RaspiVault:
Almacenamiento en la nube

IGNASI RODRIGO CERVERA 2ºASIR
CURSO: 2023/2024
IES LA VEREDA
SANTI GIMENO MARTINEZ

ÍNDICE

1. Introducciones

1.1 Introducción

1.1.1 ¿Qué es el almacenamiento en la nube?

El almacenamiento en la nube es un método de almacenamiento y organización de datos que se realiza en la nube, una red de servidores remotos a los que se puede acceder mediante una conexión a Internet. Con el almacenamiento en la nube, los usuarios y las empresas pueden almacenar, acceder y mantener sus datos desde cualquier lugar con una conexión a Internet, en lugar de tener que guardar sus archivos en una única ubicación o dispositivo.

1.1.2 ¿Qué es RaspiVault?

RaspiVault es un conjunto de raspberry pi 4 que funcionan como servidores en alta disponibilidad (HA) ofreciendo un servicio de almacenamiento en la nube, en este caso utilizaré nextcloud como software, todos los datos que maneja nextcloud (incluido el propio software) se almacenan en un synology, esté synology se encarga de almacenar la base de datos y de crear copias de seguridad para resguardar la información de los clientes.

1.1.3 Beneficios de implementar RaspiVault

- Gran portabilidad.
- Los servidores ocupan poco espacio, apenas hacen ruido.
- Rápido y sencillo de instalar.
- No tienes que pagar cuotas por el uso de almacenamiento, solamente el precio del hardware.
- No tiene costes adicionales en licencias ni software.
- El software es totalmente compatible con otro hardware, es decir podríamos cambiar el hardware para ofrecer un mayor rendimiento al servicio y un mayor almacenamiento.
- Es económico, para un entorno reducido es la mejor opción.

1.1.4 Inconvenientes de RaspiVault

- Hardware limitado
- Mala escalabilidad, se debería de cambiar el hardware
- El NAS que utilizo no soporta mucho tráfico simultáneo
- Nextcloud depende del NAS
- Arquitectura arm64, algunos software no son compatibles
- Tolerancia a fallo 1, si cae más de un servidor no daría servicio

1.2 Introducció

1.2.1 Què és l'emmagatzematge en el núvol?

L'emmagatzematge en el núvol és un mètode d'emmagatzematge i organització de dades que es realitza en el núvol, una xarxa de servidors remots als quals es pot accedir mitjançant una connexió a Internet. Amb l'emmagatzematge en el núvol, els usuaris i les empreses poden emmagatzemar, accedir i mantindre les seues dades des de qualsevol lloc amb una connexió a Internet, en lloc d'haver de guardar els seus arxius en una única ubicació o dispositiu.

1.2.2 Què és RaspiVault?

RaspiVault és un conjunt de raspberry pi 4 que funcionen com a servidors en alta disponibilitat (HA) oferint un servici d'emmagatzematge en el núvol, en este cas utilitzaré nextcloud com a programari, totes les dades que maneja nextcloud (inclòs el propi programari) s'emmagatzemen en un synology, estiga synology s'encarrega d'emmagatzemar la base de dades i de crear còpies de seguretat per a resguardar la informació dels clients.

1.2.3 Beneficis d'implementar RaspiVault

- Gran portabilitat.
- Els servidors ocupen poc espai, a penes fan soroll.
- Ràpid i senzill d'instal·lar.
- No has de pagar quotes per l'ús d'emmagatzematge, solament el preu del maquinari.
- No té costos addicionals en llicències ni programari.
- El programari és totalment compatible amb un altre maquinari, és a dir podríem canviar el maquinari per a oferir un major rendiment al servici i un major emmagatzematge.
- És econòmic, per a un entorn reduït és la millor opció.

1.2.4 Inconvenients de RaspiVault

- Maquinari limitat
- Dolenta escalabilitat, s'hauria de canviar el maquinari
- El NAS que utilitze no suporta molt trànsit simultani
- Nextcloud depén del NAS
- Arquitectura arm64, algun programari no són compatibles
- Tolerància a fallada 1, si cau més d'un servidor no donaria servici

1.3 *Introduction*

1.3.1 What is cloud storage?

Cloud storage is a method of storing and organizing data that takes place in the cloud, a network of remote servers that can be accessed via an Internet connection. With cloud storage, users and businesses can store, access and maintain their data from anywhere with an Internet connection, rather than having to store their files in a single location or device.

1.3.2 What is RaspiVault?

RaspiVault is a set of raspberry pi 4 that work as high availability (HA) servers offering a cloud storage service, in this case I will use nextcloud as software, all the data that nextcloud handles (including the software itself) is stored in a synology, this synology is responsible for storing the database and creating backups to safeguard customer information.

1.3.3 Benefits of implementing RaspiVault

- High portability.
- Servers take up very little space, hardly make any noise.
- Quick and easy to install.
- No storage usage fees, just the price of the hardware.
- No additional licensing or software costs.
- The software is fully compatible with other hardware, i.e. we could change the hardware to offer a higher performance service and more storage.
- It is economical, for a small environment it is the best option.

1.3.4 Disadvantages of RaspiVault

- Limited hardware
- Poor scalability, hardware should be changed.
- NAS I use does not support a lot of concurrent traffic
- Nextcloud depends on NAS
- Arm64 architecture, some software is not supported
- Fault tolerance 1, if more than one server goes down it would not provide service

2.Descripción del proyecto y Objetivos

2.1 Descripción del proyecto

Mi objetivo para el proyecto es crear un almacenamiento en la nube con dos raspberry pi, para ello utilizaré nextcloud como software para lograr este objetivo.

Instalando nextcloud por si solo funciona y pero no cumple con mi objetivo, mi idea es crear una infraestructura para conseguir un servicio mucho mejor que un simple nextcloud, para ello instalaré haproxy en cada raspberry, para tener un servicio en alta disponibilidad, complementaré la alta disponibilidad con keepalived para crear un ip flotante para que el cliente ni se de cuenta de si algún servidor cae.

El software de nextcloud estará alojado junto a la base de datos en el synology conectado por red, el synology solo se podrá acceder desde el servidor.

2.2 Objetivos

El objetivo de mi proyecto es crear un servicio de almacenamiento en la nube con usuarios ldap y que los servidores sean tolerante a fallos y compartan información para que no importe que servidor estés conectado, tendrás los mismos resultados.

Los objetivos técnicos que quiero cumplir en este proyecto son:

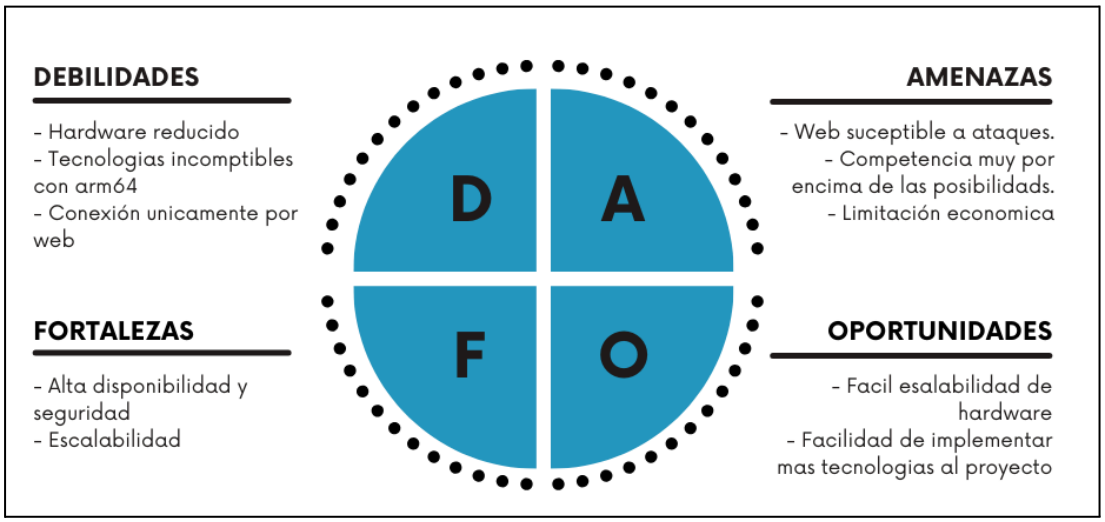
- Crear alta disponibilidad entre dos raspberry
- Hacer copias de seguridad de los archivos de los clientes con synology
- Utilizar una aplicación de almacenamiento en la nube de código abierto.
- Implementar DNS
- Aumentar la seguridad para mantener la privacidad y seguridad de los archivos.
- Implementar una app de notas con guardado en la nube.
- Adaptar todo el software a una raspberry pi de forma eficiente.
- Monitorización con nagios.
- Configurar firewall para bloqueo de accesos indebidos.
- Añadir verificación en dos pasos.

3.Estudio de viabilidad

3.1 DAFO

ANÁLISIS DAFO

El análisis **DAFO** son siglas que representan el estudio de las Debilidades, Amenazas, Fortalezas y Oportunidades de una empresa en un mercado.



3.2 Recursos de Hardware

Para poder realizar este proyecto he necesitado varios recursos hardware, entre ellos he utilizado:

- **2 x Raspberry PI 4B con 8GB de Ram:** he adaptado este hardware para crear los dos servidores que almacenan los servicios que ofrecemos a los clientes.
- **Mikrotik hEX Lite:** El mikrotik hex lite es el router que elegí para poder crear y personalizar varias redes dentro de mi entorno, para que no tengan conexión directamente.
- **Synology 212j:** El NAS que he utilizado para mi configuración y mi almacenamiento es el el synology 212j.
- **5 Cables ethernet:** para poder conectar la red y puedan comunicarse.
- **Flipper Zero:** para verificar la seguridad del sistema de doble autenticación.

3.3 Diagrama de Gantt, Inicial y Final

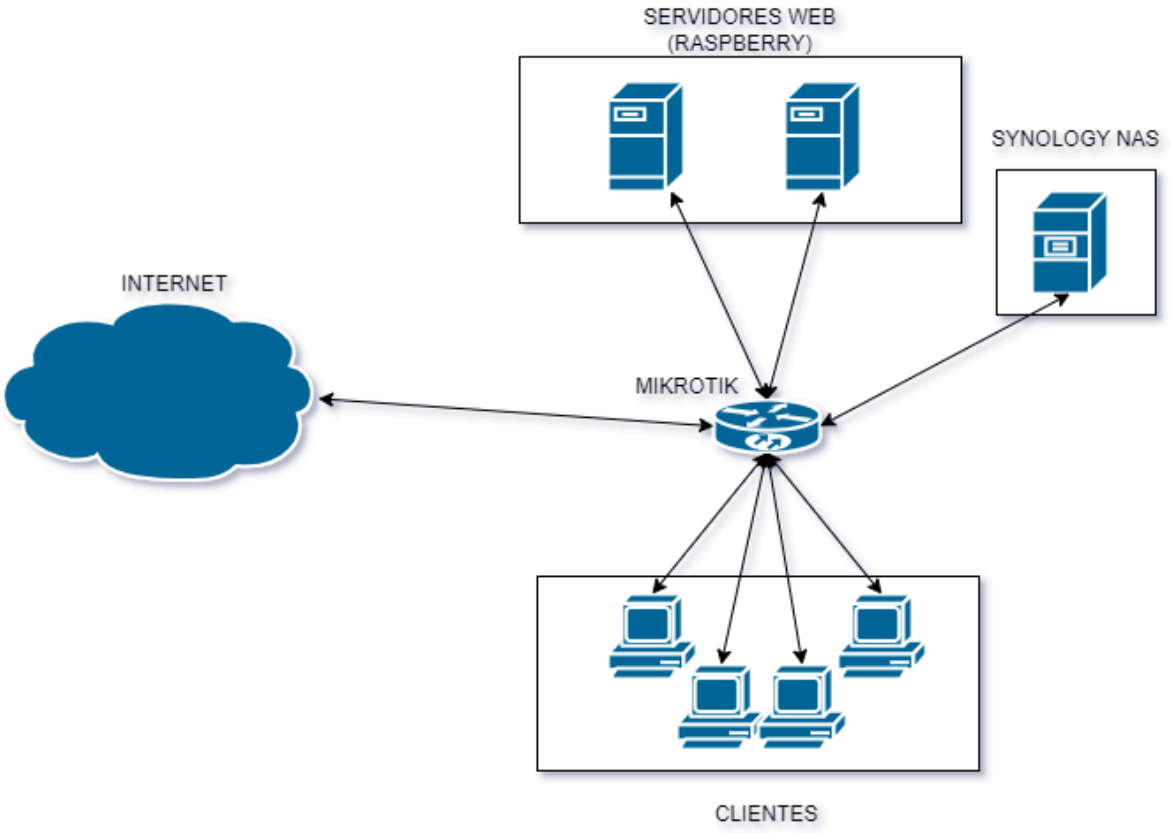
Planificación temporal diseñada inicialmente, el siguiente diagrama lo creé antes de empezar con el proyecto, antes de cambiar varias de los servicios que iba a utilizar y demás:

En el siguiente diagrama de gantt es el tiempo que he dedicado al final a cada cosa y me he dado cuenta que no todo llega a salir con el tiempo que habías planificado, ya que surgen problemas, se complica alguna cosa o una cosa da conflicto con otra que no tenías en cuenta, la primera parte de este diagrama de gantt tienen días sin asignar, eso es porque es el tiempo que dediqué a tecnologías que al final no llegué a implementar:

4.Diseño

4.1 Topología de la red

El tipo de topología que voy a usar será el tipo **estrella** para poder tener control total de la red a través del router mikrotik, todos las máquinas deben pasar por él , pudiendo bloquear el tráfico que yo desee además de restringir el acceso directo al nas y los servidores, esta topología me ofrece un gran **control sobre la red** y una **alta seguridad**, además de poder **escalar fácilmente**.



