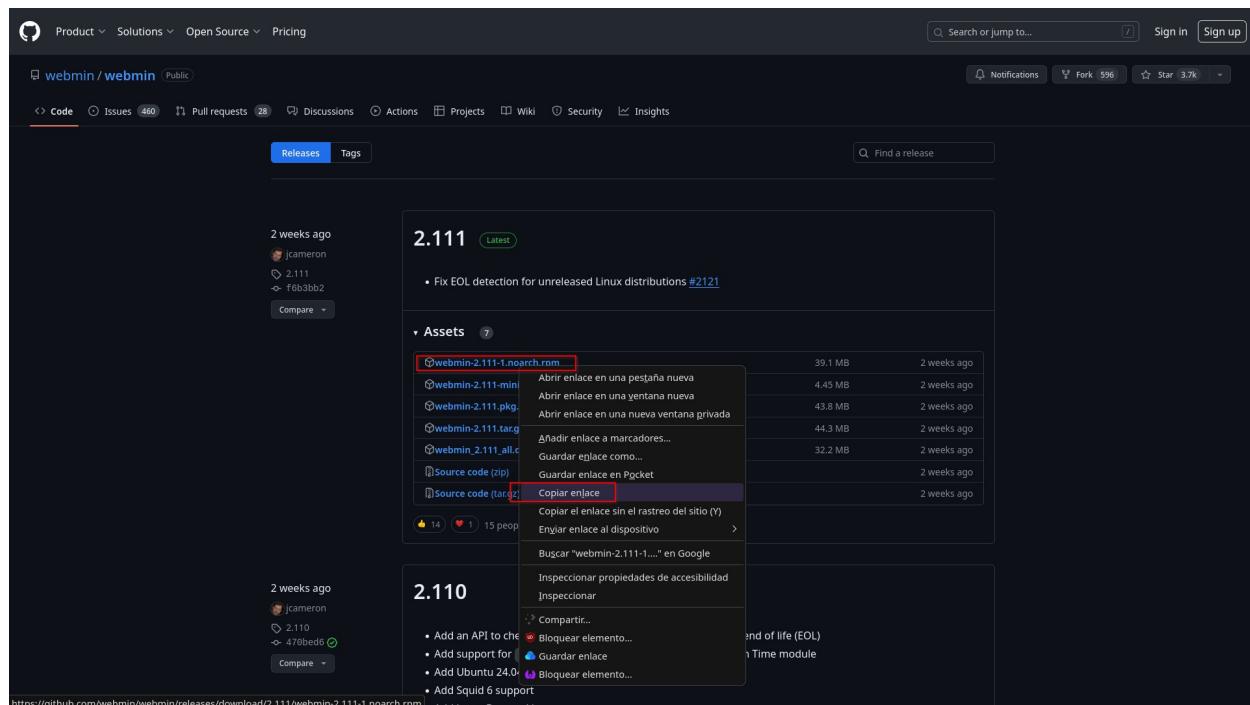
Si probamos a lanzar un ping a Google para comprobar la conectividad del sistema, vemos que la IP estática ha quedado configurada correctamente y el sistema dispone de salida a Internet.

## 7.2 La consola de administración Webmin

La consola de administración Webmin, es una vieja conocida en el mundillo de la administración de sistemas, incluyendo este ciclo formativo de ASIR, puesto que se ha usado extensamente para la gestión de los diversos servicios tratados en el temario, así como ajustes del sistema base.

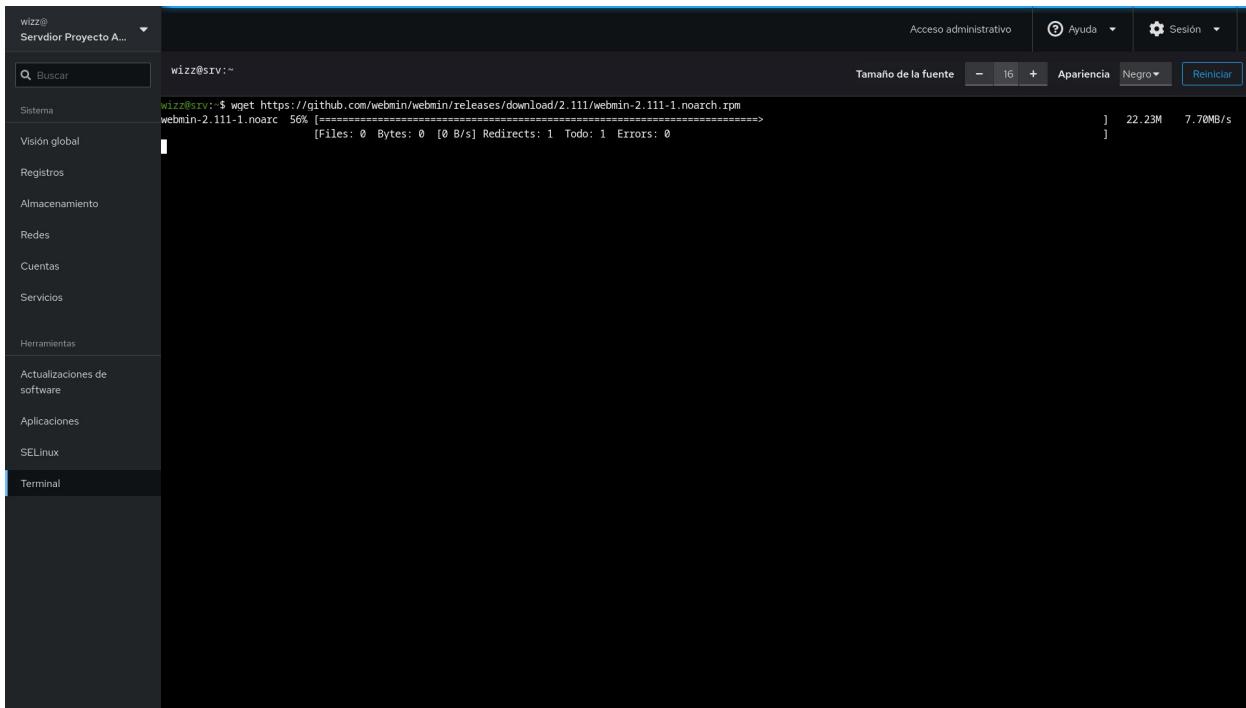
Si bien su aspecto es mucho más espartano, e incluso algo anticuado comparada con cockpit, Webmin es una herramienta increíblemente poderosa, que compensa dicho aspecto desfasado con una gestión increíblemente detallada del sistema operativo e incluso servicios individuales del sistema como DHCP, DNS o SAMBA. Es por tanto lógico que un servidor disponga de webmin en su repertorio de herramientas administrativas.

### 7.2.1 Webmin. Instalación

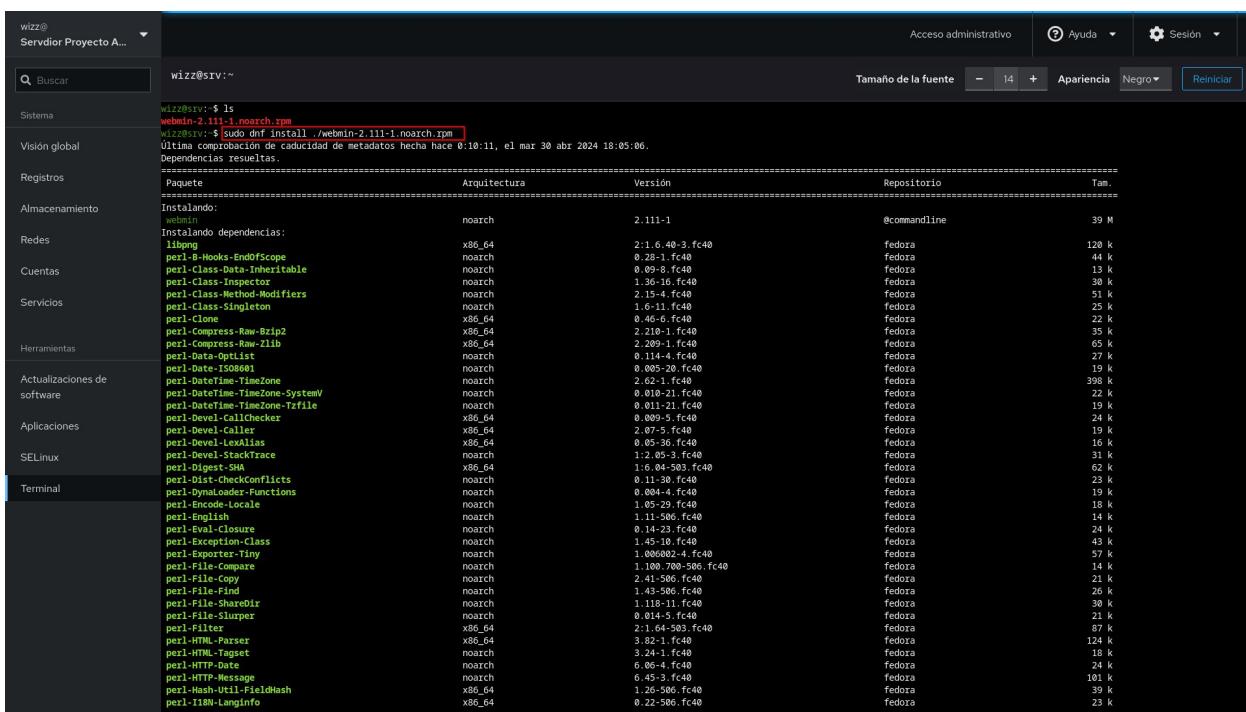


El primer paso para comenzar nuestro despliegue de Webmin, es obtenerlo de alguna fuente confiable. Webmin no está disponible en los repositorios de fedora, pero eso no supone un problema; El proyecto dispone de su repositorio de GitHub del que obtener las releases.

Para comenzar con la instalación obtendremos la última release disponible en RPM copiando su enlace de descarga



De vuelta a nuestro servidor, usaremos la herramienta de consola de Cockpit para descargar la release de Webmin, usando **wget** con el enlace que hemos copiado del GitHub oficial del mismo.



Una vez descargado dicho paquete RPM usaremos el gestor de paquetes dnf propio de Fedora para instalar webmin con **sudo dnf install ./ruta/al/paquete**; dnf se encargará de descargar e instalar cualquier dependencia que webmin necesite. Para consolidar la instalación, respondemos "si" para confirmar la transacción.

The screenshot shows the Cockpit Services page. On the left is a sidebar with navigation links: Buscar, Sistema, Visión global, Registros, Almacenamiento, Redes, Cuentas, Servicios (which is selected and highlighted in blue), Herramientas, Actualizaciones de software, Aplicaciones, SELinux, and Terminal. The main content area has tabs for Servicios, Objetivos, Sockets, Temporizadores, and Rutas. A search bar at the top allows filtering by name or description, state of activity, and state of unit file. Below is a table of services:

Nombre	Descripción	Estado	Opciones
abrt-journal-core	ABRT coredumpctl message creator	Ejecutando	Habilitado
abrt-oops	ABRT kernel log watcher	Ejecutando	Habilitado
abrt-pstoreoops		No está ejecutándose	Deshabilitado
abrt-vmcore	ABRT kernel panic detection	No está ejecutándose	Habilitado
abrt-xorg	ABRT Xorg log watcher	Ejecutando	Habilitado
abrtd	ABRT Daemon	Ejecutando	Habilitado
arp-ethers		No está ejecutándose	Deshabilitado
atd	Deferred execution scheduler	Ejecutando	Habilitado
audit-rules	Load Audit Rules	No está ejecutándose	Habilitado
audited	Security Audit Logging Service	Ejecutando	Habilitado
auth-rpcgss-module	Kernel Module supporting RPCSEC_GSS	No está ejecutándose	Estático
autofs	Automounts filesystems on demand	No está ejecutándose	Deshabilitado

El siguiente paso es comprobar si webmin está en ejecución. Usaremos la herramienta de servicios incorporada en Cockpit, filtrando el nombre del servicio en su cuadro de búsqueda.

The screenshot shows the Cockpit Services page with a search filter applied. The search bar contains 'webmin'. The results table shows one service:

Nombre	Descripción	Estado	Opciones
webmin	Webmin server daemon	Ejecutando	Habilitado

Cockpit filtrara los servicios si encuentra alguna coincidencia. Simplemente pulsamos sobre el servicio que nos interesa, en este caso Webmin.

Estado: Ejecutando (Activo desde 30 abr 2024, 18:17)  
 Ruta: /usr/lib/systemd/system/webmin.service  
 Memoria: 29,6 MB

Requiere: system.slice, sysinit.target  
 Quiere: network-online.target  
 Buscado por: multi-user.target  
 Conflictos: shutdown.target  
 Antes: shutdown.target, multi-user.target  
 Después: sysinit.target, system.slice, network-online.target, basic.target, network.target, systemd-journal.socket

Bitácoras del servicio

30 de abril de 2024

18:17 Started webmin.service - Webmin server daemon.  
 18:17 Starting webmin.service - Webmin server daemon...

Como vemos, no tenemos que hacer nada, Webmin ha quedado habilitado al arranque y en ejecución, directamente tras la instalación.

Kbps Transmitiendo

Kbps Recibiendo

Cortafuegos: Habilitado

1 zona activa

Editar reglas y zonas

Interfaces

Nombre	Dirección IP	Enviendo	Recibiendo
enp0s3	192.168.68.244/24		

Registros de redes

30 de abril de 2024

```

17:58 <info> [1714492737.3317] policy: set 'enp0s3' (enp0s3) as default for IPv6 routing and DNS
17:58 <info> [1714492737.3311] dhcpc6 (enp0s3): activation: beginning transaction (timeout in 45 seconds)
17:58 <info> [1714492736.2762] audit: op = "checkpoint-destroy" arg = "/org/freedesktop/NetworkManager/Checkpoint/1" pid=12396 uid=0 result="success"
17:58 <info> [1714492736.2760] checkpoint[0x562a1ba26c60]: destroy /org/freedesktop/NetworkManager/Checkpoint
17:58 <info> [1714492735.4433] manager: NetworkManager state is now CONNECTED_GLOBAL
17:58 <info> [1714492735.4430] device (enp0s3): activation: successful, device activated.
17:58 <info> [1714492735.4429] manager: NetworkManager state is now CONNECTED_SITE
17:58 <info> [1714492735.4427] device (enp0s3): state change: secondaries -> activated (reason 'none', sys-iface-state: 'managed')
17:58 <info> [1714492735.4425] device (enp0s5): state change: ip-check -> secondaries (reason 'none', sys-iface-state: 'managed')

```

Sin embargo, previo a su uso, debemos hacer un ajuste adicional. Por defecto, Webmin escucha en el puerto 10000, puerto que por defecto, está cerrado en el firewall de Fedora. Este puerto quedará cerrado en el resultado final del proyecto, pues organizaremos los accesos al servidor de una forma distinta, pero por el momento, lo abriremos para poder acceder a webmin.

En la herramienta de firewall, pulsamos sobre la opción para Editar reglas y zonas, resaltada en rojo en la captura superior.

Zona Publica    Interfaz enp0s3    Direcciones permitidas En toda la subred

Servicio	TCP	UDP
ssh	22	
dhcpcv6-client		546
cockpit	9090	

Añadir servicios

Por defecto, ya tenemos creada la zona publica, y podemos observar que están abiertos los puertos para ssh, DHCP V6 y Cockpit. Pulsaremos sobre añadir servicio para añadir Webmin a la lista de conexiones permitidas.

Añadir puertos a la zona FedoraServer

Servicios  Puertos específicos

TCP  Los puertos, rangos y/o servicios deben estar separados por comas para que sean válidos

UDP  Los puertos, rangos y/o servicios deben estar separados por comas para que sean válidos

ID  Si se deja vacío, se generará un ID basado en los servicios y números de puerto asociados

Descripción

Añadir puertos Cancelar

Debemos seleccionar la opción de puertos específicos, puesto que Webmin, no está disponible en la lista de servicios predeterminados. Añadimos el puerto TCP 10000, y damos un nombre y descripción a la regla, pulsando en Añadir puerto cuando hayamos terminado.

WIZZ@ Servidor Proyecto A...

Redes > Cortafuegos

Cortafuegos  Habilitado Las conexiones entrantes se bloquean por defecto. Las salientes no se bloquean.

Añadir una nueva zona

Zona Publica Interfaz enp0s3 Direcciones permitidas En toda la subred

Servicio	TCP	UDP
ssh	22	
dhcpcv6-client		546
cockpit	9090	
webmin	10000	

Puerto de acceso a la consola de administración webmin

Listo, ahora el firewall permite el acceso a Webmin.

⚠ Advertencia: riesgo potencial de seguridad a continuación

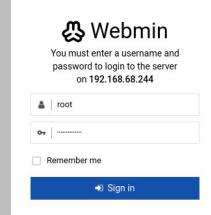
Más información...

Retroceder (recomendado) Avanzado...

192.168.68.244:10000 usa un certificado de seguridad no válido.  
No se confía en el certificado porque está autofirmado.  
Código de error: MOZILLA\_PKIX\_ERROR\_SELF\_SIGNED\_CERT  
Ver certificado

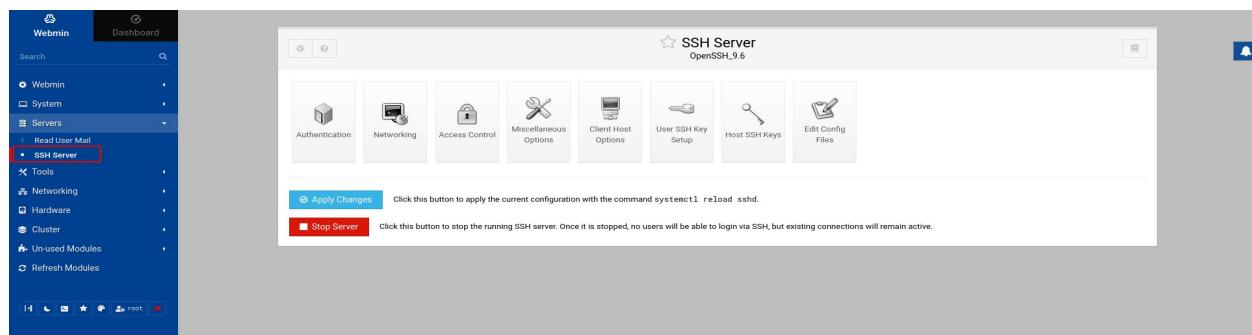
Retroceder (recomendado) Aceptar el riesgo y continuar

Una vez más, tenemos el mismo aviso de seguridad, referente al certificado auto firmado. Al igual que Cockpit, este problema lo resolveremos más adelante. Por el momento, ignoramos el aviso de seguridad y entramos.

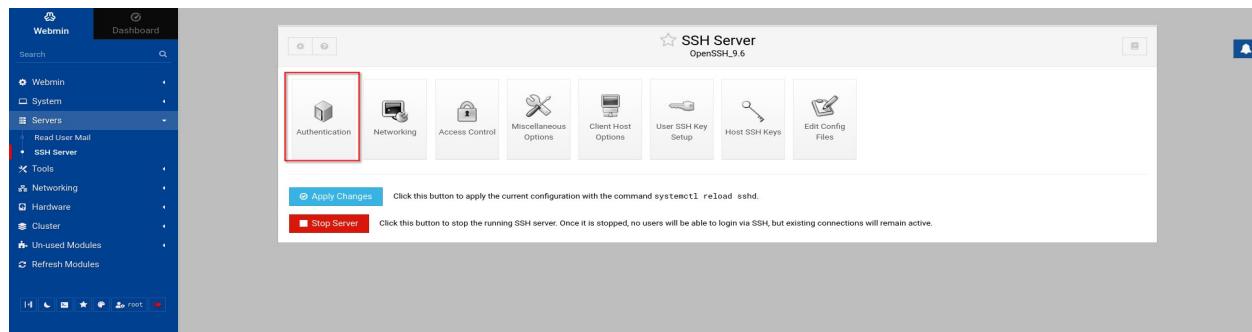


Clásica pantalla de login de Webmin. A diferencia de Cockpit, este si permite por defecto el login como usuario root, así que procedemos a entrar, lo usaremos para configurar nuestra última herramienta de administración del sistema, ssh.

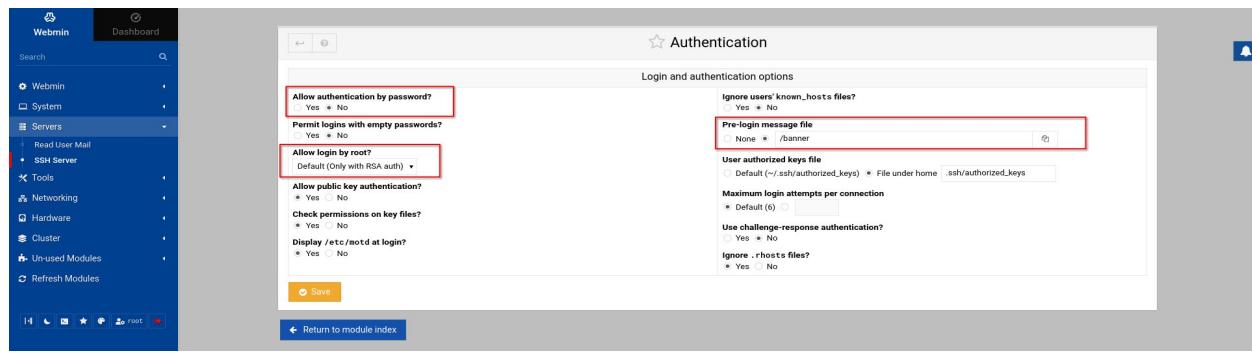
## 7.2.2 Webmin. Configuración del servicio SSH



Estoy seguro de que Webmin no necesita demasiada presentación ni justificación de su uso. Se trata de una consola gráfica web increíblemente potente, con una interfaz bastante anticuada, pero eso no invalida el hecho de que es una de las mejores herramientas de administración de sistemas Linux existentes, con infinidad de módulos para configurar el sistema en sí mismo, como servicios específicos desplegados en el servidor (Servicios clásicos, basados en paquetería standard de Linux). Precisamente haremos uso de esta cualidad para configurar nuestra última vía de gestión del servidor, el indispensable SSH. Para comenzar, seleccionamos el servidor ssh, en el apartado de servidores de la columna izquierda. Esto nos llevará al panel de gestión del servicio.



En dicho panel, seleccionaremos a continuación la opción de autenticación, para configurar dichos parámetros.



Para dejar nuestro servidor SSH seguro, y evitar a los bots, que día y noche intentanlogueos con ataques de fuerza bruta, la mejor defensa posible es deshabilitar por completo el login por contraseña. De esta forma solo equipos con claves Públicas autorizadas expresamente podrán entrar, quedando denegado cualquier intento de login restante. Permitiremos también el login de root exclusivamente mediante clave pública, y ya que estamos, seleccionaremos un archivo de banner “molón” como bienvenida pre login al servidor. Para terminar, pulsamos sobre el botón de guardar cambios, y habremos terminado.

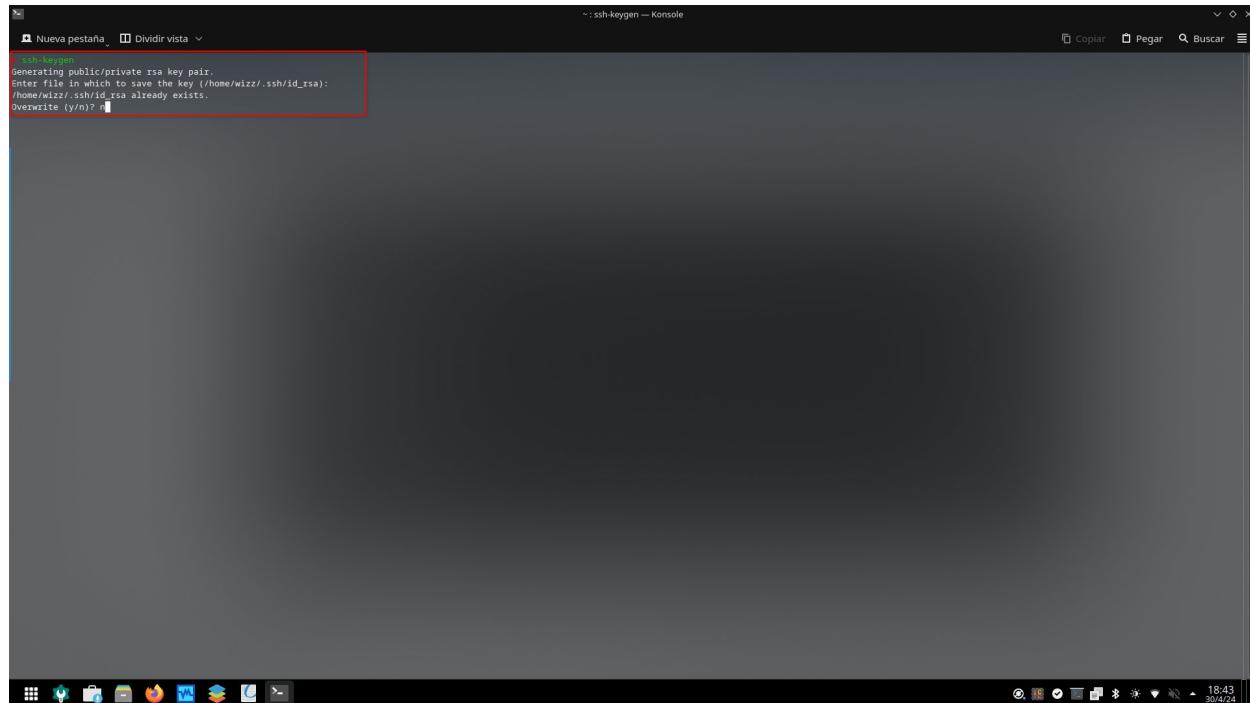
```
SSH Nueva pestaña Dividir vista ~: zsh — Konsole
SSH 192.168.68.244
the authenticity of host '192.168.68.244' (192.168.68.244) can't be established.
ED25519 key fingerprint is SHA256:oi61H0hsb8F5+Byic8pV6QWtDEN49+57KRCfwvEjISM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? [
```

Vamos a realizar una prueba de seguridad. Desde la consola de un equipo cliente, intentaremos entrar desde la red local al servidor por ssh con el clásico **ssh usuario@host**. Al ser una primera conexión, el cliente nos pregunta si confiamos en la huella criptográfica proporcionada por el servidor, aceptamos dicha huella.

Como vemos, el servidor funciona correctamente y está a la escucha, además muestra nuestro banner de bienvenida. Sin embargo, como se aprecia en la captura, prohíbe la conexión a pesar de conocer la contraseña y el usuario del servidor, por no conocer el servidor la clave pública del equipo. Este es exactamente el comportamiento que esperamos, sin una clave publica autorizada, nadie podrá acceder al equipo por SSH.

## 7.3 Consola remota SSH. Autorización del cliente con clave pública

Como vemos, nuestro servidor SSH ha ganado en seguridad enormemente tan solo con ese cambio tan simple. Sin embargo, en estos momentos nos podemos acceder a él de ninguna manera. Necesitamos por tanto generar una clave pública en nuestro equipo cliente, y autorizar dicha clave en nuestro servidor.



El primer paso es generar dicha clave, simplemente ejecutando **ssh-keygen** para generar la pareja pública-privada de claves. Nos pedirá la ruta en la que generar el id\_rsa, que en este caso dejamos por defecto. Se puede apreciar que el sistema me pregunta si deseo sobrescribir la clave que ya hay presente. Esto se debe a que ya dispongo de la clave pública autorizada en el servidor (recordemos que este servidor es una simulación del real con una máquina virtual). En mi caso, no sobrescribiré la clave y usare la ya existente, pero en el caso de la creación de una nueva, no habrá problema y esta se generará directamente.

```

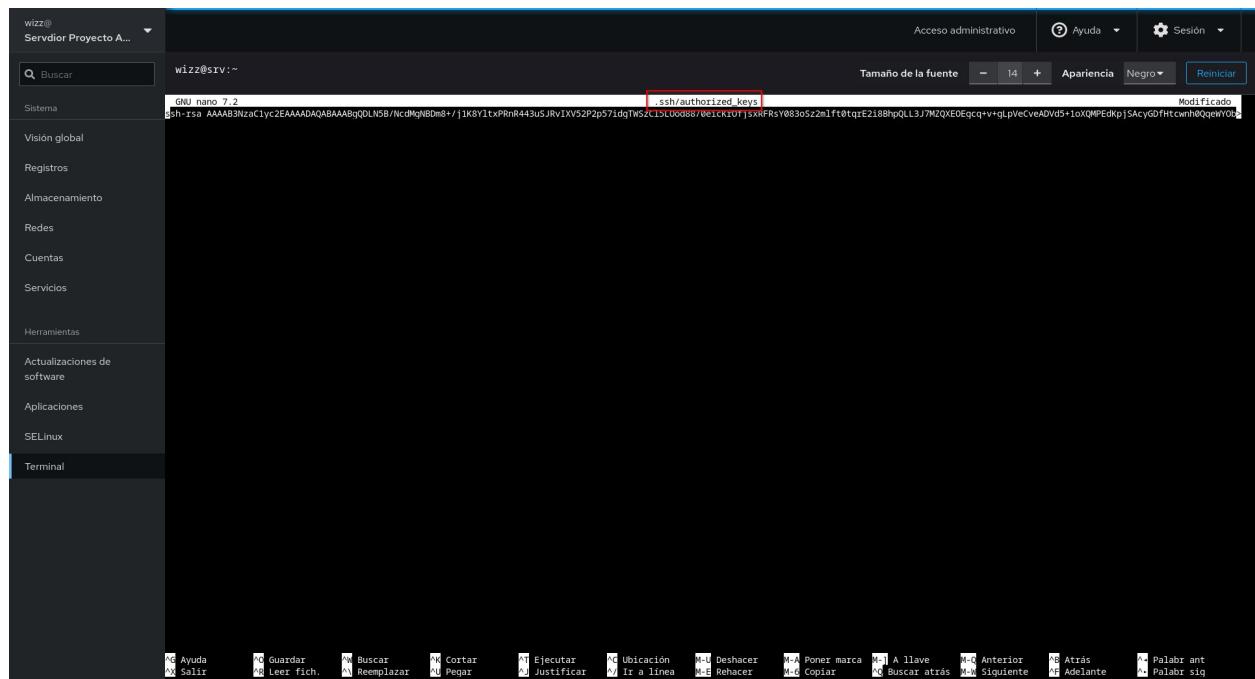
~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABgQDLSNB/NcdMgNBm8+/j1K8YlxPRnR443uSJrv1Xv52P2p57idgTwSzC15L0od88/eiCKzOfjsxRFsV083o5z2m1ft0tqrE2i8BhpQL3J7MZXQEOfgcq+v+qlpVeCveADvd5+loXQMPedkpjSacyGfHtcwnh0QqeWYob+Tdv1qcAE4P4LKZetsh0442K6PvzM
hse3Es5TkHg/emEWLSJebeem5zc1TP03fr5j+gNM1q4kzyzqyBLwiaco2LM7Mrzxz+2NHGAtmtn0Nqj/f6ITV4ew0p2Fq2md1K75zzTyInjdP5BAqmjPhdq5SwWeb+xt0zWBgA13ZkwFV+quR33vzV7R4KM0X1M2GFpxbtw6k0t1v1zNwJ1JBfaRMkNT7kwIMSiqcgfd5yX0s8ZFH2hvbcP4wJS
MSVNMIU8bbevRuptyk4ArghrJPNE1HUMp1c/mjzrLod1NloUFDg6IQztouWbUp+oyVaw58c8LCqBU= wizz@nobara-laptop

```

Una vez generada la pareja de claves, la forma más rápida y sencilla de autorizar el cliente en el servidor ssh, es mostrar el contenido de la clave publica del cliente simplemente con:

**cat ~/.ssh/id\_rsa.pub**

Copiaremos el contenido de la clave publica y nos dirigiremos al servidor, para lo que podemos utilizar la herramienta de consola de Cockpit.



Una vez en la consola del servidor, nos dirigimos al directorio del usuario en el servidor para el que deseamos autorizar acceso con clave publica (en este caso root), y editaremos el archivo **~/.ssh/authorized\_keys**. Podemos hacerlo directamente con el editor de consola nano. Pegamos el contenido de la clave publica a autorizar, y guardamos los cambios. Si dicho directorio y archivo, no existen, se pueden crear sin problema con sendos **mkdir** y **nano** o **touch**.

```
(wizz) 192.168.68.244 — Konsole
Nueva pestaña Dividir vista 
> ssh wizz@192.168.68.244
[  ] USA LA TERMINAL CON RESPONSABILIDAD
HAVE FUN!!!
Enter passphrase for key '/home/wizz/.ssh/id_rsa': [ ]
```

Ahora que hemos añadido la clave pública del cliente a la lista de claves autorizadas en el servidor, si volvemos a intentar loguearnos, en este caso como root, lo primero que el servidor nos solicita es la contraseña de desbloqueo de la clave pública (Importante, clave de desbloqueo de la pubkey, no la contraseña del usuario).

```
(wizz) 192.168.68.244 — Konsole
Nueva pestaña Dividir vista 
> ssh wizz@192.168.68.244
[  ] USA LA TERMINAL CON RESPONSABILIDAD
HAVE FUN!!!
Enter passphrase for key '/home/wizz/.ssh/id_rsa':
web console: https://svz.ismael-server.ddns.net:9090/ or https://192.168.68.244:9090/
Last login: Tue Apr 30 19:15:31 2024 from 192.168.68.235
wizz : $ whoami
wizz : $ hostname
svz.ismael-server.ddns.net
[  ]
```

¡Y voila!, el servidor ssh reconoce nuestra clave pública y nos autoriza el acceso, como se puede comprobar con sendos **hostname** y **whoami**, que confirman que somos root en el equipo `svz.ismael-server.ddns.net`

## 8. El runtime de contenedores. RedHat Podman

Llegamos al eje central de este proyecto de ASIR, la herramienta que nos permitirá ejecutar la práctica totalidad de este proyecto, el runtime de contenedores Podman, patrocinado por RedHat.

En partes anteriores de este proyecto, ya se ha explicado brevemente que es un contenedor, o un pod, no obstante, siempre es pertinente definir las cosas con un poco más de profundidad.

**Contenedor:** En sistemas informáticos, definimos un contenedor, como una unidad de software que incluye una aplicación, empaquetada junto con todas sus dependencias y que se ejecuta en un entorno aislado, aprovechando las capacidades del Kernel Linux de ejecutar dichas aplicaciones de forma aislada del resto del sistema. En la práctica, un contenedor puede ser usado para desplegar un servicio de la misma forma que se podría utilizar una máquina virtual para el mismo cometido. Si bien el contenedor no posee el hardware virtualizado que la máquina virtual si posee, restándole capacidades que no vayan más allá de ejecutar una aplicación, tiene como ventaja intrínseca la ligereza. Un contenedor, al no ser una máquina virtual, tiene un rendimiento muy superior al de una máquina virtual que cumpla la misma función, con rendimientos muy próximos al 100% del de la maquina real.

**Pod:** Concepto propio de Kubernetes y la unidad de software más pequeña que este puede desplegar.

Podemos definir un pod como un conjunto de contenedores que comparten características, hardware y recursos del host.

Para entender mejor lo que implica esta definición, pensemos en un servidor de contenedores que ejecuta 2 servicios, cada 1 en un contenedor. Si el contenedor 1 necesita acceder a datos del contenedor 2 o viceversa, es posible hacerlo mediante una conexión de red, teniendo cada uno de los contenedores una dirección IP propia; es decir, cada contenedor se comporta en la práctica como un “mini servidor” independiente que conecta con el otro.

Si esos mismos contenedores, son agrupados dentro de un pod, pasan a compartir los recursos que obtienen del host, y podrían comunicarse el uno con el otro directamente a través de su bucle local virtual 127.0.0.1; es decir que los pods pueden actuar como “mini servidores” independientes que ejecutan múltiples servicios.

Siguen además, teniendo la capacidad de conectarse con otros contenedores y Pods por conexión TCP/IP en caso de ser necesario. Esto convierte a los pods en una útil forma de agrupar contenedores de forma coherente, que de otra forma irían aislados cada uno por su lado.

Una vez hemos definido las unidades de virtualización con las que vamos a trabajar, toca responder a la pregunta **¿Por qué este proyecto se basa concretamente en Podman y no en Docker?**

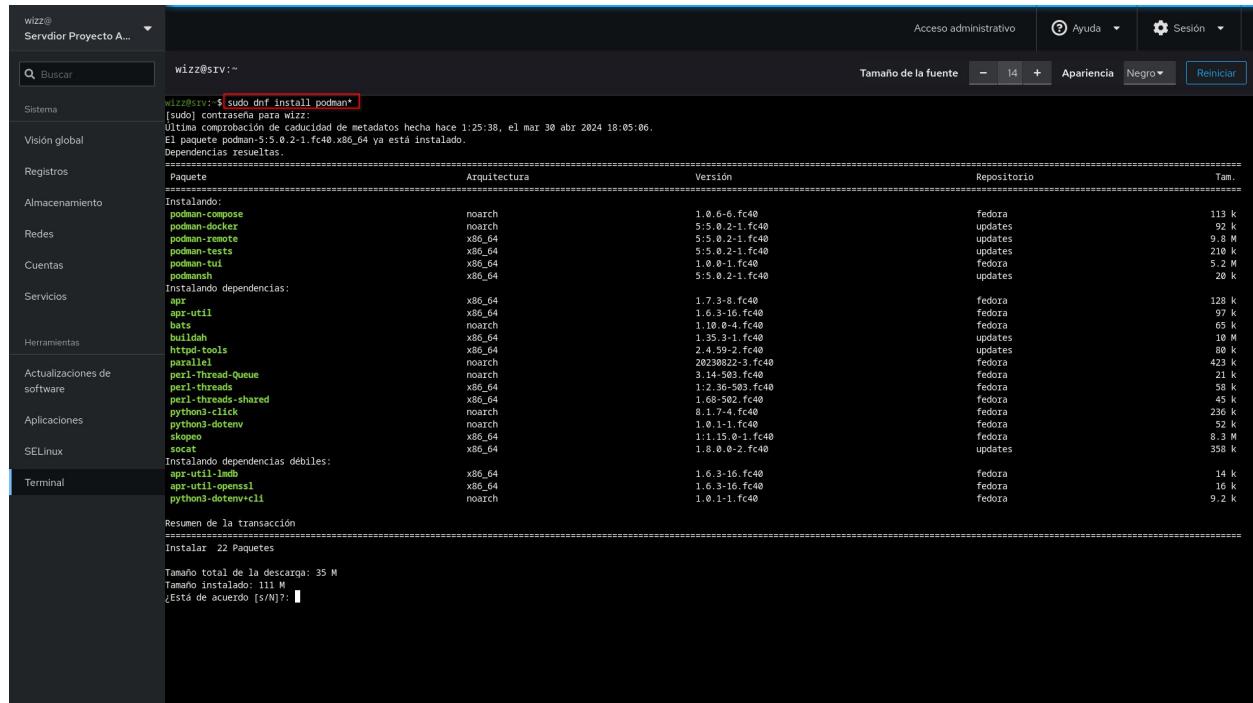
## 8.1 Podman. Diferencias respecto a Docker

Docker es quizá el runtime de contenedores más conocido desde su irrupción en 2013. Sin embargo, el autor de este proyecto opina que Podman posee algunas ventajas respecto a Docker, y por ende, es el runtime elegido para realizar este proyecto:

- **Sintaxis compatible casi al 100%:** No es una ventaja en sí misma, no obstante, sí que es una facilidad para usuarios ya acostumbrados a Docker. La gran mayoría de operaciones a realizar con Podman se realizan con un comando idéntico a Docker simplemente sustituyendo a este último por Podman, hasta el punto de que Dan Walsh, ingeniero de RedHat involucrado en el proyecto Podman, bromeo en su presentación sobre como migrar de Docker a Podman, alegando que para usar Podman lo único que había que hacer era ejecutar **alias Docker = Podman**. Si bien no pasa de una broma, es cierto que casi siempre, si se sabe hacer con Docker, se sabe hacer con Podman.
- **Runtime de contenedores Daemonless:** A diferencia de Docker, que ejecuta constantemente un demonio de sistema para el control de los contenedores, Podman prescinde de este último. Las implicaciones de seguridad de este punto son importantes, puesto que un contenedor que sea comprometido en Docker por algún tipo de vulnerabilidad grave, podría acabar dando acceso privilegiado al sistema principal. Con Podman, se elimina este riesgo al no haber un Daemon privilegiado que comprometer. No significa que una aplicación en Podman sea 100% segura, pero sí que se elimina una importante superficie de ataque.
- **Uso de Pods:** Docker como runtime no posee la capacidad de lanzar o gestionar pods, solo contenedores individuales. Si bien este proyecto sería perfectamente ejecutable usando solo contenedores, la posibilidad de usar pods aporta una funcionalidad interesante y una capa adicional de sofisticación al mismo.

## 8.2 Podman. Instalación

Pasamos pues a dejar Podman listo para operar. El soporte del mismo en Fedora es total, pues ambos productos son proyectos patrocinados por RedHat, lo que implica que Podman se encuentra por defecto en los repositorios de Fedora, y en muchas ocasiones instalado de forma predeterminada.



```
wizz@SIV: ~
[sudo] contraseña para wizz:
Última comprobación de caducidad de metadatos hecha hace 1:25:38, el mar 30 abr 2024 18:05:06.
El paquete podman-5.5.0.2-1.fc40.x86_64 ya está instalado.
Dependencias resueltas.
=====
Paquete           Arquitectura   Versión          Repositorio  Tam.
=====
Instalando:
podman-compose      noarch        1.0.6-6.fc40    fedora       113 k
podman-docker       noarch        5:5.0.2-1.fc40  updates      92 k
podman-remote       x86_64        5:5.0.2-1.fc40  updates      9.8 M
podman-tests        x86_64        5:5.0.2-1.fc40  updates      210 k
podman-tui          x86_64        1.0.6-1.fc40    fedora       5.2 M
podmanush          x86_64        5:5.0.2-1.fc40  updates      20 k
Instalando dependencias:
apr                x86_64        1.7.3-8.fc40    fedora       128 k
apr-util            x86_64        1.6.3-16.fc40   fedora       97 k
bats               noarch        1.10.0-4.fc40   fedora       65 k
buildah            x86_64        1.35.3-1.fc40   updates      10 M
httpd-tools         x86_64        2.4.59-2.fc40   updates      80 k
parallel            noarch        20230822-3.fc40 fedora       423 k
perl-thread-queue  noarch        3.1-16.fc40     fedora       21 k
perl-threads       x86_64        1.2.36-503.fc40 fedora       58 k
perl-threads-shared x86_64        1.68-502.fc40   fedora       45 k
python3-click       noarch        8.1.7-4.fc40    fedora       236 k
python3-dotenv     noarch        1.0.1-1.fc40    fedora       52 k
skopeo              x86_64        1.1.15-0.1.fc40 fedora       8.3 M
socat              x86_64        1.8.0.8-2.fc40   updates      358 k
Instalando dependencias débiles:
apr-util-lmdb      x86_64        1.6.3-16.fc40   fedora       14 k
apr-util-openrczl  x86_64        1.6.3-16.fc40   fedora       16 k
python3-dotenv+cli  noarch        1.0.1-1.fc40    fedora       9.2 k
Resumen de la transacción
=====
Instalar 22 Paquetes

Tamaño total de la descarga: 35 M
Tamaño instalado: 111 M
¿Está de acuerdo [S/N]? ■
```

Sin embargo, seguimos necesitando algunos componentes que serán usados en este proyecto, ya sean forma de librerías o aplicaciones. Por tanto, el primer paso es instalar las mismas, simplemente ejecutando **sudo dnf install podman\***, lo cual instalará de una tacada todo el software disponible junto con sus dependencias.

The screenshot shows the Cockpit application manager interface. On the left is a sidebar with navigation links: Sistema, Visión global, Registros, Almacenamiento, Redes, Cuentas, Servicios, Herramientas, Actualizaciones de software, Aplicaciones (which is selected and highlighted with a red box), SELinux, Terminal. The main area is titled "Aplicaciones" and lists several packages with their descriptions and "Instalar" (Install) buttons. The "Podman" entry is also highlighted with a red box.

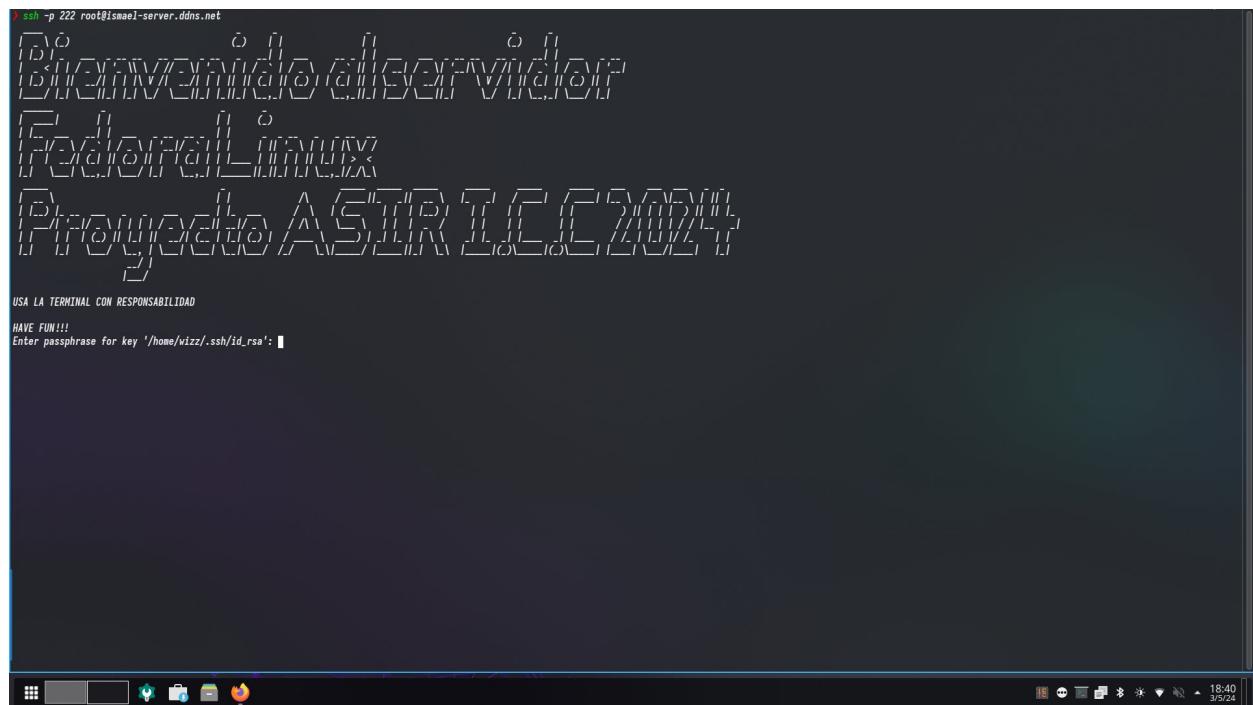
También instalaremos la aplicación para Podman en nuestra consola Cockpit desde el panel de aplicaciones. No la usaremos durante este proyecto, pues su funcionalidad es limitada, pero puede ser útil para algunas operaciones como iniciar, reiniciar, parar o eliminar contenedores y pods.

The screenshot shows the Cockpit Podman interface. On the left is a sidebar with navigation links: Propietario, Sistemas, Visión global, Registros, Almacenamiento, Redes, Contenedores de Podman (which is selected and highlighted with a red box), Cuentas, Servicios, Herramientas, Actualizaciones de software, Aplicaciones, SELinux, Terminal. The main area has two sections: "Imágenes" (Images) which shows "0 imágenes en total, 0" and "No hay imágenes"; and "Contenedores" (Containers) which shows "No hay contenedores". There are buttons for "Mostrar Todos", "Crear pod", and "Crear contenedor".

Una vez instalada tendremos disponible el panel de Podman. De momento esta vacío, pues en esta máquina virtual no hay nada ejecutándose.

## 8.3 Podman. Creación de la red

Llego la hora de meternos en harina y comenzar el despliegue en sí. Nuestros contenedores y Pods, necesitan una red para comunicarse entre sí, y a su vez con el host. Por tanto, el primer paso es la creación de una red dedicada al proyecto.



```
ssh -p 222 root@ismael-server.ddns.net
USA LA TERMINAL CON RESPONSABILIDAD
HAVE FUN!!!
Enter passphrase for key '/home/wizz/.ssh/id_rsa':
```

Llegamos por fin al servidor real en el que podman y Fedora están previamente instalados y listos para desplegar el proyecto. El primer paso es loguearnos en el servidor. Podríamos usar la consola incorporada en Cockpit, pero por razones de costumbre, buena parte del proyecto será desarrollado por SSH.

Introducimos el correspondiente comando de ssh con el usuario root y el nombre de dominio que apunta a mi IP pública. Nótese que el puerto de conexión es el 222, pues al estar ssh expuesto al exterior, tengo mi router doméstico configurado para redirigir las conexiones a ssh por ese puerto de entrada. El sistema nos solicitará el desbloqueo de la clave pública como en puntos anteriores.

```

root@srv.ismael-server.ddns.net ~
OS: Fedora Linux 39 (Server Edition) x86_64
Host: Dell Latitude 7400
Kernel: 6.8.8-0.8.8.fc39.x86_64
Uptime: 12 hours
Terminal: /dev/pts/1
CPU: Intel L5-6580T (4) @ 3.100GHz
GPU: Intel HD Graphics 530
Memory: 8671MB / 7342kB
Disk: 1TB Western Digital-Puett-Puckard Company Device [103c:8055]
CPU Usage: 38%
Local IP: 192.168.1.120
Public IP: 89.35.140.212
Users: root

```

```

Filesystem Size Used Avail Use% Mounted on
/dev/sda2 446G 188G 344G 22.3% /
tmpfs 88 0 88 /tmp

```

✓ / with root@srv at 18:45:31

Estamos logueados, como muestran **hostname** y **whoami**.

```

x podman network ls
NETWORK ID      NAME      DRIVER
2f259bab93aa   podman   bridge
710f375984cb   principal bridge
f5f33e987f88   proyecto  bridge

```

✓ / with root@srv at 18:50:37

Comenzamos listando las redes presentes en podman con el comando **podman network ls**.

Podman al igual que Docker usa una sintaxis basada en **podman/función/acción a ejecutar**. En este caso, el comando principal es obviamente **podman**, **network** especifica que vamos a realizar una acción sobre las redes de podman y **ls** es la acción a ejecutar, que en este caso al igual que el Linux, es listar dichas redes. Si por ejemplo ejecutáramos **podman container ls** podman nos mostraría una lista de contenedores en lugar de redes.

Tras la ejecución, vemos que en estos momentos las únicas redes disponibles en podman son la red “**podman**”, que es la red que este crea y usa por defecto, y “**principal**” una red con la que usualmente hago experimentos en el servidor.

```

> podman network create proyecto
> podman network ls
NETWORK ID      NAME      DRIVER
2f259bab93aa   podman   bridge
710f375984cb   principal bridge
f5f33e987f88   proyecto  bridge

```

✓ / with root@srv at 18:55:25

Una simple ejecución del comando **podman network create proyecto**, crea una nueva red para interconectar en ella nuestros contenedores. Dicha red ahora se muestra en la salida de la orden **ls**.

Con esto ya tendríamos la red lista para interconectar nuestros contenedores.

Un aspecto interesante del runtime de contenedores, es que integra automáticamente un servidor DNS interno, por lo que los contenedores y pods conectados en una red, pueden ser referenciados con su nombre de host (o lo que es lo mismo, nombre de contenedor o pod), no necesitando usar sus direcciones IP.

Estamos Listos para comenzar a desplegar nuestros servicios.

## 9. Servicios del servidor. Descripción y despliegue

Llego la hora de comenzar a desplegar los servicios que verdaderamente proporcionaran una aplicación práctica al servidor.

