Implementació d'un núvol privat virtual

Memòria treball final de grau

Estudiant: **Albert Oliveras Alvarez**
Consultor: **Joaquin Lopez Sanchez-Montañes**

Agraïments:

Com no podria ser d'un altre forma, vull agrair a la meva família, la meva dona i els meus dos fills la paciència i la tolerància per tot el temps que no els he pogut dedicar durant aquests darrers anys.

Aquest projecte vol presentar una alternativa real, lliure, segura i fiable als núvols proporcionats per grans empreses com Google, Microsoft o Amazon, dirigida a particulars i/o petits grups de treball perquè puguin mantenir el control de les seves dades. S'implementarà una infraestructura virtual privada basada en contenidors utilitzant exclusivament en programari lliure que ens proporcionarà diferents serveis accessibles de forma segura mitjançant internet des de qualsevol ubicació. S'oferiran serveis d'emmagatzematge i sincronització de dades i d'administració remota de la mateixa infraestructura.

Núvol Virtual Contenidors Internet Infraestructura Programari Lliure emmagatzematge Sincronització

This project wants to present a real, free, safe and reliable alternative to the clouds provided by large companies such as Google, Microsoft or Amazon, aimed at individuals and / or small work groups so they can keep track of their data. A private virtual infrastructure based on containers will be implemented using exclusively open source software that will provide us with different services accessible safely over the Internet from any location. Data storage and synchronization services and remote administration of the same infrastructure will be offered.

Cloud Virtual Container Internet Infrastructure Open Source Data Storage Sync

Índex

1 DESCRIPCIÓ DEL TFG....6
2 INTRODUCCIÓ AL TFG....6
2.1 Justificació....7
2.2 Objectiu....7
2.3 Planificació i metodologia....7
2.4 Programari lliure utilitzat....8
3 PREPARACIÓ DE LA INFRAESTRUCTURA....11
3.1 Instal·lació SO servidor i programari base....11
3.2 Configuració i preparació de l'emmagatzematge....11
3.3 Configuració i preparació d'usuaris....12
3.4 Instal·lació del servei de contenidors virtuals....13
3.5 Configuració infraestructura de comunicacions....14
3.5.1 Infraestructura externa....14
3.5.1.1 Servei de registre DDNS (dinamic DNS)....16
3.5.1.2 Autoritat certificadora....17
3.5.1.3 Configuració de l'encaminador local....18
3.5.2 Infraestructura interna....19
3.5.2.1 Xarxes virtuals....21
3.5.2.2 Mapeig de ports (publicació de ports)....22
3.6 Instal·lació i configuració del contenidor redirector....23
3.6.1 Instal·lació traefik....23
3.6.2 Configuració traefik....24
3.6.3 Integració de traefik amb Docker....25
3.7 Proves de connexió i redirecció....26
4 CONFIGURACIÓ DE SERVEIS....28
4.1 Sobre la creació de contenidors....28
4.2 Configuració de serveis de suport i administració....29
4.2.1 Instal·lació i configuració del contenidor de bases de dades: MariaDB....29
4.2.2 Instal·lació i configuració del contenidor d'administració de bases de dades: phpMyadmin....29
4.2.3 Proves de connexió i redirecció....30
4.3 Configuració del servei d'emmagatzematge, sincronització i compartició. 32
4.3.1 Instal·lació i configuració del contenidor: NextCloud....32
4.3.2 Proves de connexió i redirecció....33
4.4 Configuració d'altres serveis....35
4.4.1 Instal·lació, configuració i proves del contenidor de descàrregues: Transmission....35
4.4.2 Instal·lació, configuració i proves del contenidor de sessions remotes: x2go....36
4.5 Configuració del servei d'administració de contenidors....38
4.5.1 Instal·lació i configuració del contenidor: Portainer....38
4.5.2 Proves de connexió i redirecció....38

5 TEST DE SEGURETAT....42
5.1 Nextcloud Security Scan....42
5.2 Qualys SSL Labs....43
6 MANTENIMENT DE LA INFRAESTRUCTURA....48
6.1 Actualització de la infraestructura....48
6.1.1 Actualització Sistema Operatiu....48
6.1.2 Actualització dels contenidors....48
6.1.3 Manteniment de les aplicacions i troubleshoting....49
6.2 Còpies de seguretat i restauració....49
6.3 Aturada i arrancada de serveis (zero)....50
6.4 Configuració del clients....51
6.4.1 Accés web....51
6.4.2 Client d'escriptori....51
6.4.3 Client per dispositius mòbils....53
6.4.4 Accés WEBDAV....55
7 MILLORES....55
7.1 Alta disponibilitat....55
7.2 Docker-compose....55
7.3 Altres serveis....55
8 COSTOS D'IMPLEMENTACIÓ....56
8.1 Implementació....56
8.2 Planificació i documentació....56
8.3 Maquinari....57
9 BIBLIOGRAFIA....58

1 DESCRIPCIÓ DEL TFG

Implementació amb programari lliure d'una infraestructura basada en contenidors virtuals per proporcionar diferents serveis orientats a particulars o petits grups de treball amb connexions segures.

Sobre aquesta infraestructura es crearan contenidors per oferir els següents serveis:

- Servei d'emmagatzematge i sincronització de fitxers
- Servei de descàrregues via torrent
- Servei de sessions remotes

Serà necessari també crear contenidors de suport i d'administració de la infraestructura virtual

- Servei d'administració de la infraestructura de contenidors
- Servei de base de dades relacional
- Servei d'administració de bases de dades
- Servei de redirecció de connexions http/https i connexions segures

Es faran servir certificats per l'encriptació de les comunicacions que es realitzin a través d'internet mitjançant protocols segurs de comunicació.

Els requeriments per la realització del projecte estan a l'abast de tothom, es farà servir una connexió de dades per fibra (100Mb simètrics), un encaminador local i un petit servidor HPE microserver Proliant G8 (Celeron G1610T amb 2 cores i 4Gb de RAM) que disposa de quatre discos per tal d'obtenir una mínima redundància de les dades.

2 INTRODUCCIÓ AL TFG

Fa relativament poc temps, uns anys després de l'aparició d'Internet, va sorgir el servei de correu via web: gmail, yahoo i d'altres empreses van començar tímidament a oferir aquests primers serveis on les dades (els correus) es desaven distribuïdes en algun lloc indeterminat d'internet, el núvol. Actualment s'ofereixen una gran quantitat de serveis des del núvol: emmagatzematge de dades, xats, plataformes ofimàtiques, entorns col·laboratius, xarxes socials, etc. Molts d'aquests serveis són fins i tot gratuïts, amb certes limitacions, però ofereixen alta disponibilitat, còpies de seguretat, accés des de qualsevol lloc i dispositiu, certa protecció contra els atacs dels hackers i un bon grapat d'avantatges difícils de rebutjar, fins i tot una gran part d'empreses estan utilitzant aquests serveis com a estratègia per reduir costos d'infraestructura.

2.1 Justificació

Tots aquests serveis tan meravellosos descrits anteriorment no són gratuïts, tenen un cost que pot semblar ínfim per alguns i totalment intolerable per altres: la pèrdua de privacitat i el control de les dades. Tots som conscients que amb l'excusa d'oferir-nos serveis personalitzats a mida, aquestes empreses tenen accés a les nostres dades, correus, imatges, ubicacions, etc. I l'abast d'aquest accés és totalment incert.

GNU Linux i el programari lliure ens permeten implementar un núvol privat virtual i ens poden proporcionar una alternativa realment gratuïta i confiable a la utilització de serveis de núvols externs com google, dropbox, onedrive, etc., mantenint el control i seguretat de les nostres dades.

2.2 Objectiu

L'objectiu principal d'aquest TFG és presentar una possible implementació domèstica per particulars o petits grups de treball d'un núvol virtual i uns serveis bàsics d'emmagatzematge i sincronització amb l'ús exclusiu de programari lliure. Aquesta infraestructura ha de permetre una connexió fiable i segura a través d'internet i una administració remota d'aquesta.

2.3 Planificació i metodologia

El projecte es divideix en tres fases principals que no es corresponen temporalment amb les entregues de les PACs, sovint les entregues queden a cavall entre dues d'aquestes fases però aquest fet no presenta cap problema per fer el seguiment del mateix.

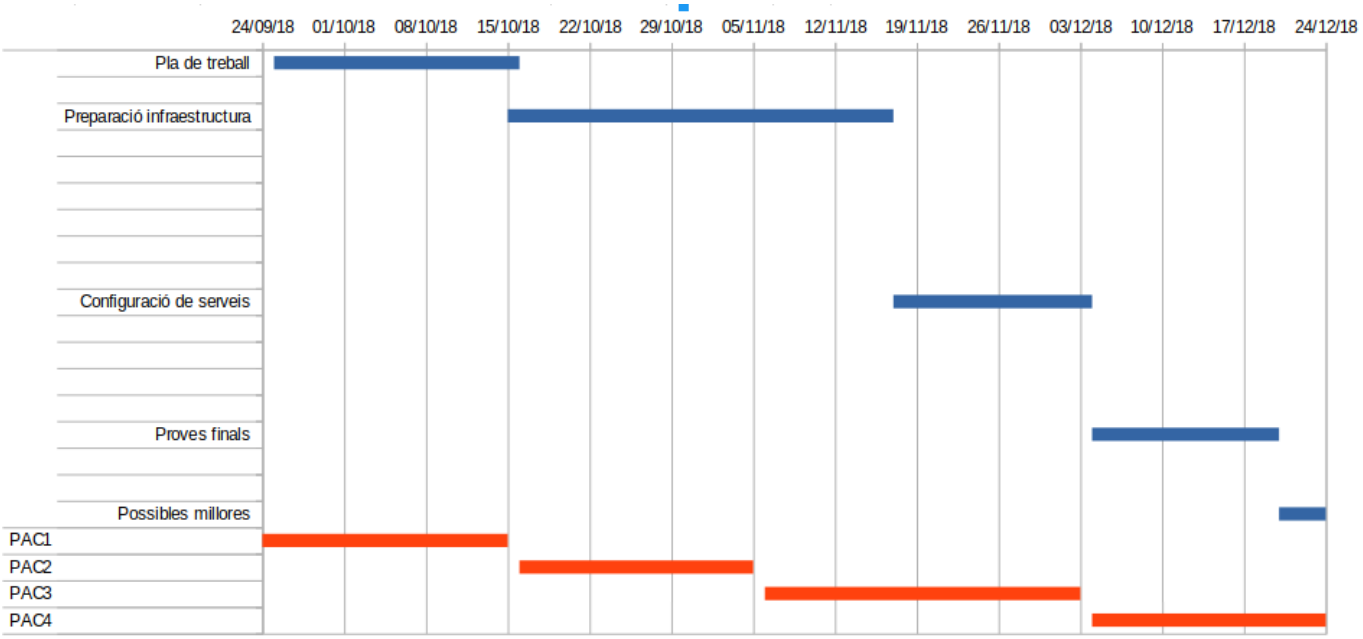
Aquestes fases són les següents:

- Preparació de la infraestructura: Consisteix en la instal·lació del sistema operatiu del servidor, configuració de l'emmagatzematge, instal·lació dels serveis de contenidors virtuals, configuració del contenidor redirector, configuració de les xarxes virtuals i configuració de l'encaminador local. Aquesta primera fase consta d'una fita per comprovar que les tasques anteriors han estat complimentades de forma satisfactòria, unes proves bàsiques de connexió i redirecció.
- Configuració de serveis: Instal·lació i configuració dels diferents contenidors que proporcionaran els serveis, aquests serveis es poden dividir en serveis de suport i administració, i el serveis d'usuari propiament dits. Cada servei instal·lat constarà d'una petita fita de comprovació d'aquest.
- Proves finals: Aturada i arrancada total (zero), proves finals dels diferents serveis i per finalitzar un test de seguretat per confirmar que les connexions són segures.
- Manteniment de la infraestructura

- Costos d'implementació
- Consideracions: Reflexió sobre possibles millores de la infraestructura des de diferents perspectives.

Full de càlcul de planificació i seguiment de les fases i tasques a realitzar:

PAC	Data d'inici	Data final	Completat	Fase	Completat	Inici	Final	Tasques	Finalitzada
PAC1	24/09/18	15/10/18	100,00 %	Pla de treball	100,00 %	24/09/18	15/10/18	Redacció estructurada idea del projecte i planificació de tasques	100,00 %
PAC2	16/10/18	05/11/18	0,00 %	Preparació infraestructura	0,00 %	16/10/18	16/11/18	Instal·lació SO servidor i programari base	0,00 %
								Configuració i preparació de l'emmagatzematge	0,00 %
								Instal·lació del servei de containers virtuals	0,00 %
								Configuració de les xarxes virtuals	0,00 %
								Instal·lació i configuració del container redirector	0,00 %
PAC3	06/11/18	03/12/18						Configuració del encaminador local i registre/publicació de noms del domini	0,00 %
								Proves de connexió i redirecció	0,00 %
				Configuració de serveis	0,00 %	17/11/18	03/12/18	Configuració de serveis de suport	0,00 %
								Configuració del servei d'emmagatzematge, sincronització i compartició	0,00 %
								Configuració del servei de descàrregues	0,00 %
								Configuració del servei de sessions remotes	0,00 %
								Configuració del servei d'administració de containers	0,00 %
PAC4	04/12/18	24/12/18		Proves finals	0,00 %	04/12/18	20/12/18	Aturada i arranc de serveis	0,00 %
								Prova final de serveis	0,00 %
								Tests de seguretat	0,00 %
				Possibles millores	0,00 %	20/12/18	24/12/18	Escalabilitat i alta disponibilitat	0,00 %



2.4 Programari lliure utilitzat

La utilització de programari lliure com a alternativa al programari propietari de pagament és una decisió basada en dos punts bàsics un, òbviament, és intentar realitzar el projecte amb cost zero més enllà del maquinari i el segon i més important, si fem servir programari propietari de codi tancat, qui ens pot assegurar que les nostres dades no continuen en perill? En tot cas el programari propietari també té les seves avantatges, no podem oblidar que el suport gratuït de la comunitat no entén de SLA's ni d'urgències si les coses van mal dades i apareixen problemes.

El ventall d'opcions dins del programari lliure és molt gran i a continuació farem una breu enumeració del programari lliure seleccionat per la implementació del projecte:

Sistema Operatiu: *CentOS 7.5* (SO linux lliure derivat de RedHat estable i confiable)

contenidors: *Docker Community* (modern contenidor engine stàndard per tota mena de serveis)

Emmagatzematge, sincronització i compartició: *Nextcloud* (clon lliure de Owncloud, permet l'ús de connectors per altres funcions com xat, col·laboració, correu, etc.)

Descàrregues torrent: *Transmission* (conegut programari de gestió de descàrregues que permet l'administració via web)

Sessions remotes: *x2go* (programari que permet la connexió remota a sessions gràfiques sense necessitat d'un gran ample de banda)

Administració de contenidors: *Portainer* (programari lliure d'administració de contenidors via web)

Reverse proxy: *Traefik* (gestor de redireccions http i https amb funcions de load balancer que s'integra amb tot tipus de tecnologies de clúster i gestió de certificats, fins i tot proporciona métriques de connexió)

Base de dades relacional: *MariaDB* (derivada lliure de MySQL, super-utilitzada bd relacional com a backend de tot tipus de serveis web, ràpida i consistent)

Gestió de bd's: *phpMyAdmin* (gestió via web de bd's com MariaDbB)

3 PREPARACIÓ DE LA INFRAESTRUCTURA

El maquinari que del es disposa pel projecte és un HPE microserver Proliant G8 (Celeron G1610T amb 2 cores i 4Gb de RAM). Aquest servidor té bahies externes per fins a quatre discos. S'han instal·lat quatre discos, dos de 500Mb i dos de 2Tb, la idea és poder tenir una mínima redundància de les dades mitjançant un raid 1 entre els discos de 2Tb.

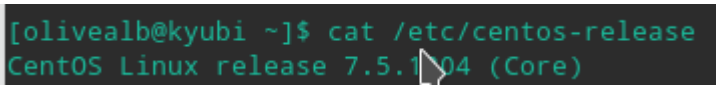
3.1 Instal·lació SO servidor i programari base

Instalació per defecte sense entorn gràfic de la darrera versió del SO Centos 7u5 des de l'imatge descarregada de:

http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-DVD-1804.iso

Atès que la controladora Raid instal·lada en el servidor (HP B120i) no disposa de suport per part del fabricant a versions de Centos/RHEL superiors a la 7u3, es considera que la millor opció és configurar els RAIDs des de el SO nativament amb mdadm ja que d'aquesta forma el suport el proporciona el mateix SO. Segurament aquesta decisió comportarà una petita pèrdua de rendiment al configurar els raids per software en lloc de per hardware però per contra guanyarem en flexibilitat atès que mdadm permet fer raids amb particions i no cal utilitzar tot el disk en un mateix raid.

S'utilitzen els dos discos de 500 Mb en raid 1 per la instal·lació del SO, idioma anglès per defecte amb disposició de teclat en castellà amb la variant catalana i el fus horari corresponent a la nostre zona.



3.2 Configuració i preparació de l'emmagatzematge

Es disposa de dos discos de 2Tb pel projecte atés que els altres dos s'han utilitzat per a sistema operatiu. En aquests dos discos es creen dues particions de 1,5Tb per configurar un raid 1, ja que amb el mirall tot i perdre el 50% de l'espai s'obté una redundància de les dades i un dels criteris del projecte és la fiabilitat (en cas de mal funcionament d'un dels discos, es pot continuar operant amb normalitat fins a reposar-lo). Amb la resta de disc crearem dues particions més de 500Gb sense redundància per emmagatzemar dades no importants pel projecte que siguin temporals.

A continuació es configura el dispositiu resultant del raid 1 (md125) com un VG (volume group) de LVM2 anomenat vg_dataraid. Utilitzar LVM és molt aconsellable atès que, entre altres avantatges, permet redimensionar els volums lógics (LV) creats de forma cómoda i segura sense utilitzar altres eines. També permetria afegir discos al VG i diposar de més espai pels lv's, per exemple, en cas necessari es podria crear un altre raid amb les particions de 500Mb esmentades anteriorment i afergir-lo al vg_dataraid i disposariem de 500Mb més d'espai pels lv's.

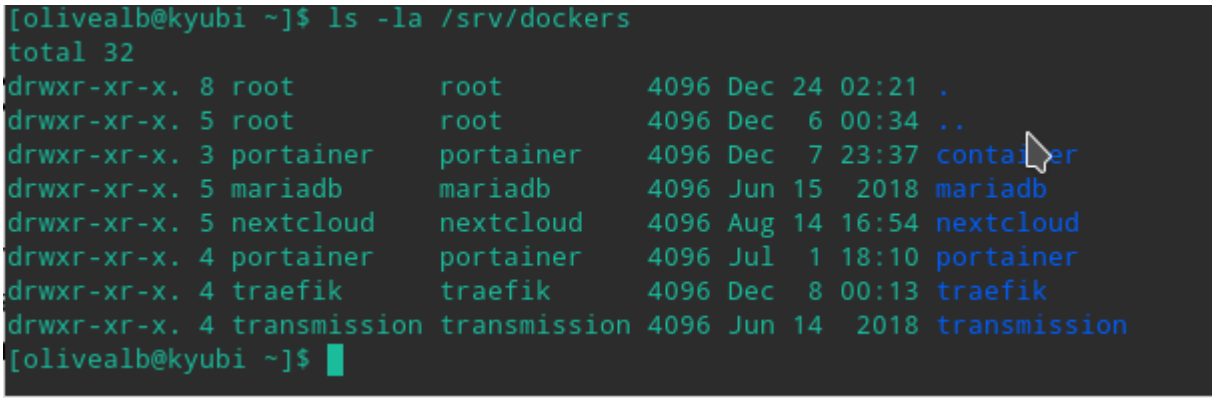
Seguidament es crearan logical volumes (lv's) pels serveis, a priori no sabem quan espai necessitarem per cadascun, però no té gens d'importància perquè podem canviar la mida en cas necessari en qualsevol moment com explicava anteriorment. D'aquesta forma l'aspecte final de la distribució dels discos és la següent:

```
[olivealb@kyubi ~]$ lsblk -f
NAME                              FSTYPE            LABEL                     UUID                                   MOUNTPOINT
sda
├─sda1                            linux_raid_member localhost.localdomain:boot 2acfe2d0-1d1a-ae8e-22f9-0f21d9e9be16
│ └─md126                         ext4                                       e3a16989-877c-4778-9dc7-1976f34a832d   /boot
└─sda2                            linux_raid_member localhost.localdomain:pv00 ce623cd5-2c8e-37fb-06e0-b35645501d98
  └─md127                         LVM2_member                                nZS8Mp-YOhc-JKlC-2ynp-nvpj-Z47L-SaNk9y
    ├─cl-root                     ext4                                       803d8d7a-f9fd-4bff-ae51-46b275bab590   /
    └─cl-swap                     swap                                       289eb978-4df3-4152-abfc-b28d248bf996   [SWAP]
sdb
├─sdb1                            linux_raid_member localhost.localdomain:boot 2acfe2d0-1d1a-ae8e-22f9-0f21d9e9be16
│ └─md126                         ext4                                       e3a16989-877c-4778-9dc7-1976f34a832d   /boot
└─sdb2                            linux_raid_member localhost.localdomain:pv00 ce623cd5-2c8e-37fb-06e0-b35645501d98
  └─md127                         LVM2_member                                nZS8Mp-YOhc-JKlC-2ynp-nvpj-Z47L-SaNk9y
    ├─cl-root                     ext4                                       803d8d7a-f9fd-4bff-ae51-46b275bab590   /
    └─cl-swap                     swap                                       289eb978-4df3-4152-abfc-b28d248bf996   [SWAP]
sdc
├─sdc1                            linux_raid_member localhost.localdomain:100  5b7cdcc2-5d5c-f769-454f-9e3911e80067
│ └─md125                         LVM2_member                                DPHRHE-Hj5I-uAbt-T3Bw-F2WW-VdsJ-rMqE1F
│   ├─vg_dataraid-lv_nextcloud    ext4                                       9c45f602-001b-4b59-87fe-ef2f1aa6e9c4   /srv/dockers/nextcloud
│   ├─vg_dataraid-lv_transmission ext4                                       0869b6ba-efb2-448e-9f11-7b3f00bd8375   /srv/dockers/transmission
│   ├─vg_dataraid-lv_mariadb      ext4                                       0580c0a8-6a62-4613-9801-db2a907e32b0   /srv/dockers/mariadb
│   ├─vg_dataraid-lv_traefik      ext4                                       d557f43f-aaaf-4779-8fe3-223f24ae6a45   /srv/dockers/traefik
│   ├─vg_dataraid-lv_portainer    ext4                                       b4f992ac-55b1-4d10-a295-0ec2973a2697   /srv/dockers/portainer
│   └─vg_dataraid-lv_documents    ext4                                       71aad055-c80c-400a-bd0c-90000b1185dd   /srv/documents
└─sdc2                            LVM2_member                                5gdTkJ-cONu-Bhhh-47A6-1IJ8-1y26-dFUp6J
  └─vg_data-lv_shared             xfs                                        df81d0a0-79a1-47b6-849e-a7a9472bf175   /srv/shared
sdd
├─sdd1                            linux_raid_member localhost.localdomain:100  5b7cdcc2-5d5c-f769-454f-9e3911e80067
│ └─md125                         LVM2_member                                DPHRHE-Hj5I-uAbt-T3Bw-F2WW-VdsJ-rMqE1F
│   ├─vg_dataraid-lv_nextcloud    ext4                                       9c45f602-001b-4b59-87fe-ef2f1aa6e9c4   /srv/dockers/nextcloud
│   ├─vg_dataraid-lv_transmission ext4                                       0869b6ba-efb2-448e-9f11-7b3f00bd8375   /srv/dockers/transmission
│   ├─vg_dataraid-lv_mariadb      ext4                                       0580c0a8-6a62-4613-9801-db2a907e32b0   /srv/dockers/mariadb
│   ├─vg_dataraid-lv_traefik      ext4                                       d557f43f-aaaf-4779-8fe3-223f24ae6a45   /srv/dockers/traefik
│   ├─vg_dataraid-lv_portainer    ext4                                       b4f992ac-55b1-4d10-a295-0ec2973a2697   /srv/dockers/portainer
│   └─vg_dataraid-lv_documents    ext4                                       71aad055-c80c-400a-bd0c-90000b1185dd   /srv/documents
└─sdd2                            LVM2_member                                cJIuU9-APvv-XhYU-JDfz-OJ9f-1VDj-9NOZFO
[olivealb@kyubi ~]$
```

Esmentar que per defecte Centos porposa XFS com a sistema de fitxers, en aquest cas optarem per ser més conservadors i es formataran els lv's amb EXT4, un sistema de fitxers més tradicional i més madur.

