



# Migració serveis de tercers d'una organització a servidors propis

**Mateu Saiós Jordana**

Grau en enginyeria informàtica

Àrea: GNU/Linux

**Tutor: Joaquin Lopez Sanchez-Montañes**

01/01/2024

## Fitxa del treball final

<b>Títol del treball:</b>	Migració serveis de tercers d'una organització a servidors propis
<b>Nom de l'autor:</b>	Mateu Saiós Jordana
<b>Nom del consultor/a:</b>	Joaquin Lopez Sanchez-Montañes
<b>Nom del PRA:</b>	Montse Serra Vizern
<b>Data d'entrega (mm/aaaa):</b>	01/2024
<b>Titulació:</b>	Grau en enginyeria informàtica
<b>Àrea del Treball Final:</b>	<i>GNU/Linux</i>
<b>Idioma del treball:</b>	Català
<b>Paraules clau</b>	Linux, Fedora, Servidor
<b>Resum del treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball.</i>	
<p>És tracta de crear un servidor per una petita organització dedicada a la creació de videojocs. L'objectiu és que no depengui de les grans corporacions. Instal·larem, configurarem i farem proves del funcionament de 6 serveis diferents:</p> <p>Un servidor web Apache que permetre accedir a la pàgina web de l'organització.</p> <p>Un servidor de fitxers Nextcloud per guardar diversos fitxers com clips dels videojocs, captures de pantalla, fulls de càlculs... També instal·larem un servidor Collabora per poder editar els fitxers directament des de Nextcloud.</p> <p>Un servidor git fent servir Forgejo. Aquest servidor ens servira per guardar els repositoris de l'organització. Per habilitar https, utilitzarem el servidor Apache com a proxy. Addicionalment, instal·larem un runner que ens permetrà executar accions, el que ens servira per construir la pàgina web després de cada 'push'.</p> <p>Un servidor de comunicació Prosody que ens permetrà enviar i rebre missatges a través del protocol xmpp. Per habilitar les trucades veu, farem servir un servidor coturn.</p> <p>Un servidor de notes Trilium en el qual podrem accedir i modificar les notes fàcilment, i tenir-les ben ordenades gràcies a la seva estructura jeràrquica.</p> <p>Per acabar, un panell de serveis Homepage per accedir a les interfícies web de la resta de serveis fàcilment.</p>	
<b>Abstract (in English, 250 words or less):</b>	

We are going to create a server for a small videogame company. The objective is to remove the dependency it currently has with big corporations. We will install, configure and test 6 different services:

An Apache web server that will allow us to access the organization's web page.

A Nextcloud file server to store various files like clips of the videogame, screenshots, spreadsheets... We will also install a Collabora server to edit the files directly from Nextcloud.

A git server using Forgejo. We will use this server to store the organization's repositories. To enable https, we will use the Apache server as a proxy. Additionally, we are going to install a runner that will allow us to execute actions, which we will use to build the webpage after each push.

A Prosody communication server to send and receive messages using the xmpp protocol. To enable voice calls, we will use a coturn server.

A Trilium notes server in which we will be able to easily access and modify the notes, and have them organized thanks to its geriatric structure.

Finally, a Homepage dashboard to easily access the web interfaces of the previous services.

# Índex

Fitxa del treball final.....	2
Índex.....	4
Índex de Figuras.....	6
Introducció.....	8
Context i justificació.....	8
Objectius.....	8
Mètode.....	8
Planificació del treball.....	8
Resum del resultat final.....	9
Resum de les seccions.....	9
Web.....	9
Fitxers.....	9
Git.....	10
Comunicació.....	10
Notes.....	10
Panell de serveis.....	10
1 Web.....	11
Opcions disponibles.....	11
Algernon.....	11
Apache.....	11
Nginx.....	11
Instal·lació i configuració.....	11
Migració.....	12
2 Fitxers.....	15
Opcions disponibles.....	15
Nextcloud.....	15
OwnCloud.....	15
Seafile.....	15
Instal·lació i configuració.....	15
Migració.....	24
3 Git.....	28
Opcions disponibles.....	28
Gogs.....	28
Gitea.....	28
Forgejo.....	28
OneDev.....	29
Instal·lació i configuració.....	29
Base.....	29
HTTPS.....	33
Accions.....	34
Migració.....	37
Repositoris.....	37
Accions.....	41
Test.....	42
Accions.....	44
4 Comunicació.....	46

Estàndards disponibles.....	46
XMPP.....	46
Matrix.....	46
Servidors XMPP.....	47
Instal·lació i configuració.....	48
Instal·lació coturn i integració amb Prosody.....	50
Posada en marxa.....	50
Tests.....	51
5 Notes.....	53
Opcions disponibles.....	53
Joplin.....	53
Nextcloud Notes.....	53
Trilium.....	54
Instal·lació i configuració.....	54
HTTPS.....	56
Migració.....	57
6 Panell de serveis.....	59
Opcions disponibles.....	59
Dashy.....	59
Homepage.....	59
Homarr.....	60
Instal·lació i configuració.....	60
Serveis.....	62
Forgejo.....	62
Nextcloud.....	63
Altres.....	64
Preferits.....	64
Altres.....	65
7 Conclusió.....	66
Glossari.....	68
Bibliografia.....	69
Web.....	69
Fitxers.....	69
Git.....	69
Comunicació.....	70
Notes.....	70
Panell de serveis.....	70

## Índex de Figuras

Figura 1: Diagrama Gantt del projecte.....	9
Figura 2: Pàgina benvinguda servidor Apache.....	12
Figura 3: Pàgina web Weküfu Studios al servidor Apache.....	14
Figura 4: Nextcloud no pot escriure al fitxer de configuració.....	17
Figura 5: Pàgina web d'instal·lació Nextcloud.....	18
Figura 6: Configuració base de dades Nextcloud.....	18
Figura 7: Pàgina d'inici de sessió Nextcloud.....	19
Figura 8: Pàgina inicial Nextcloud.....	19
Figura 9: Fitxers Nextcloud.....	20
Figura 10: Errors Nextcloud.....	20
Figura 11: Errors logging Nextcloud.....	21
Figura 12: Instal·lació servidor Collabora.....	22
Figura 13: Fitxer Invoice.odt obert fent servir el servidor Collabora.....	23
Figura 14: Pàgina domini desconegut Nextcloud.....	23
Figura 15: Pàgina login Nextcloud fent servir ip pública.....	24
Figura 16: Grup Nextcloud amb els membres de Weküfu Studios.....	25
Figura 17: Configuració carpeta compartida del grup 'Weküfu Studios'.....	25
Figura 18: Fitxers al Google Drive.....	26
Figura 19: Carpeta 'Weküfu Studios' amb els fitxers del Drive comprimits.....	26
Figura 20: Sincronitzant carpeta 'Weküfu Studios' amb Nextcloud.....	26
Figura 21: Els dos usuaris del grup 'Weküfu Studios' comparteixen la carpeta 'Weküfu Studios'.....	27
Figura 22: Configuració inicial Forgejo.....	31
Figura 23: Creació primer usuari Forgejo.....	32
Figura 24: Usuari Forgejo.....	32
Figura 25: Forgejo sense botó de registrar.....	33
Figura 26: Forgejo https.....	34
Figura 27: Administració runners Forgejo.....	37
Figura 28: Crear organització Forgejo.....	37
Figura 29: Migrant repositori a Forgejo.....	38
Figura 30: Repositori després de migrar-lo a Forgejo.....	39
Figura 31: Generar Token a GitHub.....	39
Figura 32: Repositori privat GitHub.....	40
Figura 33: Repositori privat després de migrar-lo de GitHub.....	41
Figura 34: Activem accions al repositori.....	41
Figura 35: Commit de prova al repositori Godot4RoguelikeProject.....	43
Figura 36: Resultat d'executar l'acció.....	45
Figura 37: Branca de la pàgina web sobreescrita.....	45
Figura 38: Grup de Discord amb diversos subgrups.....	47
Figura 39: Característiques Prosody IM.....	47
Figura 40: Afegir comte a Pidgin.....	51
Figura 41: Usuari mateu envia missatge a usuari socrates fent servir Pidgin.....	52
Figura 42: L'usuari socrates rep el missatge enviat per l'usuari mateu a l'aplicació Monocles Chat.....	52
Figura 43: Trucada de veu fent servir Monocles Chat.....	52
Figura 44: Captura de pantalla Joplin.....	53
Figura 45: Pàgina de configuració inicial Trilium.....	55
Figura 46: Pàgina login trilium.....	56
Figura 47: Notes Trilium creades per defecte.....	56

Figura 48: Accés a Trilium fent servir https.....	57
Figura 49: Importar notes a Trilium.....	58
Figura 50: Nota importada a Trilium.....	58
Figura 51: Trilium amb totes les notes importades.....	58
Figura 52: Exemple Panell Dashy.....	59
Figura 53: Exemple panell Homepage.....	59
Figura 54: Exemple panell Homarr.....	60
Figura 55: Panell per defecte de Homepage.....	62
Figura 56: Enllaç Forgejo a Homepage.....	63
Figura 57: Nextcloud a Homepage.....	64
Figura 58: Panell Homepage amb tots els serveis del nostre servidor.....	64
Figura 59: Preferits Homepage.....	65
Figura 60: Panell de serveis final.....	65

# Introducció

## Context i justificació

És tracta de crear un servidor per una petita organització dedicada a la creació de videojocs anomenada Weküfu Studios. L'objectiu és que no depengui de les grans corporacions. Anirem afegint diversos serveis (git, fitxers...) per substituir els serveis de tercers que hi ha actualment.

Hi ha moltes organitzacions que depenen completament en les grans corporacions i els seus serveis: Gmail, Google Drive, Microsoft Office, entre d'altres.

Penso que és preferible que les organitzacions tinguin control sobre les seves dades i l'única manera d'aconseguir això és tenint les dades i els serveis en un ordinador propi de l'organització.

## Objectius

Un cop tenim un servidor amb GNU/Linux anirem instal·lant i configurant els diferents serveis. Tindrem una secció per cada un dels serveis:

- Web
- Servidor de fitxers
- Git
- Comunicació tant de text com de veu
- Notes
- Panell de serveis

## Mètode

Seguirem el mateix mètode per a cada un dels 6 serveis. Cada servei tindrà una secció. Primer tindrem un petit resum de les diferents opcions de software que tenim pel servei en qüestió. Després d'escollir una opció, l'instal·larem, la configurarem, migrarem les dades, si és necessari, dels serveis de tercers al nou servei, i finalment, farem algun test per assegurar que el nou servei funcioni correctament.

## Planificació del treball

El següent diagrama Gantt mostra la planificació del projecte:



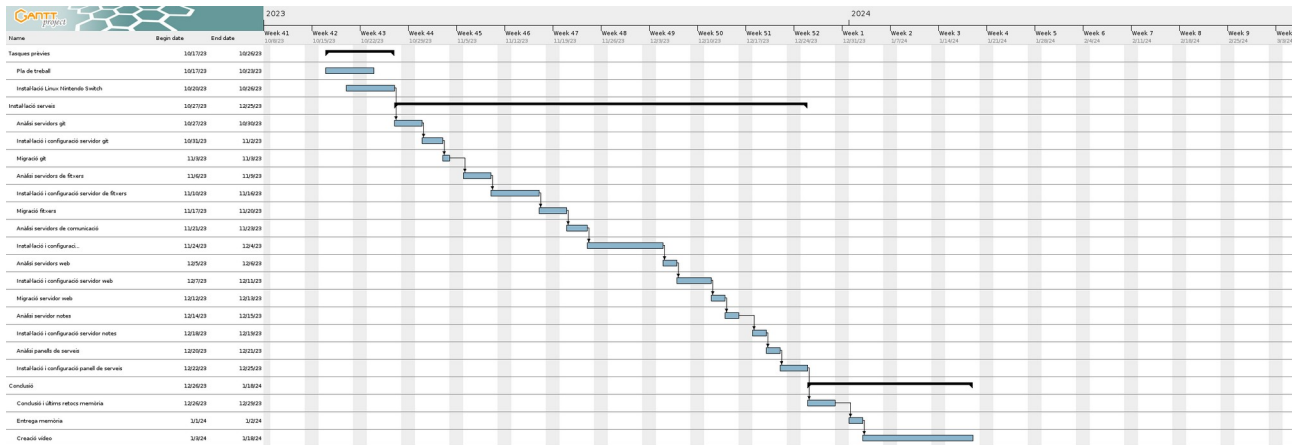


Figura 1: Diagrama Gantt del projecte

## Resum del resultat final

Al final del projecte, tindrem un servidor Fedora amb tots els serveis mencionats anteriorment funcionant correctament.

## Resum de les seccions

### Web

A la secció Web instal·larem el primer servei del servidor. Veurem 3 diferents opcions disponibles de servidors web: Algernon, Apache, i Nginx.

Un cop vistes les opcions, instal·larem Apache i publicarem la pàgina web de Weküfu Studios.

Aquesta és la secció més curta, ja que la instal·lació i configuració inicial son molt simples. El servidor web és molt important, ja que també l'utilitzarem pel servidor de Fitxers i com a proxy pel servidor Git.

### Fitxers

A la secció Fitxers, també comencem analitzant 3 opcions: Nextcloud, OwnCloud, i Seafiler.

Un cop analitzades, instal·larem un servidor Nextcloud perquè Weküfu Studios pugui guardar tots els seus fitxers. Abans, però, instal·larem i configurarem MariaSQL, la base de dades que utilitzara Nextcloud.

Adicionalment, configurarem https en el servidor Apache per poder accedir a Nextcloud de manera segura. També instal·larem un servidor Collabora per poder editar documents directament des del navegador, sense haver-los de descarregar.

Finalment, migrarem tots els fitxers que Weküfu Studios té al Google Drive al nou servidor Nextcloud.

## Git

A la secció Git, comencem fent una ullada a 4 servidors git disponibles per instal·lar: Gogs, Gitea, Forgejo, i OneDev.

El servidor que instal·larem per guardar els repositoris git de l'organització serà Forgejo. Per accedir a Forgejo farem servir Apache com un proxy, d'aquesta manera tindrem una connexió encriptada tot i que el servidor Forgejo funciona amb http.

També configurarem un 'runner' que permetrà a Forgejo executar scripts quan detecti certes accions. En aquest cas, farem servir un script que construirà la pàgina web cada vegada que fem un push al repositori d'aquesta.

Al final de la secció migrarem els repositoris que actualment estan a GitHub i farem alguns tests per comprovar que funcioni bé.

## Comunicació

A la secció Comunicació, començarem veient els dos estàndards principals dels quals podem triar: XMPP, i Matrix.

Ens decantarem per XMPP i instal·larem Prosody, un servidor que compleix amb l'estàndard xmpp, per permetre la comunicació entre els membres de l'organització.

Un cop instal·lat provarem d'enviar un missatge i fer una trucada fent servir el servidor per assegurar-nos que funciona bé.

## Notes

A la secció Notes, veurem 3 opcions: Joplin, Nextcloud Notes, Trilium.

Les dues primeres opcions tenen certes mancances, i per tant, instal·larem Trilium, un software per guardar notes jeràrquicament que permetrà als membres de Weküfu Studios crear notes fàcilment i tenir-les ordenades.

Al final de la secció, migrarem les notes que actualment estan en format md al servidor Trilium que les convertirà a html.

## Panell de serveis

Per acabar, a la secció Panell de serveis, instal·larem un panell web on ficarem enllaços als serveis que hem configurat fins ara perquè siguin fàcils d'accedir. Primer de tot analitzem 3 opcions: Dashy, Homepage, Homarr.

En aquest cas, l'elecció no té molta importància, ja que totes les opcions tenen més o menys les mateixes opcions de configuració i ens ofereixen la funcionalitat desitjada. La que instal·larem serà Homepage.

Per afegir els enllaços, modificarem els fitxers de configuració yml de Homepage.

# 1 Web

Les pàgines permeten a les companyies mostrar informació sobre elles a tothom fàcilment a través de un navegador. En el nostre cas, Weküfu Studios té una pàgina web amb informació sobre l'organització i els videojocs que ha desenvolupat i està desenvolupant actualment. A continuació migrarem aquesta pàgina web al nostre servidor.

A més a més, com veurem en les subseqüents seccions, el servidor web ens serà d'utilitat pel servidor de Fitxers i el servidor Git també.

## Opcions disponibles

### Algernon

Escrit en Go. Entre les seves característiques destaquen:

- Té suport per QUIC i HTTP/2, i varies bases de dades entre elles PostgreSQL i MariaDB.
- Suporta bastants tipus de fitxers: Markdown, Pongo2, Amber, Sass, GCSS, JSX...
- S'executa en més d'un fil d'execució per poder respondre a més d'una petició a la vegada.



### Apache

Segons [w3techs](http://w3techs.com), és el segon servidor web més usat amb un 30.9%, després de Nginx que té un 34.2%. Està escrit en C i és un dels servidors més vells, es va publicar el 1995.

Té una base molt simple, però es pot expandir amb desenes de mòduls. Per exemple, tenim un modul anomenat `mod_log_config` que guarda un log de les peticions que es fan al servidor, i un altre mod anomenat `mod_ssl` que ens permet encriptar les connexions, és a dir, fer servir https.



### Nginx

Com hem vist en l'apartat anterior, Nginx és el servidor més usat en l'actualitat. També està escrit en C.

Algunes de les diferències entre Apache i Nginx és que Apache pot executar codi php i Nginx no, per a fer-ho, a de demanar a un altre programa que ho faci per ell. Una altra diferència és que Apache pot configurar cada carpeta individualment mentre que Nginx no pot.

## Instal·lació i configuració

Farem servir Apache. És fàcil d'instal·lar i configurar i com veurem en l'apartat del servidor de fitxers, el fet que ens permeti executar codi php fàcilment ens serà útil.