Para este proyecto he separado las máquinas en diferentes redes, tengo creado tres redes diferentes, una red para servidores, otra red para la red de administración y la última red para máquinas que se conectarán al nextcloud como usuarios.

Red para servidores: En esta red se ubican los dos servidores que estará servidor web y más servicios, también estará el nas en esta red.

Red para Administración: En esta red sólo tendrá una máquina (host), será mi maquina donde configuraré todo.

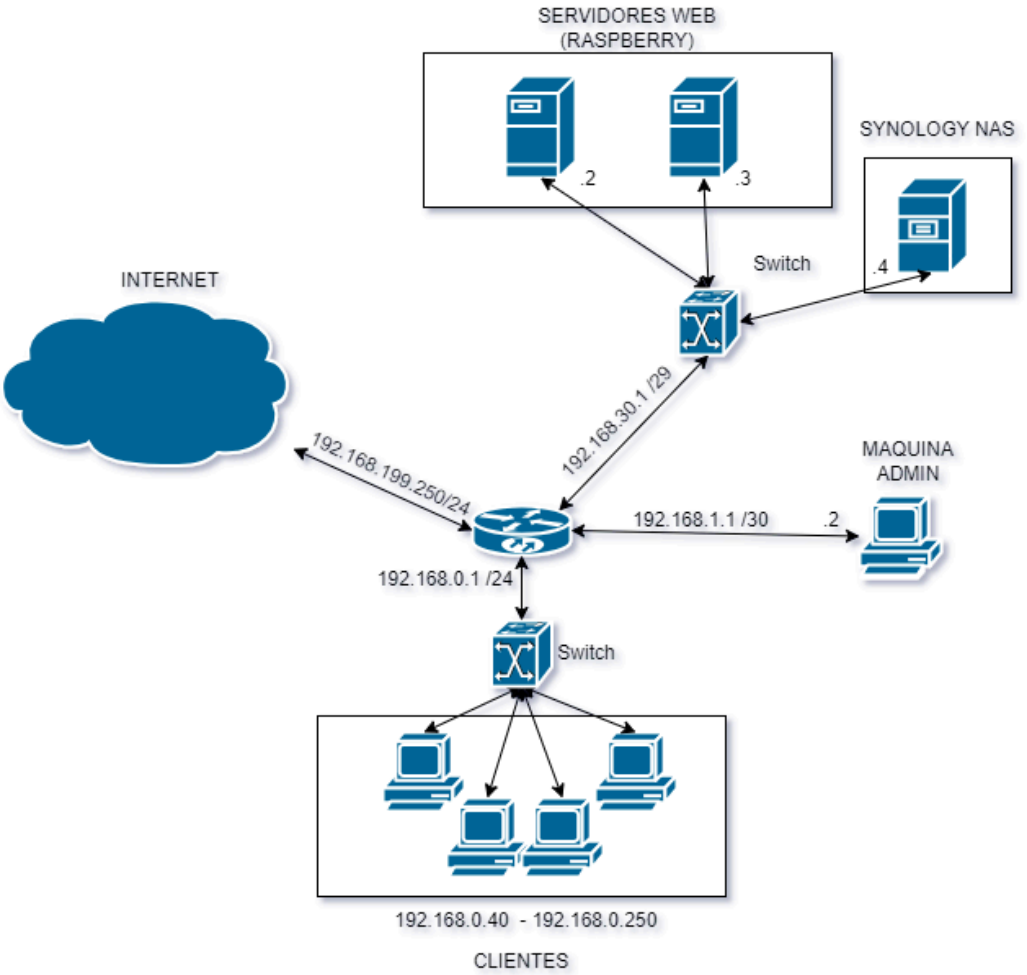
Red para Máquinas: Esta red será solamente para los usuarios que se conecten como clientes a través de las máquinas, esta red será la única que tendrá DHCP.

4.2 Diseño de red

Para la configuración de la red voy a utilizar **tres redes**, una red la utilizaré para las máquinas de los **usuarios**, otra red para el ordenador de **administración** y por último, la última red la utilizaré para los **servidores**, a continuación mostraré una tabla con las redes de mi mikrotik:

	NETWORK	IP	NETMASK	TYPE	NAME
eth1	D - 192.168.199.0	192.168.199.250	255.255.255.0 / 24	WAN	INTERNET
eth2	192.168.30.0	192.168.30.1	255.255.255.248 / 29	LAN	SERVERS
eth3	192.168.0.0	192.168.0.1	255.255.255.0 / 24	LAN	USUARIOS
eth4	—	—	—	LAN	EMPTY
eth5	192.168.1.0	192.168.1.1	255.255.255.252 / 30	LAN	ADMIN

La tabla anterior es como debería quedar, pero en mi caso al no disponer de switches tengo el eth2,eth3 y eth4 para los servidores y el eth5 para administración porque de momento no hay máquinas cliente, cuando sí haya me conectaré desde fuera del router y el eth5 pasará a ser de maquinas



5.Implementación

5.1 Tecnologías descartadas

Cuando empecé el proyecto determiné varias tecnologías y también tecnologías que me he topado que podrían haber sido candidatas para el proyecto pero por algunos inconvenientes no llegué a darles uso, en cambio he podido otras tecnologías que las sustituyeran.

5.1.1 Pimox

Proxmox es un software de virtualización que sirve para crear máquinas virtuales y contenedores, este software incluye opciones de cluster y alta disponibilidad y muchas otras funciones, investigando un poco me di cuenta que este software se adaptaba a las necesidades que yo tenía.

Intenté implementar este software en mi proyecto, pero cuando investigue más profundamente vi que este software **no** es **compatible con** la arquitectura **arm64**, que es la arquitectura de raspberry, investigando un poco mas me topé con un proyecto en github llamado "**Pimox7**", este proyecto adapta proxmox a arm64.
Cuando decidí implementar este software me surgieron varios problemas por los cuales decidí descartar esta adaptación de proxmox.

Uno de los **problemas** que tuve fue la **limitación** de **raspberry**, es decir raspberry no está pensada para virtualizar máquinas, entonces se queda muy corto en rendimiento para esta tarea, es decir se puede virtualizar y crear máquinas? Si, incluso llegué a crear un par pero iban cortas de rendimiento y no daban el mínimo que buscaba para poder realizar el proyecto que quería realizar, y por la parte de contenedores no pude crear ninguno, básicamente porque no encontré ninguna plantilla en arquitectura arm64, así que decidí no implementar este software y utilizar haproxy y keepalived.

5.1.2 Docker swarm/Docker

Docker Swarm es una herramienta software que permite ejecutar los contenedores en una granja de nodos, esto implica uno o varios balanceadores de carga implementados en uno o varios nodos maestros y los nodos que prestan el servicio, implementados en nodos trabajadores.

Cuando planteé la idea de proyecto pensé en hacerlo en docker y complementar con **docker swarm** para realizar un **cluster**, pero conforme iba realizando el proyecto pensé en hacerlo en máquinas virtuales con proxmox, y como al final no pude implementar proxmox como yo deseaba me decanté por implementar los servicios en el propio servidor **sin máquinas virtuales ni dockers**, ya que el servicios que iba a utilizar tampoco eran muy demandantes, así que realice el **cluster** con **haproxy y keepalived.**

5.1.3 Obsidian

Obsidian es una base de conocimientos personales y una aplicación para tomar **notas** que opera con archivos **Markdown**. Permite a los usuarios crear enlaces internos para las notas y luego visualizar las conexiones como un grafo.

Al principio del proyecto descubrí obsidian, un software libre que permite crear notas, me pareció buena idea **implementarlo en el nextcloud**, pero cuando ya configuré y implementé el almacenamiento en la nube nextcloud vi que el gestor de **notas de nextcloud cumplia** con mis **expectativas**, así que decidí no implementar Obsidian y utilizar las notas que proporciona nextcloud.

5.1.4 LDAP Synology

El protocolo ligero de acceso a directorios (**LDAP**) hace referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Synology tiene la opción de crear un **servicio LDAP**, para implementarlo debes descargar la aplicación de la app de aplicaciones del propio synology, cuando pensé la idea de crear el ldap dentro del synology me pareció muy buena idea, ya que estaría todo en un mismo sitio y no consumiría recursos de los servidores, pero más tarde cuando llegue a implementarlo tenía bastantes **problemas** para loguear un **usuario ldap desde fuera del synology con interfaz gráfica**, además de eso no tenía mucha opción a cambiar configuración del ldap, era bastante simple, así que decidí crear un servicio con OpenLDAP dentro de los servidores.

El **OpenLDAP** también tenía algún inconveniente, por ejemplo si lo implementaba solamente en un solo servidor no sería tolerante a fallos e iría en dirección contraría de la línea que estoy siguiendo de que todos los **servicios** sean **tolerante a fallos** y OpenLDAP no tiene una forma sencilla y rápida de tener dos servicios comunicados y que compartan información, así que me decante por hacer un **MirrorMode**, lo que hace el MirrorMode es crear dos usuarios en los dos servidores y que esos dos usuarios se comuniquen entre ellos y **repliquen** absolutamente todo.

5.1.5 Flipper zero

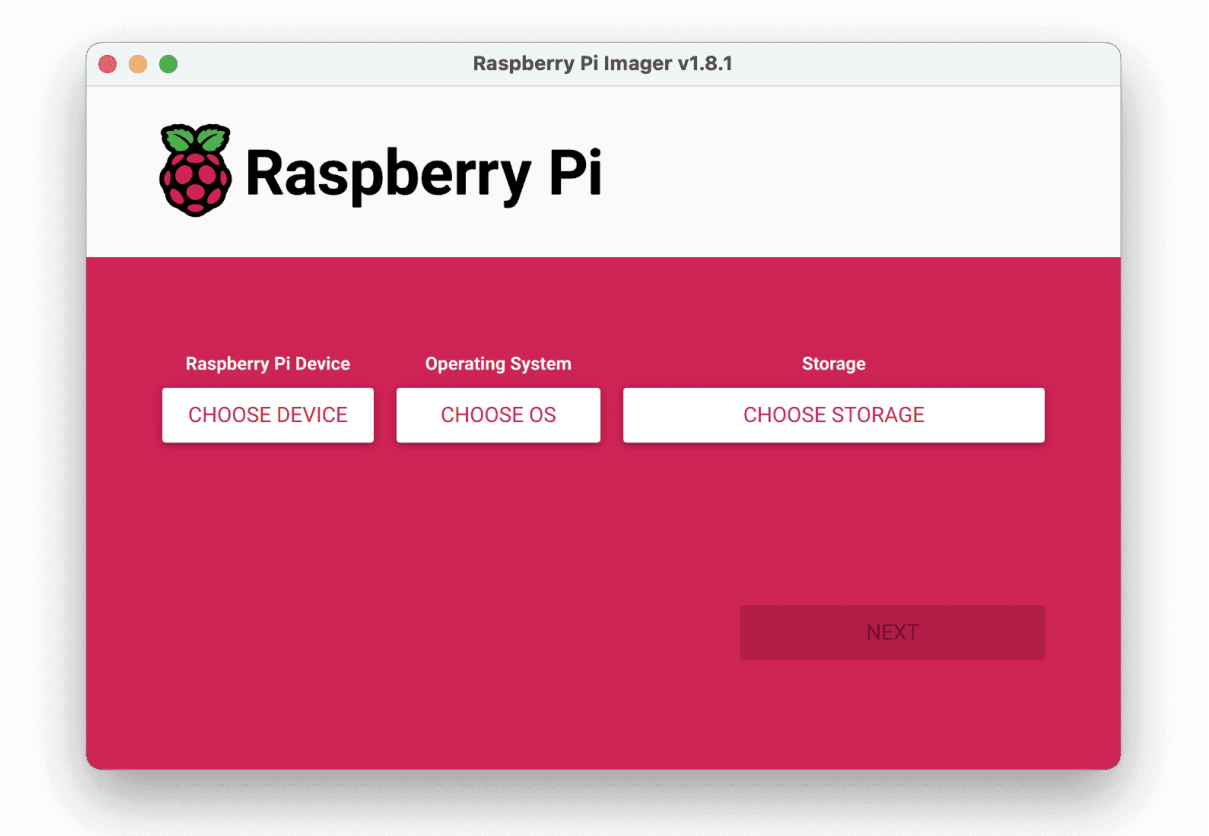
Flipper Zero es un dispositivo multifuncional portátil similar a Tamagotchi desarrollado para interactuar con sistemas de control de acceso. El dispositivo es capaz de leer, copiar y emular etiquetas RFID y **NFC**, mandos a distancia por radio, iButton y llaves de acceso digitales, junto con una interfaz GPIO.