Durante los próximos puntos, describiremos brevemente cada uno de esos servicios o aplicaciones, comentando sus principales características, dedicando un punto a cada uno de los 3 stacks que correrán en pods y al contenedor individual Caddy Server.

El despliegue de los mismos tiene un orden lógico, que será argumentado posteriormente.

### 9.1 Stack Nextcloud

Comenzaremos con el pod que da sentido al servidor en sí mismo; el servicio central que resulta de utilidad, pues el resto de aplicaciones y servicios son utilidades accesorias para el mantenimiento del servidor.

Este pod está compuesto por 3 contenedores íntimamente relacionados:

- **Nextcloud**: La aplicación pensada para ser de utilidad al usuario final
- **PostgreSQL**: El SGBD donde Nextcloud almacenara sus datos de uso interno
- **PgAdmin**: El panel de administración y editor gráfico para la gestión de dicha base de datos

Tanto si decidiéramos desplegar dicho stack como contenedores individuales, como si lo hicieramos en un pod, el primero de los contenedores debería ser PostgreSQL pues es necesario para completar la instalación de Nextcloud una vez desplegado el contenedor, no obstante y aunque mostraremos una secuencia de despliegue manual con comandos de podman de forma ilustrativa, la forma más sencilla y la que utilizaremos oficialmente con este y el resto de pods, será con un manifiesto de despliegue en Yaml y la herramienta **podman kube**.

Pero antes, pasemos a comentar brevemente PostgreSQL y PGadmin. No comentaremos al propio Nextcloud, pues ya ha sido descrito múltiples veces a lo largo de este proyecto.

## 9.1.1 El Sistema Gestor de Bases de Datos PostgreSQL

Buena parte de las aplicaciones disponibles en el mundo del software necesitan una base de datos para manejar su información interna. En no pocas ocasiones, dichas bases de datos suelen ser soluciones propietarias internas de la propia aplicación, que manejan dicha información de la forma que el desarrollador cree oportuno. Sin embargo, cuando una aplicación maneja un volumen considerable de información, los desarrolladores optan por usar soluciones específicamente dedicadas a la gestión de datos, y aquí es donde entran en juego los Sistemas Gestores de Bases de Datos.

Nextcloud es una aplicación del segundo grupo, que recurre a SGBD dedicados para manejar su información interna como usuarios, ruta a archivos, versiones, historial de actualizaciones e infinidad de datos más.

Por defecto Nextcloud permite la creación de una instancia que utiliza una instalación propia de SQLite; sin embargo, el propio instalador nos advierte de que dicha opción no es nada recomendable para entornos de producción o cargas de trabajo más allá de las más básicas, y desaconseja su uso en cualquier circunstancia si vamos a sincronizar nuestros archivos.

Si deseamos hacer uso de esa función, como es nuestro caso, nos vemos obligados a desplegar un SGBD, en el que Nextcloud pueda almacenar y gestionar su información. Además, desplegando un SGBD y una consola de administración del mismo, ganamos un servicio, que puede resultar útil para otros usos más allá de proporcionar una base de datos para Nextcloud. El SGBD que utilizará este proyecto es PostgreSQL.

Podríamos resumir a PostgreSQL simplemente diciendo que es el SGBD de licencia OpenSource más avanzado del mercado, pero... ¿Qué significa dicha afirmación?

PostgreSQL es un SGBD relacional orientado a objetos, con una robustez y flexibilidad muy altas. Permite la extensión de funcionalidades personalizadas, acceso y modificación de datos no relacionales en formatos como json, almacenamiento y cálculo de coordenadas espaciales entre otras muchas bondades.

Se trata de un SGBD que saca pecho en cuestiones de escalabilidad con bases de datos de gran tamaño, donde su rendimiento se ve mucho menos afectado que otros SGBD como MySQL.

Es 100% *ACID Compliance*, es decir que garantiza la **atomicidad, consistencia, aislamiento y durabilidad**, y garantiza una alta concurrencia gracias al sistema MVCC (Control de concurrencias multiversion), que permite a varios procesos acceder a la misma tabla sin realizar un bloqueo garantizando la consistencia.

Todas estas características le convierten en una opción a tener en cuenta para cualquier necesidad que un SGBD deba cubrir.

## 9.1.2 PostgreSQL. Despliegue del pod y primer contenedor

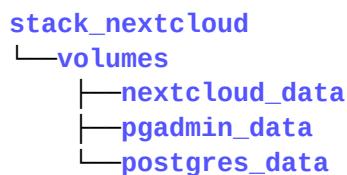
Comenzaremos desplegando este primer servicio mediante comandos manuales de podman, para realizar una prueba de concepto del uso del runtime de contenedores. Los posteriores pods y contenedores los lanzaremos mediante manifiestos de despliegue yaml.

El primer paso por supuesto es conectarnos a la consola de nuestro servidor con un usuario con privilegios administrativos tal y como prueba la captura superior. Esta captura nos muestra también con los comandos **podman pod ls** y **podman container ls** que en estos momentos no hay ningún pod ni contenedor corriendo en el servidor.

```
> mkdir -p stack_nextcloud/volumes/nextcloud_data  
> mkdir -p stack_nextcloud/volumes/postgres_data  
> mkdir -p stack_nextcloud/volumes/pgadmin_data  
> tree stack_nextcloud  
stack_nextcloud  
└── volumes  
    ├── nextcloud_data  
    ├── pgadmin_data  
    └── postgres_data  
  
5 directories, 0 files
```

Comenzaremos creando la estructura de directorios que almacenaran los volúmenes de cada contenedor, no solo de nextcloud o PostgreSQL, sino de todos nuestros pods y contenedores. Almacenaremos dichos volúmenes para la persistencia de datos de los contenedores con una estructura similar a la que se aprecia en la captura superior, con una carpeta dedicada a cada pod, y una al servidor Caddy, seguido de un directorio de volúmenes y en su interior un directorio por cada volumen necesario, con un nombre descriptivo.

Por cada pod, debería quedarnos una jerarquía similar a esta:



Una vez creada la estructura de directorios para los volúmenes del pod, podemos proceder a crearlo con su correspondiente comando.

```
[root@podman ~]# podman pod create --name proyecto_stacknc -p 80:80 -p 443:443 --network proyecto
20250bc824a9860bde1798f9ee934719595945b18b9e327aa68ac2d93e9b881
POD ID          NAME      STATUS    CREATED           INFRA ID   # OF CONTAINERS
20250bc824a9    proyecto_stacknc  Created   7 seconds ago  dbab9509e2a5  1
[root@podman ~]#
```

Listo, el pod está creado y listo para alojar los contenedores.

El comando de creación:

```
podman pod create --name proyecto_stacknc -p 80:80 -p 443:443 --network proyecto
```

Sigue la siguiente lógica:

- **podman pod create:** El comando principal con el que indicamos que vamos a crear un nuevo pod
- **—name:** Parámetro para especificar el nombre que recibirá el pod. Si no se especifica se le dará un nombre aleatorio
- **-p:** Parámetro para especificar los puertos del pod, que serán expuestos a través de los del host
- **—network:** Red a la que se desea conectar el nuevo pod. Esta debe existir previamente a la creación del mismo

```
[root@podman ~]# podman run --pod proyecto_stacknc --name postgres -e POSTGRES_PASSWORD=Asirpro2024 -e POSTGRES_USER=admin -v /root/podman/proyecto/stack_nextcloud/volumes/postgres_data:/var/lib/postgresql/data:z -d postgres:latest
? Please select an image:
  registry.fedoraproject.org/postgres:latest
  registry.access.redhat.com/postgres:latest
  docker.io/library/postgres:latest
  quay.io/postgres:latest
```

Pasamos a ejecutar el contenedor de PostgreSQL con el comando que se muestra en la captura superior. Al no existir previamente la imagen de PostgreSQL, podman nos pregunta de qué repositorio deseamos obtenerla. Usaremos el repositorio de Docker **docker.io**

Explicamos a continuación la sintaxis del comando de despliegue del contenedor

```
podman run --pod proyecto_stacknc --name postgres -e
POSTGRES_PASSWORD=Asirpro2024 -e POSTGRES_USER=admin -v
/root/podman/proyecto/stack_nextcloud/volumes/postgres_data:/var/lib/postgresql/d
ata:z -d postgres:latest
```

- **podman run:** Al igual que **Docker run**, esta es la orden básica para desplegar un contenedor en podman
- **—pod:** Este parámetro es exclusivo de podman y no existe en Docker. Con él especificamos que deseamos desplegar el contenedor dentro de un pod y a continuación especificamos el pod en el que deseamos despegarlo
- **—name:** Nombre del contenedor
- **-e:** Declaración de variables de entorno del contenedor. Estas son exclusivas de cada contenedor, y debemos consultarlas en la documentación de cada contenedor. Siguen una estructura de **CLAVE=valor** y no son obligatorias, pero simplifican en gran medida el proceso de despliegue de servicios

- v: Declaración de volúmenes. Este parámetro especifica la ruta en la que se encuentra los volúmenes que previamente hemos preparado para el contenedor. Al igual que las variables de entorno, no es un parámetro obligatorio, pero debemos usarlo si deseamos que el contenedor tenga persistencia. Su estructura básica es **/ruta/en/host:/ruta/montaje/contenedor:Z** donde :Z especifica que el volumen es exclusivo del contenedor y no es compatible con el resto.

```
> podman run --pod proyecto_stacknc --name postgres -e POSTGRES_PASSWORD=Asirpro2024 -e POSTGRES_USER=admin -v /root/podman/proyecto/stack_nextcloud/volumes/postgres_data:/var/lib/postgresql/data:z -d postgres:latest
✓ docker.io/library/postgres:latest
Trying to pull docker.io/library/postgres:latest...
Getting image source signatures
Copying blob ae3bb1b347a4 done
Copying blob 1a0a0a0a0a0a done
Copying blob 283a77db7b done
Copying blob 9729ced65421 done
Copying blob dda2a8fb5d5d done
Copying blob 91d2729fa4d5 done
Copying blob f8484d9c98ea done
Copying blob c1990ff16b05 done
Copying blob e8d55fd4d415 done
Copying blob c1cb13b19888 done
Copying blob 87352e5f8c7 done
Copying blob 050d9f8c3b1c done
Copying blob 710e142785f8 done
Copying blob cb28c2a5f899 done
Copying config 8e4fc9e184 done
Writing manifest to image destination
3ebcb69749e9cb708a28211782e7ec7a5d557554c82f4a8581d83621ad863241
  ↪ took 1m 52s ✘ with root@... at 18:39:07
```

Tras unos breves instantes, la imagen es descargada, y el contenedor ha sido desplegado. La zona resaltada en rojo representa el nombre único interno del contenedor recién desplegado.

```
> podman container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
99d354d4453 docker.io/library/postgres:latest 17 seconds ago Up 17 seconds 0.0.0.0:80→80/tcp, 0.0.0.0:443→443/tcp f6dd192e518-infra
50d391507f38 docker.io/library/postman-pause:4.9.4-1711445992 59 seconds ago Up 16 seconds 0.0.0.0:80→80/tcp, 0.0.0.0:443→443/tcp postgres
  ↪ took 1m 51s ✘ with root@... at 18:51:14
```

Un listado de los contenedores desplegados, da fe de que se ha desplegado correctamente. El otro contenedor que se muestra, es un contenedor genérico conocido como podman-pause, y actúa como “carcasa” para los pods. Siempre que se despliegue un pod, ese contenedor sera desplegado automáticamente.

Ya tenemos lista una instancia de PostgreSQL, pasemos ahora a desplegar su consola de administración.

## 9.1.3 Consola de administración del SGBD. PGAdmin

Si bien es perfectamente posible gestionar PostgreSQL mediante el intérprete de ordenes SQL... Estaremos de acuerdo en que gestionar bases de datos relacionales solo a base de ordenes SQL no es la forma más eficiente o amigable.

Se hace necesario pues una consola gráfica de administración que nos permita gestionar el SGBD y sus bases de datos de forma más agradable e intuitiva, con algún proyecto equivalente a MySQL Workbench. Dicho Proyecto es PGAdmin.

PGadmin es un proyecto OpenSource, muy ligado al proyecto PostgreSQL con una interfaz cuidada, intuitiva y muy funcional.

Se distribuye como aplicación php ejecutable en un servidor web como apache o nginx, o como contenedor que integra todo lo necesario (nuestro caso).

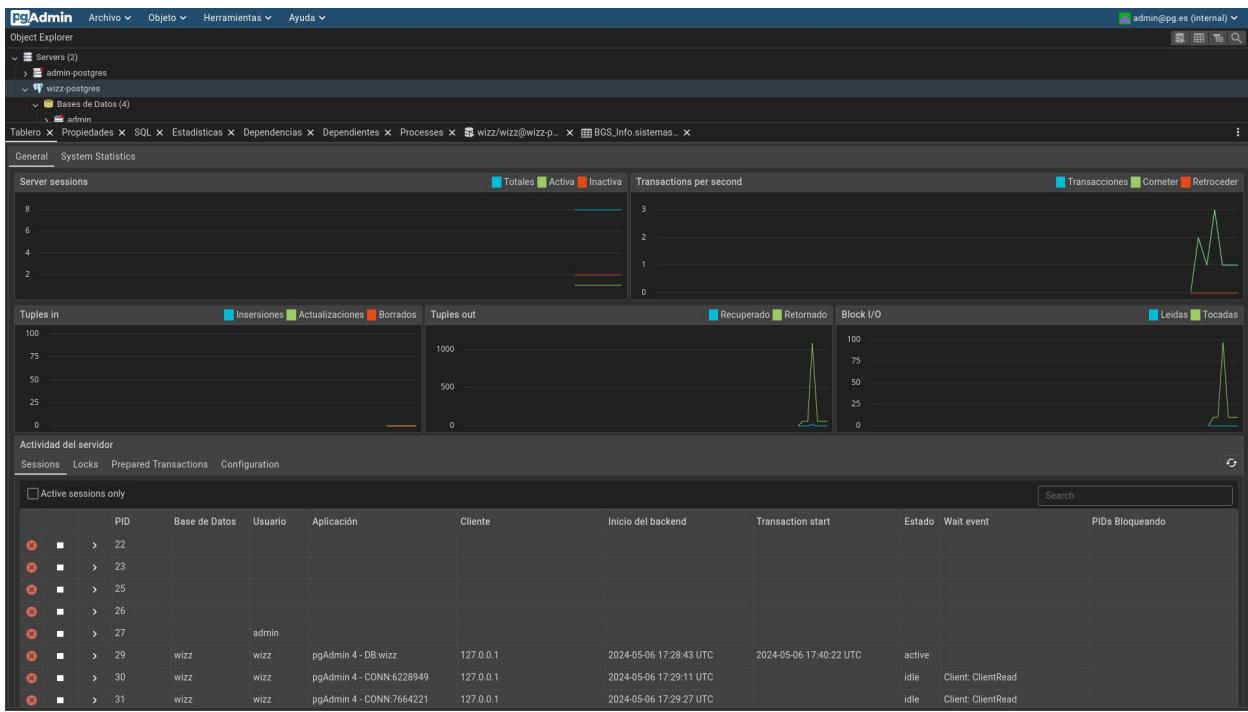
PGadmin nos ofrece:

- Editor SQL con revisión de sintaxis y autocompletado
- Vista de tabla y resultados con posibilidad de adición y edición directa
- Gestión de Roles y permisos del SGBD
- Importación / exportación de datos
- Creación, edición y gestión de propiedades de bases de datos, esquemas, tablas y columnas
- Panel de rendimiento con métricas del SGBD

The screenshot shows the PGAdmin application window. At the top is a menu bar with Archivo, Objeto, Herramientas, and Ayuda. Below the menu is the Object Explorer, which is expanded to show 'Procedimientos', 'Sequencias', 'Tablas (2)', and 'sistemas'. The 'sistemas' node is further expanded to show 'sistemas\_proxy', 'Tablas Foráneas', 'Tipos', and 'Vistas'. A red box highlights this section. The main workspace is divided into several panes: 'Explorador de objetos' (Object Explorer), 'Editor de consultas' (Query Editor) containing the SQL command 'SELECT \* FROM "BGS\_Info".sistemas ORDER BY name ASC', and 'Vista de resultados' (Results View) displaying a table of system data. The table has columns: name [PK], economy, population, res?, target\_faction, proxy\_faction, proxy\_res. The results pane also shows a message 'Total rows: 10 of 10 | Query complete 00:00:00.536' at the bottom left and 'Ln 1, Col 1' at the bottom right.

	name [PK]	economy	population	res?	target_faction	proxy_faction	proxy_res
1	Anarkyo	High Tech / Extraction	92276	[null]	Likedeler of michel	The Crusaders	5
2	Beta-2 Tucanae	Industrial / Extraction	930000	[null]	Likedeler of michel	Bluestar PMC	2
3	Divia Mocha	Industrial / Military	3730000	[null]	Likedeler of michel	United Dark Energy Astronomical Division	3
4	HIP 1173	Extraction / Agriculture	52691	[null]	Likedeler of Michel	Outlander Flight Wing	8
5	HIP 11768	Extraction / Refinery	21000	false	Likedeler of michel	Senta a pua	1
6	Mosna	Agriculture / High Tech	37662	[null]	Likedeler of Michel	Communist Party of Drakonia	4
7	Nano Picori	Industrial / Military	7738	[null]	Likedeler of Michel	United German Commanders	7
8	Nuglungbara	High Tech / Extraction	9400000	true	Likedeler of michel	Senta a pua	1
9	Sopedu	Refinery / Military	101024	[null]	Likedeler of Michel	Obsidian Order	6
10	Veturia	Extraction / High tech	27617	[null]	Likedeler of michel	Communist Party of Drakonia	4

Ejemplo de interfaz de PGAdmin



### Tablero de monitorización y métricas del SGBD

Como vemos, PGadmin será una herramienta valiosa para nuestro servidor, tanto para gestión de posibles problemas con la base de datos de Nextcloud, como para futuros usos de nuestro SGBD en futuros proyectos.

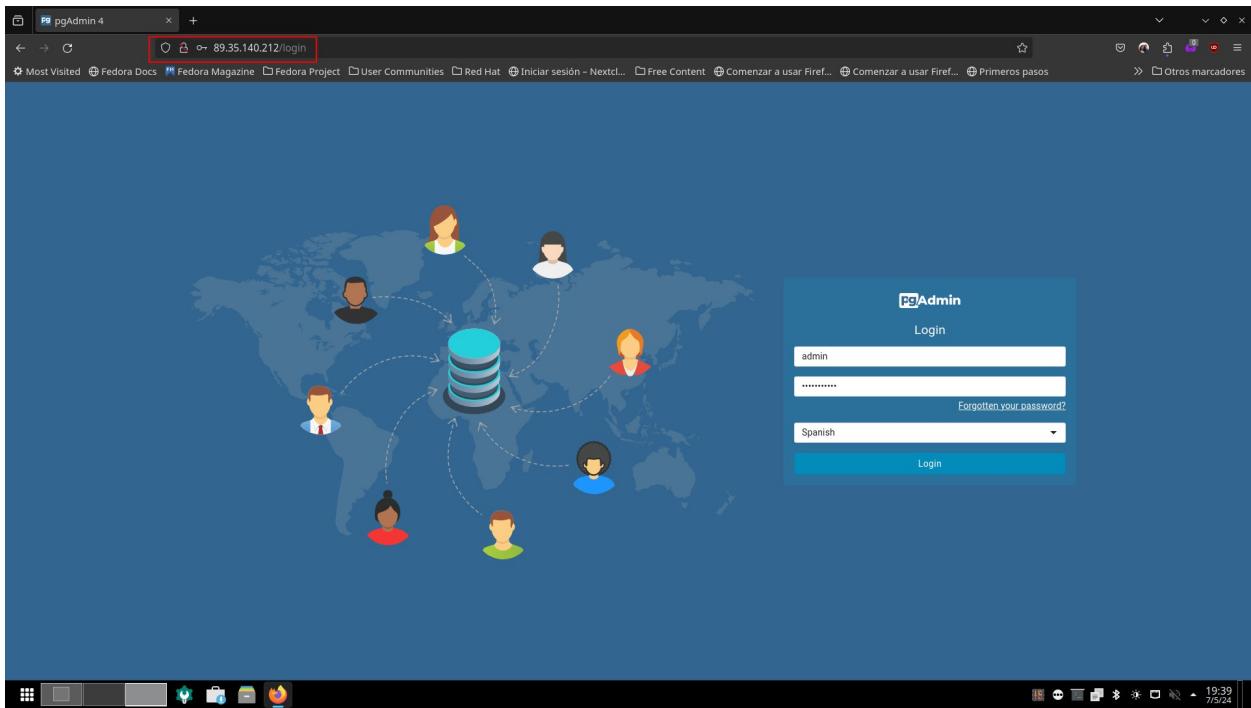
#### **9.1.4 PGAdmin. Despliegue del contenedor y preparación de la base de datos**

Pasamos ahora a desplegar el contenedor de PGAdmin en nuestro pod **stack\_nextcloud**.

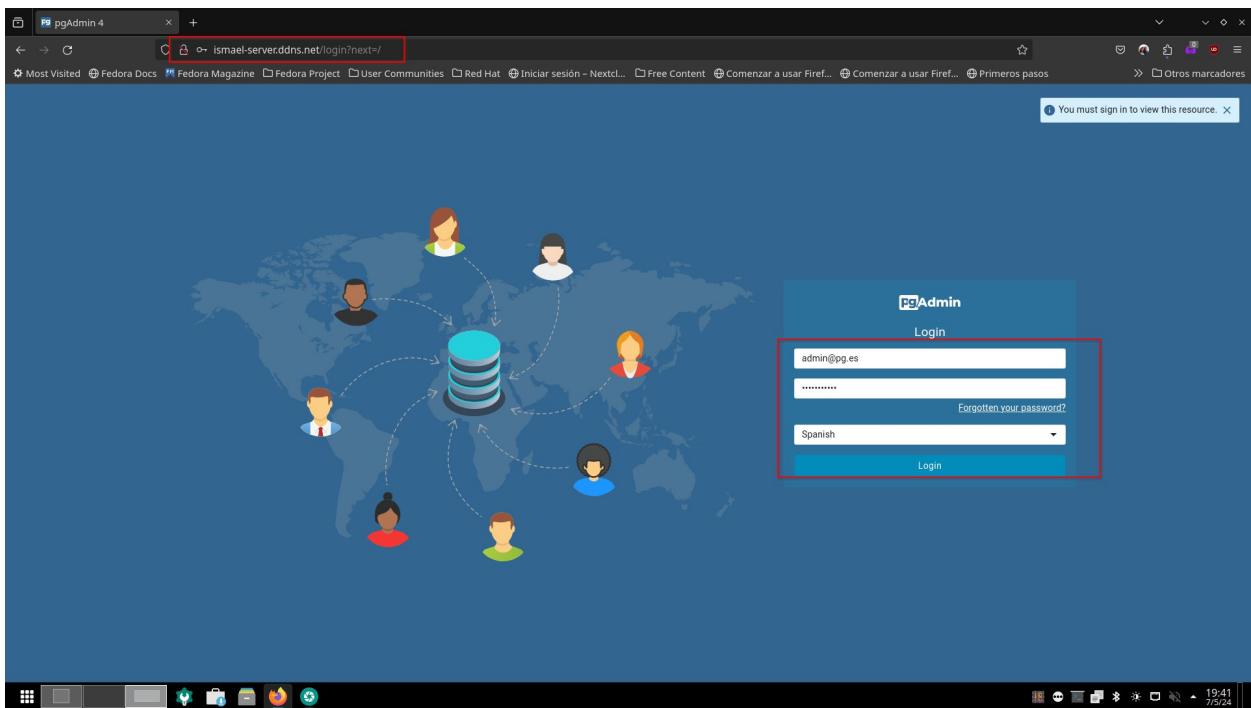
```
podman run --pod proyecto_stacknc --name pgadmin -e PGADMIN_DEFAULT_EMAIL=admin@pg.es -e PGADMIN_DEFAULT_PASSWORD=Asirpro2024 -e PGADMIN_LISTEN_PORT=8081 -v /root/podman/proyecto/stack_nextcloud/volumes/pgadmin_data:/var/lib/pgadmin2
d pgadmin/pgadmin
[docker.io/pgadmin4:latest
Trying to pull docker.io/pgadmin4:latest...
Getting image source signatures
Copying blob 4db54d640e done |
Copying blob fca837b190 done |
Copying blob 2fc81b2b245d done |
Copying blob 4abf2865143 done |
Copying blob 46712147a894 done |
Copying blob c38222c5f5f6 done |
Copying blob 4b22717c2693 done |
Copying blob 4ed37cc08cd done |
Copying blob 95e87eda23b4 done |
Copying blob 78f1580d83bc done |
Copying blob 7b6596e68888 done |
Copying blob 83cd4914ba13 done |
Copying blob ad289a3401fd done |
Copying blob 177a34239293 done |
Copying blob c346978625b6 done |
Copying blob bc295fc5bde done |
Copying config 5a75b53f24 done |
Writing manifest to image destination
35cce7a347e211fa29e7a1c5c218ed2bb57a870df918a2bb2ede877e8982
```

El proceso de despliegue del contenedor es idéntico al anterior, tan solo adaptando los datos que nos interesan del contenedor, como su nombre, variables de entorno y especificando la imagen que deseamos usar para su despliegue. En la captura se observa el despliegue correctamente. En este caso las variables de entorno del mismo definen el usuario por defecto de PGAdmin (Email), la contraseña de dicho usuario, y el puerto de escucha del servidor web de PGAdmin.

Por el momento nuestro pod está incompleto, pues aún debemos desplegar Nextcloud, pero ya tenemos una instancia de PostgreSQL y su panel de administración; así que vamos a probar que todo funciona. En este momento estamos fuera de la red local del servidor, por lo que intentaremos acceder a PGAdmin usando la IP pública del mismo.



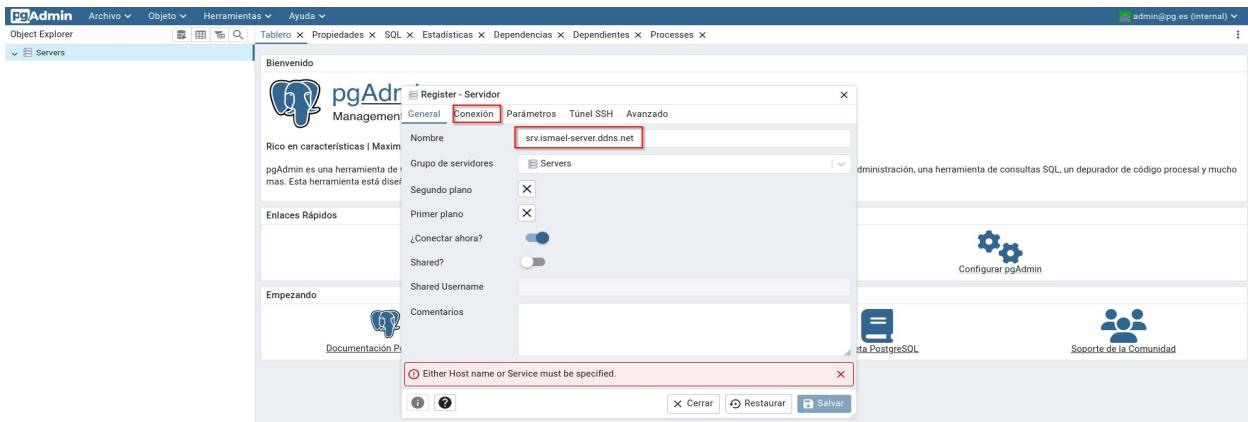
¡Perfecto! El contenedor funciona correctamente y esta redireccionando su puerto 80 al de nuestro servidor, por lo que podemos acceder a través de nuestra IP pública. Pero recordemos que este servidor va a usar un nombre de dominio que hemos adquirido en NoIP, **ismael-server.ddns.net**, probemos a usarlo.



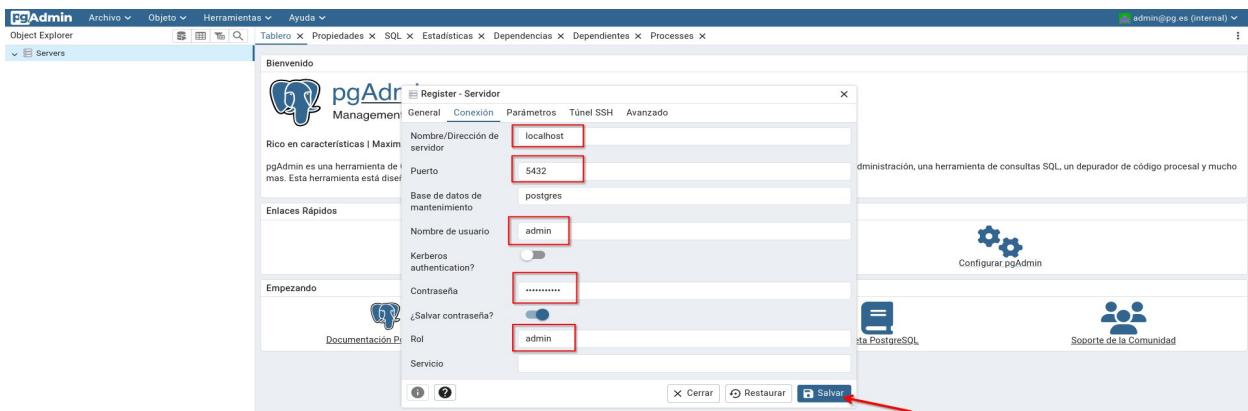
Como vemos, podemos acceder perfectamente con nuestro nombre de dominio. Vamos a loguearnos con el usuario y la contraseña que especificamos con las variables de entorno de creación del contenedor y preparemos el rol y la base de datos para nuestro siguiente y ultimo contenedor del pod, Nextcloud.



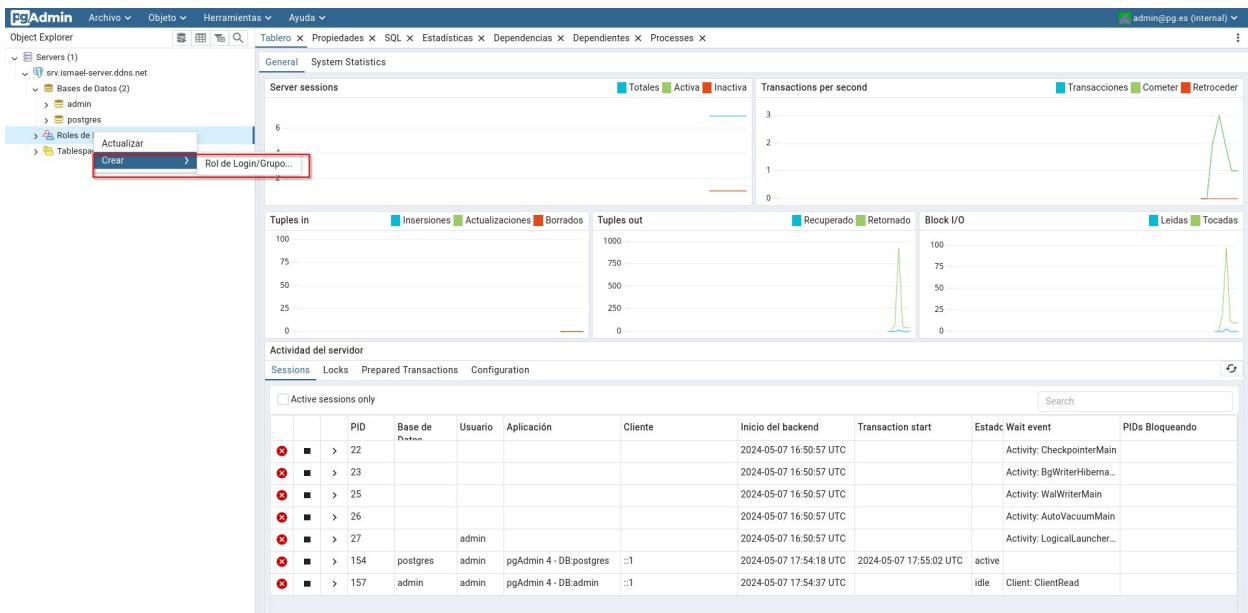
Ya estamos en PGAdmin y podemos comenzar a utilizarlo. El primer paso que debemos completar es agregar nuestra instancia de PostgreSQL a la lista de servidores usando el menú contextual de dicha lista como se muestra en la captura.



Se nos abrirá una ventana emergente en la que introduciremos el nombre identificativo de nuestro servidor. Usaremos el nombre de nuestro servidor y a continuación pasaremos a la pestaña de propiedades de conexión.

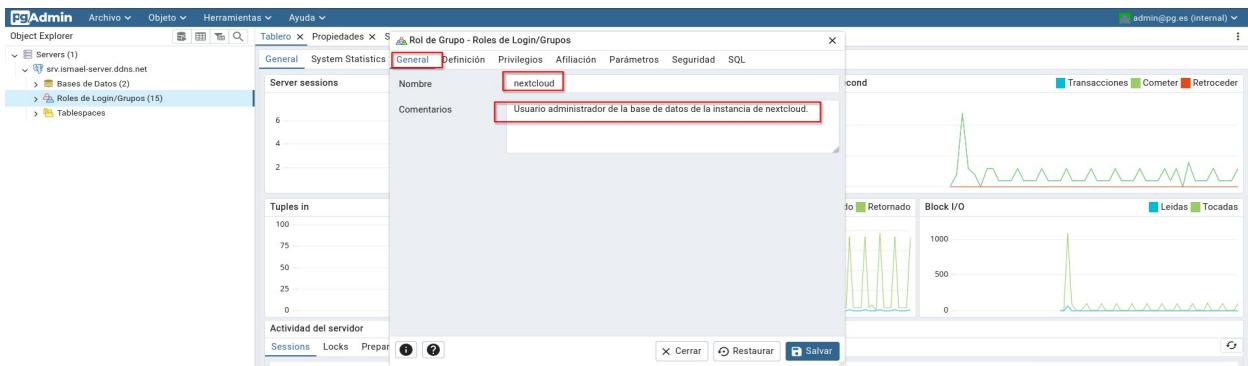


Aquí es donde podemos apreciar una de las características diferenciadoras de los pods de podman respecto a Docker. En un entorno de contenedores aislados, el SGBD tendría una dirección IP distinta a PGAdmin, pero los pods actúan a todos los efectos como un único “host virtual” por lo que en la conexión de PGAdmin con Postgre debemos especificar el servidor como localhost o 127.0.0.1. Introduciremos además el usuario admin y la contraseña, ambos especificados en las variables de entorno de despliegue del contenedor de PostgreSQL. Cuando los hayamos establecido, pulsamos sobre salvar.

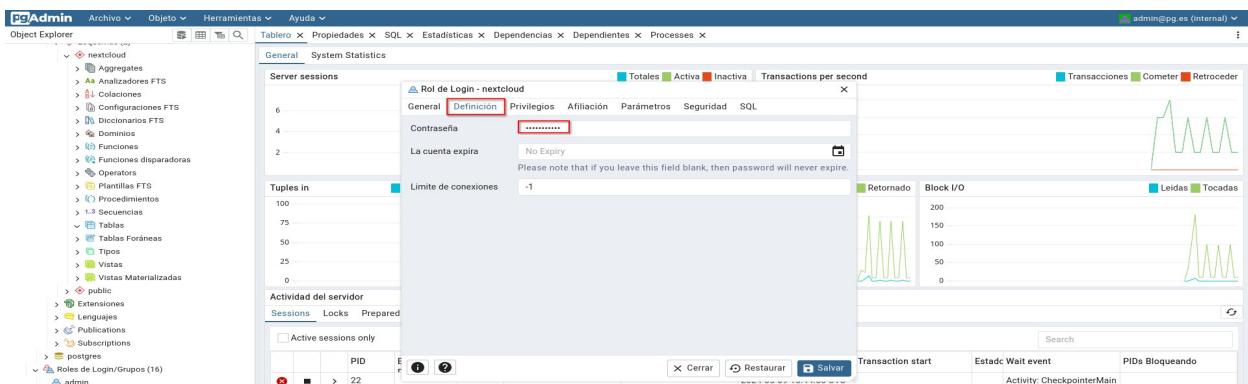


Listo. Hemos conectado PgAdmin a nuestra instancia de PostgreSQL. Desde ahora el servidor quedara grabado en la lista y podremos acceder rápidamente a él cada vez que nos logueemos en PgAdmin.

Ahora agregaremos el rol **nextcloud** a la instancia de Postgre, a la que asignaremos una base de datos homónima que será la que usará nuestra aplicación principal. Seleccionamos **crear > rol...**. En el menú contextual de **Roles de login / grupos**.



Se nos abrirá la ventana de creación de roles. Damos un nombre y una descripción al mismo en la pestaña **General** y a continuación pasamos a la pestaña **Definición**.

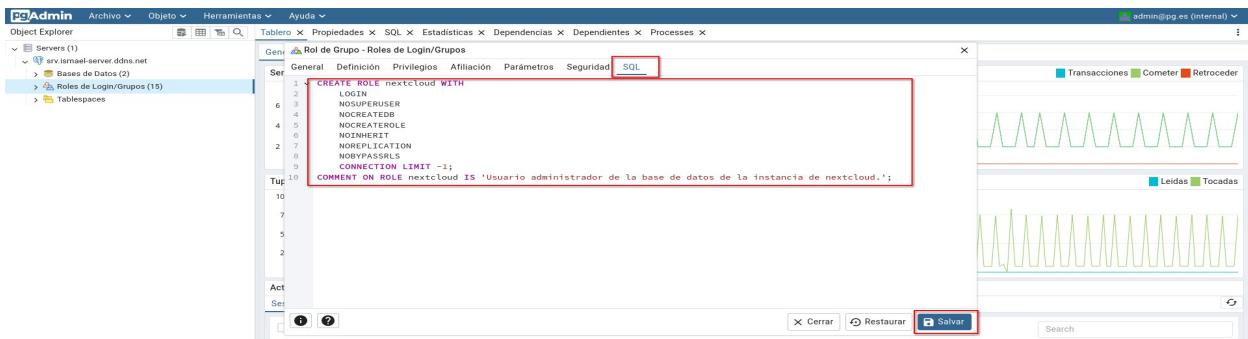


Aquí establecemos la contraseña del rol.

Administración de sistemas informáticos en red



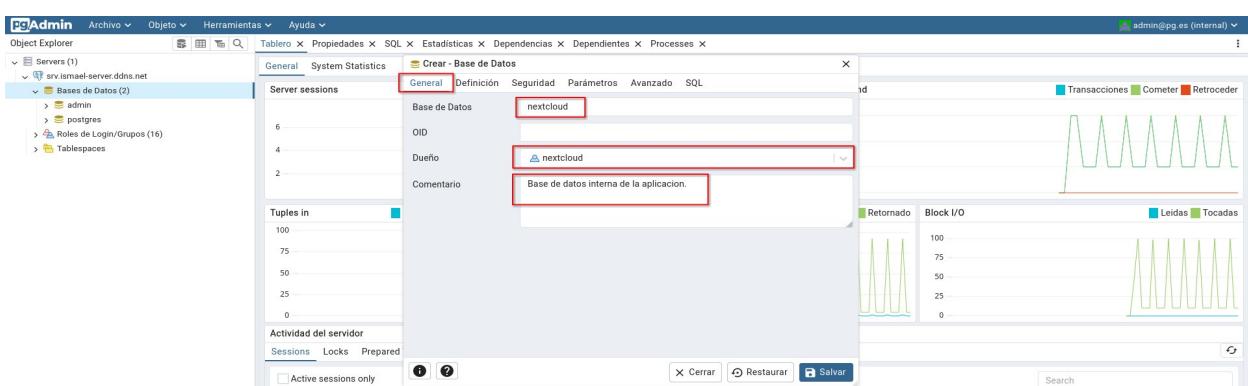
En la pestaña de privilegios le daremos permiso para iniciar sesión, para poder conectar y administrar la base de datos de nextcloud en caso de problemas con la misma.



Uno de los más interesantes detalles de PgAdmin es que para casi cualquier acción que sea realizada mediante medios gráficos, suele ofrecernos la sentencia SQL equivalente a nuestras acciones, lo que es una útil fuente de aprendizaje en el lenguaje SQL. Una vez listo el nuevo rol para nextcloud pulsamos en salvar y este quedara almacenado.



Pasemos a crear la base de datos para Nextcloud. Pulsaremos en **Crear > Bases de datos** en el menú contextual de **Bases de datos** en el explorador de objetos.

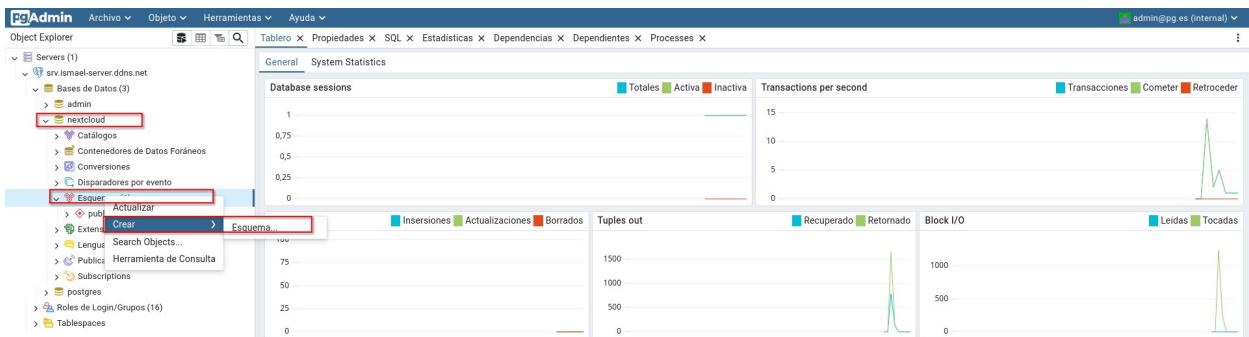


Damos a la nueva base de datos el nombre deseado, seleccionamos en el desplegable **dueño** el rol nextcloud recién creado, de esa forma dicho rol tendrá control absoluto sobre esa base de datos. Podemos además darle un comentario descriptivo a la misma. Pulsamos en **Salvar** para consolidar los cambios.

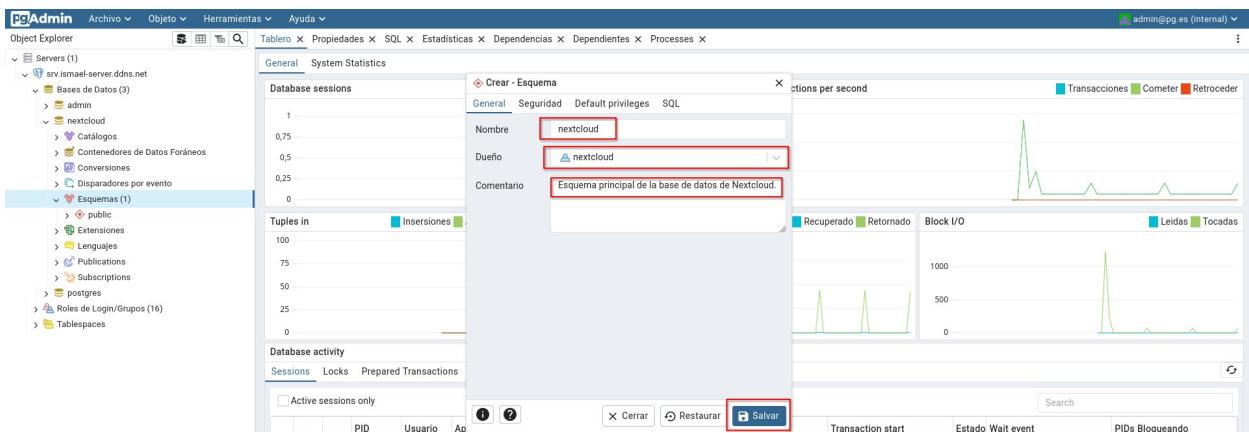
60

Administración de sistemas informáticos en red

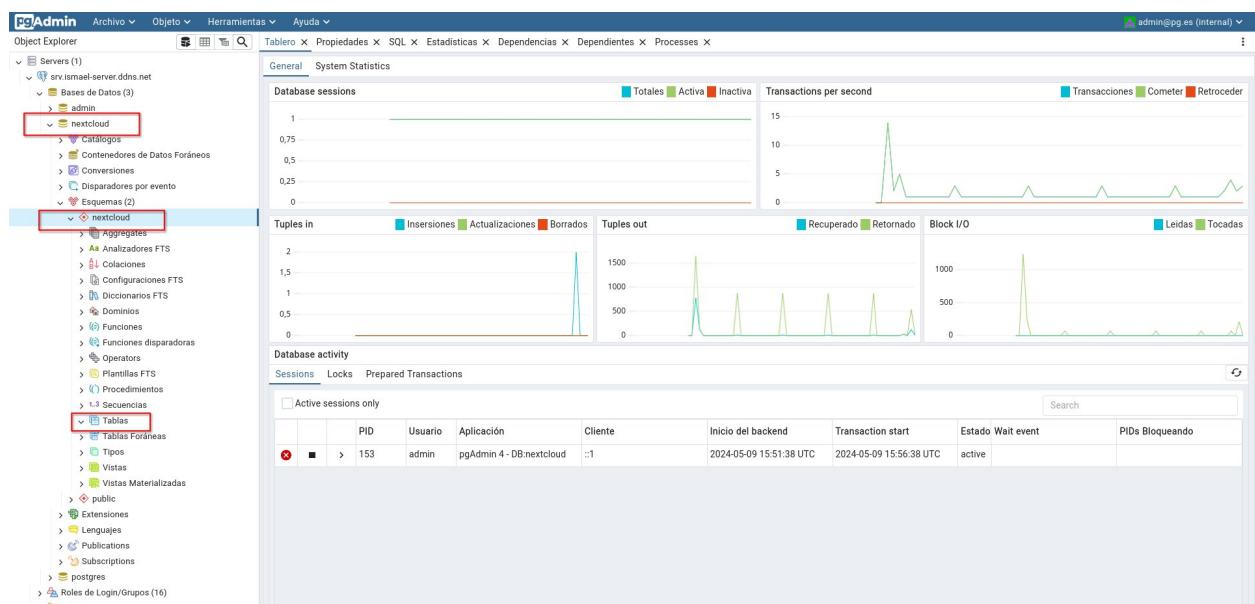
Ismael Carrasco Cubero



PostgreSQL usa un paradigma similar a Oracle, siendo las bases de datos colecciones de esquemas (donde realmente se almacenan las tablas), por lo que vamos a crear un esquema homónimo.



La creación es idéntica a las bases de datos, damos nombre al esquema, le asignamos propietario (rol nextcloud) y le damos una descripción. Pulsamos en **Salvar** y listo.



Ya tenemos el entorno de SGBD listo para desplegar nextcloud. De momento no hay tablas en el esquema, pero la propia aplicación se encargará de crear todas las necesarias; pasemos pues al siguiente contenedor.

## 9.1.5 Contenedor Nextcloud. Despliegue y configuración básica

Estamos listos para desplegar la aplicación principal. Nextcloud no necesita demasiada presentación ni justificación a estas alturas, por lo que procedemos a meternos en harina directamente.