3.3 Configuració i preparació d'usuaris

Atès que mapajarem uns sistemes de fitxers amb els diferents contenidors per tal de que puguin desar dades i configuracions que els són pròpies com veurem més endavant i que volem conservar més enllà del cicle de vida del contenidor, farem servir uns usuaris diferents d'accés per cada servei en el host de forma que aquests sistemes de fitxers quedin aïllats al màxim els uns dels altres.



L'ús d'aquests directoris el veurem més endavant.

3.4 Instal·lació del servei de contenidors virtuals

Docker proporciona dues versions del programari de contenidors, la versió Enterprise (EE) de pagament i la versió OSS Comunity (CE) gratuïta, la principal diferència entre les dues versions és que la versió EE està certificada per uns sistemes determinats i el suport el proporciona la mateixa Docker.

Seguint els criteris del projecte s'utilitzarà la versió estable de CE al ser Open Source Software. Per instal·lar aquesta versió caldrà afegir el repositori de docker-ce:

*sudo yum-config-manager --add-repo *

https://download.docker.com/linux/centos/docker-ce.repo

Nota: Abans de procedir amb la instal·lació cal comprovar que els paquets: device-mapper-persistent-data i lvm2 estan instal·lats en el nostre sistema.

Un cop realitzades les comprovacions procedirem amb la instal·lació executant:

sudo yum install docker-ce

I un cop finalitzada la instal·lació s'arrancadaa el servei i s'activarà perquè arranqui de forma automàtica en encendre el servidor amb:

sudo systemctl start docker

sudo systemctl enable docker

Per comprobar que el servei està en marxa i els paràmetres actius amb *docker info*:

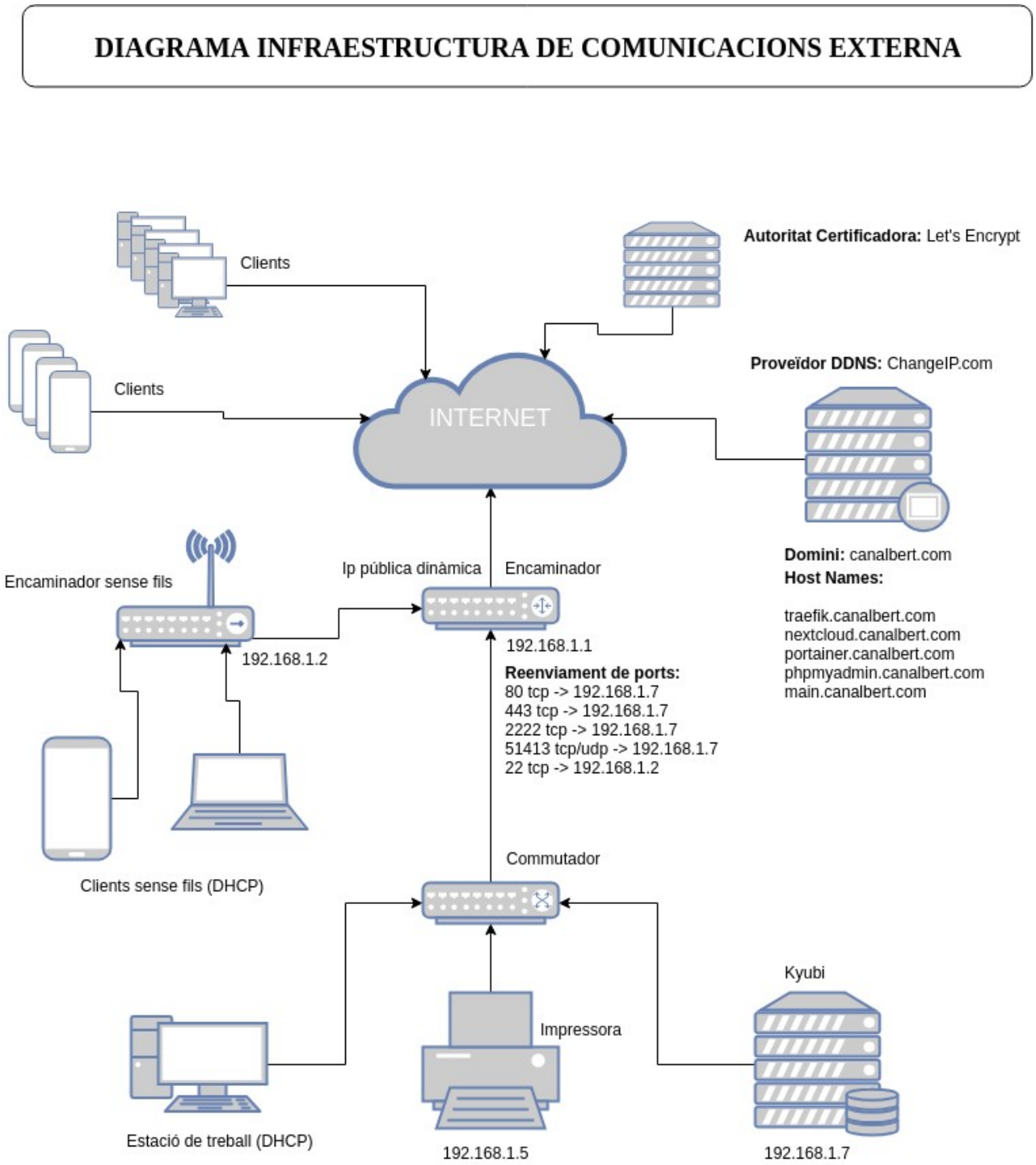
```
Server Version: 18.09.0
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: c4446665cb9c30056f4998ed953e6d4ff22c7c39
runc version: 4fc53a81fb7c994640722ac585fa9ca548971871
init version: fec3683
Security Options:
 seccomp
  Profile: default
Kernel Version: 3.10.0-862.14.4.el7.x86_64
Operating System: CentOS Linux 7 (Core)
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 3.658GiB
Name: kyubi
ID: 7BGB:CHXN:WZFI:IGRW:WDFP:UNAF:CV4Q:3VOG:MXBU:VJOU:5RNN:QH3F
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 127.0.0.0/8
Live Restore Enabled: false
Product License: Community Engine
```

3.5 Configuració infraestructura de comunicacions

Per fer més entenedora l'explicació, dividirem la infraestructura de comunicacions en dues parts: la infraestructura externa i la infraestructura interna virtual dels contenidors.

3.5.1 Infraestructura externa

La infraestructura externa inclou tota la infraestructura que no forma part dels contenidors ni del host. Començarem per la part més externa, conformada per Internet en general i els diferents serveis necessaris, per acabar amb la part local: la configuració de l'encaminador. A continuació l'esquema d'aquesta infraestructura:



Esquema realitzat amb l'utilitat Open source: draw.io (https://www.draw.io)

3.5.1.1 Servei de registre DDNS (dinamic DNS)

Òbviament si és necessari poder accedir als serveis proporcionats per la infraestructura interna des de l'exterior de la nostra xarxa, cal que els DNS d'internet resolguin els nostres hostnames, per tant cal registrar-los. Atès que no es disposa d'una Ip pública fixa cal fer el registre en un servei de DDNS (dynamic DNS) i com que no disposem tampoc de servidors DNS locals cal contractar un hosting DNS. Tots aquests serveis han estat contractats a changeIP.com (existeixen molts altres proveïdors però el preu és correcte, sobretot el de renovació)

Existien dues alternatives per fer el registre, utilitzar el servei gratuït que permet registrar un hostname amb un domini a escollir entre diversos predeterminats com: ddns.info, changeip.com, dynamic-dns.net, etc.. I llavors realitzar les redireccions internes als contenidors d'acord amb el subdomini per exemple canalbert.ddns.info/nextcloud, canalbert.ddns.info/portainer, etc.. O bé, l'opció de pagament, registrar un domini complet, pots escollir el que vulguis mentre no estigui ja registrat i utilitzar els hostnames per realitzar les redireccions als diferents contenidors per exemple: nextcloud.canalbert.com, portainer.canalbert.com, etc.. Finalment s'ha optat per aquesta segona opció tot i ser de pagamanet per dos motius principalment: la possibilitat d'escollir el nom del domini i la fiabilitat del servei de pagament. S'ha registrat el domini canalbert.com amb els següents hostnames:

traefik.canalbert.com

nextcloud.canalbert.com

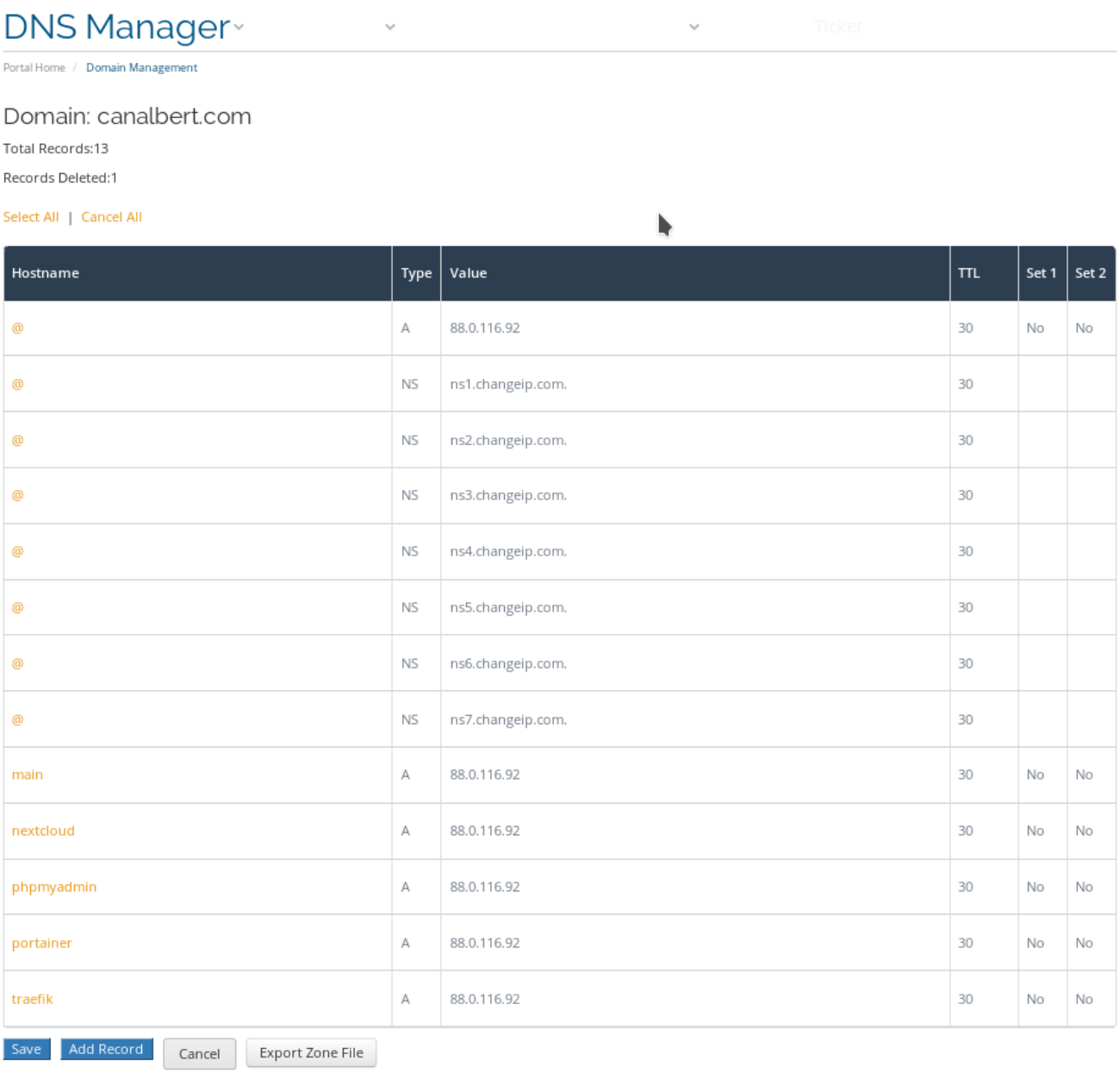
portainer.canalbert.com

phpmyadmin.canalbert.com

main.canalbert.com

Al tractar-se d'un servei DDNS caldrà configurar un mecanisme per actualitzar la ip del domini quan aquesta canviï, aquest mecanisme entre altres configuracions es tractarà en un punt posterior.

El cost de registrar el domini canalbert.com en changeIP.com és de 15$ amb un cost de renovació anual de 6$ i el cost del servei anual de hosting DNS és de 9$.



DNS Manager

Portal Home / Domain Management

Domain: canalbert.com

Total Records:13

Records Deleted:1

Select All | Cancel All

Hostname	Type	Value	TTL	Set 1	Set 2
@	A	88.0.116.92	30	No	No
@	NS	ns1.changeip.com.	30		
@	NS	ns2.changeip.com.	30		
@	NS	ns3.changeip.com.	30		
@	NS	ns4.changeip.com.	30		
@	NS	ns5.changeip.com.	30		
@	NS	ns6.changeip.com.	30		
@	NS	ns7.changeip.com.	30		
main	A	88.0.116.92	30	No	No
nextcloud	A	88.0.116.92	30	No	No
phpmyadmin	A	88.0.116.92	30	No	No
portainer	A	88.0.116.92	30	No	No
traefik	A	88.0.116.92	30	No	No

Save Add Record Cancel Export Zone File

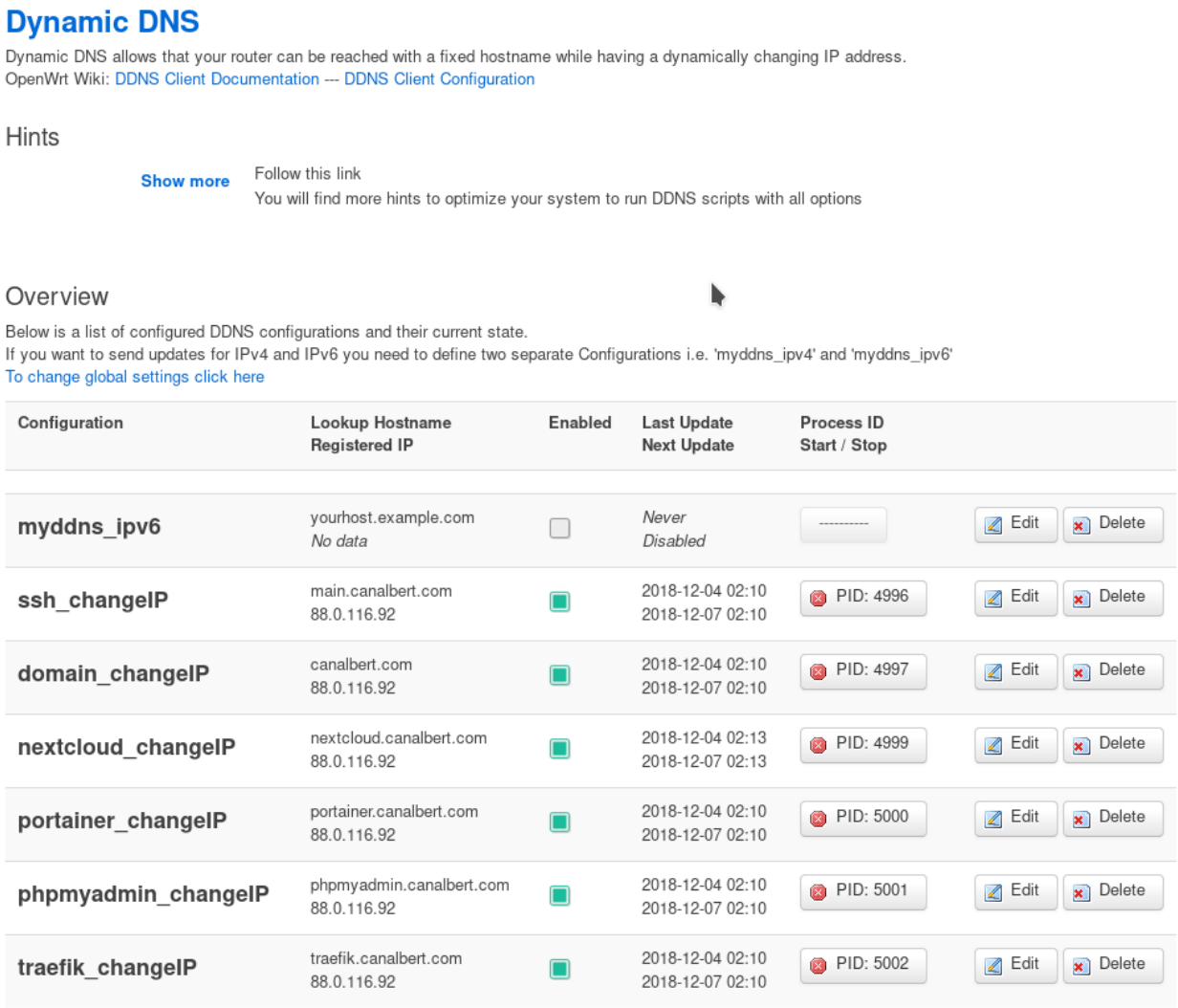
3.5.1.2 *Autoritat certificadora*

Un altre servei imprescindible pel projecte és l'autoritat certificadora per tal d'obtenir els certificats x.509 necessaris per a l'encriptació del protocol https (tls/ssl) i que verifiqui davant la resta del món que el certificat ens identifica a nosaltres. S'ha escollit Let's Encrypt atès que ofereix aquest servei de franc amb una expiració de 90 dies i que es pot renovar en qualsevol moment. Més endavant veurem que el contenidor redirector (traefik) s'encarregarà d'obtenir i renovar aquests certificats.

3.5.1.3 Configuració de l'encaminador local

És necessari configurar diferents serveis adicionals en l'encaminador local, ja que a més de servir-nos de sortida cap internet i fer de tallafocs, ara cal que també actualitzi els DNS quan canviï la Ip proporcionada pel proveïdor d'internet (O2) i que encamini les peticions externes cap al nostre servidor.

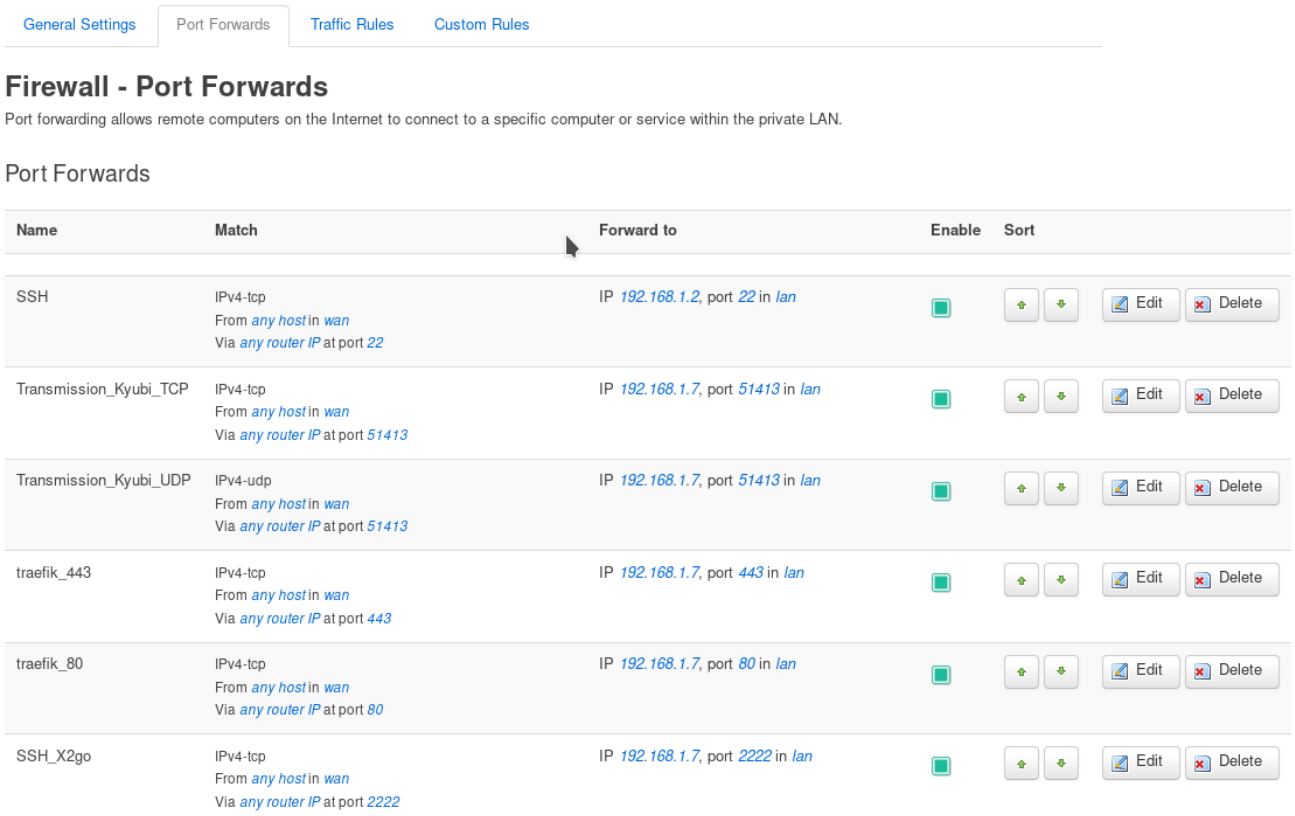
Per començar s'ha instal·lat i configurat el servei de Dynamic DNS de l'encaminador, en aquest cas l'encaminador té instal·lat el firmware de lede-17.01 (una evolució de l'OpenWRT) que permet instalar móduls adicionals. Cal crear una entrada per cada hostname registrat en el proveïdor de DNS:



Per finalitzar, la configuració del reenviament dels ports interns cap al servidor (192.168.1.7) fonamentalment el 443 i el 80 que posteriorment el contenidor de traefik s'encarregarà de reencaminar als contenidors corresponents. El servei de X2go utilitza

SSH i no pot ser redirigit per traefik per tant he decidit redirigir el port 2222 públic cap al servidor i com veurem més endavant es mapejarà cap al contenidor al port 22.