Cuando propuse la idea de proyecto pensé en **implementar un F2P con NFC** y poder vulnerar esta doble verificación con el flipper zero duplicando una tarjeta NFC, pero **no encontré la forma** de poder hacer una doble verificación con NFC en la página web de nextcloud.

5.2 Instalación y configuración de la raspberry

Para poder instalar el S.O. de raspberry en la tarjeta sd primero voy a instalarlo en un usb boot, lo iniciaré desde el usb y dentro del raspberry OS iniciado por el usb podemos instalar en ambas tarjeta el sistema operativo de raspberry con la aplicación "**imager**" (Viene preinstalada en el sistema operativo de raspberry, pero también se encuentra en su página oficial) que sirve para instalar un sistema de raspberry en cualquier almacenamiento que seleccionemos.

Un detalle a tener en cuenta es que debemos de tener conexión a la hora de realizar la instalación para que sea más sencillo, porque si tenemos conexión a internet podemos instalar un sistema operativo que nos proporcione la aplicación, si no debemos de descargar la iso por nuestra cuento y conectar otro usb con la iso.



Una vez tenemos la instalación hecha en cada micro SD ya podemos reiniciar el equipo retirando el usb que hemos utilizado para instalar los otros S.O. En mi caso he instalado el "**Raspberry OS Lite de 64 bits bookworm**" (Debian 12 con pequeñas modificaciones por parte de raspberry), he elegido la opción lite porque es la que menos recursos consume, tampoco necesitaré la interfaz gráfica ya que realizaré la mayor parte por ssh.

Una vez tenemos iniciado el sistema por la tarjeta SD vamos a hacer un update seguido de un upgrade para tener los últimos paquetes actualizados, cuando ya tengamos todo actualizado vamos a poner la ip statica (anteriormente tenía un ip asignada por DHCP), en mi caso el primer servidor tendrá la "**192.168.30.2**" y el segundo "**192.168.30.3**"

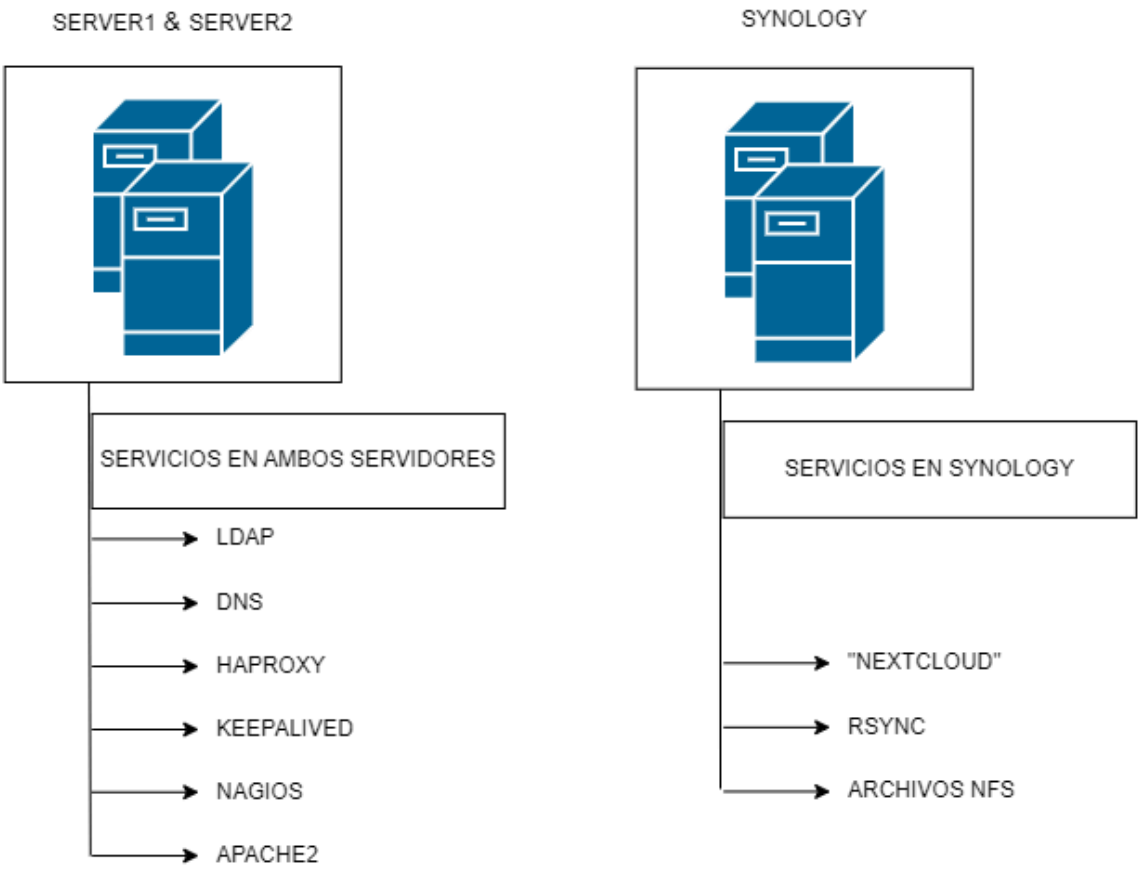
5.3 Configuración Alta disponibilidad

El objetivo es tener dos nextcloud en **HA (alta disponibilidad)** para proporcionar servicio con tolerancia a fallos, tengo configurado un servidor principal que tiene función como **master**, y el otro servidor está en modo **esclavo**, de esta forma si un servidor falla hay otro servidor a la espera para poder ofrecer servicio al instante.

Cada servidor, es decir cada raspberry pi tendrá los mismos servicios corriendo, ambos servidores tedndrán un **haproxy** para balancear la carga entre ambos servidores, se comunican entre sí y ya están a la espera de conexiones, para complementar este servicio voy a añadir **keepalived** para tener una IP flotante (ip única para dos máquinas),y configurar la prioridad de cada servidor.

Además del balanceo de carga voy a instalar un **DNS primario** para el servidor que va a ser el maestro y un **DNS secundario** para el servidor esclavo para tener servicio de dns independientemente de que servidor esté levantado.Por último pondré un servidor **LDAP** y la página web (nextcloud) en el synology.

SERVICIOS QUE UTILIZAN LOS SERVIDORES



5.3.1 Keepalived

Entramos primero a la configuración de keepalived, el archivo de configuración no está creado, debemos de crearlo y modificarlo. En mi caso tendré esta configuración:

```
# /etc/keepalived/keepalived.conf
# serv1.raspivault.com
global_defs {
        enable_script_security
        script_user haproxy
        notification_email {
                admin@raspivault.com
        }
        smtp_server 127.0.0.1
        router_id Server01
}
vrrp_script haproxy {
        script "/usr/bin/killall -0 haproxy"
        interval 2
        timeout 1
}
vrrp_instance ELB {
        state MASTER
        interface eth0
        virtual_router_id 51
        priority 255
        advert_int 1
        authentication {
                auth_type PASS
                auth_pass powerpassword@
        }
        virtual_ipaddress {
                192.168.30.5       # IP FLOTANTE
        }
        track_script {
                chk_haproxy
        }
}
```

Debemos de replicar lo mismo en el otro raspberry cambiando el “route_id” , “priority” y “state”, en mi caso el servidor 02 será el esclavo así que le bajare la prioridad “priority 205”, el stat a “state BACKUP” y “route_id Server02”.

- **router_id:** aquí añadimos el hostname de la máquina.
- **state:** MASTER para el activo y BACKUP para el pasivo.
- **interface**: El nombre de nuestra interfaz de red, podemos poner varias con el formato interfaces { … }
- **virtual_router_id:** Es un número identificativo único por cada instancia vrrp.
- **priority:** Un número entre 1 y 255, lo ideal es que el máster tenga una diferencia de 50 respecto al primer pasivo.
- **advert_int**: El intervalo en segundos del VRRP.
- **authentication:** Autenticación para sincronizar el failover.
- **virtual_ipaddress:** La IP flotante del cluster.
- **smtp_alert**: Activa la notificación por emails para esa instancia de vrrp.

5.3.2 Haproxy

El servicio de haproxy es el encargado de balancear la carga del servicio web, el es el encargado de decidir si debe correr un servidor u otro, dependiendo de la configuración el decide si se va a mostrar un servidor u otro, en mi caso lo he implementado ya que si se cae un servidor se levanta el otro servicio sin que el usuario cliente no note nada, únicamente se reinicia la página manteniendo la sesión iniciada.

Por último he añadido las siguiente configuración en el servicio de haproxy, en el archivo de configuración:

```
# /etc/haproxy/haproxy.cfg
#Arriba la configuración que ya viene por defecto

frontend http
        bind 192.168.30.5:80       # PUERTO ESCUCHA HAPROXY
        acl host_app_sysadm hdr(host) -i server01
backend app
        balance leastconn
        option httpclose
        option forwardfor
        server Server01 127.0.0.1:1000
```

- Esta configuración debe de hacerse en ambos servidores.

5.4. Configuración para Nextcloud

Nextcloud es un software de almacenamiento en la nube que utiliza dos servicios, uno es **apache2** y **PHP** para el funcionamiento de la página, además se recomienda que se use MariaDB como base de datos, pero en mi caso utilizaré **SQLITE** ya que es mucho más ligero y mi hardware, en este caso el synology no tiene suficientes recursos para mover MariaDB correctamente.

Una vez instalo el apache deshabilito el archivo default y activo mi archivo de configuración

```
root@Server01:/home/server1# a2dissite 000-default.conf
root@Server01:/home/server1# a2ensite nextcloud.conf
```

Tengo cambiado el puerto del archivo "**nextcloud.conf**", en vez de el puerto 80 usaremos el **puerto 1000**, a parte de cambiar el puerto también debemos cambiar el default root a "**/var/www/nextcloud**".
Añadir el puerto de escucha de apache2, ahora iremos al archivo "/etc/apache2/ports.conf" y comentaremos la línea de "Listen 80" y añadiremos el puerto 1000 con "Listen 1000".

5.4.1 PHP

Una vez tenemos el apache bien configurado a nuestras necesidades vamos a instalar PHP, Para instalar php para nextcloud debemos de instalar más de un paquete, ya que nextcloud necesita paquetes complementarios de PHP y un par de paquetes para sqlite para poder usar Sqlite como base de datos de nextcloud.

Estos son los parámetros que he cambiado en mi archivo de configuración de php **enfocado** para **nextcloud**:

```
#  /etc/php/8.2/apache2/php.ini
date.timezone = Europe/Madrid
memory_limit = 512M
upload_max_filesize = 500M
post_max_size = 600M
max_execution_time = 300
file_uploads = On
allow_url_fopen = On
output_buffering = Off
zend_extension=opcache
opcache.enable = 1
opcache.interned_strings_buffer = 8
opcache.max_accelerated_files = 10000
opcache.memory_consumption = 128
opcache.save_comments = 1
opcache.revalidate_freq = 1
```

- Esta configuración debe de hacerse en ambos servidores.