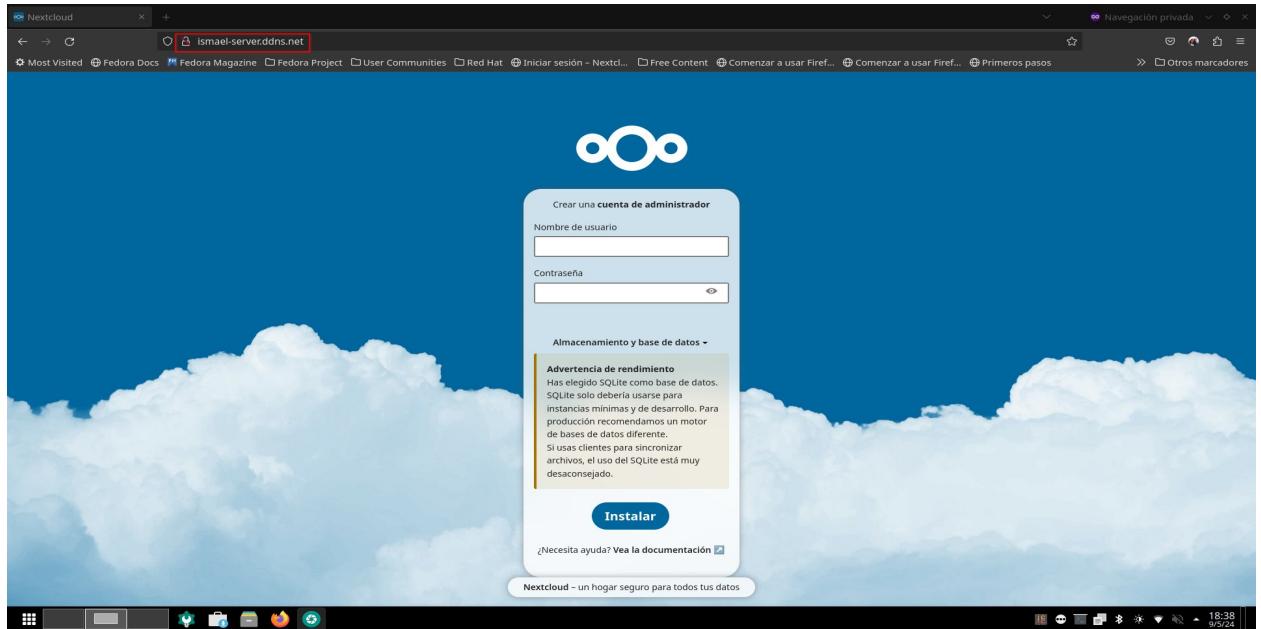
```
[root@ismael ~]# podman run --pod proyecto_stacknc --name nextcloud -d -v /root/podman/proyecto/stack_nextcloud/volumes/nextcloud_data:/var/www/html/data:z docker.io/library/nextcloud:stable-apache
ad2f5110cfcad11a1f959dec8a768fb47287b13e58da8e52dfa0babaf923...
[root@ismael ~]# podman pod ls && podman container ls
PODNAME          STATUS    CREATED             COMMAND
proyecto_stacknc Running   About a minute ago  52b8f5c1c258 4
CONTAINER ID        IMAGE               CREATED             STATUS    PORTS     NAMES
0f3d9c87c3c...   proyecto_stacknc:running   About a minute ago  Up About a minute  0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp  0f3d9c87c3c-infra
52b8f5c1c258   localhost/podman-pause:4.9.4-1711445992  About a minute ago  Up About a minute  0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp  postgres
b24771fcfa7d   docker.io/library/postgres:latest    About a minute ago  Up About a minute  0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp  postgres
c3b58505fd7d   docker.io/page/pgadmin4:latest      About a minute ago  Up About a minute  0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp  pgadmin
ad2f5110cfc...   docker.io/library/nextcloud:stable-apache  apache2-foreground...  29 seconds ago  Up 29 seconds  0.0.0.0:80-81->80-81/tcp, 0.0.0.0:443->443/tcp  nextcloud
[root@ismael ~]#
```

Lanzamos directamente Nextcloud con el siguiente comando de despliegue:

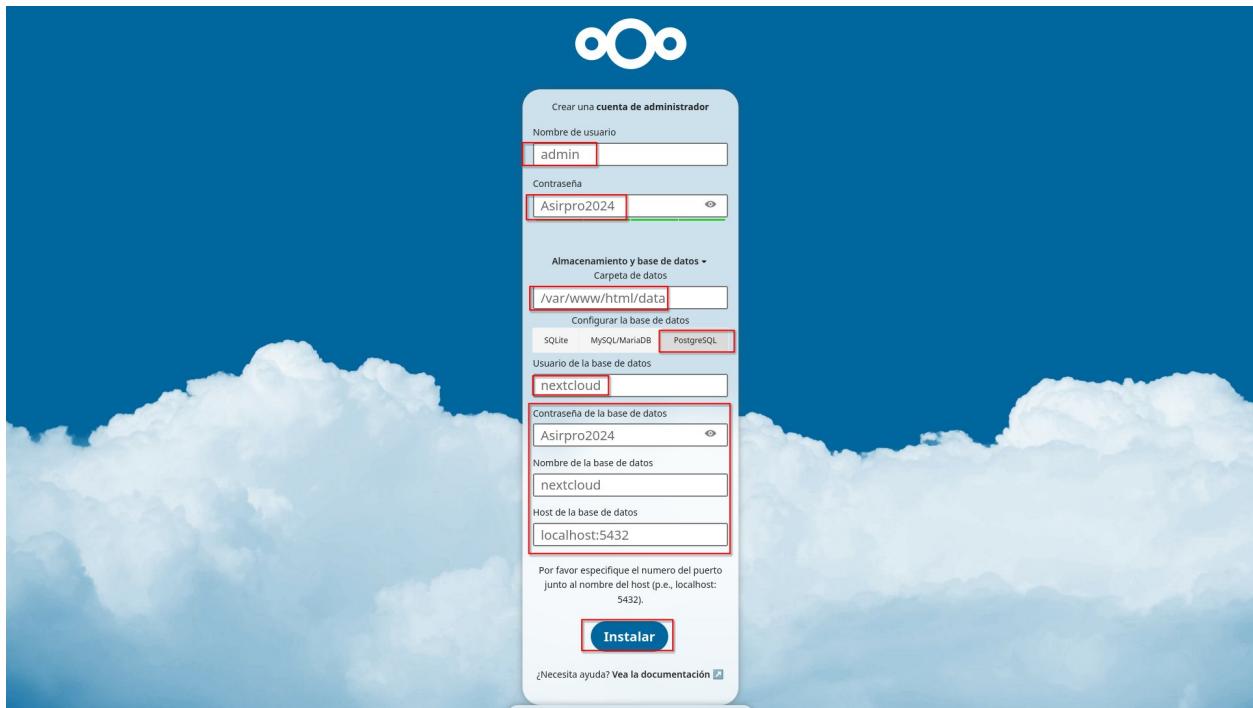
```
podman run --pod proyecto_stacknc --name nextcloud -d -v /root/podman/proyecto/stack_nextcloud/volumes/nextcloud_data:/var/www/html/data:z docker.io/library/nextcloud:stable-apache
```

Similar al resto de comandos de despliegue, usamos la orden **podman run** con los parámetros necesarios. A saber:

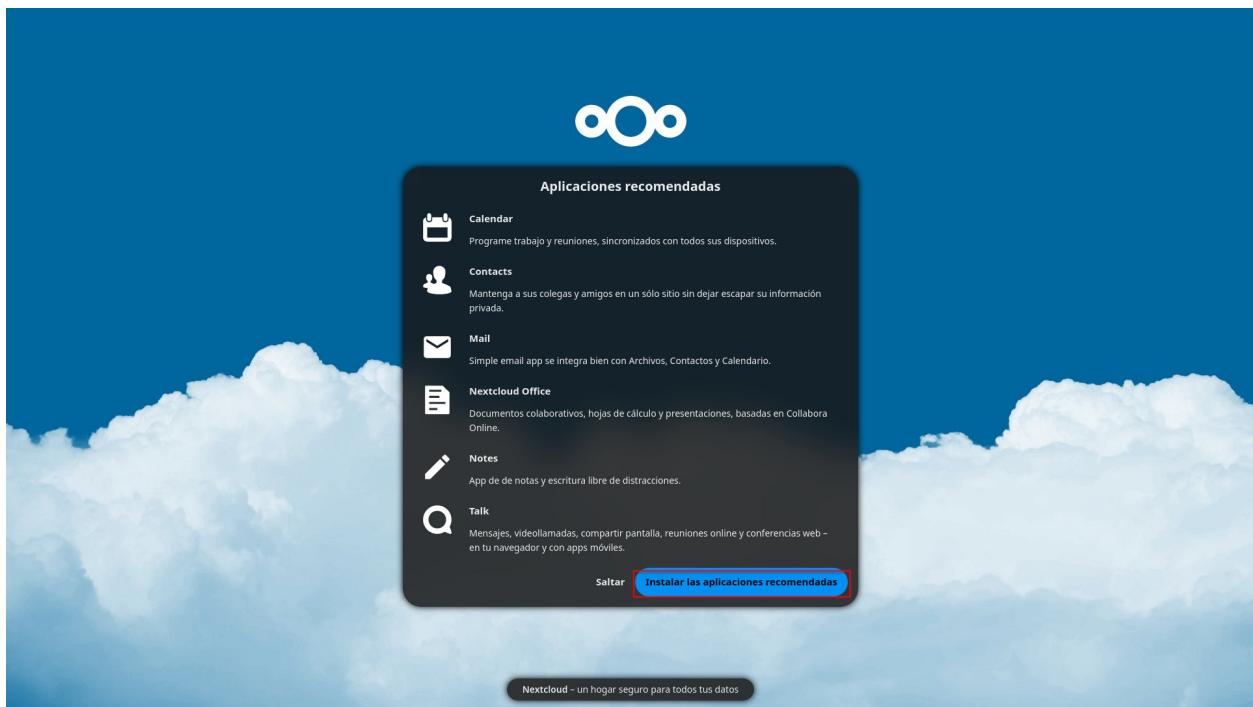
- **—pod:** Especifica el pod en el que lanzar el contenedor
- **—name:** nombre del contenedor
- **-d:** Modo “*detached*” o desacoplado. Para que el contenedor quede arrancado en segundo plano y no capture nuestra terminal
- **-v:** Declaración de volúmenes usados por el contenedor
- **docker.io/library/nextcloud:stable-apache** especifica la imagen a utilizar por el contenedor



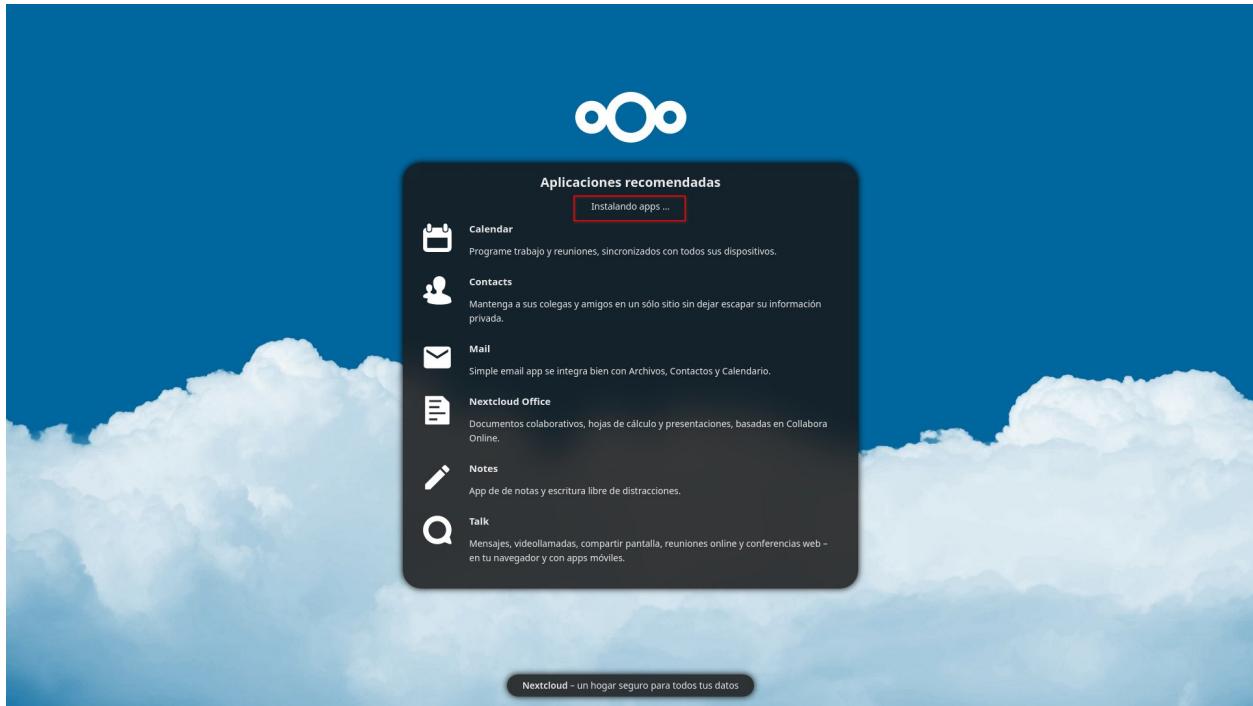
Como vemos, nuestro contenedor esta activo y atendiendo peticiones a través de nuestro nombre de domino. Nextcloud nos da la bienvenida con una pantalla para realizar la configuración inicial de la instancia. Como se ha comentado previamente, nos desaconseja encarecidamente utilizar la instancia propia de SQLite. Pasemos a configurar la instancia.



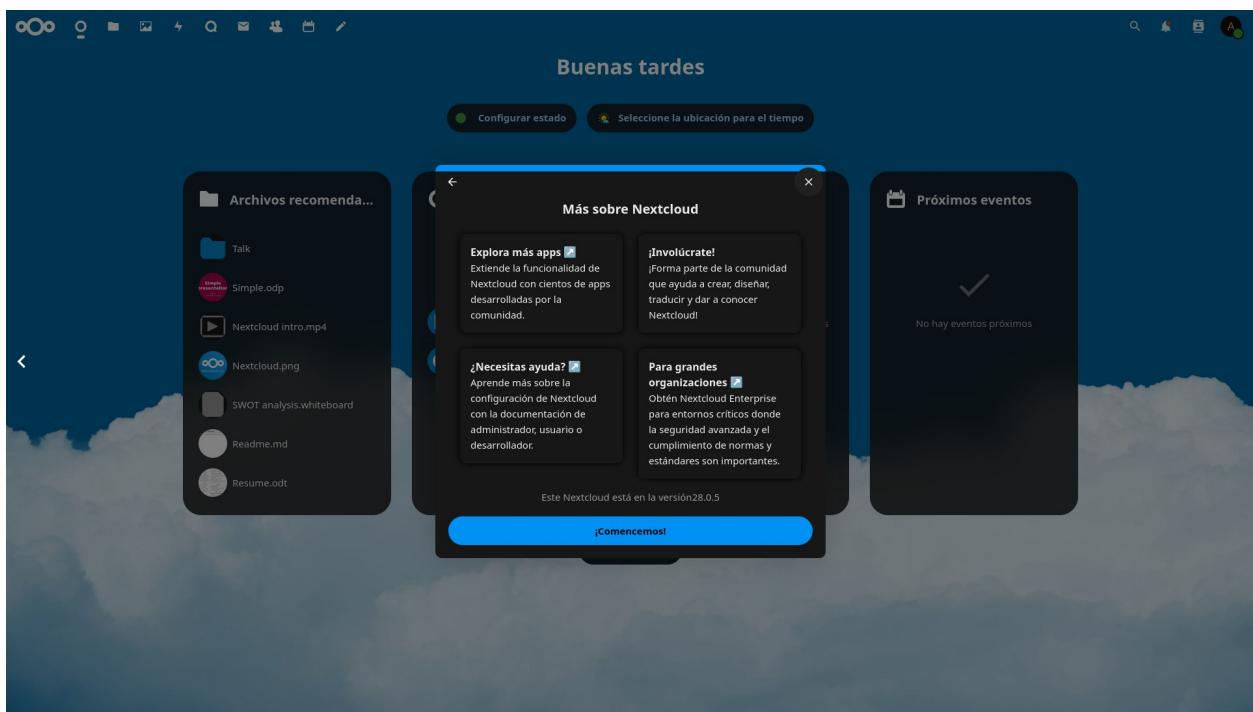
El proceso es muy sencillo, solo hemos de cumplimentar el formulario con los datos que nos pide como el nombre de usuario administrador, la contraseña del mismo, y los parámetros de acceso a la base de datos. Es importante asegurarse de escoger el SGBD que corresponda en el apartado de la base de datos. Al estar PostgreSQL en el mismo pod, una vez más hemos de especificarle como localhost añadiendo al mismo el puerto bien conocido de Postgre, el 5432 (o el que hayamos configurado manualmente). Cuando estemos listos, pulsamos sobre **Instalar**.



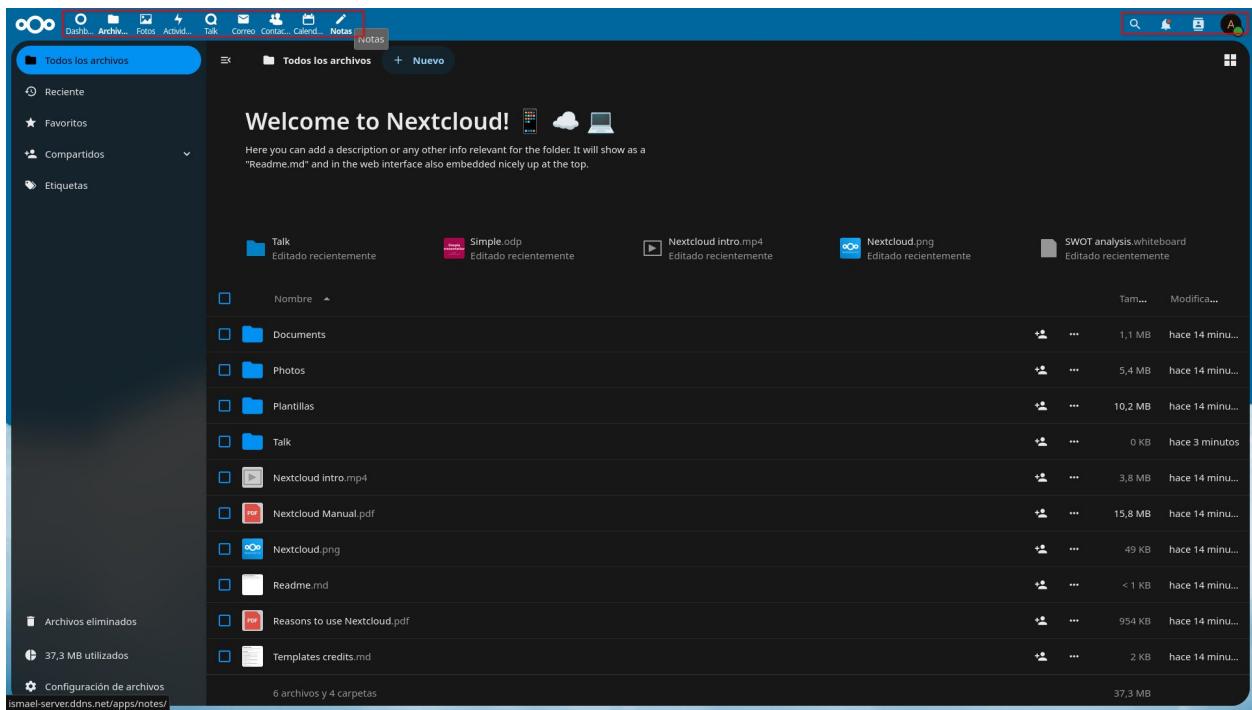
Nextcloud incorpora de serie un buen surtido de aplicaciones recomendadas de desarrollo propio del proyecto. En el siguiente paso nos preguntara si deseamos instalar dichas aplicaciones recomendadas o deseamos saltar este paso. Puesto que algunas de ellas resultan muy útiles, las instalaremos.



Ya solo queda esperar a que la instalación se complete.



Tras una breve sesión de diapositivas sobre el producto, podemos empezar a utilizarlo.



Podemos acceder a las principales aplicaciones instaladas desde el panel superior izquierdo y las opciones de cuenta, notificaciones, contactos y búsqueda desde el panel superior derecho.

## 9.1.6 Stack Nextcloud. Despliegue automatizado de pods con Podman Kube y manifiesto Yaml

Ya tenemos desplegada la aplicación principal, con su base de datos y el panel de administración de la misma, pero hemos utilizado en total 4 comandos con múltiples parámetros para realizar dicho despliegue. Este proceso es algo engorroso y además no es la forma standard para desplegar pods.

¿Podemos desplegar todo el stack con un único comando que además es más simple que los de cada contenedor por separado? La respuesta es **sí** y está en la herramienta **podman kube**. Con esta herramienta el despliegue de un pod puede automatizarse; lo único que la herramienta necesita para desplegar el pod al completo es un “guion” que le especifique que debe hacer. Dicho guion es el manifiesto Yaml basado en la sintaxis de Kubernetes.

Puesto que los manifiestos yaml pueden tener una extensión considerable, explicaremos a continuación usando capturas de un típico yaml su sintaxis básica.

```
GNU nano 7.2
$ cat stack_nextcloud.yaml
apiVersion: v1
kind: Pod
metadata:
  name: proyecto_stacknc
spec:
  volumes:
    - name: nextcloud-data
      hostPath:
        path: /root/podman/proyecto/stack_nextcloud/volumes/nextcloud_data
    - name: postgres-data
      hostPath:
        path: /root/podman/proyecto/stack_nextcloud/volumes/postgres_data
    - name: pgadmin-data
      hostPath:
        path: /root/podman/proyecto/stack_nextcloud/volumes/pgadmin_data
  containers:
    - name: nextcloud
      image: docker.io/library/nextcloud:stable-apache
      ports:
        - containerPort: 80
          hostPort: 80
      volumeMounts:
        - mountPath: /var/www/html/data/z
          name: nextcloud-data

```

Los yaml que utilizaremos se dividen principalmente en 3 áreas principales.

- Cabecera:** Área de inicio del archivo. Es obligatoria y especifica la versión del api de Kubernetes a utilizar, el tipo de despliegue (Pod), y el nombre del Pod.
- Área de especificaciones:** En ella se pueden personalizar los recursos asignados en forma de RAM, CPU o almacenamiento del host. Nosotros solo haremos uso del almacenamiento, pues no necesitamos limitar la potencia a nuestros pods.
- Definición de contenedores:** Donde se especifican los contenedores a desplegar dentro del pod, estableciendo su nombre, puertos que expondrá si los hay, variables de entorno de los mismos, volúmenes a utilizar (De los previamente declarados en especificaciones) y la imagen en la que se ha de basar el contenedor.



En la captura sobre estas líneas se aprecia con más detalle una configuración típica para contenedor, usando de ejemplo PGAdmin. Cada uno de los parámetros de configuración ha de ir en su propia categoría (**name**, **image**, **env**, **ports**, **volumemounts**).

Cuando diseñamos manifiestos yaml de Kubernetes para utilizar con podman o el propio Kubernetes es especialmente importante prestar mucha atención a la indentación del archivo, pues nos dará error de sintaxis ante la más mínima desviación en la misma.

```

$ pod ls && container ls
POD ID NAME STATUS CREATED INFRA ID # OF CONTAINERS
945d6cd9a61295294acac4dc71fc8de836bfff1b53dcacf3d64312e2b250e1f
Containers:
8c0ff23c3bec6c8eccc9d767585ac0a79413775142f14881eeccb94c84795b8cc5
272824a3dd51f23613a54e654964fa2e9cb3ed626bf83a33ed08ca8a36969bb
e8665d3f78855bbdecdaa1291da4d476c1c318b6f364b2fde859f8941e528
pod:
945d6cd9a61295294acac4dc71fc8de836bfff1b53dcacf3d64312e2b250e1f
Containers:
8c0ff23c3bec6c8eccc9d767585ac0a79413775142f14881eeccb94c84795b8cc5
272824a3dd51f23613a54e654964fa2e9cb3ed626bf83a33ed08ca8a36969bb
e8665d3f78855bbdecdaa1291da4d476c1c318b6f364b2fde859f8941e528
pod ls && container ls
POD ID NAME STATUS CREATED INFRA ID # OF CONTAINERS
945d6cd9a61295294acac4dc71fc8de836bfff1b53dcacf3d64312e2b250e1f
Containers:
8c0ff23c3bec6c8eccc9d767585ac0a79413775142f14881eeccb94c84795b8cc5
272824a3dd51f23613a54e654964fa2e9cb3ed626bf83a33ed08ca8a36969bb
e8665d3f78855bbdecdaa1291da4d476c1c318b6f364b2fde859f8941e528
pod:
945d6cd9a61295294acac4dc71fc8de836bfff1b53dcacf3d64312e2b250e1f
Containers:
8c0ff23c3bec6c8eccc9d767585ac0a79413775142f14881eeccb94c84795b8cc5
272824a3dd51f23613a54e654964fa2e9cb3ed626bf83a33ed08ca8a36969bb
e8665d3f78855bbdecdaa1291da4d476c1c318b6f364b2fde859f8941e528

```

Una vez confeccionado el manifiesto yaml y ejecutado el comando, el pod queda desplegado en breves instantes. En la captura superior se aprecia como se pasa de un entorno con 0 contenedores y pods en ejecución, a todo el stack desplegado con solo un simple comando:

### **podman kube play —network red deseada /ruta/al/archivo.yml**

Es un comando muy sencillo con la siguiente composición:

- **Podman kube:** Invocación de la herramienta kube, basada en Kubernetes
- **Play:** Indica que se va a arrancar un pod
- **—network:** necesario para indicar a que red de podman conectaremos el pod. Si no se indica, podman kube conectara el pod a la red predeterminada
- **Ruta:** Auto explicativo, se especifica la ruta del archivo yml que podman kube usara como información para desplegar el pod.

Si la sintaxis es correcta, podman kube descargara automáticamente las imágenes necesarias (si no están presentes ya en el sistema) y desplegará el pod con todos los contenedores y sus configuraciones definidas en unos instantes.

Como se puede comprobar, esta forma de realizar los despliegues es mucho más cómoda y rápida que la ejecución directa por comandos individuales.

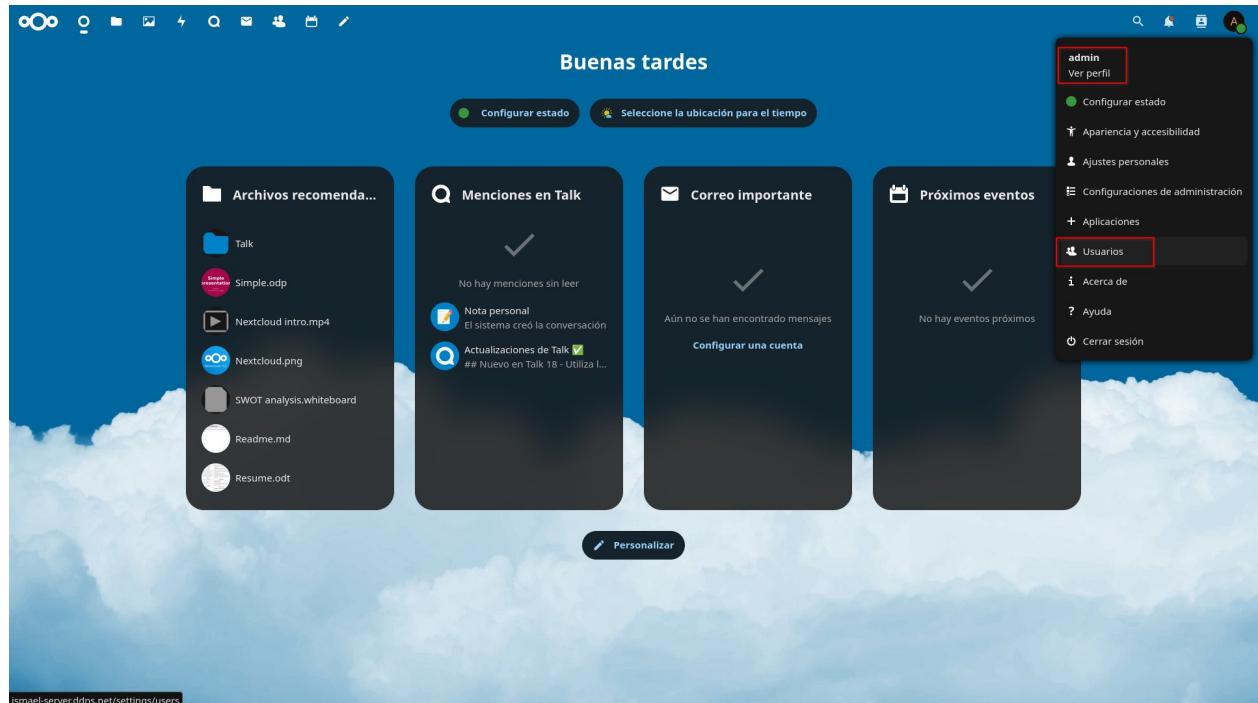
Es la que usaremos a partir de este momento para los dos stacks restantes.

## 9.1.7 Stack Nextcloud. Anexo I: Creación de usuarios en la plataforma

Si bien este manual no pretende ser una guía acerca del uso de las aplicaciones que este proyecto muestra, sí que haremos hincapié en secciones futuras a cerca del uso de un servicio de directorio OpenLDAP para permitir los accesos a la aplicación.

Es por tanto ilustrativo mostrar el proceso que pretendemos ahorrarnos con el despliegue del siguiente stack.

En las próximas líneas se hace una breve explicación del proceso de gestión de usuarios y grupos en nextcloud en un entorno aislado sin servicio de directorio.



Para empezar, debemos loguearnos en nextcloud como el administrador de la aplicación, o en su defecto un usuario al que el mismo le haya otorgado permisos de administrador.

En el botón de gestión de sesiones de la esquina superior derecha, debemos acceder al apartado **Usuarios**.

The screenshot shows the Nextcloud administration interface for users. At the top, there's a header with tabs like 'Nuevo usuario', 'Nombre para mostrar', 'Contraseña', 'Correo electrónico', 'Grupos', 'Espacio asignado', and 'Administrador'. Below the header, there's a table with columns for 'Usuarios activos' (1), 'Grupos' (1), and 'Espacio asignado' (Ilimitado). A red box highlights the 'Nuevo usuario' button in the top-left corner. Another red box highlights the 'admin' row in the table. A third red box highlights the 'Grupos' column. A red arrow points from the text 'PODEMOS AÑADIR USUARIOS O GRUPOS DESDE SENDAS OPCIONES EN LA COLUMNÁ IZQUIERDA' to the 'Nuevo usuario' button.

Este es el panel de gestión de usuarios de Nextcloud. Muestra información sobre los usuarios presentes en la plataforma, así como su cuota de almacenamiento utilizado, información sobre los grupos de los que forma parte etc.

En la captura se puede comprobar que solo el usuario administrador creado durante la instalación, está presente en el sistema.

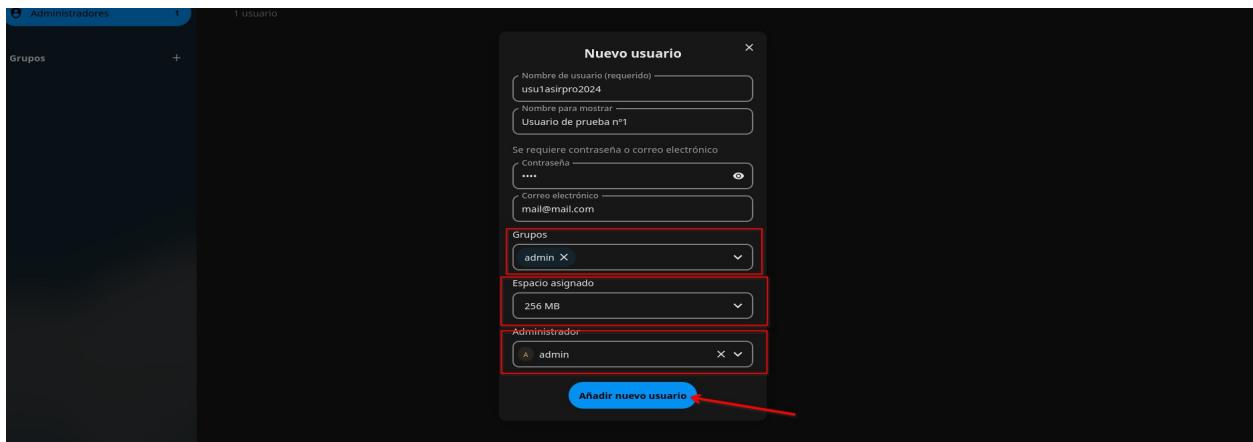
Para añadir un usuario, debemos seleccionar la opción homónima en la columna izquierda.

The screenshot shows the 'Nuevo usuario' dialog box. It contains the following fields:

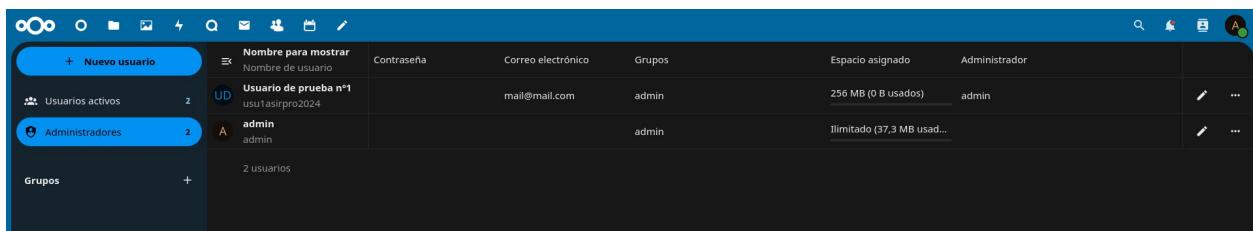
- Nombre de usuario (requerido): usu1asirpro2024
- Nombre para mostrar: Usuario de prueba nº1
- Contraseña: (redacted)
- Correo electrónico: mail@mail.com
- Grupos: Establecer grupos de usuario (dropdown menu)
- Espacio asignado: Cuota predeterminada (dropdown menu)
- Administrador: Establecer administrador de usuario (dropdown menu)

A blue button at the bottom right says 'Añadir nuevo usuario'.

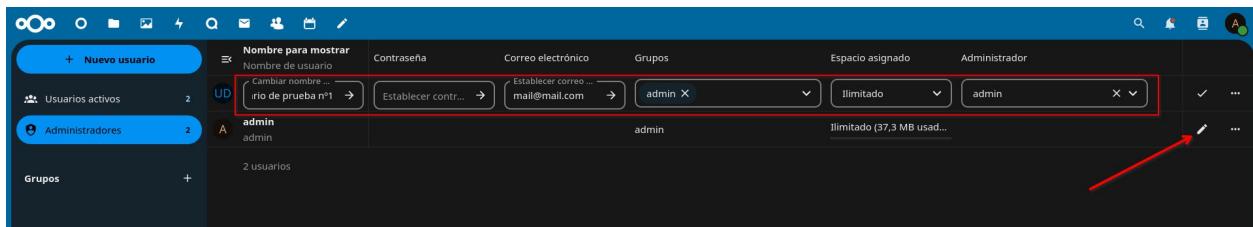
En el cuadro de diálogo cumplimentaríamos la información del usuario tal y como se muestra en la captura.



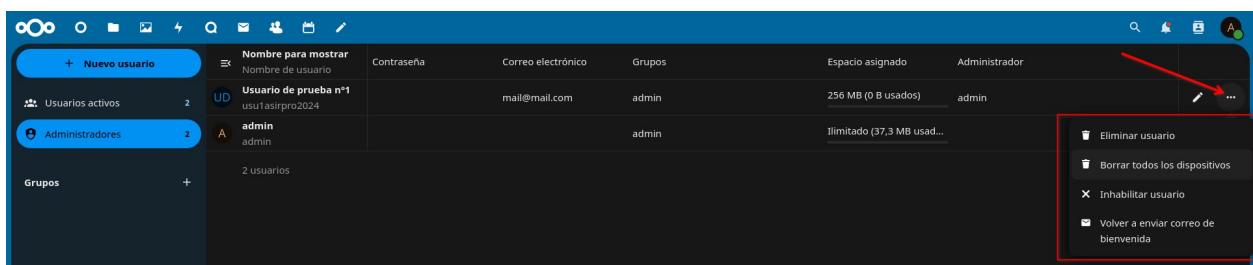
Podemos, si así lo deseamos asignar grupos al nuevo usuario, limitar su cuota de almacenamiento en el servidor, o delegar en algún usuario administrador la gestión de dicho usuario. Cuando hayamos establecido los parámetros deseados, pulsamos sobre el botón para añadir al nuevo usuario.



Como vemos, el usuario ya está presente en nextcloud y ya podría loguearse.

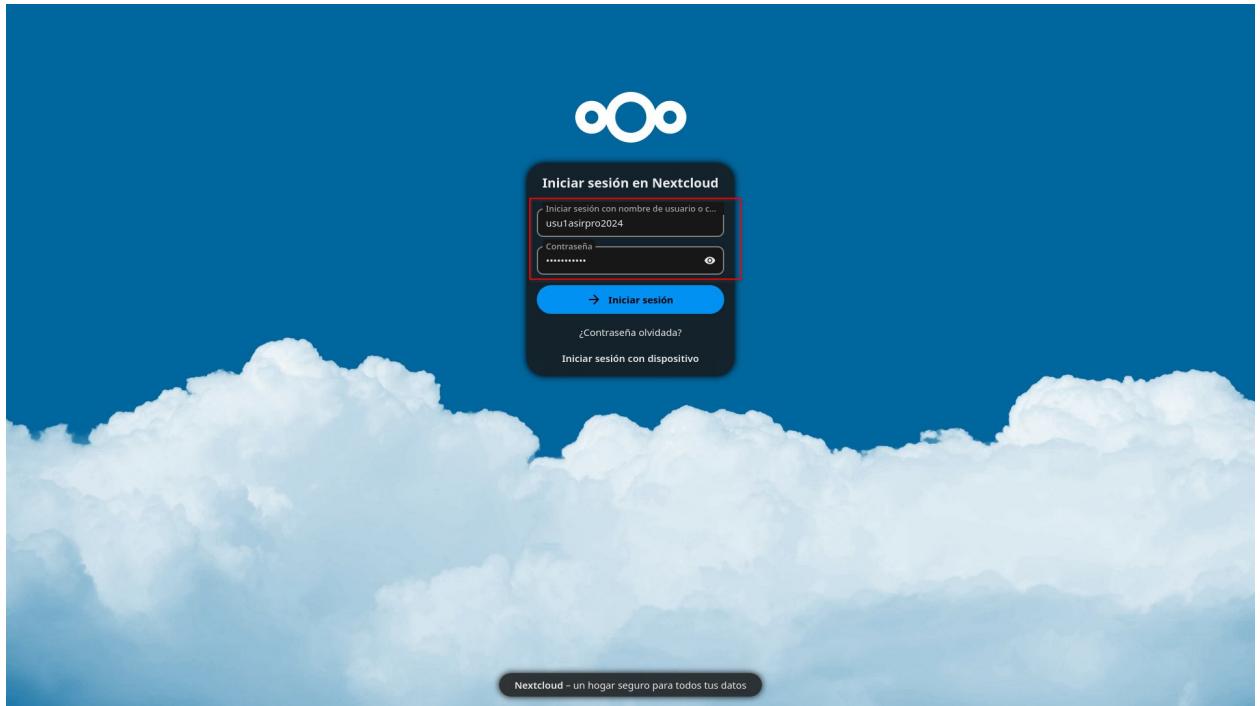


El ícono del lápiz nos permite como administradores, editar la información de los usuarios en todo momento.

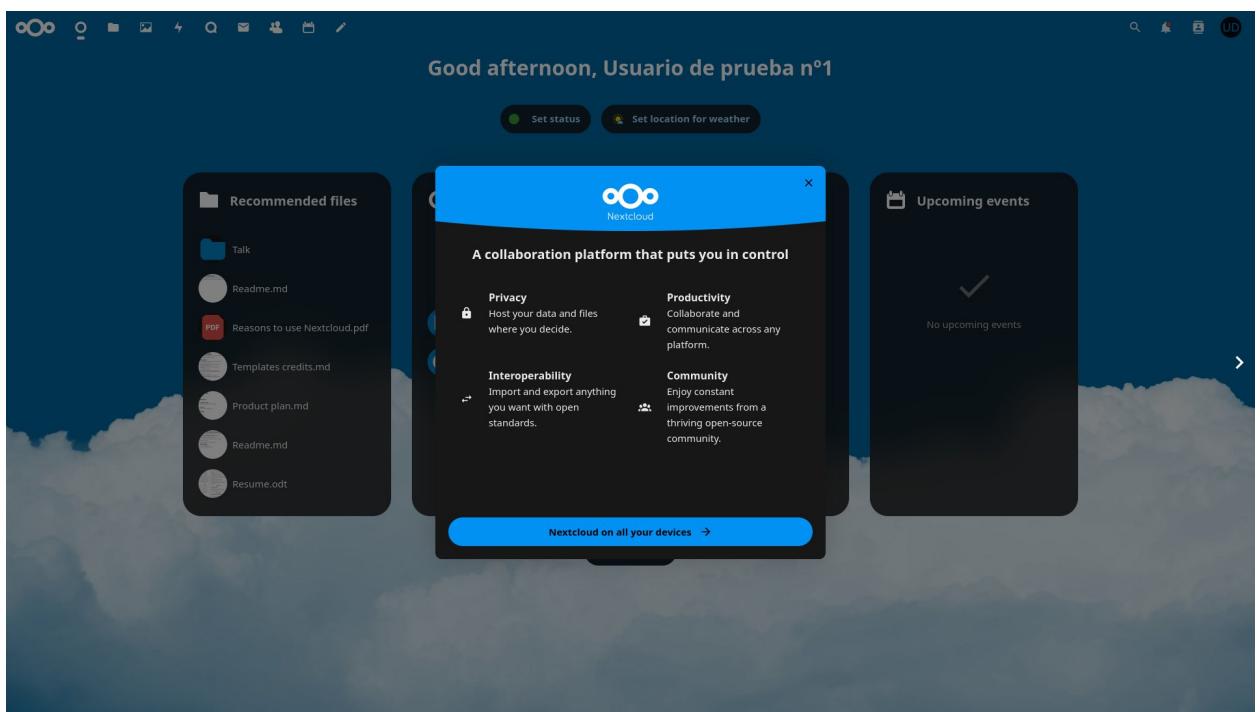


Y el botón de “más opciones” nos permitirá borrar, inhabilitar, o desloguear al usuario de todos los dispositivos que tenga sincronizados. También nos permite reenviarle el correo de bienvenida en caso de que tengamos a nextcloud asociado a nuestro propio servidor de correo. Desgraciadamente debido a la imposibilidad de controlar registros PTR, este proyecto no cubrirá el despliegue de un servidor de correo.

Pero probemos al usuario recién creado.



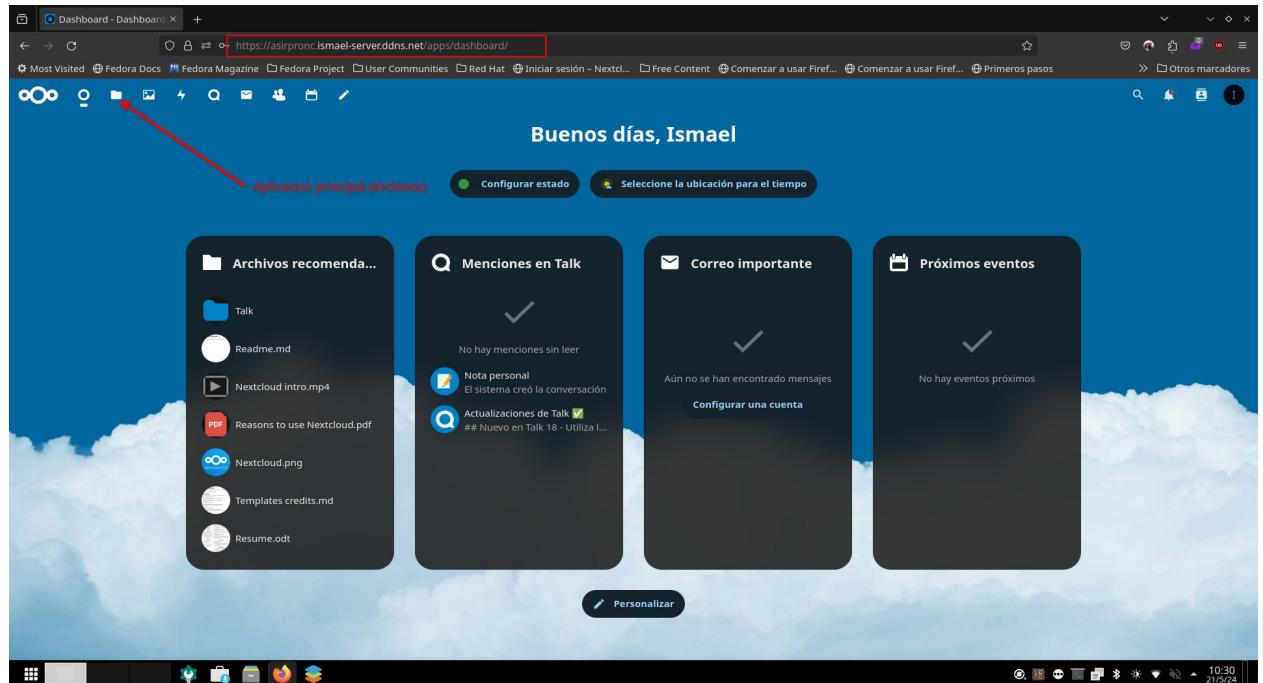
Si salimos de la sesión de administrador e introducimos las credenciales del usuario que hemos creado...



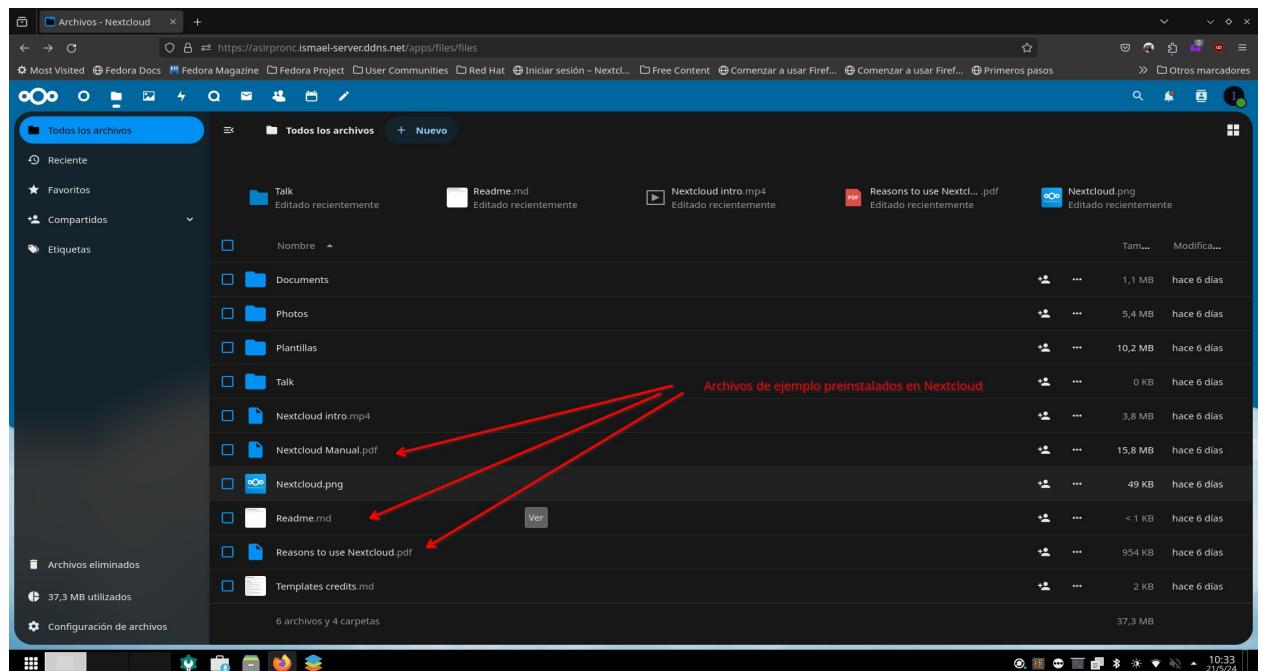
Nextcloud recibe con la bienvenida al nuevo usuario a la plataforma, y este puede empezar a utilizar el servicio.

## 9.1.8 Stack Nextcloud. Anexo II: Sincronización de archivos con cliente de escritorio

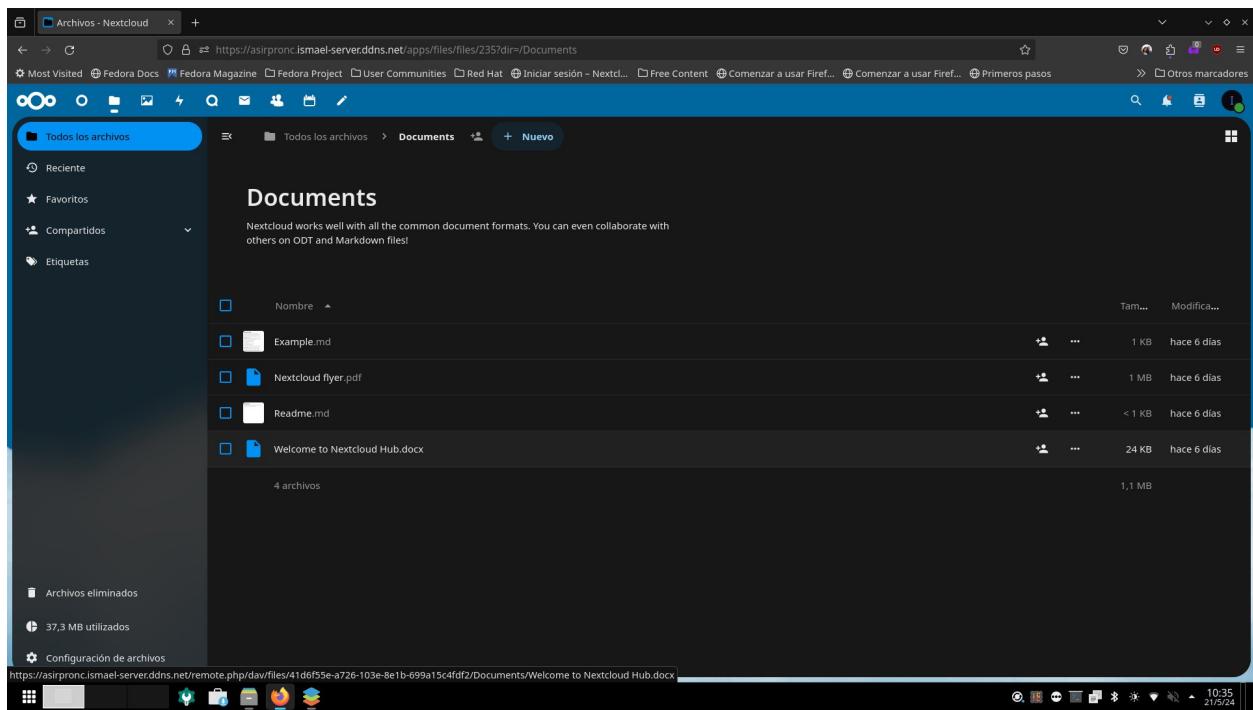
En este anexo, se mostrará en unas breves líneas el uso de la más útil funcionalidad de nuestra aplicación principal: la sincronización de archivos con el cliente de escritorio de nextcloud.



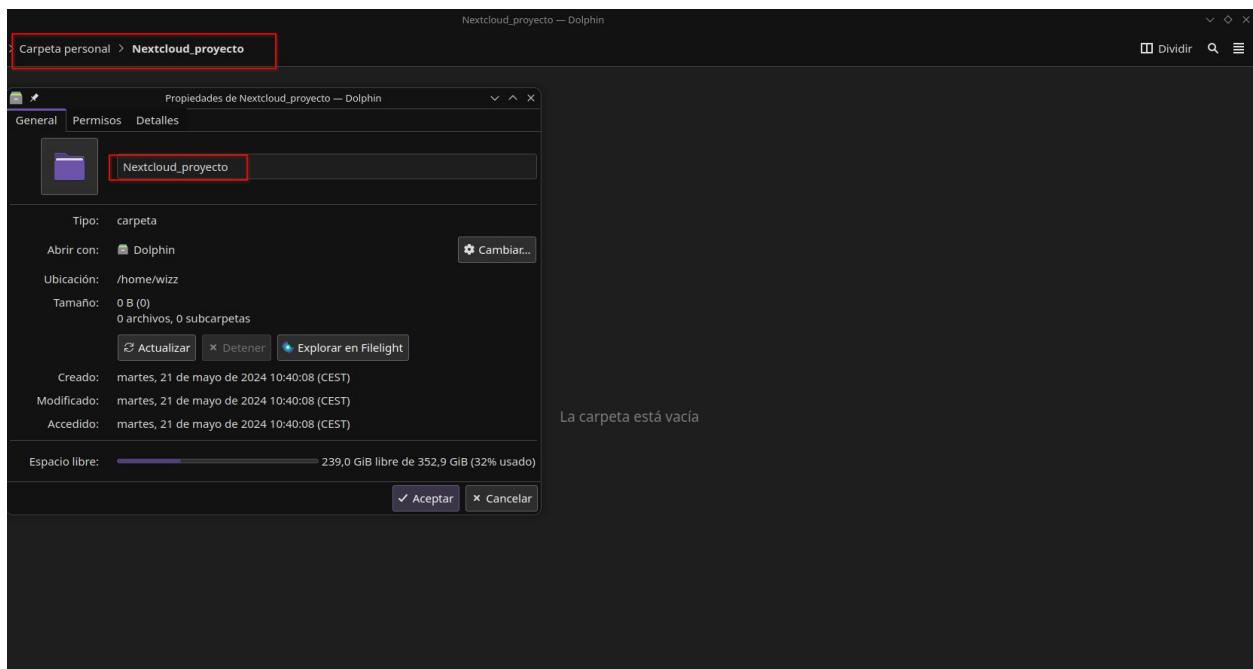
Comenzaremos logueados como uno de los usuarios del directorio LDAP con permiso para loguearse en Nextcloud (El apartado de configuración y despliegue LDAP se cubrirá en un posterior apartado). Nos iremos a la aplicación principal de Nextcloud: Archivos.



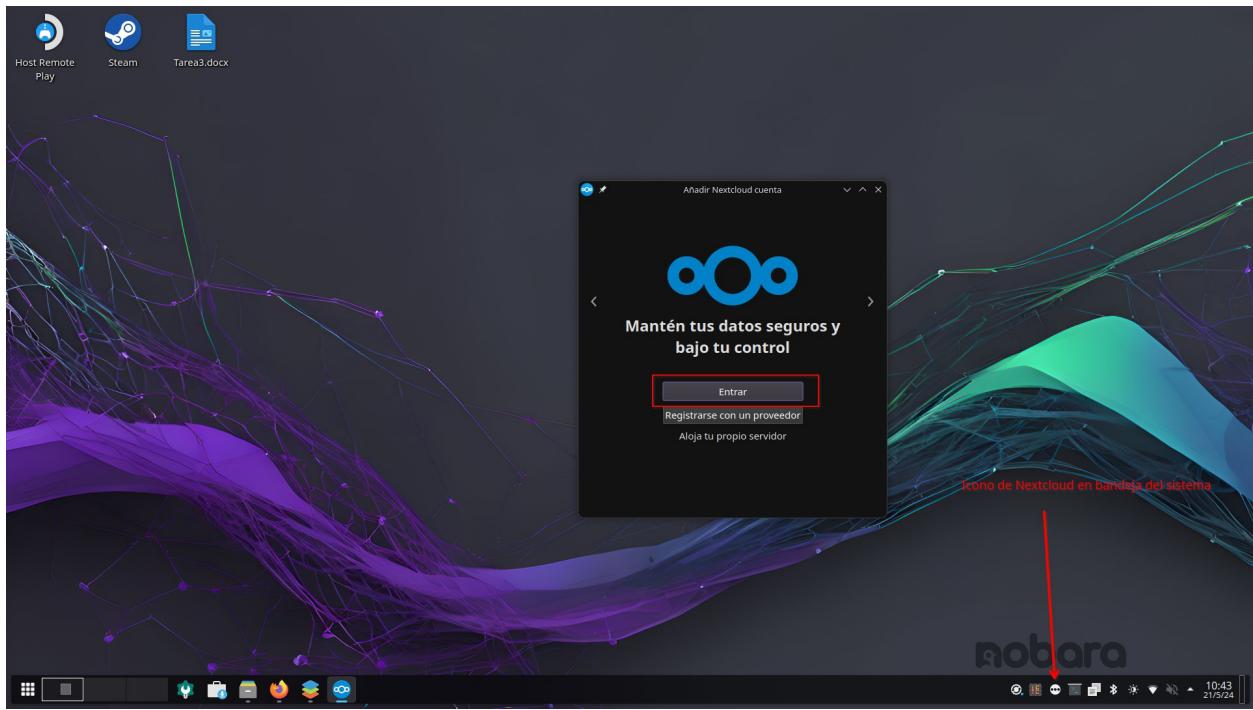
Como vemos, Nextcloud incorpora una serie de archivos de muestra en cada nueva instalación, así como como los directorios típicos de un entorno de trabajo, como Documentos y Fotos.