El port 22 públic es reencamina a un segon encaminador (amb OpenWRT també) sense fils auxiliar (192.168.1.2) amb el servei de dropbear instal·lat (un servidor SSH lleuger) per ús intern d'administració que em proporciona **una connexió SSH alternativa en cas necessari,** atès que si la resta d'accessos no estan operatius, els encaminadors sempre ho haurien d'estar.



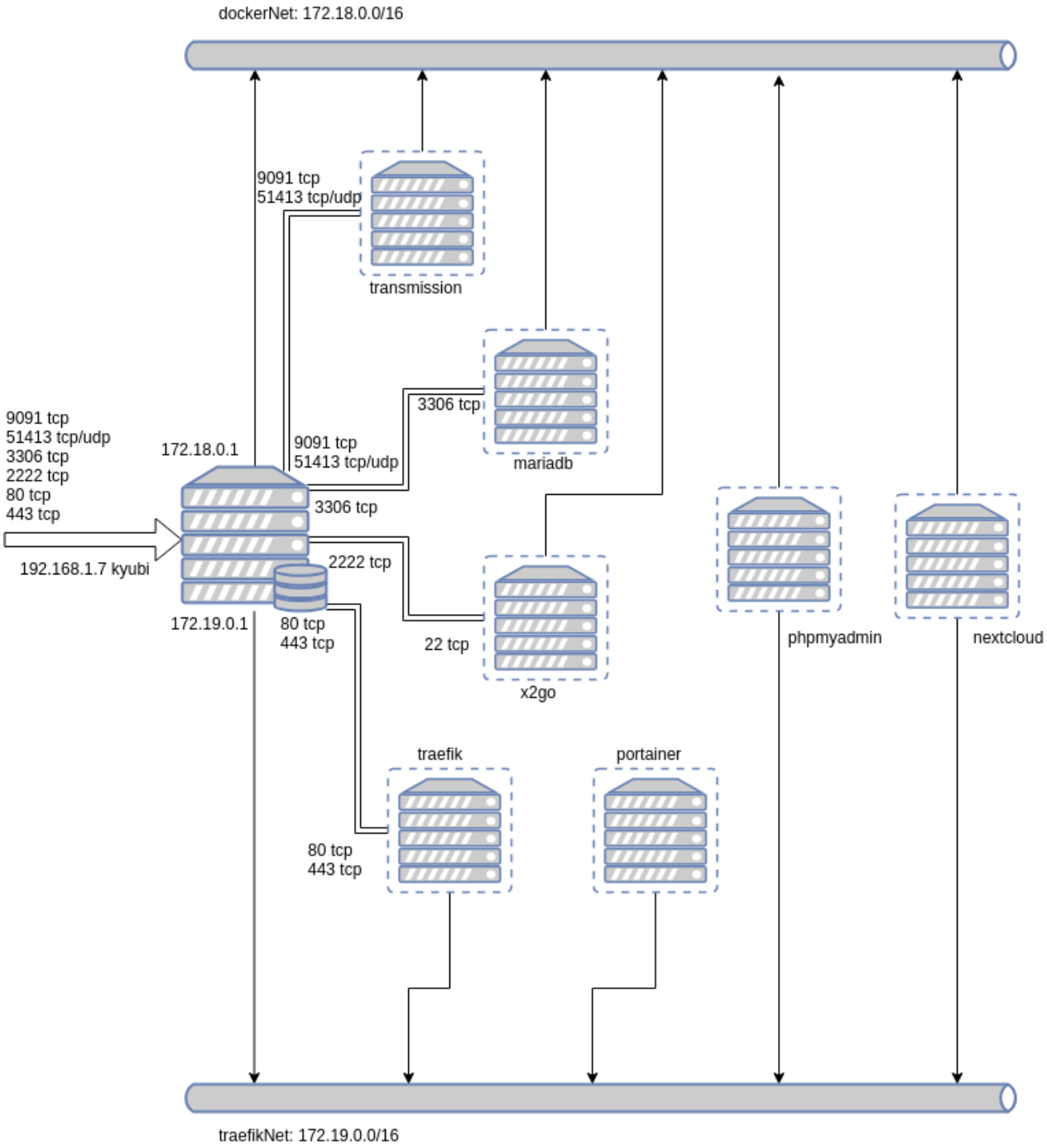
Esmentar que pel correcte funcionament del servei de transmission també es reencaminen els ports 51413 TCP/UDP.

3.5.2 Infraestructura interna

Considerarem infraestructura de comunicacions interna la proporcionada pel servei de contenidors (Docker) i que serà, per tant, virtual. Docker crea una xarxa virtual per defecte del tipus bridge i per defecte afegeix aquesta xarxa a qualsevol contenidor creat de forma que tinguin connectivitat entre ells, també aïlla els contenidors de l'exterior.

Pel nostre projecte és necessari exposar certs ports a l'exterior i crear diferents xarxes per aïllar els serveis exposats a l'exterior dels serveis auxiliars interns. A continuació un esquema d'aquesta infraestructura interna:

DIAGRAMA INFRAESTRUCTURA DE COMUNICACIONS INTERNA



Esquema realitzat amb l'utilitat Open source: draw.io (https://www.draw.io)

3.5.2.1 Xarxes virtuals

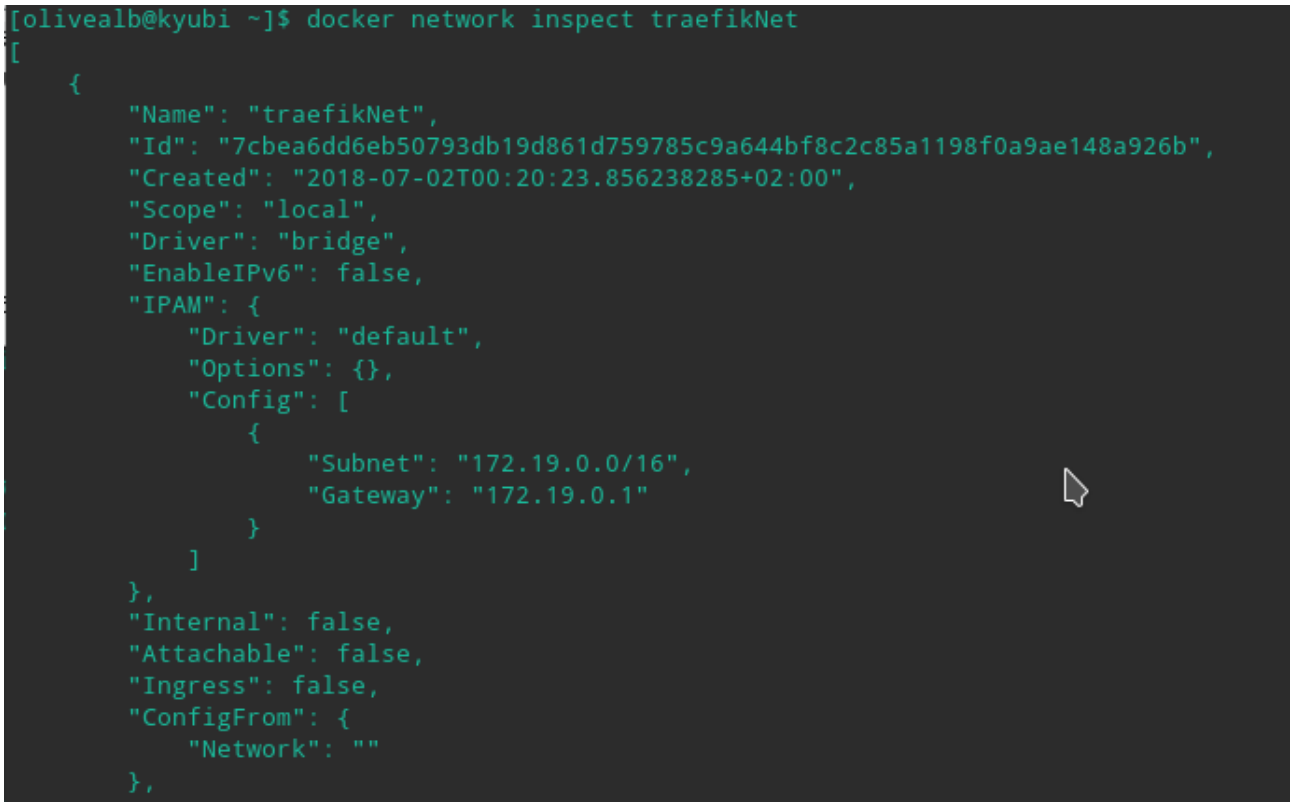
S'ha creat una xarxa virtual per realitzar les redireccions des de l'exterior i un altre per connectar els diferents contenidors de serveis, les dues xarxes restaran aïllades una de l'altre i addicionalment mantindran els contenidors del projecte aïllats d'altres possibles contenidors que poguessin crear-se posteriorment. Per crear les xarxes virtuals de tipus bridge executarem:

```
docker network create dockerNet
```

```
docker network create traefikNet
```

Per comprovar la correcta creació executarem docker network inspect:

```
[olivealb@kyubi ~]$ docker network inspect dockerNet
[
    {
        "Name": "dockerNet",
        "Id": "c088c50e4f2b0e0f76ff772544f4ef5636ee9177b4a392c3619baf96048ca7bb",
        "Created": "2018-07-02T00:20:07.001054753+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
```



Es pot obsevar que per cada xarxa virtual Docker crea per defecte un rang Ip de classe B, tot i que aquests paràmetres es poden indicar si es desitjen altres valors. En el moment de crear els contenidors caldrà indicar a quina xarxa pertany cadascun.

3.5.2.2 Mapeig de ports (publicació de ports)

L'altre aspecte a tenir en compte dins de la infraestructura interna dels contenidors és l'accés des de l'exterior als contenidors que no proporcionen els serveis http/https i que resulta interessant poder-hi accedir directament. Per defecte Docker crea els contenidors totalment aïllats de l'exterior, si cal accedir a un port d'un contenidor directament des de l'exterior, cal mapejar aquests ports amb el host, aquest mapeig es realitza en el moment de crear el contenidor com es veurà més endavant.

3.6 Instal·lació i configuració del contenidor redirector

El contenidor redirector de peticions (traefik) és un punt crític pel correcte funcionament del projecte atès que s'encarrega de reencaminar les peticions rebudes pels ports 80 i 443, i en funció del hostname, redirigir-les cap al contenidor corresponent al servei, també gestiona els certificats amb Let's Encrypt. Per aquest motiu s'ha inclòs en un punt de la preparació de la infraestructura i no en el de configuració de serveis.

3.6.1 Instal·lació traefik

Per la instal·lació del contenidor utilitzarem el repository oficial executant la comanda:

```
docker create --name=traefik -p 80:80 -p 443:443 -e PUID=1507 -e PGID=1507 -e TZ=Europe/Madrid --network traefikNet -v /var/run/docker.sock:/var/run/docker.sock -v /srv/dockers/traefik:/etc/traefik -v /srv/shared:/shared -l "traefik.enable=true" -l "traefik.frontend.rule=Host:traefik.canalbert.com" -l "traefik.port=8080" -l "traefik.docker.network=traefikNet" traefik
```

Veure el punt 4.1 per l'explicació dels paràmetres utilitzats en la creació dels contenidors.

Mapejarem els ports 80 i 443 amb el host, el contenidor es conectarà a la xarxa virtual traefikNet, informarem les variables d'entorn corresponents el PUID i el PGID i la zona horària. Muntarem el directori /srv/dockers/traefik en /etc/traefik del contenidor per desar les configuracions específiques de traefik i els certificats, també muntarem el directori /var/run/docker.sock en /var/run/docker.sock del contenidor perquè tingui accés al End Point local de Docker i pugui accedir als contenidors creats i finalment muntarem el directori /srv/shared per desar el fitxer d'autenticació .htpasswd que haurem generat prèviament.

Per acabar, setejarem les etiquetes específiques de l'aplicació:

traefik.enable=true → activades les redireccions per ell mateix.

traefik.frontend.rule=Host:traefik.canalbert.com → redireccionament de les peticions pel hostname.

traefik.port=8080 → traefik ha de reencaminar el port 443 al 8080

traefik.docker.network=traefikNet → la xarxa utilitzada per redirigir les peticions.

Un cop creat, per arrancar el contenidor només cal executar: **docker start traefik**.

3.6.2 Configuració traefik

Traefik desa la configuració en el fitxer traefik.toml que com s'ha comentat en el punt anterior està muntat en /srv/dockers/traefik/traefik.toml i que es crearà amb les opcions per defecte automàticament en arrencar el primer cop el contenidor. Cal modificar aquesta configuració perquè el comportament sigui el desitjat, comentarem les parts més rellevants de la configuració.

```
[olivealb@kyubi ~]$ cat /srv/dockers/traefik/traefik.toml
logLevel = "WARN" #DEBUG, INFO, WARN, ERROR, FATAL, PANIC
defaultEntryPoints = ["http", "https"]

InsecureSkipVerify = true

# WEB interface of Traefik - it will show web page with overview of frontend and backend configurations
[web]
address = ":8080"
  [web.auth.basic]
  usersFile = "/shared/.htpasswd"

# Force HTTPS
[entryPoints]
  [entryPoints.http]
     address = ":80"
  [entryPoints.http.redirect]
     entryPoint = "https"

  [entryPoints.https]
     address = ":443"

 [entryPoints.https.tls]

[file]
  watch = true
  filename = "/etc/traefik/rules.toml"

# Let's encrypt configuration
[acme]
  email = "aoliveras@gmail.com" #any email id will work
  storage="/etc/traefik/acme/acme.json"
  entryPoint = "https"
  acmeLogging=true
  onDemand = false #create certificate when container is created
  onHostRule = true

  # Use a HTTP-01 acme challenge rather than TLS-SNI-01 challenge
  [acme.httpChallenge]
    entryPoint = "http"

  [[acme.domains]]
  main = "canalbert.com"
  sans = ["phpmyadmin.canalbert.com","nextcloud.canalbert.com","portainer.canalbert.com","traefik.canalbert.com"]

# Connection to docker host system (docker.sock)
[docker]
  endpoint = "unix:///var/run/docker.sock"
  domain = "canalbert.com"
  watch = true
# This will hide all docker containers that don't have explicitly
# set label to "enable"
  exposedbydefault = false
```

En primer lloc s'ha configurat una autenticació sencilla creant un fitxer .htpasswd amb un usuari i password encriptats per la pàgina web de traefik a la qual s'accedeix pel port 8080, tot i que des de l'exterior nosaltres accedirem de forma segura sempre pel port 443 a tots els serveis web (veure el següent punt 3.5.3).

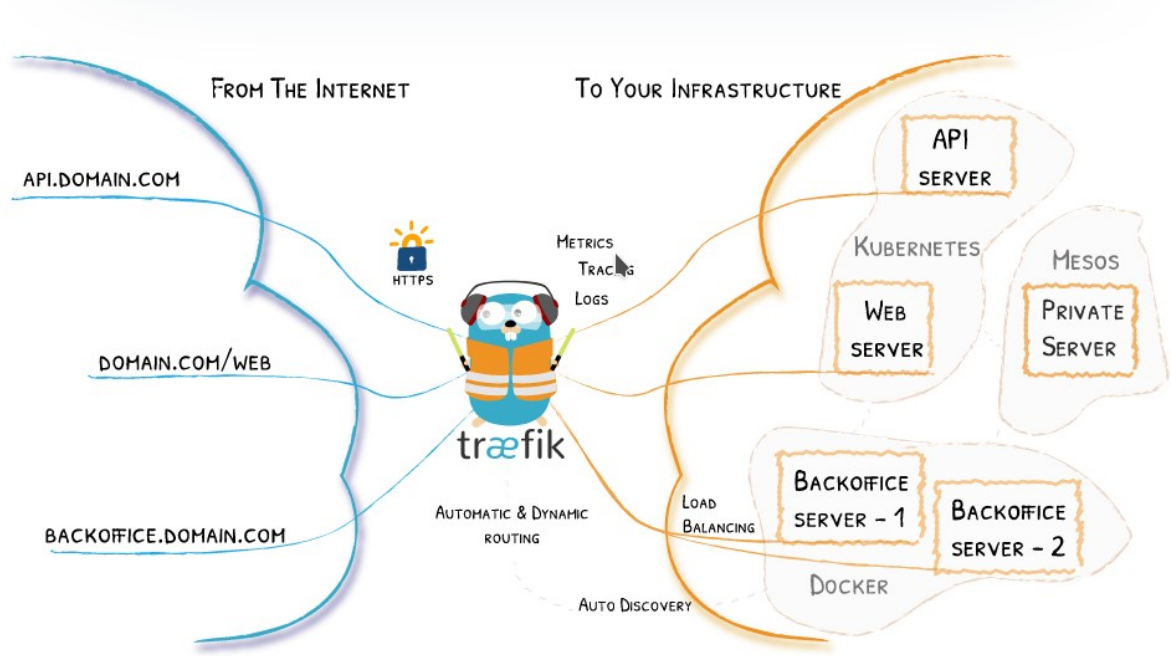
A continuació s'ha forçat la redirecció del port 80 contra el 443 que és sobre el que es realitzaran les redireccions cap a l'interior.

Seguidament s'informa d'on ha de desar els certificats de Let's encrypt i que s'obtingui certificats pels contenidors segons la regla que s'indiqui a l'etiqueta **traefik.frontend.rule** durant la creació del contenidor. Let's Encrypt utilitza el protocol Acme per comprovar que el certificat és correcte i en el nostre cas farà servir el HTTP-01 Acme challenge per realitzar aquesta comprovació (l'explicació d'aquest protocol queda fóra de l'abast del projecte, per més informació: https://letsencrypt.org/how-it-works/). Cal especificar també els hostnames i domini vàlid dels contenidors que seran reencaminats, si el proveïdor DNS ho soporta i té una api compatible amb Let's Encrypt, es pot especificar un altre tipus de protocol que accepti wildcards's pel domini de forma que no calgui especificar cada hostname del domini, no és el nostre cas.

Finalment s'indica que no per tots els contenidors creats traefik gestionarà els certificats i la redirecció corresponent, només els contenidors que tinguin l'etiqueta **traefik.enable=true** en els paràmetres de creació ho farà.

3.6.3 Integració de traefik amb Docker

Atès que traefik té accés al End point de Docker, comprova cada cop que es crea i s'engega un contenidor si conté l'etiqueta **traefik.enable=true** i en cas afirmatiu crea una redirecció de forma automàtica cap al contenidor creat segons les etiquetes **traefik.frontend.rule** i **traefik.port** actuant com un reverse proxy de forma que les comunicacions exteriors són sempre encriptades i segures actuant ell mateix com a frontal.

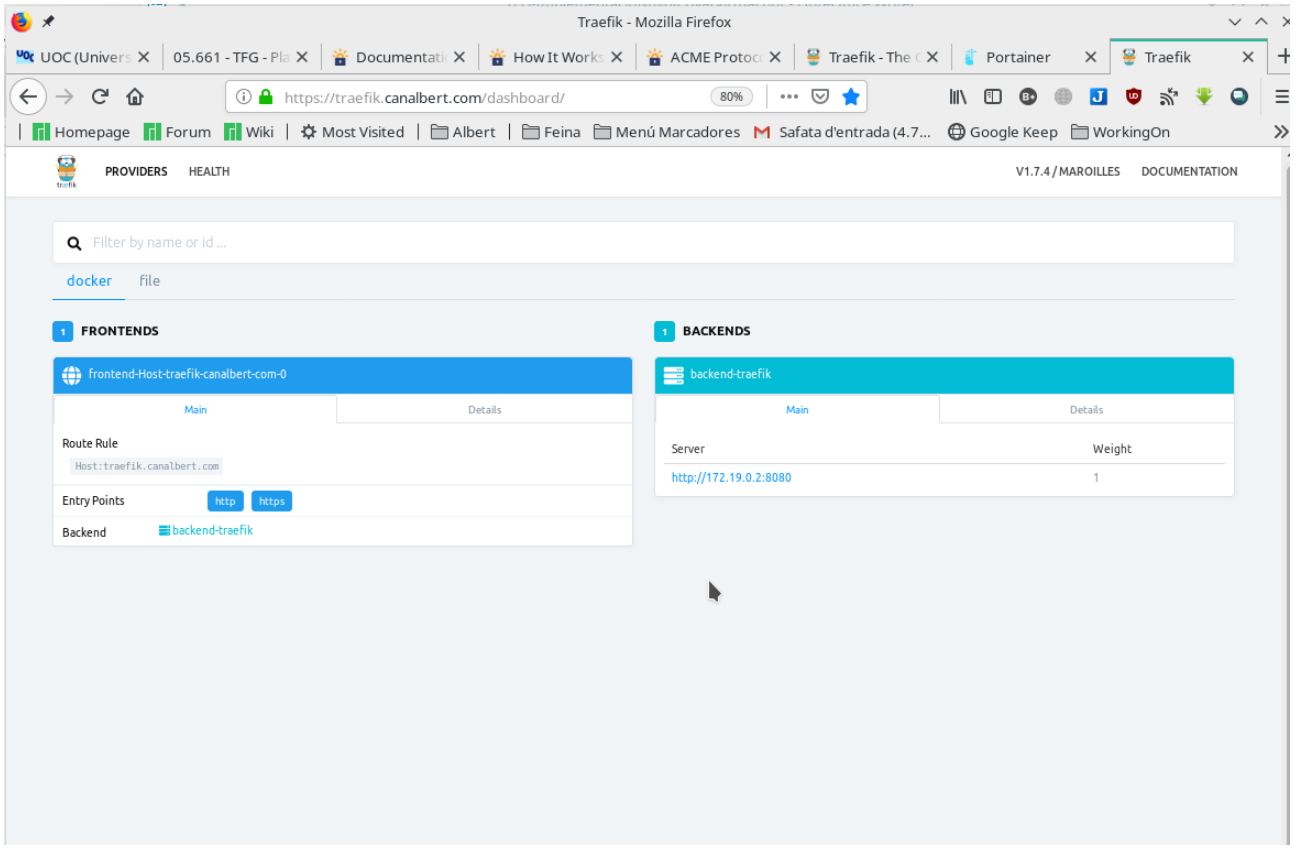


Imatge extreta de https://traefik.io/

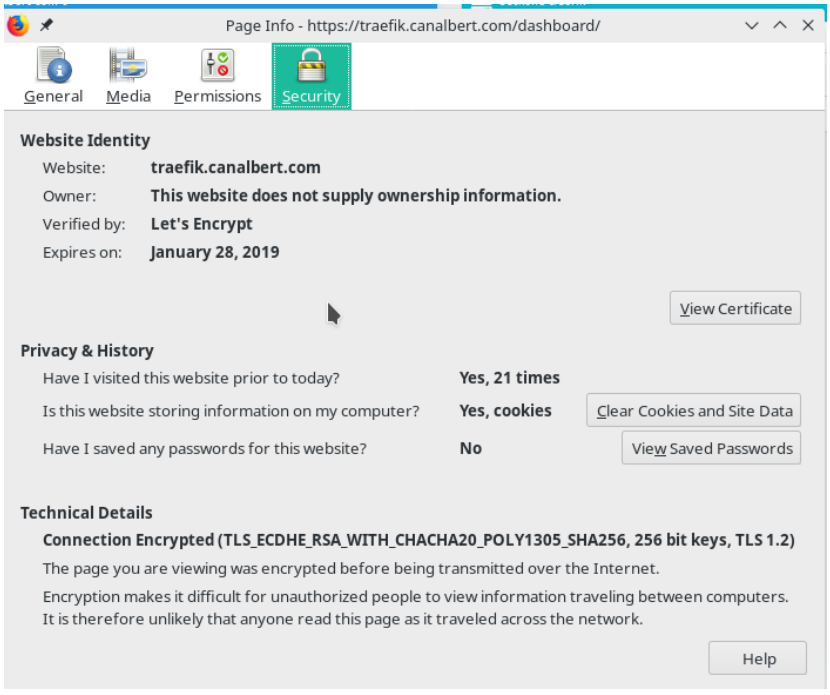
3.7 Proves de connexió i redirecció

Un cop arrancat aquest primer contenidor, ja podem comprovar si traefik està fent la seva feina correctament atès que ell mateix té un frontal web que presenta un dashboard dinàmic que mostra els frontends i els seus corresponents backends actius en temps real. Aquest frontal web està reencaminat per ell mateix (443 → 8080) i també tindrà assignat un certificat verificat per Let's Encrypt.