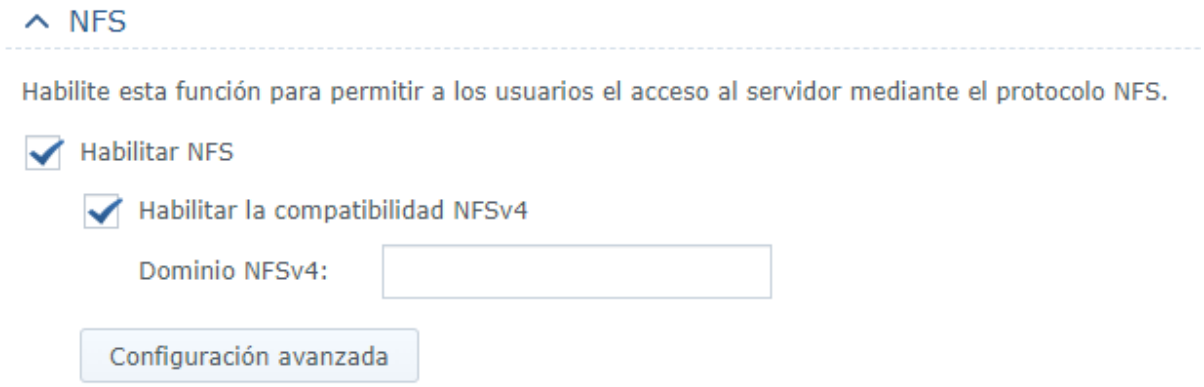
5.5 Instalación y configuración del NAS

Nextcloud necesita un almacenamiento para almacenar sus datos y las bases de datos, este almacenamiento puede ser local o remoto, en mi caso al querer sistema tolerante a fallos ubicaré los ficheros y las BBDD de nextcloud en remoto, para ello utilizare un equipo synology, en concreto el DS212J, para hacer el almacenamiento remoto para el nextcloud lo que haré es crear directorios compartidos, usaré el **protocolo NFS** para poder compartir directorios por red (synology con IP: 192.168.30.4 / 29), con este sistema para compartir directorios mis servidores estarán conectados a este directorio y solo tendré que cambiar la configuración del Nextcloud una vez al igual que la base de datos, ya que ambos servidores utilizarán el mismo recursos.

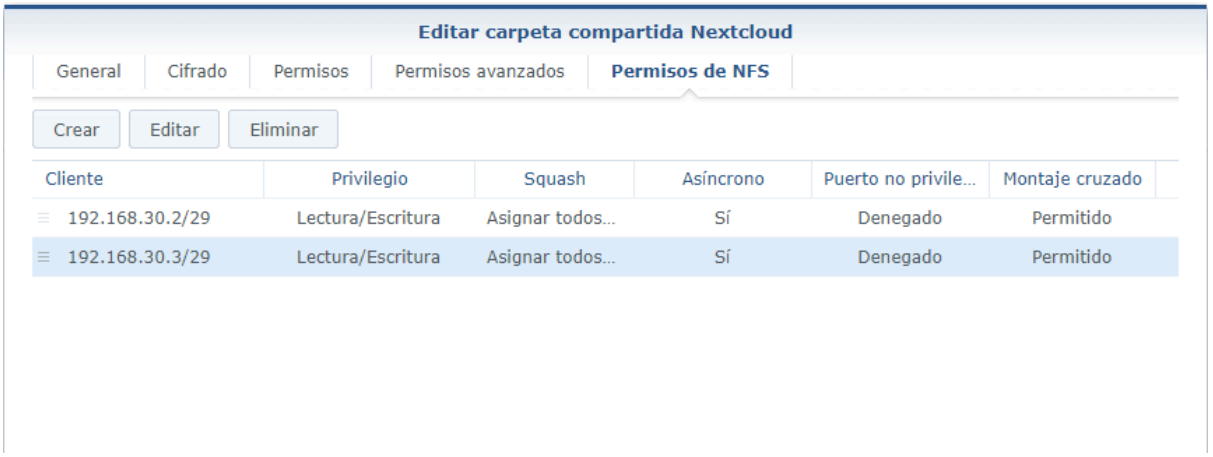
5.5.1 Synology

Para poder compartir la carpeta desde el synology hay que habilitar la opción de NFS, crear la carpeta, compartirla y dar permisos a las máquinas que deseemos.

Debe de tener el NFS habilitado:



Cuando ya esté el NFS habilitado debemos de tener la carpeta creada y con los permisos que toque, yo he creado un usuario específico en el synology para esta carpeta, aparte hay que asignar que IPs tienen acceso a la carpeta compartida:



En la anterior captura muestro las dos IPs que tienen acceso, que son las IPs de los servidores.

- **Nombre de host o IP:**
  Introduzca la dirección IP del cliente NFS que accede a la carpeta compartida.
- **Privilegio:** Seleccione permisos de lectura/escritura para el cliente NFS.
- **Squash:** Este campo le permite controlar los privilegios de acceso del usuario del cliente NFS.
- **Habilitar asíncrono:**
  Si activa esta opción, su Synology NAS podrá responder a las peticiones de clientes NFS antes de que se complete cualquier cambio en los archivos, lo que proporciona un mayor rendimiento.
- **Permitir conexiones desde puertos no privilegiados (puertos superiores a 1024):**
  Si activa esta opción permitirá a sus clientes de NFS utilizar puertos no privilegiados (es decir, puertos mayores de 1024) cuando se conecten a Synology NAS.
- **Permitir a los usuarios acceder a las subcarpetas montadas:**
  Si activa esta opción permitirá a sus clientes de NFS que accedan a subcarpetas montadas.

5.5.2 NFS en Raspberry

Para hacer que el synology comparta archivos con ambos servidores voy a utilizar el protocolo NFS, crearé una carpeta en el synology y la compartiré por NFS y les daré permiso solamente a ambos servidores, y desde los servidores solamente montaré las carpetas y apuntaré con un enlace simbólico a la ruta de montaje.

Para montar manualmente la carpeta lo he hecho de esta manera:

```
sudo mount -t nfs 192.168.30.4:/volume1/nextcloud /mnt/synology
```

Primero indicamos la ip de la vm seguido de dos puntos y la ubicación de la carpeta dentro del synology y luego la carpeta que acabamos de crear para tener el directorio externo representado en nuestro sistema.

Para terminar vamos a indicar en "**/etc/fstab**" el montaje del directorio para que se realice automáticamente cada vez que arranquemos el sistema:

```
192.168.30.4:/nextcloud /mnt/synology nfs defaults,noatime 0 0
```

```
root@Server01:/home/server1# cat /etc/fstab
proc            /proc           proc    defaults          0       0
PARTUUID=e1752efc-01  /boot/firmware  vfat    defaults          0       2
PARTUUID=e1752efc-02  /               ext4    defaults,noatime  0       1
192.168.30.4:/volume1/Nextcloud /mnt/synology/  nfs     defaults,noatime 0 0
192.168.30.4:/volume1/homes /mnt/ldap/ nfs defaults,noatime 0 0
# a swapfile is not a swap partition, no line here
#   use  dphys-swapfile swap[on|off]  for that
root@Server01:/home/server1#
```

5.5.3 Nextcloud

Para instalar **nextcloud** hay que descargar un comprimido de la página oficial de nextcloud y descomprimirla en el lugar que necesitemos, en mi caso lo he extraido en la carpeta compartida que he creado antes, debemos de descomprimir la **carpeta dentro del synology** para que ambos servidores tengan una misma configuración compartida.

```
cd /mnt/synology            # Creo nextcloud en el synology
curl -o nextcloud.zip
https://download.nextcloud.com/server/releases/latest.zip
unzip nextcloud.zip
sudo chown -R www-data:www-data nextcloud
```

Ahora vamos a **aplicar el nextcloud al apache** que ya teníamos creado, cambiaremos el document root, añadiremos nameserver y además una configuración del directorio creado en “/var/www/nextcloud”:

```
<VirtualHost *:1000>
        ServerName nextcloud.natxo.rasp
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/nextcloud
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /var/www/nextcloud/>
        Options +FollowSymlinks
        AllowOverride All
        <IfModule mod_dav.c>
        Dav off
        </IfModule>
        SetEnv HOME /var/www/nextcloud
        SetEnv HTTP_HOME /var/www/nextcloud

</Directory>
</VirtualHost>
```

Para que el apache apunte a la página web sin salirse directamente de /var/www he creado un enlace simbólico desde /var/www a /mnt/synology, para que desde el apache pueda acceder a la página de nextcloud

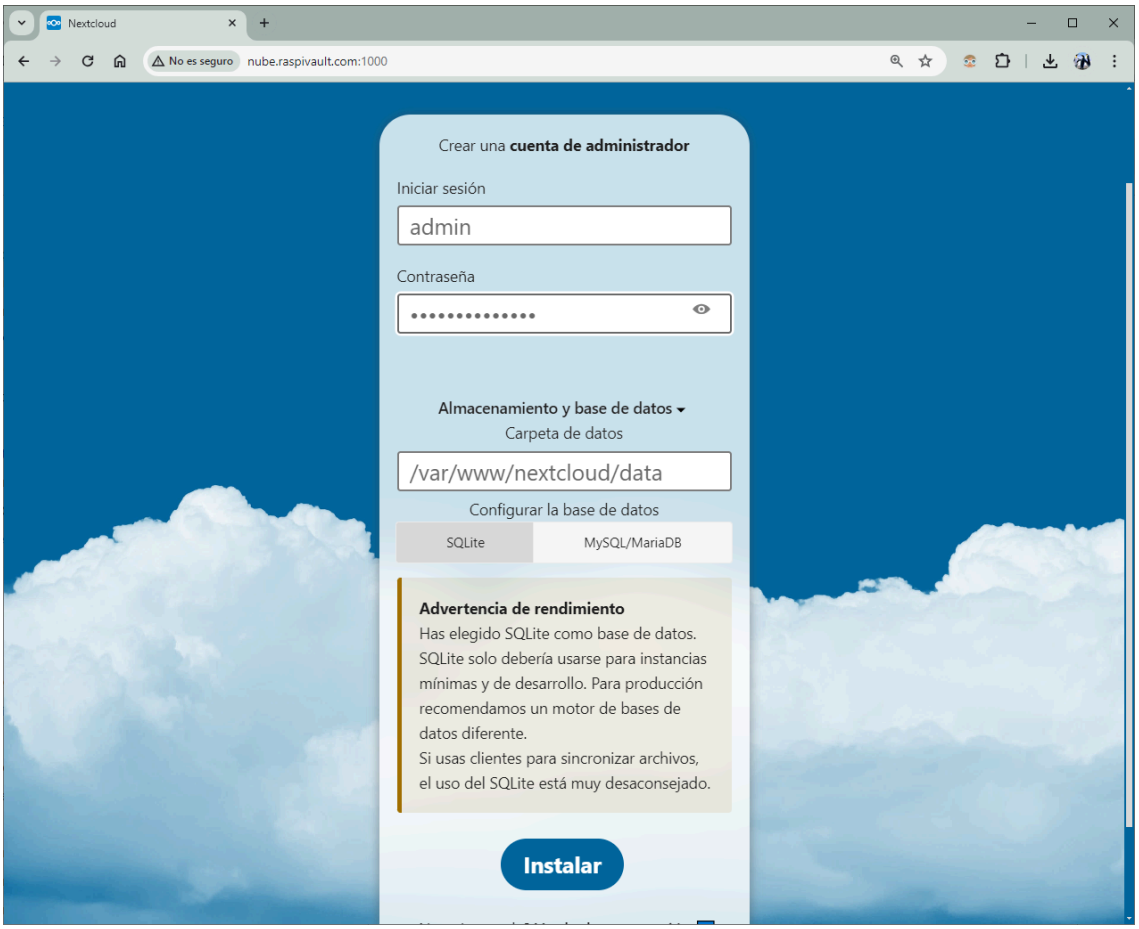
```
cd /var/www

ln -s /mnt/synology/nextcloud ./nextcloud

chown www-data:www-data /var/www/nextcloud
```

Con este comando ya tengo creado el enlace simbólico y el apache2 tendrá acceso a la página.

```
root@Server02:/home/server2# ls -l /var/www/
total 8
drwxr-xr-x 2 root     root     4096 Apr 28 21:51 html
lrwxrwxrwx 1 www-data www-data   23 May  2 12:38 nextcloud -> /mnt/synology/nextcloud
drwxr-xr-x 2 root     root     4096 Apr 26 19:38 pagina2
root@Server02:/home/server2#
```

Una vez tenemos el enlace simbólico creado y el montaje de la carpeta NFS creado podemos ejecutar nextcloud a través de la página web, podemos hacer de ambas formas, o bien poniendo la ip flotante o bien poniendo el nombre que la hayamos asignado en el DNS.