Dichos directorios también están poblados con archivos de muestra, como por ejemplo el directorio Documents. Procederemos a sincronizar el servidor con el cliente y haremos varias pruebas con algunos archivos.

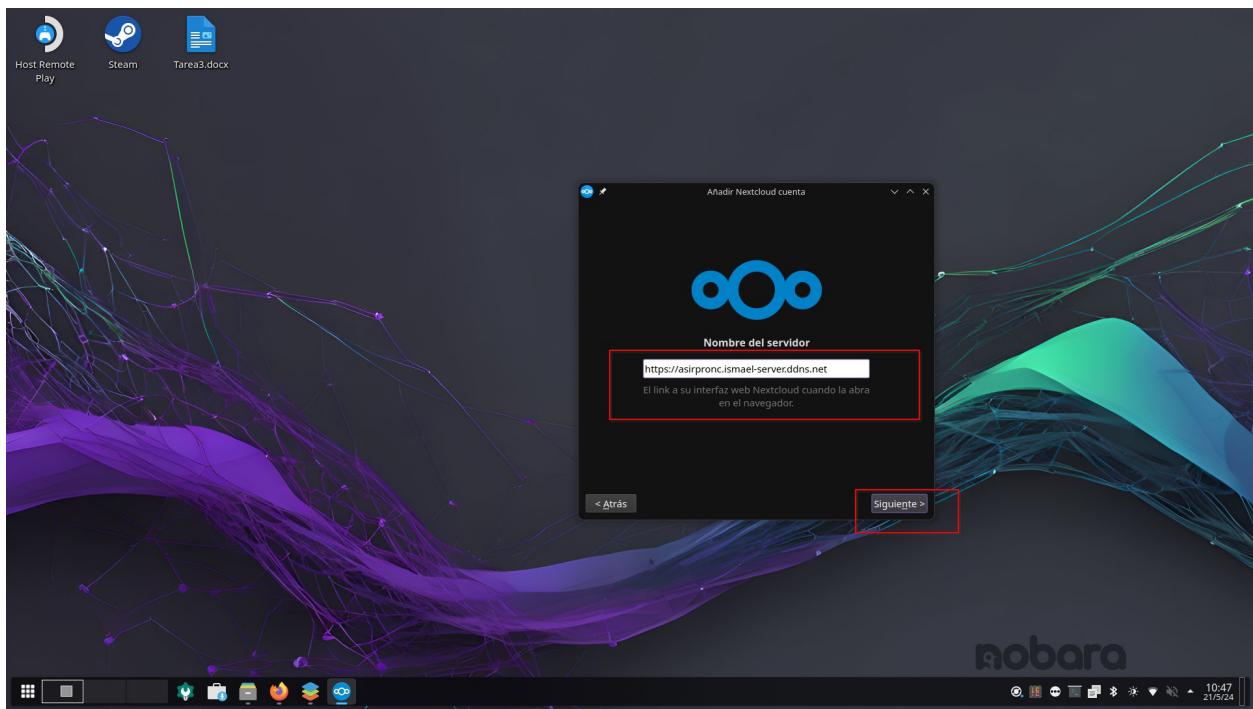


El primer paso es dedicar un directorio para la sincronización de Nextcloud. Este directorio puede tener cualquier nombre y estar ubicado en cualquier ruta accesible por el usuario que pretende sincronizar contenido. En nuestro caso, crearemos un directorio llamado "Nextcloud\_proyecto". Se puede apreciar que el directorio está vacío en la captura superior.

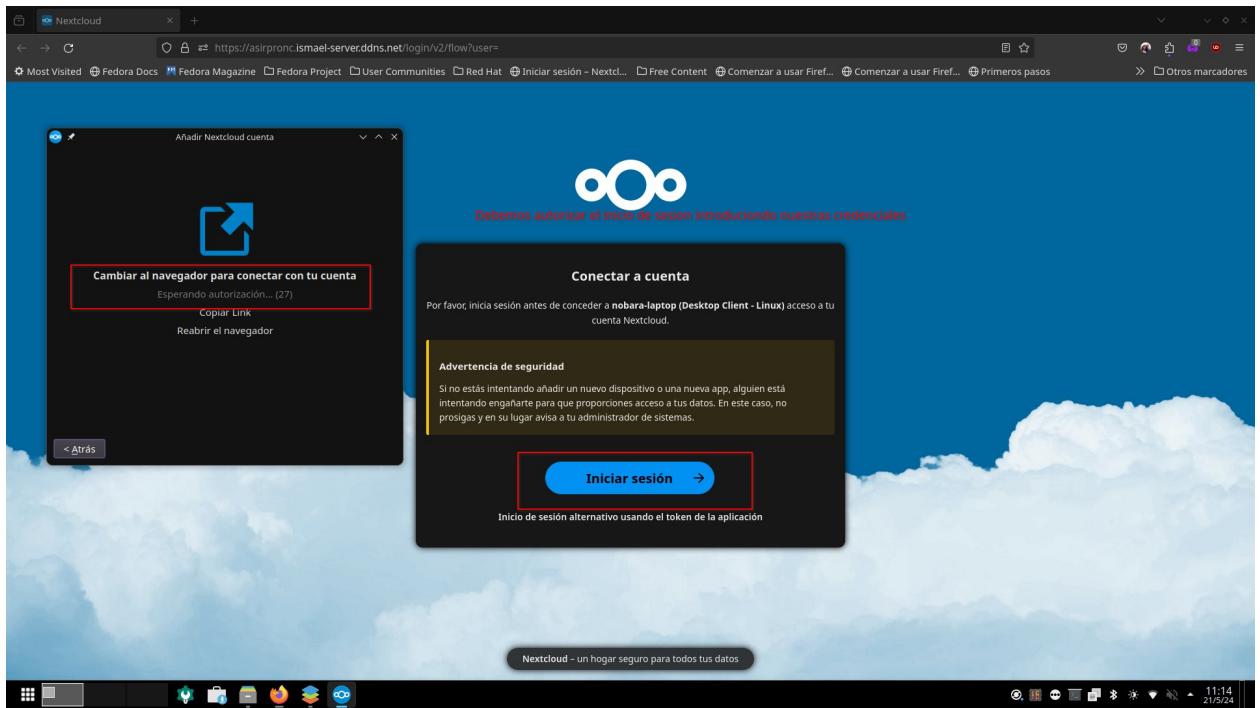


Debemos descargar la aplicación de “Cliente de escritorio Nextcloud”. Dicha aplicación tiene distribuciones para los principales sistemas operativos, ya sea Windows, Linux, MacOS, IOS y Android. La aplicación se distribuye como típico paquete autoinstalable en Windows, y como paquetes Flathub o AppImage para Linux, siendo esta la versión que usaremos en este apartado.

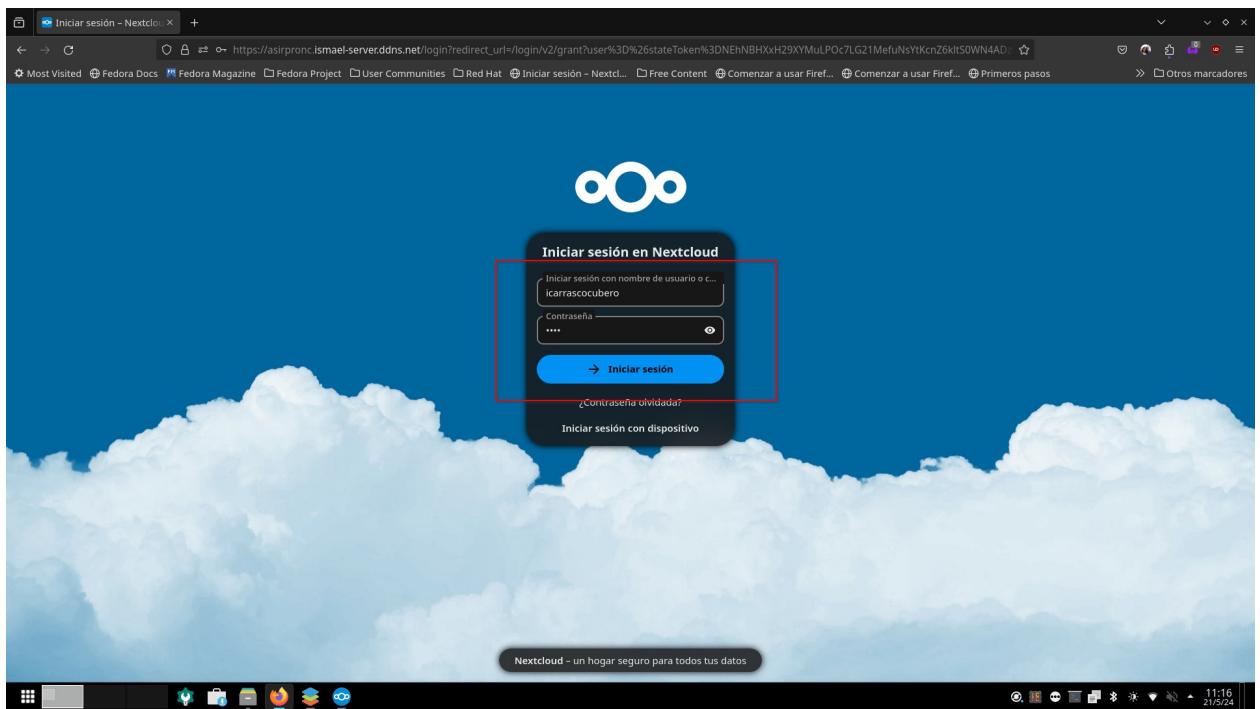
Una vez instalada y ejecutada, pulsaremos en el botón “Entrar” para loguearnos en nuestro servidor Nextcloud.



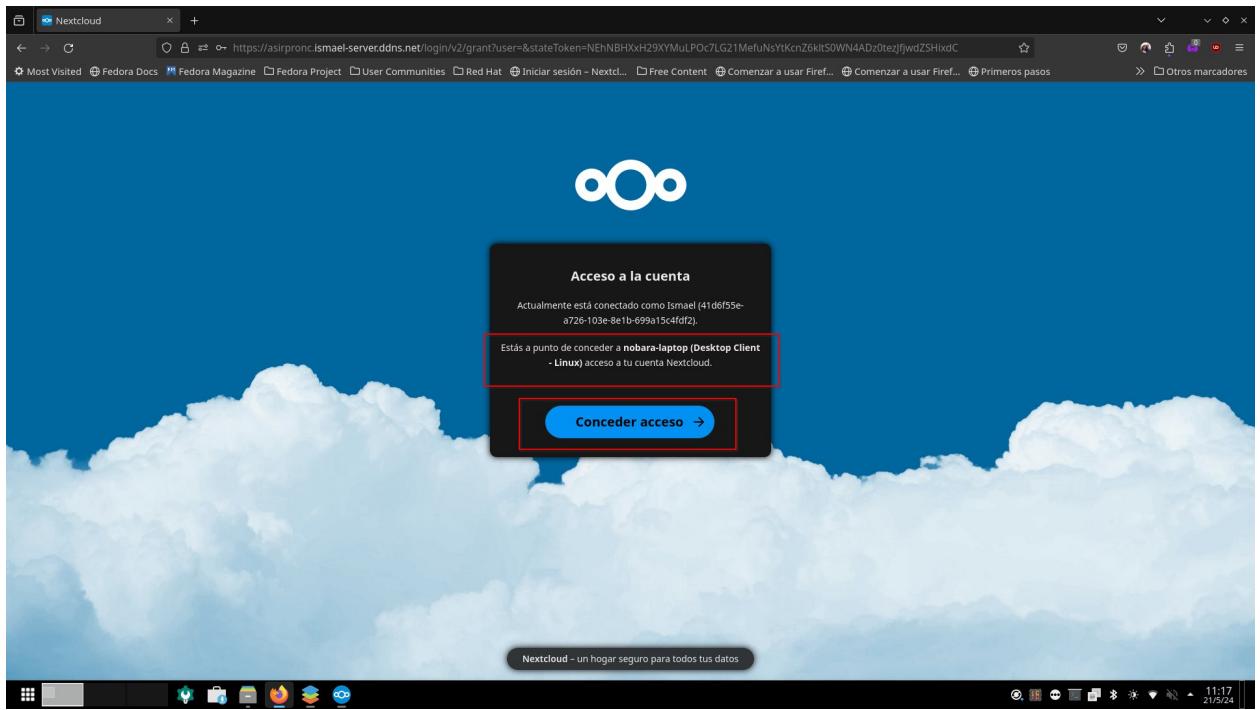
En nuestro caso, usaremos la URL del nombre de dominio de nuestro servidor Nextcloud, aunque también es válida la dirección IP y el puerto de escucha si pretendemos conectar solo a través de red local.



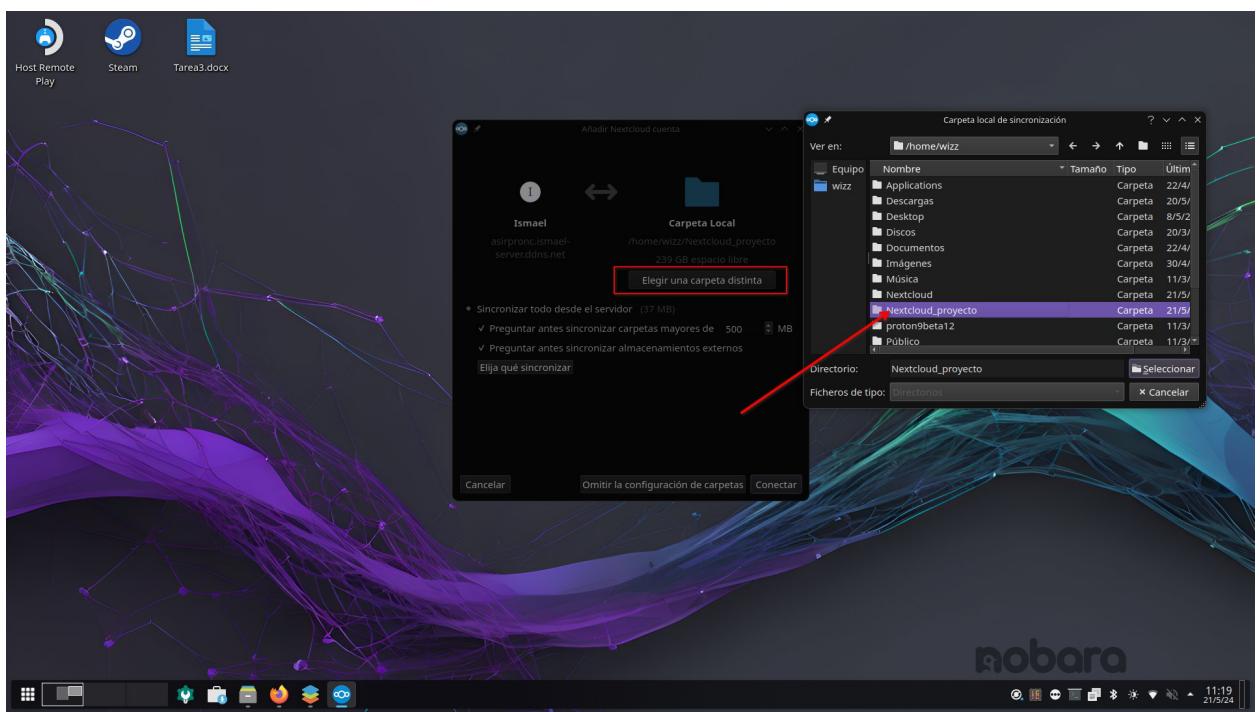
El cliente de escritorio nos abrirá en nuestro navegador la web de Nuestra instancia de Nextcloud para autorizar el login, debemos pulsar sobre iniciar sesión.



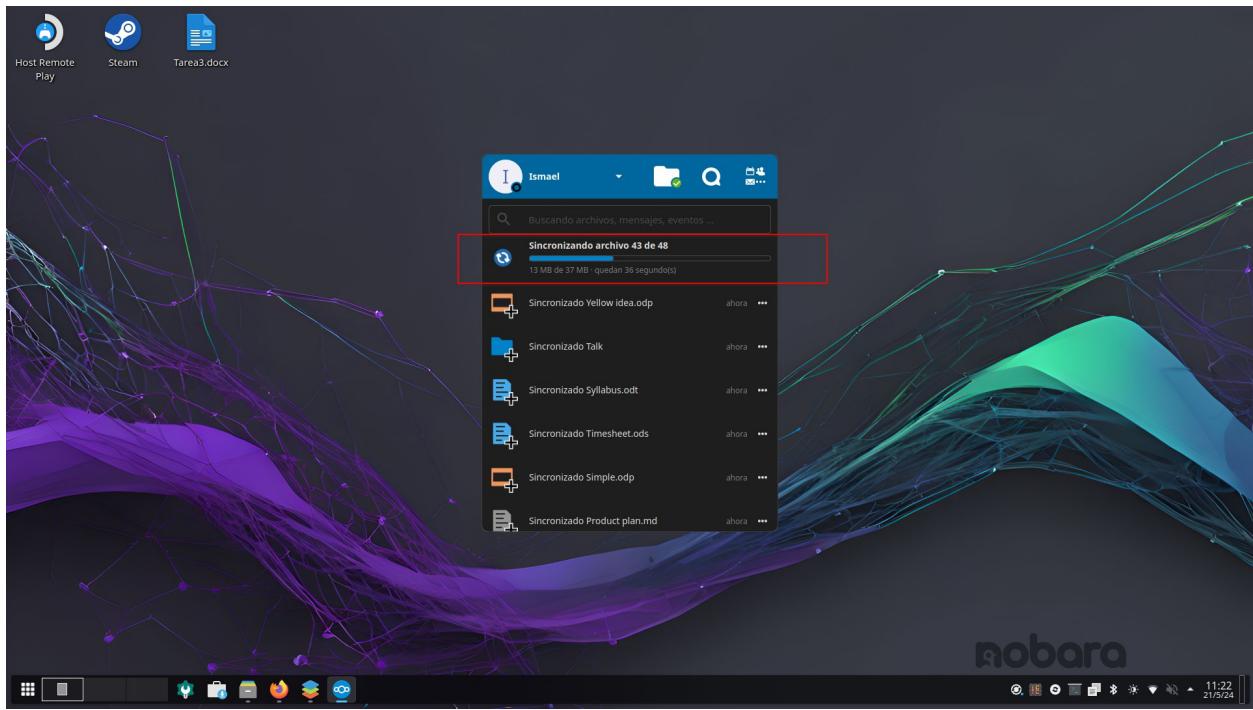
Introducimos nuestras credenciales exactamente igual que si fuéramos a entrar en la aplicación web.



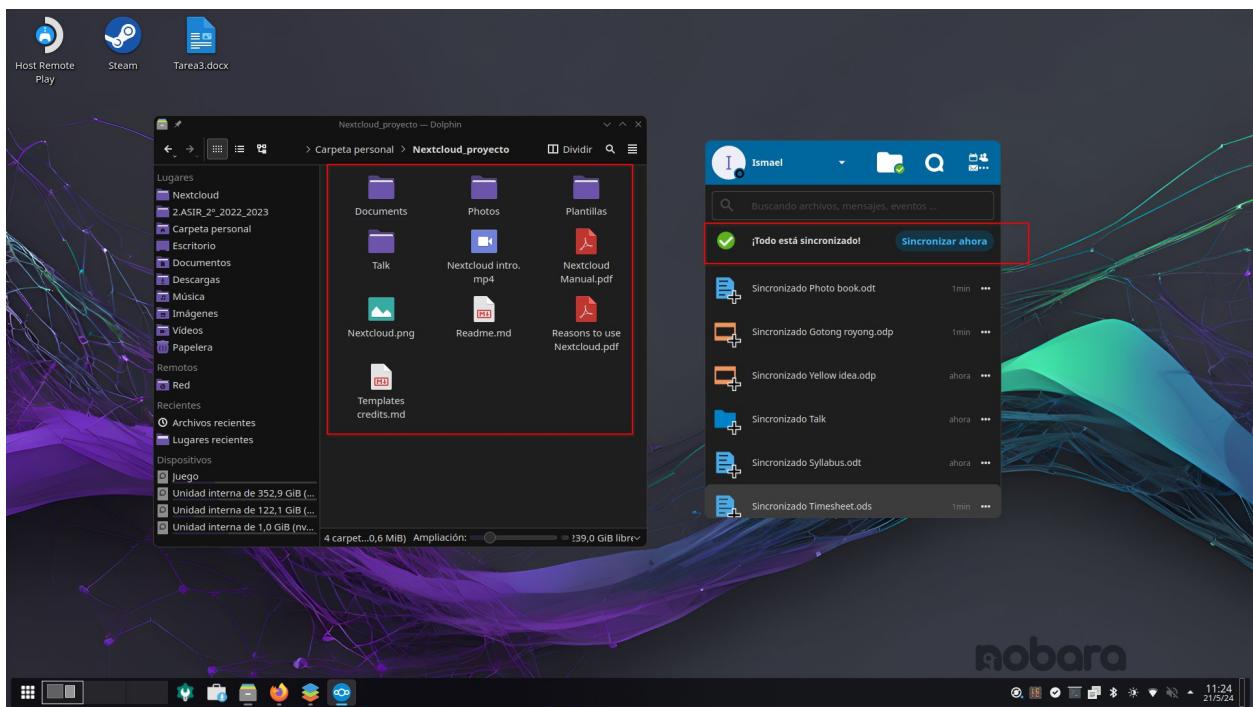
Una vez logueados, ya solo nos queda autorizar el acceso al cliente con el botón “Conceder acceso” para terminar el proceso.



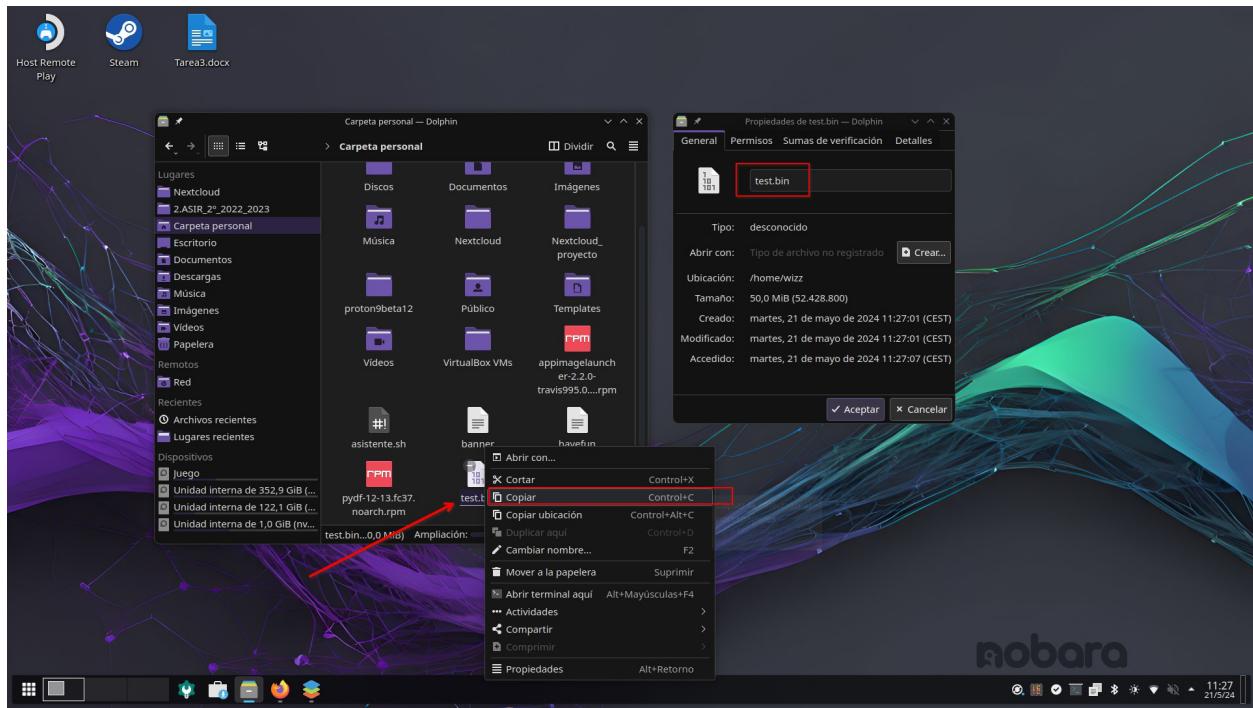
El cliente nos abre un breve panel de configuración en el que se nos permite elegir la ubicación en la que queremos seleccionar los archivos usando un explorador de directorios, así como otras opciones de sincronización. Una vez elegida la ubicación pulsamos en “Conectar”.



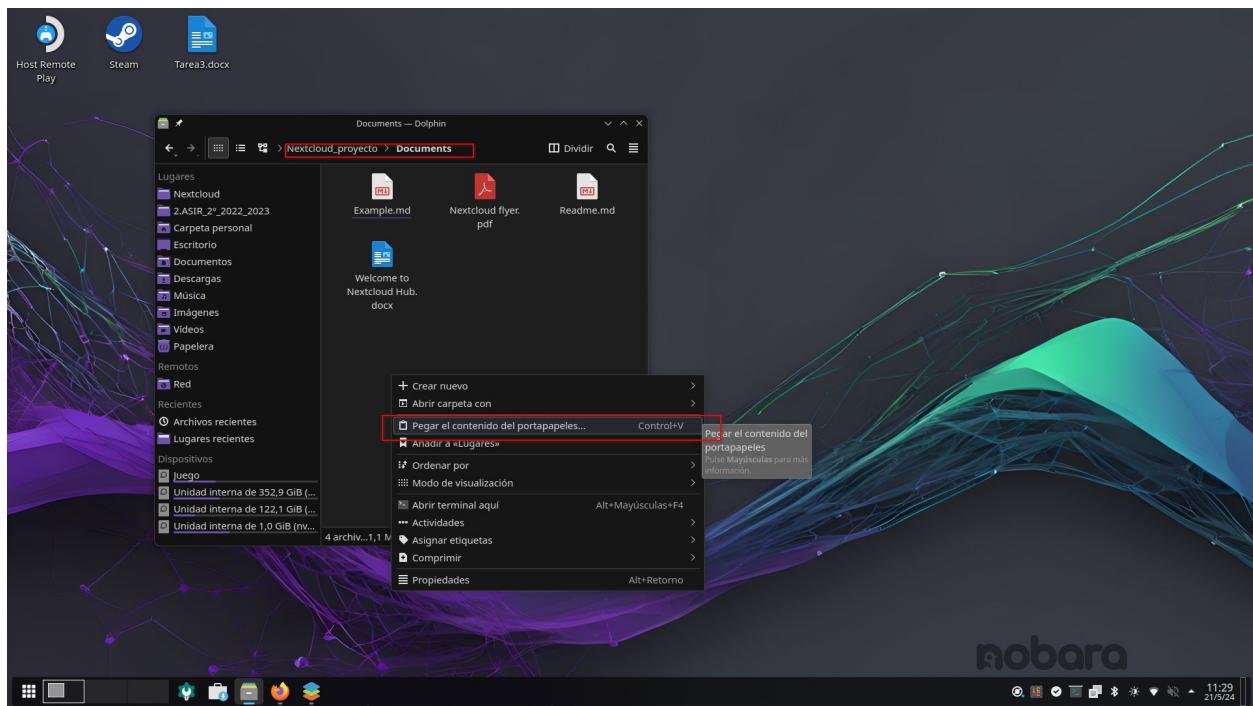
En cuanto todo está configurado, el cliente comenzara a sincronizar todos los archivos entre el directorio local y el directorio del servidor, descargando o subiendo según el caso, como se puede apreciar en la descarga de los archivos de ejemplo del servidor en la captura superior.



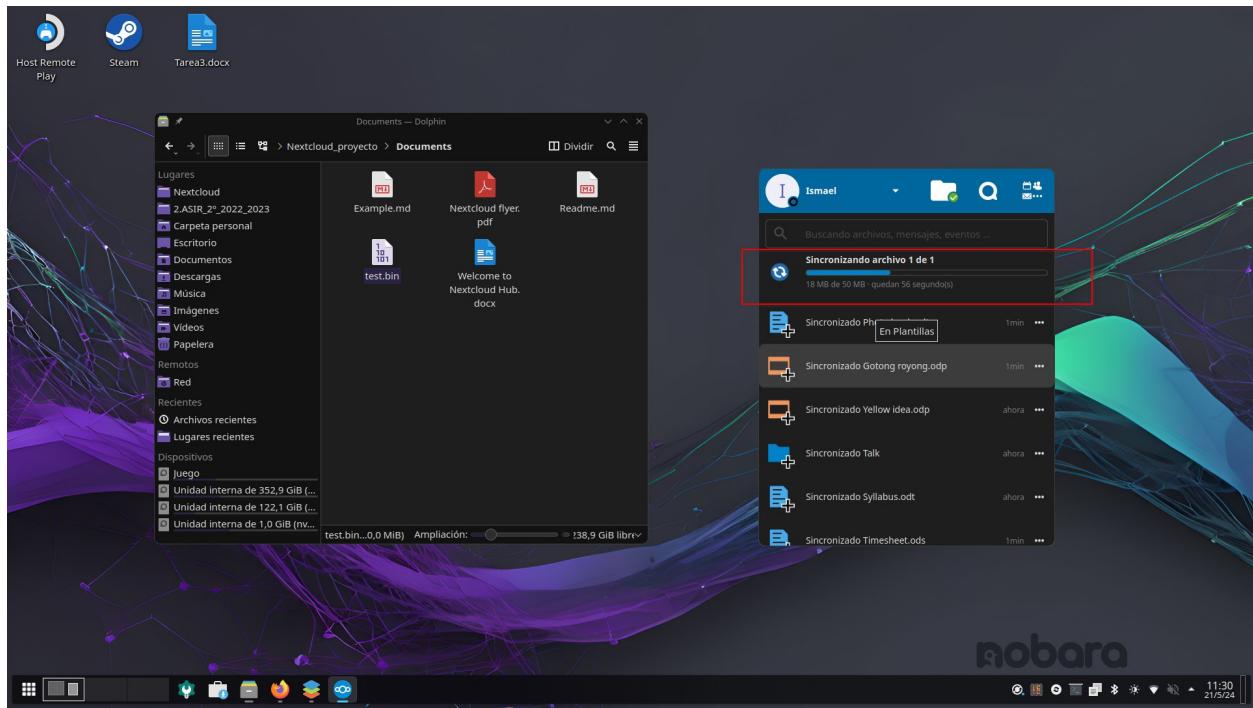
La sincronización se ha completado. Probemos ahora con algunos archivos propios en ambas direcciones.



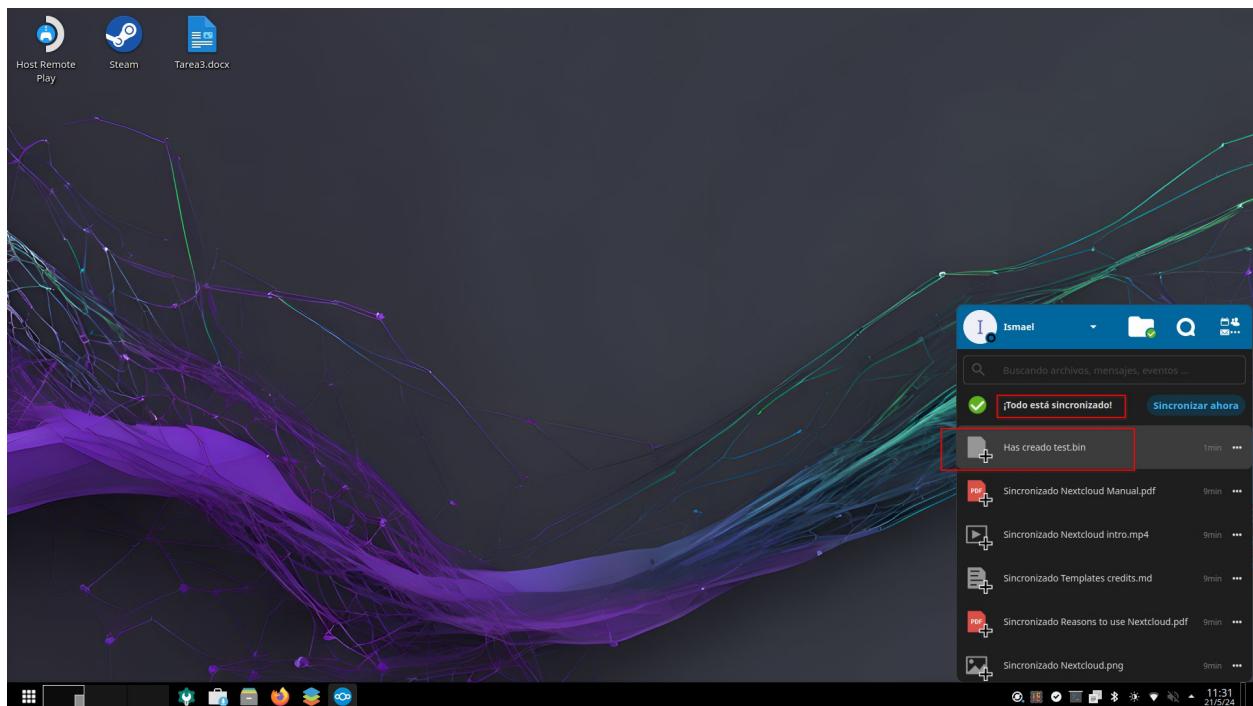
Probemos a copiar un archivo de 50MB llamado test.bin a nuestro directorio de nextcloud.



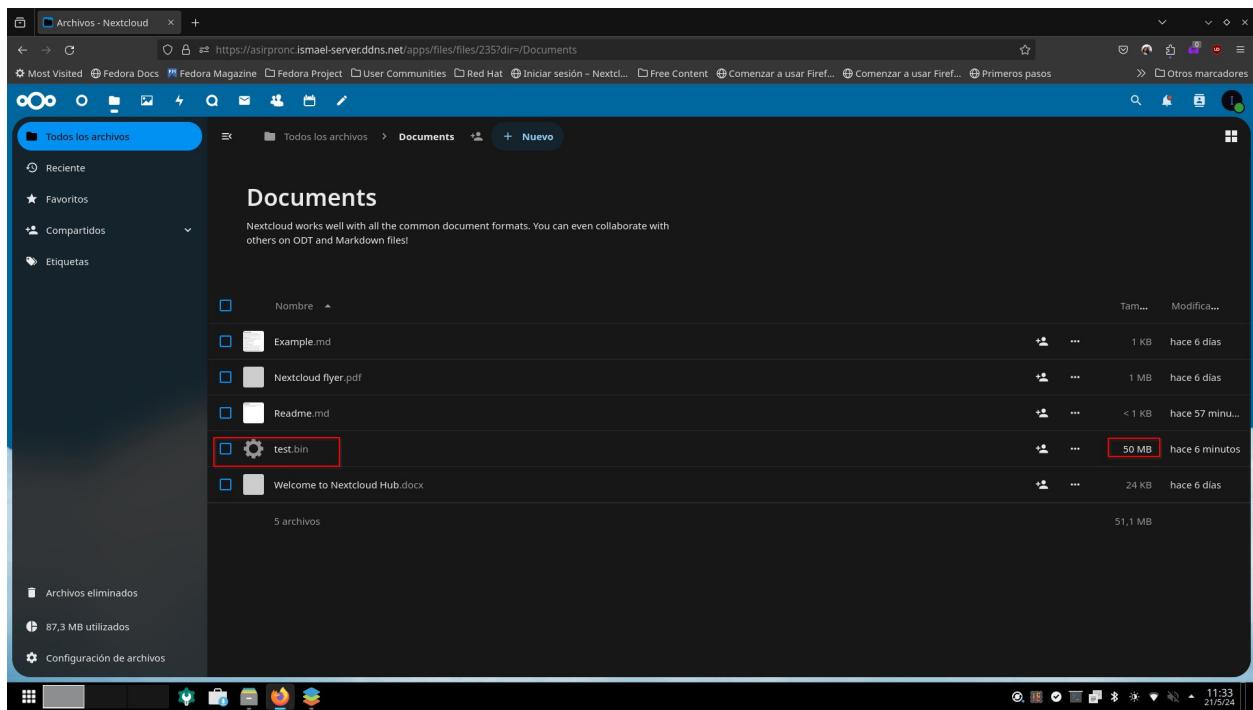
Si pegamos la copia en el directorio de documentos de nuestro directorio sincronizado con el servidor...



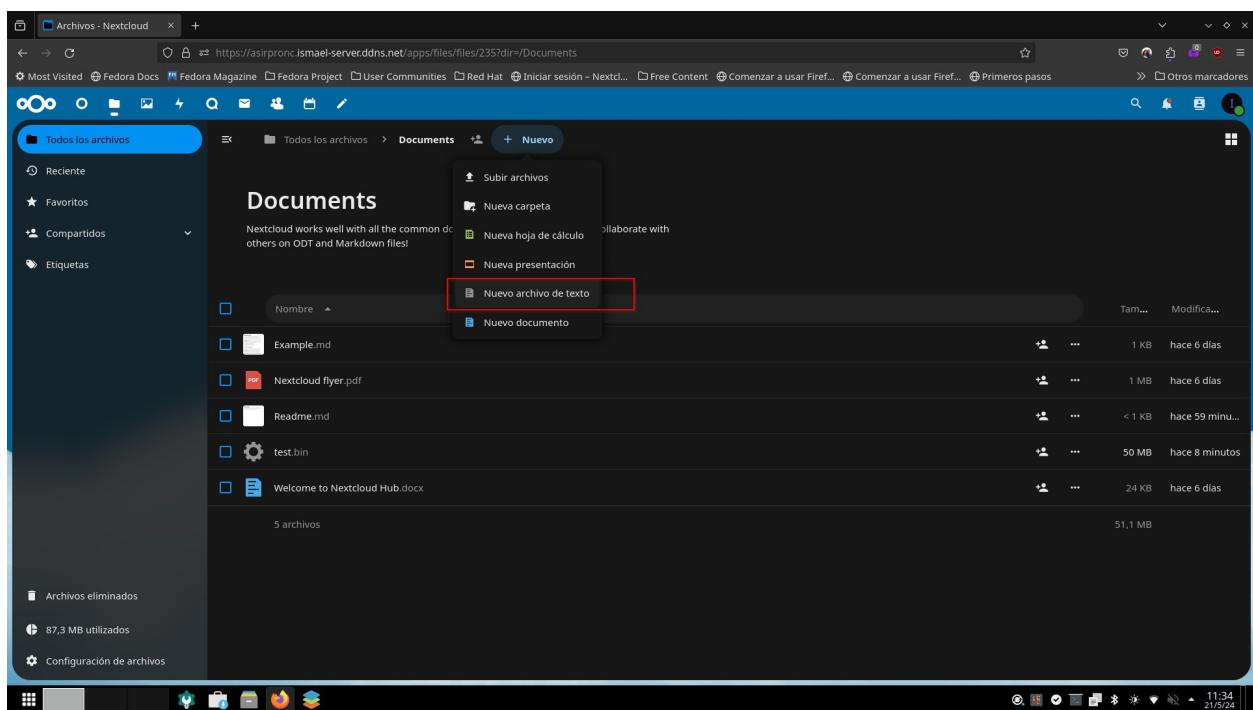
La sincronización comienza en unos instantes.



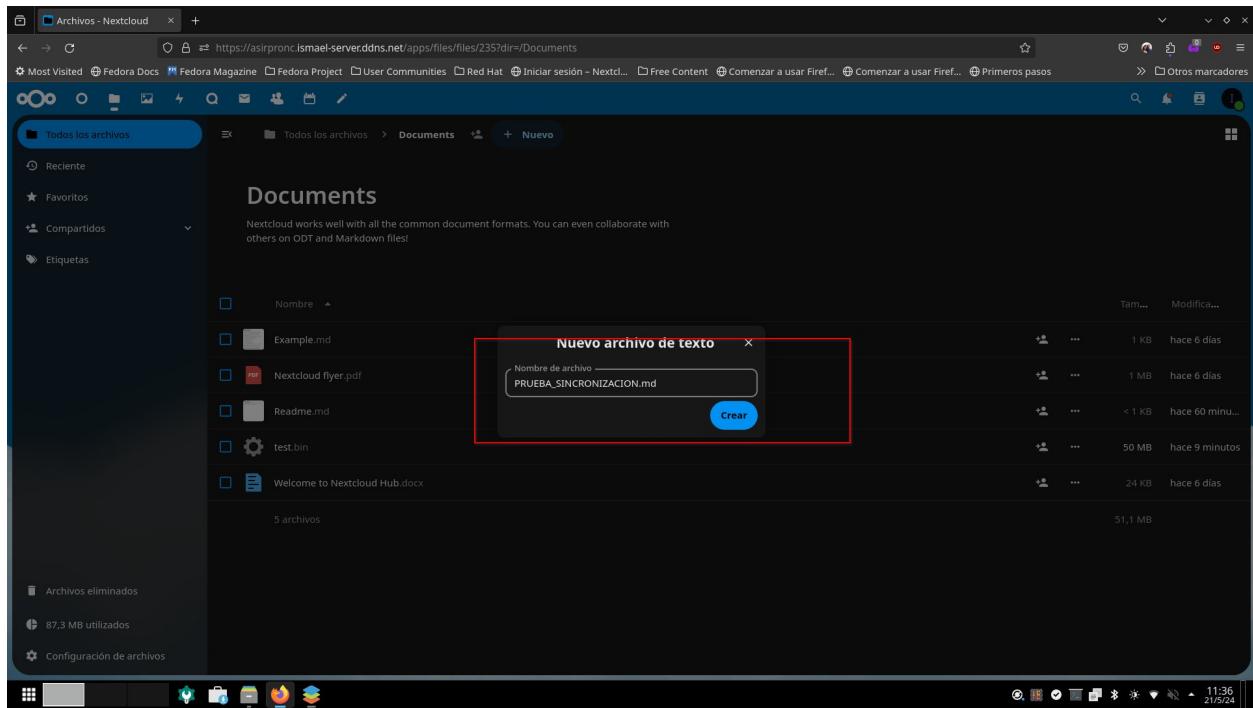
La bandeja del cliente de sincronización nos informa de que el archivo se ha sincronizado con éxito.



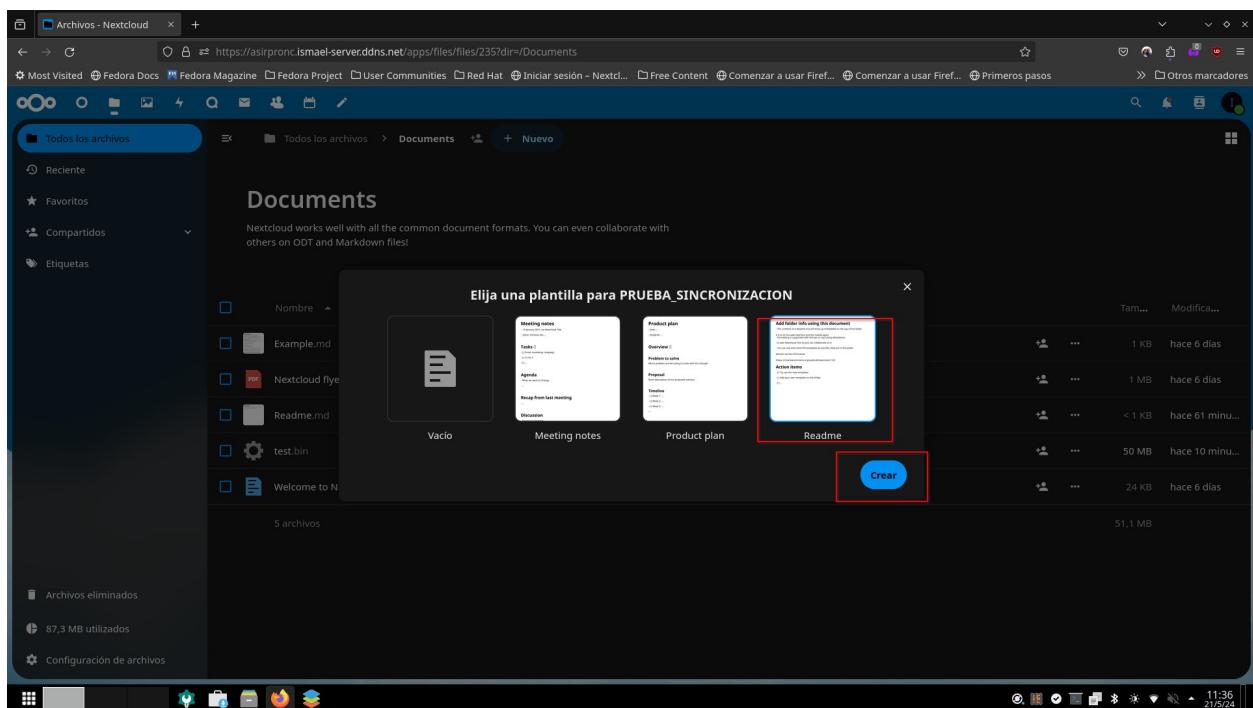
Si navegamos en la app web al directorio de documentos, confirmamos la presencia del test.bin de 50MB.



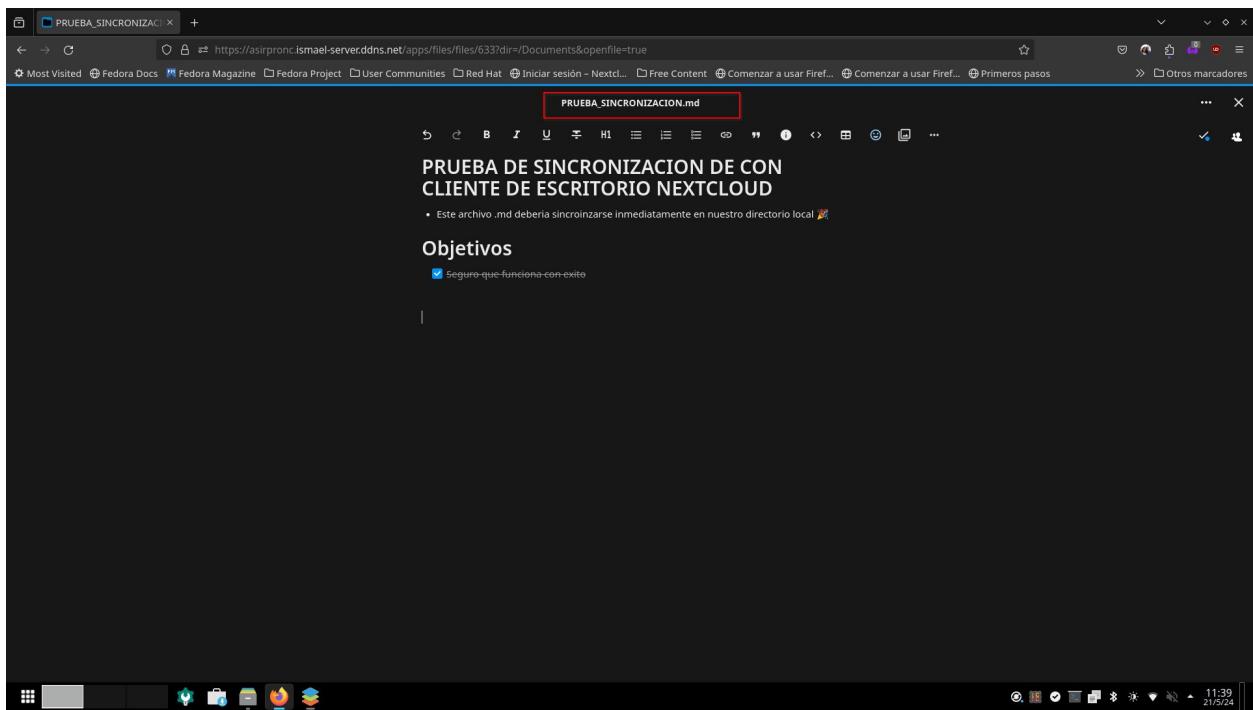
Probemos el proceso opuesto creando un nuevo archivo de texto en el directorio de documentos con el editor de texto integrado.



Le damos nombre...



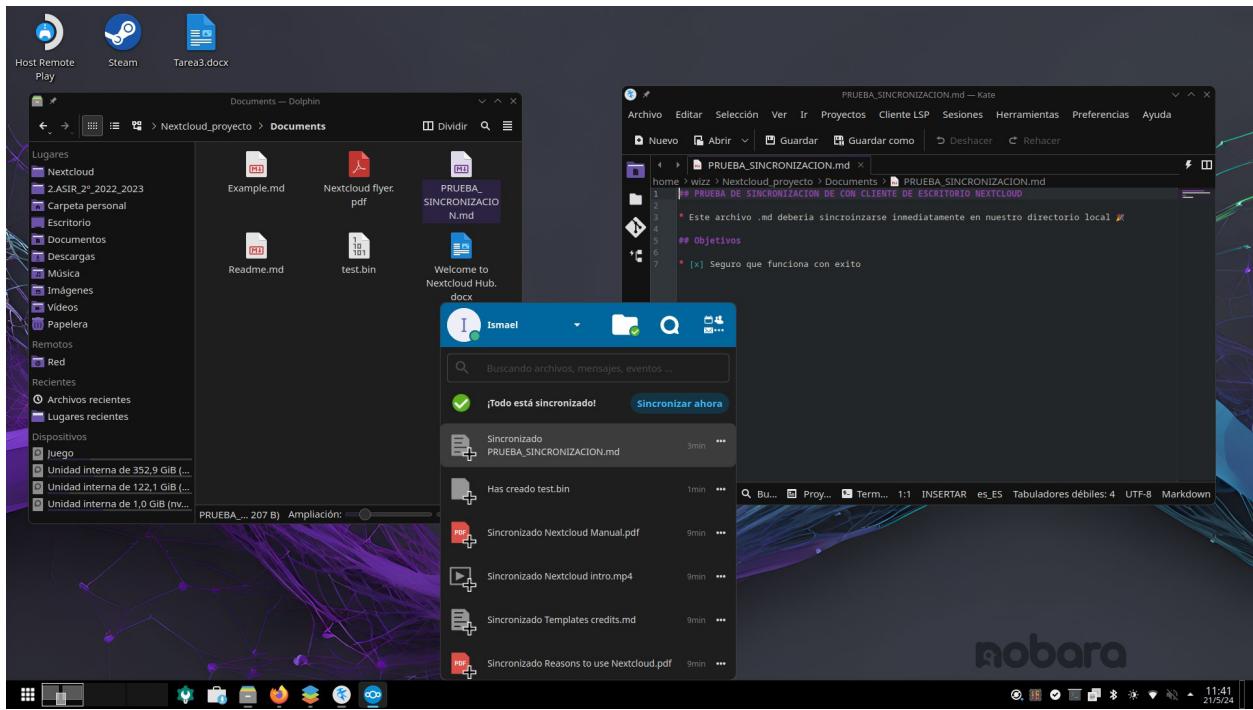
Ya que Nextcloud nos ofrece amablemente el uso de una plantilla, escojamos una para probar. Pulsamos sobre crear.



Ponemos algo de contenido cualquiera. Nextcloud guarda automáticamente todos los cambios en sus editores integrados.