Per accedir al frontal de traefik introduirem en el navegador la URL: https://traefik.canalbert.com tal com es va indicar amb l'etiqueta **traefik.frontend.rule=Host:traefik.canalbert.com** en crear el contenidor.



Un altre detall important és el cadenat verd que es mostra a l'esquerra de la barra de navegació que ens indica que el protocol és segur i que el certificat està verificat per una autoritat certificadora que confirma que el domini és el que correspon amb el certificat.



4 CONFIGURACIÓ DE SERVEIS

A continuació es procedirà a la creació dels contenidors que proporcionaran els diferents serveis dels quals consta el projecte, però abans, un primer punt per descriure els principals paràmetres que s'utilitzaran en la creació dels mateixos i que s'aniran repetint per cada contenidor.

4.1 Sobre la creació de contenidors

En general, per cada contenidor haurem aprovisionat prèviament un o més d'un sistema de fitxers en el host (veure punt 3.2) i un usuari propi amb permisos per accedir-hi. Bàsicament la creació del contenidor es realitza amb la crida docker create i una sèrie de paràmetres que identifiquen els contenidors o els proporcionen certes característiques, unes variables d'entorn, unes etiquetes que ajuden a la configuració del servei de traefik i finalment la imatge que s'utilitzarà com a base i que s'obtindrà del repositori oficial de Docker.

Els paràmetres més importants són:

--name → El nom que volem proporcionar al contenidor. Obligat.

-p → El port que es mapejarà amb el host per poder tenir accés des de l'exterior, consta d'una tupla separada per **:** on el primer element representa el número de port del host i el segon element el número de port del contenidor aquests números poden ser iguals o diferents. Un contenidor pot tenir més d'un port publicat repetint el paràmetre. Opcional.

--network → Per defecte docker connecta els contenidors a una mateixa xarxa virtual per defecte, si volem connectar el contenidor a una xarxa específica prèviament creada hem d'informar aquest paràmetre. Cada contenidor pot estar connectat a una única xarxa virtual durant la creació però després es poden connectar d'altres de forma dinàmica. Opcional.

-v → Com ja s'ha comentat els contenidors són imatges invariables i si volem guardar una part variable com la configuració o les dades, cal muntar els directoris corresponents en un filesystem determinat. Aquest paràmetre consta d'una tupla d'elements separada també per **:** , el primer element és el directori del host on es muntarà el segon element que és un directori del contenidor. Opcional.

-e → Serveix per assignar variables d'entorn permeses per la imatge del contenidor, per exemple TZ per assignar el Time Zone, PUID i PGID per indicar quin usuari farà servir el contenidor, d'aquesta forma és poden aïllar encara més els contenidors en

el host i resulta imprescindible per configurar els permissos dels directoris indicats amb el paràmetre -v. Opcional.

-l → Serveix per assignar valors a certes etiquetes que després utilitzarà traefic per configurar les redireccions, obtenir els certificats, etc.. Com ja es va explicar en el punt 3.5.1. Opcional.

Per últim el nom de la imatge a partir de la qual es generarà el contenidor, si la imatge ja es troba en el repositori local s'utilitzarà i si no es baixarà del repositori oficial de Docker https://hub.docker.com/_/traefik/. També es pot indicar a més del nom de la imatge, un TAG de versió, si no s'indica res, utilitzarà l'última (latest). Obligat.

4.2 Configuració de serveis de suport i administració

4.2.1 Instal·lació i configuració del contenidor de bases de dades: MariaDB

Per la instal·lació d'aquest contenidor hem creat prèviament l'usuari mariadb amb PUID=1503 i GUID=1503 i utilitzarem el directori /srv/dockers/mariadb per desar la informació variable del contenidor. La crida per generar el contenidor és la següent:

```
docker create --name=mariadb -p 3306:3306 -e PUID=1503 -e PGID=1503 -e MYSQL_ROOT_PASSWORD=XXXXXXX -e TZ=Europe/Madrid --network dockerNet -v /srv/dockers/mariadb:/config linuxserver/mariadb
```

Es pot apreciar que amb la variable d'entorn MYSQL_ROOT_PASSWORD=XXXXXXX s'informa del password que volem que tingui l'usuari root de la instància de mariadb. Aquest contenidor no ha de ser accesible des d'internet per tant no utilitzem cap de les etiquetes de traefik, però sí mapejarem el port 3306 pel que respon la instància ja que ens permetrà fer connexions des de la xarxa local per fer consultes i/o backups. Per arrancadaar el contenidor executarem:

```
docker start mariadb
```

Comprovarem que la instància ha arrancadaat correctament en el següent punt quan comprovem que el servei de phpmyadmin ha arrancadaat i connecta amb la instància.

4.2.2 Instal·lació i configuració del contenidor d'administració de bases de dades: phpMyadmin

Aquest contenidor no necessita guardar cap dada ni configuració i per tant no necessita tampoc cap usuari amb permisos. La crida per generar el contenidor és la següent:

```
docker create --name=phpmyadmin -e PMA_HOST=mariadb -e MYSQL_ROOT_PASSWORD=XXXXXXX -e TZ=Europe/Madrid -e PMA_ABSOLUTE_URI=https://phpmyadmin.canalbert.com --network traefikNet -l "traefik.enable=true" -l "traefik.frontend.rule=Host:phpmyadmin.canalbert.com" -l "traefik.port=80" -l "traefik.docker.network=traefikNet" phpmyadmin/phpmyadmin
```

Es pot apreciar que mitjançant les variables d'entorn s'informa del password del usuari root (MYSQL_ROOT_PASSWORD) de la bd i del contenidor on està ubicada (PMA_HOST) la instància de mariaDB. Atès que a l'aplicació web d'aquest contenidor si s'ha de poder accedir des d'internet fem servir les etiquetes corresponents perquè traefik faci la redirecció pertinent. Per acabar, connectarem aquest contenidor també a la xarxa virtual dockerNet perquè tingui connectivitat amb el contenidor de bases de dades creat en el punt anterior executant:

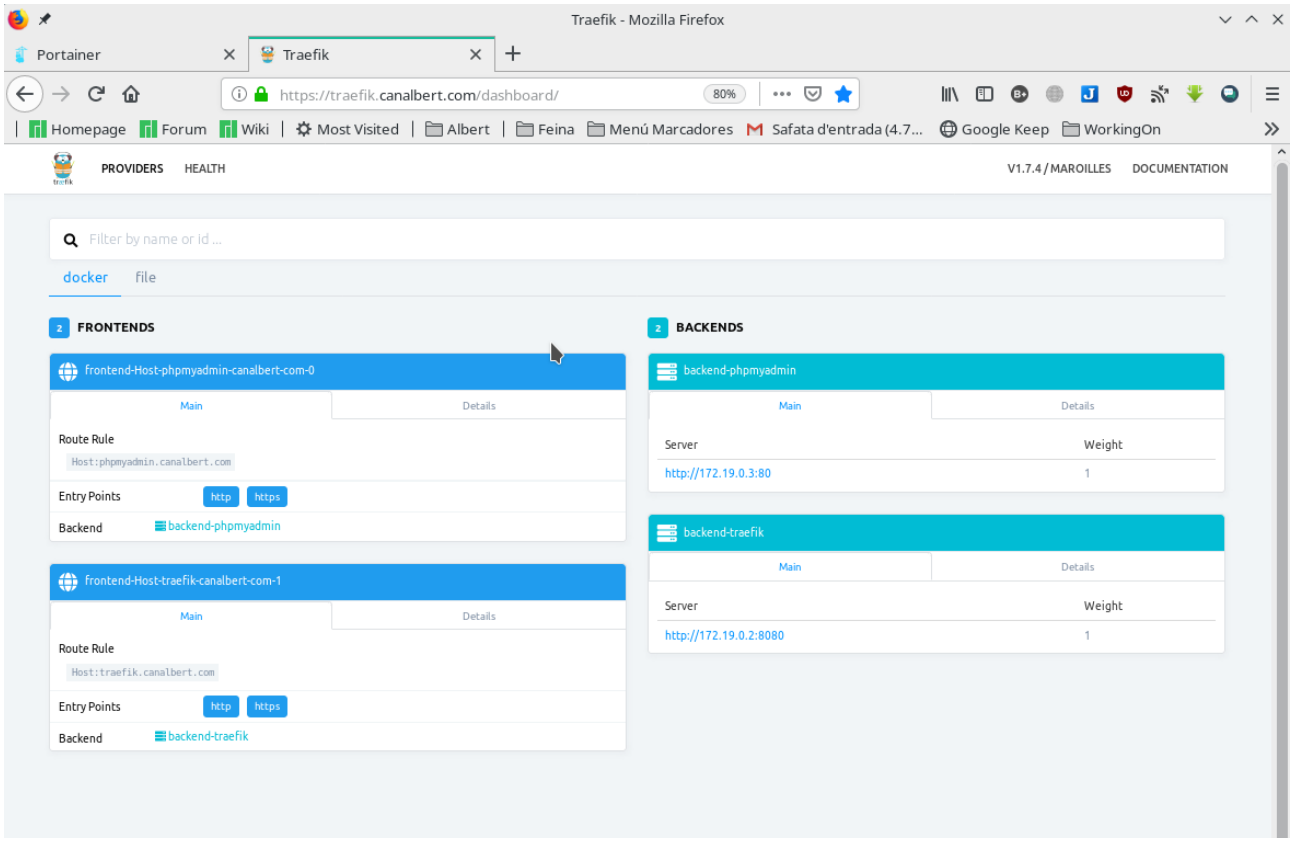
```
docker network connect dockerNet  phpmyadmin
```

i tot seguit arrancadaarem el contenidor amb:

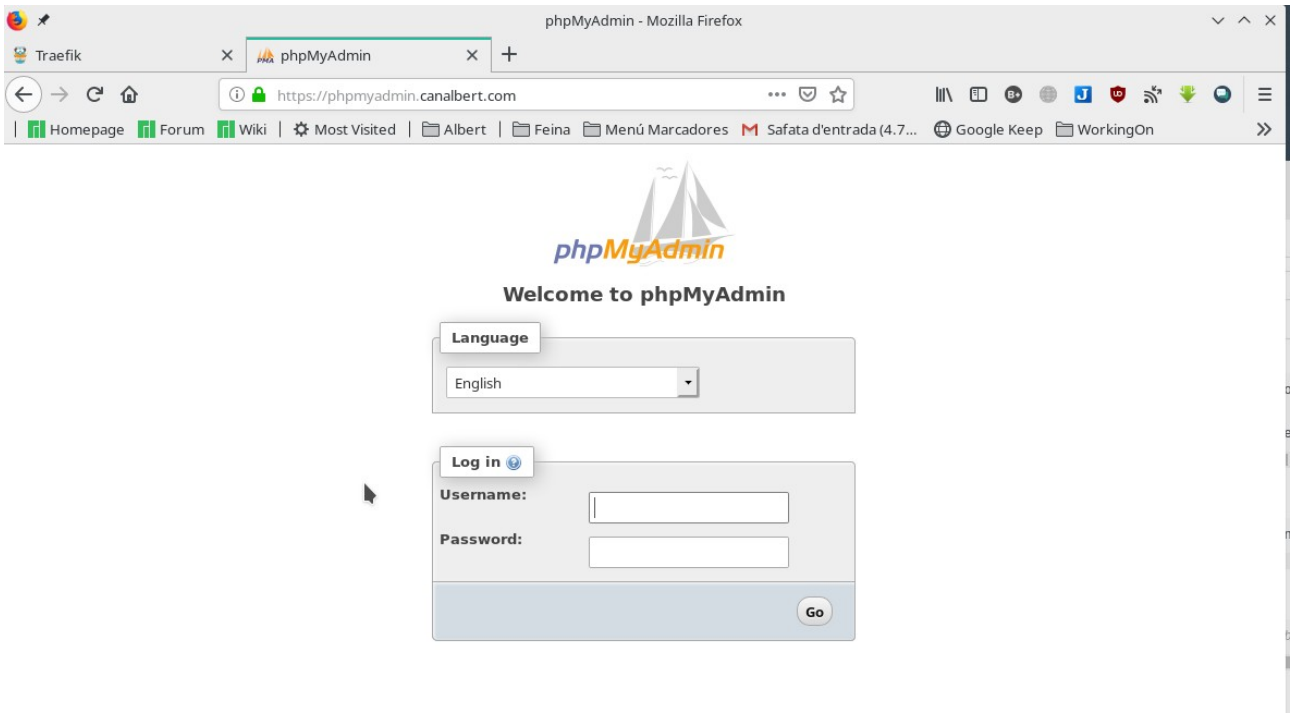
```
docker start phpmyadmin
```

4.2.3 Proves de connexió i redirecció

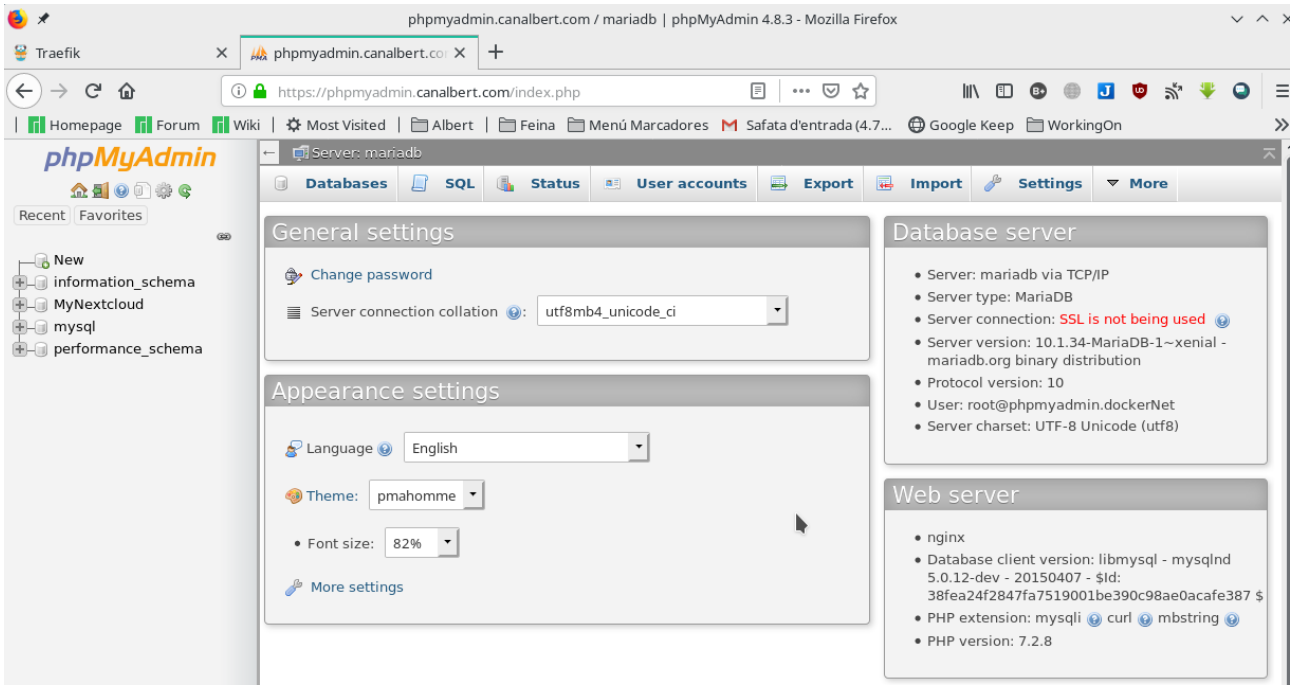
Si observem ara el dashboard de traefik podem comprovar que es mostra un nou frontal corresponent al servei de phpmyadmin.



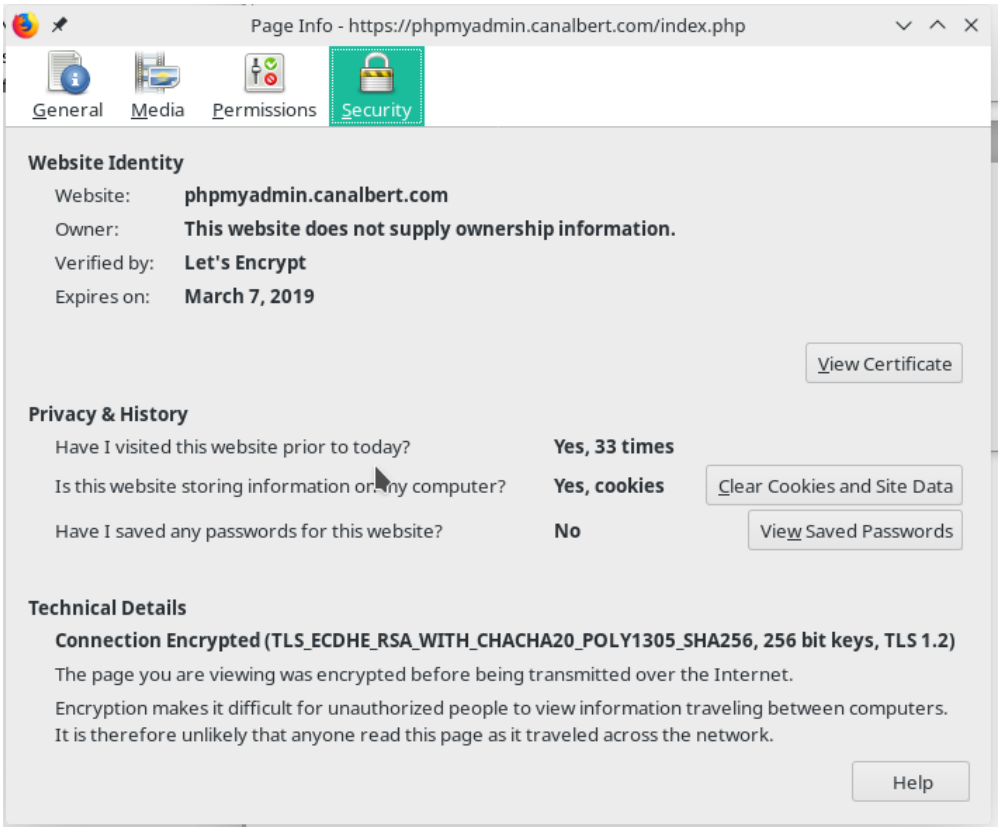
I si naveguem fins a la web https://phpmyadmin.canalbert.com



i si ens validem amb l'usuari root podem comprovar la connexió a la instància de mariaDB.



Destaquem un cop més el cadenat verd que indica la connexió segura i verificada per Lets Encrypt.