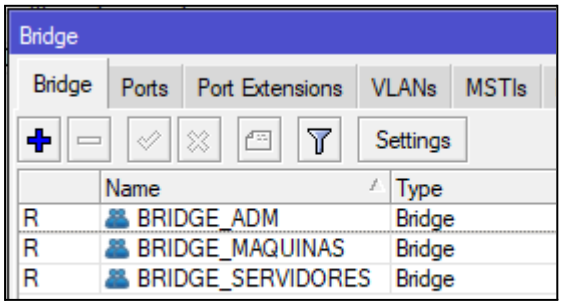
La primera vez que entramos no pedirá la base de datos y el usuario admin:

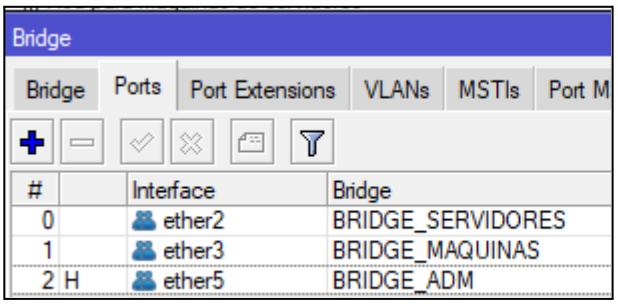


5.6. Configuración de la red

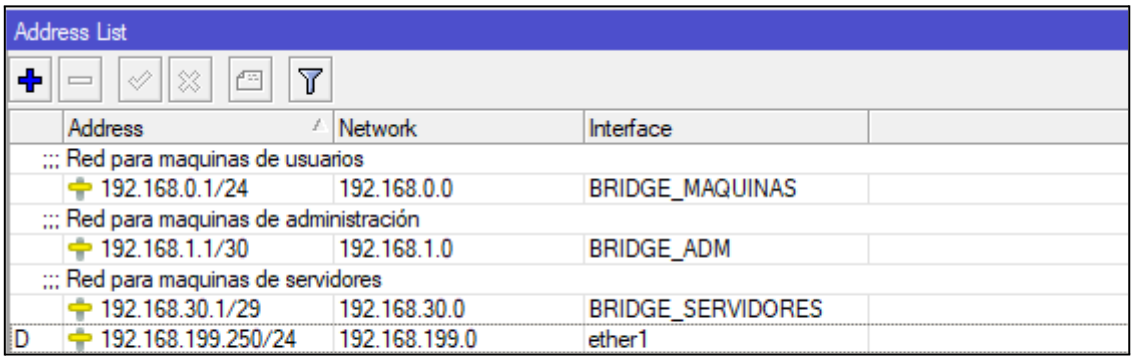
Para gestionar la red y crear varias redes con sus reglas de firewall necesitaré un router, en mi caso he utilizado un **Mikrotik**, y configuraré las redes tal como he dicho en el punto "4.diseño".

Para cada red he creado un bridge que les representa, este bridge irá asignado a su puerto ethernet correspondiente, y tendrá una red asignada, estos serán los bridge con su puerto:

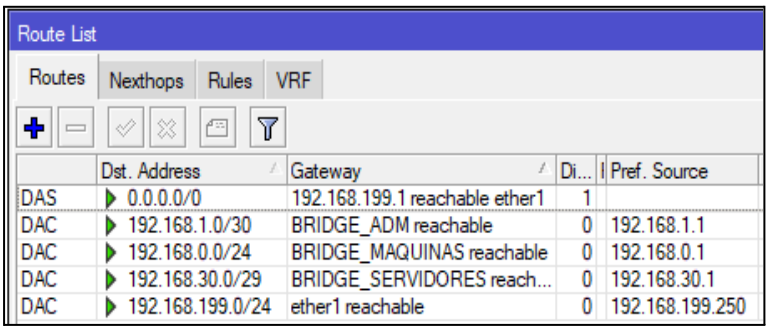


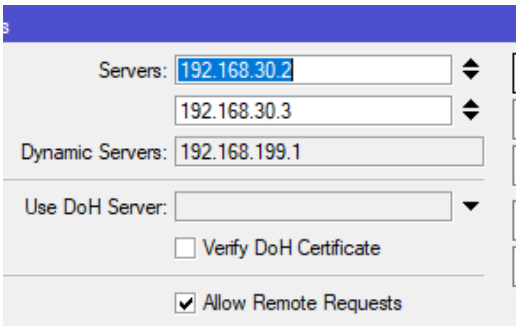


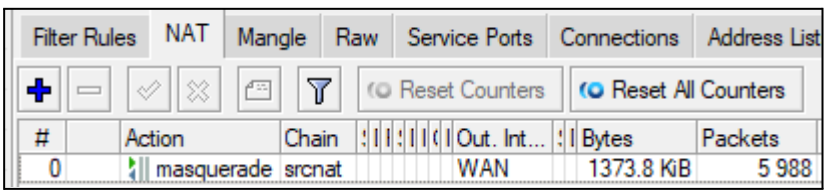
Una vez tengo creado los "bridge" debo asignarles una red y una Ip a cada uno, que será la puerta de enlace de cada red, debe de estar alineado con el apartado de diseño para que no haya confusiones de red y se puedan comunicar correctamente, si se pusiera la red mal las máquinas tendrían diferentes redes y no tendrían conexión con el exterior y además en el firewall se bloquean máquinas que no son. Este es el resumen de direccionamiento de mi esquema:



Aquí podremos ver las rutas que deberán seguir cada máquina de la red para poder salir a internet, y los DNS que deben de seguir para tener resolución de nombres:



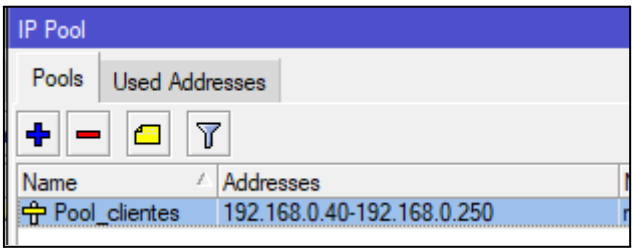




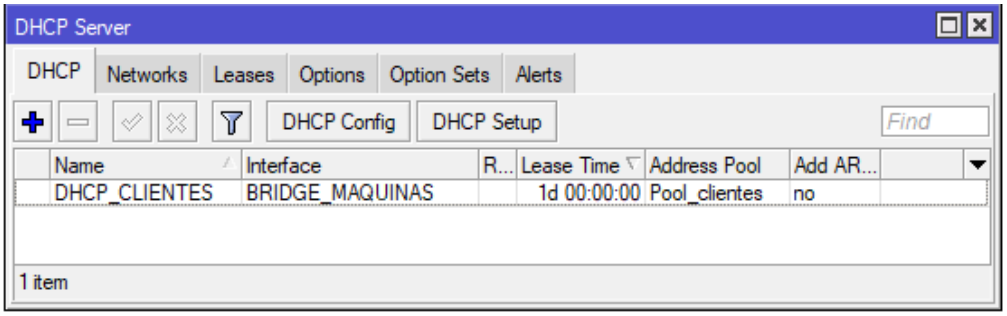
5.6.1 DHCP

Para el “Bridge” de las máquinas utilizaré dhcp, aprovechando que el mikrotik que poseo tiene opción a crear dhcp utilizaré esta herramienta para dar IP a los clientes que se conecten.

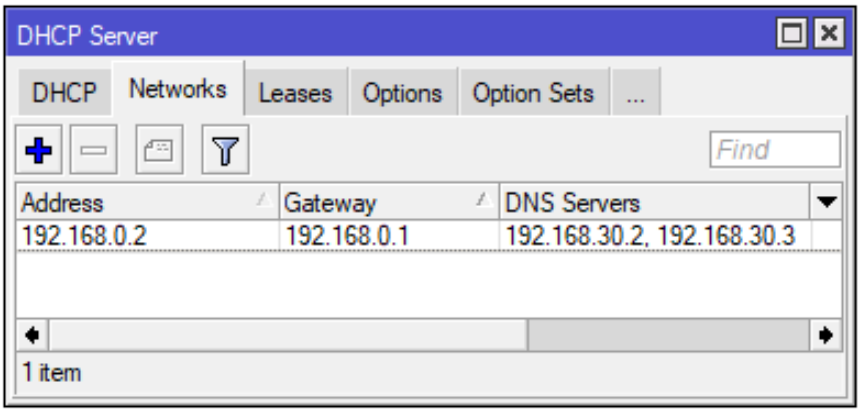
La red que he asignado a los clientes es “192.168.0.0/24”, para esta red usaré este rango de IPs que utilizará el DHCP: 192.168.0.40 - 192.168.0.250, este será el pool:



Y asignará el pool anterior al Bridge de máquinas la cual estará conectada a la interfaz correspondiente a las máquinas clientes, también pondré que la IP asignada a una máquina esté reservada durante 24h:



También he indicado el network, porque si no la máquina tendrá la IP que se le asigne pero no tendrá salida afuera ni podrá comunicarse por la red:



5.6.2. Firewall

Un firewall es un sistema de seguridad de red de las computadoras que restringe el tráfico de Internet entrante, saliente o dentro de una red privada. Este software o esta unidad de hardware y software dedicados funciona bloqueando o permitiendo los paquetes de datos de forma selectiva.

Como he decidido crear mis propias redes privadas también debo de securizar también el conjunto de redes, ya que si no asigno reglas para que filtre el router cualquiera podría ver y acceder a los servidores y otras máquinas desde el exterior, para ello he creado nueve reglas que permitirán o no conexiones y tráfico entre redes.

1. **Regla 1:** Aceptar todas las conexiones que estén establecidas y relacionadas, es decir si mi maquina de administrador o cualquier otra accede a internet y se establece una conexión con algún servicio ftp, este propio servicio puede mantener conexión y además puede transferir datos por otro puerto, ya que el servicio FTP donde se crea una conexión para comandos y otra diferente para la transferencia de datos.
2. **Regla 2:** En esta regla bloqueo las conexiones entrantes que sean inválidas, las reglas que bloquean conexiones inválidas rechazan paquetes que no pueden ser identificados como parte de una sesión establecida o relacionada, o que de alguna manera no siguen el protocolo esperado (como un paquete SYN en respuesta a un paquete SYN ya enviado).
3. **Regla 3:** En esta regla permito el tráfico que esté establecido y relacionado, esta regla es muy parecida a la primera regla, pero esta en cambio es permitir todo tráfico que atraviesa el router y que sea establecido y/o relacionado, en la primera regla permito la conexión que es de entrada y esté relacionado y/o establecido.
4. **Regla 4:** En esta regla permito el protocolo UDP por los puertos 111, 892, 2049 que son del servicio NFS que tiene el synology y el puerto 53 que es el puerto de DNS.
5. **Regla 5:** En esta regla permite los mismos puertos (11, 892, 2049 que son del servicio NFS que tiene el synology y el puerto 53 que es el puerto de DNS) más el puerto 389 que es el puerto que utiliza LDAP pero en este caso por TCP.
6. **Regla 6:** A parte de permitir los anteriores puerto permito la ip “192.168.30.5” que es la ip que tiene el keepalived (IP flotante).
7. **Regla 7:** En esta regla bloqueo todo tráfico que no haya permitido anteriormente en los puertos, para que las máquinas de los usuarios no tengan más acceso a los servidores.
8. **Regla 8:** En esta regla permito que la red de Administración pueda tener tráfico con otras redes, ya sea dentro del mikrotik o fuera.
9. **Regla 9**: En esta regla permito que la red de Servidores pueda tener tráfico con otras redes, ya sea dentro del mikrotik o fuera.
10. **Regla 10:** En esta regla permito que la red de Máquinas pueda tener tráfico con otras redes, ya sea dentro del mikrotik o fuera.

11. **Regla 11:** En esta regla corto todo el tráfico restante, es decir todo tráfico que no cumpla una regla permisiva que esté por encima de esta regla será bloqueada.