Nombre	Tam...	Modifica...
Example.md	1 KB	hace 6 días
Nextcloud flyer.pdf	1 MB	hace 6 días
PRUEBA_SINCRONIZACION.md	< 1 KB	hace 3 minutos
Readme.md	< 1 KB	hace 64 min...
test.bin	50 MB	hace 13 min...
Welcome to Nextcloud Hub.docx	24 KB	hace 6 días

Una vez creado y guardado el archivo...



La sincronización se lleva a cabo inmediatamente.

Como vemos, Nextcloud es una aplicación de productividad, especializada en el almacenamiento de archivos increíblemente útil. Y todo alojado y mantenido por nosotros, si necesidad de depender de servicios externos siempre que dispongamos de nuestro propio hardware.

## 9.2 Stack LDAP

Ya tenemos en funcionamiento nuestra aplicación principal, y podemos gestionar su base de datos mediante una potente consola de gestión de PostgreSQL. Con este simple despliegue ya tenemos una aplicación extremadamente útil para gestionar nuestros archivos, compartir documentos, gestionar nuestro calendario entre otras muchas posibilidades, pero... Imaginemos por un momento que estamos en un entorno empresarial, y que dicha empresa tiene un número muy elevado de empleados que gestionar.

En un entorno de ese tipo, el tener que gestionar las cuentas, información y credenciales de los usuarios para cada aplicación o servicio puede ser un auténtico infierno, con infinidad de redundancias y en el peor de los casos inconsistencias.

Siempre es mucho más útil disponer de todo nuestro entorno de usuarios centralizado en una base de datos única, que permita los accesos a cualquier aplicación o servicio compatible con dicha base de datos.

Este sencillo pod se encargará de resolver esa problemática, desplegando una instancia del servicio de directorio OpenLDAP y su panel de gestión LDAP Account Manager, ofreciendo un nivel de funcionalidad básica que permite el login centralizado de usuarios a la aplicación, así como permitir o denegar los accesos en función a la pertenencia o no de diferentes grupos.

## 9.2.1 Stack LDAP. OpenLDAP

Comencemos pues describiendo a OpenLDAP, el cual no necesita mucha presentación.

Se trata de un conocido de nuestro ciclo de ASIR; un servicio de directorio de licencia Open source basado en el proyecto LDAP original, basado este a su vez en X500, que nos permite recopilar de forma completamente centralizada la información de los objetos de nuestra organización, sean estos usuarios, equipos, impresoras etc.

Su gestión mediante comandos y archivos .ldif es bastante árida y compleja, razón por la que a este contenedor le acompañara uno adicional para proporcionar un panel gráfico de administración.

Es particularmente útil como base de datos para el almacenamiento de usuarios que permita logins a aplicaciones capaces de contactar con el servidor.

Este último uso será el que le daremos en nuestro servidor.

## 9.2.2 Stack LDAP. LDAP Account Manager (LAM)

El despliegue de un servicio de directorio LDAP nos lleva a la necesidad de desplegar un medio para poder gestionar el directorio de forma intuitiva y agradable, pues la gestión del mismo con herramientas como **ldapadd**, **ldapmodify** o **slapcat** y archivos .ldif es bastante compleja.

LDAP Account Manager es la solución a este problema, ofreciendo una interfaz web no excesivamente bonita, pero si muy simple de entender y usar. Se trata de una aplicación web desarrollada principalmente en PHP que hace uso de un servidor web interno para mostrar la interfaz de usuario.

The screenshot shows the LAM interface for creating a new user. At the top, there's a red error message: 'Apellido' (Last name) is required. Below that are buttons for 'Guardar' (Save), 'Establecer contraseña' (Set password), and 'Regresar a la lista de usuarios' (Return to user list). On the right, there's a placeholder for a profile picture with the text 'Añadir foto'. The main form has sections for 'Nuevo usuario' (New user), 'Sufijo' (Suffix), 'Identificador RDN' (Identifier RDN), and 'uid'. On the left, there's a sidebar with categories: Personal, Unix, and Sombra. The 'Personal' category is selected. The main form contains fields for 'Nombre' (Name), 'Apellido' (Last name), 'Iniciales' (Initials), 'Nombre a mostrar' (Display name), 'Descripción' (Description), 'Calle' (Street), 'Oficina de correos' (Post office box), 'Código postal' (Zip code), 'Ubicación' (Location), 'Estado' (State), 'Dirección postal' (Postal address), 'Dirección registrada' (Registered address), and 'Nombre de la oficina' (Office name). Each field has a '+' icon to add more entries.

Pantalla de creación de usuario de LAM

Dicha interfaz es en opinión del autor, un poco espartana, pero resulta sencilla de utilizar y cubre sobradamente las necesidades básicas de gestión de un directorio LDAP, por lo que es una herramienta muy útil para grandes organizaciones que hagan uso de una instancia de OpenLDAP como servicio de directorio.

Al igual que el resto de software usado en este proyecto, es de licencia Open Source y gratuita, libre de descargar y utilizar; aunque se ofrece una versión Pro con funcionalidad ampliada

## 9.2.3 Stack LDAP. Despliegue y configuración básica

Pasamos al despliegue del pod propiamente dicho. Usaremos un manifiesto yaml y **podman** para automatizar el proceso, no obstante, explicaremos brevemente los aspectos más importantes del manifiesto yaml, pues introducirlo textualmente en este documento abarca más espacio del deseado.

Este pod requiere de un poco de configuración adicional, que será cubierta y justificada durante la guía de despliegue.

```
> podman -p stack_ldap/volumes/lam/lam_config
> edit: -p stack_ldap/volumes/openldap/config
> edit: -p stack_ldap/volumes/openldap/database
> tree -L 2 stack_ldap
stack_ldap
└── volumes
    └── lam
        └── openldap
            4 directories, 0 files
            > tree -L 3 stack_ldap
            stack_ldap
            └── volumes
                └── lam
                    └── openldap
                        ├── config
                        └── database
                            7 directories, 0 files
                            • / ~ ~/podman/proyecto
```

Comenzamos creando la estructura de directorios para albergar los volúmenes de persistencia de los contenedores del pod. LAM utiliza un único volumen para albergar la configuración, mientras que OpenLDAP utiliza 2 volúmenes, 1 para la configuración y otro para albergar la base de datos. En la captura superior se muestra la estructura final del directorio del pod.

```
> chown -Rv 911:911 stack_ldap/volumes/openldap
el propietario de 'stack_ldap/volumes/openldap/config' permanece como 911:911
el propietario de 'stack_ldap/volumes/openldap/database/load_memberof.ldif' permanece como 911:911
el propietario de 'stack_ldap/volumes/openldap/database/add_memberof.overlay.ldif' permanece como 911:911
el propietario de 'stack_ldap/volumes/openldap/database' permanece como 911:911
el propietario de 'stack_ldap/volumes/openldap' permanece como 911:911
• / ~ ~/podman/proyecto
```

Los contenedores de este pod, son exigentes en cuanto a acceso a sus volúmenes se refiere, y ambos requieren permiso de acceso total 777 (Inviable, pues es una mala práctica de seguridad) o la propiedad de los directorios. Esta última opción es mucho más adecuada por lo que una vez creados los directorios, cambiaremos la propiedad del directorio de los volúmenes de OpenLDAP al usuario 911 y al grupo 911 (UID y GID del usuario de OpenLDAP dentro del contenedor). De esta forma otorgaremos acceso al contenedor sin modificar los permisos por defecto 755, mucho más seguros que un 777.

LDAP Account Manager < https://www.ldap-account-manager.org/lamcms/releases

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Iniciar sesión – Nextcl... Free Content Comenzar a usar Fire... Comenzar a usar Fire... Primeros pasos >>> Otros marcadores

# LDAP Account Manager

Log in Search

Download Live Demo LAM Pro Shop Documentation Support Developers Partners

Take a look at [LDAP Account Manager Pro](#) for more powerful features.

LAM Pro customers can download the LAM Pro packages [here](#) (requires authentication).

The DEB/RPM packages are signed with our [PGP key](#).

If you use FreeBSD/Docker then take a look at our [additional downloads](#).

## LDAP Account Manager 8.7

 [Tar.bz2](#)

 [Debian/Ubuntu package](#)

 [Debian/Ubuntu package \(lamdaemon only\)](#)

 Ubuntu [Ubuntu requires php-psr-log 1.x](#). See [here](#).

## Cart

0 items

## News

LAM 8.7 with PHP 8.3 compatibility and passwordless SSO login for self service  
Sat, 2024-03-16

LAM 8.6 with new "Request access" module for self service and Docker update  
Tue, 2023-12-05

LAM 8.5 with accessibility improvements and better multi edit tool  
Fri, 2023-09-29

LAM celebrates its 20th anniversary  
Tue, 2023-09-19

La imagen del contenedor oficial de LAM no dispone en su interior los archivos de configuración básicos para el funcionamiento de la aplicación, por lo que debemos aprovisionarlo antes de ejecutar el contenedor. El primer paso para aprovisionarlo es descargar la release oficial de LAM desde su web oficial en formato tar.bz2:

Release oficial de LAM en formato tar.bz2

```
[root] ~ % curl https://github.com/1LDAPAccountManager/lam/releases/download/8.7/ldap-account-manager-8.7.tar.bz2  
- 2024-05-13 17:47:49 (67,9 MB/s) - <https://github.com/1LDAPAccountManager/lam/releases/download/8.7/ldap-account-manager-8.7.tar.bz2>  
Resolving github.com (github.com...): 186.82.121.3  
Connecting to github.com (github.com)[186.82.121.3]:443... connected.  
Petición HTTP enviada, esperando respuesta... 302 Found  
Localización: https://objects.githubusercontent.com/github-production-release-asset-2e65be/54287867/09661c9a-ec47-47c2-aed6-de96c4be1ba2?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCDYLSA53PQK4ZAB2F20240513%2Fus-east-1%2Fus-west-2%2Fus-south-1%2Faws4_request&X-Amz-Date=20240513T174502Z&X-Amz-Expires=30084X-Amz-Signature=1275591a9a321e56c6ee4248de3d2c5d56b5c9c9b587c5f6f9d4a6e7feF8X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=54387867&response-content-disposition=attachment%3Dfilename%3D0Jdap-account-8.7.tar.gz&X-Amz-Content-Type=application%2Foctet-stream [siguiendo]  
- 2024-05-13 17:47:50 - <https://objects.githubusercontent.com/github-production-release-asset-2e65be/54287867/09661c9a-ec47-47c2-aed6-de96c4be1ba2?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAVCDYLSA53PQK4ZAB2F20240513%2Fus-east-1%2Fus-west-2%2Fus-south-1%2Faws4_request&X-Amz-Date=20240513T174502Z&X-Amz-Expires=30084X-Amz-Signature=1275591a9a321e56c6ee4248de3d2c5d56b5c9c9b587c5f6f9d4a6e7feF8X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=54387867&response-content-disposition=attachment%3Dfilename%3D0Jdap-account-8.7.tar.gz&X-Amz-Content-Type=application%2Foctet-stream  
Resolviendo objetos.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...  
Conectando a objetos.githubusercontent.com (objects.githubusercontent.com)[185.199.109.133]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 26981454 (26M) [application/octet-stream]  
Guardando como: <ldap-account-manager-8.7.tar.gz>  
  
ldap-account-manager-8.7.tar.bz2 100%-----] 25,65M 67,9MB/s en 8,4s  
  
2024-05-13 17:47:50 (67,9 MB/s) - <https://github.com/1LDAPAccountManager/lam/releases/download/8.7/ldap-account-manager-8.7.tar.bz2> guardado [26981454/26981454]  
  
$ ls  
clean.sh grafana.old ldap-account-manager-8.7.tar.bz2 stack_grafana stack_ldap stack_nextcloud trash  
o ~ ~/podman/projects
```

Podemos por supuesto descargarlo directamente a nuestro servidor usando wget.

```
> tar xf ldap-account-manager-8.7.tar.bz2  
> ls  
clean.sh grafana.old ldap-account-manager-8.7 ldap-account-manager-8.7.tar.bz2 stack_grafana stack_ldap stack_nextcloud trash  
> mv ldap-account-manager-8.7  
confira| configure configure.ac COPYING copyright docs graphics help HISTORY index.html install.sh lib locale Makefile.in pwa_worker.js README sess style templates tmp VERSION  
o ~ ~/padman/projetos
```

Descomprimimos el archivo tar con **tar xjf nombrearchivo**. Se nos crea el directorio comprimido con toda la release, pero solo necesitamos el directorio **config**.

```
> ls ldap-account-manager-8.7/config  
addressbook.sample.conf config.cfg.sample language samba3.sample.conf selfService templates unix.sample.conf windows_samba4.sample.conf
```

Aun así, la release no viene completamente preparada, pues el archivo de configuración principal **config.cfg** no viene como tal por defecto, sino que viene una “muestra” llamada **config.cfg.sample**.

```
![[{"text": "tar -cvf ldap-account-manager-8.7/config\naddressbook.sample.conf config.cfg.sample language samba3.sample.conf selfService templates unix.sample.conf windows_semba4.sample.conf\n> -v ldap-account-manager-8.7/config/ldap.cfg.sample' \u2192 'ldap-account-manager-8.7/config/config.cfg'}], [{"text": "\u2192 ~/podman/proyecto"}], [{"text": "\u2192 with netfilter at 17:54:22"}]]
```

La solución a ese problema es muy simple, sacamos una copia renombrada de **config.cfg.sample** a **config.cfg** y listo.

```
![[{"text": "tar -cvf ldap-account-manager-8.7/config stack_ldap/volumes/lam/lam_config\n'ldap-account-manager-8.7/config' \u2192 'stack_ldap/volumes/lam/lam_config/config'\n'ldap-account-manager-8.7/config/htaccess' \u2192 'stack_ldap/volumes/lam/lam_config/config/htaccess'\n'ldap-account-manager-8.7/config/addressbook.sample' \u2192 'stack_ldap/volumes/lam/lam_config/config/addressbook.sample.conf'\n'ldap-account-manager-8.7/config/config.cfg.sample' \u2192 'stack_ldap/volumes/lam/lam_config/config/config.cfg.sample'\n'ldap-account-manager-8.7/config/language' \u2192 'stack_ldap/volumes/lam/lam_config/config/language'\n'ldap-account-manager-8.7/config/samba3.sample.conf' \u2192 'stack_ldap/volumes/lam/lam_config/config/samba3.sample.conf'\n'ldap-account-manager-8.7/config/selfService' \u2192 'stack_ldap/volumes/lam/lam_config/config/selfService'\n'ldap-account-manager-8.7/config/placeholder' \u2192 'stack_ldap/volumes/lam/lam_config/config/placeholder'\n'ldap-account-manager-8.7/config/templates' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates'\n'ldap-account-manager-8.7/config/templates/pdf' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf'\n'ldap-account-manager-8.7/config/templates/pdf/default.alias.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.alias.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.asteriskExt.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.asteriskExt.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.automountType.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.automountType.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.bind.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.bind.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.customType.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.customType.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.dhcp.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.dhcp.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.gon.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.gon.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.group.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.group.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.host.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.host.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.kolabSharedFolderType.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.kolabSharedFolderType.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.kopanoAddressListType.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.kopanoAddressListType.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.kopanoAddressListType.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.kopanoAddressListType.xml'"}]]
```

Una vez listo el contenido del directorio config, lo copiamos a nuestro directorio lam\_config, el volumen de persistencia de la aplicación.

```
![[{"text": "'ldap-account-manager-8.7/config/templates/pdf/default.gon.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.gon.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.group.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.group.xml'\n'ldap-account-manager-8.7/config/templates/pdf/default.host.xml' \u2192 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.host.xml'\n'> -v 33:33 stack_ldap/volumes/lam/lam_config"}, {"text": "cambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/htaccess' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/addressbook.sample.conf' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/config.cfg.sample' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/language' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/samba3.sample.conf' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/selfService' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.alias.xml' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.asteriskExt.xml' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.automountType.xml' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.bind.xml' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.customType.xml' de root:root a 33:33\n\tcambiado el propietario de 'stack_ldap/volumes/lam/lam_config/config/templates/pdf/default.dhcp.xml' de root:root a 33:33"}]]
```

Por último, al igual que con ldap, debemos dar al usuario interno del contenedor la propiedad del directorio del volumen para que tenga acceso al mismo; en este caso con el UID 33 y GID 33 mediante otro **chown**.

Turbo ahora de desplegar el pod mediante **podman kube** y nuestro previamente confeccionado archivo yaml. Dado que los archivos yaml son muy grandes y una captura para este documento no es capaz de abarcarlos, se muestran a continuación solo fragmentos de dicho yaml y la explicación de los parámetros más importantes.

Este junto con los 2 yaml restantes de los otros pods se adjuntarán al completo con el resto de documentos del proyecto.

```
![[{"text": "GNU nano 7.2\napiVersion: v1\nkind: Pod\nmetadata:\n  creationTimestamp: \"2024-03-17T00:16:33Z\"\n  labels:\n    app: proyecto_stackldap\n  name: proyecto_stackldap"}, {"text": "stack_ldap/stack_ldap.yaml"}, {"text": "Modificado"}], [{"text": "Cabeecera", "x": 380, "y": 630}, {"text": "Nombre para el pod", "x": 380, "y": 650}]]
```

La cabecera contiene datos básicos del archivo yaml, despliegue de tipo pod, la fecha de creación de la primera versión del yaml y el nombre que tomara el pod.

```
![[{"text": "spec:\n  volumes:\n    - name: lam_config\n      hostPath:\n        path: /root/podman/proyecto/stack_ldap/volumes/lam/lam_config\n      name: openldap_config\n      hostPath:\n        path: /root/podman/proyecto/stack_ldap/volumes/openldap/config\n      name: openldap_database\n      hostPath:\n        path: /root/podman/proyecto/stack_ldap/volumes/openldap/database\n      name: openldap_database"}, {"text": "DEFINICION DE LOS VOLUMENES DE PERSISTENCIA"}], [{"text": "Ruta del volumen en el host", "x": 480, "y": 750}, {"text": "Nombre del volumen", "x": 480, "y": 765}]]
```

En el apartado de specs definiremos los volúmenes previamente preparados, indicando la ruta en la que se encuentran en el servidor y el nombre asignado a cada uno.

```

containers:
  - env:
    - name: LDAP_DOMAIN
      value: ismael-server.ddns.net
    - name: LDAP_BASE_DN
      value: dc=dcn,dc=dcn,dc=dcn
    - name: LDAP_SERVER
      value: ldap://127.0.0.1:389
    - name: LAM_PASSWORD
      value: Astirpo024
    - name: LAM_LANG
      value: es_ES
  # name: LAM_SKIP_PRECONFIGURE
  image: ghar.io/ldapaccountmanager/lam:latest
  securityContext:
    fsGroup: 33
  # runAsUser: 33 # UID para este contenedor
  # runAsGroup: 0
  name: lam
  tty: true
  volumeMounts:
    - mountPath: /var/lib/ldap-account-manager/config/z
      name: lam_config

```

Ventana de terminal con la definición de un contenedor llamado 'lam' que ejecuta la imagen 'ghar.io/ldapaccountmanager/lam:latest'. La configuración incluye variables de entorno para el dominio LDAP ('LDAP\_DOMAIN'), la base DN ('LDAP\_BASE\_DN'), el servidor LDAP ('LDAP\_SERVER'), la contraseña ('LAM\_PASSWORD') y el idioma ('LAM\_LANG'). Se incluyen también las variables 'LAM\_SKIP\_PRECONFIGURE' y 'LAM\_CONFIG'. Los recursos se gestionan mediante 'volumeMounts' para el directorio '/var/lib/ldap-account-manager/config/z' en el contenedor.

En la sección de contenedores es donde definiremos los contenedores propiamente dichos, indicando el nombre que deseamos asignarle, las variables de entorno del mismo, la imagen en la que se basa, el grupo de seguridad (el usuario que ejecuta internamente el contenedor) y la asignación de los volúmenes previamente definidos y su punto de montaje interno. La captura sobre estas líneas muestra la definición del contenedor de LAM.

```

env:
  - name: LDAP_DOMAIN
    value: Ismael-Server.ddns.net
  - name: LDAP_ADMIN_PASSWORD
    value: Astirpo024
  - name: KEEP_EXISTING_CONFIG
    value: "true"
  - name: LDAP_ORGANISATION
    value: Ismael - 2º ASIR - IES Aguadulce
  - name: LDAP_ADMIN_PASSWORD
    value: Astirpo024
image: docker.io/ostixia/openldap:1.5.0
securityContext:
  # runAsUser: 33 # UID para este contenedor
  # runAsGroup: 33
  name: openldap
  ports:
    - containerPort: 389
      hostPort: 389
    - containerPort: 636
      hostPort: 636
  tty: true
  volumeMounts:
    - mountPath: /etc/ldap/slapd.d/z
      name: openldap_config
    - mountPath: /var/lib/ldap/z
      name: openldap_database

```

Ventana de terminal con la definición de un contenedor llamado 'openldap' que ejecuta la imagen 'docker.io/ostixia/openldap:1.5.0'. La configuración incluye variables de entorno para el dominio LDAP ('LDAP\_DOMAIN'), la contraseña de administrador ('LDAP\_ADMIN\_PASSWORD') y la opción 'KEEP\_EXISTING\_CONFIG' establecida en 'true'. El contexto de seguridad es 'openldap'. Los puertos 389 y 636 están expuestos al host. Los volúmenes 'openldap\_config' y 'openldap\_database' se montan en '/etc/ldap/slapd.d/z' y '/var/lib/ldap/z' respectivamente.

El contenedor de Openldap se define de una forma muy similar. Sin embargo, como se muestra en la captura superior se definen para él los puertos necesarios por el servicio. Estos puertos solo serían necesarios si necesitáramos exponerlos al exterior para conectar al servicio de directorio otras máquinas, por lo que en la versión definitiva del despliegue serán comentados y deshabilitados.

```

$ pod ls && podman container ls
POD ID          NAME           STATUS        CREATED     INFRA ID   # OF CONTAINERS
2cf3a8b9e9...   proyecto_stacknc   Running      About an hour ago  512cba277e28  4
CONTAINER ID    IMAGE          COMMAND      CREATED     STATUS      PORTS          NAMES
512cba277e28  localhost/podman-pause:4.9.4-171445992  /            About an hour ago  Up About an hour  0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  2cf3a8b9e9-infra
656dc78f305a  docker.io/library/nextcloud:stable-apache  apache2-foreground...  About an hour ago  Up About an hour  0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  proyecto_stacknc-nextcloud
b8219bc18ac...  docker.io/library/postgres:12test  postgres      About an hour ago  Up About an hour  0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  proyecto_stacknc-postgres
62d54449fe57  docker.io/dpage/pgadmin4:1test       pgadmin      About an hour ago  Up About an hour  0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp  proyecto_stacknc-pgadmin

```

Ventana de terminal mostrando la lista de pods y containers. Hay un pod llamado 'proyecto\_stacknc' que contiene cuatro containers: '2cf3a8b9e9-infra' (que ejecuta 'localhost/podman-pause:4.9.4-171445992'), 'proyecto\_stacknc-nextcloud' (que ejecuta 'docker.io/library/nextcloud:stable-apache'), 'proyecto\_stacknc-postgres' (que ejecuta 'docker.io/library/postgres:12test') y 'proyecto\_stacknc-pgadmin' (que ejecuta 'docker.io/dpage/pgadmin4:1test'). Los containers están en ejecución.

Procedamos pues al despliegue. Se aprecia en la captura superior que en este momento solo está en ejecución el pod del stack nextcloud.

```

$ kubectl play --network proyecto_stack ldap/stack ldap.yml
Trying to pull docker.io/ostixia/openldap:1.5.0...
Getting image source signatures
Copying blob aef12f184516 done
Copying blob f8b134d16ba5 done
Copying blob 27f335cced6e done
Copying blob 866d24991091 done
Copying blob 8e6d62499765 done
Copying blob 8a1e25c7e4f4 done
Copying blob d2837a1b1080 done
Copying blob 52e8676423 done
Copying blob cc711da6e514f done
Copying blob 52e8676423 done
Copying blob 6899e0fd8884 done
Copying blob 87132130a140 done
Copying blob 8ff52aa1eb7b done
Copying blob cbc5524e7ea9 done
Copying blob e1cb4647e74c done
Copying blob 74476080e0 skipped: already exists
Writing manifest to image destination
Trying to pull docker.io/ostixia/openldap:1.5.0...
Getting image source signatures
Copying blob 45542c59b833 done
Copying blob 55443d9da5d5 done
Copying blob ae7f185973b done
Copying blob b862b291946a done
Copying blob 3731e12f1fa4 done
Copying blob b1fe1a476881 done
Copying blob 70342a1214a1 done
Copying blob 2da3cf6448 done
Writing manifest to image destination
Pod:
Containers:
 707345279a5938df7ed7847268263944d625a98c1661f848251e82e15c7
  a7aa9babe478c2594f314e6da568f51df15f7f16fa18ee495597e5f3e4dc

```

Ventana de terminal con el comando 'kubectl play --network proyecto\_stack ldap/stack ldap.yml'. Se intenta pullar la imagen 'docker.io/ostixia/openldap:1.5.0'. Se muestran los blobs que se están copiando y la creación de los manifestos para los containers.

Ejecutamos **podman play kube —network red /ruta/all.yml** y el pod queda desplegado.

89  
Administración de sistemas informáticos en red

Y podman pod ls && curl curl container ls	POD ID	NAME	STATUS	CREATED	INFRA ID	# OF CONTAINERS
	9404351e2347	proyecto_stackldap	Running	54 seconds ago	32cbf038919a	3
	2cff3ab5b9e9	proyecto_stacknc	Running	About an hour ago	512cbf0277e2d	4
	c2126a77a28	localhost/podman-pause	Up	About an hour ago	COMMAND	
	656dc78f305a	docker.io/library/nexcloud:stable-apache	apache2-foreground	About an hour ago	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp	2cff3ab5b9e9-infra
	b2819dc18c	docker.io/library/postgres:latest	postgres	About an hour ago	0.0.0.0:5432->5432/tcp, 0.0.0.0:5433->5433/tcp	proyecto_stacknc-nextcloud
	62d54449fe67	docker.io/pgadmin4:latest	pgadmin	About an hour ago	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp	proyecto_stacknc-postgres
	32cbf038919a	localhost/podman-pause:4.9.4-1711445992	Up	54 seconds ago	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp	proyecto_stacknc-pgadmin
	157a8b2a2032	ghcr.io/ldapcontroller/ldap:latest	/usr/local/bin/st...	52 seconds ago	Up 52 seconds	9404351e2347-infra
	423817fa3d8	docker.io/sidix/openldap:1.5.0	openldap	52 seconds ago	Up 52 seconds	proyecto_stackldap-jam
						proyecto_stackldap-openldap

¡Listo! El pod ha sido desplegado y los contenedores están en ejecución.

Sin embargo es necesario realizar un ajuste adicional para permitir o denegar los logins a Nexcloud de los futuros usuarios en base a la pertenencia a un grupo específico. Para ello debemos cargar en el servidor LDAP un módulo llamado memberof overlay.

```
GNU nano 2.2
dn: olcOverlay-memberof,cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: memberof
```

El primer paso es crear un archivo ldif con el contenido que se muestra en la captura (se adjuntara con el resto de scripts y archivos). Dicho ldif modifica la base de datos de configuración del ldap para cargar dicho modulo.

```
GNU nano 7.2
dn: olcOverlay-memberof,olcDatabase={1}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
olcAccess: {1}labeledMemberof
olcOverlay: memberof
olcMemberOfRefint: TRUE
```

A continuación, creamos otro archivo ldif con el contenido mostrado. Este ldif añade al esquema del directorio un nuevo objeto llamado “memberof”, que usaremos como atributo para permitir o denegar que un usuario pueda loguearse en Nexcloud.

```
root@stack_ldap:/# cp stack_ldap/load_memberof.ldif stack_ldap/volumes/openldap/database
root@stack_ldap:/# cp stack_ldap/add_memberof_overlay.ldif stack_ldap/volumes/openldap/database
root@stack_ldap:/# cp add_memberof_overlay.ldif data.ldb load_memberof.ldif lock.ldb
```

Necesitamos cargar dichos archivos dentro de la Shell del contenedor, por lo que los copiamos a uno de los volúmenes del mismo para que este tenga acceso.

```
root@podman:~# podman exec -it proyecto_stackldap-openldap /bin/bash
root@proyecto_stackldap:[#]
```

Ejecutando `podman exec -it proyecto_stackldap-openldap /bin/bash` ganamos acceso a una terminal directa en el contenedor, vemos como el prompt de nuestra terminal cambia.

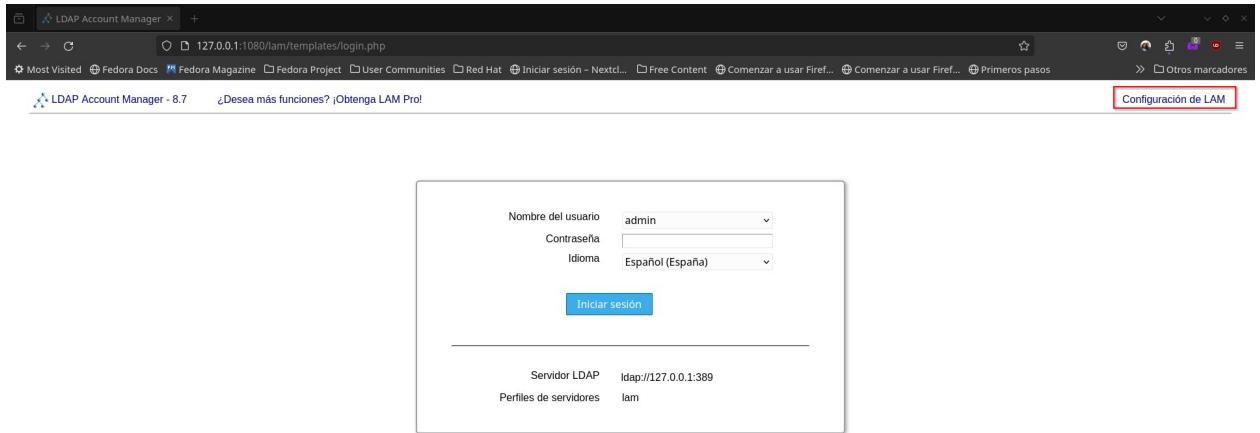
```
root@proyecto_stackldap:/# cd /var/lib/ldap/
root@proyecto_stackldap:/var/lib/ldap# ldapmodify -Y EXTERNAL -H ldap:// -f load_memberof.ldif
SASL/EXTERNAL authentication started
SASL/PLAIN: gidiMember+@uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=module{0},cn=config"
root@proyecto_stackldap:/var/lib/ldap# slapcat -n 0 | grep olcModuleLoad
olcModuleLoad: {0}back_ldap
olcModuleLoad: {1}memberof
root@proyecto_stackldap:/var/lib/ldap#
```

Navegamos a la ubicación del volumen donde hemos copiado nuestros ldif y ejecutamos la herramienta ldapmodify con el archivo `load_memberof.ldif`. Una consulta con slapcat muestra que el módulo esta correctamente cargado a partir de ahora.

```
root@proyecto_stackldap:/var/lib/ldap# ldapmodify -Y EXTERNAL -H ldap:// -f add_memberof_overlay.ldif
SASL/EXTERNAL authentication started
SASL/PLAIN: gidiMember+@uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "olcOverlay=memberof,olcDatabase={1}mdb,cn=config"
```

Volvemos a usar ldapmodify con el archivo `add_memberof_overlay.ldif` y el nuevo atributo **memberof** queda listo para ser usado. Ya podríamos cerrar la terminal del contenedor con un **exit** o **logout**.

Listo nuestro servidor de OpenLDAP pasemos a echar un vistazo a su cara visible, LAM.



Si introducimos la dirección IP del contenedor (En este caso se está usando un tunel ssh para acceder al contenedor mediante localhost) o del host en caso de una redirección de puertos, accederemos a la pantalla de login de LAM. El primer paso que debemos realizar es entrar al apartado de configuración de la aplicación.



Entraremos a la sección de ajustes generales.



Se nos solicitará la contraseña maestra de la aplicación (no confundir con la contraseña del perfil de conexión o la contraseña de acceso al servidor ldap). Dicha contraseña es por defecto "lam". Vamos a cambiarla.

Longitud mínima de la contraseña: 0

Caracteres mínimos con minúscula: 0

Caracteres mínimos con mayúsculas: 0

Mínimo de caracteres numéricos: 0

Caracteres mínimos con símbolos: 0

Clases de caracteres mínimos: 0

Número de reglas que deben coincidir: todos

La contraseña no puede tener el nombre del usuario

La contraseña no debe contener partes del primer/segundo nombre ni el nombre de usuario

Verificación externa de contraseña:

Iniciando sesión

Nivel de trazas: Advertencia

Destino de las trazas: Registrando en el sistema.

Reporte de errores PHP: predeterminado

Cambiar la contraseña maestra

Nueva contraseña maestra: Asirpro2024

Vuelva a introducir la contraseña:

Guardar Cancelar

Cambiamos la contraseña a la que deseemos y pulsamos en guardar. Este panel contiene múltiples opciones para cambiar parámetros generales de la aplicación, no vamos a explicarlos pues no necesitamos cambiar ninguno más allá de la contraseña.

Ajustes generales Tipos de cuentas Módulos Preferencias del módulo

Preferencias del servidor

Dirección del servidor: ldap://127.0.0.1:389

Activar TLS: no

Límite de búsqueda LDAP: no

Parte del DN a ocultar:

Método del inicio de sesión: Lista fija

Lista de usuarios válidos: cn=admin,dc=ismael-server,dc=ddns,dc=net

Opciones Avanzadas +

Configuración del idioma

Idioma por defecto: Español (España)

Zona horaria: Europe/London

La sección de perfiles de servidor contiene ajustes específicos de conexión con un servidor concreto en dichos perfiles. La sección más importante es la de conexión al servidor LDAP, pero una de las comodidades de nuestro yaml es que dichos ajustes de conexión ya están definidos como variables de entorno, por tanto, la dirección del servidor, contraseña, usuario, o dominio base ya están previamente configuradas en cuanto se realiza el despliegue.

LDAP Account Manager - 8.7 ¿Desea más funciones? ¡Obtenga LAM Pro!

Configuración de LAM

Nombre del usuario: admin

Contraseña:

Idioma: Español (España)

Iniciar sesión

Servidor LDAP: ldap://127.0.0.1:389

Perfiles de servidores: lam

Hora de loguearnos en el perfil, para contactar con nuestro servidor de LDAP. Hemos definido durante el despliegue el uso del usuario administrador del directorio, por tanto, introducimos su usuario y contraseña.

LAM también cuenta con un útil visor de árbol, para apreciar la estructura jerárquica de los objetos presentes en el directorio.

Como vemos, en estos momentos nuestro directorio esta vacío. Por lo que debemos poblarlo con objetos.

Vamos a poblar nuestro directorio usando la herramienta de importación/exportación de Idif. Usaremos un texto Idif previamente generado mayormente con objetos inventados. Dicho Idif también se adjuntará a la entrega del proyecto.

Una vez en la herramienta tan solo tenemos que pegar el contenido de nuestro Idif con los objetos del directorio en la caja de texto, seleccionar la fuente como entrada de texto y pulsar en enviar.

Importar  Exportar**Importar**

Estado: listo

Nueva importación

**Entrada creada**  
ou=equipos-informaticos,dc=ismael-server,dc=ddns,dc=net

**Entrada creada**  
cn=ImpresoraDepartamental,ou=equipos-informaticos,dc=ismael-server,dc=ddns,dc=net

**Entrada creada**  
cn=ImpresoraPrincipal,ou=equipos-informaticos,dc=ismael-server,dc=ddns,dc=net

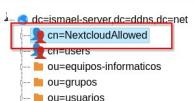
**Entrada creada**  
cn=LaptopEjecutivo,ou=equipos-informaticos,dc=ismael-server,dc=ddns,dc=net

LAM nos informa en su salida del resultado de la importación, en este caso todas las entradas han sido importadas.

**uid=icarrascocubero,ou=usuarios,dc=ismael-server,dc=ddns,dc=net****Atributos**

cn \* Ismael  
 description El creador  
 gidNumber \* 10000  
 givenName Ismael  
 homeDirectory \*/home/icarrascocubero  
 initials ICC  
 l Córdoba  
 loginShell /bin/bash  
 mobile 684093179  
 objectClass \* posixAccount  
 inetOrgPerson  
 organizationalPerson

Ya tenemos nuestro directorio poblado con objetos simulando un entorno empresarial (más o menos). Desde el visor de árbol podemos ver detalles de cada objeto o modificarlos en el panel de la derecha, como se muestra en la captura encima de estas líneas.

**cn=NextcloudAllowed,dc=ismael-server,dc=ddns,dc=net****Atributos**

cn \* NextcloudAllowed  
 description Usuarios con permiso de acceso a Nextcloud  
 member \* uid=agutierrez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net  
 uid=icarrascocubero,ou=usuarios,dc=ismael-server,dc=ddns,dc=net  
 objectClass \* groupOfNames

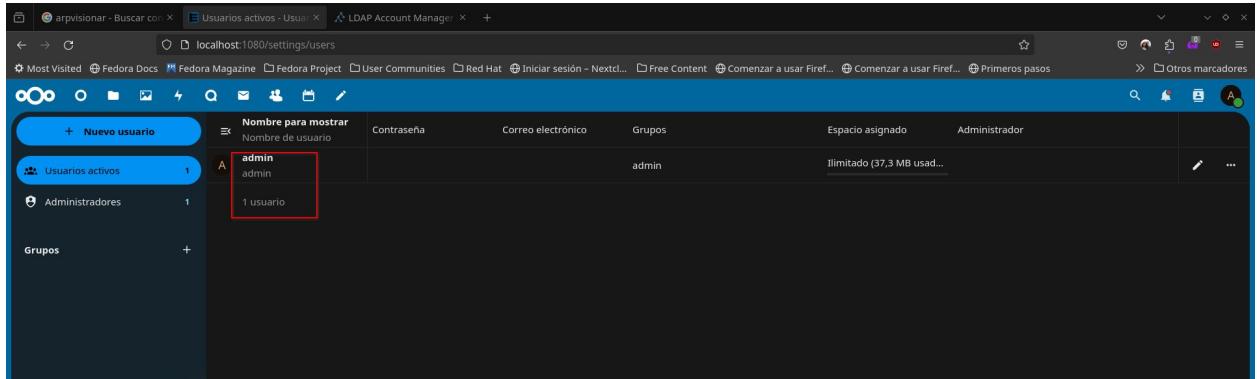
Una de las entradas más importantes de nuestro directorio es “NextcloudAllowed”, un objeto de tipo groupOfNames que funciona de forma similar a un grupo UNIX.

Gracias al módulo memberof overlay, cargado con anterioridad, podemos usar este grupo para permitir o denegar el acceso a Nextcloud en base a su pertenencia o no.

Podríamos por supuesto crear otros groupOfNames con distinto nombre y usarlos para el mismo cometido.

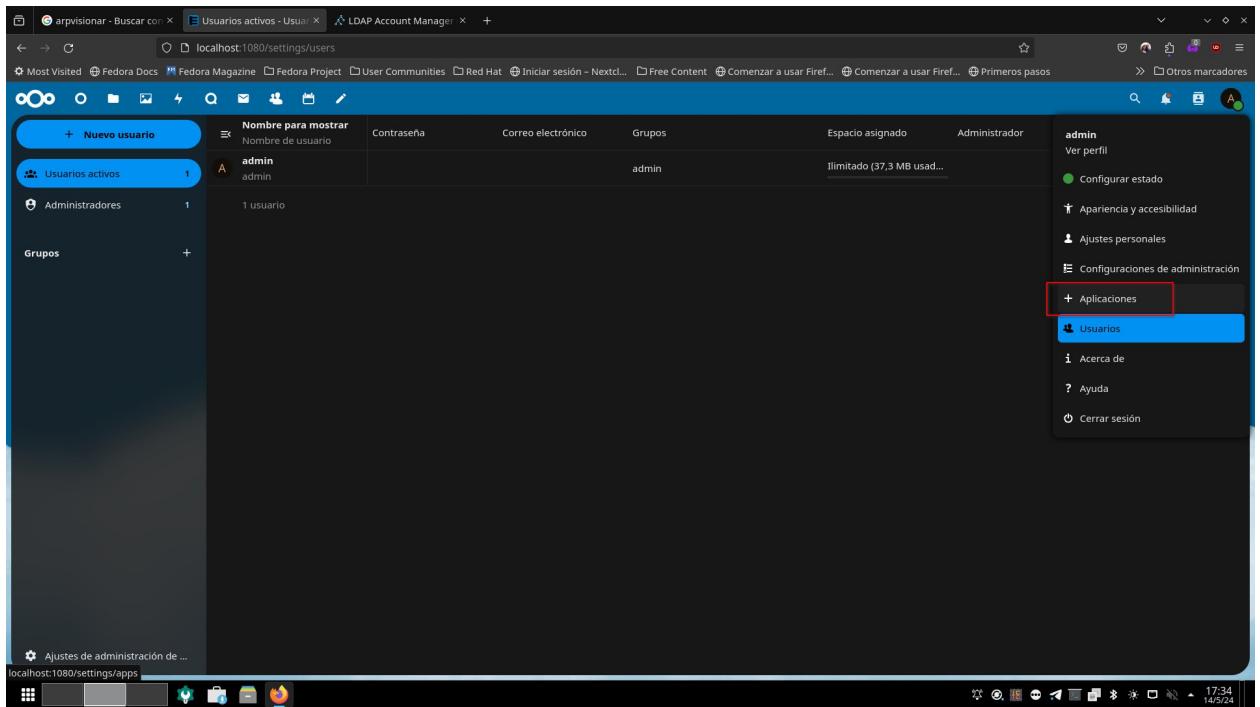
## 9.2.4 Stack LDAP. Conexión De Nextcloud al servicio de directorio

Una vez que hemos desplegado nuestro pod con el servicio de directorio LDAP, y este está activo y con usuarios, podemos conectar a Nextcloud con una aplicación interna al servicio LDAP y usar el directorio como base de datos de login.



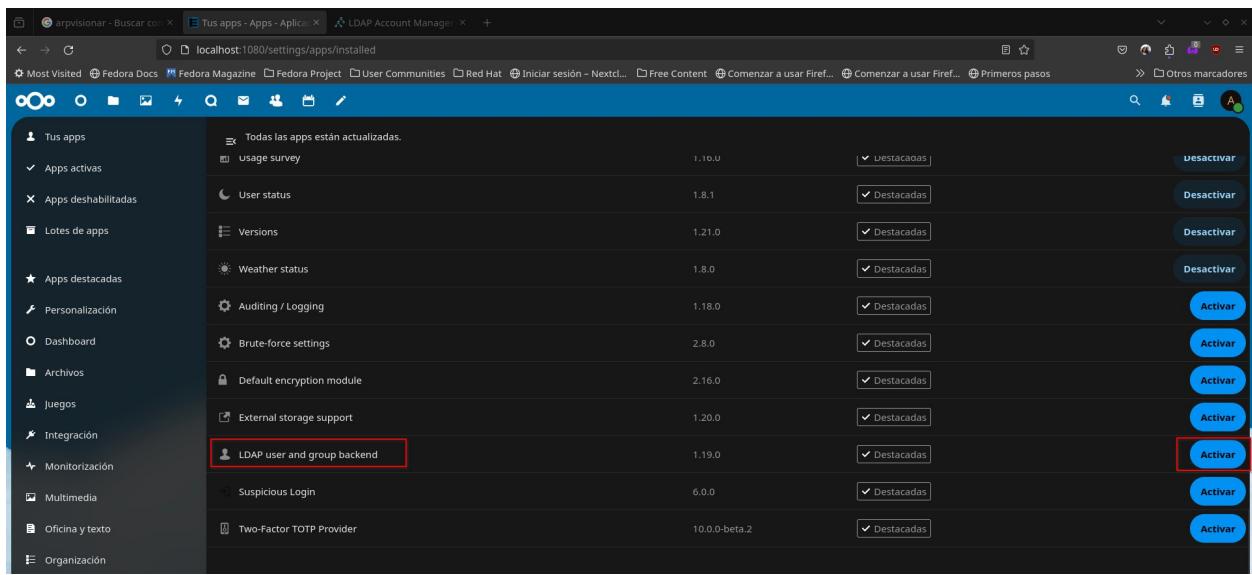
The screenshot shows the 'Users active' section of the LDAP Account Manager. It lists one user: 'admin' (Nombre para mostrar: admin, Nombre de usuario: admin). The 'Grupos' section shows 'Administradores' with 1 member. The 'Espacio asignado' column indicates 'Ilimitado (37,3 MB usad...)'. The 'Administrador' column shows 'admin'. A red box highlights the user 'admin' in the list.

Si accedemos a nuestra instancia de Nextcloud, al apartado de usuarios, podemos comprobar que el único usuario presente es el propio administrador. Conectemos la instancia al directorio.

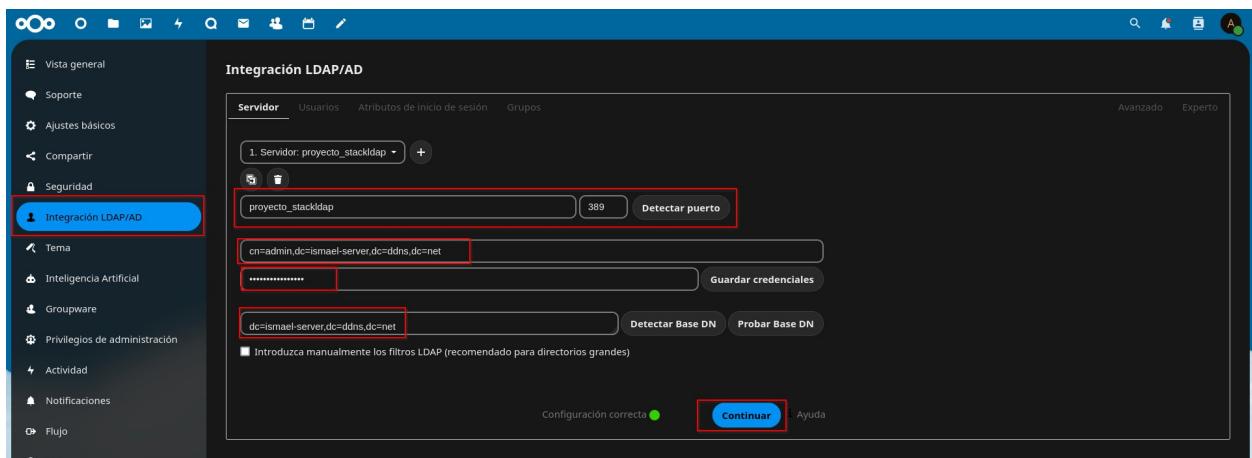


The screenshot shows the Nextcloud Admin sidebar. The 'Aplicaciones' section is highlighted with a red box. Other options visible include 'Ver perfil', 'Configurar estado', 'Apariencia y accesibilidad', 'Ajustes personales', 'Configuraciones de administración', 'Usuarios', 'Acerca de', 'Ayuda', and 'Cerrar sesión'.

Para ello comenzamos por entrar en el apartado de aplicaciones para añadir la app necesaria.



Si hacemos scroll hasta la parte baja, encontraremos la aplicación “LDAP user and group backend”. Normalmente viene preinstalada en la instancia por defecto (aunque no activada) por lo que simplemente debemos activarla mediante el correspondiente botón.



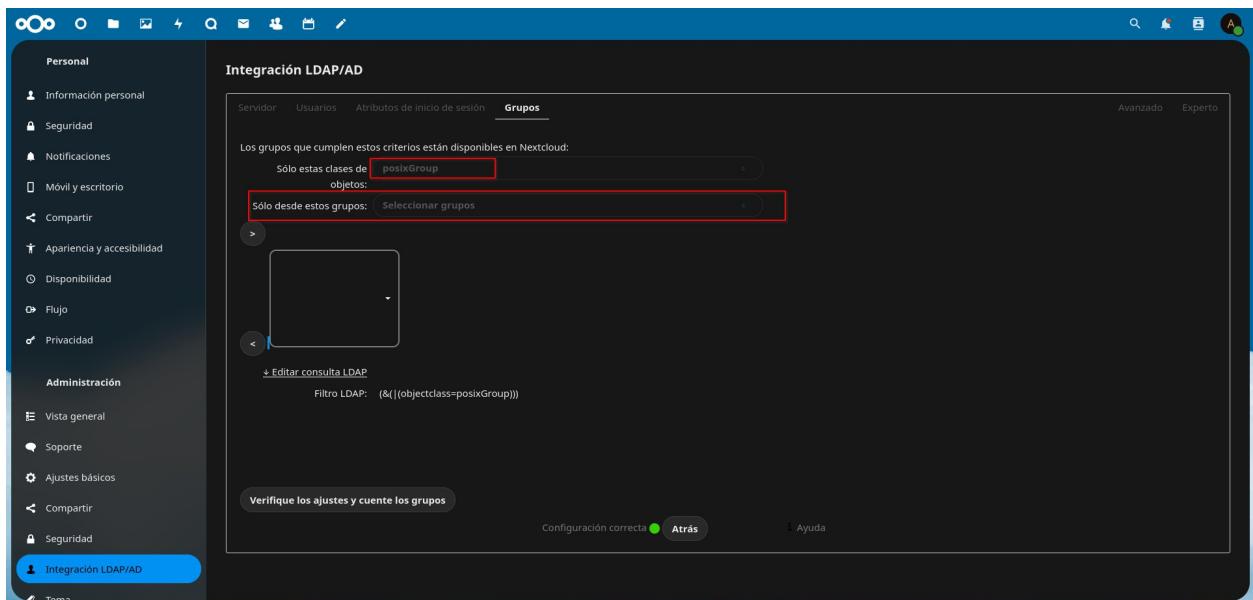
Una vez activa, debemos entrar en el panel de configuración de administrador y navegar hasta la sección de integración LDAP. Comenzaremos cumplimentando la información del servidor como se muestra en la captura, introduciendo la dirección IP (o nombre DNS, como es en este caso con el nombre de nuestro pod LDAP) y el puerto del servidor LDAP, la raíz del dominio, el usuario que accederá y su contraseña. Nextcloud nos informará si es capaz de conectar con el servidor LDAP y de ser el caso pulsamos en continuar para pasar a la siguiente pestaña.

La siguiente pestaña nos permite configurar el login de usuarios. Debemos escoger que clases de objeto del directorio están permitidas como usuarios dentro de nextcloud (para evitar por ejemplo que un objeto de tipo posixGroup aparezca como usuario).

La siguiente opción “Solo desde estos grupos” es la opción relacionada con los groupOfNames, en nuestro caso seleccionamos nuestro groupOfNames “NextcloudAllowed”. Si bien el panel de selección detecta grupos de tipo Posix, seleccionarlos no tiene ningún efecto; la discriminación solo funciona con groupOfNames.

Tras configurar este apartado, vemos que Nextcloud está comenzando a detectar usuarios validos en base a nuestros criterios si ejecutamos una prueba de configuración. Pulsamos en continuar para pasar al siguiente paso.

En la siguiente pestaña, podríamos seleccionar que atributos de los usuarios disponibles pueden ser usados para el login, como el nombre, apellido, correo electrónico, nombre de pila etc. Sin embargo, la opción más lógica es dejarlo por defecto y usar los nombres de usuario de LDAP (UID) por lo que solo nos aseguraremos de que están marcadas las dos opciones que se muestran y podemos continuar.



Por último, en la pestaña de grupos podemos configurar de qué forma Nexcloud reconoce a los grupos de usuarios presentes en LDAP. Debemos seleccionar la clase de objeto que queramos que sea integrada (posixGroup) y podríamos seleccionar en el desplegable si solo queremos grupos específicos. En nuestro caso, en esta pantalla la configuración por defecto nos sirve, por lo que hemos terminado la conexión con LDAP.