El servidor Apache està en el paquet anomenat `httpd`, l'instal·lem fent servir `dnf`:

```
[root@switchroot ~]# dnf install httpd
...
Complete!
```

Tot seguit, com hem fet amb Forgejo, executem `systemctl start` per iniciar el servidor i `systemctl enable` per iniciar-lo automàticament quan el servidor s'engega:

```
[root@switchroot ~]# systemctl start httpd.service
[root@switchroot ~]# systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

Ara el servidor ja està funcionant. Si escrivim l'adreça IP del servidor veurem la pàgina de benvinguda:

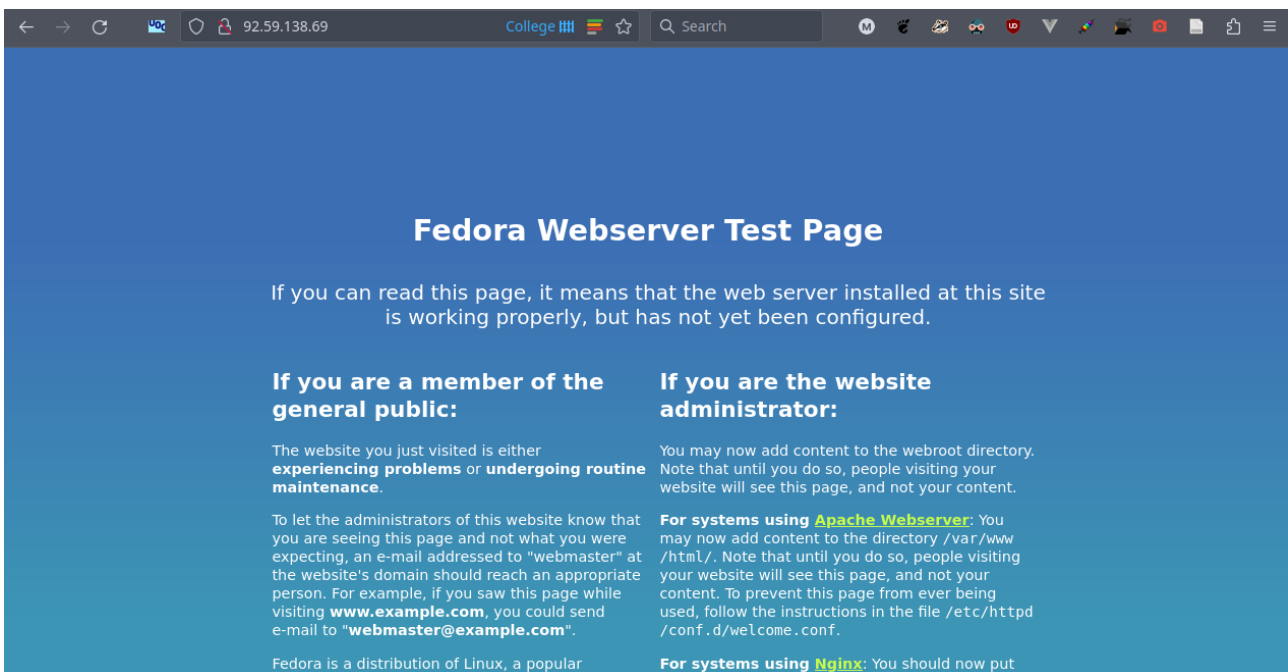


Figura 2: Pàgina benvinguda servidor Apache

De moment no configurarem https, ja que la pàgina de l'organització és una pàgina estàtica, i per tant, no es transmet cap dada sensible.

## Migració

Mes endavant, a la secció Git apartat Migració migrarem el repositori de la pàgina web de GitHub al servidor que escollim, de moment baixarem la pàgina fent servir el repositori de GitHub. Aquest repositori té una branca anomenada `gh-pages` amb la pàgina web. Així doncs si copiem el contingut d'aquesta branca al directori `/var/www/html`, la carpeta on hem de ficar els fitxers del servidor web, el servidor ens hauria de servir la web. Per copiar el contingut de la branca del repositori, podem fer servir git clone:

```
[root@switchroot html]# git clone https://github.com/WekufuStudios/WekufuStudios.github.io --
branch gh-pages
Cloning into 'WekufuStudios.github.io'...
remote: Enumerating objects: 661, done.
remote: Counting objects: 100% (661/661), done.
remote: Compressing objects: 100% (304/304), done.
remote: Total 661 (delta 331), reused 661 (delta 331), pack-reused 0
Receiving objects: 100% (661/661), 187.92 MiB | 504.00 KiB/s, done.
Resolving deltas: 100% (331/331), done.
Updating files: 100% (179/179), done.
```

El procés és el mateix que faríem si ja tinguéssim configurat el nostre servidor git, l'única diferencia és que hauríem d'afegir l'opció ``-c http.sslVerify=false`` perquè ignori que el certificat està signat per nosaltres mateixos.

Amb ``--branch gh-pages`` li indiquem que volem baixar la branca anomenada gh-pages.

Ara tenim la branca dintre una carpeta amb el nom del repositori, però volem que els continguts de la carpeta estiguin directament a la carpeta html, així que els movem tots a la carpeta pare (html) i eliminem la carpeta del repositori:

```
[root@switchroot html]# cd WekufuStudios.github.io/
[root@switchroot WekufuStudios.github.io]# mv * ../
[root@switchroot WekufuStudios.github.io]# ls
[root@switchroot WekufuStudios.github.io]# cd ..
[root@switchroot html]# rm -rf WekufuStudios.github.io
```

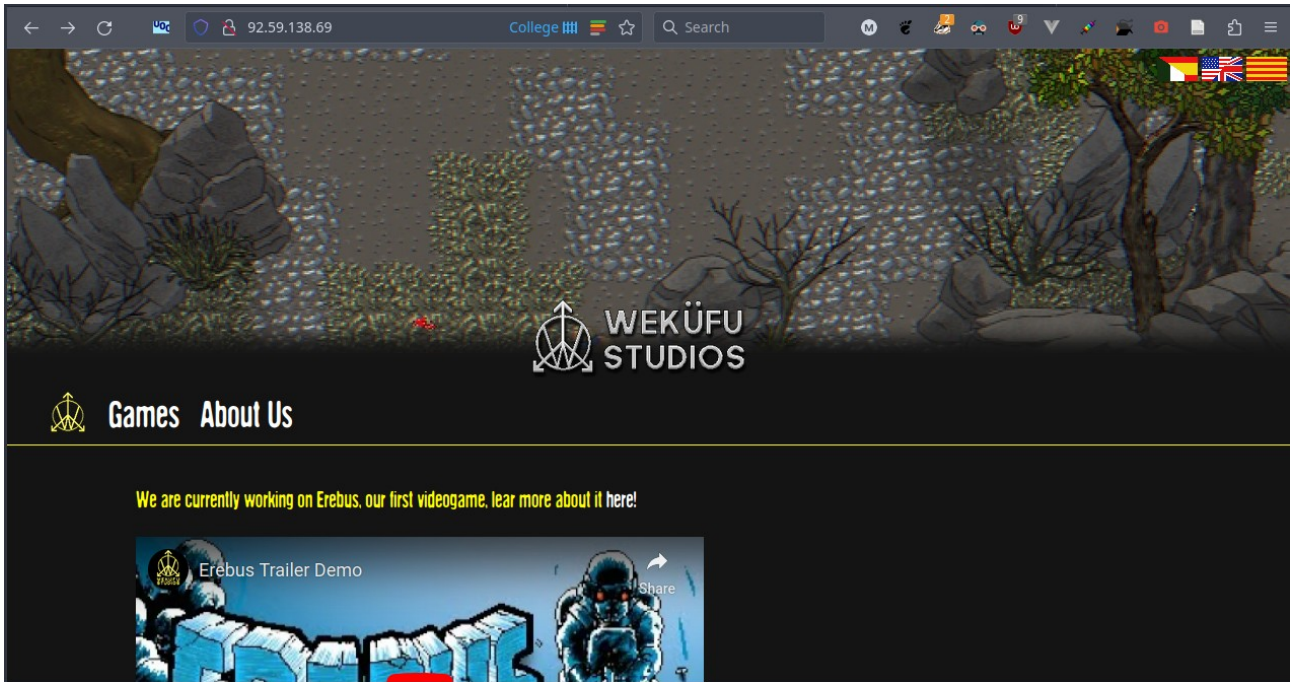
Per eliminar la carpeta WekufuStudios.github.io, fem servir l'opció `'r'` perquè elimini tot el contingut recursivament, ja que la carpeta `.git` s'ha quedat a dins al ser una carpeta oculta. La podem eliminar sense por, ja que no té cap ús en el servidor web. L'opció `'f'`, de `'force'`, serveix perquè no ens preguntin si volem eliminar els fitxers un per un.

A continuació canviem el propietari dels fitxers a mateus, el meu usuari, i el grup a apache, el grup que el servidor Apache va servir. També canviem els permisos a 750, de manera que jo tinc tots els permisos sobre els fitxers, el servidor apache els pot executar i llegir però no escriure, i els altres usuaris no tenen cap permís:

```
[root@switchroot www]# chown -R mateus html/
[root@switchroot www]# chgrp -R apache html/
[root@switchroot www]# chmod -R 750 html/
```

L'opció `'R'` serveix per aplicar els canvis recursivament.

Si recarreguem la pestanya amb la pàgina de benvinguda, veurem la pàgina web de Weküfu Studios:



*Figura 3: Pàgina web Weküfu Studios al servidor Apache*

## 2 Fitxers

Weküfu Studios té molts documents i fulls de càlcul a Google Drive, a part d'imatges i vídeos mostrant els videojocs, entre d'altres. És preferible que el servidor ens permeti modificar els fitxers sense haver de baixar-los, similar a Google Drive.

### Opcions disponibles



L'opció més popular, programat en php i JavaScript. Està format per 4 serveis:

- Nextcloud Files: servidor de fitxers
- Nextcloud Talk: Comunicació de text i veu
- Nextcloud Groupware: calendari, contactes i correu
- Nextcloud Office: Col·laboració en temps real de documents.



OwnCloud, també programat en php i JavaScript, té una interfície molt similar a Google Drive. La seva pàgina menciona que disposa d'integració d'OnlyOffice, de manera que podríem modificar i col·laborar en documents fàcilment.



Finalment, tenim Seafile, programat en C i la seva pàgina principal afirma que té un gran rendiment. La seva interfície també és bastant similar a la del Google Drive.

Adicionalment, disposa d'una característica que et permet crear wikis, però això no és tant important, ja que Forgejo ja ens permet crear wikis i també ho podríem fer des del servidor web que hem configurat en la secció 1.

### Instal·lació i configuració

A continuació instal·larem Nextcloud, ja que és l'opció més completa. Ens permet guardar fitxers i col·laborar en temps real. Adicionalment, ens permet afegir serveis més endavant si els necessitem, per exemple, si en el futur necessitem un calendari, ho podrem configurar fàcilment amb Nextcloud.

Per instal·lar Nextcloud necessitem un servidor web, farem servir el servidor Apache que hem instal·lat en la secció anterior.

Nextcloud també necessita una base de dades MariaDB/MySQL. Per instal·lar MariaDB podem fer servir dnf, després iniciem el servei i el configurem perquè s'executi automàticament quan el servidor s'engega, i finalment executem `mysql_secure_installation` per configurar la base de dades:

```
[root@switchroot /]# dnf install mariadb-server
...
Complete!
[root@switchroot /]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service →
/usr/lib/systemd/system/mariadb.service.
[root@switchroot /]# systemctl start mariadb
[root@switchroot /]# mysql_secure_installation
...
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Tot seguit hem de crear un usuari i una base de dades per Nextcloud. Hem de donar tots els privilegis de la nova base de dades al nou usuari:

```
[root@switchroot /]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'nextcloud_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.021 sec)

MariaDB [(none)]> CREATE DATABASE IF NOT EXISTS nextcloud;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nextcloud.* TO 'nextcloud_user'@'localhost'
IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.011 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.008 sec)

MariaDB [(none)]> quit
Bye
```

Instal·lem algunes dependències que Nextcloud necessita:

```
[root@switchroot /]# dnf install php php-gd php-mbstring php-intl php-mysqlnd php-opcache php-
json php-zip php-xml
...
Complete!
```



Ara anem a la carpeta arrel del servidor Apache i descarreguem la ultima versió de Nextcloud fent servir wget:

```
[root@switchroot html]# wget https://download.nextcloud.com/server/releases/latest.zip
```

Un cop hem descomprimit el fitxer, eliminem el fitxer comprimit:

```
[root@switchroot html]# unzip latest.zip  
[root@switchroot html]# rm latest.zip
```

Després de descomprimir el fitxer, tenim una carpeta amb el nom 'nextcloud'. Canviem el propietari a mateus i el grup a apache per que tingui el mateix propietari i grup que la resta de fitxers del servidor web. També canviem els permisos perquè tant el propietari com el grup puguin executar, escriure i llegir.

```
[root@switchroot html]# chown -R mateus:apache nextcloud/  
[root@switchroot html]# chmod -R 770 nextcloud/
```

Al fer el servidor web, hem donat a apache permisos de lectura i execució, però en aquest cas, com que nextcloud necessita escriure a una configuració, li hem de donar permisos d'escriptura també. Si no ho fem, ens trobarem amb el següent error que ens diu que no pot escriure al fitxer de configuració:

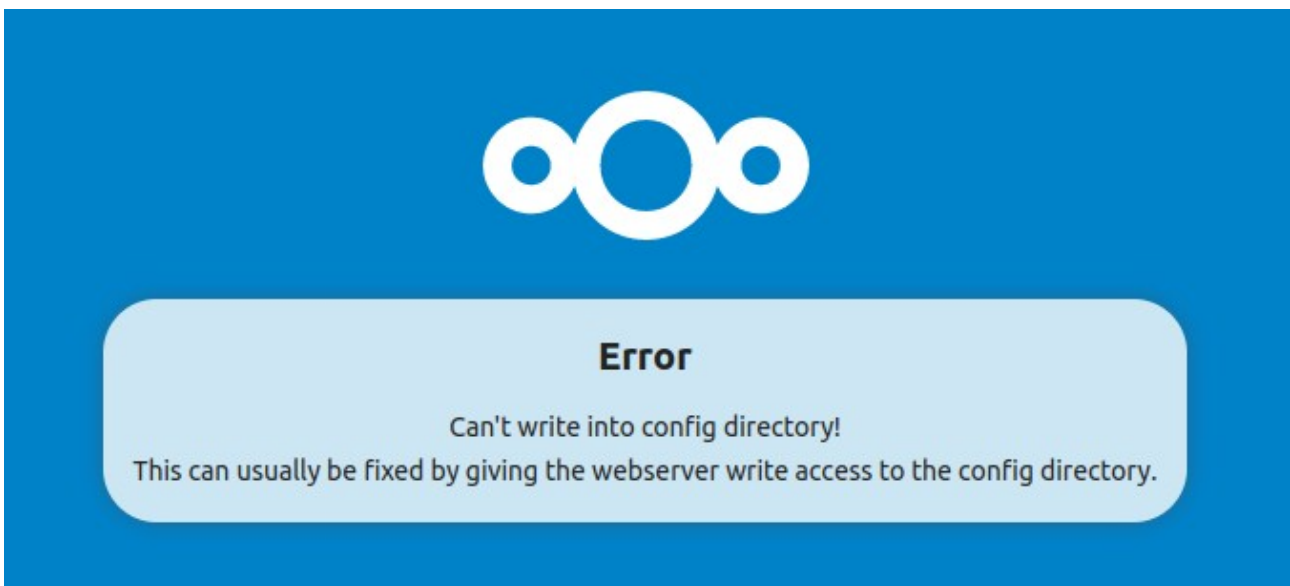


Figura 4: Nextcloud no pot escriure al fitxer de configuració

Reiniciem el servidor Apache i anem a l'adreça 192.168.1.71/nextcloud, ja que hem ficat la carpeta 'nextcloud' a l'arrel del servidor web. Apache ens redirigeix automàticament a 192.168.1.71/nextcloud/index.php. Per configurar Nextcloud hem desconnectat el servidor d'internet, per això estem fent servir l'adreça IP privada.

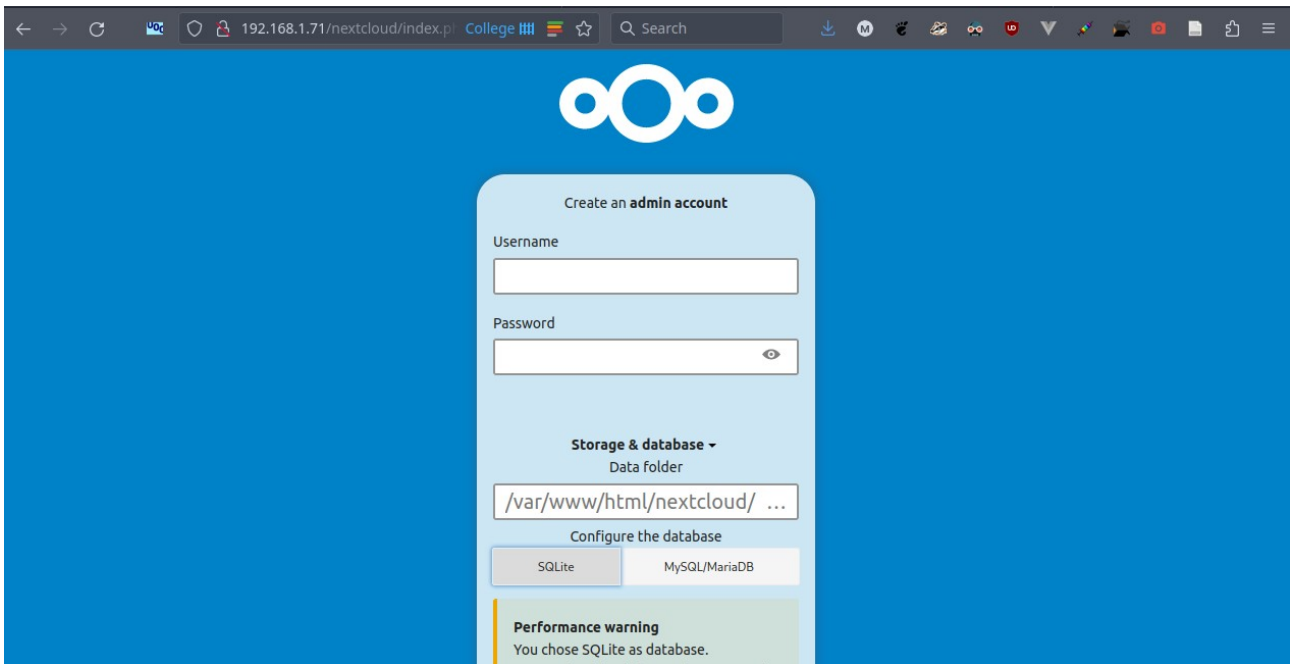


Figura 5: Pàgina web d'instal·lació Nextcloud

Omplim l'usuari i contrasenya del compte administratiu i configurem la base de dades perquè es connecti amb la base de dades que hem creat fa un moment. L'usuari és 'nextcloud\_user', el nom de la base de dades és 'nextcloud', i el port que fa servir MariaDB per escoltar peticions és 3306:

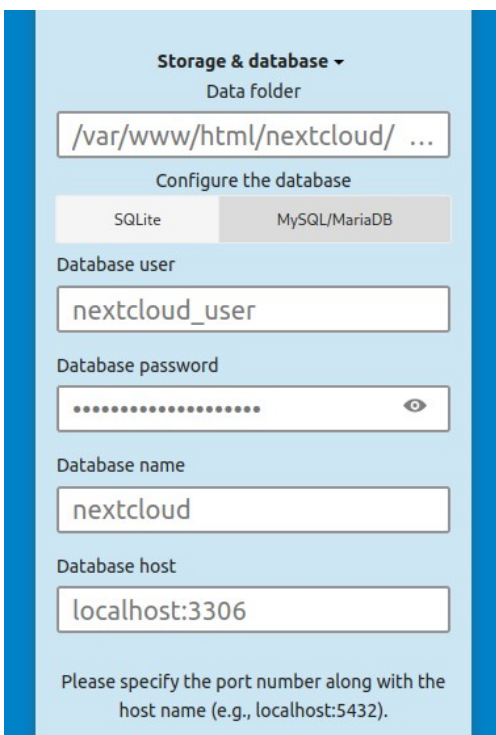


Figura 6: Configuració base de dades Nextcloud

Cliquem el botó 'install'. I al cap d'uns segons ens redirigeix a la pàgina d'inici de sessió:

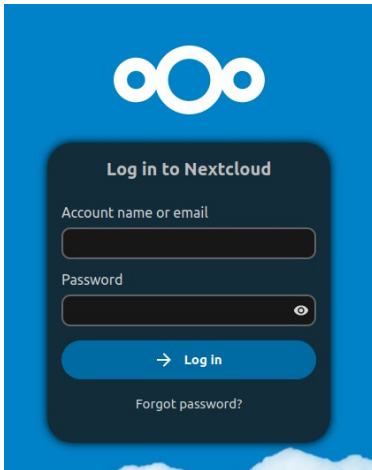


Figura 7: Pàgina d'inici de sessió Nextcloud

Escribim l'usuari i contrasenya que hem definit pel compte d'administrador i accedim a la pàgina d'inici de Nextcloud:

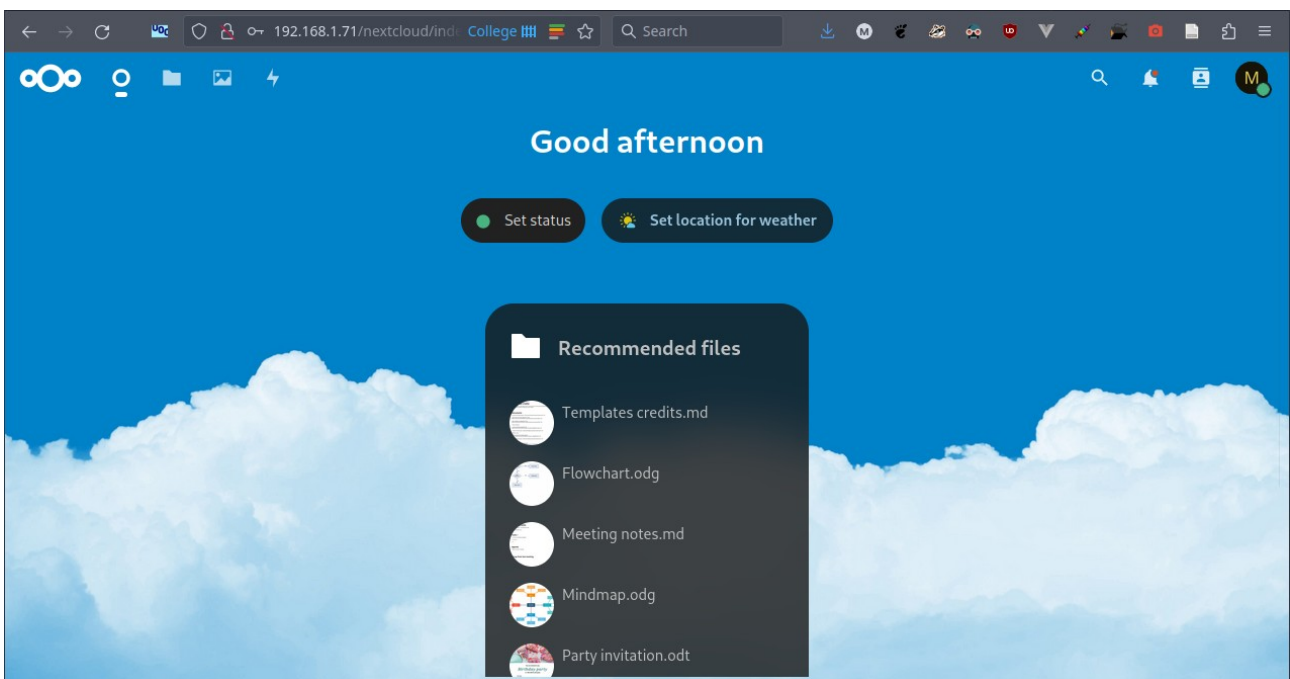


Figura 8: Pàgina inicial Nextcloud

A la part superior tenim una barra de navegació que ens permet anar a l'emmagatzematge de fitxers. Per defecte Nextcloud hi ha ficat un manual i uns quants fitxers més:

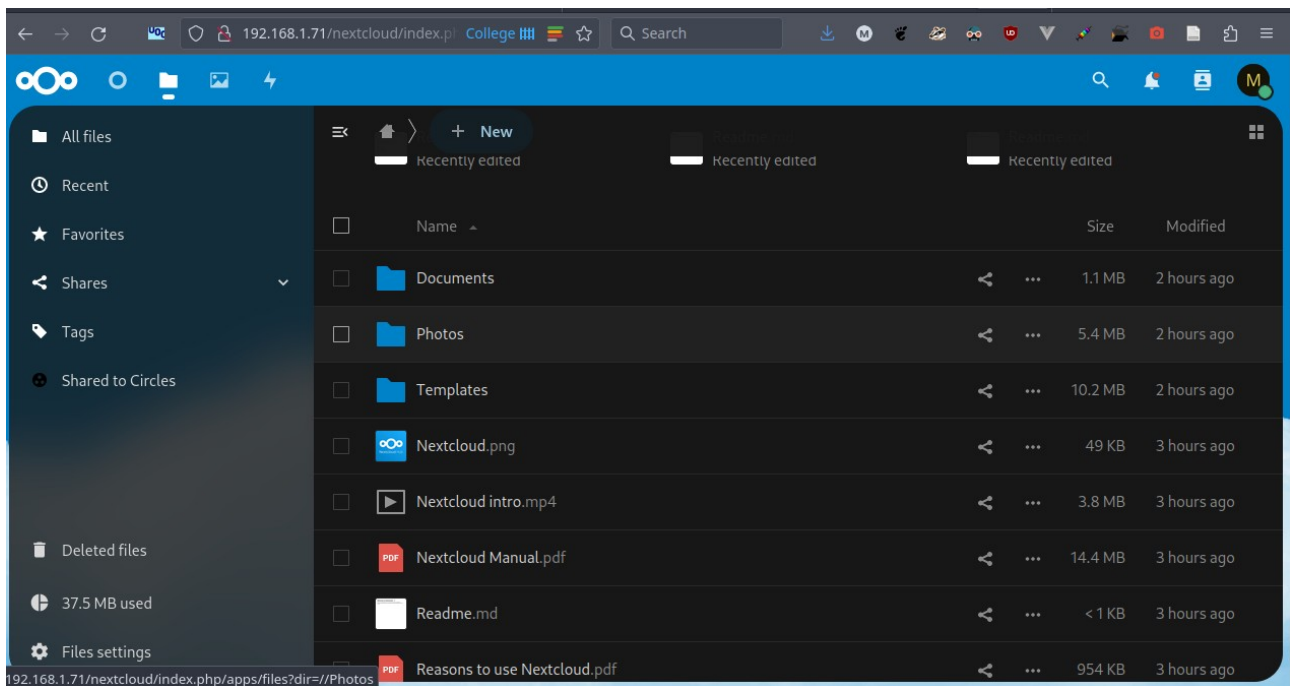


Figura 9: Fitxers Nextcloud

Per defecte quan cliquem un fitxer en format docx, ods... el descarrega localment. Per poder editar els fitxers directament des del navegador haurem d'instal·lar una extensió. Però primer arreglarem uns errors que apareixen a la configuració administrativa:

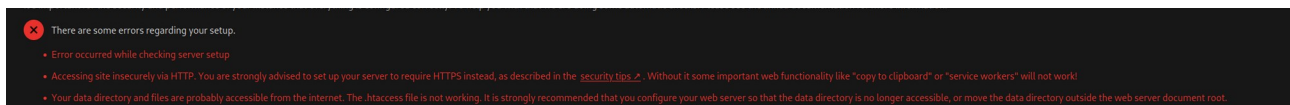


Figura 10: Errors Nextcloud

Començarem per l'últim. Diu que el directori data i els fitxers probablement són accessibles des d'internet. La carpeta 'nextcloud' té un fitxer anomenat '.htaccess' que hauria de sobreescrivir certs paràmetres del servidor web només pels fitxers d'aquesta carpeta. El problema és que, per defecte, el servidor Apache està ignorant aquest fitxer.

Hem d'entrar a la configuració d'Apache i canviar 'AllowOverride' a 'All' en la configuració del directori /var/www/html. D'aquesta manera es tindran en compte els fitxers .htaccess en el directori html.

```
...
<Directory "/var/www/html">
    ...
    AllowOverride All
    ...
```

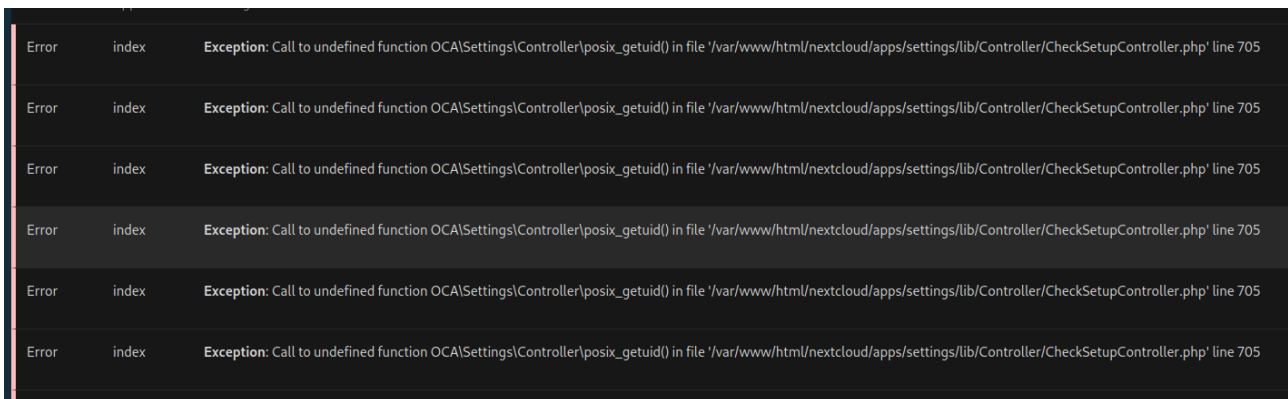
Després de canviar aquest paràmetre, reiniciem el servidor Apache i ja no ens surt l'error en qüestió a Nextcloud.

Anem a solucionar el següent error. Diu que la connexió no és segura perquè estem fent servir http. Per solucionar-ho, hem d'habilitar https en el servidor Apache. Necessitem el mòdul `mod_ssl` d'Apache, l'instal·lem fent servir `dnf`:

```
[root@switchroot httpd]# dnf install mod_ssl
...
Complete!
```

Un cop instal·lat el mòdul, podem trobar la seva configuració a `/etc/httpd/conf.d/ssl.conf`. Però no hem de configurar res més, ja que per defecte fa servir un certificat signat per nosaltres mateixos que està al directori `/etc/pki/tls/certs/`.

Seguim tenint el primer error que l'únic que ens diu que hi ha hagut un problema amb la configuració del servidor. Si anem a la pestanya 'logging' de la configuració de Nextcloud trobarem més informació:



```
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
Error index Exception: Call to undefined function OCA\Settings\Controller\posix_getuid() in file '/var/www/html/nextcloud/apps/settings/lib/Controller/CheckSetupController.php' line 705
```

Figura 11: Errors logging Nextcloud

Nextcloud prova d'executar la funció `posix_getuid` però no la troba. Gràcies a [aquest pregunta al fòrum](#), sabem que ens falta el paquet `php-posix`. L'instal·lem amb `dnf` i ja no ens apareix més l'error.

Però ara ens trobem amb un nou error que diu que la memòria utilitzada per php està per sota el valor recomanat de 512MB. Per arreglar aquest error hem d'accedir a `/etc/php/php.ini` i canviar 'memory\_limit' a 512M o un valor superior:

```
...
memory_limit = 512M
...
```

Després de fer aquest canvi, reiniciem php i el servidor web:

```
[root@switchroot config]# systemctl restart php-fpm.service
[root@switchroot config]# systemctl restart httpd
```

Ja no tenim més errors.

Per editar els documents des del navegador instal·larem l'extensió de [Collabora](#). Anem a 'Apps' i instal·lem el servidor Collabora (la versió ARM64, ja que aquesta és l'arquitectura de la Switch):

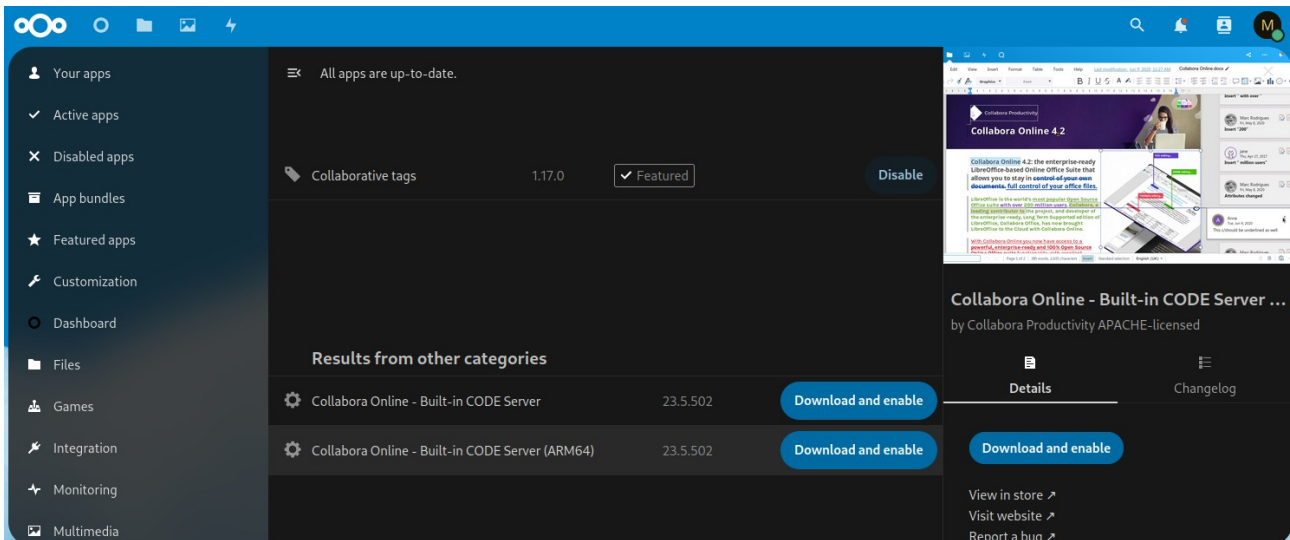


Figura 12: Instal·lació servidor Collabora

El procès d'instal·lació acaba amb error. Sembla ser que el temps de resposta és superior al límit per defecte de Nextcloud. La descripció de l'extensió ens proposa instal·lar-ho fent servir el terminal, així que això és el que farem:

```
[root@switchroot nextcloud]# sudo -u apache php -d memory_limit=512M ./occ app:install richdocumentscode_arm64 richdocumentscode_arm64 23.5.502 installed richdocumentscode_arm64 enabled
```

Ara sí que s'ha instal·lat correctament. Ara tenim un servidor Collabora, però també necessitem l'extensió 'Nextcloud Office' que ens serveix de connector. Un cop instal·lat 'Nextcloud Office', ja podem accedir i modificar fitxers des del navegador. Per exemple, si cliquem el fitxer d'exemple Invoice.odt:

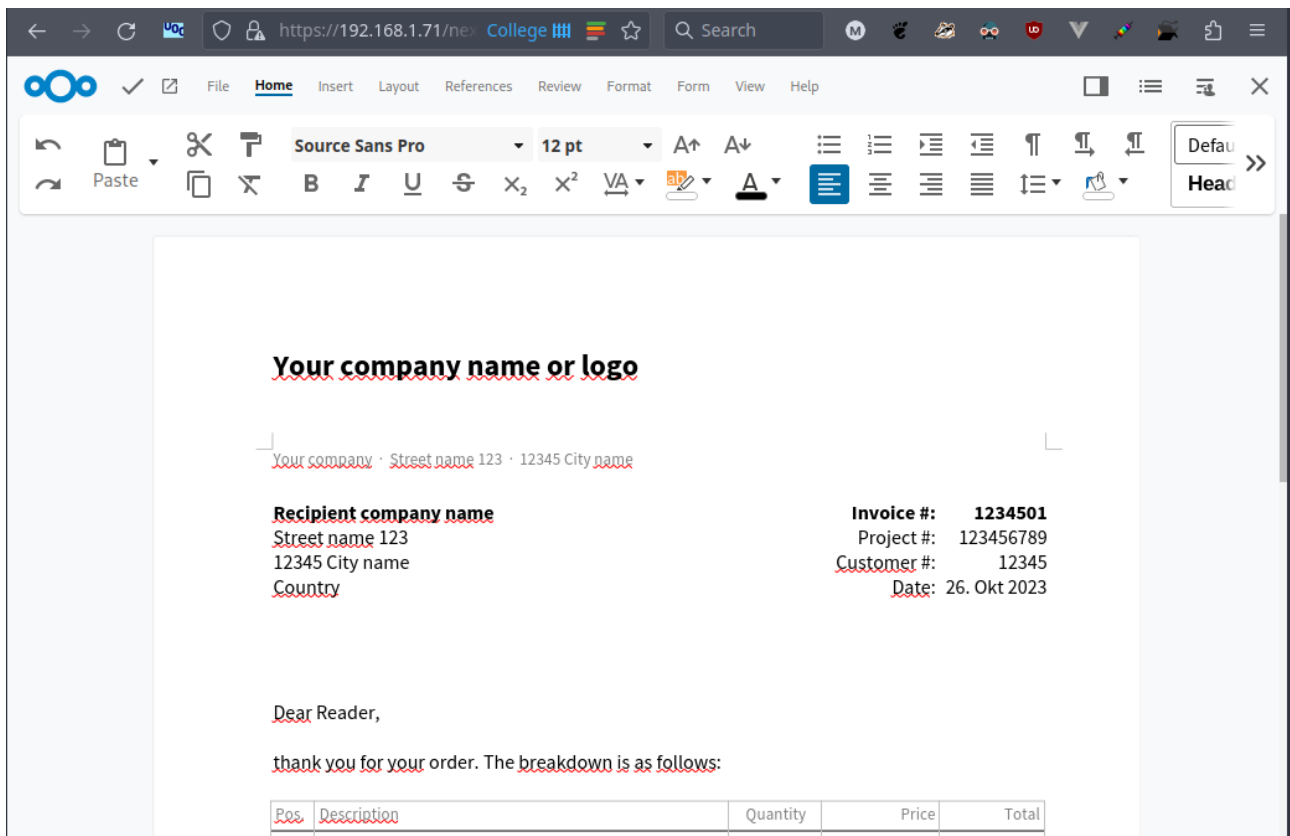


Figura 13: Fitxer Invoice.odt obert fent servir el servidor Collabora

Ens queda una ultima cosa per configurar. Si intentem accedir a Nextcloud fent servir l'adreça IP pública, no ens deixa entrar perquè diu que estem intentant accedir des d'un domini desconegut:

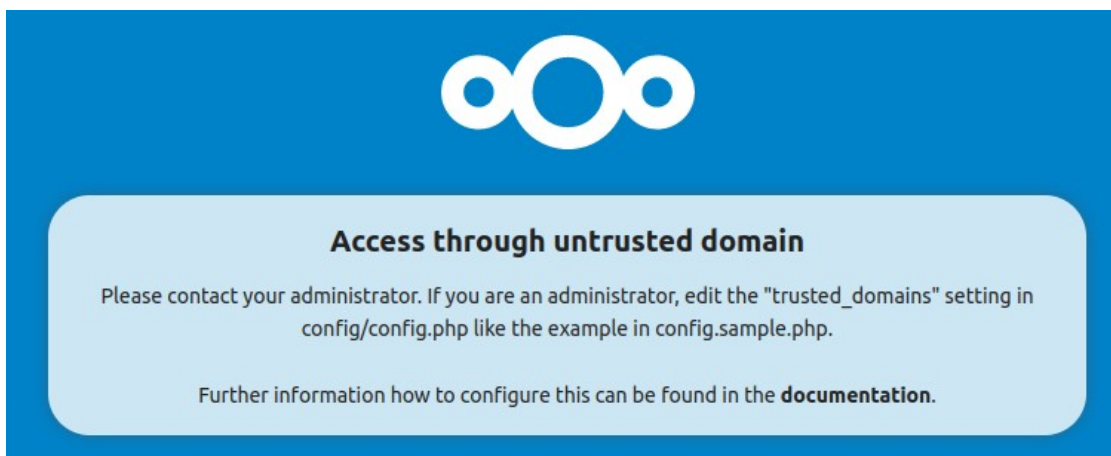


Figura 14: Pàgina domini desconegut Nextcloud

Per donar accés al servidor des de qualsevol part d'internet, hem de modificar el fitxer config/config.php. Modifiquem la propietat 'trusted domains' amb una nova línia amb el següent contingut:

```
'trusted_domains' =>
array (
  0 => '192.168.1.71',
  1 => '*',
),
```

L'asterisc fa que Nextcloud accepti tots els dominis. Si recarreguem la pàgina, ara sí que podem accedir fent servir la IP pública:

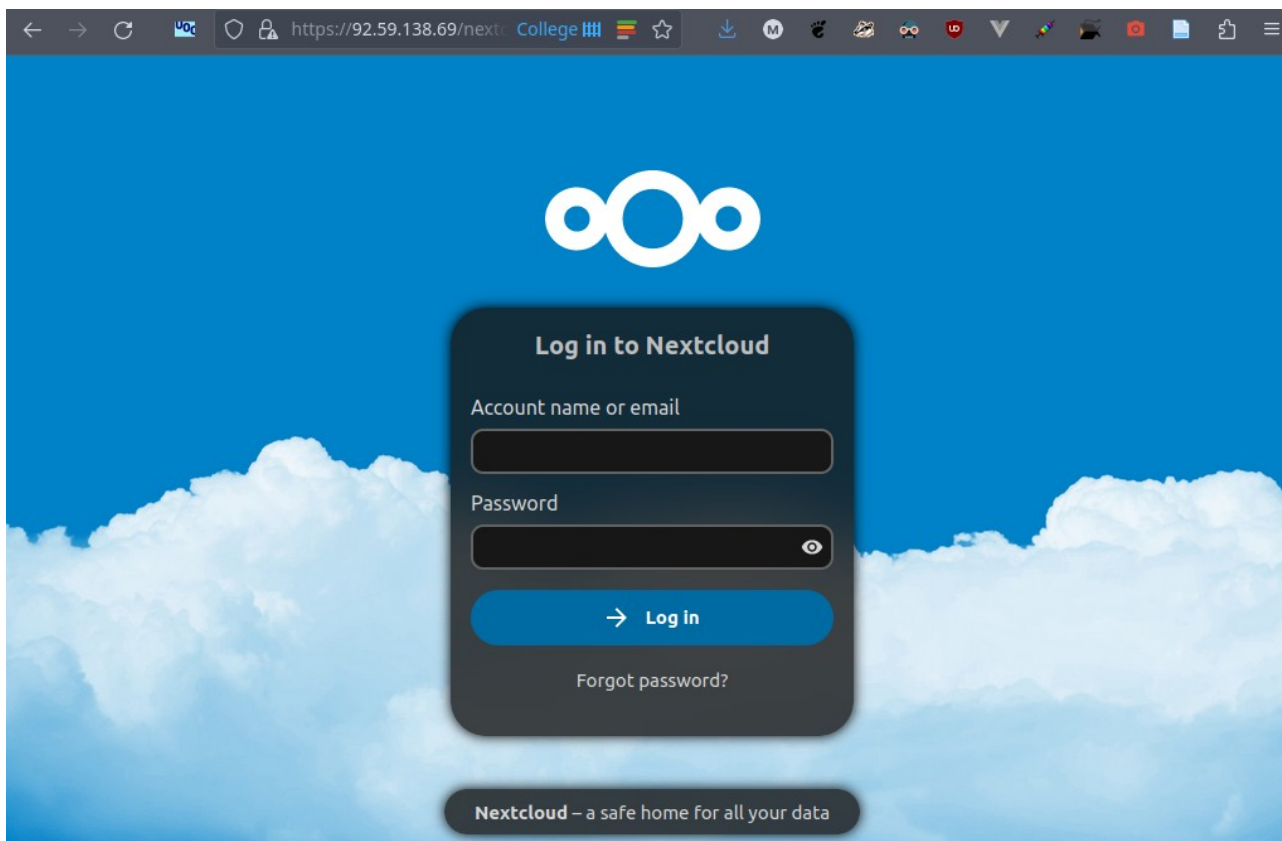


Figura 15: Pàgina login Nextcloud fent servir ip pública

## Migració

Abans de moure els fitxers del Google Drive a Nextcloud, hi ha un parell de coses que hem de fer. Primer de tot, anem a la configuració d'usuaris i creem un nou grup pels membres de l'organització i afegim aquests al grup. També creem tants usuaris com sigui necessari:



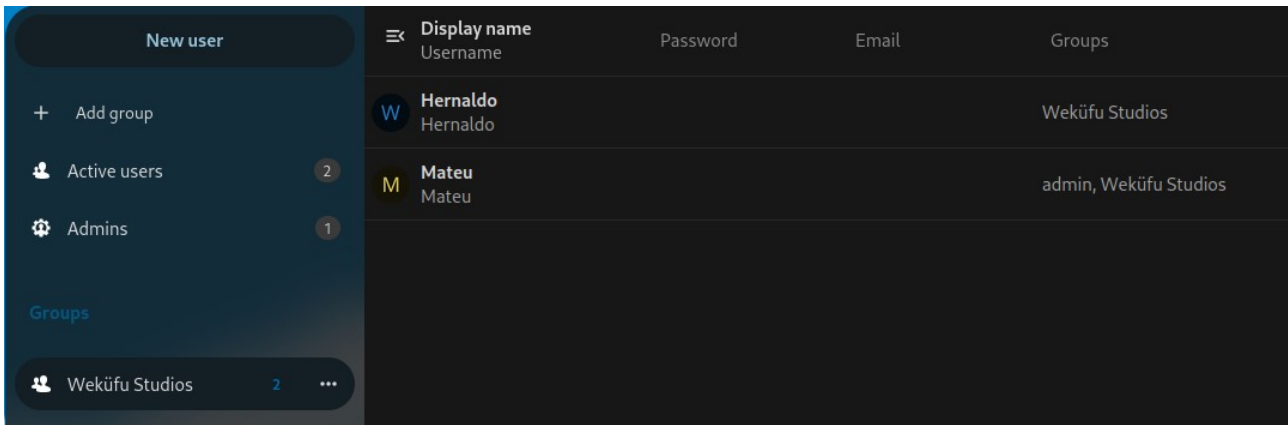


Figura 16: Grup Nextcloud amb els membres de Weküfu Studios

Farem servir aquest grup per compartir els fitxers que hi ha actualment al Google Drive entre tots els integrants de Weküfu Studios. Però per fer això, primer hem d'instal·lar l'extensió 'Group folders', ja que és aquesta extensió la que ens permet compartir carpetes entre tots els membres d'un grup.

Un cop tenim l'extensió instal·lada ens apareix una nova opció de configuració, 'Group folders', a la configuració administrativa. Hi entrem, creem una nova carpeta, i assignem aquesta carpeta al grup dels membres de l'organització, 'Weküfu Studios':

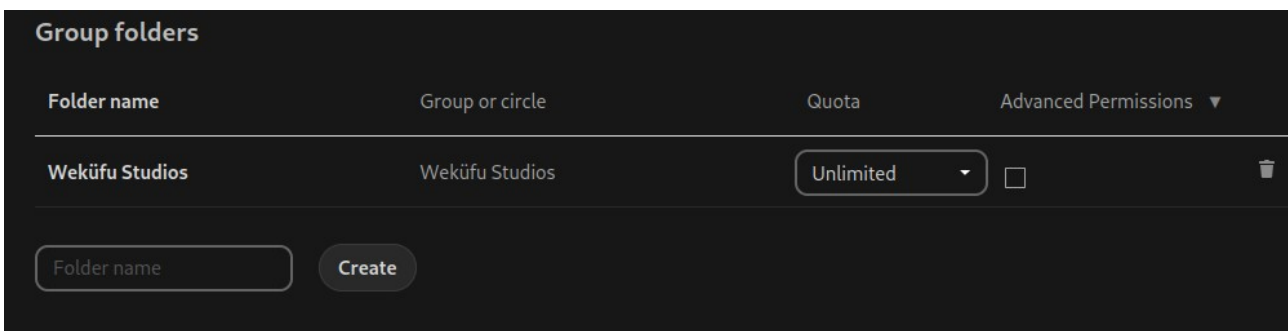


Figura 17: Configuració carpeta compartida del grup 'Weküfu Studios'

Ara, aquesta carpeta apareixerà a tots els usuaris membres del grup 'Weküfu Studios'.

Ara que tenim això enllestit, ja podem migrar tots els fitxers. Entrem al Google Drive i seleccionem tots els fitxers, ho podem fer clicant el primer element i tot seguit clicant l'últim element mentre mantenim shift pressionat.

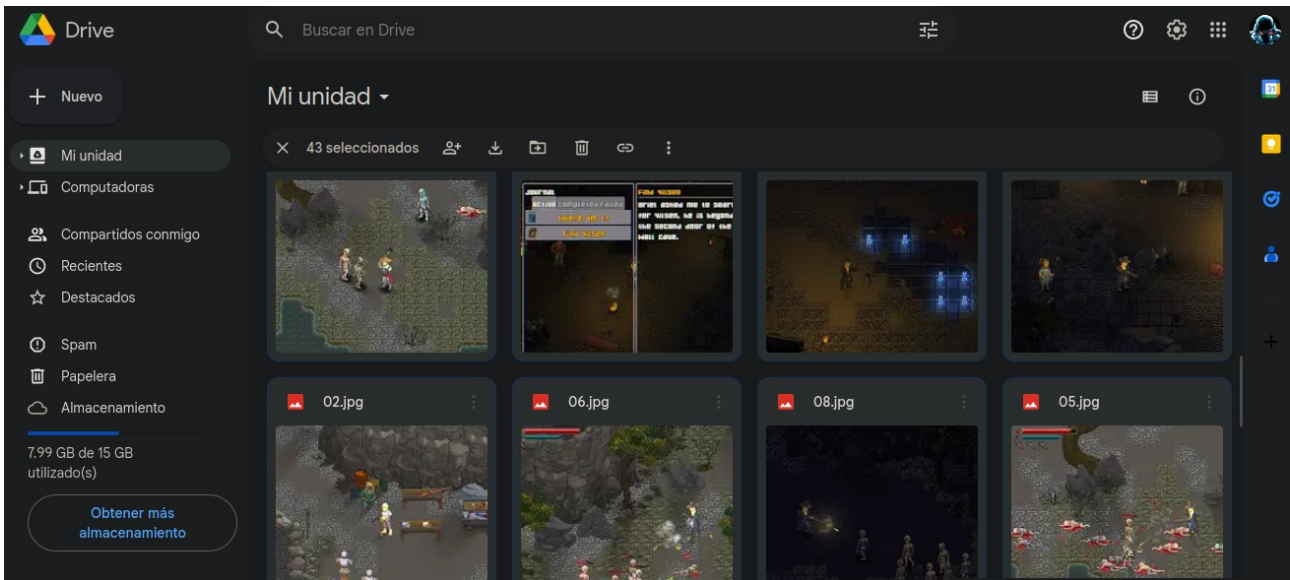


Figura 18: Fitxers al Google Drive

Baixem les carpetes comprimides, més d'una, ja que Drive divideix els fitxers en varies carpetes comprimides si ocupen molt, amb tots els fitxers al nostre sistema, dintre una carpeta que sincronitzarem amb Nextcloud. He anomenat aquesta carpeta 'Weküfu Studios' i l'he col·locada dintre la carpeta Home.

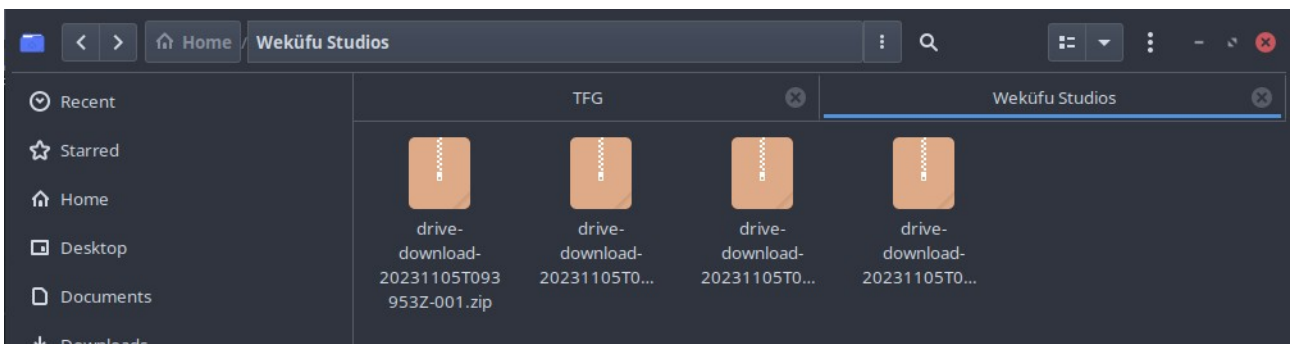


Figura 19: Carpeta 'Weküfu Studios' amb els fitxers del Drive comprimits

Descomprimim els fitxers i eliminem els fitxers comprimits. Ja tenim la carpeta a punt, només queda sincronitzar-la amb Nextcloud, lo qual podem fer amb el [client oficial](#). Seleccionem la carpeta 'Weküfu Studios' del nostre ordinador i la carpeta compartida 'Weküfu Studios' de Nextcloud i comencem la sincronització. Tardara una mica, ja que són 8 GB de fitxers.

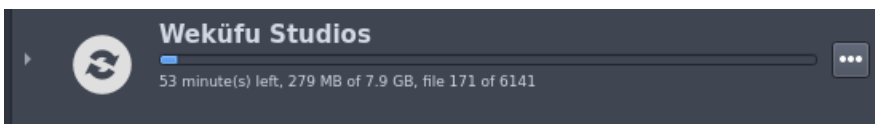


Figura 20: Sincronitzant carpeta 'Weküfu Studios' amb Nextcloud

A la següent imatge podem veure com els dos usuaris del grup 'Weküfu Studios' tenen accés a tots els fitxers del Drive que acabem de pujar:

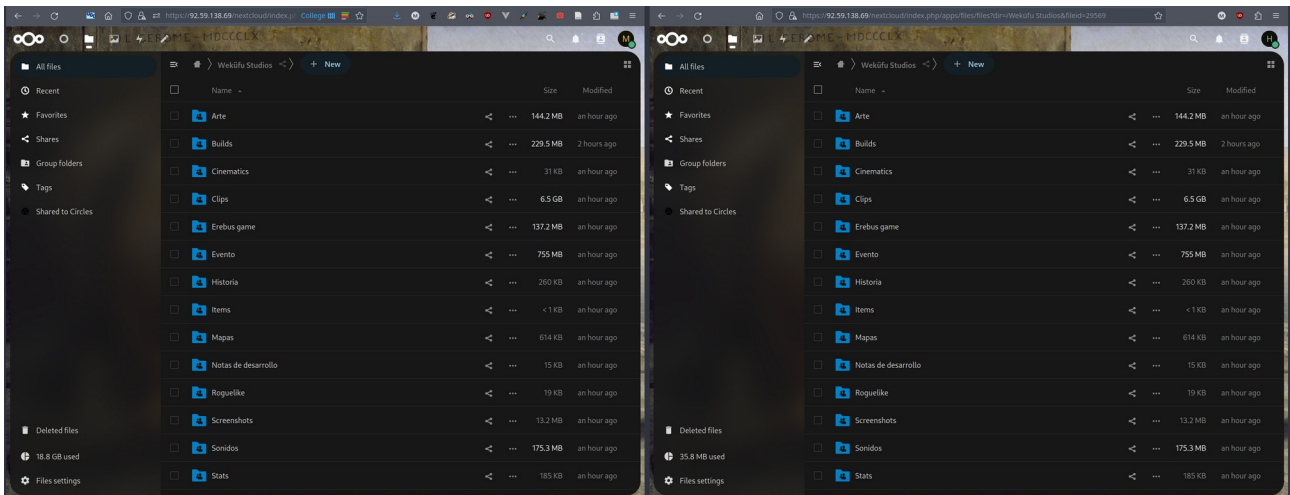


Figura 21: Els dos usuaris del grup 'Weküfu Studios' comparteixen la carpeta 'Weküfu Studios'

Qualsevol modificació d'aquest fitxers es veurà reflectida en tots els usuaris i fins i tot en les carpetes locals que estiguin sincronitzades fent servir el client de Nextcloud.

## 3 Git

Git és un software de control de versions, el més popular actualment. És un software molt important en les organitzacions que es dediquen al desenvolupament de software, ja que permet tenir un control de les diferents versions i podem veure qui ha fet cada canvi i quan, i, fins i tot recuperar un estat anterior si hi ha algun problema.

Es pot utilitzar git completament en local, però normalment és fa servir juntament amb un servidor per donar accés al codi a varies persones i que aquestes puguin col·laborar en la creació del software.

La organització per la qual estem fent el servidor té diversos repositoris a GitHub amb el codi dels seus videojocs i la seva pàgina web. A continuació buscarem una alternativa de codi obert i migrarem els repositoris.