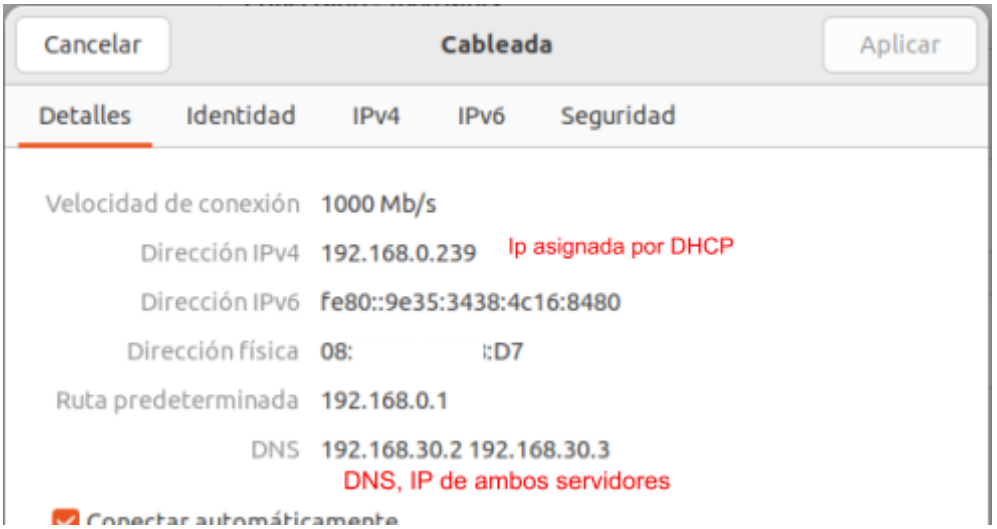
Captura de como quedaría:

#		Action	Chain	S..	D..	Proto...	S..	Dst. Port	In. Interface	Out. Interface	I.	O	S.	Dst. Addr...	Bytes	Packets
;;; Permitir conexiones establecidas y relacionadas																
0		acc...	input												102.9 KiB	1 048
;;; Denegar conexiones invalidas																
1		drop	input												0 B	0
;;; Permitir trafico relacionado y establecido																
2		acc...	forward												625.6 MiB	845 928
;;; Permitir conexión desde las maquinas por nfs y dns por udp a los servidores																
3		acc...	forward			17 (u...		111,892,...	BRIDGE_MAQUINAS					Servers	0 B	0
;;; Permitir conexión desde las maquinas por ldap, nfs y dns por tcp a los servidores																
4		acc...	forward			6 (tcp)		389,111,...	BRIDGE_MAQUINAS					Servers	0 B	0
;;; Permitir conexión entre maquinas y la ip flotatne																
5		acc...	forward						BRIDGE_MAQUINAS					ipFlotante	0 B	0
;;; Denegar conexión desde la red de maquinas a todos los servidores																
6		drop	forward						BRIDGE_MAQUINAS	BRIDGE_SERV...					0 B	0
;;; Permitira trafico a Administración																
7		acc...	forward						BRIDGE_ADM						6.4 MiB	15 665
;;; Permitir trafico a Servidores																
8		acc...	forward						BRIDGE_SERVIDO...						2336.6 KiB	25 965
;;; Permitir trafico a Maquinas clientes																
9		acc...	forward						BRIDGE_MAQUINAS						0 B	0
;;; Deneger resto de trafico																
10		drop	forward												0 B	0

5.7. Instalación y configuración de clientes

Para conectar un cliente a nuestra red y que obtenga el acceso a nuestro servicio de almacenamiento en la nube y el servicio adicional.

Para empezar primero de todo debemos de instalar un sistema operativo, en mi caso usaré un ubuntu con interfaz gráfica, luego se debe de configurar la red de la máquina cliente, para ello en la red debemos de configurar que obtenga ip a través de dhcp y asignarles los dos DNS que hemos creado, quedaría de la siguiente forma:

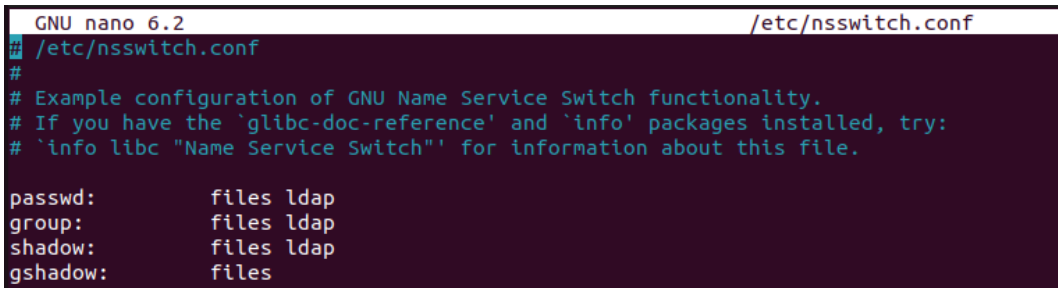


Una vez tenemos la red y hace ping a internet y a nuestro servidor DNS debemos de configurar el directorio compartido por NFS, para ello pondremos el montaje en “/etc/fstab”, este es la configuración que he asignado a la carpeta de HomeDirectory de los usuarios LDAP (para poder usar el formato nfs necesitaremos el paquete “nfs-kernel-server”):

```
192.168.30.4:/volume1/ldap                /mnt/ldap                nfs
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

Una vez tenemos el volumen montado en la carpeta correspondiente debemos tener instalados los paquetes necesarios (libnss-ldap libpam-ldap ldap-utils nslcd), cuando instalemos los paquetes debemos de poner las direcciónes de los servidores LDAP, con el dominio correspondiente y con su usuario administrador y contraseña.

Cuando ya tenemos la instalación correctamente vamos a cambiar el archivo "/etc/nsswitch.conf" e implementaremos donde queremos que se obtenga la información cuando iniciemos sesión, así debería de quedar el documento:



- Podemos comprobar que el sistema detecta los usuarios LDAP con el comando "getent passwd"

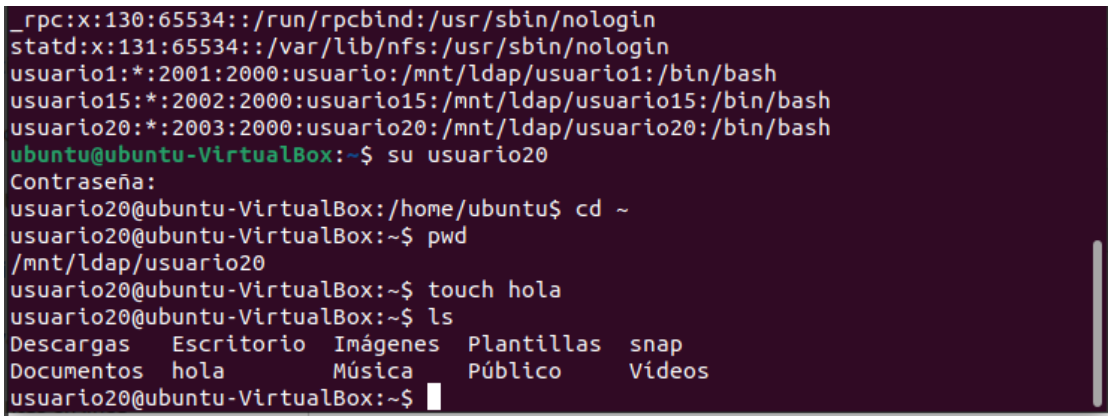
Además debemos de modificar el archivo "/etc/pam.d/common-session" para añadir la opción de loguearse con un usuario LDAP debería de parecerse a esta línea:

```
session optional        pam_mkhomedir.so skel=/etc/skel umask=077
```

Para poder loguearte correctamente con interfaz gráfica debemos de ejecutar el siguiente comando para que el sistema detecte que este directorio forma parte de los HOMEs de:

```
sudo snap set system homedirs=/mnt/ldap
```

- Con esto ya podremos loguearnos tanto en terminal como de forma gráfica



6.Tecnologías adicionales

6.1. Instalación y configuración de DNS

DNS corresponde a las siglas en inglés de "Domain Name System", es decir, "Sistema de nombres de dominio". Este sistema es básicamente la agenda telefónica de la Web que organiza e identifica dominios.

He decidido implementar un DNS para que los usuarios que se conecten al almacenamiento en la nube desde un nombre en vez de que se conecten por IP, para ello he configurado el servicio DNS con “bind9”, he asignado al “server1” como servidor DNS primario y el “Server2” como servidor DNS secundario,he decidido hacerlo de esta forma para que se complemente bien con la alta disponibilidad de los servidores, ya que si solo hiciese un servidor DNS en un servidor y ese mismo servidor se cae dejaría a los usuarios sin servicio DNS.
Para ello he configurado una zona para el dominio “**raspivault.com**” y su correspondiente zona inversa, esta es la configuración:

```
# /etc/bind/named.conf.local
zone "raspivault.com" {
    type master;
    file "/etc/bind/zones/db.raspivault.com";
    allow-transfer { 192.168.30.3; };  // IP DNS Secondary
};
zone "30.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.30.168.192";
    allow-transfer { 192.168.30.3; };  // IP DNS Secondary
};
```

Para cada máquina de la red de los servidores he creado un nombre, cada nombre identifica a cada máquina/IP, a continuación mostraré una tabla con la máquina y su nombre correspondiente:

	SERVIDOR 1	SERVIDOR 2	SYNOLOGY	KEEPALIVED
IP	192.168.30.2 / 29	192.168.30.3 / 29	192.168.30.4 / 29	192.168.30.5 / 29
NOMBRE ASIGNADO	serv1.raspivault.com	serv2.raspivault.com	syn.raspivault.com	nube.raspivault.com

Así tengo mi configuración de cada máquina:

```
# /etc/bind/zones/db.raspivault.com
$TTL    604800
@       IN      SOA     serv1.raspivault.com. admin.raspivault.com. (
                              4         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      serv1.raspivault.com.
@       IN      NS      serv2.raspivault.com.
serv1.raspivault.com.    IN      A       192.168.30.2
serv2.raspivault.com.    IN      A       192.168.30.3
syn.raspivault.com.       IN      A       192.168.30.4
nube.raspivault.com.     IN      A       192.168.30.5
```

Y esta la configuración de la zona inversa:

```
@       IN      SOA     serv1.raspivault.com. admin.raspivault.com. (
                              4         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      serv1.raspivault.com.
@       IN      NS      serv2.raspivault.com.
2       IN      PTR     serv1.raspivault.com.
3       IN      PTR     serv2.raspivault.com.
4       IN      PTR     syn.raspivault.com.
5       IN      PTR     nube.raspivault.com.
```

Por último he configurado las redes en las que confío para que el servicio DNS permite consultar nombres, esta es su configuración:

```
# /etc/bind/named.conf.local
acl "trusted_networks" {
        192.168.30.0/29
       // Red Servidores       192.168.30.0    255.255.255.248
        192.168.1.0/30
       // Red Administración   192.168.1.0     255.255.255.252
        192.168.0.0/24
       // Red Clientes         192.168.0.0     255.255.255.0
};
```

```
#  /etc/bind/named.conf.local
options {
        directory "/var/cache/bind";

        recursion yes;
        allow-recursion { trusted_networks; };  // allows recursive
queries from "trusted"
        listen-on { 192.168.30.3; };            //  server1  private
IP address
        allow-transfer { none; };
        forwarders {                    // DNS Public google
                8.8.8.8;
                8.8.4.4;
        };
```

Ahora desde el segundo servidor debemos de crear una zona igual pero esta vez asignarle como esclavo e indicar la ip del servidor1, el maestro.

6.2. Instalación y configuración de Nagios

Nagios es una solución de monitorización de código abierto que permite a las organizaciones identificar y resolver problemas de infraestructura de TI antes de que afecten a los procesos críticos de negocio. Para los desarrolladores de software y administradores de sistemas que usan Debian 12, configurar Nagios puede ser un movimiento estratégico para asegurar la fiabilidad y disponibilidad de las aplicaciones y servicios.

Para poder instalar nagios primero debemos de instalar varias dependencias que tiene, estos son los paquetes que son necesarios instalar:

```
apt install -y autoconf gcc libc6 make wget unzip apache2 php
libapache2-mod-php7.4 libgd-dev
```

- Para configurar el nagios y para el propio uso del servicio crearemos un usuario y grupo llamado “nagios” el usuario y “nagcmd” el grupo.

Para instalar nagios debemos de descargar el nagios desde la propia página web oficial de nagios:

```
wget
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6
.tar.gz
tar xzf nagios-*.tar.gz
cd nagios-*
```

Una vez tenemos descargado la carpeta de nagios debemos de compilar el paquete por nuestra propia cuenta, en este paquete no lo compila el sistema, debemos de hacerlo manualmente:

```
sudo ./configure --with-nagios-group=nagios
--with-command-group=nagcmd
sudo make all
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
```

- La primera línea sirve para ejecutar el archivo “configure” con el usuario y grupo que hemos creado anteriormente.
- Las demás líneas estamos compilando el nagios.

Una vez tenemos creado el nagios vamos a crear la página web donde vamos a consultar los avisos y el estado de los servicios que estamos monitorizando y además vamos a añadir el usuario admin de nagios que utilizaremos más adelante para loguearnos en la página.

```
sudo make install-webconf
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
sudo a2enmod rewrite cgi #Activar módulo de apache
sudo systemctl restart apache2
```

Una vez tenemos creado el nagios vamos a añadir los plugins necesarios que necesitan nagios.