Nombre para mostrar	Nombre de usuario	Contraseña	Correo electrónico	Grupos	Administrador de grupo para	Espacio asignado	Admi...
D	David	2bcdcd57b-a4a9-103e-82...		RecursosHumanos		Ilimitado (37,3 MB usad...	...
S	Sandra	2bceb01e-a4a9-103e-82...		GrupoDesarrollo		Ilimitado (0 B usados)	...
M	Marta	2bcfa12c-a4a9-103e-82...		EquipoAdmin		Ilimitado (37,3 MB usad...	...
J	José	2bd09e7a-a4a9-103e-82...		EquipoVentas		Ilimitado (37,3 MB usad...	...
D	Daniel	add3d1ca2-a4a5-103e-82...				Ilimitado (37,3 MB usad...	...
I	Ismael	ad3e0144-a4a5-103e-82...				Ilimitado (37,3 MB usad...	...
S	Sebastián	ad3ef9f0-a4a5-103e-82...				Ilimitado (37,3 MB usad...	...
J	Juan José	ad3fe0e2-a4a5-103e-82...				Ilimitado (37,3 MB usad...	...
M	Manuel	ad40e468-a4a5-103e-82...				Ilimitado (0 B usados)	...
A	Alejandro	ad41c72a-a4a5-103e-82...		GrupoDesarrollo		Ilimitado (37,3 MB usad...	...
A	admin	admin		admin		Ilimitado (37,3 MB usad...	...
I	Ismael	fb466ab0-a4b1-103e-97...		users		Ilimitado (37,3 MB usad...	...

En el momento en el que terminamos la conexión al directorio, si nos dirigimos al panel de administración de usuarios, vemos como ya aparecen los usuarios presentes en el directorio. Estos usuarios son únicamente aquellos que tienen el atributo memberof referenciando al groupofnames "NextcloudAllowed". Pero es mejor verlo en acción que explicarlo.

Tomemos como ejemplo al usuario “Alejandro García” (agarcia), que se muestra sobre estas líneas.

Si observamos sus atributos internos, vemos que no hay mención alguna a la pertenencia a “NextcloudAllowed” ...



Si intentamos hacer login con su usuario de LDAP y contraseña...



Nextcloud no lo reconoce como usuario valido y deniega el acceso.

Administración de sistemas informáticos en red

100

LDAP Account Manager - 8.7 admin

Cuentas Herramientas Cerrar sesión

dc=ismael-server,dc=ddns,dc=net

- cn=NextcloudAllowed
- cn=users
- ou=equipos-informaticos
- ou=grupos
- ou=usuarios
  - uid=agarcia
  - uid=aguileraez
  - uid=amarinez
  - uid=crodriguez
  - uid=dcano
  - uid=dperalta
  - uid=eperez
  - uid=icarrascocubero
  - uid=lopez
  - uid=jgomez
  - uid=jmartin
  - uid=jrodriguez
  - uid=jsanchez**
  - uid=mgonzalez
  - uid=mlopez
  - uid=mserrano
  - uid=slopez
  - uid=sruiz

**uid=jsanchez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net**

Atributos	
cn*	Juan José
gidNumber*	1004
givenName	Juan José
homeDirectory*	/home/jsanchez
loginShell	/bin/bash
objectClass*	inetOrgPerson posixAccount shadowAccount
shadowLastChange	19855
sn*	Sánchez Céspedes
uid*	jsanchez
uidNumber*	1004
userPassword	[REDACTED] SSHA

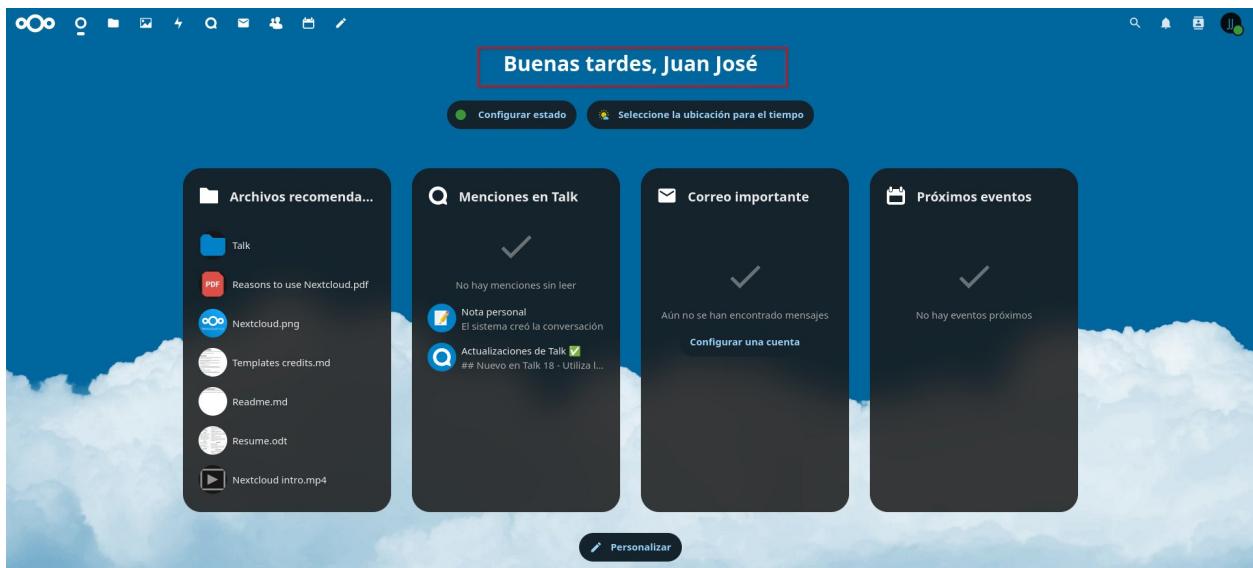
Probemos ahora con otro usuario distinto, en este caso el de Juan José (Todos los profesores tendréis acceso, obviamente).

Atributos internos	
createTimestamp	20240513175457Z
creatorsName	cn=admin,dc=ismael-server,dc=ddns,dc=net
entryCSN	20240513175457.589374Z#00000#000#00000
entryDN	uid=jsanchez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net
entryUUID	a7de9e4-a59d-103e-86b8-4d450b65219c
hasSubordinates	FALSE
memberOf	cn=NextcloudAllowed,dc=ismael-server,dc=ddns,dc=net
modifiersName	cn=admin,dc=ismael-server,dc=ddns,dc=net
modifyTimestamp	20240513175457Z
structuralObjectClass	inetOrgPerson
subschemaSubentry	cn=Subschema

Si echamos un vistazo a sus atributos internos, este usuario si dispone de pertenencia a NextcloudAllowed referenciando el groupOfNames con el atributo interno memberOf.



Si intentamos loguearle con sus credenciales de LDAP...



Nextcloud encuentra al usuario en LDAP, y comprueba su pertenencia a NextcloudAllowed, y a continuación permite el acceso.

Con esto ya tendríamos operativa una instancia de Nextcloud lista para trabajar y una útil forma centralizada para gestionar a los usuarios de nuestra organización, por lo que hemos completado el despliegue de nuestro stack LDAP.

## 9.3 Stack Grafana

Hasta el momento, tenemos un servidor completamente operativo, con una buena pléthora de utilidades de administración y una aplicación de productividad lista para usarla.

Sin embargo, todo servidor que se precie debe tener alguna herramienta para recopilar métricas del sistema y poder tenerlo controlado en todo momento.

El siguiente pod, desplegará un stack con 3 contenedores: **Grafana**, **Prometheus** y **Node Exporter**; con los que podremos recopilar métricas del hardware del sistema, como la carga del mismo, el uso de CPU o memoria, I/O de Disco, red etc.

### 9.3.1 Stack Grafana. El panel de métricas Grafana

Grafana es una aplicación web que nos permite la creación de paneles de mando o “Dashboards” en los que agrupar métricas de nuestros sistemas computacionales.

Posee una interfaz sencilla pero muy completa, con la posibilidad de crear desde 0 nuestros propios paneles o importarlos de fuentes externas, incluido su propio repositorio comunitario en el que los usuarios comparten sus creaciones.

Permite la recopilación de métricas de multitud de fuentes, incluyendo Sistemas Gestores de bases de datos, hardware del sistema, runtimes u orquestadores de contenedores como Kubernetes, así como un largo etc.

Esta licenciada bajo licencia Apache 2.0, por lo que es de código abierto y gratuito.

Todo lo anterior convierte a Grafana en una aplicación muy útil para mantener vigilados los valores de nuestro servidor.

### 9.3.2 Stack Grafana. Prometheus

Grafana por sí, no es más que una plataforma de visualización que transforma datos recopilados por otras vías en datos fácilmente visualizables de forma agradable, pero por sí solo es incapaz de hacer nada. Es aquí donde entra en juego Prometheus.

Podemos definir a Prometheus como el sistema de recolección y monitorización de datos, que recopila, ordena y almacena las métricas que pretendemos monitorizar; para a continuación pasárselas a Grafana para su visualización. Es decir, sin un contenedor de Prometheus, Grafana no es más que una aplicación inútil sin fuente de datos.

Prometheus es uno de los sistemas de monitorización y alerta más populares de la industria, flexible y configurable. Es además de código abierto bajo licencia Apache 2.0 al igual que Grafana, por lo que era una elección obvia para esta parte del proyecto.

### 9.3.3 Stack Grafana. Node Exporter

Si bien este contenedor no estaba contemplado en un principio, el desarrollo del proyecto llevo a un amargo descubrimiento: Por defecto Prometheus no es capaz de recopilar las métricas de hardware que deseamos recopilar.

Puesto que la intención original de este pod era precisamente esa, se hacía necesaria la incorporación de algún medio para obtener esas métricas y representarlas en Grafana.

Es ahí donde entra Node Exporter.

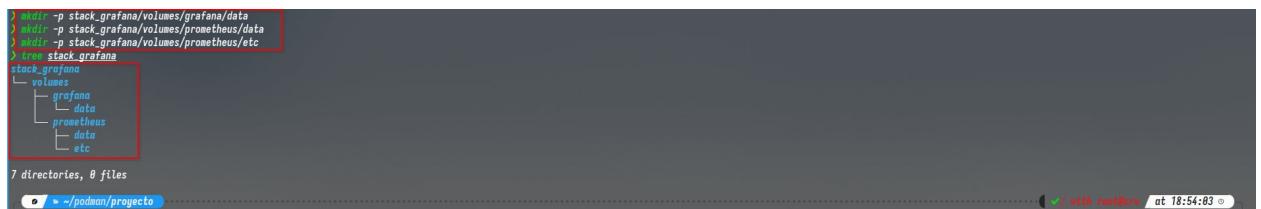
Se trata de un exportador de métricas de sistema, al pool de datos de Prometheus, el cual a su vez las transfiere a Grafana.

Al igual que las otras 2 piezas de este stack, Node Exporter esta licenciado bajo Apache 2.0, lo que lo hace ideal para usar en cualquier proyecto al ser de libre uso.

### 9.3.4 Stack Grafana. Despliegue y configuración básica

Llegamos al momento del despliegue, el cual documentaremos de la misma forma que con el anterior pod de LDAP.

Describiremos la creación de volúmenes, los parámetros del yaml de despliegue y registraremos los pasos de configuración básica para que la aplicación quede operativa.



```
└── stack_grafana
    └── volumes
        ├── grafana
        │   └── data
        ├── prometheus
        │   └── data
        └── etc
            └── stack_grafana
                └── grafana
                    └── data
                    └── prometheus
                        └── data
                        └── etc
7 directories, 0 files
```

Como es habitual comenzaremos creando la estructura de directorios para alojar los volúmenes del pod, siguiendo la estructura habitual como se muestra en la captura superior.



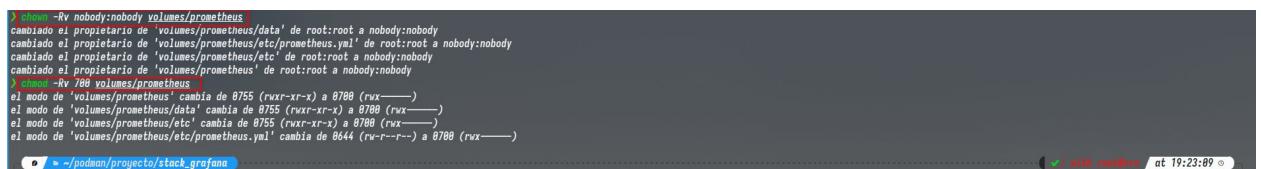
```
global:
  scrape_configs:
    - job_name: 'prometheus'
      static_configs:
        - targets: [ 'localhost:9090' ]
    - job_name: 'node'
      static_configs:
        - targets: [ 'srv.ismael-server.ddns.net:9100' ]
```

El siguiente paso es crear un archivo de configuración Yaml para prometheus en su volumen etc. En él se definen los trabajos que recopilara, incluido el futuro node exporter. Dicho yaml será incluido en la entrega del proyecto.



```
└── stack_grafana
    └── volumes
        ├── grafana
        └── prometheus
            └── etc
                └── prometheus
                    └── yaml
```

Al igual que con los contenedores del pod LDAP, debemos cambiar el propietario del volumen de Grafana al usuario y grupo 472:472, el usuario interno del contenedor.



```
└── stack_grafana
    └── volumes
        ├── grafana
        └── prometheus
            └── etc
                └── prometheus
                    └── yaml
```

Respecto al contenedor de Prometheus, este por diseño lee el contenido de los volúmenes como el usuario nobody, por lo que debemos cambiar el propietario a este mismo. Esta decisión de diseño está considerada por parte de la comunidad como un error y una brecha de seguridad, por lo que para mitigarla en la medida de lo posible restringiremos los permisos de dichos volúmenes exclusivamente a dicho usuario con un **chmod 700**.

```

GNU nano 7.2
stack_grafana.yaml
kind: Pod
metadata:
  labels:
    app: proyecto.stackgrafana
    name: proyecto_stackgrafana
spec:
  containers:
    - name: prometheus
      image: docker.io/prom/prometheus:latest
      args:
        - --web.listen-address=127.0.0.1:9091
      ports:
        - containerPort: 9090
          hostIP: 127.0.0.1
          hostPort: 9090
      resources: {}
    securityContext:
      capabilities:
        drop:
          - CAP_MKNOD
          - CAP_NET_RAW
          - CAP_AUDIT_WRITE
      volumeMounts:
        - mountPath: /etc/prometheus:Z
          name: prometheus-etc-host-0
        - mountPath: /prometheus:Z
          name: srv-prometheus-data-host-0

```

Preparados los directorios y permisos, pasamos a confeccionar el archivo yaml que se adjuntara a la entrega. Los argumentos de arranque son importantes en este pod, pues le indican a prometheus que ha de cambiar el puerto de escucha del por defecto 9090 a 9091; debido a que nuestro host ya tiene ocupado dicho puerto con la consola de administración cockpit. El argumento **config.file** especifica a prometheus donde buscar su archivo de configuración yaml, donde debemos especificar la ruta al mismo en el volumen creado.

Otro punto a destacar en este yaml es la presencia de “capabilities” o privilegios de sistema. Especificándolos en el apartado de contexto de seguridad bajo una etiqueta **drop**, se limitan ciertos privilegios de sistema a los contenedores del pod.

- **CAP\_MKNOD**: Deshabilita la creación de dispositivos especiales
- **CAP\_NET\_RAW**: Deshabilita capacidades de red de bajo nivel
- **CAP\_AUDIT\_WRITE**: Deshabilita la capacidad para escribir en el log del sistema

```

GNU nano 7.2
stack_grafana.yaml
- name: grafana
  image: docker.io/grafana/grafana:latest
  ports:
    - containerPort: 3000
      hostIP: 127.0.0.1
      hostPort: 3000
  resources: {}
  securityContext:
    capabilities:
      drop:
        - CAP_MKNOD
        - CAP_NET_RAW
        - CAP_AUDIT_WRITE
    privileged: false
  volumeMounts:
    - mountPath: /var/lib/grafana:Z
      name: srv-grafana-data-host-0

```

La declaración del contenedor Grafana es bastante simple. Contiene su nombre, imagen, los mismos privilegios desactivados y el volumen declarado más adelante.

```

- args:
    - --path.rootfs=/host
  image: docker.io/prom/node-exporter:latest
  name: node_exporter
  securityContext:
    capabilities:
      drop:
        - CAP_MKNOD
        - CAP_NET_RAW
        - CAP_AUDIT_WRITE
    volumeMounts:
      - mountPath: /host
        name: root-host-0
        readOnly: true
  restartPolicy: unless-stopped
  volumes:
    - hostPath:
        path: /root/padman/proyecto/stack_grafana/volumes/prometheus/etc
        type: Directory
        name: srv-prometheus-etc-host-0
    - hostPath:
        path: /root/padman/proyecto/stack_grafana/volumes/prometheus/data
        type: Directory
        name: srv-prometheus-data-host-0
    - hostPath:
        path: /
        type: Directory
        name: root-host-0
        enableServiceLinks: false
        hostNetwork: true

```

Por último, el contenedor de node-exporter se define igual que los demás con su nombre e imagen, mismos permisos deshabilitados. Sin embargo, su volumen es la raíz del host, pues

necesita acceder a los registros del mismo para poder recopilar las métricas. Véase que su asignación de volumen tiene una etiqueta adicional **readOnly** que le da permiso exclusivamente a leer, pero no escribir.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8a8e02159488	localhost/podman-pause:4.9.4-1711445992	/usr/local/bin/st...	About a minute ago	Up About a minute		f3592f4ed5a-infra
98fe0e177345	ghcr.io/ldapaccountmanager/lam:latest	/usr/local/bin/st...	About a minute ago	Up About a minute		projeto_stackldap-lam
76997317484f	docker.io/ossia/openidap:1.5.0		About a minute ago	Up About a minute		projeto_stackldap-openidap
9779142882c	localhost/podman-pause:4.9.4-1711445992		About a minute ago	Up About a minute		d942a8a1b6f2-infra
7228369e1d65	docker.io/library/nextcloud:stable-apache	apache2-foreground...	About a minute ago	Up 59 seconds		projeto_stacknc-nextcloud
ae23a77fd295	docker.io/library/postgres:latest	postgres	About a minute ago	Up 59 seconds		projeto_stacknc-postgres
2b845a5a6b8b	docker.io/dpage/pgadmin4:latest		About a minute ago	Up 58 seconds		projeto_stacknc_pgadmin

Finalmente, listo nuestro manifiesto yaml de despliegue del pod, ya solo nos queda ejecutar el comando **podman kube play** que hemos realizado con el resto de pods para completar el despliegue del stack de grafana.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8a8e02159488	localhost/podman-pause:4.9.4-1711445992	/usr/local/bin/st...	About a minute ago	Up About a minute		f3592f4ed5a-infra
98fe0e177345	ghcr.io/ldapaccountmanager/lam:latest	/usr/local/bin/st...	About a minute ago	Up About a minute		projeto_stackldap-lam
76997317484f	docker.io/ossia/openidap:1.5.0		About a minute ago	Up About a minute		projeto_stackldap-openidap
9779142882c	localhost/podman-pause:4.9.4-1711445992		About a minute ago	Up About a minute		d942a8a1b6f2-infra
7228369e1d65	docker.io/library/nextcloud:stable-apache	apache2-foreground...	About a minute ago	Up 59 seconds		projeto_stacknc-nextcloud
ae23a77fd295	docker.io/library/postgres:latest	postgres	About a minute ago	Up 59 seconds		projeto_stacknc-postgres
2b845a5a6b8b	docker.io/dpage/pgadmin4:latest		About a minute ago	Up 58 seconds		projeto_stacknc_pgadmin

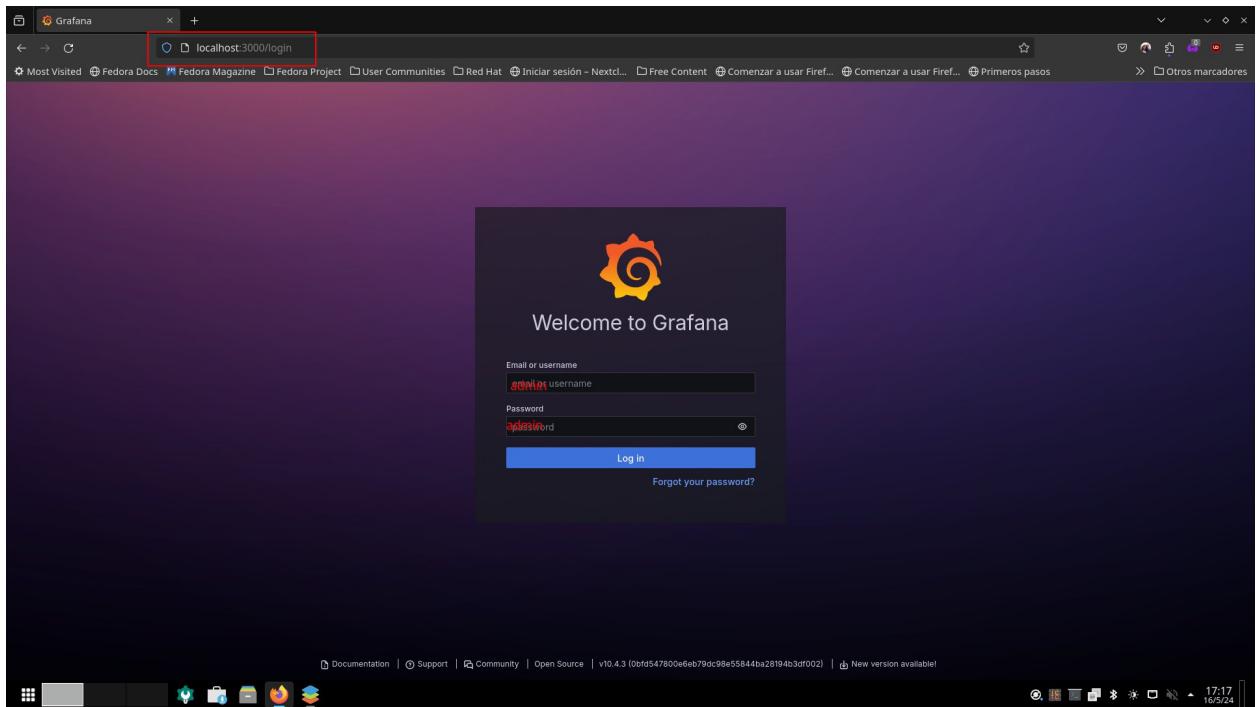
  

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1d858fc7a4a1592dbbb1fd3e2dee644fb5735839e0d76c822717a14fe4592a						
Containers:						
2d05c6e4d38ba58c9c7d7a66b39f1fca8bf74a6a9b1d224c3acbf19174c						
aee0b85bd8c1ad71cfdb8c884527424df363471fa6d3a2392329346ceba3						
0016991fb0369c24d3ac826f5a341825918dec59b98fffa0833c3b42ed56738						

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8a8e02159488	localhost/podman-pause:4.9.4-1711445992	/usr/local/bin/st...	3 minutes ago	Up 3 minutes		f3592f4ed5a-infra
98fe0e177345	ghcr.io/ldapaccountmanager/lam:latest	/usr/local/bin/st...	3 minutes ago	Up 3 minutes		projeto_stackldap-lam
76997317484f	docker.io/ossia/openidap:1.5.0		3 minutes ago	Up 3 minutes		projeto_stackldap-openidap
9779142882c	localhost/podman-pause:4.9.4-1711445992		2 minutes ago	Up 2 minutes		d942a8a1b6f2-infra
7228369e1d65	docker.io/library/nextcloud:stable-apache	apache2-foreground...	2 minutes ago	Up 2 minutes		projeto_stacknc-nextcloud
ae23a77fd295	docker.io/library/postgres:latest	postgres	2 minutes ago	Up 2 minutes		projeto_stacknc-postgres
2b845a5a6b8b	docker.io/dpage/pgadmin4:latest		2 minutes ago	Up 2 minutes		projeto_stacknc_pgadmin
829e782bde4c	localhost/podman-pause:4.9.4-1711445992		8 seconds ago	Up 5 seconds		1d858fc7a4a1-infra
2d05c6e4d38ba58c9c7d7a66b39f1fca8bf74a6a9b1d224c3acbf19174c		--web.listen-address...	7 seconds ago	Up 5 seconds		projeto_stackgrafana-prometheus
aee0b85bd8c1ad71cfdb8c884527424df363471fa6d3a2392329346ceba3			5 seconds ago	Up 4 seconds		projeto_stackgrafana-grafana
0016991fb0369c24d3ac826f5a341825918dec59b98fffa0833c3b42ed56738		--path.rootfs=/ho...	4 seconds ago	Up 4 seconds		projeto_stackgrafana-node_exporter

De esta forma el pod comienza a ejecutarse y está listo para que le hagamos su configuración básica.



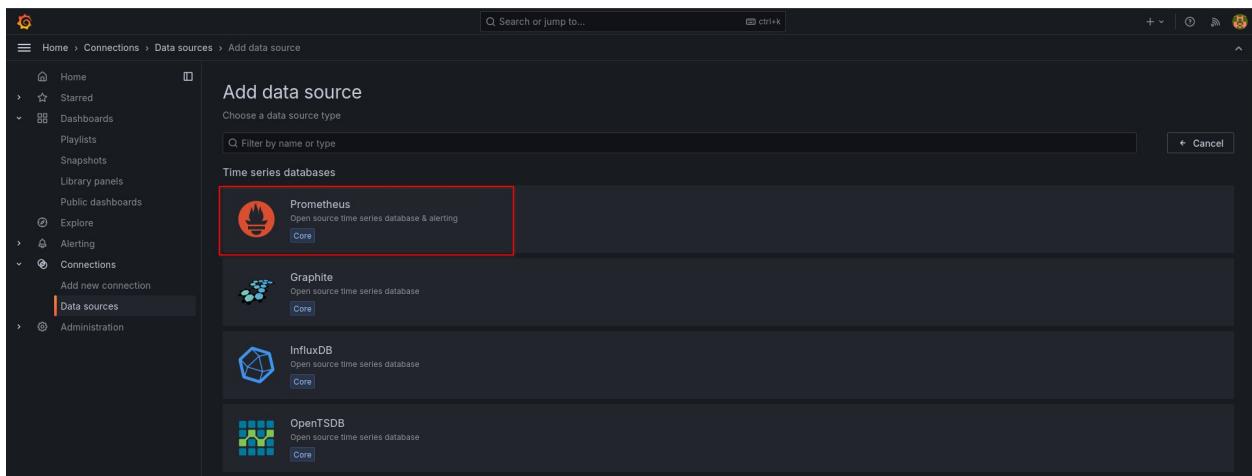
Ahora ya podemos entrar a grafana a través de su puerto de escucha (3000) y loguearnos con el usuario administrador. Por defecto las credenciales de Grafana son **admin** y **admin** tanto para contraseña como para usuario.



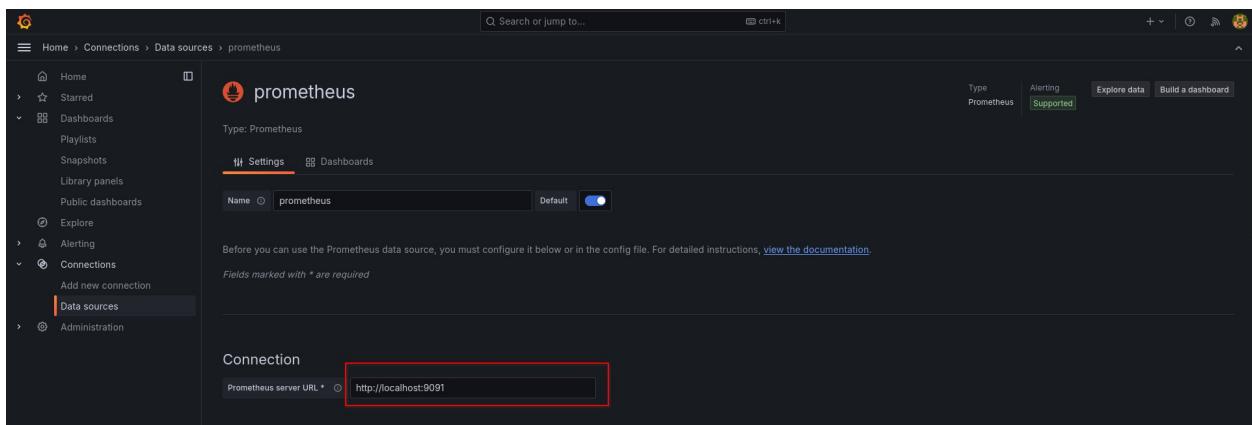
En el momento en el que introducimos dichas credenciales, grafana nos solicita que cambiemos la contraseña del administrador por una más segura, por lo que procedemos a hacerlo.

Ya estamos en el panel principal de Grafana, y lo primero que debemos hacer es dirigirnos a la sección de fuentes de datos (data sources) en la columna izquierda.

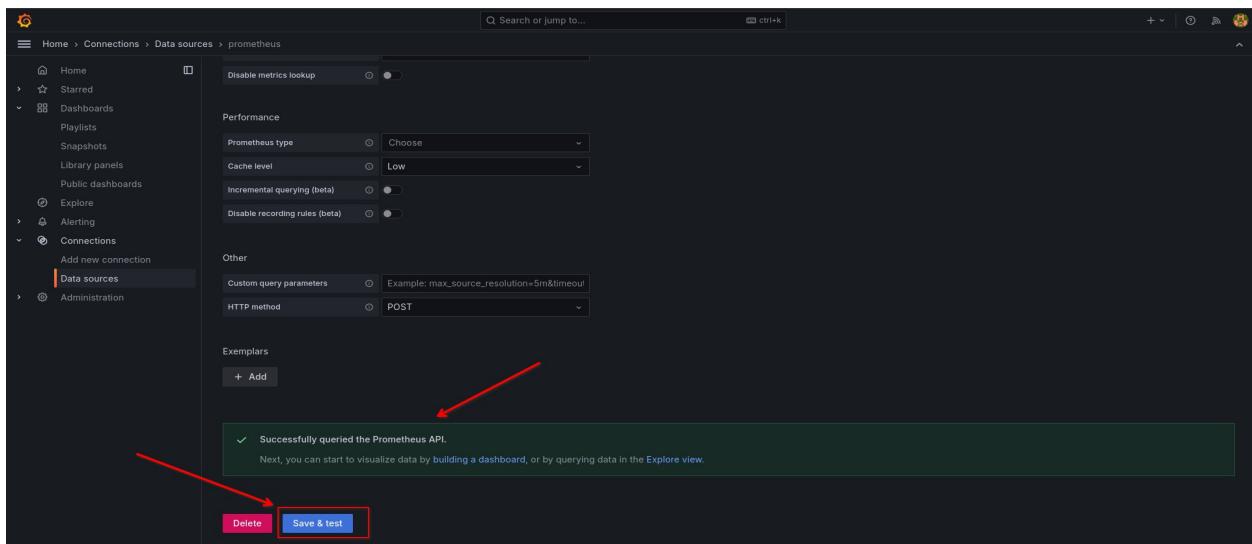
Pulsamos sobre **Add data source...**



Pulsamos sobre la fuente de datos que hemos desplegado junto con Grafana, **Prometheus**.



En el apartado de conexión con Prometheus, debemos introducir la url hacia el host local por el puerto en el que hemos configurado prometheus durante el despliegue con su archivo de configuración yaml, <http://localhost:9091>



Hacemos scroll hasta la parte baja del panel de Prometheus y pulsamos sobre "Guardar y probar". Si todo funciona correctamente, un aviso nos confirmara que se ha establecido correctamente la configuración con Prometheus y la conexión quedara guardada.

Welcome to Grafana

Basic

TUTORIAL DATA SOURCE AND DASHBOARDS  
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE Add your first data source

DASHBOARDS Create your first dashboard

Ahora necesitamos un Dashboard que muestre los datos que estamos recopilando. Pulsamos sobre el apartado de Dashboards en la columna izquierda.

Home > Dashboards

Dashboards

Create and manage dashboards to visualize your data

Search for dashboards and folders

No dashboards yet. Create your first!

+ Create Dashboard

Podríamos crear nuestro propio dashboard, pero es una labor relativamente compleja y larga que queda fuera de las competencias de este proyecto; sin embargo existen multitud de dashboards compartidos por la comunidad que pueden ser utilizados libremente ([Biblioteca de Dashboards de Grafana](#)), por lo que usaremos dicha opción. Pulsaremos en la opción para importar en el desplegable “Nuevo” de la esquina superior derecha.

Home > Dashboards > Import dashboard

Import dashboard

Import dashboard from file or Grafana.com

Upload dashboard JSON file

Drag and drop here or click to browse Accepted file types: .json, .txt

Find and import dashboards for common applications at [grafana.com/dashboards](#)

13978

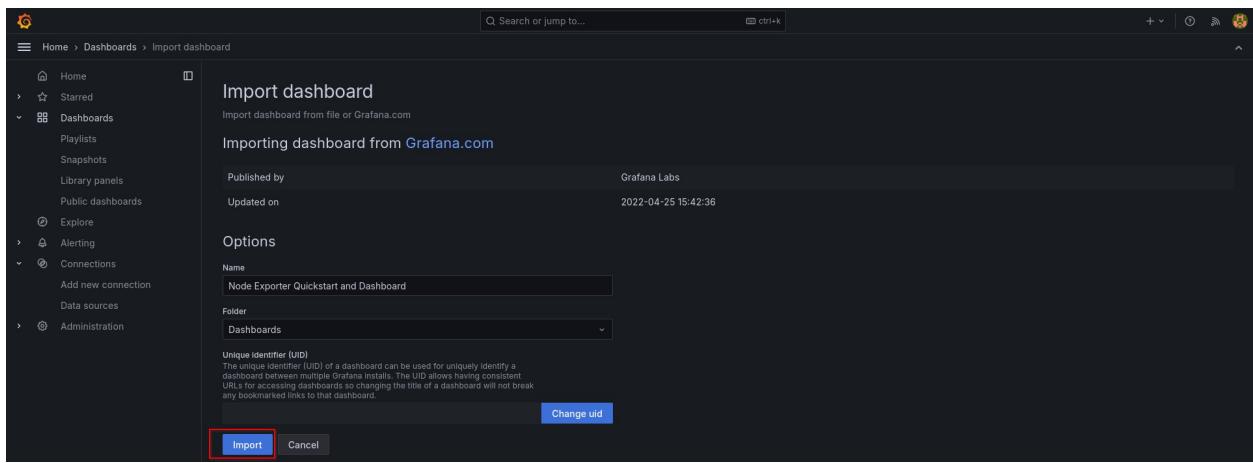
Load

Import via dashboard JSON model

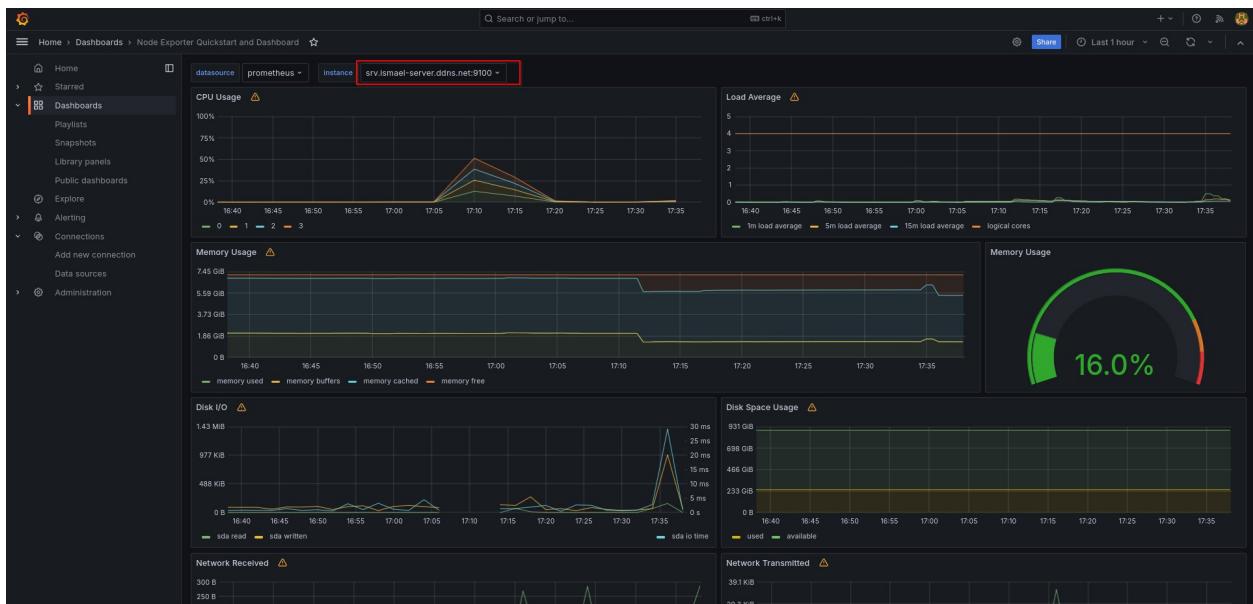
```
{
  "title": "Example - Repeating Dictionary variables",
  "uid": "OHnEoM4z",
  "panels": [...]
}
```

Load Cancel

Una vez hayamos elegido un Dashboard acorde a nuestras necesidades en la biblioteca de grafana, introduciremos su ID en el cuadro de búsqueda del panel de importación y pulsamos en cargar.



Pulsamos en importar...



Y ¡¡listo!! Ya tenemos un dashboard de monitorización gracias que NodeExporter está recopilando métricas del sistema y enviándolas a Grafana. A partir de ahora podremos detectar consumos de recursos, picos inusuales en el hardware e incluso configurar alertas.

## 9.4 Contenedor Caddy Server

Por el momento hemos desplegado todas las aplicaciones de nuestro servidor y ya están listas para ser usadas pero, el usuario atento habrá reparado en un pequeño detalle que aún no se ha resuelto: Hasta ahora accedíamos a dichas aplicaciones bien a través de la red local del servidor apuntando a cada uno de sus puertos de escucha, o bien mediante túneles ssh si necesitábamos acceder remotamente. Además, ninguna de las aplicaciones desplegadas tiene tráfico cifrado HTTPS o un certificado válido firmado por alguna autoridad de certificación válida.

Solo disponemos de un nombre de dominio (ismael-server.ddns.net) con wildcard y cada aplicación web tiene su propio servidor web independiente así que... ¿Cómo poder asignar distintos subdominios a cada una de las aplicaciones sin tener que abrir puertos para cada una de las aplicaciones?

La respuesta es **Caddy Server**.

### 9.4.1 Caddy server. El servidor multiuso

Podemos considerar a Caddy Server como una auténtica navaja suiza de los servidores.

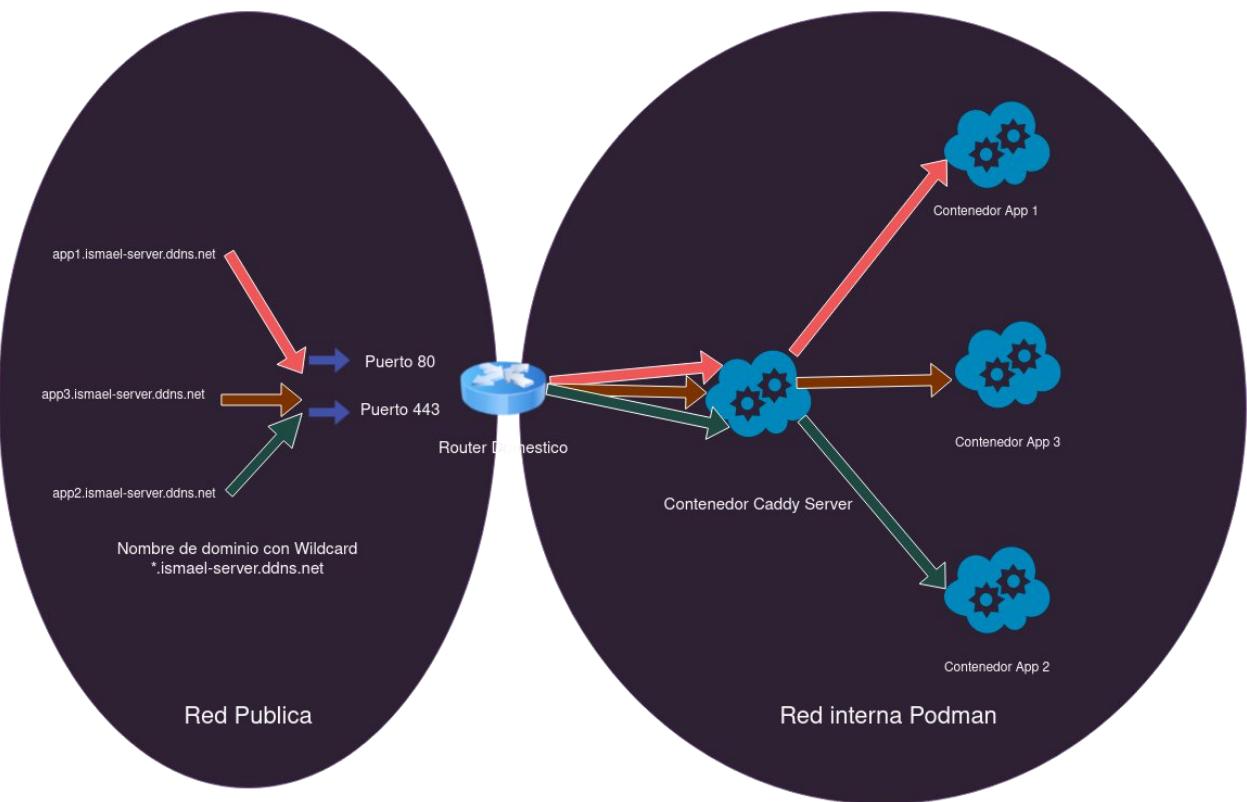
Esta pequeña y ligera pieza de software, pensada principalmente como servidor web ligero y fácil de configurar, puede ser configurado como servidor de archivos, balanceador de carga y proxy inverso.

Es extremadamente sencillo de configurar a través de directivas en un único archivo de configuración: el **Caddyfile**, y posee una característica increíblemente útil: Negociación, obtención y renovación automática de certificados HTTPS mediante Let's encrypt y ZeroSSL para todas las webs que aloje o redirija.

Es además software libre licenciado bajo Apache 2.0, por lo que es una elección más que obvia para las necesidades de este proyecto.

## 9.4.2 Caddy Server. Esquema lógico de la red

Antes de proceder a su despliegue, el autor considera que es una buena idea ofrecer al lector un esquema visual a cerca de cómo es el flujo de red desde el exterior hasta los contenedores. Se ofrece a continuación pues, un esquema del flujo de red de las solicitudes a nuestro servidor.



El uso de un proxy inverso como Caddy nos proporciona una conveniente forma de poder utilizar distintos subdominios para redireccionar a cada una de las aplicaciones o consolas de administración de nuestro servidor, mientras al mismo tiempo cifra todas las conexiones mediante SSL. Su uso tiene además la ventaja añadida de eliminar la necesidad de exponer al exterior la multitud de puertos que nuestras aplicaciones usan (9090 para Cockpit, 10000 para Webmin, 3000 para Grafana etc), quedando estos restringidos únicamente a los puertos 80 y 443 para el protocolo HTTP/S y un puerto para las conexiones por ssh redirigido al 22 interno.

## 9.4.3 Caddy Server. Aprovisionamiento del Caddy file y despliegue del contenedor

Llega el momento de desplegar el que será nuestro último contenedor.

A diferencia del resto de nuestra pila de software, Caddy será desplegado como un contenedor aislado, pues está relacionado con todos los pods y a la vez no está relacionado “conceptualmente” con ninguno de ellos. Pero antes, debemos crear sus volúmenes y aprovisionar su sencillo archivo de configuración, el Caddyfile.

```
# cd /etc/caddy
# mkdir -p caddy_proxy/volumes/config
# mkdir -p caddy_proxy/volumes/data
# touch caddy_proxy/volumes/Caddyfile
# tree caddy_proxy
caddy_proxy
└── volumes
    ├── Caddyfile
    └── config
        └── data
4 directories, 1 file
```

Como con el resto de nuestro software, Caddy necesita sus volúmenes para tener persistencia. Creamos la habitual estructura de directorios con sendos **mkdir** y **touch** para generar un archivo Caddyfile vacío en la raíz de volúmenes.

```
GNU nano 7.2
# Directivas
asirprnc.ismael-server.ddns.net {
    reverse_proxy proyecto_stacknc:80
}

# LDAP Account Manager
asirprolm.ismael-server.ddns.net {
    reverse_proxy proyecto_stackldap:80
}

# Consola Webmin
asirproxw.ismael-server.ddns.net {
    reverse_proxy 192.168.1.120:10800
}

# Consola Cockpit
asirpropcc.ismael-server.ddns.net {
    reverse_proxy 192.168.1.120:9090
}

# PGAdmin
asirpropga.ismael-server.ddns.net {
    reverse_proxy proyecto_stacknc:81
}

# Grafana
asirprogrf.ismael-server.ddns.net {
    reverse_proxy proyecto_stackgrafana:3000
}
```

Aprovisionar el Caddyfile para que actué como Proxy inverso es extremadamente fácil. Tan solo debemos crear una entrada por cada “host” que queramos redirigir con el siguiente formato como se aprecia en la captura superior:

```
#Comentario

nombre.de.dominio.o.subdominio {

    reverse_proxy direcciónIP/nombreDNS:puerto

}
```

Debemos crear una entrada para cada aplicación o Host al que queramos redirigir peticiones, incluyendo en cada caso su FQDN e introducir entre llaves {} la directiva `reverse_proxy` con el host a continuación y su puerto de escucha.

Caddy admite tanto direcciones IP como nombres DNS.

```

podman run -d
--name=projeto_caddyproxy \
--network=principal \
--network=projecto \
-p 80:80 \
-p 443:443 \
-v /root/podman/projeto/caddy_proxy/volumes/Caddyfile:/etc/caddy/Caddyfile:z \
-v /root/podman/projeto/caddy_proxy/volumes/data:/data:z \
-v /root/podman/projeto/caddy_proxy/volumes/config:/config:z \
caddy:latest

```

Una vez aprovisionado el Caddyfile, ejecutamos un comando de arranque de contenedor simple con **podman run** y el parámetro **-d** para la ejecución desacoplada (segundo plano), especificando la red a la que conectara el contenedor, que debe ser la misma a la que están conectados nuestros pods (También se conecta a mi red principal para conectar otros contenedores de uso personal); así como el montaje de sus volúmenes para la persistencia, la imagen a usar y los puertos a exponer.

Respecto a estos últimos, es importante que Caddy no tenga restricciones de salida y escucha a los puertos 80 y 443, pues de otra forma no será capaz de negociar, obtener y renovar los certificados para el tráfico cifrado HTTPS.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8a8e2150498	localhost/podman-pause:4.9.4-1711445992	/usr/local/bin/st...	18 hours ago	Up 37 minutes		f3592faed6a-infra
9f7fe017345	ghcr.io/distrobuildanger/luci:latest		18 hours ago	Up 37 minutes		projeto_stackluci-luci
75997317484	docker.io/cisis/openldap:1.5.8		18 hours ago	Up 37 minutes		projeto_stackluci-openldap
9777e14282c	localhost/podman-pause:4.9.4-1711445992		18 hours ago	Up 37 minutes		d92a8a1b6f2-infra
7228369e1de5	docker.io/library/nextcloud:stable-apache	apache2-foreground...	18 hours ago	Up 37 minutes		projeto_stacknc-nextcloud
ae23a7ff295	docker.io/library/postgres:latest	postgres	18 hours ago	Up 37 minutes		projeto_stacknc-postgres
2b845a5a5bb8	docker.io/dapgmnl4:latest		18 hours ago	Up 37 minutes		projeto_stacknc-pgadmin
B29e782bdc4	localhost/podman-pause:4.9.4-1711445992		18 hours ago	Up 37 minutes		1d858fc7c4a01-infra
2d05c6dd38b	docker.io/prom/prometheus:latest	--web.listenAddr...	18 hours ago	Up 37 minutes		projeto_stackgrafana-prometheus
aeeb055bd8c	docker.io/grafana/grafana:latest		18 hours ago	Up 37 minutes		projeto_stackgrafana-grafana
0016091fb836	docker.io/node-exporter:latest	--path.rootfs=/no...	18 hours ago	Up 37 minutes		projeto_stackgrafana-node_exporter
fcabddc1eb0b3	docker.io/library/caddy:latest	caddy run -conf...	11 seconds ago	Up 11 seconds	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp	projeto_caddyproxy

POD ID	NAME	STATUS	CREATED	INFRA ID	# OF CONTAINERS
1d858fc7e4a1	projeto_stackgrafana	Running	18 hours ago	829e782bdc2	4
d942ada1b6f2	projeto_stacknc	Running	18 hours ago	9779e14282c	4
f3592faed6a	projeto_stackluci	Running	18 hours ago	8a8e2150498	3

Caddy es el único contenedor de toda la pila que expone puertos al exterior

Y ¡¡listo!! Caddy ha sido desplegado. Este es el aspecto final de toda nuestra pila de software, que a partir de ahora solo expondrá los puertos 80 y 443, quedando todos los demás restringidos a la red interna virtual de podman. A partir de ahora, Caddy se encargará de encaminar las peticiones a cada una de nuestras aplicaciones.

```

container logs projeto_caddyproxy
[{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "caddy", "msg": "using provided configuration", "config_file": "/etc/caddy/Caddyfile", "config_adapter": "caddyfile"}, {"[{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "caddy", "msg": "Caddyfile input is not formatted; run `caddy fati...overr...` to fix inconsistencies", "adapter": "caddyfile", "file": "/etc/caddy/Caddyfile", "line": 3}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "caddy", "msg": "admin endpoint started", "address": "localhost:2019", "enforce_origin": false, "origins": "[\"/localhost:2019\", \"/[::1]:2019\", \"/127.0.0.1:2019\"]"}, {"[{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http_auto_https", "msg": "Server is Listening only on the HTTPS port but has no TLS connection policies; adding one to enable TLS", "server_name": "srv0", "https_port": 443}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http_auto_https", "msg": "enabling automatic HTTP->HTTPS redirects", "server_name": "srv0"}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "started background certificate maintenance", "cache": "0x0000166800"}, {"[{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "enabling HTTP/2 listen", "addr": ":443"}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "failed to sufficiently increase receive buffer size (was: 2048 kB, wanted: 2048 kB, got: 416 kB). See https://github.com/quic-go/quic-go/wiki/UDP-Buffer-Sizes for details."}, {"[{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "server running", "name": "srv0", "protocols": ["h1", "h2", "h3"]}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "remaining auto_https_redirects", "protocols": ["h1", "h2", "h3"]}], [{"level": "info", "ts": "2019-07-17T15:35:37.739+00:00", "logger": "http", "msg": "enabling automatic TLS certificate management", "domains": ["asirprogrf.ismael-server.ddns.net", "asirprnc.ismael-server.ddns.net", "asirprow.ismael-server.ddns.net", "asirpro"}]

```

Si lanzamos un comando **podman container logs nombrecontenedor**, podemos ver los registros de Caddy para ver cómo va la negociación y obtención de certificados (puede tardar unos minutos tras el despliegue)

```

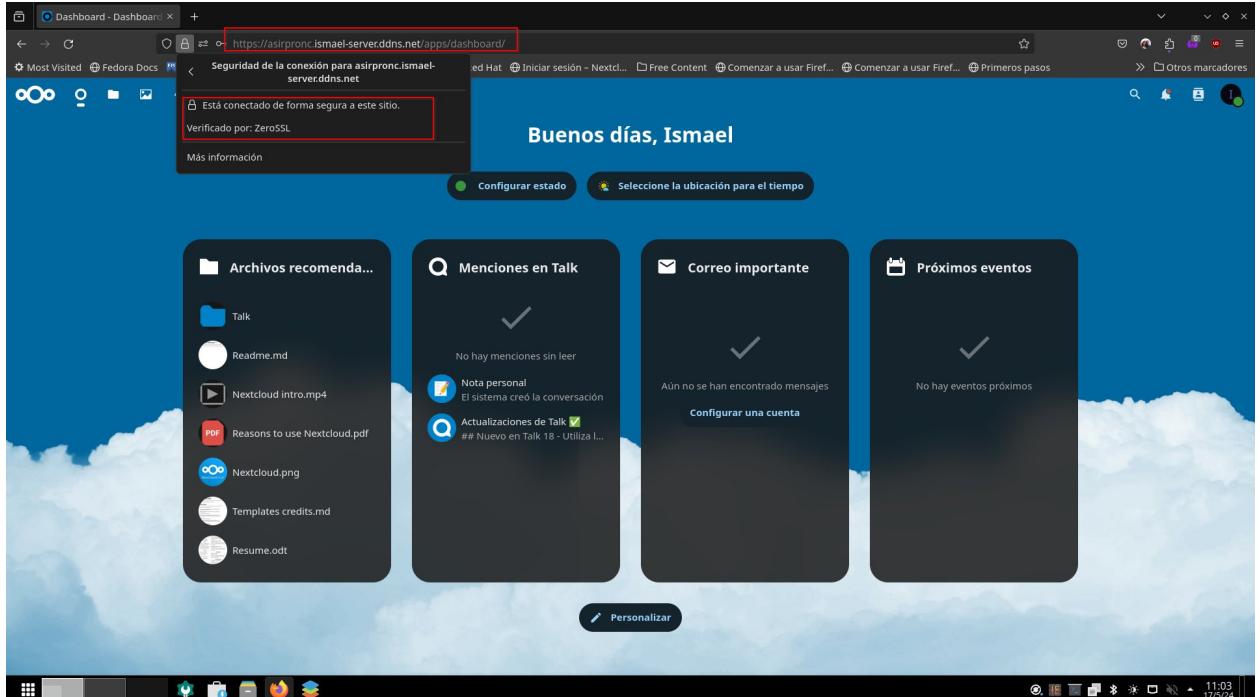
{"level": "info", "ts": "1715935829.1119354", "logger": "http.acme.client", "msg": "trying to solve challenge", "identifier": "asirpronc.ismael-server.ddns.net", "challenge_type": "http-01", "ca": "https://acme.zerossl.com/v2/DV98"}
{"level": "info", "ts": "1715935829.1387894", "logger": "http.acme.client", "msg": "trying to solve challenge", "identifier": "asirpronc.ismael-server.ddns.net", "challenge_type": "http-01", "ca": "https://acme.zerossl.com/v2/DV98"}
{"level": "info", "ts": "1715935829.1796787", "logger": "http.acme.client", "msg": "trying to solve challenge", "identifier": "asirpronc.ismael-server.ddns.net", "challenge_type": "http-01", "ca": "https://acme.zerossl.com/v2/DV98"}
{"level": "info", "ts": "1715935838.8872495", "logger": "http", "msg": "served key authentication", "identifier": "asirprop.ismael-server.ddns.net", "challenge": "http-01", "remote": "91.199.212.132:56516", "distributed": "false"}
{"level": "info", "ts": "1715935838.9387241", "logger": "http", "msg": "served key authentication", "identifier": "asirpronc.ismael-server.ddns.net", "challenge": "http-01", "remote": "91.199.212.132:56522", "distributed": "false"}
{"level": "info", "ts": "1715935838.3581414", "logger": "http.acme.client", "msg": "authorization finalized", "identifier": "asirprop.ismael-server.ddns.net", "authz_status": "valid"}
{"level": "info", "ts": "1715935838.3981404", "logger": "http.acme.client", "msg": "validations succeeded", "finalizing_order": "order": "https://acme.zerossl.com/v2/DV98/order/GtYvlgJzbJK0_uF0-EQQ"}
{"level": "info", "ts": "1715935838.4004404", "logger": "http.acme.client", "msg": "authorization finalized", "identifier": "asirpronc.ismael-server.ddns.net", "authz_status": "valid"}
{"level": "info", "ts": "1715935838.3728634", "logger": "http.acme.client", "msg": "validation succeeded", "finalizing_order": "order": "https://acme.zerossl.com/v2/DV98/order/0y0f1hL7Pz25jJBuUkRA"}
{"level": "info", "ts": "1715935838.5246944", "logger": "http.acme.client", "msg": "authorization finalized", "identifier": "asirpronc.ismael-server.ddns.net", "authz_status": "valid"}
{"level": "info", "ts": "1715935838.5258947", "logger": "tls.obtain", "msg": "certificate obtained successfully", "identifier": "asirprop.ismael-server.ddns.net"}
{"level": "info", "ts": "1715935838.5587728", "logger": "tls.obtain", "msg": "releasing lock", "identifier": "asirpronc.ismael-server.ddns.net"}
{"level": "info", "ts": "1715935838.5258947", "logger": "tls.obtain", "msg": "successfully downloaded available certificate chains", "count": "1", "first_url": "https://acme.zerossl.com/v2/DV98/cert/gexLfzYqCJz8gH8iqFarA"}
{"level": "info", "ts": "1715935838.5592315", "logger": "tls.obtain", "msg": "certificate obtained successfully", "identifier": "asirprop.ismael-server.ddns.net"}
{"level": "info", "ts": "1715935838.6487578", "logger": "http.acme.client", "msg": "successfully downloaded available certificate chains", "count": "1", "first_url": "https://acme.zerossl.com/v2/DV98/cert/oWxWjrNmCdp7dCzrEtTw"}
{"level": "info", "ts": "1715935838.6487578", "logger": "tls.obtain", "msg": "certificate obtained successfully", "identifier": "asirpronc.ismael-server.ddns.net"}
{"level": "info", "ts": "1715935851.6481799", "logger": "tls.obtain", "msg": "releasing lock", "identifier": "asirpronc.ismael-server.ddns.net"}
```

Tras una breve espera, los logs muestran que Caddy ya ha negociado y obtenido satisfactoriamente los certificados.