### Opcions disponibles



**Gogs**

Gogs està escrit en Go i destaca per la seva fàcil instal·lació, només has de descarregar el binari correcta i executar-lo.

Té suport tant per MySQL com per PostgreSQL.

No té suport CI/CD, això és un punt negatiu, ja que la companyia fa servir CI/CD per compilar la seva pàgina web.



**Gitea**

Gitea és un dels servidors git de codi obert més coneguts. Va començar com un 'fork' de Gogs, però segons la seva pàgina web, gairebé tot el codi ha estat canviat. Te moltes de les característiques de GitHub, a més de emmagatzemar el codi, permet:

- Fer revisions de codi abans d'integrar aquest definitivament.
- Definir CI/CD: un seguit d'instruccions que s'executa de manera automàtica quan fem certes accions com per exemple 'push'.
- Organitzar el projecte en taules de tasques.
- Gestió de paquets.

També disposa d'una opció de pagament en que ens configuren el servidor per nosaltres, però a nosaltres el que ens interessa és instal·lar-ho en el nostre servidor.



**Forgejo**

Quan el domini i la marca de Gitea van ser adquirits per una empresa amb anim de lucre, es va crear Forgejo com un fork de Gitea, gestionat per una organització sense anim de lucre.

Les seves característiques són pràcticament les mateixes que Gitea.

## OneDev

Com a ultima opció tenim OneDev que, per variar, no està basat en Gogs i està programat en Java. A [la seva pàgina principal](#) destaca CI/CD i Kanban. Com hem vist les altres dues opcions anteriors també disposen de CI/CD i Kanban. Kanban és una manera d'organitzar les tasques en columnes: una columna per les tasques pendents, una altre per les tasques en procés, una més per les tasques que s'han de provar, i la final que indica les tasques completades.

Les altres opcions també tenen una opció per definir projectes de la mateixa manera.

La principal diferencia és que OneDev té un buscador molt potent: permet buscar variables i funcions. També permet buscar fent servir expressions regulars.

Com Gitea, disposa d'una opció de pagament. Aquesta opció de pagament ens ofereix varis extrems com suport, el buscador pot buscar a més d'un projecte a la vegada...

## Instal·lació i configuració

Tot seguit instal·larem Forgejo, ja que em sembla l'opció més completa: té moltes característiques necessàries com CI/CD i projectes, i a diferencia de Gitea, està gestionat per una empresa sense anim de lucre, el que el fa una opció més atractiva.

## Base

Primer de tot, descarreguem el binari i verifiquem la seva signatura amb GPG:

```
[mateus@switchroot ~]$ wget https://codeberg.org/forgejo/forgejo/releases/download/v1.20.5-0/forgejo-1.20.5-0-linux-arm64
--2023-10-23 16:26:15--
...
forgejo-1.20.5-0-linux-arm 100%[=====>]
94.00M 4.14MB/s in 25s

2023-10-23 16:26:41 (3.70 MB/s) - 'forgejo-1.20.5-0-linux-arm64' saved
[98567208/98567208]
```

```
[mateus@switchroot ~]$ gpg --keyserver keys.openpgp.org --recv
EB114F5E6C0DC2BCDD183550A4B61A2DC5923710
...
gpg: key A4B61A2DC5923710: public key "Forgejo <contact@forgejo.org>" imported
...
[mateus@switchroot ~]$ wget
https://codeberg.org/forgejo/forgejo/releases/download/v1.20.5-0/forgejo-1.20.5-0-linux-arm64.asc
--2023-10-23 16:27:26-- https://codeberg.org/forgejo/forgejo/releases/download/v1.20.5-0/forgejo-
1.20.5-0-linux-arm64.asc
...
forgejo-1.20.5-0-linux-arm 100%[=====>] 228 --.-
KB/s in 0.001s
...
[mateus@switchroot ~]$ gpg --verify forgejo-1.20.5-0-linux-arm64.asc forgejo-1.20.5-0-linux-arm64
gpg: Signature made Fri 06 Oct 2023 06:28:22 PM CEST
gpg: using EDDSA key F7CBF02094E7665E17ED6C44E381BF3E50D53707
gpg: Good signature from "Forgejo <contact@forgejo.org>" [unknown]
gpg: aka "Forgejo Releases <release@forgejo.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: EB11 4F5E 6C0D C2BC DD18 3550 A4B6 1A2D C592 3710
Subkey fingerprint: F7CB F020 94E7 665E 17ED 6C44 E381 BF3E 50D5 3707
```

A continuació fem el fitxer executable amb chmod:

```
[mateus@switchroot ~]$ chmod +x forgejo-1.20.5-0-linux-arm64
```

Ara que sabem que el programa no s'ha corromput durant la descarrega i que és executable, el podem moure amb els altres programes a /usr/local/bin:

```
[mateus@switchroot ~]$ sudo mv forgejo-1.20.5-0-linux-arm64 /usr/local/bin/forgejo
```

Segons la documentació necessitem els paquets git i git-lfs. Git ja està instal·lat per defecte, però git-lfs no, així que l'hem d'instal·lar. Ho podem fer amb `sudo dnf install git-lfs`.

A continuació necessitem un nou usuari anomenat git que Forgejo farà servir per les connexions ssh:

```
[mateus@switchroot ~]$ sudo groupadd --system git
[mateus@switchroot ~]$ sudo adduser --system --shell /bin/bash --comment 'Git Version
Control' --gid git --home-dir /home/git --create-home git
```

Ara hem de crear el directori on Forgejo guardara els repositoris i el directori on guardara la seva configuració. També hem de canviar els permisos, el propietaris i els grups d'aquest directoris als indicats a la documentació:

```
[mateus@switchroot ~]$ su -
Password:
[root@switchroot ~]# mkdir /var/lib/forgejo
[root@switchroot ~]# chown git:git /var/lib/forgejo && chmod 750 /var/lib/forgejo
[root@switchroot ~]# mkdir /etc/forgejo
[root@switchroot ~]# chown root:git /etc/forgejo && chmod 770 /etc/forgejo
```

Forgejo ja està punt, l'únic que queda és executar l'executable. Però en comptes d'executar-lo manualment podem instal·lar un scrip de systemd per poder controlar-ho més fàcilment i executar-lo automàticament quan el servidor s'engega:

```
[root@switchroot ~]# wget -O /etc/systemd/system/forgejo.service
https://codeberg.org/forgejo/forgejo/raw/branch/forgejo/contrib/systemd/forgejo.service
--2023-10-23 16:58:36--
https://codeberg.org/forgejo/forgejo/raw/branch/forgejo/contrib/systemd/forgejo.service
...
Saving to: '/etc/systemd/system/forgejo.service'

/etc/systemd/system/forgej 100%[=====]
2.44K --.-KB/s in 0s
...
```

Amb systemctl enable fem que l'script s'executi quan el servidor s'engega i amb systemctl start l'executem en aquest moment:

```
[root@switchroot ~]# systemctl enable forgejo.service
Created symlink /etc/systemd/system/multi-user.target.wants/forgejo.service →
/etc/systemd/system/forgejo.service.
[root@switchroot ~]# systemctl start forgejo.service
```

Si accedim al port 3000 del servidor des d'un navegador, podem accedir a Forgejo. Primer de tot, trobem una pàgina per acabar de configurar-lo. Deixarem tots els paràmetres per defecte excepte la base de dades que la canviarem a sqlite, ja que les altres requereixen configuracions addicionals:

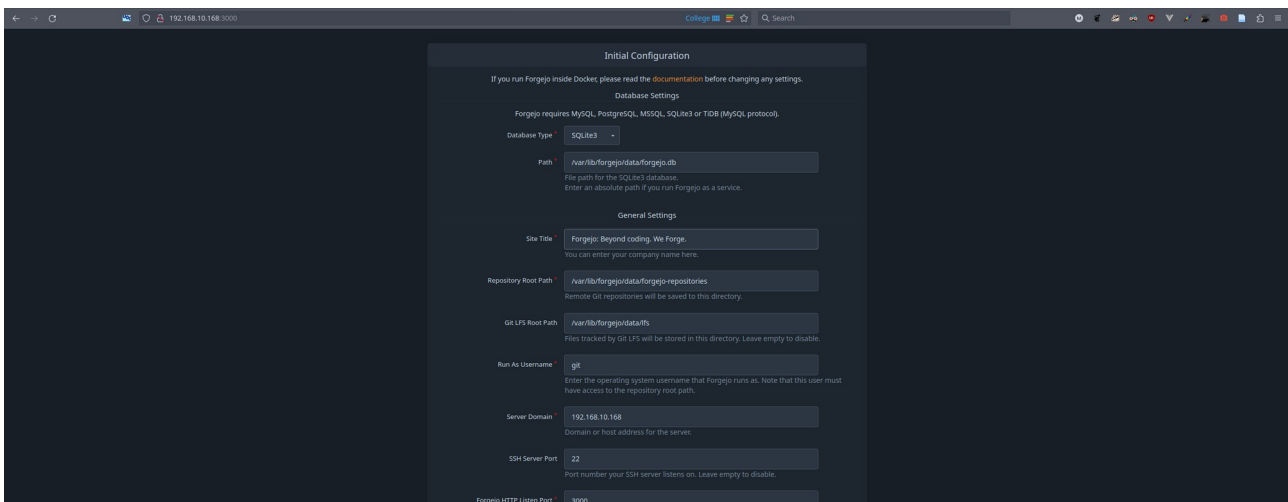
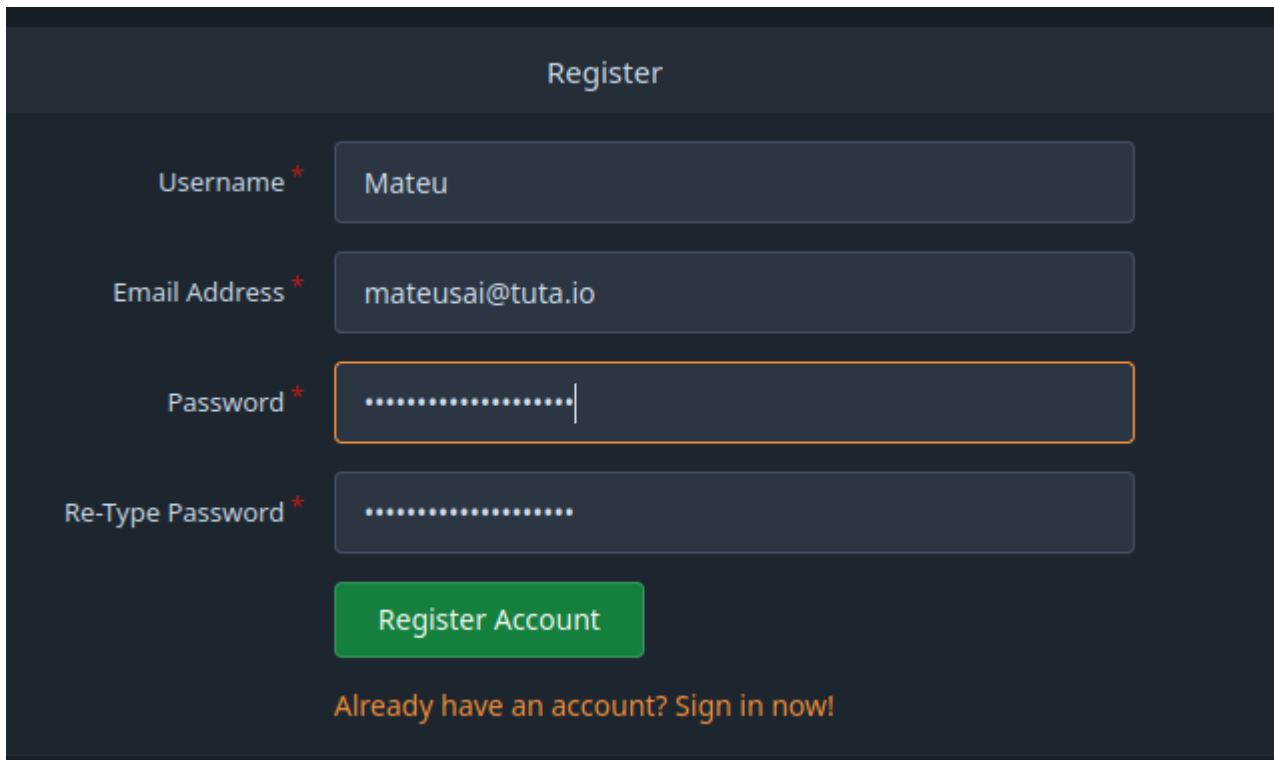


Figura 22: Configuració inicial Forgejo

A continuació ens registrem, el primer usuari tindrà permisos d'administració:



The image shows a registration form titled "Register" on a dark background. It contains four input fields: "Username" with the value "Mateu", "Email Address" with "mateusai@tuta.io", "Password" (masked with dots), and "Re-Type Password" (also masked). A green "Register Account" button is positioned below the fields. At the bottom, there is a link that says "Already have an account? Sign in now!".

Figura 23: Creació primer usuari Forgejo

Aquí podem veure el nou usuari:

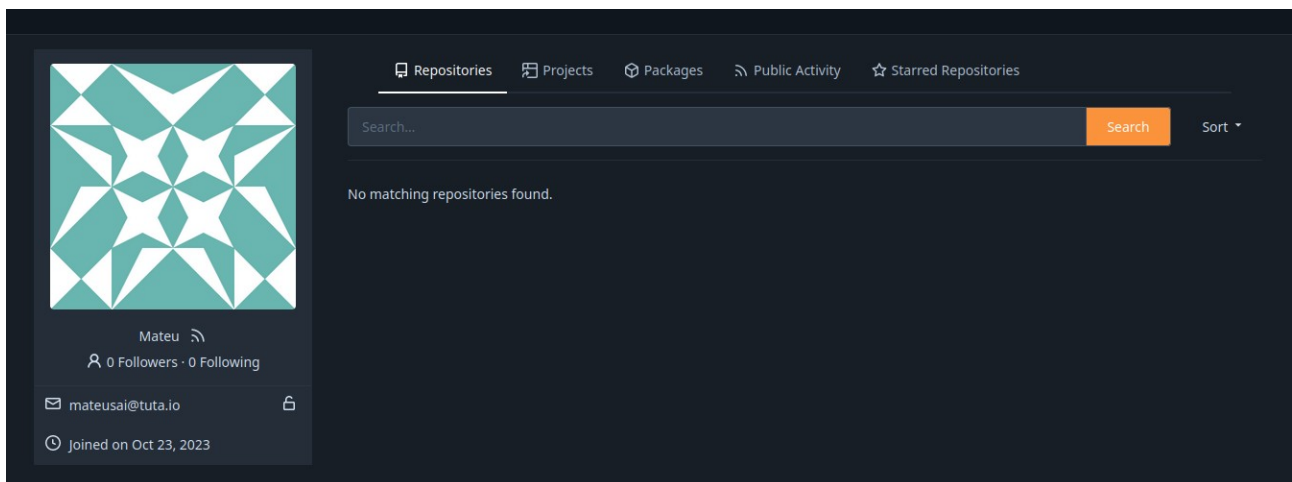


Figura 24: Usuari Forgejo

Tal i com està ara, tothom qui vulgui pot crear un usuari, però volem limitar els usuaris als membres de l'organització. Per això, hem de modificar el fitxer de configuració de Forgejo a `/etc/forgejo/app.ini` i definir `'DISABLE_REGISTRATION'` com a `'true'`, d'aquesta manera només podran crear usuaris els administradors:



```
...  
[service]  
REGISTER_EMAIL_CONFIRM = false  
ENABLE_NOTIFY_MAIL = false  
DISABLE_REGISTRATION = true  
...
```

Perquè els canvis tinguin efecte hem de reiniciar el servidor. Ho podem fer amb `systemctl restart forgejo.service`. Ara ja no surt el botó de registrar:




Figura 25: Forgejo sense botó de registrar

## HTTPS

També habilitarem https. Podríem configurar Forgejo perquè faci servir https directament, però això portaria problemes amb el 'runner' que ha d'executar les accions, ja que no funciona amb certificats signats per un mateix. Per encriptar la connexió farem servir el servidor Apache, configurat anteriorment, com a proxy. Això vol dir que quan ens connectem a Forgejo, els paquets seguiran el següent recorregut.

1. Del navegador de l'usuari al servidor Apache del servidor, a través de una connexió encriptada https.
2. Del servidor Apache al servidor Forgejo, sense encriptar a través de http

I de la mateixa manera, els paquets que rebem de Forgejo, primer passaran pel servidor Apache per encriptar-se.

Per configurar-ho, modificarem la configuració ssl del servidor apache, que està en el fitxer `/etc/httpd/conf.d/ssl.conf`. Afegim les següents línies al final del fitxer:

```
Listen 3002 https  
<VirtualHost *:3002>  
  
SSLEngine on  
SSLHonorCipherOrder on  
SSLCipherSuite PROFILE=SYSTEM  
SSLProxyCipherSuite PROFILE=SYSTEM  
  
SSLCertificateFile /etc/pki/tls/certs/localhost.crt  
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key  
  
ProxyPass "/" "http://127.0.0.1:3000/"  
ProxyPassReverse "/" "http://127.0.0.1:3000/"  
  
</VirtualHost>
```

Primer li diem que escolti en el port 3002, ja que no podem fer servir el mateix port que el servidor Forgejo. Tot seguit creem una secció 'VirtualHost \*:3002' que el que fa és aplicar tots els paràmetres del seu interior a les connexions que tinguin el format especificat, en aquest cas, totes les connexions amb port 3002.

Dintre el 'VirtualHost' tenim les configuracions necessàries per encriptar la connexió amb SSL. Indiquem que volem que SSL estigui actiu i els xifratges que pot fer servir.

Finalment tenim la configuració del proxy, dintre de 'VirtualHost' també, ja que només volem fer servir el proxy en connexions amb port 3002. Simplement li diem que redirigeixi totes les connexions (que es connectin al port 3002) a <http://127.0.0.1:3000/>, l'adreça local del servidor Forgejo.