```
cd /tmp
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
tar xzf nagios-plugins-*.tar.gz
cd nagios-plugins-*
```

Compilamos los plugins:

```
sudo ./configure --with-nagios-user=nagios
--with-nagios-group=nagios --with-openssl
sudo make
sudo make install
```

Una vez tenemos el nagios y los plugins descargados y compilados ya podremos entrar dentro del nagios por medio de la página web, poniendo la ip del servidor y añadiendo /nagios al final de la url podremos entrar, y en el apartado servicios de la columna derecha podemos ver los estados de los servicios.

6.3. Configuración de LDAP con replicación

Para los usuarios que se conecten a través de la red de usuarios crearé un LDAP para que los usuarios se puedan autenticar, para ello haré el **LDAP tolerante a fallos**, como **OpenLDAP** no trae una opción por defecto de que dos o más servidores se comuniquen entre sí debemos de hacer una **replicación**, es decir estará un servidor como principal y otro replicando lo que haga el servidor principal.

Como sabrán las maquinas cliente a cual conectarse y en qué momento? En el archivo de configuración pondré ambos servidores como opción de ldap, y el mismo sistema intentará conectarse primero a uno y si no hay conexión se conectará al otro.

6.3.1 Configuración de OpenLDAP

Para configurar el LDAP que usaré para que los usuarios se logueen en el sistema usaré OpenLDAP, primero configuraré el servidor LDAP “Maestro”, esta será la configuración del servidor1:

```
root@Server01:/home/server1# ldapsearch -x -b "dc=raspivault,dc=com"
```

```
# raspivault.com
dn: dc=raspivault,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: natxorod
dc: raspivault

# Usuarios, raspivault.com
dn: ou=Usuarios,dc=raspivault,dc=com
objectClass: top
objectClass: organizationalUnit
ou: People
ou: Usuarios

# Usuarioscli, Usuarios, raspivault.com
dn: cn=Usuarioscli,ou=Usuarios,dc=raspivault,dc=com
objectClass: top
objectClass: posixGroup
gidNumber: 2000
cn: Usuarioscli

# usuariol, Usuarios, raspivault.com
dn: uid=usuariol,ou=Usuarios,dc=raspivault,dc=com
objectClass: top
objectClass: posixAccount
objectClass: inetOrgPerson
objectClass: person
cn: usuario
uid: usuariol
uidNumber: 2001
gidNumber: 2000
loginShell: /bin/bash
sn: uno
mail: usuariol@gmail.com
givenName: usuariol
homeDirectory: /mnt/ldap/usuariol
```

6.3.2 Configuración directorios usuarios LDAP

La configuración y el servicio de **LDAP** se guarda en los propios servidores, pero en cambio los **directorios y los archivos** de los usuarios que se conecten al ldap se guardarán en el **synology** a través de **NFS**.

Para ello debemos de crear una carpeta nueva en el synology y dar permisos de NFS a los dos servidores, llamaré a la carpeta ldap.
Así tengo los permisos del homeDirectory de cada usuario en el synology:

Para terminar vamos a indicar en "**/etc/fstab**" el montaje del directorio para que se realice automáticamente cada vez que arranquemos el sistema:

```
192.168.30.4:/ldap /mnt/ldap nfs defaults,noatime 0 0
```

6.3.3 Replicación de OpenLDAP

Para realizar **alta disponibilidad en LDAP** para que ambos servidores compartan y escriban la misma información es necesario que ambos se comuniquen y escriban la misma información, para ello usaremos un **usuario ldap** que **replique** la información en ambas direcciones, ya que no hay otra forma "sencilla" de realizar una alta disponibilidad entre dos servidores LDAP

El método de réplica que se va a utilizar en nuestro caso es el modo **MirrorMode**, este método nos permite tener una solución de alta disponibilidad para escrituras de directorio, mientras el maestro esté operativo las escrituras se pueden exceptuar con seguridad.
Además los nodos maestros se replican entre sí para que estén siempre actualizados los nodos y así pueden estar listos para hacer cargo de cualquier tipo de carga de información o consulta de la misma

```
dn: uid=mirrormode,dc=raspivault,dc=com
objectClass: top
objectClass: account
objectClass: simpleSecurityObject
uid: mirrormode
description: LDAP replication user
userPassword:: e1NTSEF9Tll2SE1iS0ZCKysyQU5DbFE1Sk43M21CUG1CK2d4cC8=
structuralObjectClass: account
entryUUID: e42f99f2-a715-103e-9d55-7576eddac251
creatorsName: cn=admin,dc=raspivault,dc=com
createTimestamp: 202405151448092
entryCSN: 20240515144809.553708Z#000000#000#000000
modifiersName: cn=admin,dc=raspivault,dc=com
modifyTimestamp: 202405151448092
```

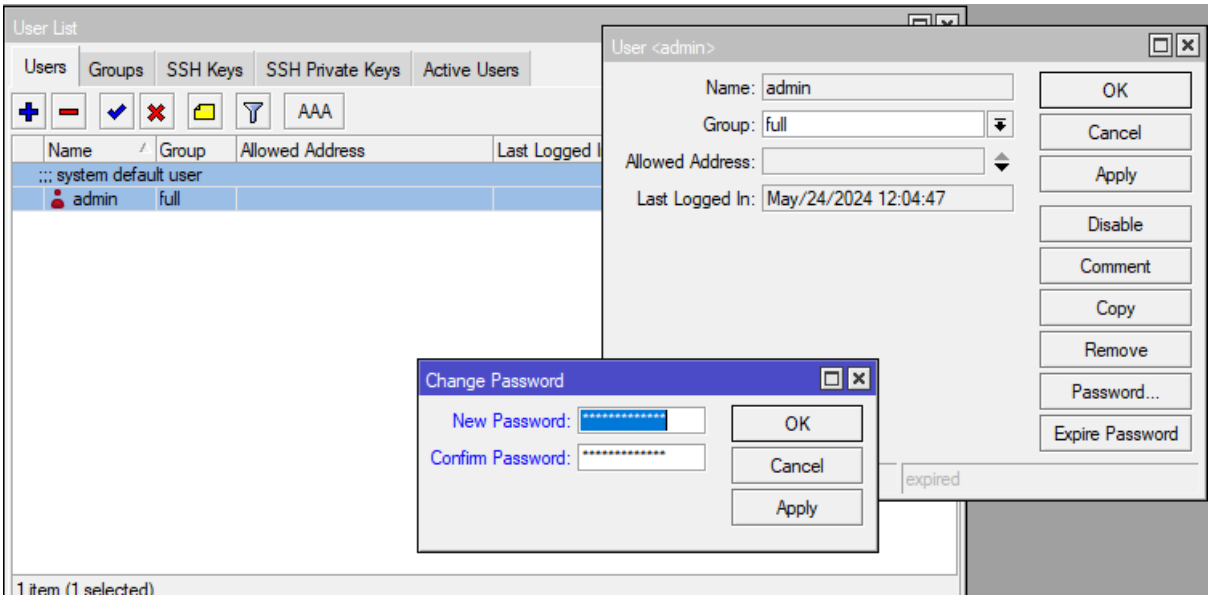
7.Administración

7.1. Gestión de usuarios y permisos

Para el uso de mikrotik he cambiado la contraseña del usuario admin para securizar el acceso al mikrotik y para la replicación del ldap he usado un usuario para administrar el replicado entre servidores.

7.1.1 Usuario mikrotik

Para cambiar la contraseña del admin debemos de ir a Systema > Users > seleccionar el usuario que queremos cambiar > Password… > Apply > OK



7.1.2 Usuario de replicación ldap

Antes de replicar el LDAP debemos de tener instalado y configurado el OpenLDAP en el servidor 1 y una vez lo tenemos debemos de crear un usuario en LDAP (Mirormode) en cada servidor y configurar varios archivos en cada uno.

Para el replicado de LDAP necesitamos crear un usuario (Fichero .ldif) y añadir más opciones, permisos y configuración.

Estos 6 archivos debemos de realizarlos en ambos servidores, después de escribir los 6 ficheros .ldif debemos de aplicar cada archivo de la siguiente manera:

```
ldapmodify -D "cn=admin,dc=raspivault,dc=com" -W -f [nombre de archivo]ldif
```

#Archivo1.ldif : Creación de usuario

```
dn: uid=mirrormode,dc=raspivault,dc=com
objectClass: top
objectClass: account
objectClass: simpleSecurityObject
description: LDAP replication user
userPassword: "contraseña LDAP"
```

#Archivo2.ldif : Permisos para el usuario creado anteriormente

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcAccess
olcAccess: to attrs=userPassword
by self =xw
by dn.exact="cn=admin,dc=raspivualt,dc=com" =xw
by dn.exact="uid=mirrormode,dc=raspivault,dc=com" read
by anonymous auth
by * none
olcAccess: to *
by anonymous auth
by self write
by dn.exact="uid=mirrormode,dc=raspivault,dc=com" read
by users read
by * none
```

#Archivo3.ldif : Cargar modulo syncprov

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
```

#Archivo4.ldif : Añadir el módulo syncprov al ldap

```
dn: olcOverlay=syncprov,olcDatabase={1}mdb,cn=config
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
```

#Archivo5.ldif : Añadir el número identificativo del server

```
dn: cn=config
changetype: modify
add: olcServerId
olcServerId: 1 #1 en caso del primer server,en el segundo un 2
```

#Archivo6.ldif : Habilitaremos la replicación en el servidor 1 indicando el otro equipo

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcSyncrepl
olcsyncrepl: rid=000
provider=ldap://serv1.raspivault.com # serv1 o serv2
type=refreshAndPersist
retry="5 5 300 +"
searchbase="dc=raspivault,dc=com"
attrs="*,+"
bindmethod=simple
binddn="uid=mirrormode,dc=raspivault,dc=com"
credentials="contraseña LDAP"
add: olcDbIndex
olcDbIndex: entryUUID eq
olcDbIndex: entryCSN eq
replace: olcMirrorMode
olcMirrorMode: TRUE
```

Como he dicho anteriormente debemos de poner el siguiente comando cambiando solamente el nombre del archivo para aplicar los archivos:

```
ldapmodify -D "cn=admin,dc=raspivault,dc=com" -W -f [archivo].ldif
```

7.2. Monitorización

Para monitorizar los servicios de ambas máquinas he utilizado dos nagios, cada uno monitorea su propio sistema y además los servicios del otro servidor, ya que si se cae el nagios de un servidor podríamos saber que servicios tiene levantado el otro servidor.

Así se mostraría si todo funcionara bien:

Host	Status	Last Check	Duration	Status Information
serv1.raspivault.com	UP	05-30-2024 09:48:29	0d 1h 2m 45s	PING OK - Packet loss = 0%, RTA = 0.09 ms
serv2.raspivault.com	UP	05-30-2024 09:48:29	0d 18h 54m 57s	PING OK - Packet loss = 0%, RTA = 0.27 ms

Host	Service	Status	Last Check	Duration	Attempt	Status Information
serv1.raspivault.com	Current Load	OK	05-30-2024 09:48:29	0d 1h 0m 23s	1/3	OK - load average: 0.02, 0.09, 0.09
	HTTP SERVER 1	OK	05-30-2024 09:50:29	0d 0h 0m 30s	1/3	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time
	Root Partition	OK	05-30-2024 09:48:29	0d 0h 56m 38s	1/3	DISK OK - free space: / 106776 MiB (95.22% inode=98%):
	SSH	OK	05-30-2024 09:48:29	0d 0h 59m 46s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u2 (protocol 2.0)
	Total Processes	OK	05-30-2024 09:48:29	0d 0h 58m 31s	1/3	PROCS OK: 78 processes with STATE = RSZDT
serv2.raspivault.com	Current Load	OK	05-30-2024 09:48:29	0d 0h 49m 54s	1/3	OK - load average: 0.02, 0.09, 0.09
	HTTP SERVER 2	OK	05-30-2024 09:48:29	0d 0h 40m 22s	1/3	HTTP OK: HTTP/1.1 200 OK - 944 bytes in 0.003 second response time
	PING	OK	05-30-2024 09:48:29	0d 1h 6m 12s	1/3	PING OK - Packet loss = 0%, RTA = 0.28 ms
	Root Partition	OK	05-30-2024 09:48:29	0d 0h 57m 17s	1/3	DISK OK - free space: / 106776 MiB (95.22% inode=98%):
	SSH	OK	05-30-2024 09:48:29	0d 0h 53m 52s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u2 (protocol 2.0)

Aquí he apagado el servidor 1 y el keepalived me ha redirigido al servidor 2, podemos comprobar los CRITICAL que tenemos:

Host	Status	Last Check	Duration	Status Information
serv1.raspivault.com	DOWN	05-30-2024 09:23:26	0d 0h 10m 53s	CRITICAL - Host Unreachable (192.168.30.2)
serv2.raspivault.com	UP	05-30-2024 09:21:56	0d 0h 13m 33s	PING OK - Packet loss = 0%, RTA = 0.08 ms

Host	Service	Status	Last Check	Duration	Attempt	Status Information
serv1.raspivault.com	Current Load	OK	05-30-2024 09:21:26	0d 0h 11m 32s	1/4	OK - load average: 0.03, 0.03, 0.00
	HTTP SERVER 1	CRITICAL	05-30-2024 09:20:36	0d 0h 7m 22s	4/4	connect to address 192.168.30.2 and port 1001: No route to host
	PING	CRITICAL	05-30-2024 09:18:50	0d 0h 4m 8s	4/4	CRITICAL - Host Unreachable (192.168.30.2)
	Root Partition	OK	05-30-2024 09:18:26	0d 0h 9m 32s	1/4	DISK OK - free space: / 108564 MiB (96.81% inode=99%):
	SSH	CRITICAL	05-30-2024 09:19:26	0d 0h 3m 32s	4/4	connect to address 192.168.30.2 and port 22: No route to host
	Total Processes	OK	05-30-2024 09:20:26	0d 0h 7m 32s	1/4	PROCS OK: 70 processes with STATE = RSZDT
serv2.raspivault.com	Current Load	OK	05-30-2024 09:21:56	0d 0h 11m 2s	1/4	OK - load average: 0.07, 0.04, 0.00
	HTTP SERVER 2	OK	05-30-2024 09:17:56	0d 0h 10m 2s	1/4	HTTP OK: HTTP/1.1 302 Found - 1567 bytes in 0.825 second response time
	Root Partition	OK	05-30-2024 09:19:56	0d 0h 8m 2s	1/4	DISK OK - free space: / 108564 MiB (96.81% inode=99%):
	SSH	OK	05-30-2024 09:20:26	0d 0h 7m 32s	1/4	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u2 (protocol 2.0)

7.3. Implementación de backups

En mi caso tengo los archivos de configuración de nextcloud, sus bases de datos y los HOMEs de los usuarios ldap dentro del synology, partiendo de esa base haré un backup con rsync desde otra maquina al synology entero, creando una regla en crontab que automatice este proceso semanalmente.