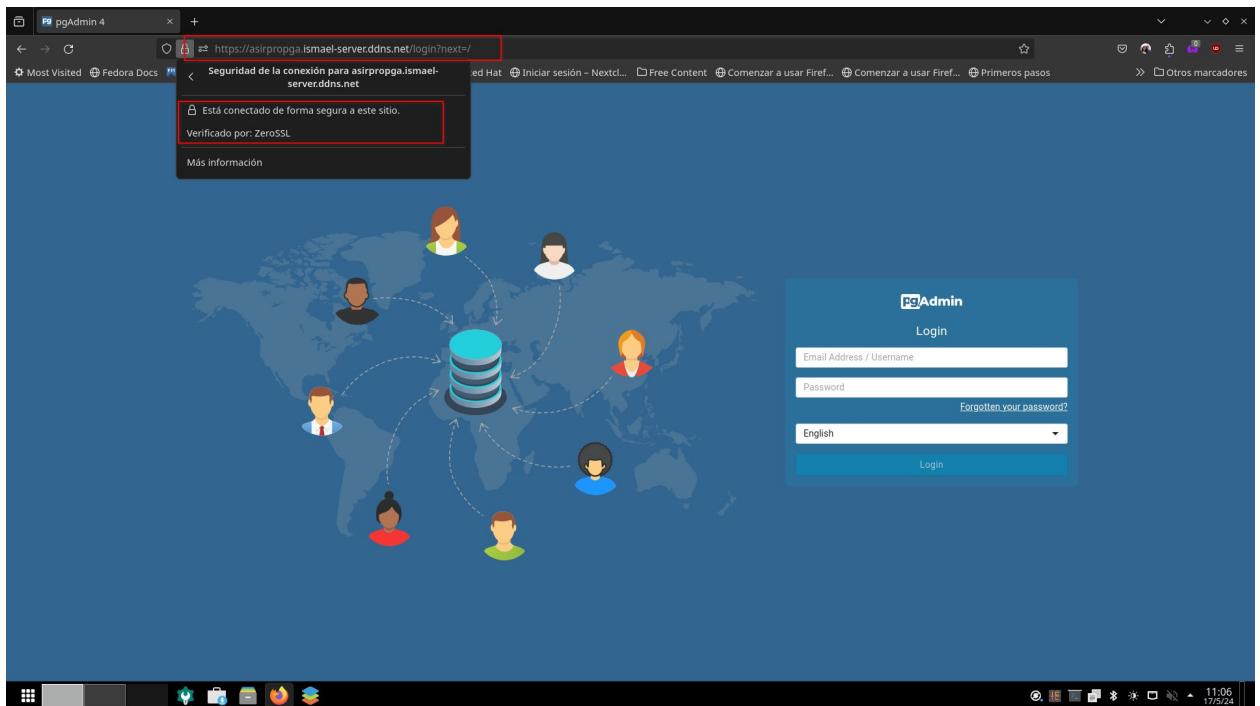
Ya solo nos queda comprobar que todo funciona a la perfección y todas nuestras peticiones son redirigidas a nuestras aplicaciones, y el tráfico es cifrado.

A continuación, se muestran capturas accediendo a cada una de las aplicaciones desplegadas, con su URL en la barra de navegación e información del certificado.

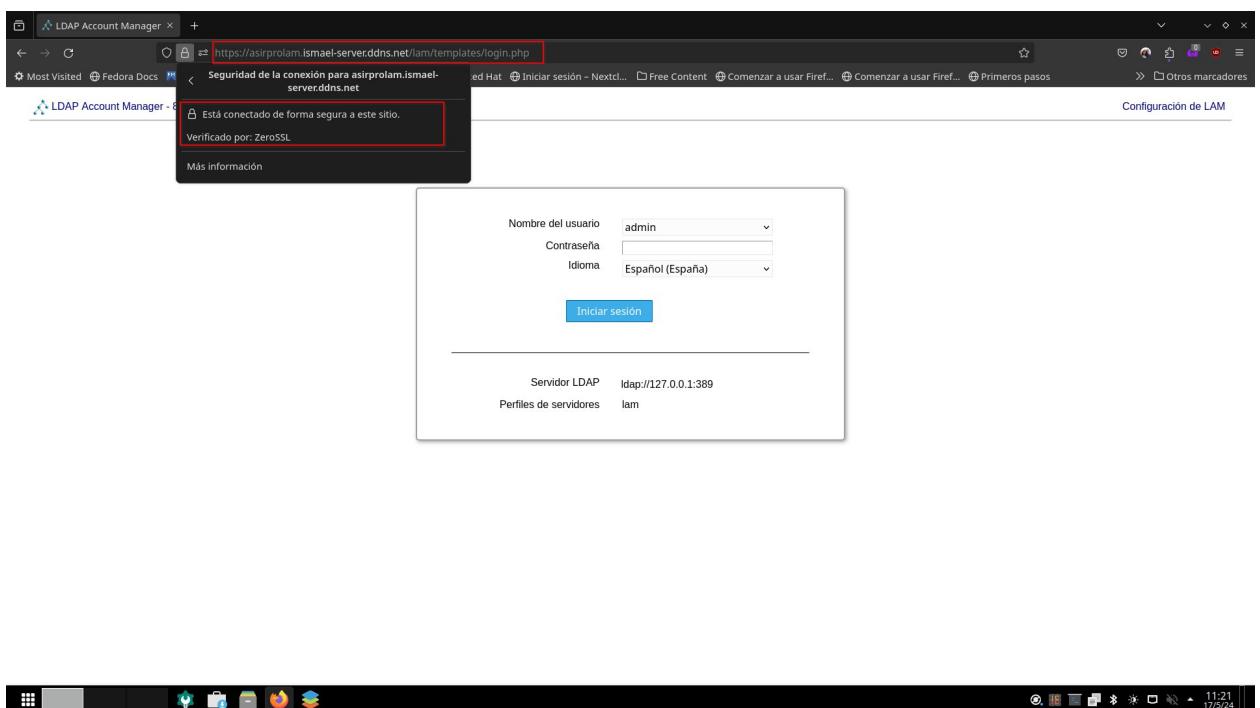
Nextcloud: <https://asirpronc.ismael-server.ddns.net>



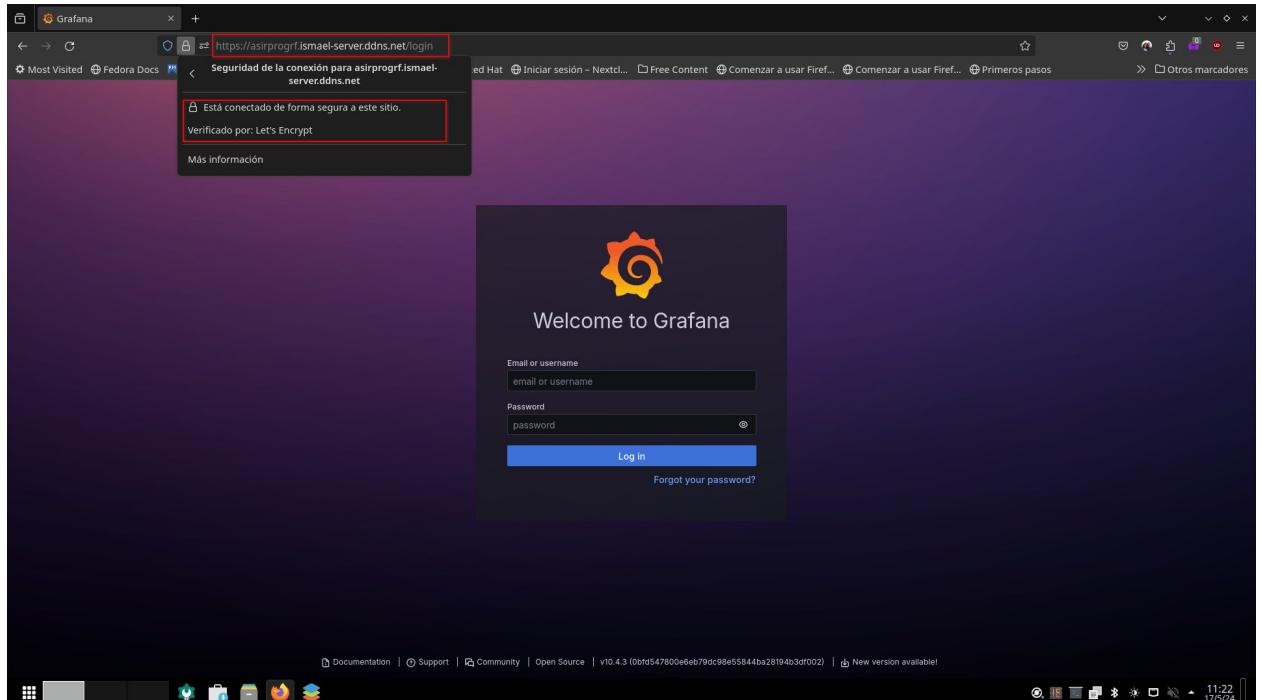
[PGAdmin: https://asirpropga.ismael-server.ddns.net](https://asirpropga.ismael-server.ddns.net)



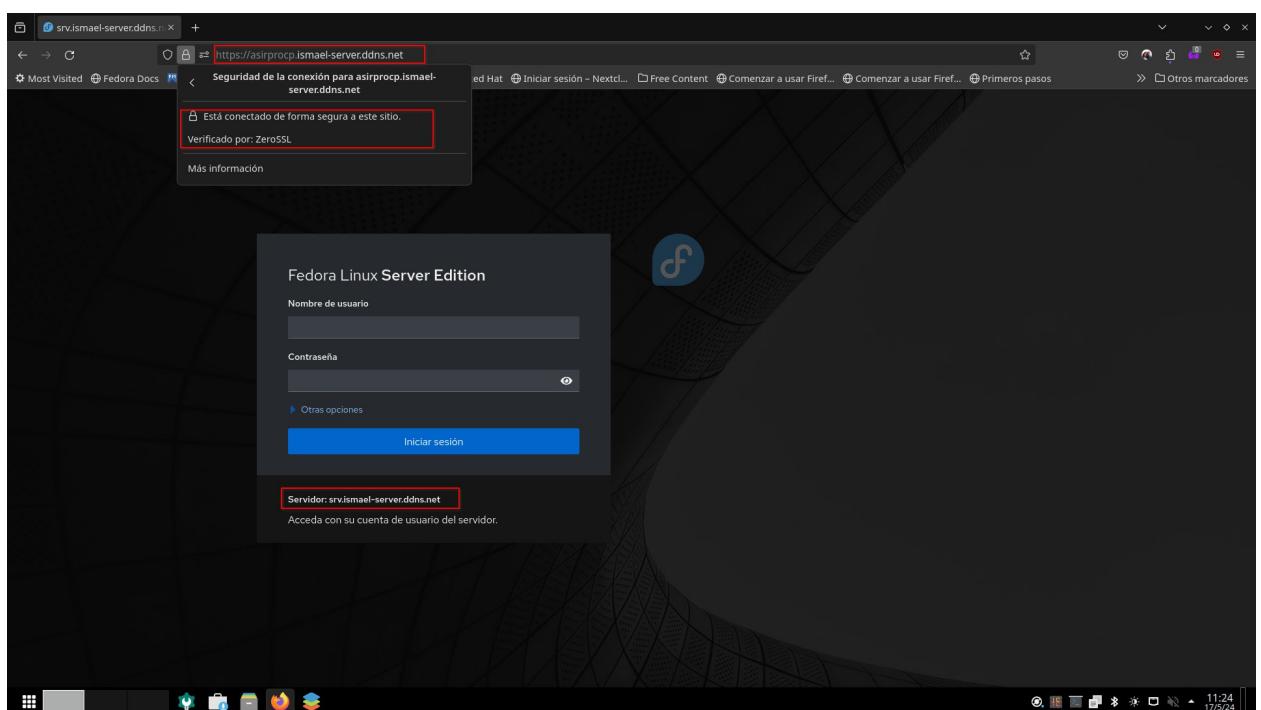
[LDAP Account Manager: https://asirprolam.ismael-server.ddns.net](https://asirprolam.ismael-server.ddns.net)



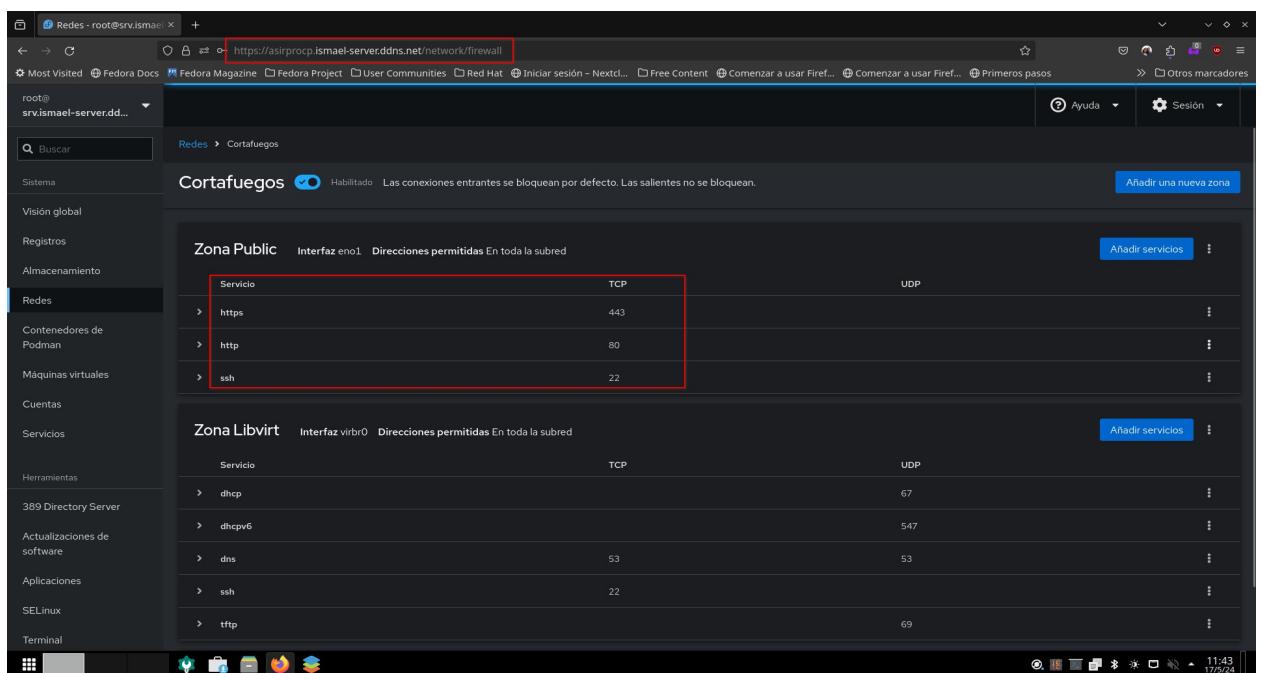
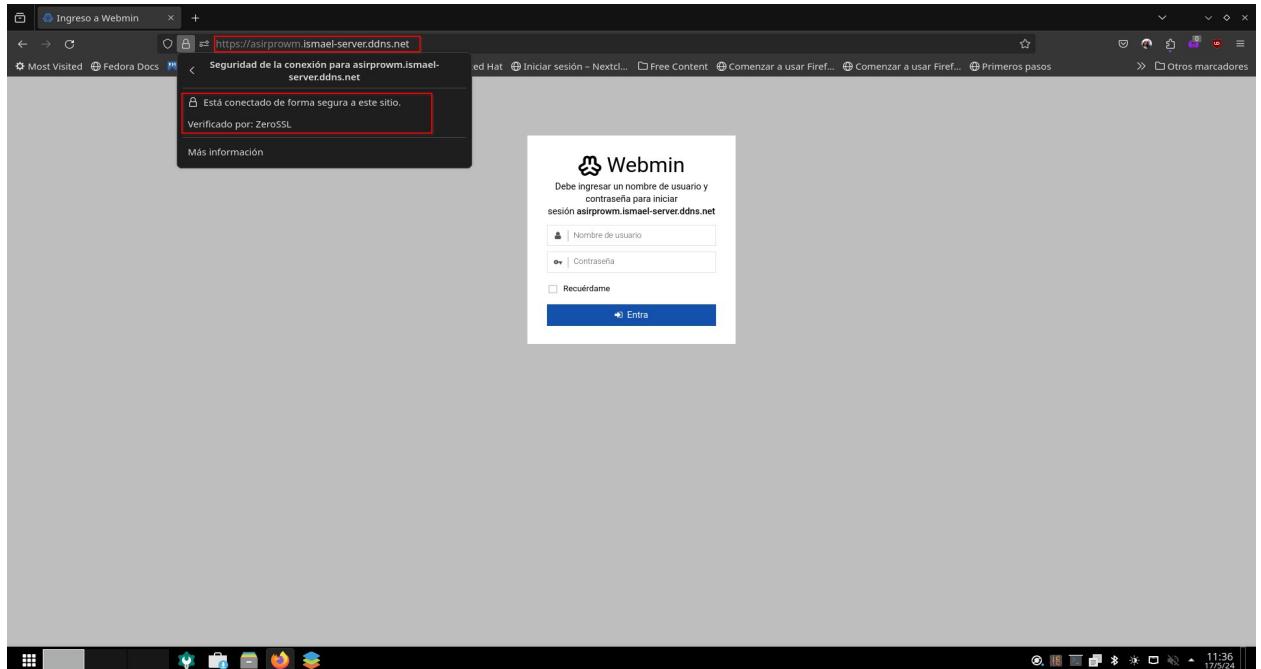
Grafana: <https://asirprogrf.ismael-server.ddns.net>



Cockpit console: <https://asirproc.ismael-server.ddns.net/>



## [Webmin Console: https://asirprowm.ismael-server.ddns.net/](https://asirprowm.ismael-server.ddns.net/)



Finalmente, este es el aspecto de los puertos abiertos al exterior de nuestro servidor, quedando abiertos únicamente los estrictamente necesarios, lo que nos deja un servidor con la mínima superficie de ataque posible, en cuanto a conexiones se refiere.

Llegados a este punto, tenemos una completa pila de software desplegada y lista para usar, completamente accesible desde Internet en cualquier lugar, y con un nivel de seguridad muy aceptable, gracias a la baja exposición de puertos, y la prohibición de login por ssh si no se dispone de una clave criptográfica autorizada.

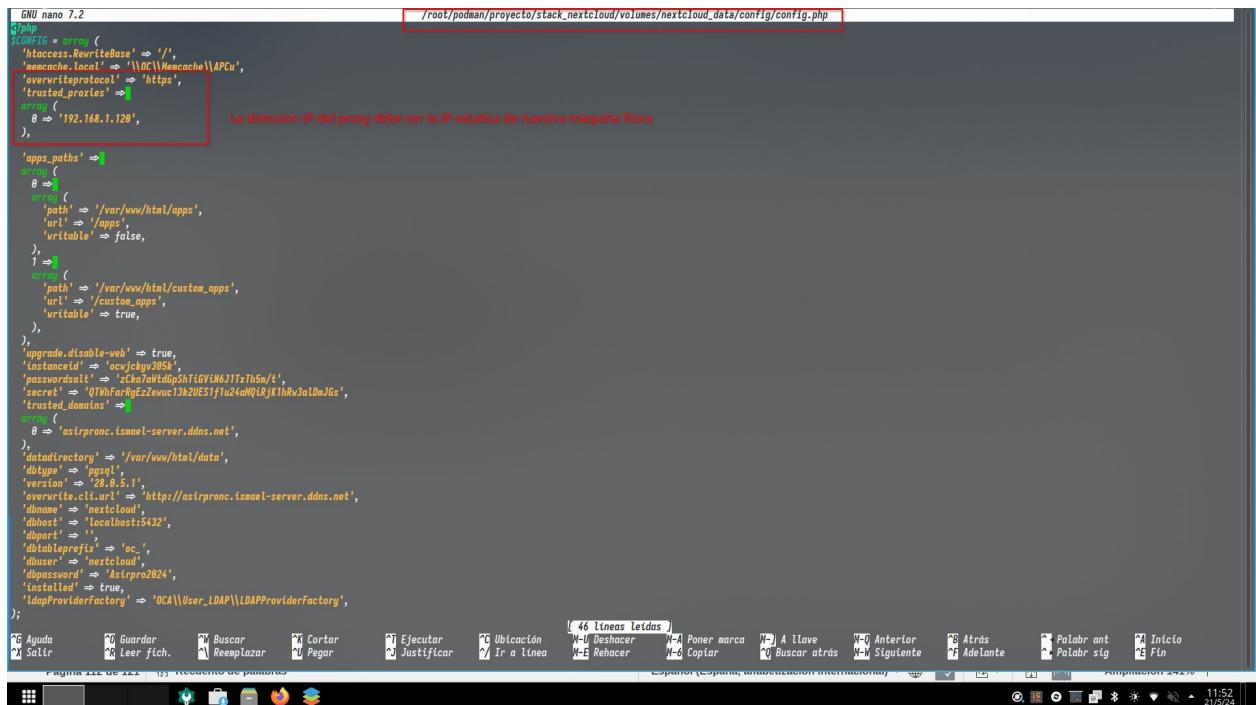
Podemos pues dar por finalizado el despliegue de este proyecto, aunque aún hay ciertos objetivos que exploraremos en los próximos puntos.

119

Administración de sistemas informáticos en red

## 9.4.4 Caddy Server. Anexo I: Configuración adicional de nextcloud para el proxy Inverso

Una vez desplegado el proxy inverso, la pila esta lista para funcionar. Sin embargo, para que todo funcione a la perfección debemos hacer un pequeño ajuste adicional que he considerado importante añadir en este anexo.



```
GNU nano 7.2
$php
$CONFIG = array (
    'htaccess.rewriteBase' => '/',
    'memcache.local' => '\\DC\\Memcache\\APCu',
    'overwriteprotocol' => 'https',
    'trusted_proxies' => [
        0 => '192.168.1.128',
    ],
    'apps_paths' => [
        0 => [
            'array' => [
                'path' => '/var/www/html/apps',
                'url' => '/apps',
                'writable' => false,
            ],
            '0' => [
                'path' => '/var/www/html/custom_apps',
                'url' => '/custom_apps',
                'writable' => true,
            ],
        ],
        'upgrade.disable-web' => true,
        'instanceid' => 'ocvjcbgv305k',
        'passwordsalt' => 'zckazVatd0pShfGVlW6J1TxTh5n/t',
        'secret' => 'QTMhFmrgEzZewuc13b20ESfIu24aMqIRjkK1hRw3aL0mJ6s',
        'trusted_domains' => [
            'array' => [
                0 => 'asirprnc.ismael-server.ddns.net',
            ],
            'datadirectory' => '/var/www/html/data',
            'dbtype' => 'pgsql',
            'version' => '2.8.0.5.1',
            'overrite.cli.url' => 'http://asirprnc.ismael-server.ddns.net',
            'domain' => 'nextcloud',
            'dbhost' => 'localhost:5432',
            'dbport' => '',
            'dbtableprefix' => 'oc_',
            'dbuser' => 'nextcloud',
            'dbpassword' => 'Asirpro2024',
            'installed' => true,
            'ldapProviderfactory' => 'OCA\\User_LDAP\\LDAPProviderFactory',
        ],
    ],
    'Ayuda'      : '<F1> Ayuda',
    'Guardar'   : '<F2> Guardar',
    'Buscar'    : '<F3> Buscar',
    'Cortar'     : '<F4> Cortar',
    'Ejecutar'  : '<F5> Ejecutar',
    'Ubicación' : '<F6> Ubicación',
    'Justificar': '<F7> Justificar',
    'Ir a línea': '<F8> Ir a línea',
    'Deshacer'  : '<F9> Deshacer',
    'Poner marca': '<F10> Poner marca',
    'Añadir llave': '<F11> Añadir llave',
    'Anterior'  : '<F12> Anterior',
    'Siguiente' : '<Shift> Siguiente',
    'Atrás'     : '<Alt> Atrás',
    'Adelante'  : '<Shift> Adelante',
    'Palabra anterior': '<Shift> Palabra ant',
    'Palabra siguiente': '<Shift> Palabra sig',
    'Inicio'    : '<Shift> Inicio',
    'Fin'       : '<Shift> Fin'
);

```

Tan solo debemos editar el archivo **config/config.php** en el volumen de Nextcloud, añadiendo las etiquetas '**overwriteprotocol**' => '**https**' y un array PHP con la lista numerada de proxys (En nuestro caso solo uno) en formato **0 => 'DireccionIP del host'**.

Tras añadir dichas lineas, guardamos los cambios y reiniciamos el contenedor, y el funcionamiento de Nextcloud con nuestro proxy inverso será perfecto.

## 10. Script de automatización de tareas. El asistente de despliegue en bash

Durante todo este proyecto hemos desplegado toda la pila de software mediante comandos de terminal usando Podman, usando en la primera parte comandos de despliegue individuales de pods y contenedores, y usando la herramienta **podman kube** y manifiestos Yaml en el resto.

Si bien considero que es importante exponer dichas vías de despliegue de este proyecto, el uso de esos medios presenta tres problemas principales:

- Es un proceso relativamente lento y complejo.
- Resulta poco flexible, pues los manifiestos Yaml están diseñados para un dominio en concreto.
- Las contraseñas están escritas en un texto plano en los manifiestos Yaml, lo que no es precisamente una buena idea.

De los tres problemas expuestos, la flexibilidad es el que tiene más fácil solución, pues a fin de cuentas los manifiestos Yaml son archivos de texto plano, y su sintaxis es lo suficientemente sencilla para que cualquier persona pueda editarlos amoldándolos a sus necesidades. Sin embargo, siguen debiendo ejecutarse comandos individuales por cada pod más el contenedor de Caddy, y las contraseñas seguirían siendo almacenadas en los Yaml.

Como solución a estas problemáticas se ha desarrollado un útil script en bash, que automatiza algunas de las tareas más frecuentes a utilizar con la pila de software, siendo la más útil de ellas la opción “Despliegue inicial”.

Este capítulo no pretende ser una inspección al código de dicho script, que será entregado junto con toda la documentación y resto de archivos del proyecto, sino un breve repaso a su funcionalidad y modo de uso.

Para esta guía se seguirá el siguiente orden de procedimientos:

1. Formateo de volúmenes de la instancia, de la que previamente se hará una copia de seguridad
2. Despliegue inicial de esa instancia no configurada
3. Importación de los datos del servicio de directorio
4. Restauración de la copia de seguridad y redespliegue de la instancia previamente respaldada

## 10.1 Script de automatización de tareas. Despliegue inicial



Comenzamos creando una copia de toda la instancia, copiando el directorio que contiene todos los volúmenes, manifiestos y resto de archivos necesarios.

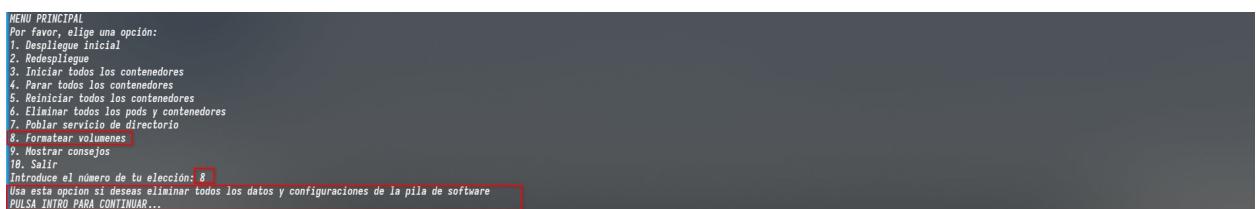


Situados en el directorio del proyecto, ejecutamos el script **asistente.sh**



AUTOR: Ismael Carrasco Cubero  
Antes de comenzar, asegurate de que este directorio y todo su contenido estan en la ruta /root/podman y de haber creado en podman una red llamada "proyecto"  
para poder ejecutar el comando "podman network create proyecto"  
Por razones de compatibilidad de permisos con los volúmenes, este asistente debe ejecutarse como root  
PULSA INTRO PARA CONTINUAR

La pantalla de bienvenida nos explica algunos requisitos previos y nos insta a pulsar intro para continuar.



MENU PRINCIPAL  
Por favor, elige una opción:  
1. Despliegue inicial  
2. Redespliegue  
3. Iniciar todos los contenedores  
4. Parar todos los contenedores  
5. Reiniciar todos los contenedores  
6. Eliminar todos los pods y contenedores  
7. Poblar servicio de directorio  
8. Formatear volúmenes  
9. Mostrar consejos  
10. Salir  
Introduce el número de tu elección: 8  
Usa esta opción si deseas eliminar todos los datos y configuraciones de la pila de software  
PULSA INTRO PARA CONTINUAR...

El script está estructurado con un menú principal con múltiples opciones que podemos escoger para realizar diversas tareas. Puesto que vamos a realizar un despliegue inicial, comenzaremos “formateando” los volúmenes para dejar la instancia completamente libre de datos previos seleccionando la opción deseada.

El script nos da una breve explicación de que es lo que se va a realizar y cuál es su utilidad.



MENU PRINCIPAL  
Por favor, elige una opción:  
1. Despliegue inicial  
2. Redespliegue  
3. Iniciar todos los contenedores  
4. Parar todos los contenedores  
5. Reiniciar todos los contenedores  
6. Eliminar todos los pods y contenedores  
7. Poblar servicio de directorio  
8. Formatear volúmenes  
9. Mostrar consejos  
10. Salir  
Introduce el número de tu elección: 8  
Usa esta opción si deseas eliminar todos los datos y configuraciones de la pila de software  
PULSA INTRO PARA CONTINUAR...  
Formatando volúmenes...  
4579b3fcce9e34252a87dd8df5b26d9e727a42e183f6b90c2d20b6f811e839c  
0be822a3d34c1cdfe01c8bd21a8d8391a9ba64fd01e389f869a837b598d  
2542708be881a6a884e57a8919286ed3d6c28d57bb7db3f31af8b8dee34c62e  
proyecto\_caddyproxy  
Todos los datos de los volúmenes han sido eliminados!  
Usa la opción de parar y eliminar todos los contenedores y redespela la instancia desde el menu principal  
PULSA INTRO PARA CONTINUAR...

Tras unos instantes, el script detiene toda la pila y elimina los datos, y nos indica que a continuación debemos eliminar todos los contenedores y hacer un despliegue inicial.

122

Administración de sistemas informáticos en red

```

MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 6
Eliminando contenedores...
0be822e3da364c1cfdf2e018ba621a8d839ba19ba64fd0f0e389f86ba837b598d
4579b3fc9e3a252a870dddf5b25bd9d9727a742e113fb6b8c2d2b6b8f81e839c
2542708be801a6ba84e57d8918236ed3d6c2d57b5dbd3f1afabdeee34c62e2
0be822e3da364c1cfdf2e018ba621a8d839ba19ba64fd0f0e389f86ba837b598d
4579b3fc9e3a252a870dddf5b25bd9d9727a742e113fb6b8c2d2b6b8f81e839c
2542708be801a6ba84e57d8918236ed3d6c2d57b5dbd3f1afabdeee34c62e2
proyecto caddyprod
Todas las contenedores y pods han sido eliminados! PULSA INTRO PARA CONTINUAR...

```

De vuelta en el menú principal, usaremos la opción indicada anteriormente y eliminaremos todos los contenedores.

```

MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 1
Despliegue inicial!!!
Esta opción realiza una instalación nueva de toda la pila y la deja lista para funcionar.
Se te solicitarán los datos de tu instancia para realizar el despliegue
PULSA INTRO PARA CONTINUAR

```

Procedemos ahora a desplegar la instancia “de nuevas” seleccionando la opción 1. El script nos informa de que nos solicitará una serie de datos para realizar el despliegue.

```

MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 1
Despliegue inicial!!!
Esta opción realiza una instalación nueva de toda la pila y la deja lista para funcionar.
Se te solicitarán los datos de tu instancia para realizar el despliegue
PULSA INTRO PARA CONTINUAR

Aprobando el contenedor LDAP Account manager...
Estableciendo permisos en volúmenes...
A continuación se le solicitarán los datos de configuración del servicio de directorio
Introduzca el FQDN de su dominio en formato ejemplo.local: ismael-server.ddns.net
Introduzca la raíz de su dominio en formato dc=example,dc=local: dc=ismael-server,dc=ddns,dc=net
Introduzca la contraseña de usuario administrador de LDAP:
Introduzca la contraseña de usuario administrador de LDAP:
Confirme la contraseña de usuario administrador de LDAP:
Introduzca la contraseña de configuración de LDAP:
Confirme la contraseña de configuración de LDAP:
Introduzca la contraseña para el perfil de LAM
Confirme la contraseña para el perfil de LAM

```

El script irá realizando internamente las tareas necesarias y nos irá informando en todo momento de la acción que se está realizando. A continuación, comienza a solicitarnos los datos de configuración, comenzando con el FQDN de nuestro dominio, la raíz en formato **dc=example,dc=com** y las contraseñas necesarias de los distintos componentes. Es importante mencionar que el script tienen ciertas limitaciones: es capaz de ocultar los campos de introducción de contraseñas y pide que se confirmen para evitar errores, pero no comprueba patrones de texto, por lo que es importante introducir los datos correctamente. Hacer dichas comprobaciones habría supuesto un esfuerzo y tiempo que a estas alturas no está disponible.

```

Generando configuración del servicio de directorio...
Desplegando servicio de directorio
Pods:
c18cd568b354480dd5b25d945e3b94d9e5a94dc378ff9f9d8eefb6c2b82f21d4
Containers:
5846b77fc998f783f0753ba5c5fd8a6826ac1cb35b32cdafa73d047312f68291
2169bb7768ab241d4a68185c23931b7642695ebf4ea766c256eefea96e65

Activando memberOf Overlay
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=module[0],cn=config"

SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "oIdOverlay=memberof,oIdDatabase=[{}],cn=config"

Limpiando...
A continuación se le solicitarán los datos de configuración para Nextcloud y la instancia de base de datos
Introduzca el usuario administrador de PostgreSQL: 

```

El script despliega el pod de LDAP generando un manifiesto yaml temporal usando variables, activa el memberOf Overlay y a continuación elimina dicho manifiesto temporal.

```

A continuación se le solicitarán los datos de configuración para Nextcloud y la instancia de base de datos
Introduzca el usuario administrador de PostgreSQL: admin
Introduzca el email del usuario predeterminado para PGAdmin: admin@pg.es
Introduzca la contraseña :
Introduzca la contraseña para el usuario administrador de PostgreSQL:
Confirme la contraseña :

```

A continuación, el script comienza el despliegue del pod que contiene Nextcloud y PostgreSQL con PGAdmin. Nos solicita los datos de configuración de la base de datos y su panel de gestión. Al igual que antes, hemos de ser cuidadosos con la entrada de datos, por las limitaciones antes mencionadas.

```

Generando configuración para Nextcloud y la base de datos ...
Desplegando Nextcloud y la base de datos ...
Pod:
8fe6718a642139a8bd86d54acd0ce0d5bf0841a2ab8ffeedc0547d14fab91e69
Containers:
8e579a291c1c264aad1967e7b95e471149b855f87dc651423e6386c2a89dc4cb
8e19ae55b8bc6fbfa8c6981e81beacda6b95388198bc54bf918ff9a
1c358a867286f221a0b18bd1f8b9e7b574ec69b87616436f2abe9e97223c17bd3

Limpieando...
Desplegando Grafana...
Pod:
82a196ab16f498b4782c1115f1d28b2a29879edd9bd4020176a984f83188df00
Containers:
8e0b78e21fdcd55cc34328197158ef49718287483da824b9bc27fd9116a42
7cacbac8292f924a369a8857f2c165df2e5ad0a152db0ba7edc6c529fffb15
d4ff1caf1cfceec540bd798f617a88349c2c33a2dd2ecbb0db42a8fc46f

Desplegando proxy reverso Caddy Server...
dc1f6522072c77149b9ef346a74a6171e1980e14ac3fd762241e999321ce1b
Reiniciando proxy inverso...
proyecto_caddyproxy
Despliegue completado! El proxy inverso necesita algo de tiempo para obtener los certificados https. Si recibes errores al acceder a las aplicaciones espera unos minutos
PULSA INTRO PARA CONTINUAR...

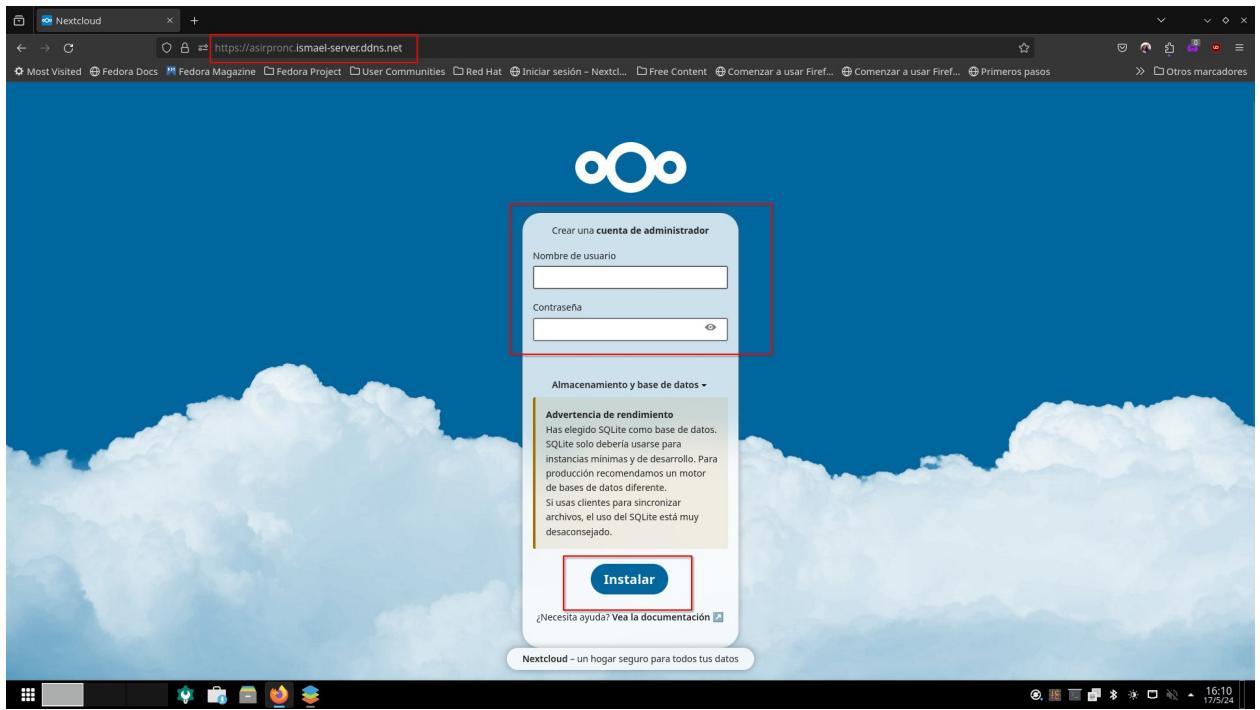
```

Introducidos los datos de configuración, el script genera el manifiesto temporal con las variables introducidas, despliega el pod y elimina el manifiesto temporal. Una vez desplegado el pod de Nextcloud y puesto que el pod de Grafana y el contenedor de Caddy no necesitan datos adicionales, el asistente continua con el despliegue automáticamente.

Una vez finalizado, se reinicia el proxy para que este detecte todos los hosts presentes en su Caddyfile, y el asistente nos informa que el despliegue ha finalizado y podemos comenzar a configurar nuestras aplicaciones. A partir de este momento la pila de software esta lista para comenzar a ser configurada.

1. Podman pod ls & 2. Podman container ls	POD ID	NAME	STATUS	CREATED	INFRA ID	# OF CONTAINERS					
	CONTAINER ID	IMAGE	CMD	CREATED	STATUS	PORTS	NAMES				
	82a196ab16f4	projeto.stackgrafana	Running	4 minutes ago	f2912b3883e4	4	c18cd5b60354-infra				
	c8fe6718a643	projeto.stacknc	Running	9 minutes ago	96f58d5f642b	4	projeto_stacknc-lua				
	c1c358a867286f	projeto.stackdc	Running	9 minutes ago	4585da5f59dc	3	projeto_stackdc-openldap				
	495f8d6f59dc	localhost/podman-pause:4.9.4-1711445992	Up 9 minutes	9 minutes ago			d8fed7718d43-infra				
	5846d577c98f	ghcr.io/library/postgreSQL:11-mariadb	/usr/local/bin/st...	9 minutes ago			projeto_stacknc-postgres				
	513690501542	ghcr.io/library/postgreSQL:11-mariadb-5.0	/usr/local/bin/st...	9 minutes ago			projeto_stacknc-pgadmin				
	94f58d6f642b	localhost/podman-pause:4.9.4-1711445992		9 minutes ago			82a196ab16f4-infra				
	84579a291c2c	docker.io/library/nextcloud:stable-apache	apache2-foreground...	4 minutes ago							
	8e19ae55b8bc	docker.io/library/postgres:latest	postgres	4 minutes ago							
	1c358a867286f	docker.io/dpdk/pgadmin4:latest		4 minutes ago							
	F2912b3883e4	localhost/podman-pause:4.9.4-1711445992		4 minutes ago							
	Beb078e21fd0	docker.io/prom/prometheus:latest	--web.listen-add...	4 minutes ago							
	7cacbac8292f	docker.io/grafana/grafana:latest		4 minutes ago							
	d4ff1caf1cf0	docker.io/node-exporter:latest	--path.rootfs=/ho...	4 minutes ago							
	dc1f6522072c	docker.io/library/caddy:latest	caddy run --confi...	4 minutes ago	Up 4 minutes	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp	projeto_caddyproxy				

Podemos comprobar que se ha realizado del despliegue completo, listando los pods y contenedores en ejecución.



Como vemos, los volúmenes ya no contienen ningún dato más allá de los generados durante el despliegue inicial, como prueba el asistente inicial de instalación de Nextcloud.

## 10.2 Script de automatización de tareas. Introducción de datos en servicio de directorio

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 7
Esta opción automatiza la importación de datos LDIF al servicio de directorio LDAP desplegado
El archivo LDIF adjunto solo es válido para el dominio ismael-server.ddns.net, por lo que si deseas importar tu propio ldif, renombralo a directorio.ldif
y añadelo al directorio /root/podman/proyecto/stack_ldap
PULSA INTRO PARA CONTINUAR...
```

Con la pila ya desplegada, pasamos a poblar el servicio de directorio. El script nos avisa de que el Ldif incorporado solo es válido para mi dominio, y explica como adjuntar el nuestro propio si disponemos otro dominio.

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 7
Esta opción automatiza la importación de datos LDIF al servicio de directorio LDAP desplegado
El archivo LDIF adjunto solo es válido para el dominio ismael-server.ddns.net, por lo que si deseas importar tu propio ldif, renombralo a directorio.ldif
y añadelo al directorio /root/podman/proyecto/stack_ldap
PULSA INTRO PARA CONTINUAR...
Introduzca el usuario administrador del directorio en formato cn (cn=admin,dc=example,dc=com) : cn=admin,dc=ismael-server,dc=ddns,dc=net
Introduzca la contraseña del administrador del directorio: ■
```

Nos solicita el usuario administrador de nuestro dominio y su contraseña.

```
adding new entry "cn=GrupoDesarrollo,ou=grupos,dc=ismael-server,dc=ddns,dc=net"
adding new entry "cn=RecursosHumanos,ou=grupos,dc=ismael-server,dc=ddns,dc=net"
adding new entry "cn=SoporteTecnico,ou=grupos,dc=ismael-server,dc=ddns,dc=net"
adding new entry "cn=Nextcloudallowed,dc=ismael-server,dc=ddns,dc=net"
adding new entry "cn=users,dc=ismael-server,dc=ddns,dc=net"
adding new entry "ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=agarcia,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=agutierrez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=amarantinez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=drodriguez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=dcano,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=dperalta,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=eperez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=jcarrascocubero,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=ilopez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=jgomez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=jmartin,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=jrodriguez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=jsanchez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=mgonzalez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=mlopez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=mserrano,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=slopez,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
adding new entry "uid=sruiz,ou=usuarios,dc=ismael-server,dc=ddns,dc=net"
Entradas añadidas al directorio! PULSA INTRO PARA CONTINUAR...
```

Y en un instante nuestro directorio queda poblado con entradas.

# 10.3 Script de automatización de tareas. Otras opciones de utilidad

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 4
Usa esta opción si deseas detener los servicios de la pila
PULSA INTRO PARA CONTINUAR...

Parando contenedores y pods...
c8fe6718a43139abdd54cdccbd8fbf841a2ab8ffec8547d14fab91e69
c18cd5b68354486bd5b29b45e83b94ed9e594dc3789ff9cd8eef6b2082f21d4
82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
proyecto_caddyproxy
Todos los contenedores y pods han sido detenido! PULSA INTRO PARA CONTINUAR...