Reiniciem el servidor Apache i com podem veure, ara podem accedir-hi fent servir https (després d'un avis que ens indica que el certificat està signat per nosaltres mateixos):

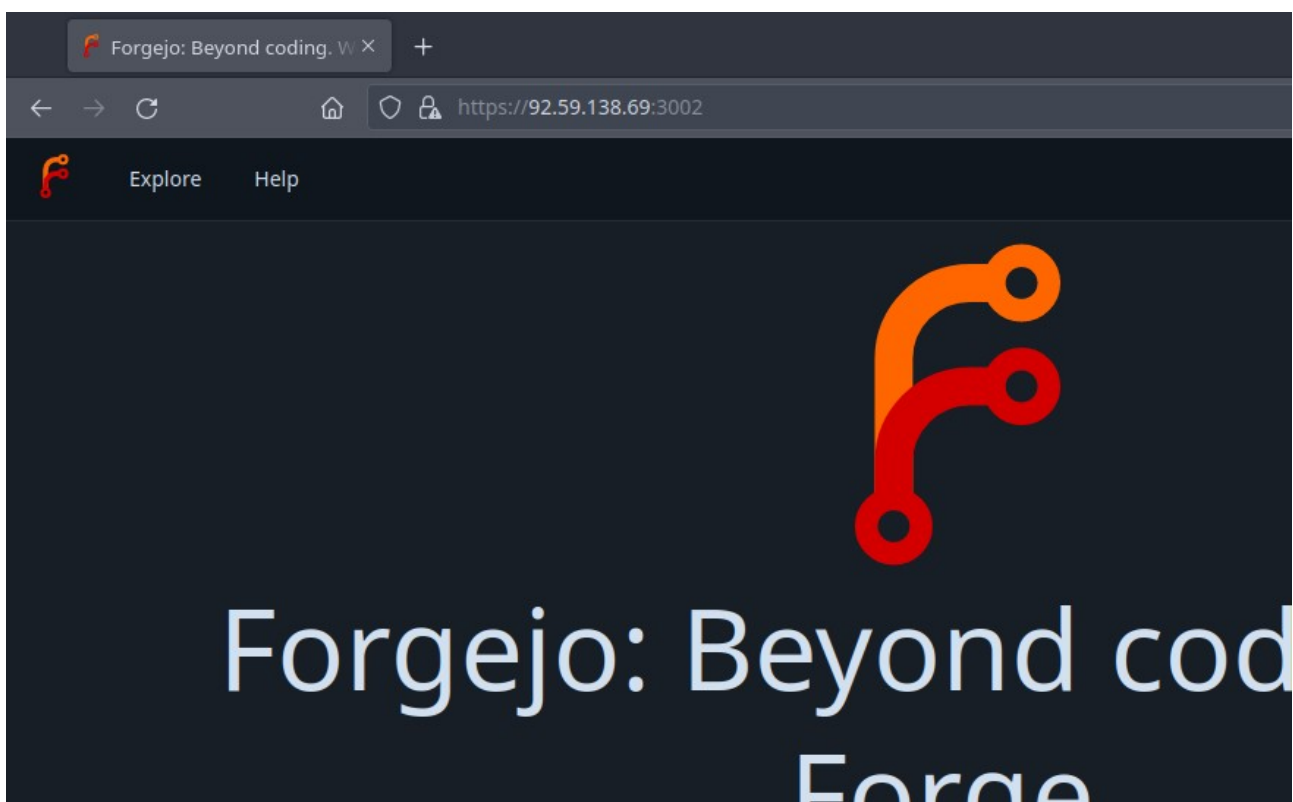


Figura 26: Forgejo https

## Accions

Com hem vist abans, un dels requisits del servidor git és que pugui utilitzar CI/CD. Més tard entrarem en detalls i migrarem l'script CI/CD, però primer hem d'instal·lar el programa que executa les accions, que és un programa a part anomenat 'Forgejo runner'.

La instal·lació de 'Forgejo runner' és fa de manera similar a la de Forgejo. Baixem el programa i el seu resum, comprovem que el programa no estigui corrupte, i li donem permís d'execució:

```
[root@switchroot ~]# wget -O forgejo-runner
https://code.forgejo.org/forgejo/runner/releases/download/v3.0.0/forgejo-runner-3.0.0-linux-arm64
...
2023-11-07 13:27:39 (6.71 MB/s) - 'forgejo-runner' saved [16711680/16711680]
[root@switchroot ~]# wget -O forgejo-runner.asc
https://code.forgejo.org/forgejo/runner/releases/download/v3.0.0/forgejo-runner-3.0.0-linux-arm64.asc
...
2023-11-07 13:28:55 (1.08 MB/s) - 'forgejo-runner.asc' saved [228/228]
[root@switchroot ~]# gpg --keyserver keys.openpgp.org --recv
EB114F5E6C0DC2BCDD183550A4B61A2DC5923710
[root@switchroot ~]# gpg --verify forgejo-runner.asc forgejo-runner
...
gpg: Good signature from "Forgejo <contact@forgejo.org>" [unknown]
gpg:      aka "Forgejo Releases <release@forgejo.org>" [unknown]
...
[root@switchroot ~]# chmod +x forgejo-runner
```

Movem el fitxer a /usr/local/bin, juntament amb l'executable de Forgejo i eliminem el resum:

```
[root@switchroot ~]# mv forgejo-runner /usr/local/bin/forgejo-runner
[root@switchroot ~]# rm forgejo-runner.asc
```

Com que és un executable a part, hem de crear un script de systemd i activar-lo perquè s'executi quan iniciem el servidor. Podem fer servir el mateix script de Forgejo amb algunes modificacions:

```
[root@switchroot ~]# cp /etc/systemd/system/forgejo.service /etc/systemd/system/forgejo-
runner.service
[root@switchroot ~]# vi /etc/systemd/system/forgejo-runner.service
[Unit]
Description=Forgejo Runner
After=syslog.target
After=network.target
Wants=forgejo.service
After=forgejo.service

[Service]
RestartSec=2s
Type=simple
User=git
Group=git
WorkingDirectory==/usr/local/bin/
ExecStart=/usr/local/bin/forgejo-runner daemon
Restart=always
Environment=USER=git HOME=/home/git GITEA_WORK_DIR=/var/lib/forgejo

[Install]
WantedBy=multi-user.target
```

Només he afegit dues línies per indicar que el 'runner' necessita el servei Forgejo i s'ha d'executar un cop Forgejo estigui iniciat, ja que no té sentit que iniciem el runner si Forgejo no s'està executant.

També he canviat el camí cap a l'executable perquè apunti a l'executable del runner en comptes de l'executable de Forgejo. Amb el paràmetre 'daemon' li diem que volem iniciar el 'runner'.

Forgejo-runner necessita un programa que executi les accions en contenidors, farem servir docker.

Per fer servir accions, hem de canviar 'ENABLED' a 'true' de la secció 'actions' de la configuració de Forgejo. També especifiquem URL d'on agafara les accions si no especifiquem una url absoluta. Farem servir <https://code.forgejo.org>, que té accions com 'checkout' que permet modificar repositoris:

```
[actions]
ENABLED = true
DEFAULT_ACTIONS_URL = https://code.forgejo.org
```

Ara hem d'afegir un 'runner', un contenidor on s'executaran les accions. Podem fer servir el de l'[exemple de la documentació](#). Podem obtenir el token des de la configuració de Forgejo:

```
[root@switchroot ~]$ /usr/local/bin/forgejo-runner register --no-interactive --token token --name
runner --instance http://192.168.1.71:3001 --labels docker:docker://node:16-bullseye,self-hosted
INFO Registering runner, arch=arm64, os=linux, version=v3.0.0.
DEBU Successfully pinged the Forgejo instance server
INFO Runner registered successfully.
```

Com que tant el 'runner' com Forgejo estan al mateix sistema, no hi ha cap necessitat de connectar-nos a internet ni d'encriptar la connexió, per això, he fet servir l'adreça ip privada 192.168.1.71 i http.

Movem el fitxer .runner generat a la carpeta /usr/local/bin, ja que aquesta és la carpeta on el servei forgejo-runner el buscara:

```
[mateus@switchroot ~]$ sudo mv .runner /usr/local/bin/
```

Si provem d'iniciar el servei ens donara error perquè l'usuari git no té accés a docker. Per donar-li accés, l'afegim al grup docker:

```
[mateus@switchroot ~]$ sudo usermod -a -G docker git
```

Iniciem i activem el servei:

```
[root@switchroot ~]# systemctl enable forgejo-runner
Created symlink /etc/systemd/system/multi-user.target.wants/forgejo-runner.service →
/etc/systemd/system/forgejo-runner.service.
[root@switchroot ~]# systemctl start forgejo-runner
```

Per comprovar que Forgejo identifiqui el runner correctament, podem anar a l'administració de Forgejo, a 'Actions' → 'Runners':

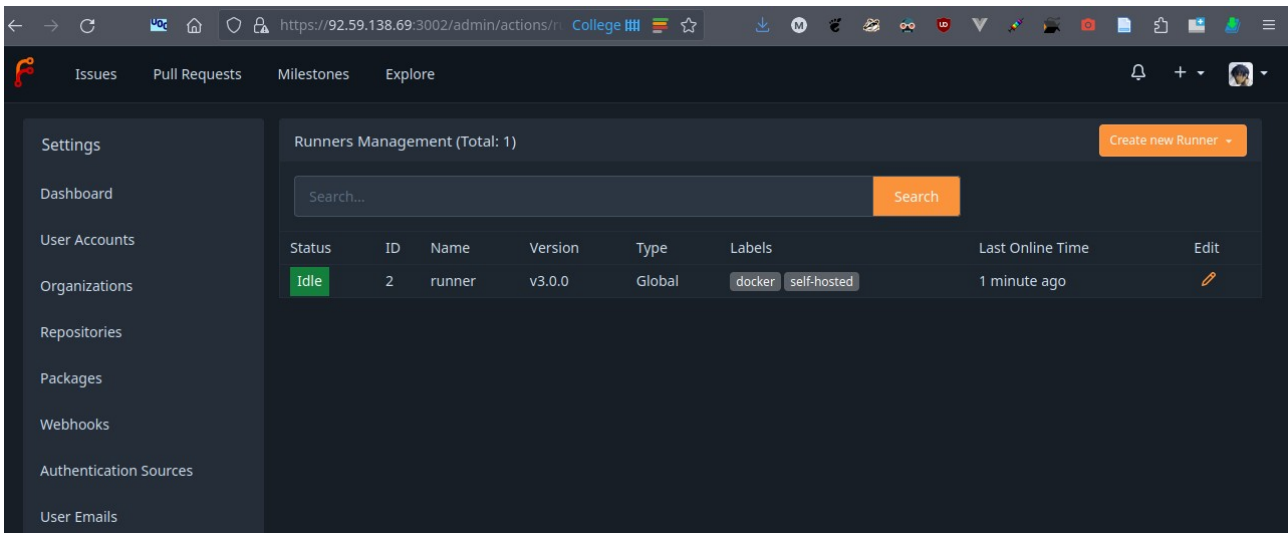


Figura 27: Administració runners Forgejo

A continuació migrarem els repositoris de l'organització al nou servidor.

## Migració

### Repositoris

Primer de tot creem l'organització a Forgejo:

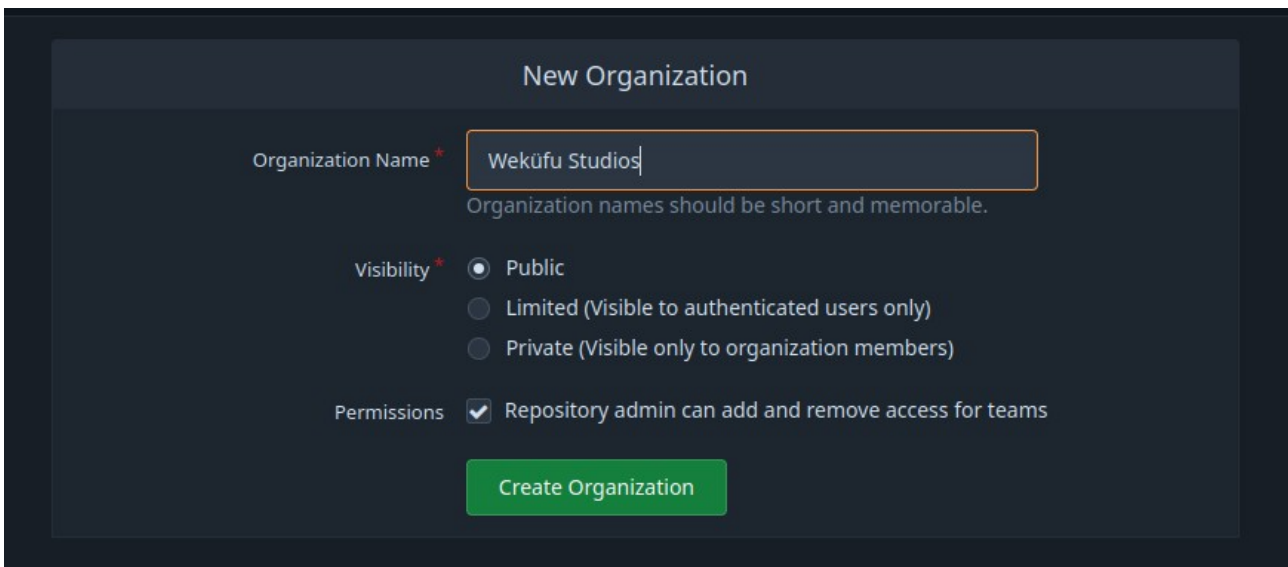


Figura 28: Crear organització Forgejo

Tot seguit, migrem els repositoris de l'organització. Forgejo disposa de una opció per migrar repositoris, només l'hi hem de donar la url. Per exemple, a continuació migro el repositori de la pàgina web:


### Migrate From GitHub

Migrate / Clone From URL \*   
The HTTP(S) or Git 'clone' URL of an existing repository

Access Token  ?  
You can put one or more tokens with comma separated here to make migrating faster because of GitHub API rate limit. WARN: Abusing this feature may violate the service provider's policy and lead to account blocking.

Migration Options  This repository will be a mirror  
 Migrate LFS files

Migration Items  Wiki  
Access Token is required to migrate additional items  
 Labels  Issues  
 Pull Requests  Releases  
 Milestones

Owner \* 

Repository Name \*

Visibility  Make Repository Private

Description

Figura 29: Migrant repositori a Forgejo

Al cap de uns segons, ja tenim el repositori a punt:

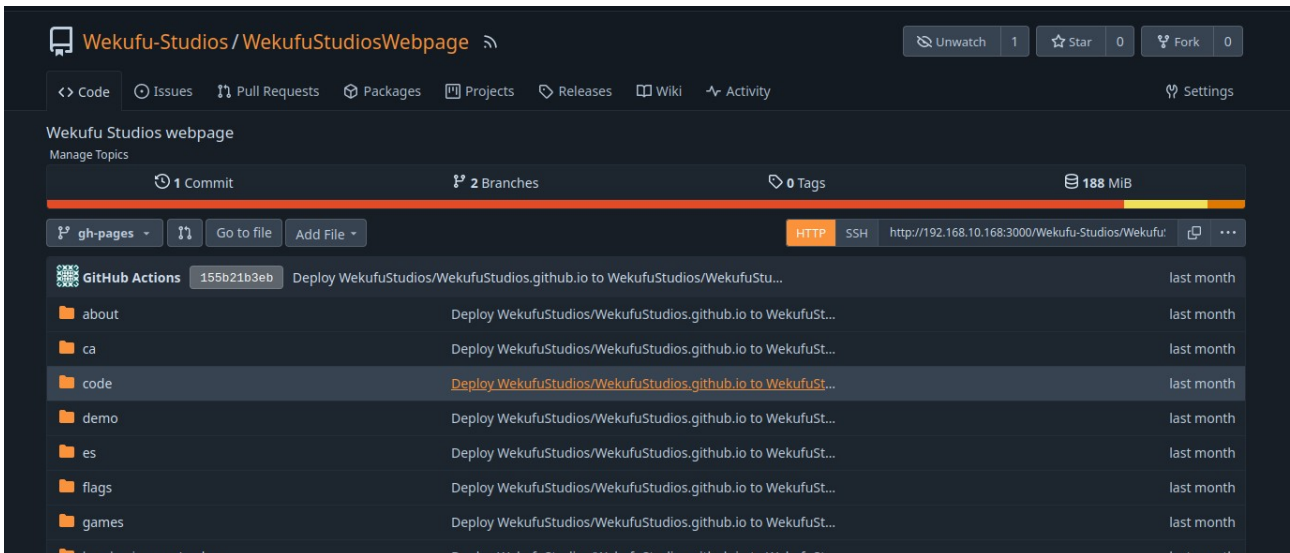


Figura 30: Repositori després de migrar-lo a Forgejo

Per migrar els repositoris privats, necessitem una clau d'accés. Per generar aquesta clau d'accés, hem d'anar a la configuració de GitHub i accedir a la configuració de desenvolupador. Un cop allà, anem a 'Tokens' i en generem un de nou. Li he ficat 'Forgejo' com a nom:

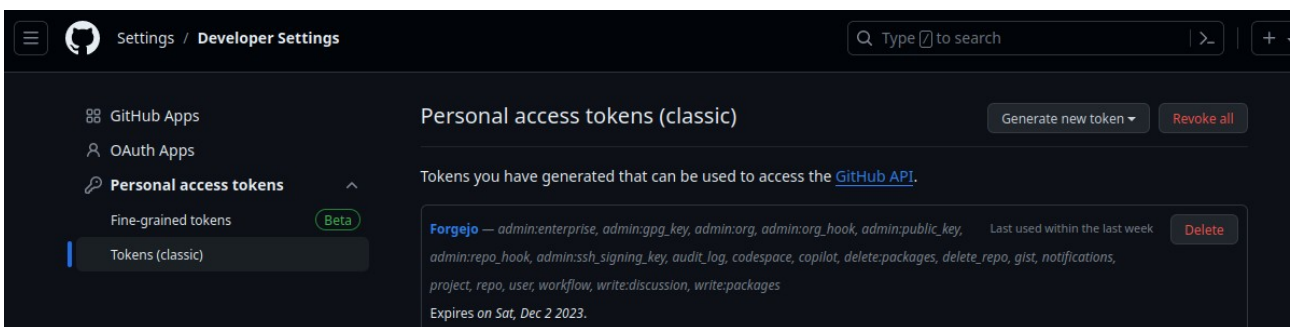


Figura 31: Generar Token a GitHub

Ara migrarem el repositori d'un roguelike en el qual Weküfu Studios està treballant actualment. Com podem veure, el repositori és privat:

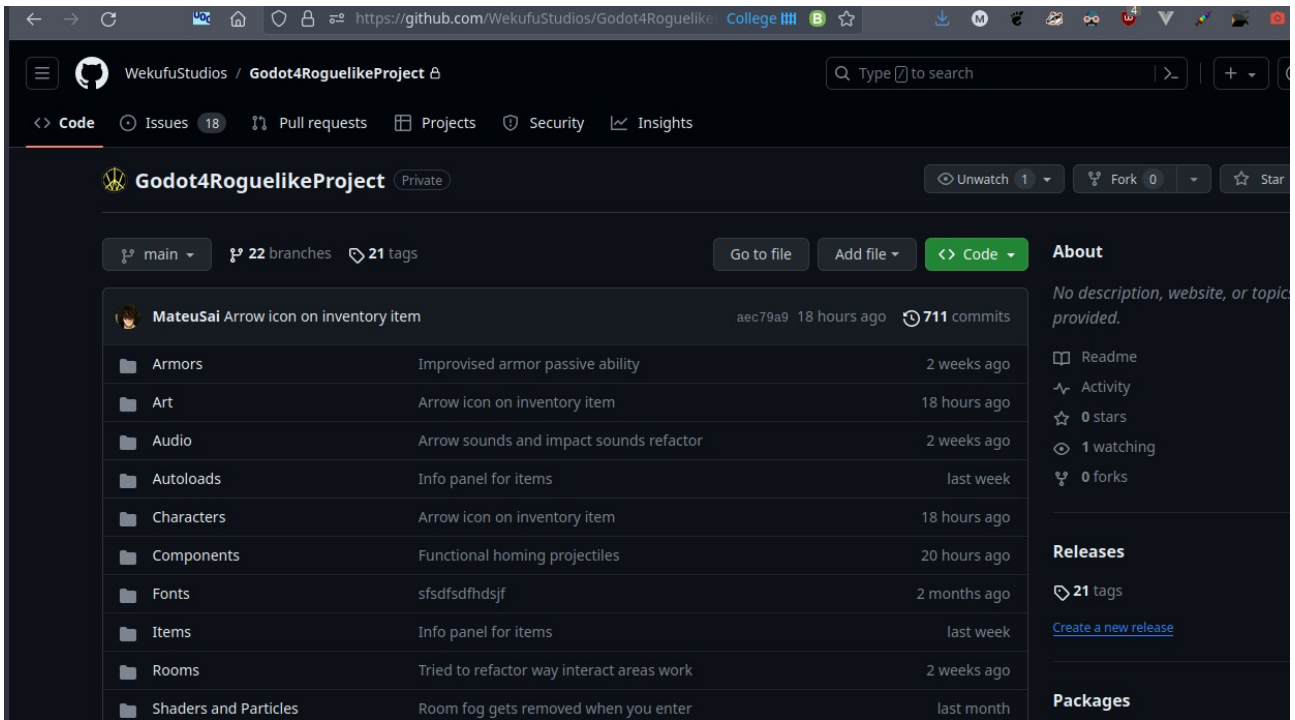
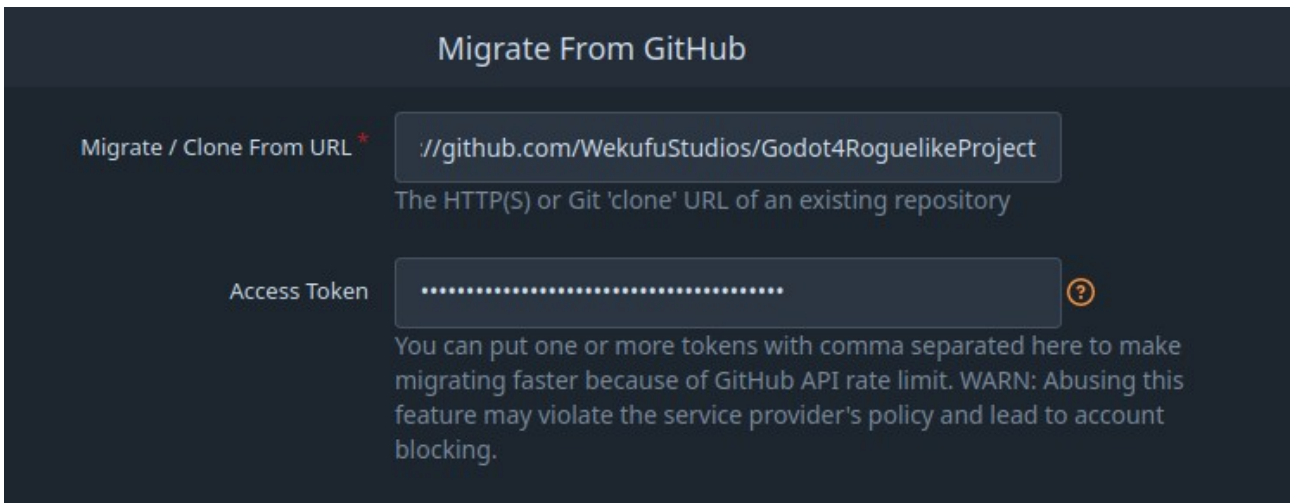


Figura 32: Repositori privat GitHub

A continuació, seguim els mateixos passos que hem seguit per migrar el repositori públic: copiem l'enllaç, especifiquem el propietari, li diem que volem que sigui privat... Però aquesta vegada omplim la casella anomenada 'Access Token' amb el token que hem generat abans, el qual té accés al repositori privat en qüestió:



Al cap d'uns segons, ja tenim el repositori a Forgejo:



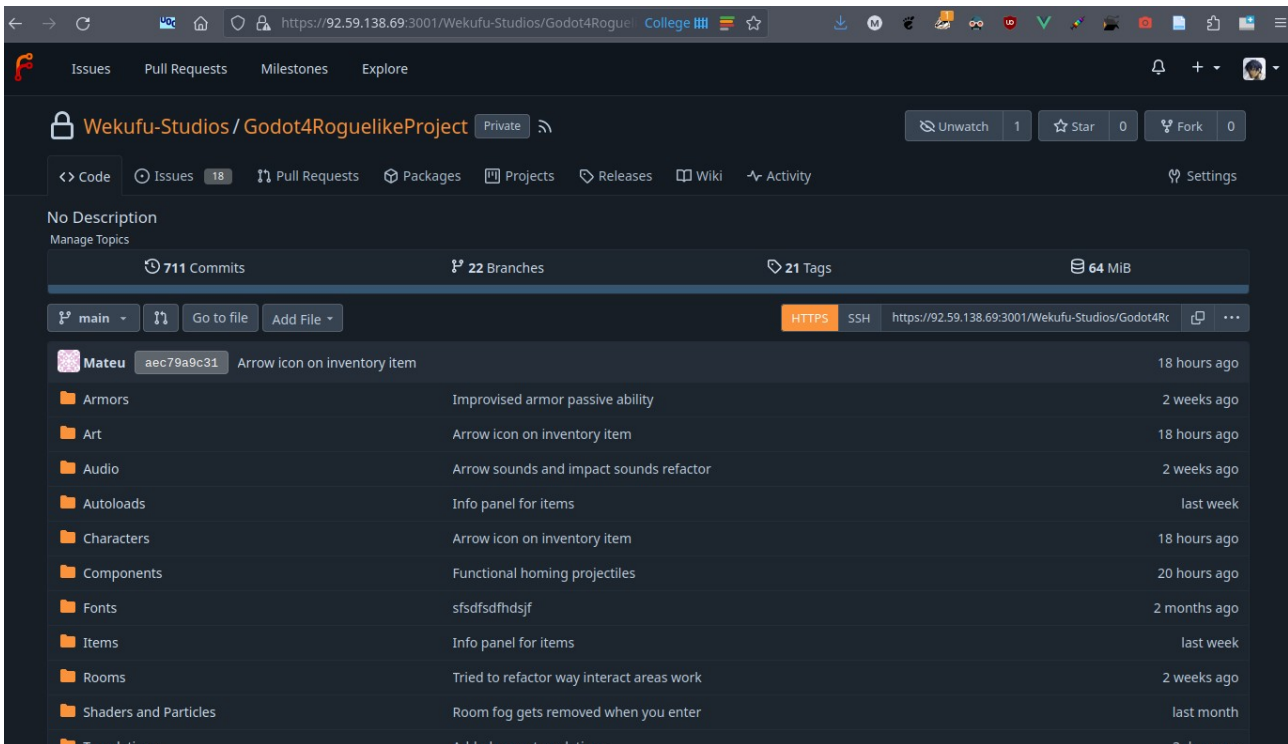


Figura 33: Repositori privat després de migrar-lo de GitHub

## Accions

El repositori de la pàgina web de Weküfu Studios utilitza les accions per compilar la pàgina web i guardar-la en una altre branca del projecte. Primer de tot, hem d'habilitar les accions al repositori, ho podem fer des de la seva configuració:

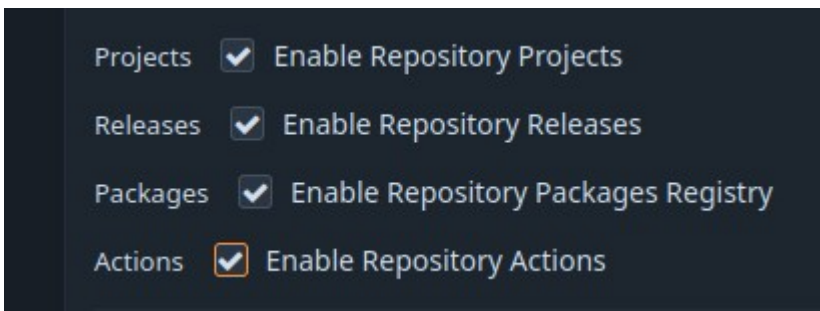


Figura 34: Activem accions al repositori

A GitHub, les accions han d'estar a la carpeta `.github/workflows/`, però a Forgejo han d'estar a la carpeta `.forgejo/workflows/`. Per tant, perquè Forgejo trobi l'acció, hem de canviar el nom del directori a `.forgejo`:

```
[21ms][Code][~/Web/WekufuStudiosWebpage]$ mv .github/ .forgejo/
```

Perquè el fitxer `main.yml` amb les instruccions de l'acció funcioni a Forgejo, li hem de fer algunes modificacions primer. El fitxer `main.yml` que utilitzava abans de la migració, es pot consultar a <https://github.com/WekufuStudios/WekufuStudios.github.io/blob/Code/.github/workflows/main.yml>.

```
jobs:
  build:
    runs-on: docker
    if: github.ref == 'refs/heads/code'
    steps:
      - name: checkout
        uses: actions/checkout@v3.0.0
      - name: build_and_deploy
        uses: http://192.168.1.71:3001/Mateu/zola-deploy-action-unsafe@v0.17.2-2
    env:
      # Target branch
      PAGES_BRANCH: gh-pages
      # Provide personal access token
      #TOKEN: ${{ secrets.TOKEN }}
      # Or if publishing to the same repo, use the automatic token
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
      GITHUB_HOSTNAME: 192.168.1.71:3002
```

1. A la propietat 'runs-on', especifiquem 'docker' perquè l'acció s'executi fent servir el label docker del runner que hem definit abans.
2. En comptes de fer servir [el repositori original](#) per construir i enviar a una nova branca la pàgina web, [he copiat el repositori al servidor Forgejo](#) i he fet una petita modificació perquè permeti fer 'push' sense verificar el certificat (al cap i a la fi, el runner està a la mateixa xarxa local que el servidor Forgejo)
3. Per defecte, l'acció utilitzaria github.com com a host del repositori, per això, he hagut de sobre escriure-ho amb la propietat 'GITHUB\_HOSTNAME'.

## Test

Per finalitzar la secció, farem un push de prova al repositori 'Godot4RoguelikeProject' per comprovar que funcioni correctament:

```

mateus@pop-os ~/G/Godot4RoguelikeProject (main) [1]> vi WizardHut.gd
mateus@pop-os ~/G/Godot4RoguelikeProject (main)> git add -A
mateus@pop-os ~/G/Godot4RoguelikeProject (main)> git commit
Aborting commit due to empty commit message.
mateus@pop-os ~/G/Godot4RoguelikeProject (main) [1]> git commit
[main fc8b9e1] Test commit
file changed, 1 insertion(+), 1 deletion(-)
mateus@pop-os ~/G/Godot4RoguelikeProject (main)> git -c http.sslVerify=false push
https://92.59.138.69:3002/Wekufu-Studios/Godot4RoguelikeProject main
Username for 'https://92.59.138.69:3001': Mateu
Password for 'https://Mateu@92.59.138.69:3002':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: . Processing 1 references
remote: Processed 1 references in total
To https://92.59.138.69:3002/Wekufu-Studios/Godot4RoguelikeProject
aec79a9..fc8b9e1 main -> main

```

He modificat el fitxer WizardHut.gd del projecte que tinc localment al meu ordinador. L'únic que he fet és afegir una tabulació sense importància.

A continuació, afegim els canvis i fem un commit. Un cop tenim el commit fem, podem fer el push. ``-c http.sslVerify=false`` serveix per indicar a git que no ens importa que el certificat estigui signat per nosaltres mateixos. Especifiquem l'adreça del repositori i la branca en la qual volem fer el push. Ens demana l'usuari i la contrasenya. Els escrivim i el push es fa correctament.

Si tornem al repositori, veurem que s'ha actualitzat amb el canvi que hem fet. A continuació podem veure el canvi que he fet en el fitxer:

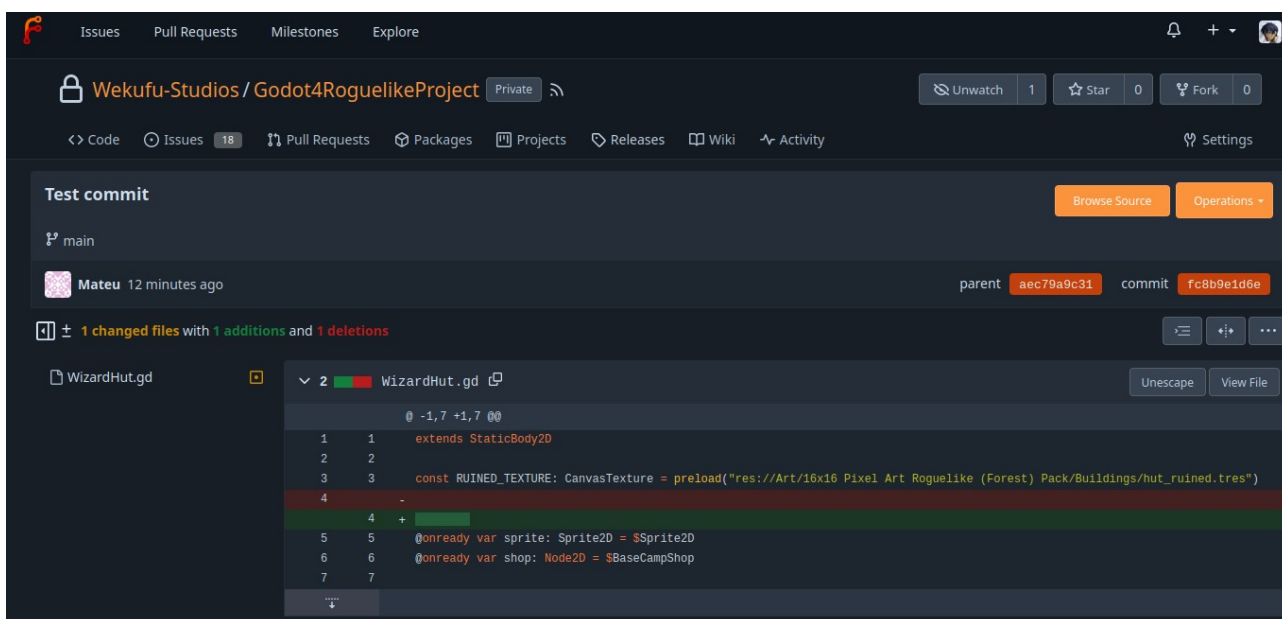


Figura 35: Commit de prova al repositori Godot4RoguelikeProject

## Accions

Per provar el funcionament de l'acció, només hem de fer un push al repositori de la pàgina web (ja que l'acció està configurada per executar-se a cada push). En aquest punt he trobat el problema en el qual li he dedicat més temps de tot el treball.

Al intentar construir l'acció zola (el software que construeix i envia a una altre branca la pàgina web), l'acció fallava amb els següent missatges:

```
Step 10/13 : RUN apt-get update && apt-get install -y wget git
---> Running in c3e563998b8e
Ign:1 http://deb.debian.org/debian stable InRelease
Ign:2 http://deb.debian.org/debian stable-updates InRelease
Ign:3 http://deb.debian.org/debian-security stable-security InRelease
Ign:1 http://deb.debian.org/debian stable InRelease
Ign:2 http://deb.debian.org/debian stable-updates InRelease
Ign:3 http://deb.debian.org/debian-security stable-security InRelease
Ign:1 http://deb.debian.org/debian stable InRelease
Ign:2 http://deb.debian.org/debian stable-updates InRelease
Ign:3 http://deb.debian.org/debian-security stable-security InRelease
Err:1 http://deb.debian.org/debian stable InRelease
Temporary failure resolving 'deb.debian.org'
Err:2 http://deb.debian.org/debian stable-updates InRelease
Temporary failure resolving 'deb.debian.org'
Err:3 http://deb.debian.org/debian-security stable-security InRelease
Temporary failure resolving 'deb.debian.org'
Reading package lists...
W: Failed to fetch http://deb.debian.org/debian/dists/stable/InRelease Temporary failure resolving
'deb.debian.org'
W: Failed to fetch http://deb.debian.org/debian/dists/stable-updates/InRelease Temporary failure
resolving 'deb.debian.org'
W: Failed to fetch http://deb.debian.org/debian-security/dists/stable-security/InRelease Temporary
failure resolving 'deb.debian.org'
W: Some index files failed to download. They have been ignored, or old ones used instead.
Reading package lists...
Building dependency tree...
Reading state information...
E: Unable to locate package wget
E: Unable to locate package git
The command '/bin/sh -c apt-get update && apt-get install -y wget git' returned a non-zero code: 100
```

Gràcies a l'ajuda de l'Earl Warren al servidor de Forgejo de Matrix, he vist que és culpa del sistema dns, el qual per alguna raó no funciona. Passant '--network host' quan es construeix el contenidor, llavors sí que funciona, però, és clar, el runner no ho crida amb aquest paràmetre i sembla que no és pot canviar. He provat moltes coses per intentar arreglar docker, però no he set capaç de trobar una solució. També ho he preguntat al fòrum de Fedora, però al moment d'escriure aquest paràgraf encara no he obtingut resposta: <https://discussion.fedoraproject.org/t/dns-does-not-work-on-docker/97343>.

Així doncs, l'única solució que he trobat es moure el runner al meu ordinador personal en comptes de tenir-lo al servidor, ja que docker funciona correctament allà.