4.3 Configuració del servei d'emmagatzematge, sincronització i compartició

4.3.1 Instal·lació i configuració del contenidor: NextCloud

Per la instal·lació d'aquest contenidor hem creat prèviament l'usuari nextcloud amb PUID=1504 i GUID=1504 i utilitzarem el directori /srv/dockers/nextcloud per desar la informació variable del contenidor bàsicament la configuració i les dades. La crida per generar el contenidor és la següent:

```
docker create --name=nextcloud -e PUID=1504 -e PGID=1504 -v /srv/dockers/nextcloud/config:/var/www/html/config -v /srv/dockers/nextcloud/data:/var/www/html/data --network traefikNet -l "traefik.enable=true" -l "traefik.frontend.rule=Host:nextcloud.canalbert.com" -l "traefik.port=80" -l "traefik.docker.network=traefikNet" nextcloud
```

Tal i com s'ha explicat amb punts anteriors, podem observar les etiquetes per activar la gestió de traefik i atès que necessita desar informació en una bd de la instància de mariaDB el connectarem també a la xarxa virtual de dockerNet:

```
docker network connect dockerNet  nextcloud
```

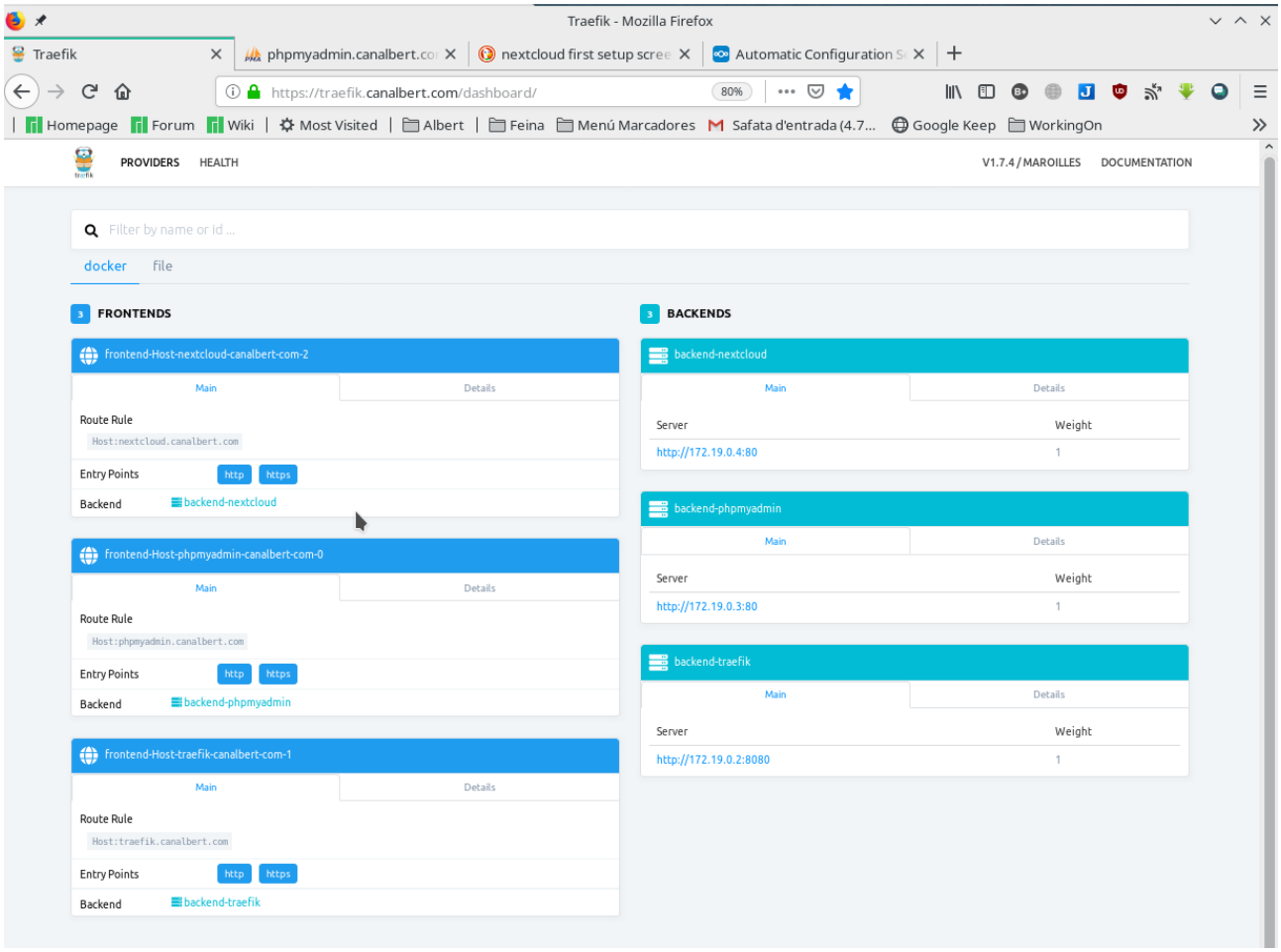
Tot seguit arrancarem el contenidor executant:

```
docker start nextcloud
```

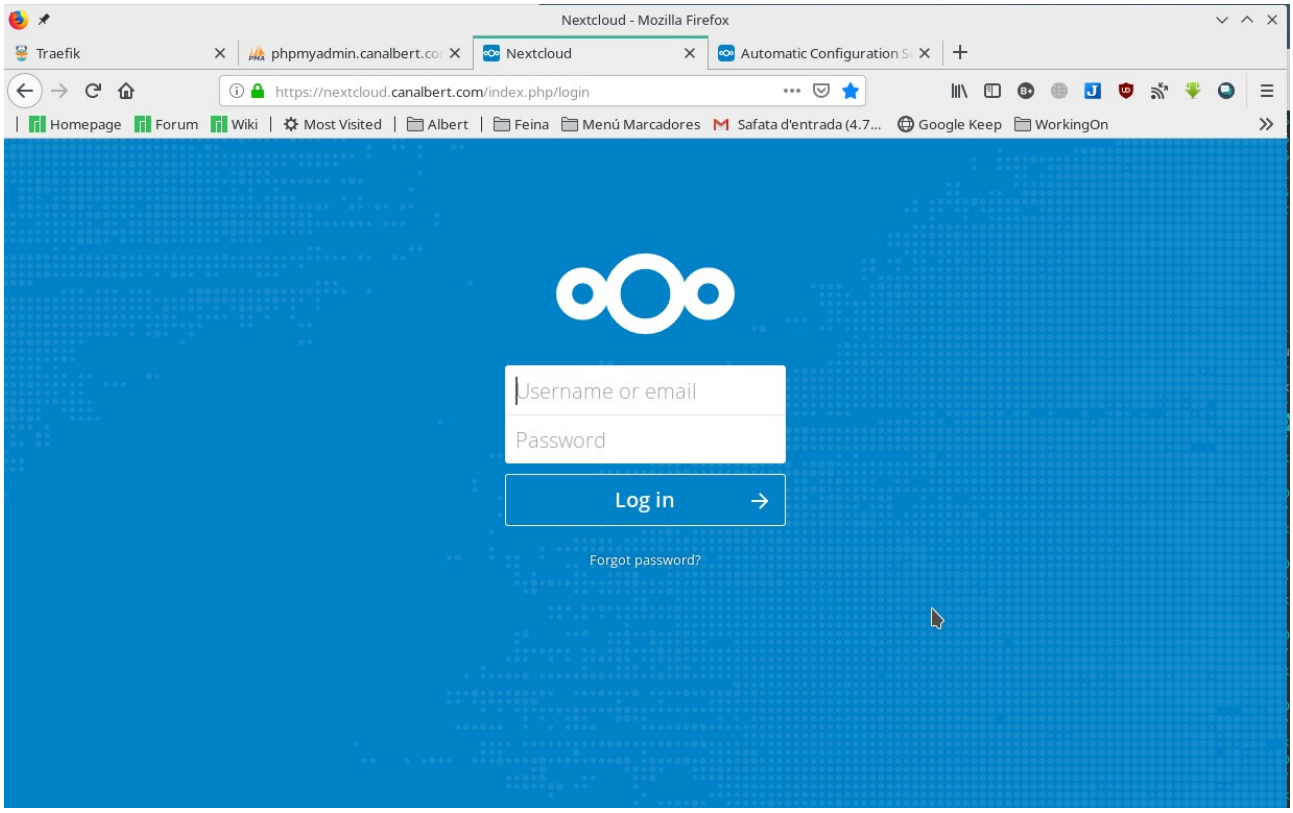
La primera vegada que arranca nextcloud mostra una pantalla de configuració sol·licitant un usuari i password administrador i les dades de connexió a la instància de mariaDB. Un cop realitzada la configuració inicial ja es pot realitzar la connexió normal via web.

4.3.2 Proves de connexió i redirecció

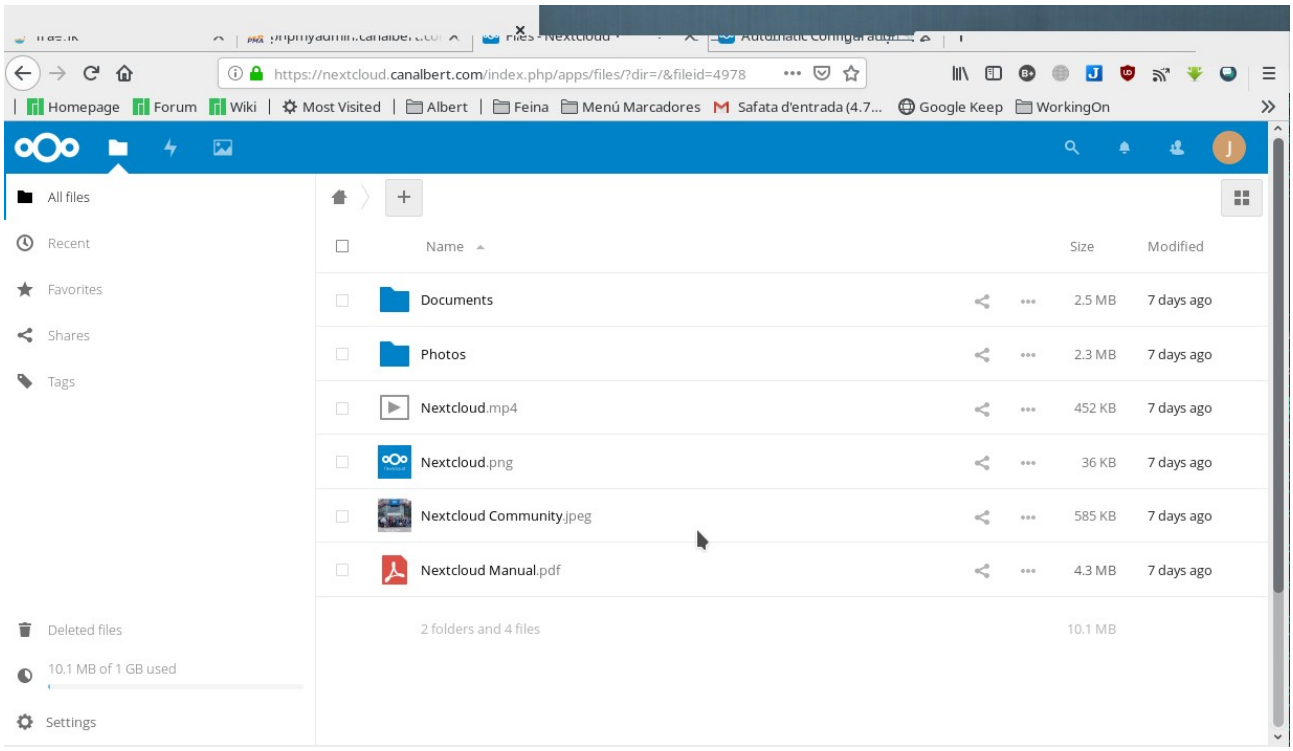
En primer lloc comprovarem que el frontal de traefik està aixecat:



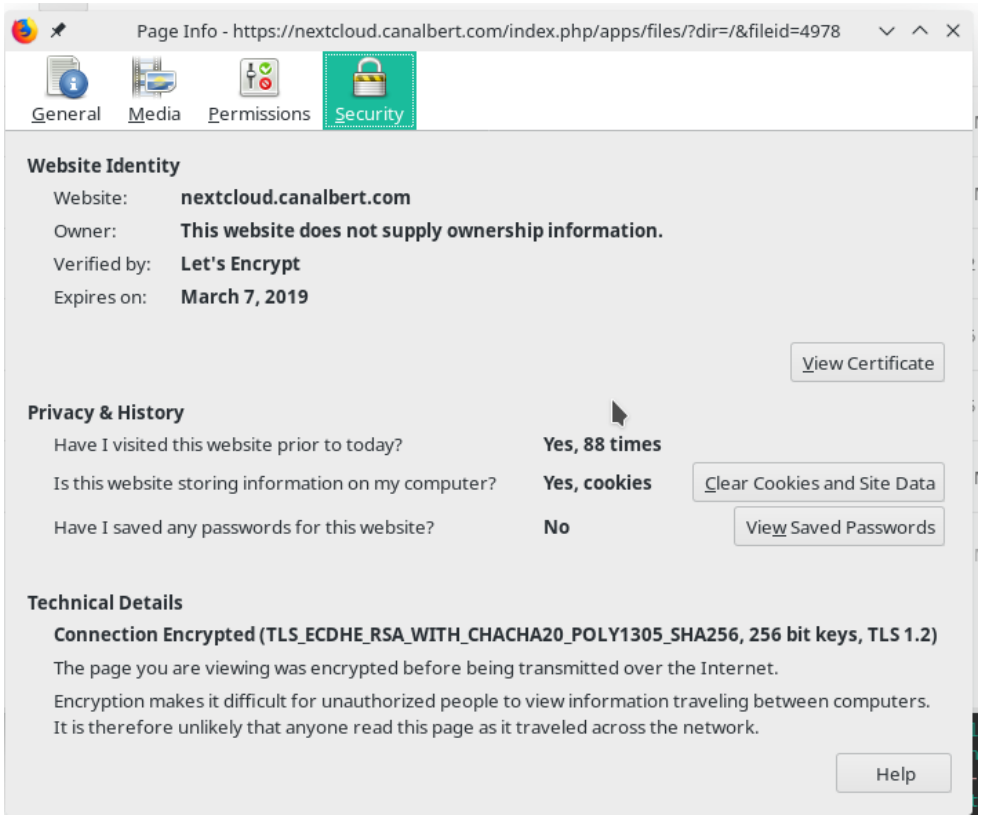
i a continuació la connexió a https://nextcloud.canalbert.com:



i si ens validem amb l'usuari de prova consultor/Consult0r:



El certificat de la pàgina:



Nextcloud disposa també d'un client multiplataforma d'escriptori que permet mantenir els directoris que vulguem sincronitzats amb el servidor i també un bon grapat de plug-ins per afegir serveis: notes, col·laboració, xat, etc..

4.4 Configuració d'altres serveis

Alguns d'aquests serveis no es publicaran a internet mitjançant traefic atès que només és capaç de reencaminar tràfic http i https com per exemple el servei de sessions remotes que utilitza el protocol ssh. El cas del servei de descàrregues es podria publicar amb traefik però he decidit no fer-ho, ja que aquest servei només té sentit des de la xarxa interior. En tot cas, sempre es podria utilitzar el servei de sessions remotes per connectar amb el transmission des d'internet en cas necessari.

4.4.1 Instal·lació, configuració i proves del contenidor de descàrregues: Transmission

Per la instal·lació d'aquest contenidor hem creat prèviament l'usuari transmission amb PUID=1500 i GUID=1500 i utilitzarem el directori /srv/dockers/transmission i /srv/shared/downloads per desar la informació variable del contenidor bàsicament la configuració i les descàrregues. La crida per generar el contenidor és la següent:

```
docker create --name=transmission -p 9091:9091 -p 51413:51413 -p 51413:51413/udp --network dockerNet -v /srv/dockers/transmission/config:/config -v
```

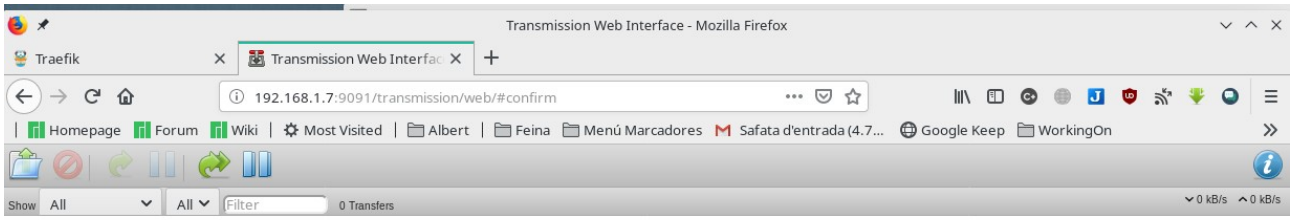
/srv/dockers/transmission/watch:/watch -v /srv/shared/downloads:/downloads -e PUID=1500 -e PGID=1500 -e TZ=Europe/Madrid linuxserver/transmission

En aquest cas ja que no es publica amb traefik es podrà accedir a través de la ip del host:9091 o bé des de la xarxa virtual interna dockerNet transmission:9091.

Un cop instal·lat el contenidor es pot accedir a la configuració del mateix que està desada a /srv/dockers/transmission/config/settings.json on podrem configurar un usuari i password per la autenticació.

arrancadaarem el contenidor com és habitual executant:

docker start transmission



Si instal·lem un plugin del navegador anomenat torrent control i es configura amb les dades indicades es poden afegir els torrents directament des del navegador a la cua de descàrregues.

4.4.2 Instal·lació, configuració i proves del contenidor de sessions remotes: x2go

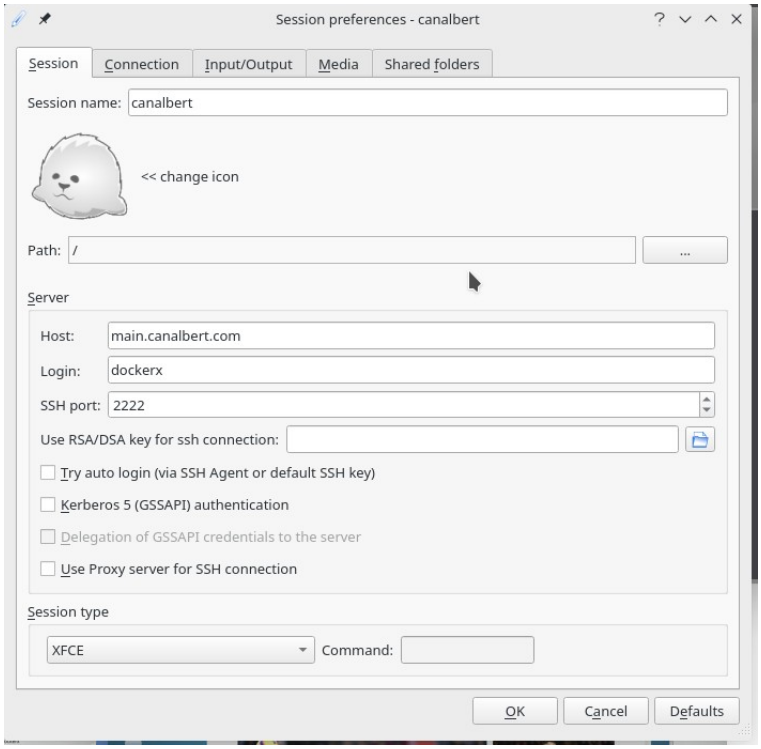
Aquest contenidor no necessita guardar cap dada ni configuració i per tant no necessita tampoc cap usuari amb permisos. La crida per generar el contenidor és la següent:

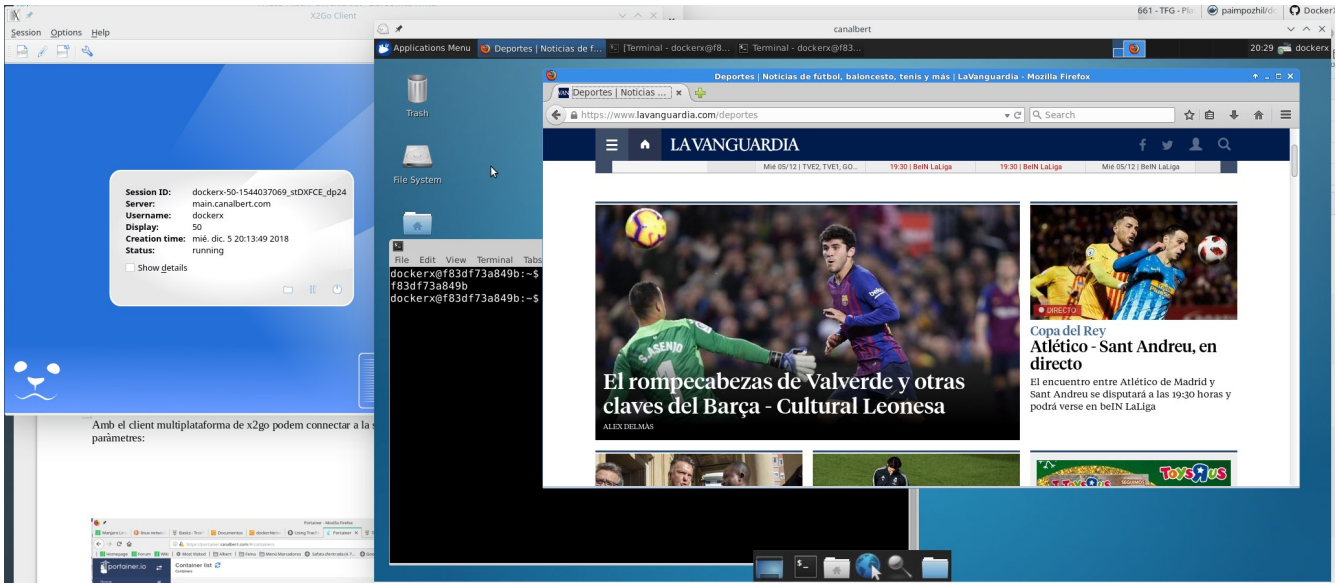
docker create --name=x2go -p 2222:22 -e TZ=Europe/Madrid --network dockerNet paimpozhil/docker-x2go-xubuntu

En aquest cas podem destacar que es publica el port 2222 del host i es mapeja amb el port 22 (SSH) del contenidor i que hi ha diferents tipus de sabors com imatge base del contenidor: xubuntu, xlde, etc.. Engeguem el contenidor com sempre executant:

docker start x2go

La configuració d'autenticació es genera automàticament i s'ha de consultar el log del contenidor per saber el password inicial, després es pot canviar. Per connectar amb el servidor cal fer servir el client multiplataforma de x2go i que es pot trobar a https://wiki.x2go.org/doku.php/download:start i configurant els següents paràmetres:





4.5 Configuració del servei d'administració de contenidors

4.5.1 Instal·lació i configuració del contenidor: Portainer

Per la instal·lació d'aquest contenidor hem creat prèviament l'usuari portainer amb PUID=1509 i GUID=1509 i utilitzarem el directori /srv/dockers/portainer/data per desar la informació variable del contenidor bàsicament la configuració. La crida per generar el contenidor és la següent:

```
docker create --name=portainer -e PUID=1509 -e PGID=1509 -e TZ=Europe/Madrid -v /var/run/docker.sock:/var/run/docker.sock -v /srv/dockers/portainer/data:/data --network traefikNet -l "traefik.enable=true" -l "traefik.frontend.rule=Host:portainer.canalbert.com" -l "traefik.port=9000" -l "traefik.docker.network=traefikNet" portainer/portainer
```

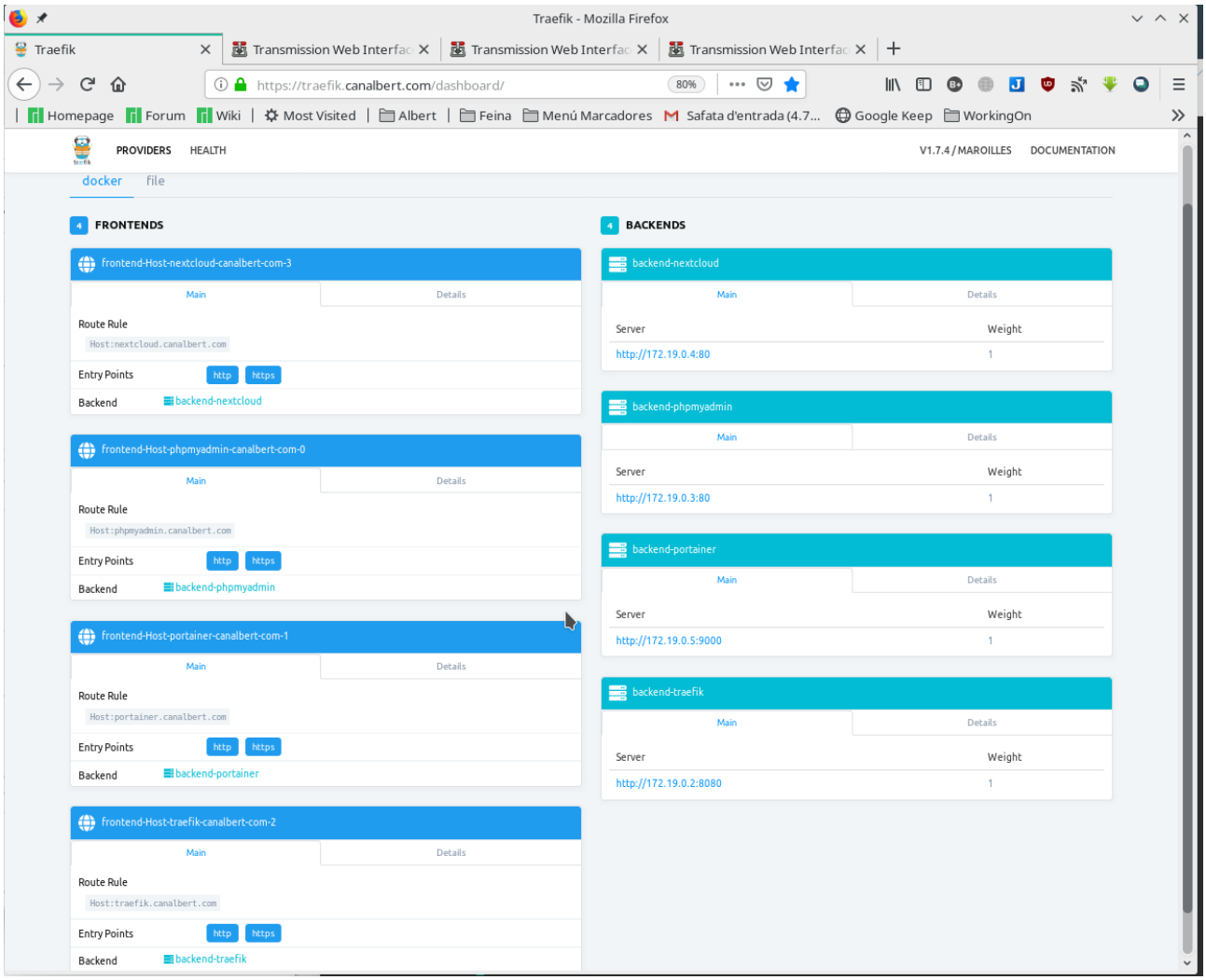
Únicament destacar que es dóna accés al directori /var/run/docker.sock perquè el contenidor tingui accés al End Point de docker.