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4583da594dc localhost/podman-pause:4.9.4-1711445992 3 hours ago Exited (0) 58 seconds ago c18cd5b68354-infra
584b67c7c90f għċiex/0/ldapaccountmanager:lam:latest /usr/local/bin/st... 3 hours ago Exited (0) 58 seconds ago proyecto_stackldap-lam
2169bb7768ab docker.io/osixia/openldap:1.5.0 3 hours ago Exited (0) 58 seconds ago proyecto_stackldap-openldap
96f58d5f642b localhost/podman-pause:4.9.4-1711445992 2 hours ago Exited (0) 59 seconds ago c8fe6718a43-infra
84579a29a1c2 docker.io/library/nginxcloud:stable-apache apache2-foregroun... 2 hours ago Exited (0) 59 seconds ago proyecto_stacknc-nextcloud
8a19ae5bb8c0 docker.io/library/postgres:latest postgres 2 hours ago Exited (0) 59 seconds ago proyecto_stacknc-postgres
1c35a8a7286f docker.io/dpage/pgadmin4:latest 2 hours ago Exited (0) 59 seconds ago proyecto_stacknc-pgadmin
f2972b3883e4 localhost/podman-pause:4.9.4-1711445992 2 hours ago Exited (0) 59 seconds ago 82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
0e0b78e21fde docker.io/prom/prometheus:latest --web.listen-addr... 2 hours ago Exited (0) 59 seconds ago proyecto_stackgrafana-prometheus
7eacbac829f2 docker.io/grafana/grafana:latest 2 hours ago Exited (0) 59 seconds ago proyecto_stackgrafana-grafana
d4ff1cacfbfc docker.io/prom/node-exporter:latest --path.rootfs=/ho... 2 hours ago Exited (2) 57 seconds ago proyecto_stackgrafana-node_exporter
dc1f6522072c docker.io/library/caddy:latest caddy run --confi... 2 hours ago Exited (0) 57 seconds ago 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp proyecto_caddyproxy

PULSA INTRO PARA CONTINUAR...

```

Otras opciones de utilidad en el asistente son las funciones para parar toda la pila...

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 3
Usa esta opción para iniciar los contenedores si los has detenido
PULSA INTRO PARA CONTINUAR...

Iniciando contenedores y pods...
c8fe6718a43139abdd54cdccbd8fbf841a2ab8ffec8547d14fab91e69
c18cd5b68354486bd5b29b45e83b94ed9e594dc3789ff9cd8eef6b2082f21d4
82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
proyecto_caddyproxy
Todos los contenedores y pods han sido iniciados! PULSA INTRO PARA CONTINUAR...

> c1
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4583da594dc localhost/podman-pause:4.9.4-1711445992 3 hours ago Up 28 seconds c18cd5b68354-infra
584b67c7c90f għċiex/0/ldapaccountmanager:lam:latest /usr/local/bin/st... 3 hours ago Up 28 seconds proyecto_stackldap-lam
2169bb7768ab docker.io/osixia/openldap:1.5.0 3 hours ago Up 28 seconds proyecto_stackldap-openldap
96f58d5f642b localhost/podman-pause:4.9.4-1711445992 2 hours ago Up 29 seconds c8fe6718a43-infra
84579a29a1c2 docker.io/library/nginxcloud:stable-apache apache2-foregroun... 2 hours ago Up 28 seconds proyecto_stacknc-nextcloud
8a19ae5bb8c0 docker.io/library/postgres:latest postgres 2 hours ago Up 28 seconds proyecto_stacknc-postgres
1c35a8a7286f docker.io/dpage/pgadmin4:latest 2 hours ago Up 28 seconds proyecto_stacknc-pgadmin
f2972b3883e4 localhost/podman-pause:4.9.4-1711445992 2 hours ago Up 28 seconds 82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
0e0b78e21fde docker.io/prom/prometheus:latest --web.listen-addr... 2 hours ago Up 28 seconds proyecto_stackgrafana-prometheus
7eacbac829f2 docker.io/grafana/grafana:latest 2 hours ago Up 28 seconds proyecto_stackgrafana-grafana
d4ff1cacfbfc docker.io/prom/node-exporter:latest --path.rootfs=/ho... 2 hours ago Up 27 seconds proyecto_stackgrafana-node_exporter
dc1f6522072c docker.io/library/caddy:latest caddy run --confi... 2 hours ago Up 27 seconds 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp proyecto_caddyproxy

PULSA INTRO PARA CONTINUAR...

```

Iniciarla de nuevo...

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir
Introduce el número de tu elección: 5
Usa esta opción si deseas reiniciar los contenedores, si por ejemplo alguno de ellos esta fallando
PULSA INTRO PARA CONTINUAR...

Reiniciando contenedores y pods...
c8fe6718a43139abdd54cdccbd8fbf841a2ab8ffec8547d14fab91e69
c18cd5b68354486bd5b29b45e83b94ed9e594dc3789ff9cd8eef6b2082f21d4
82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
proyecto_caddyproxy
Todos los contenedores y pods han sido reiniciados! PULSA INTRO PARA CONTINUAR...

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4583da594dc localhost/podman-pause:4.9.4-1711445992 3 hours ago Up 29 seconds c18cd5b68354-infra
584b67c7c90f għċiex/0/ldapaccountmanager:lam:latest /usr/local/bin/st... 3 hours ago Up 29 seconds proyecto_stackldap-lam
2169bb7768ab docker.io/osixia/openldap:1.5.0 3 hours ago Up 32 seconds proyecto_stackldap-openldap
96f58d5f642b localhost/podman-pause:4.9.4-1711445992 3 hours ago Up 38 seconds c8fe6718a43-infra
84579a29a1c2 docker.io/library/nginxcloud:stable-apache apache2-foregroun... 3 hours ago Up 32 seconds proyecto_stacknc-nextcloud
8a19ae5bb8c0 docker.io/library/postgres:latest postgres 3 hours ago Up 32 seconds proyecto_stacknc-postgres
1c35a8a7286f docker.io/dpage/pgadmin4:latest 3 hours ago Up 31 seconds proyecto_stacknc-pgadmin
f2972b3883e4 localhost/podman-pause:4.9.4-1711445992 3 hours ago Up 29 seconds 82a1b6ab16498b4782c1115fd28b2a29870ed9bd4820176a984f83108df99
0e0b78e21fde docker.io/prom/prometheus:latest --web.listen-addr... 3 hours ago Up 28 seconds proyecto_stackgrafana-prometheus
7eacbac829f2 docker.io/grafana/grafana:latest 3 hours ago Up 28 seconds proyecto_stackgrafana-grafana
d4ff1cacfbfc docker.io/prom/node-exporter:latest --path.rootfs=/ho... 3 hours ago Up 28 seconds proyecto_stackgrafana-node_exporter
dc1f6522072c docker.io/library/caddy:latest caddy run --confi... 3 hours ago Up 27 seconds 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp proyecto_caddyproxy

PULSA INTRO PARA CONTINUAR...

```

O reiniciarla.

Administración de sistemas informáticos en red

127

## 10.4 Script de automatización de tareas. Restauración de una instancia previa

Pasemos ahora a simular una restauración de instancia previamente almacenada en un backup, usando la función de redespliegue del asistente.

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir

Introduce el número de tu elección: 6
Eliminando contenedores...
cfe6718a643139a8bd8d54acdcebd5f8fd841a2ab8ffaec8547d14fab91e69
c18cd568b3548bb6d5b2b945e83b394ed9e594dc3789ff9cd80efb6c2882f21d4
82a106ab164f98b47821115fd128b2a29878edd9bd4820176a984f83188df00
proyecto_caddyprox
cfe6718a643139a8bd8d54acdcebd5f8fd841a2ab8ffaec8547d14fab91e69
c18cd568b3548bb6d5b2b945e83b394ed9e594dc3789ff9cd80efb6c2882f21d4
82a106ab164f98b47821115fd128b2a29878edd9bd4820176a984f83188df00
proyecto_caddyprox
Todos los contenedores y pods han sido eliminados! PULSA INTRO PARA CONTINUAR...
[el & el]
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
POD ID NAME STATUS CREATED INFRA ID # OF CONTAINERS
[✓] ~ ~/podman/proyecto
```

El primer paso es eliminar todos los contenedores presentes en uso, para lo que podemos utilizar la función incluida en el asistente para tal fin.

```
> cd ..
> rm -rf proyecto && mv backup_proyecto proyecto
> ls
config.bak  heatmap  jellyfin  joomla  mongodb  proxy  proyecto  stack_nextcloud  stack_postgres
[✓] ~ ~/podman
```

Eliminamos el directorio actual, y restauramos la copia de seguridad que hicimos en pasos anteriores.

```
MENU PRINCIPAL
Por favor, elige una opción:
1. Despliegue inicial
2. Redespliegue
3. Iniciar todos los contenedores
4. Parar todos los contenedores
5. Reiniciar todos los contenedores
6. Eliminar todos los pods y contenedores
7. Poblar servicio de directorio
8. Formatear volúmenes
9. Mostrar consejos
10. Salir

Introduce el número de tu elección: 2
Esta opción realiza el despliegue de una instancia previamente configurada que ya contiene configuraciones almacenadas en los volúmenes
Resulta útil si has eliminado los contenedores, y deseas redesplicarlos conservando todos sus datos para ejemplo actualizar las aplicaciones a sus nuevas imágenes
PULSA INTRO PARA CONTINUAR...

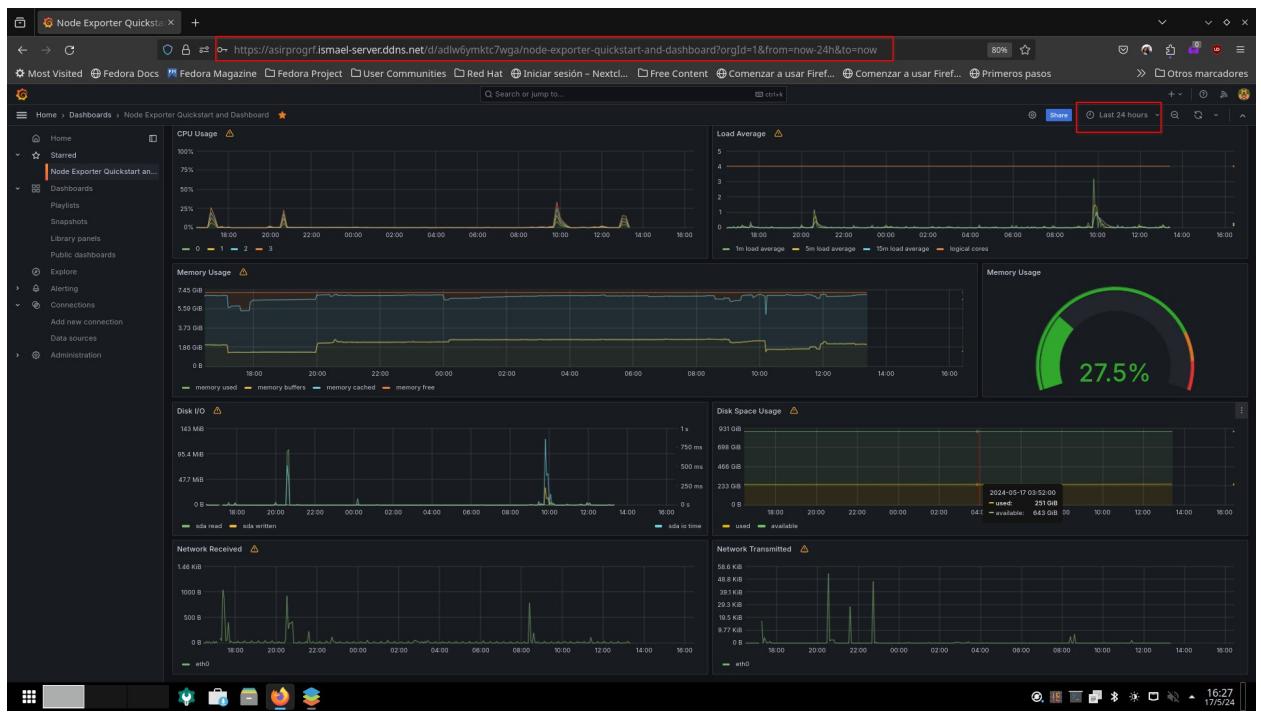
Estableciendo permisos en volúmenes...
Ejecutando comandos de redespiego.
Desplegando el servicio de directorio
Pod:
03c295b912fc2c6c8893d9237e4c80bdb812887fdaddee1c695f78688ea9e6ef
Containers:
7539852ea671ade79b9b3396490e2049c81a8db643883abf3864da3ed0fccb14d
4e284912b7f3abcf854ddaa2f9cd02978c7586c219bcd7e9d4d8335bc96523b

Despliegando Nextcloud y base de datos
Pod:
bb443d49feef1ebc5732763aa7ff3af83da7ff09361c2cbaed77832ede775f2a8
Containers:
6814983496749ec51fb229bda1b644383fa25976fc757d8fa#f438a1327ff
6b66e4e075a47c2ace771e7ec0a0a05d3bcbe2c8e668334ab06e820dd12ee847
01f596749da31a2eb5c7f9b68f9432dfdc38bfbe3efccb43fe6cc0fc3c188e0

Despliegando Grafana y Prometheus
Pod:
ac2e489db07a7f35548c591bae7f3f2fb7b855f3285b9439728ff7db7a80fe24f
Containers:
7983d1d2c97459eb1e1b1f83de47b756bb9503837ah83e78a943a41205c1be
0a9782311c1459a98c1c66afad932735773fe0c5fd8c029989788e2ed56
4e62269216597bae801d4349342c8658547338441c81c6a843a588e8875f0f

Despliegando proxy inverso Caddy
48a957867abab43a825c13a977182a6dad9b9c393f9ae5c3886440664a24bda
Despliegue completado! PULSA INTRO PARA CONTINUAR...
```

Y tras restaurar el directorio, usamos la opción de redespliegue del asistente, el cual desplegará la pila al completo, usando manifiestos de despliegue especiales para el mismo, que no contienen contraseñas, nombres de usuario y otros parámetros. Los nuevos contenedores utilizarán los datos almacenados en los volúmenes para retomar su actividad exactamente donde lo dejaron.



Esto es fácilmente comprobable si entramos a Grafana, y observamos que este contiene datos de las ultimas 24h, que han sido restaurados con la instancia.

# 11. Copias de seguridad y resiliencia

Todo sistema que ofrezca un servicio, y más aún uno destinado principalmente a almacenar datos, debe estar preparado para un eventual desastre con el almacenamiento de los mismos.

Se hace necesario pues establecer algún mecanismo para garantizar la supervivencia de los datos almacenados de toda la pila mediante copias de seguridad.

En los siguientes apartados se establecerá dicho mecanismo, y se simulará de forma ilustrativa la enorme facilidad de redespliegue de toda la pila ante un eventual desastre si disponemos de copias de seguridad.

## **11.1 Copias de seguridad y resiliencia. El comando rsync**

No disponemos de infraestructura dedicada exclusivamente a copias de seguridad, como los NAS, así que... ¿Como establecemos algún método de copias de seguridad efectivo y automatizado? El comando **rsync** es la respuesta a estas necesidades.

Se trata de un potente programa que como su nombre indica, “sincroniza” archivos y directorios, en el más estricto sentido de la palabra.

A diferencia de un comando de copia tradicional como **cp.**, rsync no se limita a hacer una copia exacta de una ubicación a otra (si bien puede funcionar como una copia tradicional si lo deseamos), sino que puede ser ejecutado para realizar copias incremétales, copias solo de archivos modificados, copia a través de dispositivos de red y otras muchas características. Una ventaja añadida a rsync, es que sabe en todo momento en que punto de la transferencia o copia se encuentra, por lo que si durante el proceso de copia, este se detiene por algún motivo, rsync puede retomar dicha copia exactamente por donde se quedó en cuanto las circunstancias lo permiten.

En nuestro caso usaremos rsync para sincronizar la pila completa con todos sus volúmenes en un HDD mecánico SATA III de 2TB destinado a las copias de seguridad. Dicho HDD se conectará a nuestro servidor mediante un Dock SATA III – USB 3.2 externo.



***El dock SATA III – USB 3.2 junto al HDD de 2TB y un SSD de 256 GB***

## 11.1.1 Comando rsync. Montaje de la unidad externa.

El primer paso para nuestras copias de seguridad es montar la unidad externa en la que serán almacenadas. Pasamos a describir el proceso usando nuestra consola de administración Cockpit.

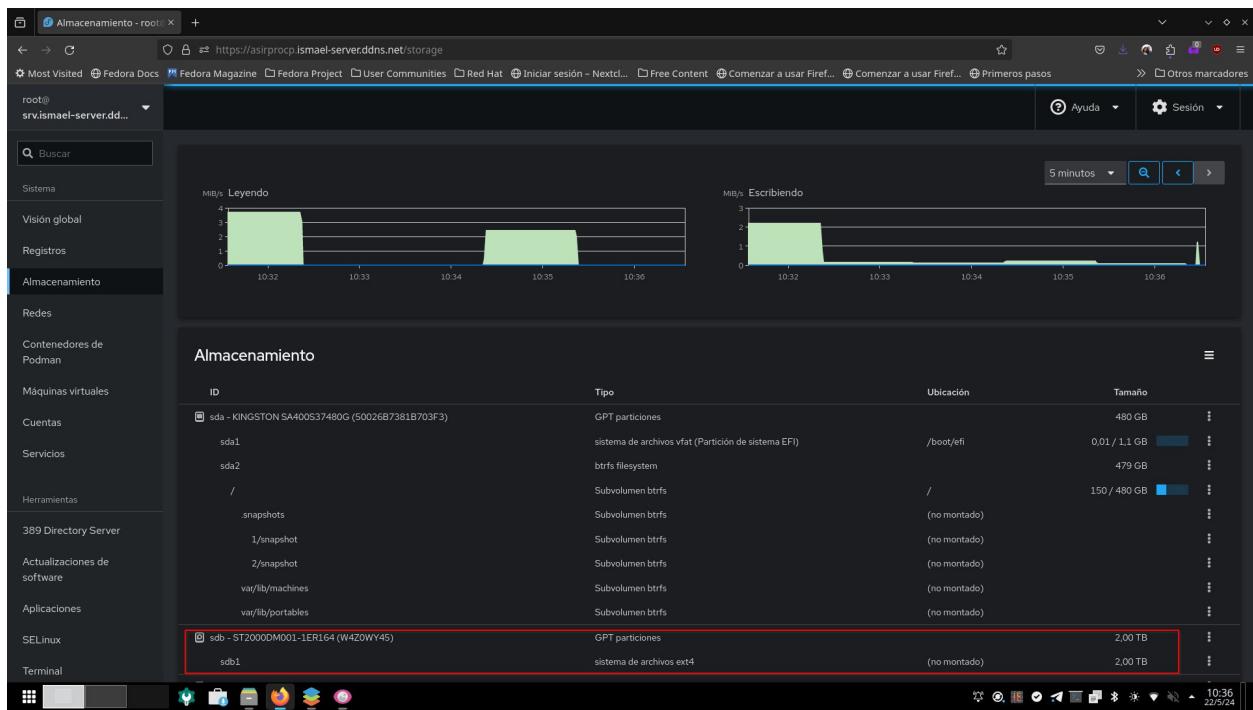
```
root@srv.ismael-server.ddns.net ~
OS: Fedora Linux 39 (Server Edition) x86_64
Host: HP EliteDesk 800 G2 DM 35W
Kernel: 5.18.200-532.8.64
Uptime: 18 days, 10 hours, 21 mins
Terminal: /dev/pts/0
CPU: Intel i5-6500T (4) @ 3.100GHz
GPU: Intel HD Graphics 530
Memory: 1378MB / 7342MB
GPU Driver: Hewlett-Packard Company Device [103c:8055]
CPU Usage: 85%
Local IP: 192.168.1.109
Public IP: 89.35.140.212
Users: root

Filesystem Size Used Avail Use% Mounted on
/dev/sda2 446G 137G 307G 30.8% /root
[root@srv.ismael-server.ddns.net ~]# mount -t ext4 -o defaults /dev/sda2 /root
[OK] 2024-05-22T10:26:06
```

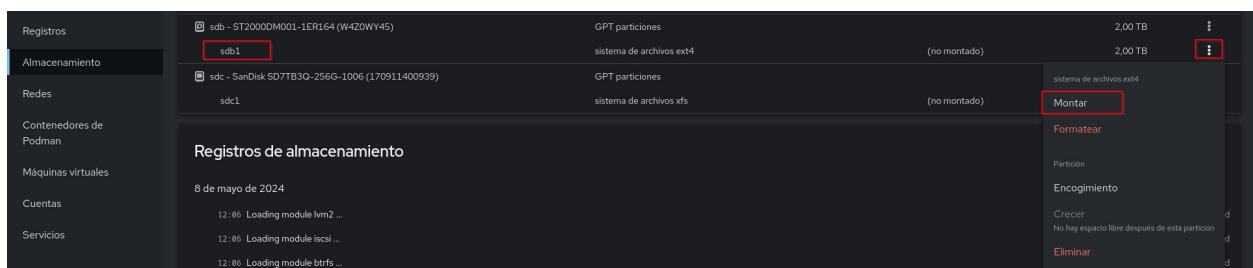
Antes de proceder al montaje de la unidad, crearemos un directorio dedicado exclusivamente a su punto de montaje, en un directorio padre dedicado a puntos de montaje.

The screenshot shows the Cockpit web interface for a Fedora Linux 39 Server Edition. The left sidebar has a 'Almacenamiento' (Storage) item highlighted. The main content area is divided into four sections: 'Salud' (Health), 'Uso' (Usage), 'Información del sistema' (System Information), and 'Configuración' (Configuration). The 'Uso' section shows CPU and memory usage. The 'Información del sistema' section provides details like model (HP EliteDesk 800 G2 DM 35W), serial number (CZC720PC9), and uptime (18 days, 10 hours, 21 mins). The 'Configuración' section includes options for changing the storage name to 'srv.ismael-server.ddns.net editor', setting the system time to '22 may 2024, 10:34', and selecting a performance profile ('throughput-performance'). The bottom status bar shows network and battery information.

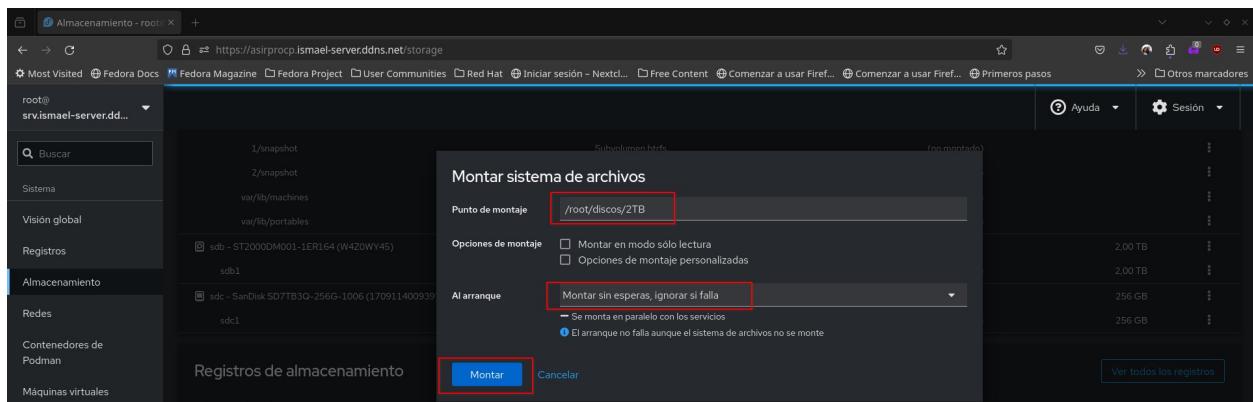
Nos loquearemos en nuestra consola Cockpit y nos dirigimos a el apartado de “Almacenamiento” en la columna izquierda.



En este apartado podemos obtener información sobre las unidades conectadas al equipo y métricas sobre las mismas. La unidad que nos interesa es /dev/sdb.



Pulsaremos sobre el botón de opciones adicionales (3 puntos) y escogeremos la opción “Montar” en la partición sdb1.



Definimos nuestro punto de montaje, las opciones de montaje durante el arranque deseada y pulsamos sobre montar.

sdb - ST2000DM001-1ER164 (W4ZOWY45)	GPT particiones	2,00 TB	⋮
sdb1	sistema de archivos ext4	/root/discos/2TB	0,67 / 2,0 TB
sdc - SanDisk SD7TB3Q-256G-1006 (1709111400939)	GPT particiones	256 GB	⋮
sdc1	sistema de archivos xfs	(no montado)	256 GB
> <code>lsblk &amp;&amp; cat /etc/fstab</code>			
<code>NAME MAJ:MIN RM SIZE TYPE MOUNTPOINTS</code>			
<code>sda 8:0 0 447,1G 0 disk</code>			
<code>└─sda1 8:1 0 1G 0 part /boot/efi</code>			
<code>└─sda2 8:2 0 446,1G 0 part /var/lib/containers/storage/overlay</code>			
<code>/</code>			
<code>sdb 8:16 0 1,0T 0 disk</code>			
<code>└─sdb1 8:17 0 1,0T 0 part /root/discos/2TB</code>			
<code>sdc 8:32 0 238,5G 0 disk</code>			
<code>└─sdc1 8:33 0 238,5G 0 part</code>			
<code>zram0 252:0 0 7,2G 0 disk [SWAP]</code>			
<code>#</code>			
<code># /etc/fstab</code>			
<code># Created by anaconda on Sun Feb 25 20:17:58 2024</code>			
<code>#</code>			
<code># Accessible filesystems, by reference, are maintained under '/dev/disk/'.</code>			
<code># See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.</code>			
<code>#</code>			
<code># After editing this file, run 'systemctl daemon-reload' to update systemd</code>			
<code># units generated from this file.</code>			
<code>#</code>			
<code>UUID=c8435ce2-9fd8-44c3-813d-07c1c5ac7567 /          btrfs defaults    0 0</code>			
<code>UUID=0738-5259   /boot/efi          vfat  umask=0077,shortname=winnt 0 0</code>			
<code>UUID=f9df3e39-5585-43b9-b3bd-3a05947a63bd /root/discos/2TB auto nofail 0 0</code>			

Tras unos breves instantes la unidad queda montada y lista para ser usada. Un vistazo rápido del comando lsblk y el contenido de /etc/fstab dan fe de que la unidad esta correctamente configurada.

## 11.1.1 Comando rsync. Ejecución y automatización

Ya tenemos nuestra unidad de backups lista para almacenarlas. A continuación, se muestra el comando rsync utilizado, su ejecución y su posterior automatización.

cd media/datos/2TB/backup\_proyecto

Pero antes, crearemos un directorio en la unidad, dedicado a las copias de seguridad.

```
GNU nano 7.2                                backups.sh
#!/bin/bash

# Directorio de origen
SRC="/root/podman/proyecto"                  Ruta a copiar

# Directorio de destino de las copias
DEST="/root/datos/2TB/backup_proyecto"        Ruta de destino.

# Obtener la fecha y la hora de la copia
FECHA=$(date +%Y%m%d_%H%M%S)                 Variable con la hora de ejecucion

# Patron para el nombre de la copia de seguridad
BACKUP_DIR="${DEST}/srv_backup_${FECHA}"       Patron para el nombre de la copia

# Ejecución de sync
rsync -a ${SRC} ${BACKUP_DIR}                   Ejecución de sync

# Mantener las copias de seguridad de los últimos 3 días y eliminar las más antiguas
find ${BACKUP_DIR} -maxdepth 1 -mtime +3 -type d -exec rm -rf {} \;      Borrado automático de los backups de más de 3 días
```

Creamos un sencillo script que automatizara la gestión de las copias de seguridad. Dicho script contiene variables que especifican los directorios de origen y destino, obtiene la fecha de realización de la copia, establece su nombre, copia el contenido con rsync y a continuación comprueba si existen copias más antiguas de 3 días; si estas existen procede a borrarlas.

```
ls -la ./discos/2TB/backup_proyecto
total 8
drwxr-xr-x. 2 root root 4096 may 22 10:49 .
drwxr-xr-x. 10 wizz wizz 4096 may 22 10:49 ..

```

Pasemos a probar si el script funciona correctamente. Como se ve en la captura, el directorio de backups esta vacío en estos momentos. A continuación, ejecutamos el script.

```
> $0 backups.sh  
> tar -zcvf ./discos/2TB/backup_proyecto  
./.../discos/2TB/backup_proyecto  
└─ srv_backup_20240522_113842  
    └─ proyectos  
  
3 directories, 0 files  
  
[ 0% ~ /podium/proyectos ] ✓ with rootpriv at 11:39:47
```

Tras unos instantes, la copia queda realizada en nuestra unidad de copias de seguridad. Ya solo debemos automatizar el proceso mediante cron.

The screenshot shows the Webmin interface on a Fedora Linux system. The left sidebar is a navigation tree with sections like 'Sistema' (selected), 'Actualizaciones de paquetes de software', 'Arranque y apagado', etc. The main content area displays 'Información del sistema' with four circular progress bars: CPU (0%), MEMORIA REAL (24%), MEMORIA VIRTUAL (16%), and ESPACIO EN EL DISCO LOCAL (37%). Below this, detailed system stats are shown, including the host name (srv.ismael-server.ddns.net), kernel version (Fedora Linux 39), and various system metrics. A section for 'Tareas Planificadas (Cron)' is highlighted in red, showing a list of recent logins from IP 10.89.1.188 and 127.0.0.1. The bottom status bar shows the date and time (22/5/24 11:51).

Para dicha automatización con Cron, nuestra consola Webmin, dispone de un completo módulo de gestión de tareas automatizadas. Lo seleccionaremos en la columna izquierda en el apartado de “Sistema”.

The screenshot shows the 'Tareas Planificadas (Cron)' module. The left sidebar has the 'Cron' option selected under 'Sistema'. The main panel lists scheduled cron jobs with columns for 'Usuario', 'Activa?', 'Comando', and 'Mover'. A red box highlights the 'Crear una nueva tarea de cron en catalogo' button. Other buttons include 'Crear una nueva variable de entorno', 'Controlar el acceso de usuarios a tareas de cron', 'Editar manualmente trabajos cron', 'Eliminar trabajos seleccionados', 'Deshabilitar trabajos seleccionados', 'Habilitar trabajos seleccionados', 'Detener a Cron Daemon', and 'Iniciar Cron Daemon Boot'. A note at the bottom explains the cron daemon's behavior during boot.

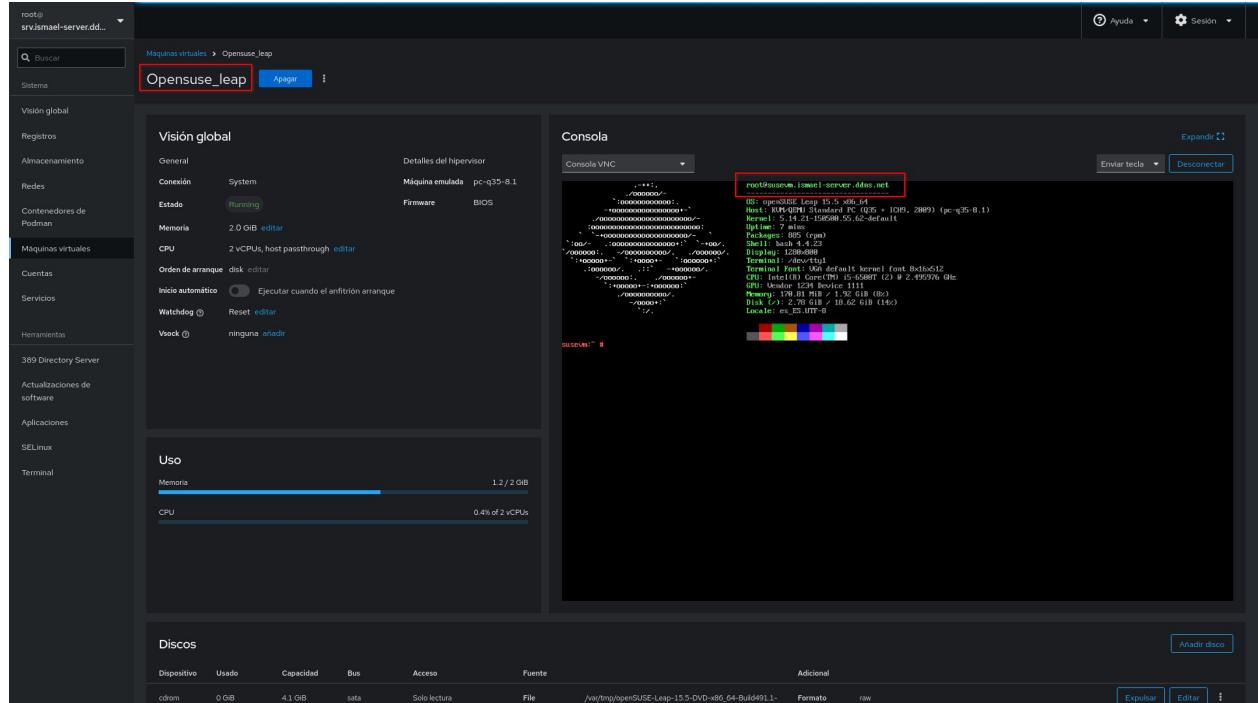
Una vez en el módulo de tareas planificadas, pulsaremos sobre “Crear una nueva tarea de cron en catalogo”.

Como vemos en la captura, el módulo de gestión de tareas de Cron en Webmin es increíblemente potente y completo. Debemos definir el usuario que ejecutara el comando, el comando propiamente dicho (la ejecución de nuestro script) y definimos que el script se ejecute en cualquier día, mes y día de la semana; quedando establecido que se ejecute a las 4:00 de la madrugada todos los días; pulsaremos en crear cuando hayamos definido la tarea programada.

De esta forma nuestras copias de seguridad se harán en un momento de baja actividad en el servidor, y las copias más antiguas se irán borrando automáticamente.

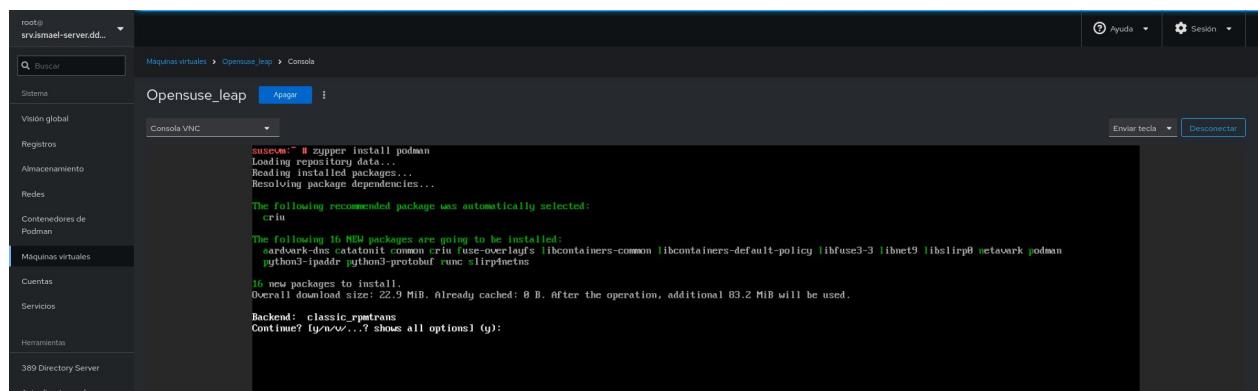
## 11.2 Copias de seguridad y resiliencia. Puesta a prueba en casos de desastres o migración

Para la prueba de resiliencia de nuestras copias de seguridad traspasaremos una de las copias de seguridad a otro sistema y redesplegaremos la instancia actual con una de las copias de seguridad que nuestro script realiza. Redesplegaremos la instancia usando el script de automatización descrito en puntos anteriores.

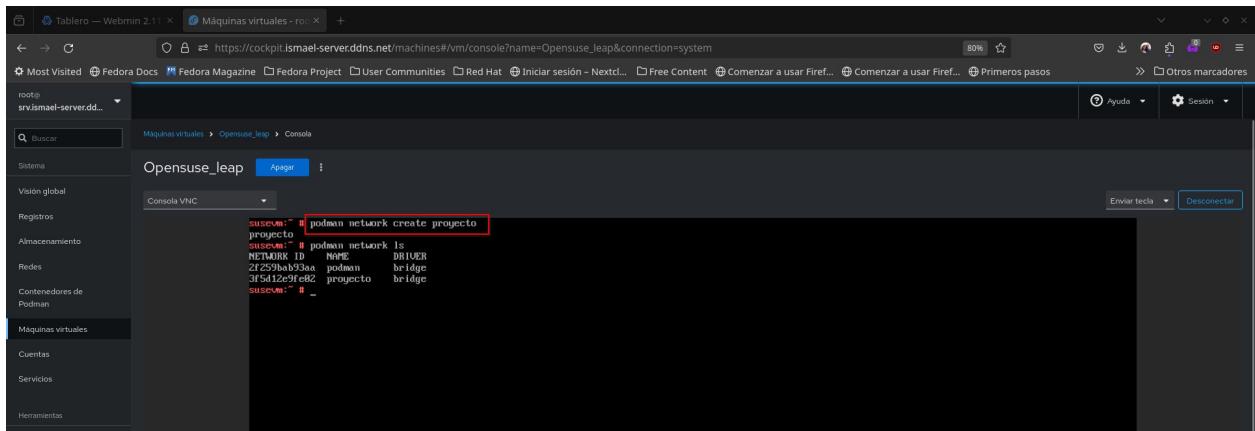


Para dicha prueba, hemos dispuesto una máquina virtual OpenSUSE Leap, gestionada dicho sea de paso en el servidor con la excelente herramienta de máquinas virtuales incorporada en Cockpit. Esta máquina simula ser un nuevo sistema desplegado debido a la perdida por desastre de nuestro anterior sistema Fedora 39.

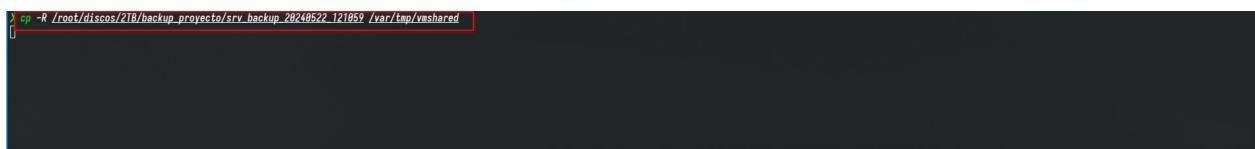
Esta simulación también nos sirve para ilustrar una migración completa de toda la pila a un nuevo sistema instalado voluntariamente desde cero.



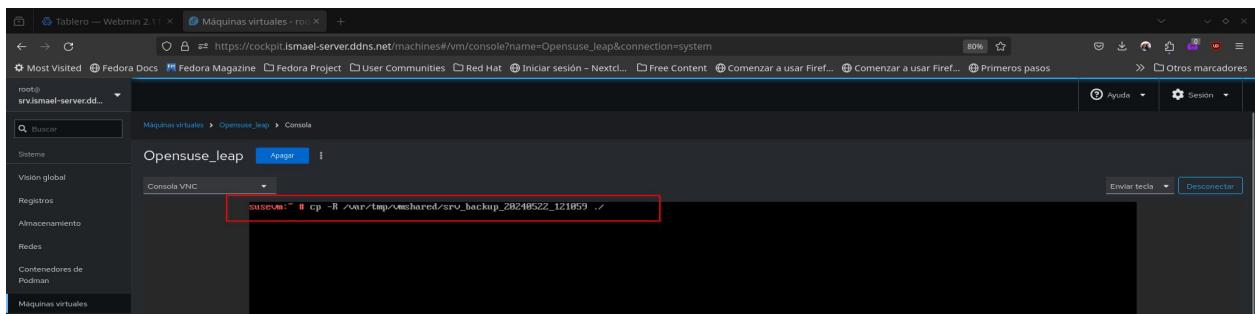
Una vez tuviéramos nuestra maquina instalada tan solo debemos instalar podman...



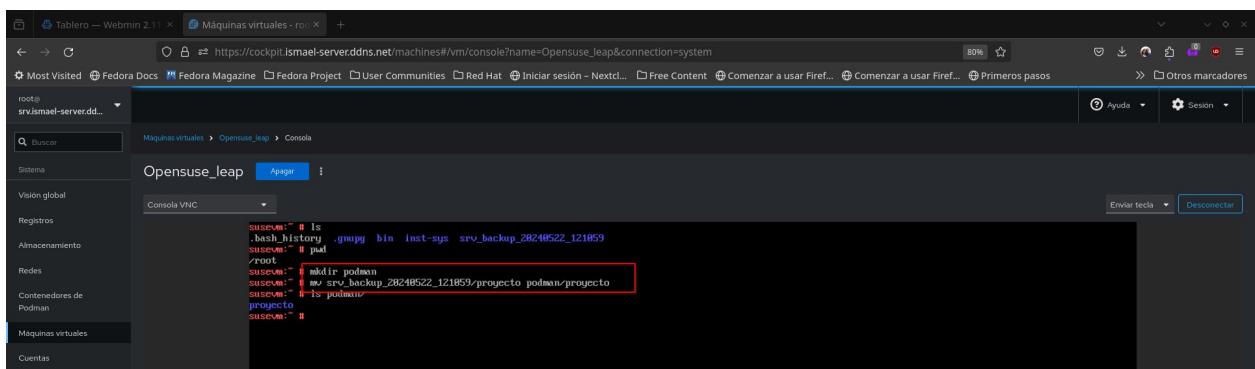
Creamos la red “proyecto” tal y como nos solicita nuestro script de automatización en su pantalla de bienvenida.



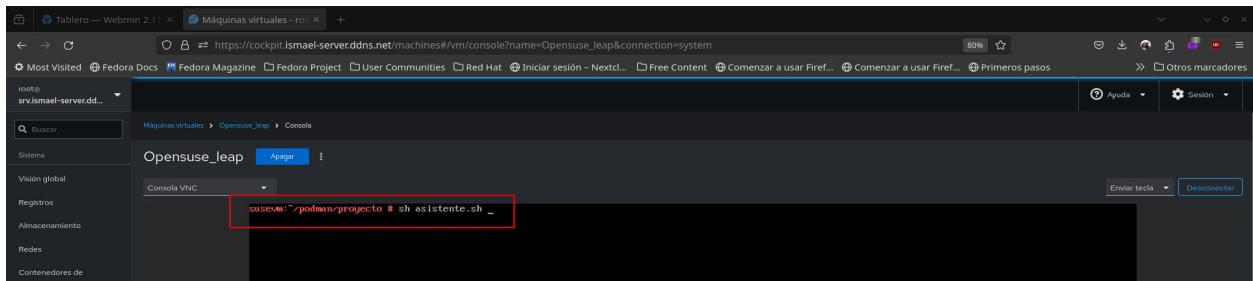
Debemos traspasar el directorio de nuestro backup a la nueva máquina. En este caso hemos compartido un directorio del sistema principal con la máquina virtual en /var/tmp, por lo que el primer paso es trasladarlo a dicha ubicación. Este paso dependerá de nuestras circunstancias ante el hipotético problema que enfrentemos, en cualquier caso, el paso se reduce a traspasar nuestra copia de seguridad del directorio de backups a la nueva máquina (O la maquina actual si solo pretendemos una restauración en la misma).



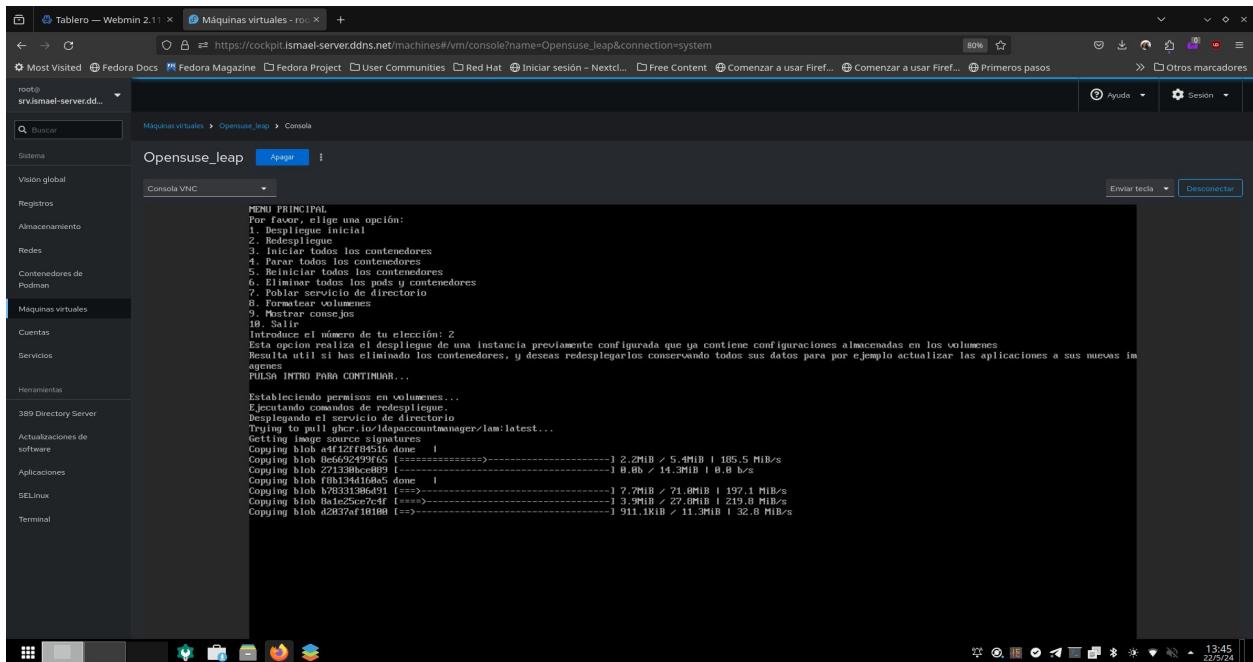
Ahora sí, trasladamos la copia de seguridad al nuevo host.



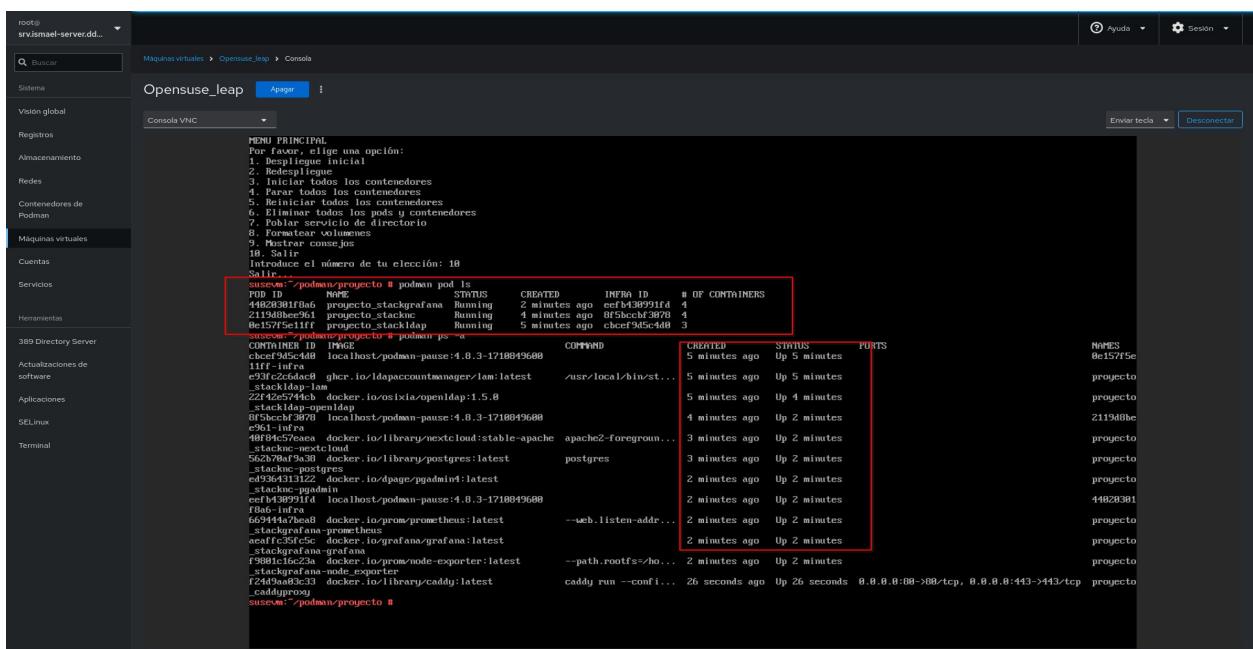
Y movemos el directorio “proyecto” de nuestra copia de seguridad al directorio podman tal y como se especifica en las instrucciones de nuestro script.



Entramos en el directorio y ejecutamos el asistente.



Escogemos nuestra opción de redespliegue y esperamos a que el script haga el resto por nosotros.



Redespliegue exitoso tal y como muestra nuestra lista de contenedores.

Administración de sistemas informáticos en red

140

Acciones	Nombre del usuario	Nombre	Apellido	Número UID	Número GID
<input type="checkbox"/>	agarcia	Alejandro	García	1014	2004
<input type="checkbox"/>	agutierrez	Alejandro	Gutiérrez	1006	2001
<input type="checkbox"/>	amartinez	Ana	Martínez	1020	2005
<input type="checkbox"/>	crodriguez	Carlos	Rodríguez	1008	2003
<input type="checkbox"/>	dcano	Daniel	Cano Verdú	1001	1001
<input type="checkbox"/>	dperalta	David	Peralta	1015	2005
<input type="checkbox"/>	eperez	Elena	Pérez	1022	2002
<input type="checkbox"/>	icarrascocubero	Ismael	Carrasco Cubero	10000	10000
<input type="checkbox"/>	ilopez	Ismel	López García	1002	1002
<input type="checkbox"/>	jgomez	Juan	Gómez	1019	2004
<input type="checkbox"/>	jmartin	José	Martín	1018	2003
<input type="checkbox"/>	jrodriguez	Javier	Rodríguez	1021	2001
<input type="checkbox"/>	jsanchez	Juan José	Sánchez Céspedes	1004	1004
<input type="checkbox"/>	mgonzalez	Maria	González	1007	2002
<input type="checkbox"/>	mlopez	Marta	López	1017	2002
<input type="checkbox"/>	mserrano	Manuel	Serrano Fernández	1005	1005
<input type="checkbox"/>	scuberomartinez	Sofia	Cubero Martínez	10001	10000
<input type="checkbox"/>	slopez	Sebastián	López Ojeda	1003	1003
<input type="checkbox"/>	sruiz	Sandra	Ruiz	1016	2001

Podemos comprobar entrando a nuestra aplicación LAM que nuestro directorio sigue ahí tal y como estaba. En un supuesto de migración a otra máquina, o reimplementación completa de la misma, es importante mencionar que la pila necesita de cierta configuración adicional, principalmente debido a los ajustes de red (Como en este caso, que se está accediendo al contenedor con un túnel ssh, al no estar disponible la IP privada a la que apuntan los nombres de dominio). No obstante, podemos poner en funcionamiento toda la pila en muy poco tiempo, con poca configuración adicional, quedándonos un sistema resiliente ante fallos.

## 12. Bibliografía

1. Sobre Nextcloud. (n.d.). <https://nextcloud.com/es/about/>
2. POUL-HENNING KAMP y ROBERT WATSON. ACM QUEUE. (2004). *Building Systems to be Shared Securely*. <https://dl.acm.org/doi/pdf/10.1145/1016998.1017001>
3. La “Contenerización” de aplicaciones. (n.d.).  
<https://www.plotandesign.com/sistemas/contenerizacion-de-aplicaciones/>
4. The new Debian Linux 7.0 is now available. (n.d.). <https://www.zdnet.com/article/the-new-debian-linux-7-0-is-now-available/>
5. Arch Linux, Arch wiki. (n.d.). [https://wiki.archlinux.org/title/Arch\\_Linux](https://wiki.archlinux.org/title/Arch_Linux)
6. Proyecto OpenSuse. (n.d.). <https://www.opensuse.org/>
7. RedHat Enterprise Server. (n.d.). <https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/server>
8. Flujo de desarrollo de Fedora Linux. (n.d.). <https://docs.fedoraproject.org/en-US/quick-docs/fedora-and-red-hat-enterprise-linux/>
9. Fedora Server. (n.d.). <https://fedoraproject.org/es/server/>
10. Proyecto Cockpit. (n.d.). <https://cockpit-project.org/>
11. Sobre Webmin. (n.d.). <https://webmin.com/about/>
12. Web oficial de Podman. (n.d.). <https://podman.io/>
13. Ionos - Artículo sobre diferencias entre Podman y Docker. (n.d.).  
<https://www.ionos.es/digitalguide/servidores/know-how/podman-vs-docker/>
14. Documentación oficial PostgreSQL. (n.d.). <https://www.postgresql.org/docs/current/intro-whatis.html>
15. Serializable Snapshot Isolation in PostgreSQL. (n.d.). <https://drkp.net/papers/ssi-vldb12.pdf>
16. Web Oficial Pgadmin - Características. (n.d.). <https://www.pgadmin.org/features/>
17. Web Oficial - LDAP Account Manager. (n.d.). <https://www.ldap-account-manager.org/lamcms/>
18. Web Oficial - Grafana. (n.d.). <https://grafana.com/grafana/>

19. ¿Que es Prometheus? (n.d.). <https://prometheus.io/docs/introduction/overview/>
20. Proyecto NodeExporter - GitHub. (n.d.). [https://github.com/prometheus/node\\_exporter](https://github.com/prometheus/node_exporter)
21. Web Oficial Caddy Server - Caracteristicas. (n.d.). <https://caddyserver.com/features>
22. Manual oficial rsync - samba.org. (n.d.). <https://download.samba.org/pub/rsync/rsync.1>