Teniendo instalado rsync en la máquina donde guardaremos los backups vamos a realizar el siguiente comando rsync en mi caso utilizo una clave ssh para poder acceder más tarde en el script sin poner contraseña.

```
rsync -avze "ssh -i /root/.ssh/rsync_clave"
"root@192.168.30.4:/volume1/" "/backups/backup"
```

- De esta manera podemos hacer una copia entera del synology manualmente, pero en este caso queremos automatizarlo para que se haga solo sin que nadie intervenga.

Este es el script que voy a ejecutar en el crontab para que se realice una copia una vez por semana:
SCRIPT:

```
#!/bin/bash
# Variables
BACKUP_DIRECTORIO="/backups"
ANTIGUO_BACKUP_DIRECTORIO="${BACKUP_DIRECTORIO}/copiasAntiguas"
LOG_DIR="${BACKUP_DIRECTORIO}/log"
LOG_FILE="${LOG_DIR}/rsync_log_$(date +'%Y%m%d_%H%M%S').log"
ORIGEN="root@192.168.30.4:/volume1/"
DEST="${BACKUP_DIRECTORIO}/backup"
TAR_FILE="${ANTIGUO_BACKUP_DIRECTORIO}/backup_$(date
+'%Y%m%d_%H%M%S').tar.gz"

# Comprobar si existe una copia anterior para comprimir
if [ -d "${BACKUP_DIRECTORIO}/backup" ]; then
    # Comprimir la anterior copia
    tar -czf "${TAR_FILE}" -C "${DEST}" backup
```

```
# Comprobar si se ha comprimido correctamente la copia de seguridad
    if [ $? -eq 0 ]; then
        echo "Archivo comprimido: ${TAR_FILE}" >> "${LOG_FILE}"
    else
        echo "Error al comprimir" >> "${LOG_FILE}"
        exit 1
    fi
else
    echo “No se ha encontrado archivo para comprimir, omitiendo..."
>> "${LOG_FILE}"
fi
# Comando que realiza la copia de seguridad con rsync
rsync -avze "ssh -i /root/.ssh/rsync_clave" "${ORIGEN}" "${DEST}"
&>> "${LOG_FILE}"

# Comprobar que la copia de seguridad se ha completado correctamente
if [ $? -eq 0 ]; then
    echo "Copia realizada" >> "${LOG_FILE}"
else
    echo "Copia fallida" >> "${LOG_FILE}"
    exit 1
fi
```

Una vez tenemos el script le añadimos permisos para que se pueda ejecutar y añadimos la línea dentro del crontab para que se pueda crear los backups semanalmente:

```
30 23 * * 5 /bin/bash /root/scriptBackups.sh
```

En este comando indicamos que todos los viernes de todos los meses y de todos los años se ejecute el script que hemos indicado anteriormente a las 23:30 de la noche.

7.4 Script inicio de servidores

Cuando tengo apagados todos los servidores y enciendo todos las raspberrys se encienden mucho antes que el synology, entonces cuando se inicia el synology 5 minutos después de las raspberrys estas no tienen los montajes de las carpetas compartidas por NFS.

He solucionado este problema, ahora puedo encender los servidores en el orden que quiera, lo que he hecho es un script sencillo que compruebe si el synology está encendido (mediante un ping), si lo está que haga un “mount -a” y si no está encendido que vuelva a comprobarlo dentro de 30 segundos.

Script:

```
#!/bin/bash
ping() {#Función para comprobar si hay conexión con la máquina
    ping -c 1 192.168.30.4 > /dev/null
    return $?
}
mount() { # Función para montar los sistemas de archivos
    mount -a
}
while true; do # Bucle Para comprobar estado de la máquina
    # Comprobamos si hay ping con la máquina
    if ping; then
        # Montamos los sistemas de archivos
       sleep 20      #Esperar que se levanten las carpetas
        mount
        break
    else
        # Esperamos 30 segundos antes de volver a intentar
        sleep 30
    fi
done &> /var/log/script_log_nfs.log &
```

Línea dentro del crontab (crontab -e con root):

```
@reboot /root/nfs.sh
```

8.Conclusiones

Antes de empezar el proyecto decidí que utilizaré unos servicios y unas tecnologías, únicamente investigué que tecnologías podría usar, pero cuando empecé la parte práctica del proyecto me di cuenta que las cosas son bastante diferentes de como funciona un servicio o tecnología mirando por internet que realizar una infraestructura y combinar varios servicios y que todos funcionen a la perfección sin que den conflictos entre sí, así que las tecnologías que he usado han sido los servicios que más se han adaptado a mi estilo, los más “sencillos” que me han parecido y con los que más familiarizado estaba.

8.1 Problemas encontrados

El primer problema que encontré fue proxmox, quería instalar los servicios dentro de un proxmox por raspberry, pero cuando empecé a investigar vi que proxmox no soportaba la arquitectura arm64 que es la arquitectura que está hecho la raspberry, encontré un proyecto de github que adaptaba a arm64, pero tampoco me llegó a servir ya que necesitaba más recursos ya que no existen plantillas con sistema arm.

Otro error que me encontré fue el ldap del synology, el synology trae un servicio de ldap, lo instalé y configuré pero no me llegaba a conectar con el usuario ldap fuera del propio synology.

También me tope con un error en el https en nextcloud, ya que al intentar crear el certificado con certbot me daba continuamente problemas, y como me estaba tomando demasiado tiempo y no me quedaba mucho decidí abandonar la opción de implementar https.

8.2 Mejoras

Las mejoras que yo haría serían las siguientes:
Lo primero que mejoraría sería el firewall del mikrotik, no quiero decir que sea insuficiente, pero dedicando mucho más tiempo al firewall se podría afinar mucho más y poder restringir mucho más los accesos indebidos.

A parte del firewall también mejoraría la parte del nagios, pondría chequeos más específicos para mi proyecto para los servidores, y añadiría que también pudiera monitorear a los clientes.

Si este proyecto fuera destinado a un mayor número de clientes y un mayor tráfico también mejoraría los servidores, es decir cambiaría las raspberrys por servidores dedicados a la función de mantener servicios las 24 horas del dia.

9.Bibliografía y Webgrafía

- Página web oficial de Raspberry OS donde he descargado el sistema operativo
  https://www.raspberrypi.com/software/

- Instalación y configuración de Keepalived y haproxy
  https://sysadm.es/haproxy-balanceador-activo-pasivo/

- Instalación y Configuración de Nextcloud en debian 12
  https://howtoforge.es/como-instalar-nextcloud-en-debian-12/

- Configuración de carpeta compartida NFS en el cliente
  https://somebooks.es/nfs-parte-5-acceder-a-la-carpeta-compartida-desde-un-cliente-ubuntu-20-04-lts/

- Configuración de dhcp para Mikrotik
  https://jhonatanlamina.com/configuracion-basica-de-un-router-mikrotik/

- Configuración de Firewall del Mikrotik
  http://foroisp.com/threads/1643-Firewall-B%C3%A1sico-Para-Equipos-Mikrotik

- Instalar y configurar DNS primario y secundario
  https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-private-network-dns-server-on-ubuntu-18-04-es

- Instalar y compilar Nagios
  https://reintech.io/blog/installing-configuring-nagios-monitoring-debian

- Tecnología descartada Pimox7
  https://github.com/pimox/pimox7

- Instalación y configuración del LDAP Principal
  https://juanjoselo.wordpress.com/2017/10/29/instalacion-y-configuracion-de-ldap-en-debian-stretch-con-directorio-ldap-basico/

- Configuración de LDAP secundario utilizando MirrorMode
  https://juanjoselo.wordpress.com/2017/11/30/configuracion-de-openldap-para-ldap-secundario/

- Configuración de LDAP cliente
  Inicio de sesión solamente en terminal
  https://somebooks.es/ldap-parte-6-configurar-un-cliente-ubuntu-para-autenticarse-en-el-servidor-openldap/

Inicio de sesión con interfaz gráfica
https://somebooks.es/ldap-parte-7-iniciar-sesion-grafica-en-el-equipo-cliente-con-un-usuario-ldap/

Añadir homedirs al sistema operativo del cliente, sirve para poder tener el home directory del usuario fuera de "/home"
https://forum.snapcraft.io/t/home-directories-outside-of-home/19224

- Instalación y configuración de Rsync
https://www.proxadmin.es/blog/rsync-10-ejemplos-practicos-de-comandos-rsync/

HERRAMIENTAS USADAS:

- Imagen para la portada:
https://jartigag.blog/nextcloudpi

- Herramienta web utilizada para el diagrama de gantt
https://canva.com

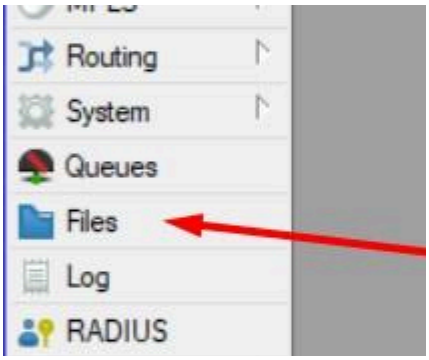
- Herramienta web utilizada para el diseño de topología de red
https://draw.io

10. Anexos

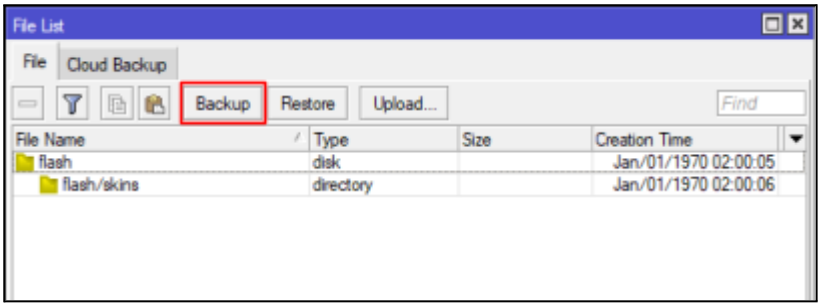
10.1 Backup mikrotik:

Para guardar configuración y el estado del mikrotik por si se corrompe algún archivo o por si se borran por algún motivo las configuraciones del mikrotik debemos de hacer un backup del mikrotik, debemos de hacer lo siguiente:

1.Vamos a “files”

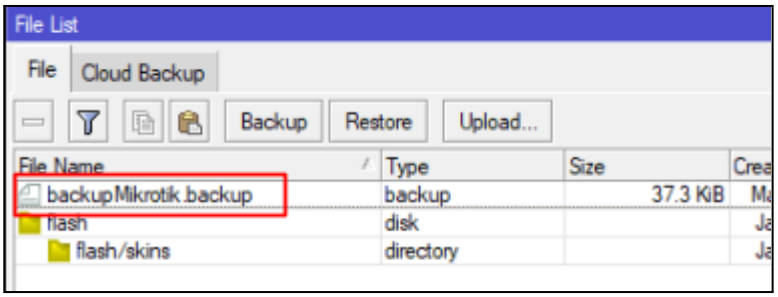


2.Le damos a “Backup”



3.Ponemos Nombre y confirmamos

4. Podemos ver el backup en los “files” mikrotik



5.Por último lo podemos guardar en cualquier otro sitio, en este caso he seleccionado y arrastrado al escritorio.