En primer cop que s'accedeix a l'aplicació web et demana unes credencials per l'usuari administrador i la connexió al End Point local per connectar amb el servei de docker. arrancadaarem el contenidor executant:

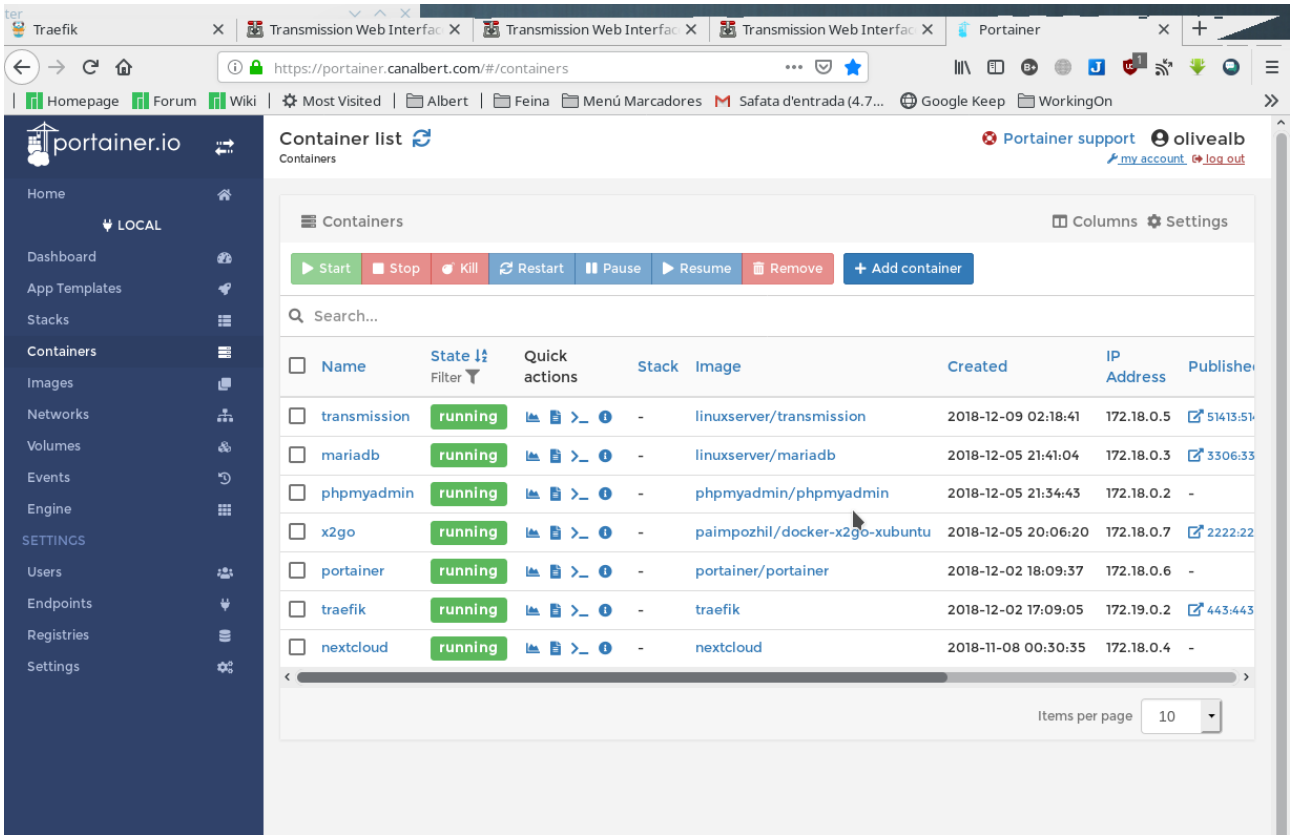
```
docker start portainer
```

4.5.2 Proves de connexió i redirecció

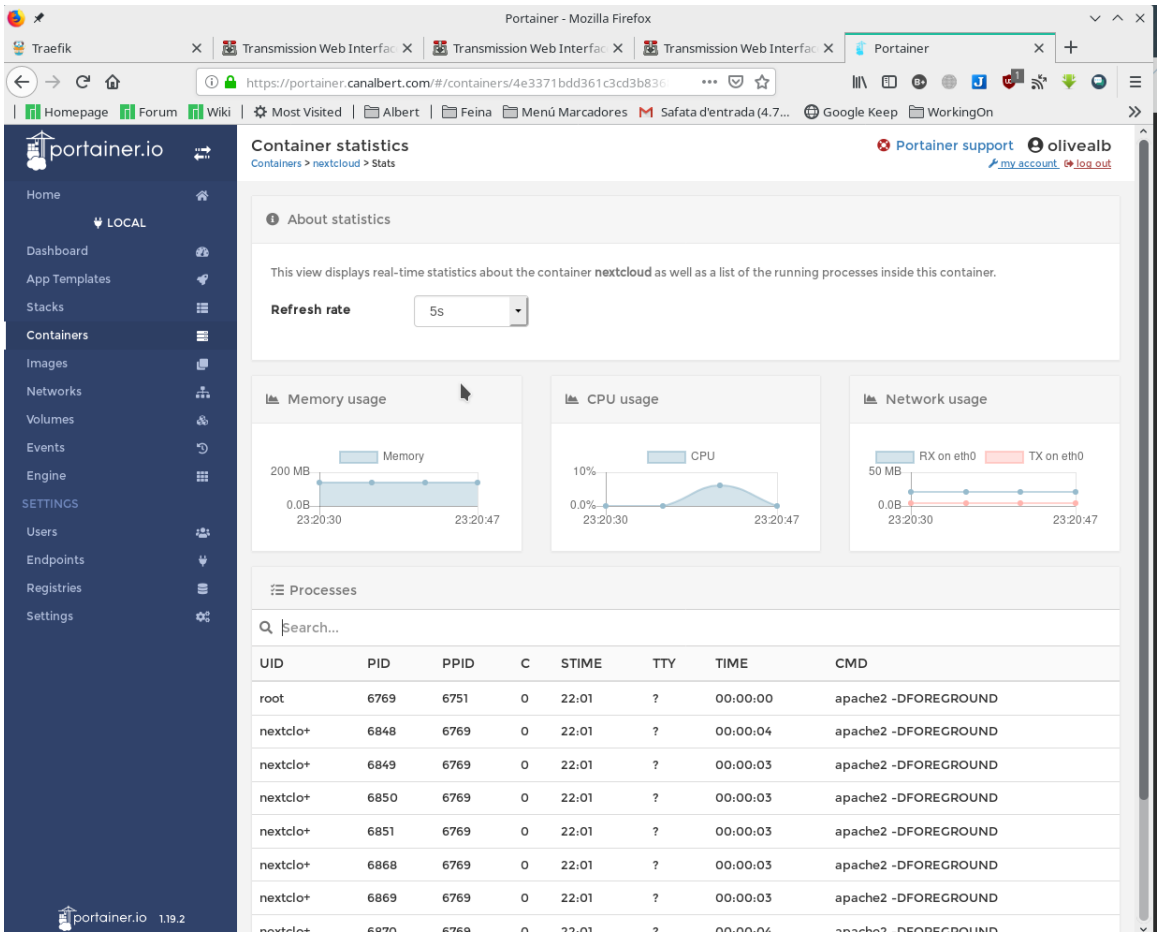
Com en anteriors ocasions primer comprovarem que traefik ha aixecat el nou frontal per aquèsta aplicació:



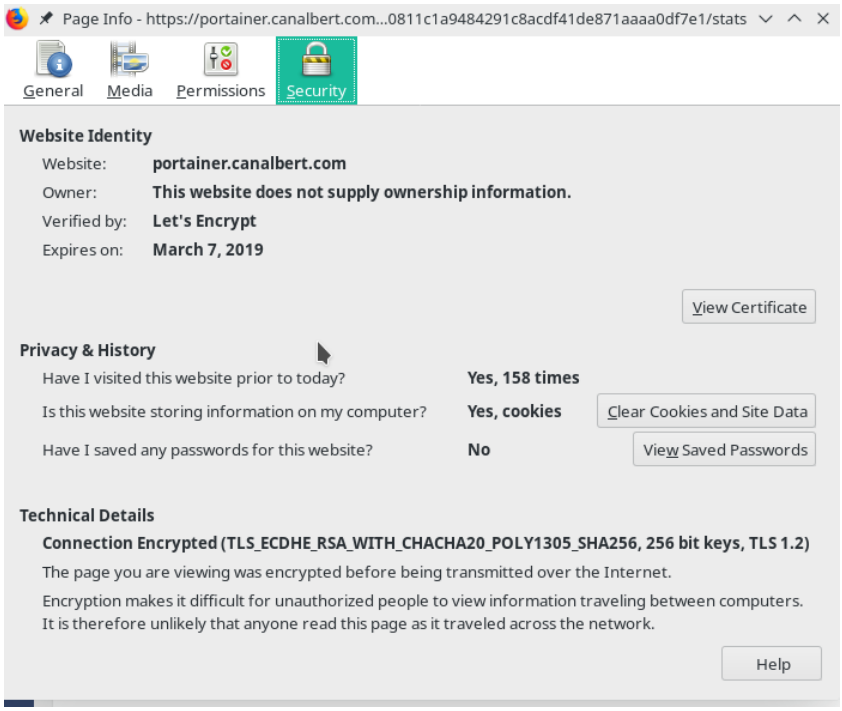
i la connexió accedint a la url: https://portainer.canalbert.com



Amb aquesta aplicació via web es poden administrar els contenidors i monitoritzar mínimament el seu estat, consum de recursos, etc..



I per últim comprovar el certificat digital:

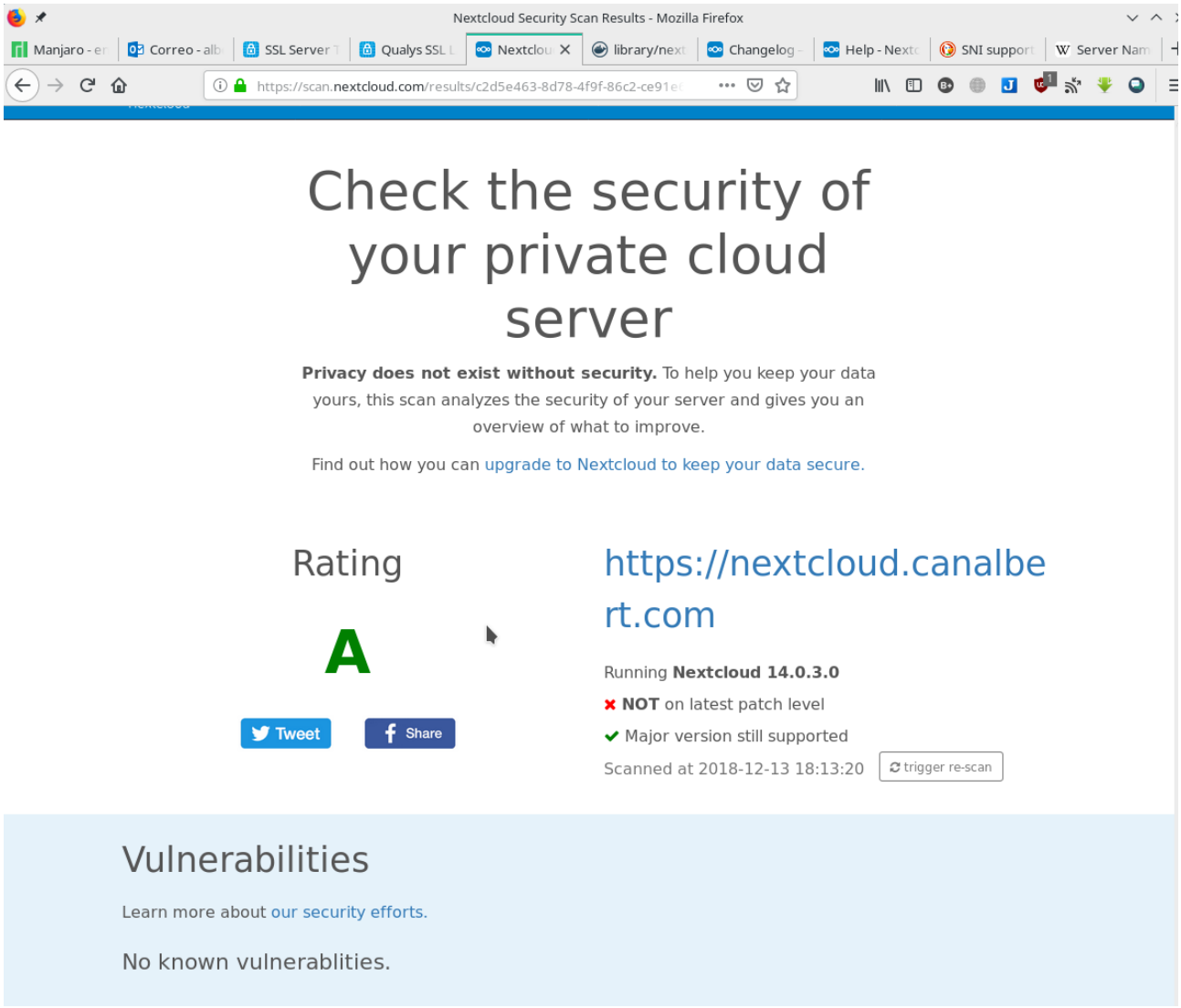


5 TEST DE SEGURETAT

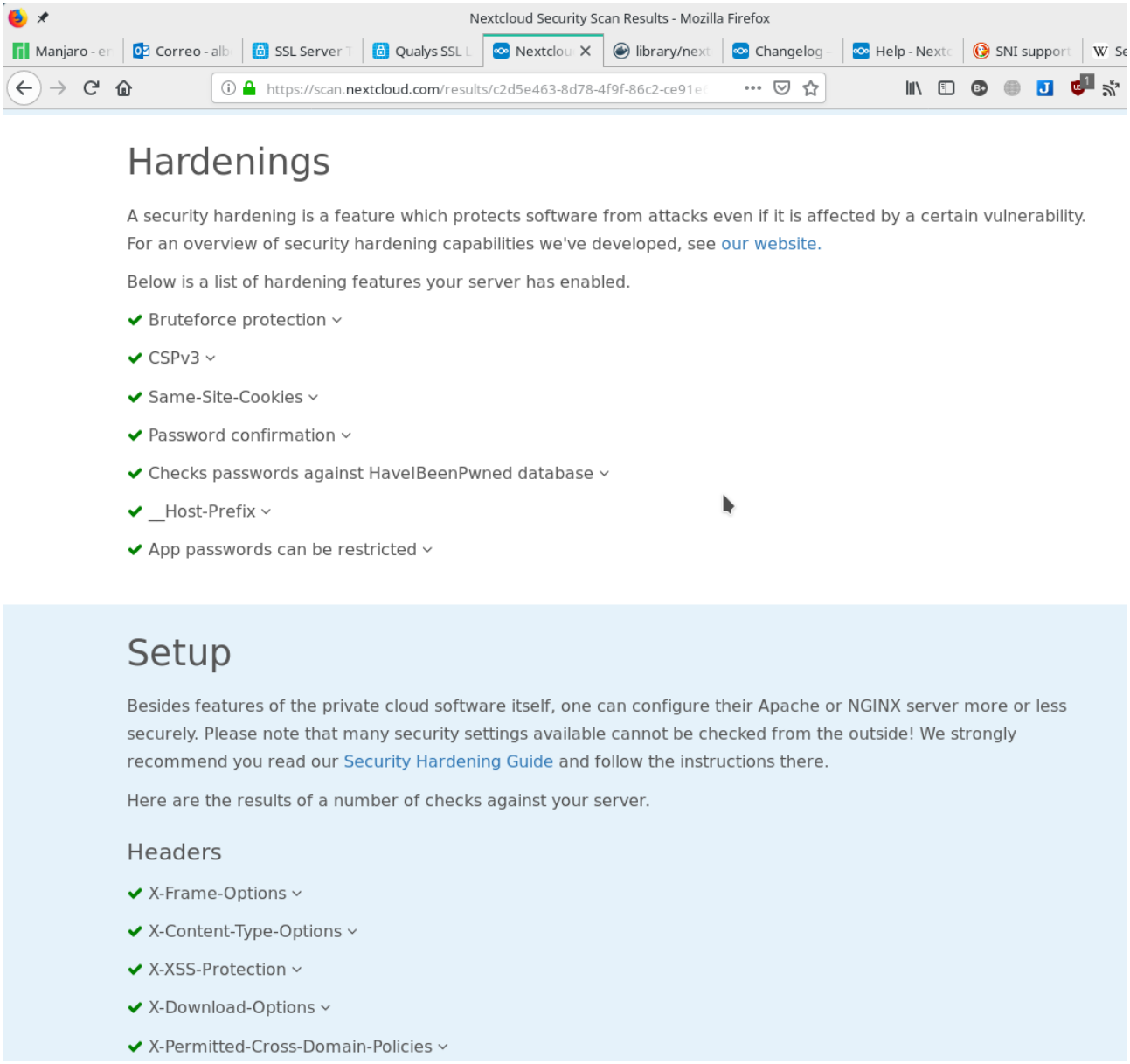
Un cop finalitzada la implementació de la infraestructura cal comprovar que aquesta asoleix uns nivells de seguretat acceptables cumplint així amb els criteris d'acceptació del projecte.

5.1 Nextcloud Security Scan

Nextcloud compromesa amb la seguretat del producte i dels usuaris, ofereix un check de seguretat del seu producte https://scan.nextcloud.com/ .

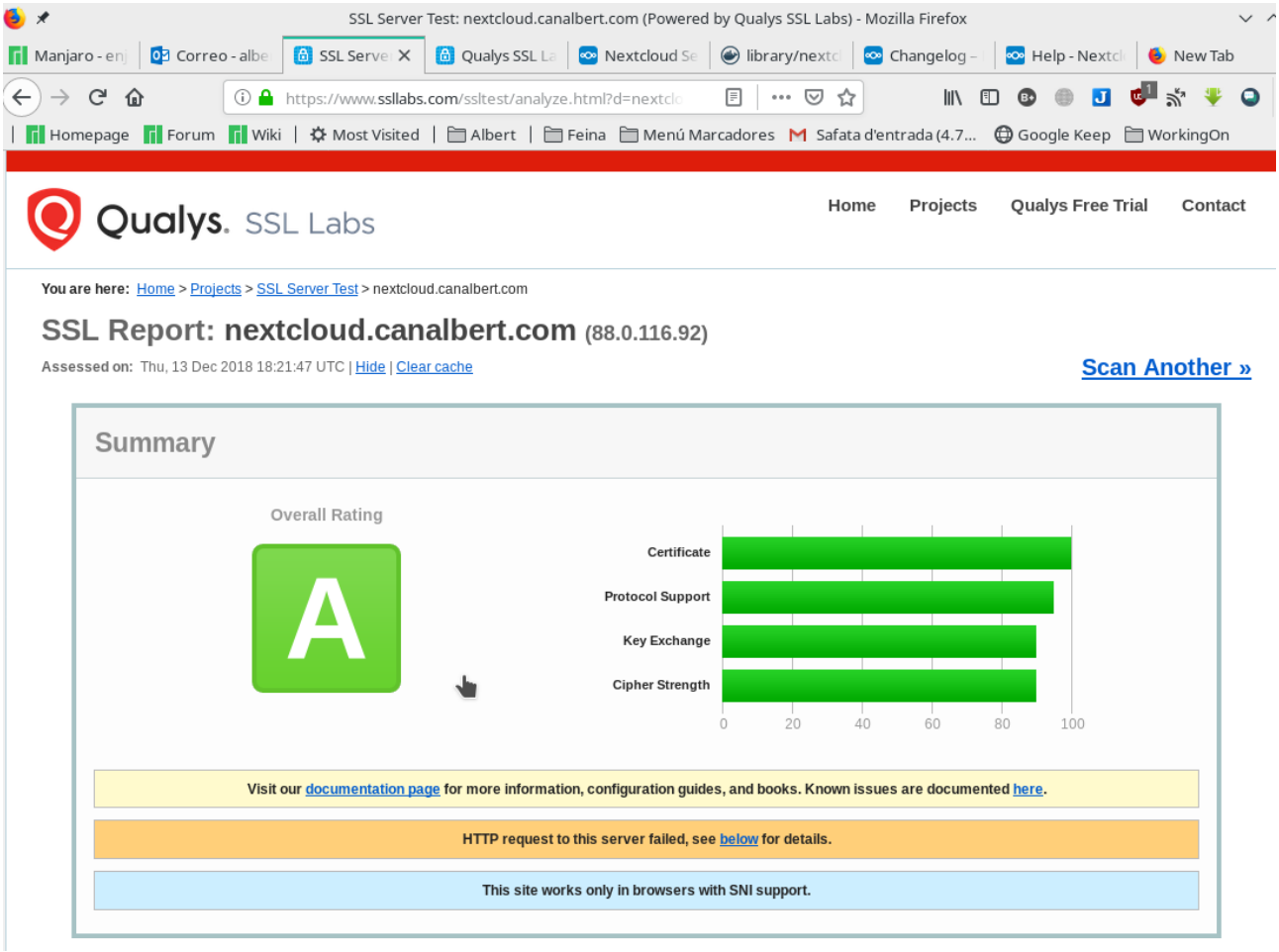


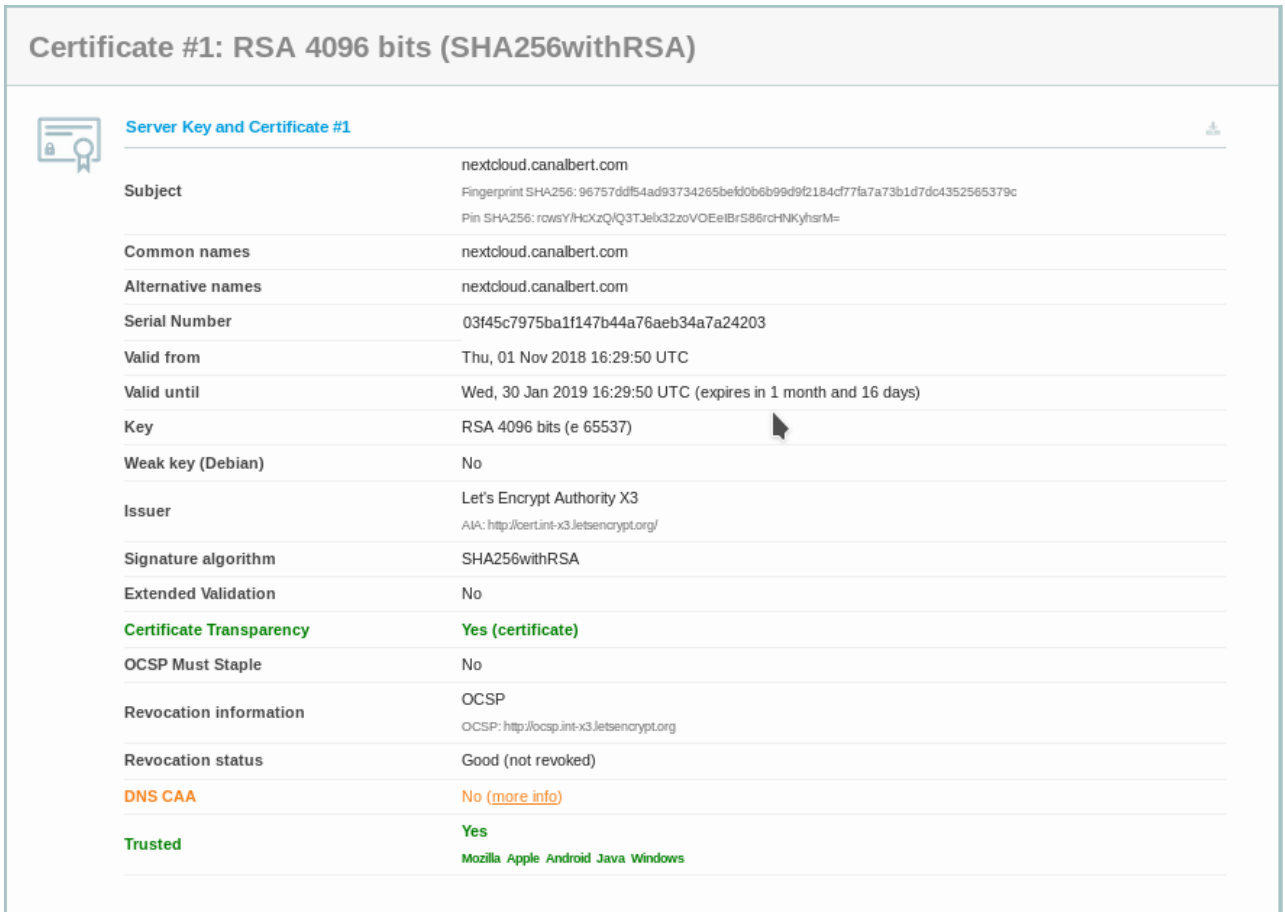
Podem observar que el check de seguretat adverteix que existeix una versió superior de Nextcloud, la 15.0.0, que va alliberar-se el 10 de Decembre, però la versió 14.0.3.0 està totalment soportada tal i com s'indica a https://nextcloud.com/changelog/ . En tot cas, actualitzar la versió de nextcloud costa ben poc com veurem més endavant.



5.2 Qualys SSL Labs

SSL Labs és una organització dedicada a la correcta implementació i millora del protocol SSL i també ofereix un servei d'escaneig gratuït de hostnames.


SSL Server Test: nextcloud.canalbert.com (Powered by Qualys SSL Labs) - Mozilla Firefox
https://www.ssllabs.com/ssltest/analyze.html?d=nextcl
Qualys. SSL Labs
Home
Projects
Qualys Free Trial
Contact
You are here: Home > Projects > SSL Server Test > nextcloud.canalbert.com
SSL Report: nextcloud.canalbert.com (88.0.116.92)
Assessed on: Thu, 13 Dec 2018 18:21:47 UTC | Hide | Clear cache
Scan Another »
Summary
Overall Rating
A
Certificate
Protocol Support
Key Exchange
Cipher Strength
0
20
40
60
80
100
Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.
HTTP request to this server failed, see below for details.
This site works only in browsers with SNI support.



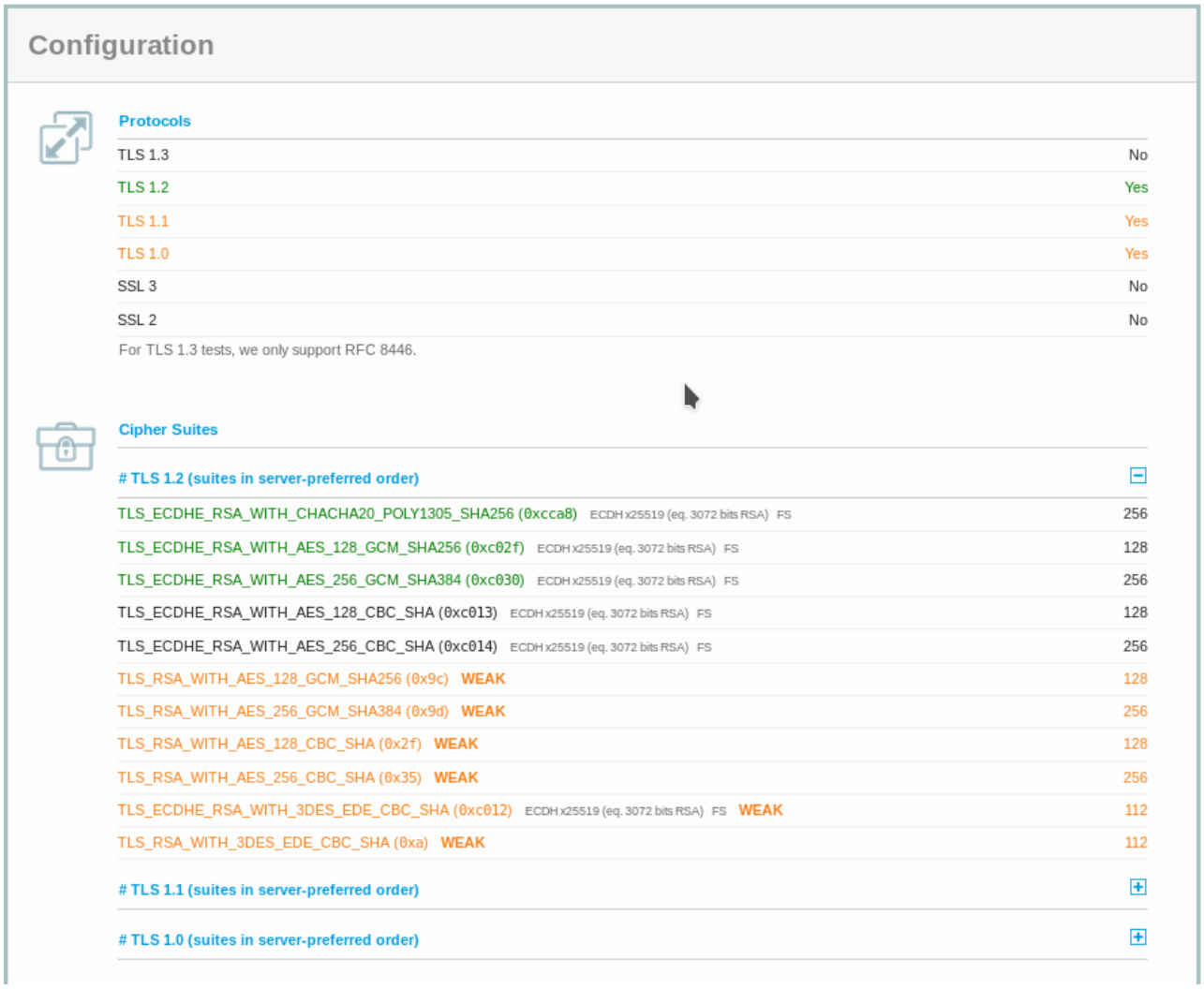
És important remarcar que com indica l'scan, cal que el navegador suporti SNI (Server Name Indication) perquè la web funcioni de forma segura, el motiu és que aquesta extensió del protocol TLS permet precisament presentar múltiples certificats sobre la mateixa IP i port, que és exactament el que estem fent nosaltres.

Exemples de programari antic que **NO** soporta SNI:

Android 2.3.7 i inferior

IE 8/6 - Windows XP

Java 6u45 i inferior



Configuration

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we only support RFC 8446.

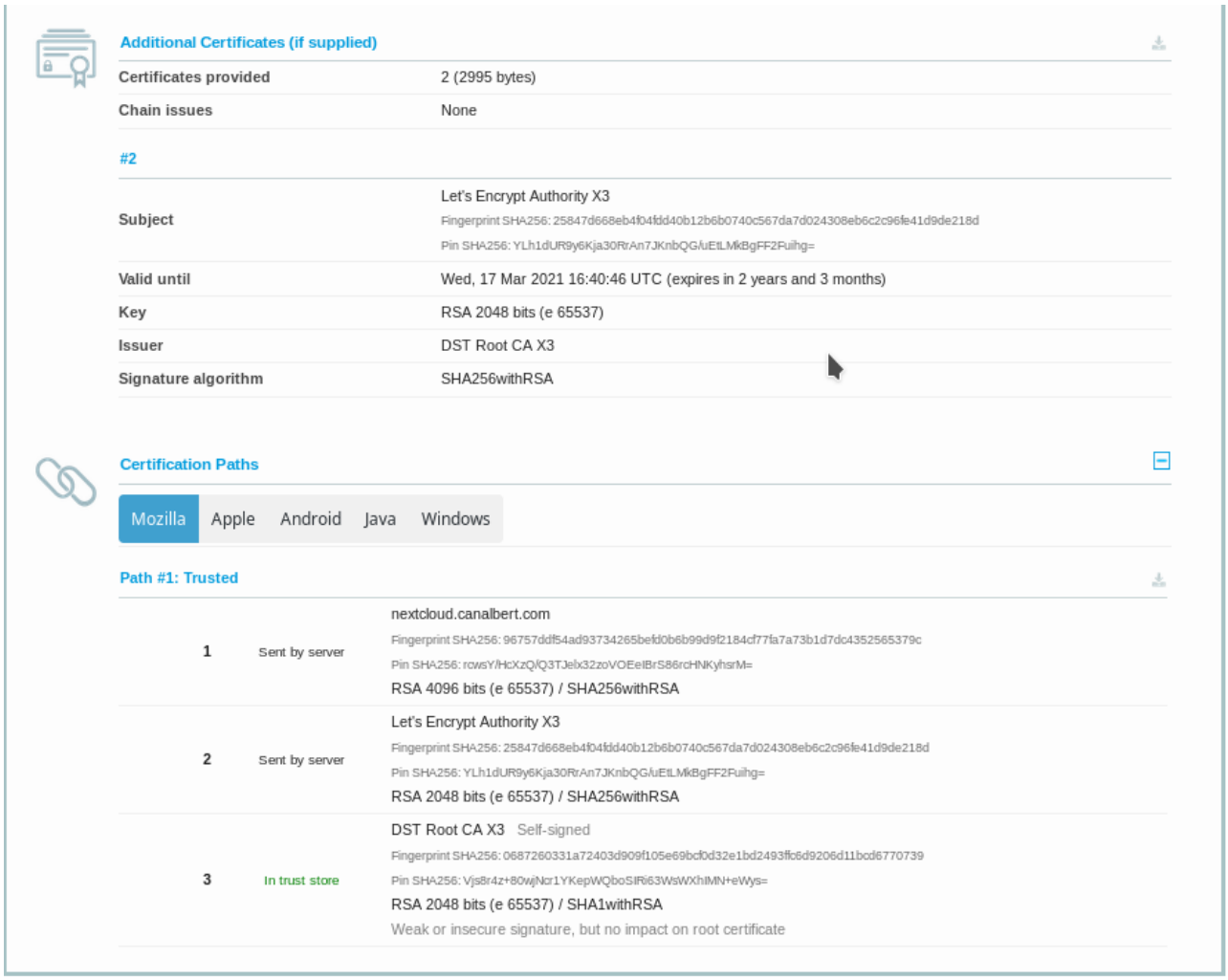
Cipher Suites

TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK		128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK		256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK		128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK		256
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012)	ECDH x25519 (eq. 3072 bits RSA) FS WEAK	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK		112

TLS 1.1 (suites in server-preferred order)

TLS 1.0 (suites in server-preferred order)


Additional Certificates (if supplied)
Certificates provided
2 (2995 bytes)
Chain issues
None
#2
Subject
Let's Encrypt Authority X3
Fingerprint SHA256: 25847d668eb4f04fdd40b12b6b0740c567da7d024308eb6c2c96fe41d9de218d
Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
Valid until
Wed, 17 Mar 2021 16:40:46 UTC (expires in 2 years and 3 months)
Key
RSA 2048 bits (e 65537)
Issuer
DST Root CA X3
Signature algorithm
SHA256withRSA
Certification Paths
Mozilla
Apple
Android
Java
Windows
Path #1: Trusted
1
Sent by server
nextcloud.canalbert.com
Fingerprint SHA256: 96757ddf54ad93734265befd0b6b99d9f2184cf77fa7a73b1d7dc4352565379c
Pin SHA256: rcwsY/HcXzQ/Q3TJelx32zoVOEeIBrS86rcHNKyhsrM=
RSA 4096 bits (e 65537) / SHA256withRSA
2
Sent by server
Let's Encrypt Authority X3
Fingerprint SHA256: 25847d668eb4f04fdd40b12b6b0740c567da7d024308eb6c2c96fe41d9de218d
Pin SHA256: YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=
RSA 2048 bits (e 65537) / SHA256withRSA
3
In trust store
DST Root CA X3 Self-signed
Fingerprint SHA256: 0687260331a72403d909f105e69bcf0d32e1bd2493ffc6d9206d11bcd6770739
Pin SHA256: Vjs8r4z+80wjNcr1YKepWQboSIRi63WsWXhIMN+eWys=
RSA 2048 bits (e 65537) / SHA1withRSA
Weak or insecure signature, but no impact on root certificate

6 MANTENIMENT DE LA INFRAESTRUCTURA

Aquest és un dels punts més importants dins d'una infraestructura, el manteniment i l'actualització de versions, la màxima que sovint s'aplica en moltes organitzacions de:

«Si funciona, no ho toquis» és totalment falsa, és crític actualitzar les versions i aplicar els pegats de seguretat i no fer-ho suposa un risc molt important perquè tard o d'hora deixarà de funcionar. També cal seguir sempre un procediment segur que permeti una marxa enrere en cas necessari i aquí intervé l'altre gran puntal per mantenir la infraestructura estable i segura, les còpies de seguretat.

6.1 Actualització de la infraestructura

Com hem vist fins ara, la infraestructura del projecte consta de diferents capes, el Sistema Operatiu, els contenidors i les aplicacions dins dels contenidors. A continuació veurem com realitzar les actualitzacions que afecten en cada nivell.

6.1.1 Actualització Sistema Operatiu

Com en qualsevol instal·lació linux per upgradejar el sistema i les aplicacions cal utilitzar el programari de paqueteria propi, Centos utilitza YUM, per tant per actualitzar el sistema caldrà executar:

```
sudo yum update
```

6.1.2 Actualització dels contenidors

En aquest document s'ha fet esment en diverses ocasions de la immutabilitat dels contenidors i que per desar canvis cal fer-ho en un sistema de fitxers del host, recordem que diversos contenidors munten directoris del host per guardar configuracions i dades. Doncs aquesta és una de les grans avantatges a l'hora d'actualitzar els contenidors i les aplicacions que hi corren, atès que l'únic que cal fer és aturar el contenidor, eliminar-lo i tot seguit esborrar la imatge que es va baixar quan es va generar. Un cop realitzat, el tornem a crear amb els mateixos paràmetres, i com que s'ha eliminat la imatge inicial, docker anirà un altre cop al repositori d'imatges i es baixarà la darrera versió, recordem que si no indiquem el contrari amb un TAG, baixa la darrera versió (latests version), un cop arracat el contenidor recuperarà les configuracions i les dades que estaven desades en el directori del sistema i funcionarà amb tota normalitat.

Veiem un exemple, abans hem vist que el contenidor del servei de nextcloud es podia actualitzar a la versió 15.0.0, doncs per actualitzar-lo executaríem el següent:

```
docker stop nextcloud
```

```
docker rm nextcloud
```

```
docker rmi dbcf87f7f289
```

(abans hauriem de d'executar un `docker image ls` per llistar les imatges)

```
docker create --name=nextcloud -e PUID=1504 -e PGID=1504 -v /srv/dockers/nextcloud/config:/var/www/html/config -v /srv/dockers/nextcloud/data:/var/www/html/data --network traefikNet -l "traefik.enable=true" -l "traefik.frontend.rule=Host:nextcloud.canalbert.com" -l "traefik.port=80" -l "traefik.docker.network=traefikNet" nextcloud
```

i llestos, ja està el contenidor actualitzat.

Òbviament primer caldria fer una còpia de seguretat de les dades que es guarden en el directori del host que utilitza nexcloud i de la base de dades mariaDB per si cal tornar enrere.

6.1.3 Manteniment de les aplicacions i troubleshoting

En algún moment podria ser necessari fer una analisi del funcionament d'alguna aplicació per algún problema puntual, en aquest cas es pot accedir al log dels contenidors executant

```
docker logs <nom del contenidor>
```

També es pot interactuar amb els contenidors obrint una sessió interactiva de shell amb ells, executant:

```
docker exec -it <non del contenidor> bash
```

En el cas especial de nexcloud, és possible que sigui necessàri posar el servei en mode manteniment per realitzar operacions específiques amb la base de dades, executant:

```
docker exec -u www-data nextcloud php occ maintenance:mode --on
```

```
docker exec -u www-data nextcloud php occ maintenance:mode --off
```

Com en qualsevol instal·lació normal.

6.2 Còpies de seguretat i restauració

Com hem vist en punts anteriors, només caldria fer còpia de seguretat de les dades muntades en el host, això representa també una gran avantatge atès que no cal accedir als contenidors per fer les còpies, fent una còpia des del sistema operatiu del host n'hi ha prou amb un tar/gz en un medi extern. Per mantenir la coherencia de les dades, es poden aturar els contenidor abans del realitzar la còpia o tirar d'snapshot del sistema de fitxers lvm.

Adicionalment seria molt aconsellable crear unes tasques de manteniment via cron per realitzar export periòdics de la base de dades de mariadb, recordem que com que vam publicar el port 3306 aquesta còpia es pot fer des de qualsevol dispositiu de la xarxa interna.

mysqldump -h 192.168.1.7 -u root -p <Nom_base_de _dades> > backup.sql

6.3 Aturada i arrancada de serveis (zero)

Docker permet establir unes polítiques perquè els contenidor s'arranquin automàticament en cas d'aturada sobtada del sistema, s'ha decidit no configurar aquesta política atès que és preferible en cas de problemes fer les comprovacions prèvies abans d'arrancar els serveis i també perquè existeix certa dependència entre contenidors i per tant un ordre d'aturada i d'arrancada determinat.

L'ordre d'aturada seria: x2go, transmission, phpmyadmin, nextcloud sense prioritats, després mariadb i portainer i per finalitzar traefik.

L'ordre d'arrancada seria l'invers a l'anterior, primer traefik, mariadb i portainer i després la resta.

Per aturar els contenidors es pot utilitzar el mateix servei de portainer excepte per aturar els serveis de traefik i d'ell mateix. L'alternativa és la connexió per SSH al host i aturar els contenidors amb línia de comandes:

```
docker ps -a
```

per llistar els contenidors i executar:

```
docker stop <nom_container>
```

en l'ordre indicat anteriorment.

Per arrancar els contenidors de traefik i portainer de la mateixa manera es pot executar:

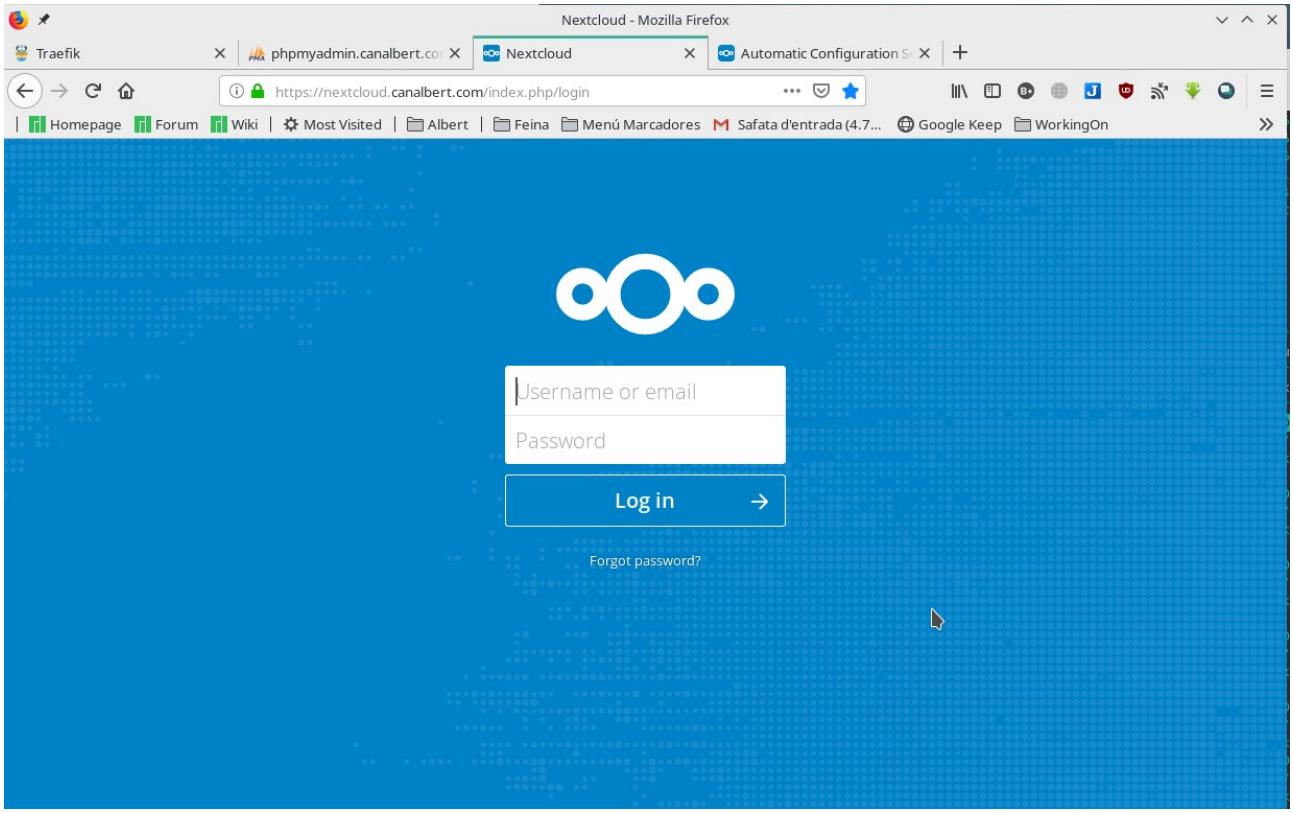
```
docker start <nom_container>
```

6.4 Configuració del clients

Seguidament una breu explicació sobre la configuració dels clients del servei de Nextcloud i mètodes d'accés al servei.

6.4.1 Accés web

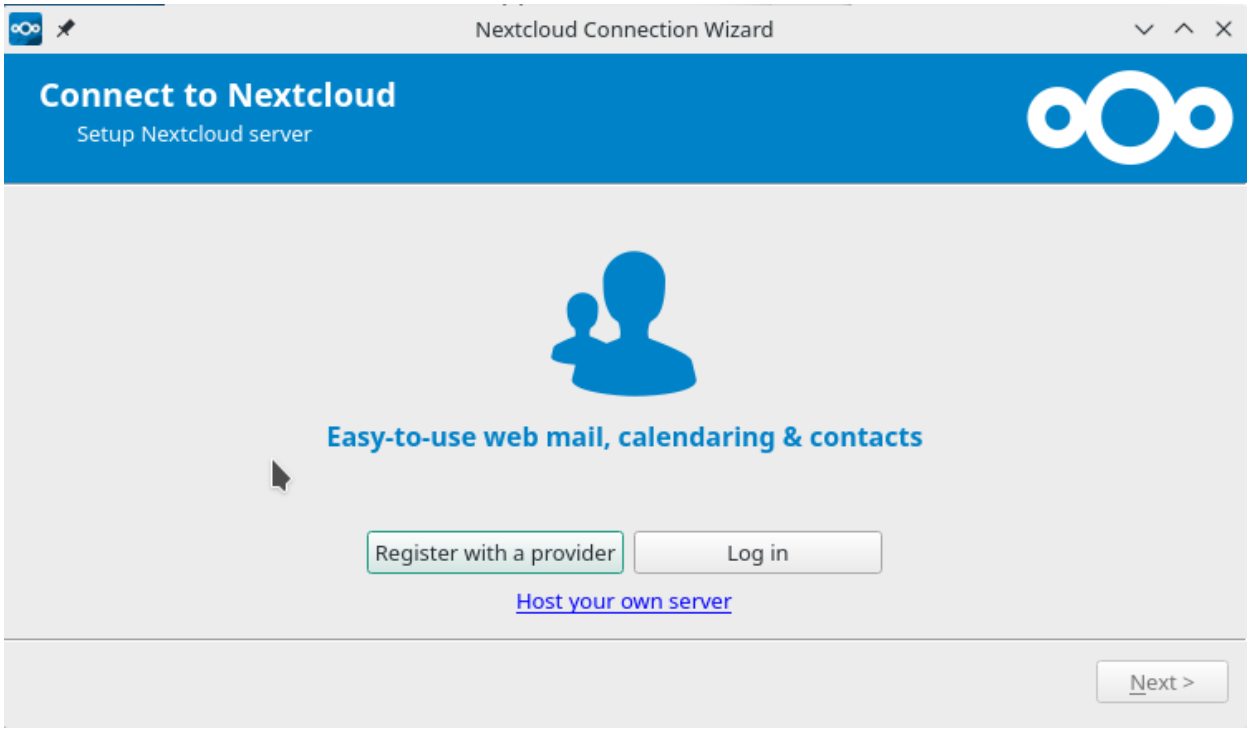
Per accedir per https als fitxers emmagatzemats en el núvol només cal utilitzar un navegador i accedir a l'URL https://nextcloud.canalbert.com introduint les credencials correctes.



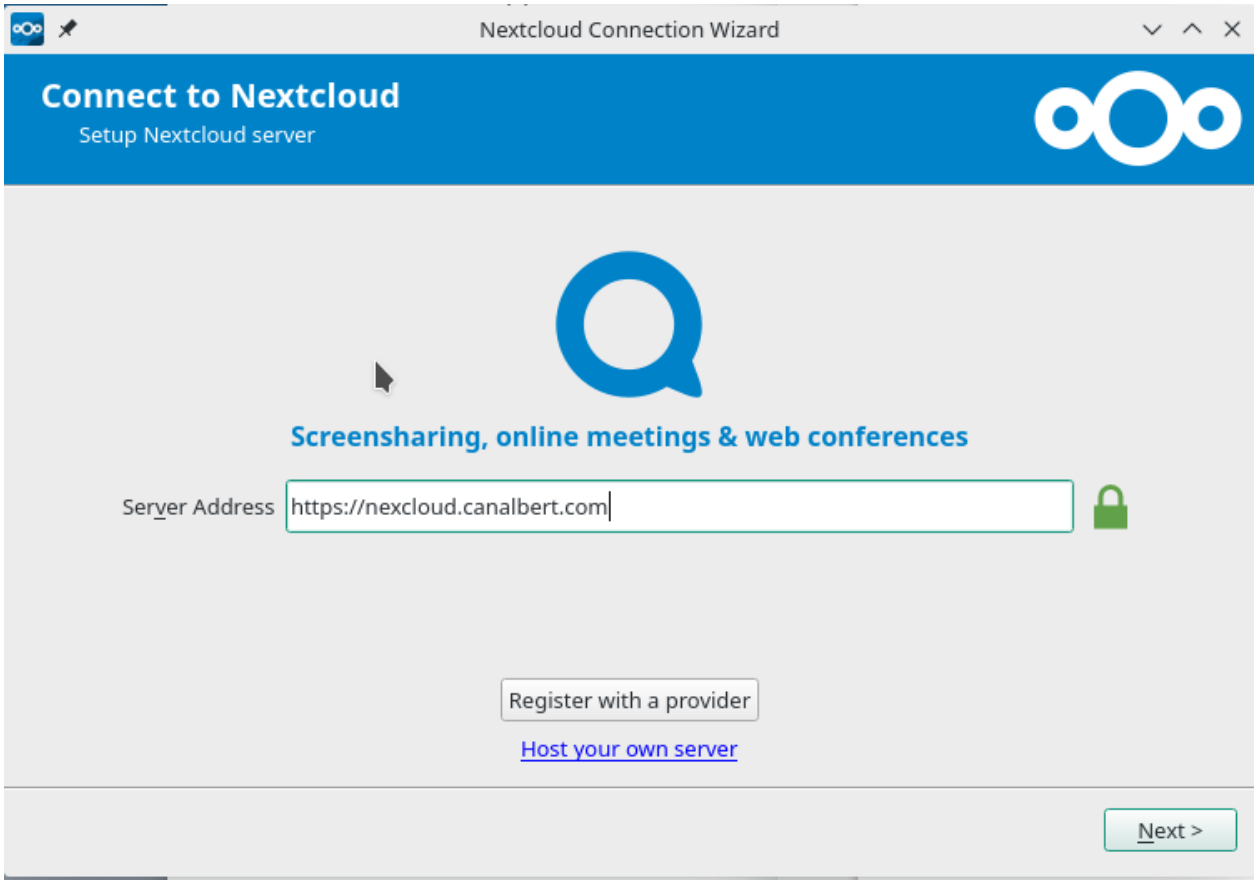
6.4.2 Client d'escriptori

Es pot accedir i sincronitzar selectivament els fitxers emmagatzemats en el núvol amb un directori local utilitzant el client multiplataforma (Windows/macOS/Linux) que podem obtenir de https://nextcloud.com/install/#install-clients dins de <Download for desktop>.

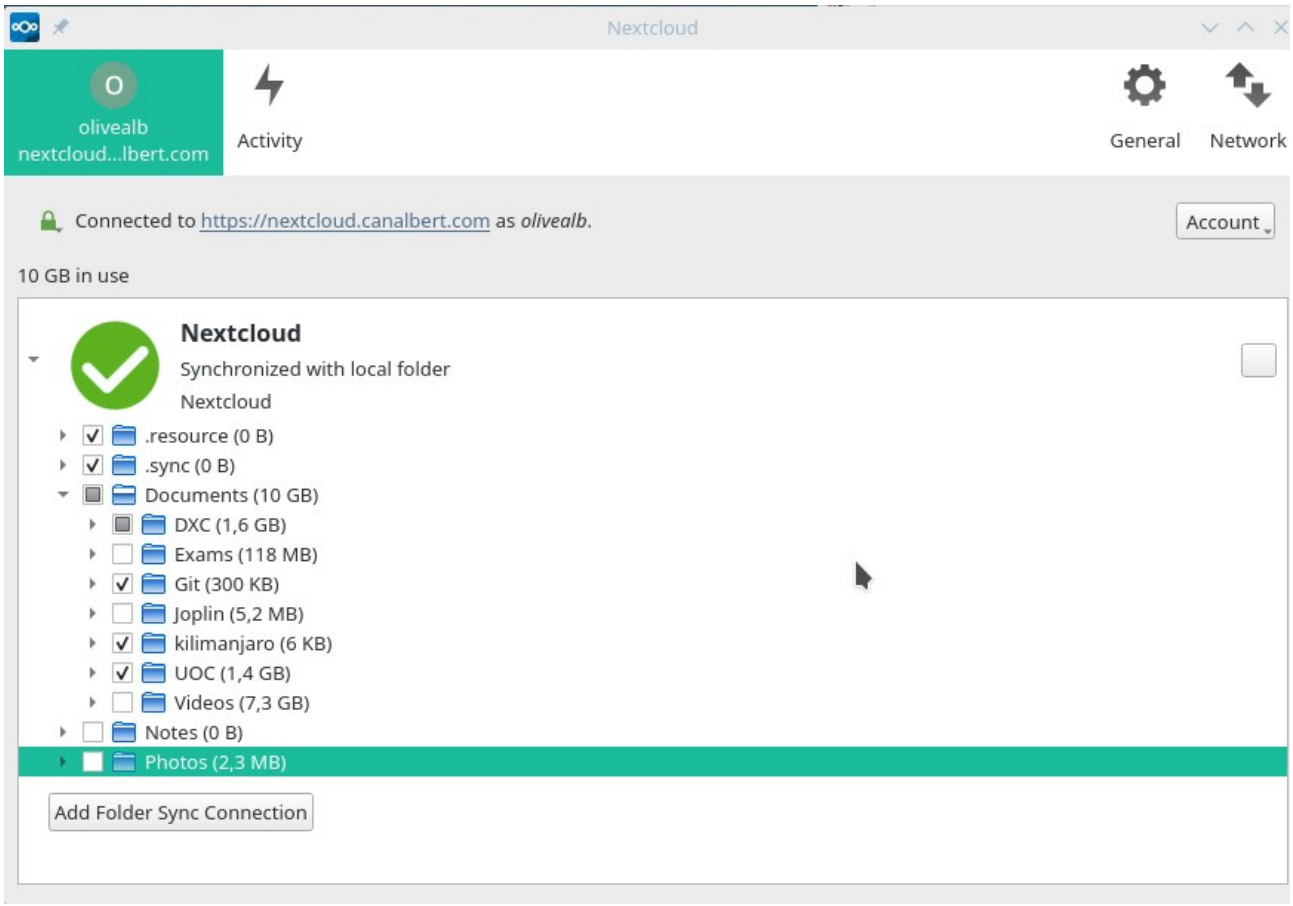
Un cop instal·lat el client, afegim un account:



Fem clic en Log in i informem el camp Server Address amb l'adreça del nostre servidor https://nextcloud.canalbert.com i indroduïm les credencials.



Seguidament només cal indicar quins directoris/fitxers volem sincronitzar i la carpeta local d'on penjaran:



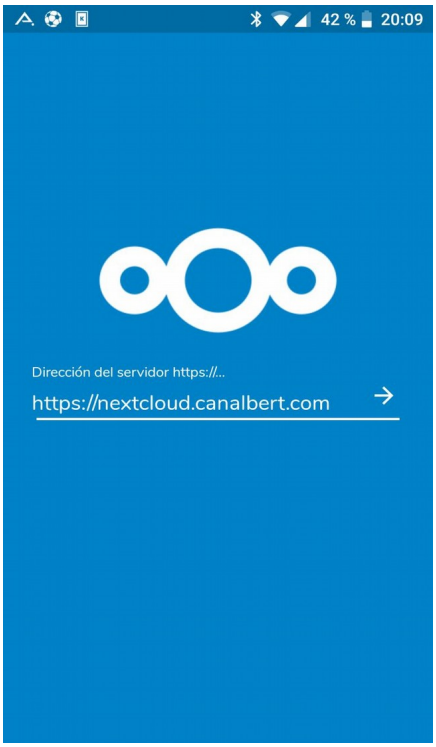
També es poden configurar filtres pel nom, extensió o mida dels fitxers que volem sincronitzar així com consultar l'activitat de sincronització i/o els possibles problemes.

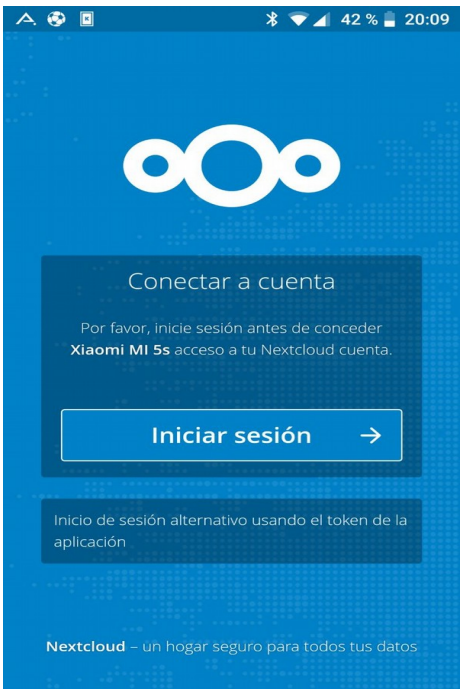
6.4.3 Client per dispositius mòbils

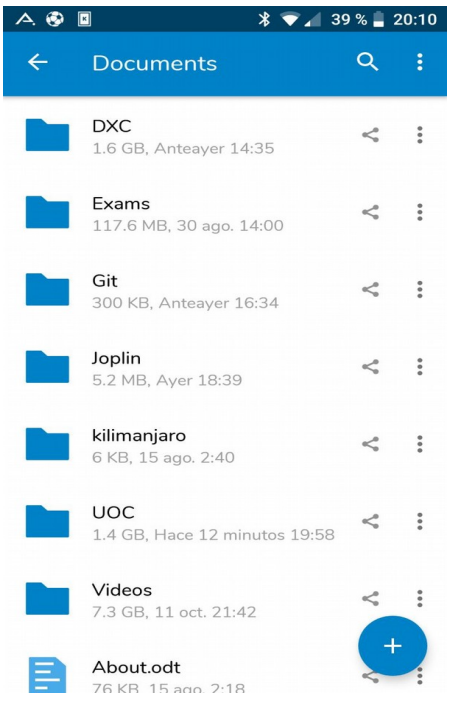
A la mateixa adreça web https://nextcloud.com/install/#install-clients dins de <Download for mobile> o també accedint als diferents stores: Google Play, AppStore o Windows Store i cercant l'aplicació Nexcloud.

La instal·lació és molt similar a les anteriors:


43 % 20:07
Mantén tus datos seguros y bajo tu control
INICIAR SESIÓN
REGISTRARSE CON UN PROVEEDOR
Aloja tu propio servidor


42 % 20:09
Dirección del servidor https://...
https://nextcloud.canalbert.com


42 % 20:09
Conectar a cuenta
Por favor, inicie sesión antes de conceder Xiaomi MI 5s acceso a tu Nextcloud cuenta.
Iniciar sesión
Inicio de sesión alternativo usando el token de la aplicación
Nextcloud – un hogar seguro para todos tus datos


39 % 20:10
Documents
DXC
1.6 GB, Anteayer 14:35
Exams
117.6 MB, 30 ago. 14:00
Git
300 KB, Anteayer 16:34
Joplin
5.2 MB, Ayer 18:39
kilimanjaro
6 KB, 15 ago. 2:40
UOC
1.4 GB, Hace 12 minutos 19:58
Videos
7.3 GB, 11 oct. 21:42
About.odt

6.4.4 Accés WEBDAV

Per finalitzar comentarem que també es pot accedir a les dades sense cap client específic, fent servir el protocol Webdav en qualsevol explorador de fitxers que tingui suport per aquest protocol introduint l'adreça i les credencials:

webdavs://nextcloud.canalbert.com:443/remote.php/webdav/Documents

7 MILLORES

En aquest punt es comentaran possibles millores de la infraestructura sota diferents criteris i altres possibles serveis per instal·lar.

7.1 Alta disponibilitat

Seguint un criteri de disponibilitat dels serveis seria molt interessant afegir redundància a la infraestructura més enllà de muntar discos en raid 1, simplement configurant un altre equip similar i utilitzant algún tipus de rèplica de fitxers entre els dos, un cluster de Ceph per exemple. El següent pas seria algún programari per gestionar el balanceig de contenidors entre diferents engines docker com swarm o kubernetes de forma que en cas de fallada d'un dels dos equips s'aixequessin en l'altre. Afegint un SAI, fàcilment podria ser una sol·lució per petites empreses.

7.2 Docker-compose

Les configuracions de contenidors realitzades en aquest projecte han estat individuals, s'han creat els contenidors un a un i s'han anat arrancant, però fent servir docker-compose es pot configurar en un sol firxer tota la configuració de contenidors i xarxes virtuals de forma que es pot desplegar i aturar tot l'entorn de cop i configurar múltiples entorns repetits per desenvolupament, preproducció i producció.

7.3 Altres serveis

Els serveis definits incialment pel projecte s'han implementat correctament però les possibilitat són enormes, es podrien haver implementat també molts altres com: servei de correu, servei git de control de versions, un servei plex per reproduir video en streamig, un servei de missatgeria online, un wordpress, les possibilitats són realment molt grans.

8 COSTOS D'IMPLEMENTACIÓ

A continuació es mostrarà una taula amb els costos aproximat d'implementació de la infraestructura asociada al projecte i en punts a part es valoraran els costos de planificació, documentació i maquinari.

8.1 Implementació

Els costos d'implementació per tasques són aproximadament els següents:

Tasques	Cost (en hores)
Instal·lació SO servidor i programari base	2
Configuració i preparació de l'emmagatzematge	5
Instal·lació del servei de containers virtuals	2
Configuració de les xarxes virtuals	1
Instal·lació i configuració del container redirector	2
Configuració del encaminador local i registre/publicació de noms del domini	5
Proves de connexió i redirecció	5
Configuració de serveis de suport	3
Configuració del servei d'emmagatzematge, sincronització i compartició	5
Configuració del servei de descàrregues	1
Configuració del servei de sessions remotes	2
Configuració del servei d'administració de containers	1
Aturada i arranc de serveis	3
Prova final de serveis	5
Tests de seguretat	1
Total:	43

8.2 Planificació i documentació

Els costos anteriors són estimats tenint en compte un disseny i planificació prèvia que òbviament també té un cost (al menys la primera vegada). Aquest cost és d'entre 32 i 40 hores.

La documentació d'una implementació (no unes memòries) seria de 8 hores més.

Per tant el cost total aproximadament seria de 80 hores.

8.3 Maquinari

El cost del maquinari utilitzat, sense comptar els elements de comunicacions com l'encaminador o el commutador atès que són elements que no són específics (tot i que sí necessàris) d'aquesta implementació, seria:

Element	Unitats	Preu unitari (IVA incl.)	Cost (IVA Incl.)
HP ProLiant MicroServer 8th Gen - Servidor (Intel Celeron G1610T Dual-Core a 2.3 GHz, 4GB de RAM)	1	199	199
Seagate Barracuda - Disco Duro Interno de 2 TB (3,5, 64 MB de caché SATA de 6 GB/s hasta 210 MB/s)	2	61,5	123
Kingston SSD A400 - Disco duro sólido, 2.5", SATA 3, 480 GB	2	64,9	129,8
			451,8 €

Esmentar que el micro servidor utilitzat està descatalogat en l'actualitat, els nous models de MicroServer 10th Gen es serveixen amb processadors AMD Opteron més potents (i més cars) però sense ILO, que és una característica que permet la connexió via xarxa a una cónsola de la BIOS, on podem apagar i engegar el servidor de forma remota, per exemple, o comprovar l'estat d'aquest, molt interessant per administració remota tot i que la llicència completa amb totes les característiques s'ha d'adquirir de forma separada a HP.

9 BIBLIOGRAFIA

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/logical_volume_manager_administration/LVM_components

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/logical_volume_manager_administration/lv_overview

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/logical_volume_manager_administration/raid_volumes

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/physvol_admin.html#physvol_create

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/VG_admin.html#VG_create

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/VG_grow.html

Product Documentation/RHEL. redhat, https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/

Getting docker (docker docs). Docker,https://docs.docker.com/install/linux/docker-ce/centos/

Manage networks (docker docs). Docker, https://docs.docker.com/network/

Manage application data (docker docs). Docker, https://docs.docker.com/storage/volumes/

Manage application data (docker docs). Docker, https://docs.docker.com/storage/bind-mounts/

Manage application data (docker docs). Docker, https://docs.docker.com/samples/library/nextcloud/

Traefik Reference (GitHub). Traefik, https://docs.traefik.io/basics/

Traefik Reference (GitHub). Traefik, https://docs.traefik.io/configuration/acme/

Traefik Reference (GitHub). Traefik, https://docs.traefik.io/configuration/entrypoints/

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/configuration_database/index.html

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/configuration_server/config_sample_php_parameters.html

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/maintenance/backup.html

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/maintenance/backup.html#maintenance-mode

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/maintenance/backup.html#backup-database

Docs support (GitHub). Nextcloud, https://docs.nextcloud.com/server/15/admin_manual/installation/source_installation.html

MariaDB Documentation. MariaDB, https://mariadb.com/kb/en/library/getting-installing-and-upgrading-mariadb/

MariaDB Documentation. MariaDB, https://mariadb.com/kb/en/library/backing-up-and-restoring-databases/

MariaDB Documentation. MariaDB, https://mariadb.com/kb/en/library/server-monitoring-logs/

Portainer Documentation (GitHub). Portainer, https://portainer.readthedocs.io/en/stable/configuration.html#admin-password

Portainer Documentation (GitHub). Portainer, https://portainer.readthedocs.io/en/stable/deployment.html#persist-portainer-data

Portainer Documentation (GitHub). Portainer, https://portainer.readthedocs.io/en/stable/deployment.html#secure-portainer-using-ssl

Documentation (Wiki). X2Go, https://wiki.x2go.org/doku.php/doc:installation:start

Documentation (Wiki). X2Go, https://wiki.x2go.org/doku.php/doc:de-compat

Public Community. Stack Overflow, https://stackoverflow.com/questions/51262403/dynamic-change-traefik-frontend-configuration-i-docker

Public Community. Stack Overflow, https://stackoverflow.com/questions/53845370/traefik-entire-wildcard-certificate-chain-configuration

HP's Generation 8 Microserver. Homeservershow.com, https://homeservershow.com/forums/forum/88-microserver-gen-8/

Multi HTTPS sub domain with Traefik and Docker, Patrik Cyvoct. https://blog.ptrk.io/multi-https-sub-domain-with-traefik-and-docker/

How to install Nextcloud on your server with Docker. Dwijadas Dey, https://blog.ssdnodes.com/blog/installing-nextcloud-docker/