Aquest és el resultat de l'acció un cop solucionat el problema anterior:

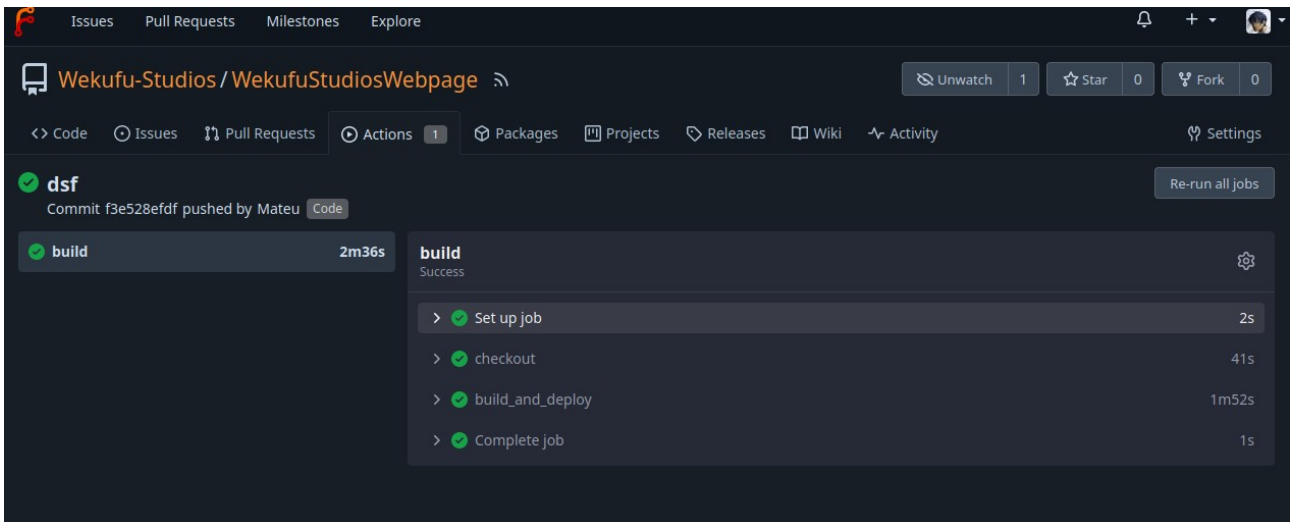


Figura 36: Resultat d'executar l'acció

La pàgina web no haurà canviat perquè no he canviat el seu codi, però podem veure que la branca s'ha sobreescrit:

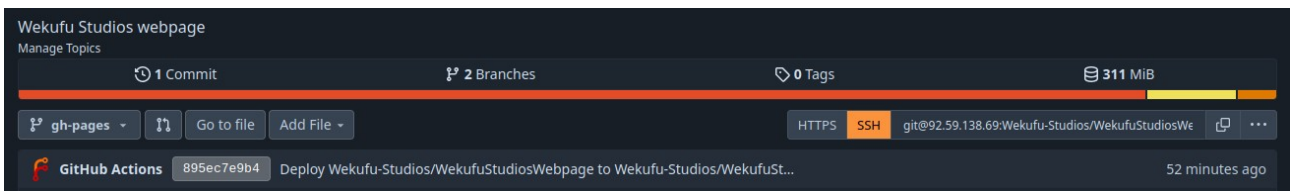


Figura 37: Branca de la pàgina web sobreescrita

## 4 Comunicació

Actualment, Weküfu Studios està fent servir Discord per la comunicació. A més a més de comunicació de text també és necessària comunicació de veu per fer reunions telemàtiques.

### Estàndards disponibles

Bàsicament, tenim 2 estàndards oberts per triar: XMPP i Matrix. Els dos estàndards són descentralitzats, és a dir, no només ens podem comunicar amb els usuaris que facin servir el nostre servidor, si no que també ens podem connectar amb qualsevol altre usuari que faci servir un servidor amb el mateix estàndard.



XMPP (Extensible Messaging and Presence Protocol) va ser desenvolupat el 1998, des de llavors s'ha anat actualitzant i s'hi han afegit moltes extensions. Dues de les extensions més importants són:

- Jingle que ens permet comunicar-nos a través de sessions multimèdia i
- MUC (Multi-User-Chat) que ens permet crear canals amb múltiples usuaris.

XMPP encripta els missatges abans d'enviar-los, de manera que ningú que intercepti els paquets podrà veure el seu contingut.

Centenars de milions d'usuaris utilitzen xmpp a través de serveis com WhatsApp o Zoom.



Igual que xmpp, matrix també encripta els missatges i permet comunicació de veu i vídeo. El que el fa diferent és que permet crear subgrups dintre de grups similar a Discord i té una extensió que permet connectar-se amb Discord i fer de pont entre les dues plataformes.

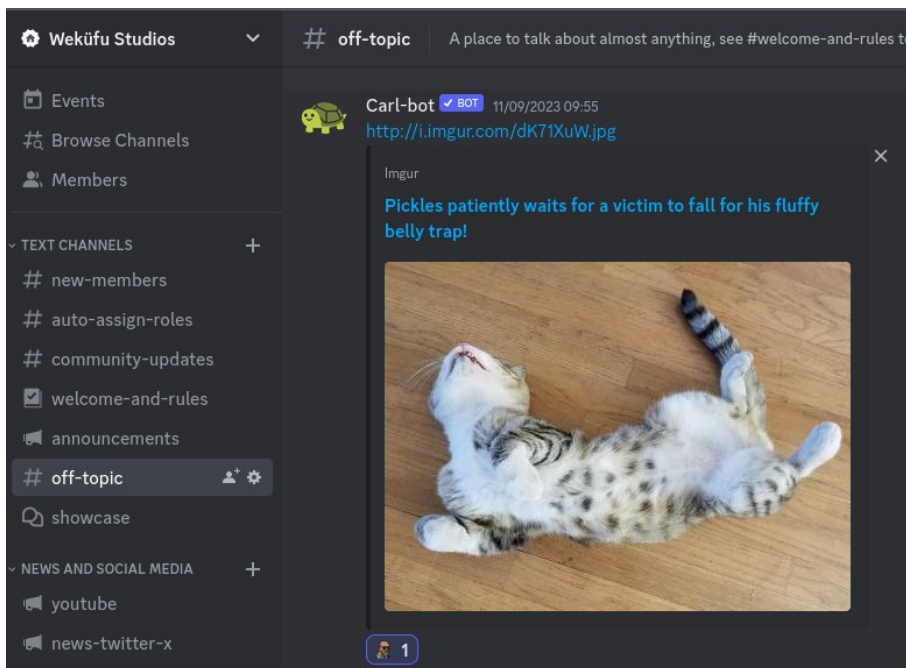


Figura 38: Grup de Discord amb diversos subgrups

Tot i que aquestes dues característiques podrien ser bastant útils, Matrix comparteix moltes metadades entre servidors, entre elles el numero de telèfon i correu. A més a més, Matrix té una historia molt tèrbola, pel que sembla va ser finançat i desenvolupat per una company d'«Israel» anomenada Amdocs amb connexions amb el servei d'intel·ligència «israelià». A causa d'aquestes raons, instal·larem un servidor xmpp.

## Servidors XMPP

Hi ha bastants servidors xmpp disponibles, des de l'[apartat 'software' de la seva pàgina web](#) podem veure els diferents servidors i les característiques de cada un. Instal·larem Prosody IM, ja que és un dels servidors més complets.

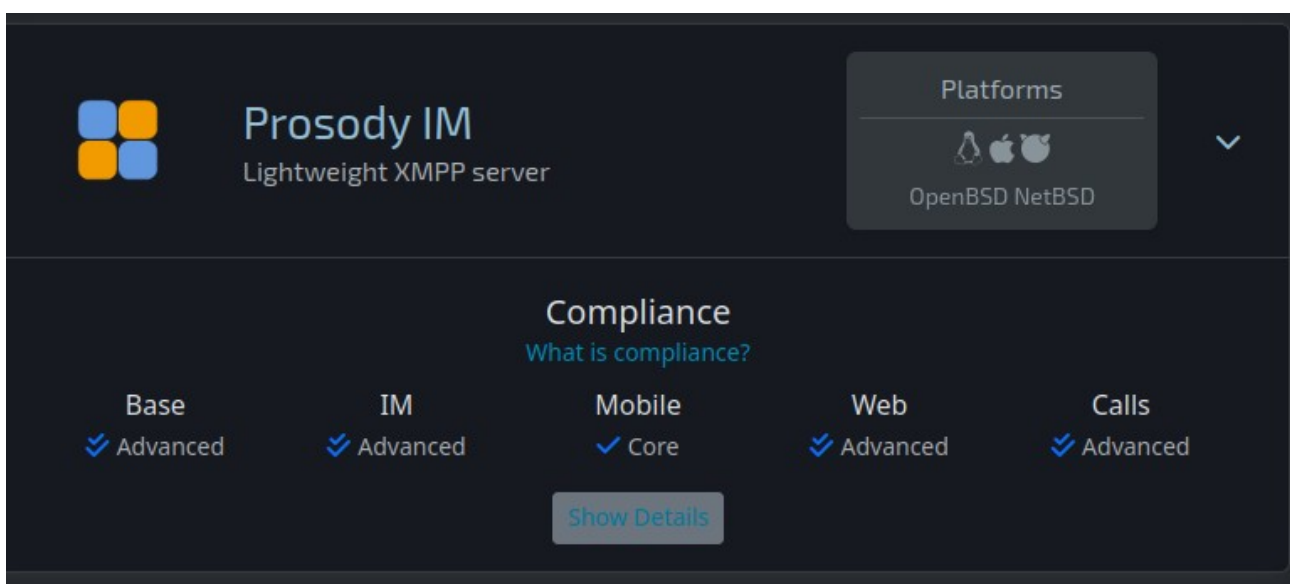


Figura 39: Característiques Prosody IM

## Instal·lació i configuració

Fedora disposa d'un paquet amb Prosody IM, l'instal·lem amb dnf:

```
[root@switchroot ~]# dnf install prosody
...
Complete!
```

La configuració està a /etc/prosody/prosody.cfg.lua. Hem de configurar el domini, farem servir l'IP pública:

```
...
VirtualHost "92.59.138.69"
...
```

Podem fer servir `prosodyctl check config` per assegurar-nos que no hi ha cap problema amb la configuració:

```
[root@switchroot ~]# prosodyctl check config
Checking config...
The following configuration files have been loaded:
- /etc/prosody/conf.d/localhost.cfg.lua
- /etc/prosody/conf.d/example.com.cfg.lua
- /etc/prosody/prosody.cfg.lua
Done.

All checks passed, congratulations!
```

Per crear un nou usuari fem servir la comanda `prosodyctl adduser`, especificant el nom d'usuari i el domini:

```
[root@switchroot ~]# prosodyctl adduser mateu@92.59.138.69
Enter new password:
Retype new password:
```

XMPP utilitza https per encriptar les connexions, així que necessitem un certificat. Prosody ens proporciona una comanda per generar un certificat signat per nosaltres mateixos i la seva clau:



```
[root@switchroot certs]# prosodyctl cert generate 92.59.138.69
Choose key size (2048):
Key written to /var/lib/prosody/92.59.138.69.key
Please provide details to include in the certificate config file.
Leave the field empty to use the default value or '.' to exclude the field.
countryName (GB):
localityName (The Internet):
organizationName (Your Organisation):
organizationalUnitName (XMPP Department):
commonName (92.59.138.69):
emailAddress (xmpp@92.59.138.69):

Config written to /var/lib/prosody/92.59.138.69.cnf
Certificate written to /var/lib/prosody/92.59.138.69.crt
```

S'ha generat un certificat i la seva configuració i s'han guardat a la carpeta /var/lib/prosody. Però Prosody busca el certificat a la carpeta /etc/pki/prosody/, per això, hem de copiar aquest fitxers a /etc/pki/prosody/:

```
[root@switchroot prosody]# cp /var/lib/prosody/92.59.138.69.cnf ./92.59.138.69.cnf
[root@switchroot prosody]# cp /var/lib/prosody/92.59.138.69.crt ./92.59.138.69.crt
[root@switchroot prosody]# cp /var/lib/prosody/92.59.138.69.key ./92.59.138.69.key
[root@switchroot prosody]# ls
92.59.138.69.cnf 92.59.138.69.key localhost.cnf localhost.key openssl.cnf
92.59.138.69.crt GNUmakefile localhost.crt makefile
```

Per defecte els fitxers són propietat de root i estan en el grup root, però la documentació de prosody ens diu que han d'estar en el grup prosody. Per canviar-ho, executem chown:

```
[root@switchroot prosody]# chown root:prosody 92.59.138.69.*
```

Executem `prosody check certs` per estar segurs que pot trobar el certificat:

```
[root@switchroot prosody]# prosodyctl check certs
Checking certificates...
Checking certificate for 92.59.138.69
Certificate: /etc/pki/prosody//92.59.138.69.crt
Checking certificate for localhost
Certificate: /etc/pki/prosody//localhost.crt
Not valid for client connections to localhost.
Not valid for server-to-server connections to localhost.

For more information about certificates please see https://prosody.im/doc/certificates
Problems found, see above.
```

No hi ha problemes amb el certificat de 92.59.138.69. Sí que hi ha problemes amb el certificat de localhost, però no ens afecta, ja que no farem servir el domini localhost.

Per defecte, no podem fer trucades. Per habilitar aquesta opció, primer necessitem un servidor TURN, un servidor que s'encarregue de transmetre imatges i so en temps real. Tenim varies opcions de servidors TURN de codi obert, el que instal·larem serà coturn, ja que la documentació ens diu que ha estat provat i funciona bé amb prosody.

## Instal·lació coturn i integració amb Prosody

Hi ha un paquet amb coturn disponible als repositoris de Fedora, així que l'instal·lem fent servir dnf:

```
[root@switchroot ~]# dnf install coturn
...
Complete!
```

La configuració de coturn està a /etc/coturn/turnserver.conf. Definim el domini, i una clau que farem servir per connectar-lo amb prosody:

```
Realm=92.59.138.69
use-auth-secret
static-auth-secret=secret
```

Ara anem al fitxer de configuració de prosody i activem el mòdul turn\_external, el mòdul que permet a prosody connectar-se amb un servidor turn i oferir així trucades de veu. També especifiquem l'adreça i el secret que hem definit a la configuració de coturn:

```
modules_enabled = {
    ...
    "turn_external"; -- Provide external STUN/TURN service for e.g. audio/video calls
    ...
}
...
turn_external_host = "127.0.0.1"
turn_external_secret = "secret"
```

Com que el servidor turn està al mateix servidor que prosody, podem fer servir l'adreça localhost o 127.0.0.1 directament per connectar-nos a ell.

Per acabar, iniciem coturn i l'activem perquè s'iniciï automàticament cada vegada que encenem el servidor:

```
[root@switchroot ~]# systemctl enable coturn
Created symlink /etc/systemd/system/multi-user.target.wants/coturn.service →
/usr/lib/systemd/system/coturn.service.
[root@switchroot ~]# systemctl start coturn
```

## Posada en marxa

Per activar el servei, utilitzem systemctl com hem fet amb els altres serveis:

```
[root@switchroot ~]# systemctl enable prosody
Created symlink /etc/systemd/system/multi-user.target.wants/prosody.service →
/usr/lib/systemd/system/prosody.service.
[root@switchroot ~]# systemctl start prosody
```

## Tests

He creat un altre usuari anomenat 'socrates' per provar que tot funcioni bé. He fet servir la mateixa comanda que hem fet servir per crear el primer usuari.

Per accedir al servidor xmpp, podem fer servir un dels molts clients disponibles. Per provar el servidor farem servir [Pidgin](#). Afegim un nou compte i omplim les dades que ens demanen: usuari, domini i contrasenya:

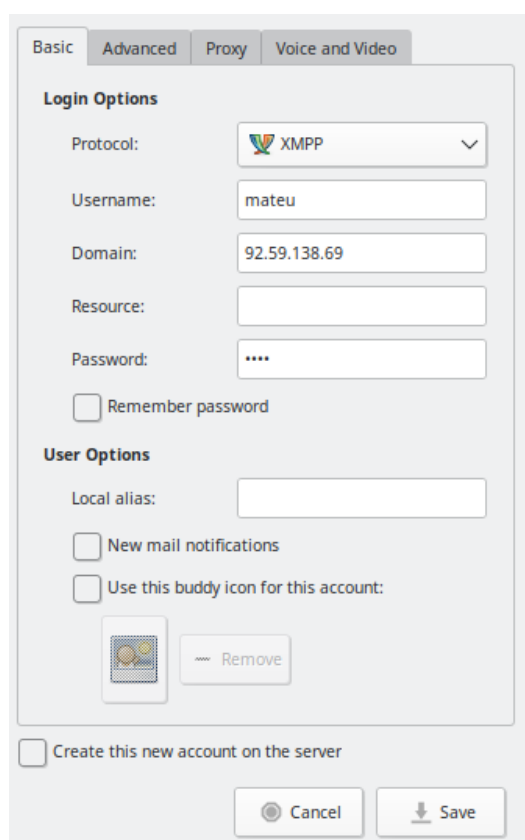


Figura 40: Afegir comte a Pidgin

En el meu telèfon mòbil he instal·lat [Monocles Chat](#), un client xmpp per a Android, i he iniciat sessió amb l'usuari Socrates. Per provar el funcionament, he enviat un missatge amb l'usuari mateu a l'usuari socrates:

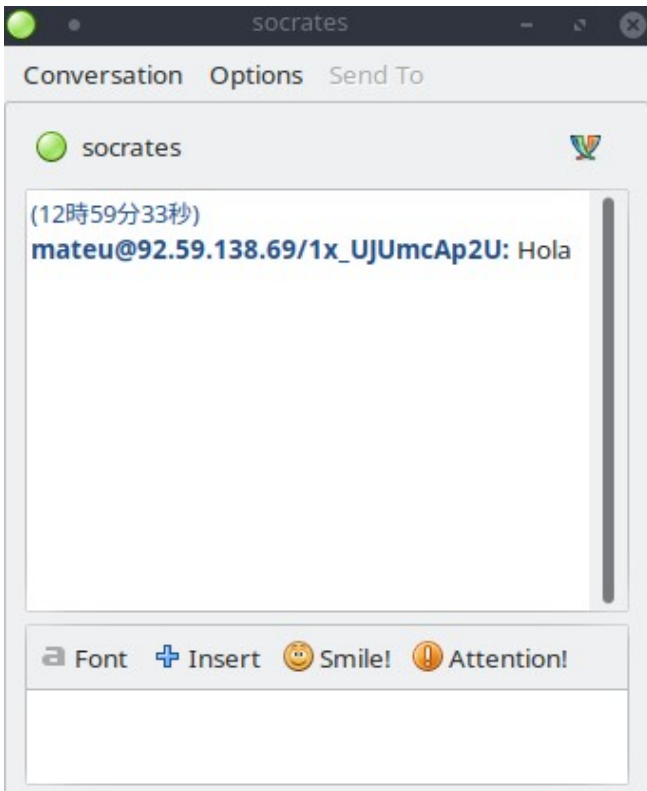


Figura 41: Usuari mateu envia missatge a usuari socrates fent servir Pidgin

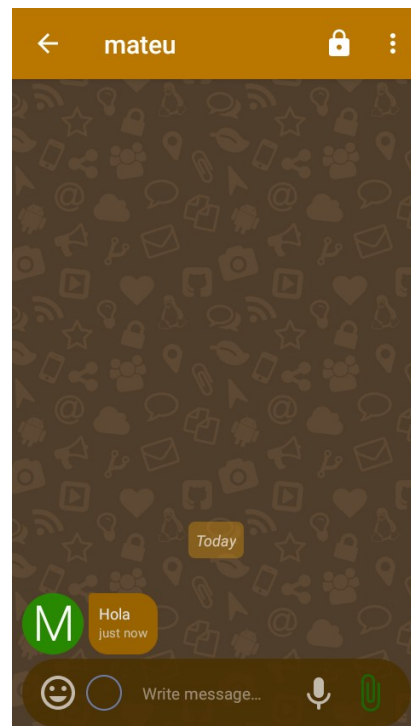


Figura 42: L'usuari socrates rep el missatge enviat per l'usuari mateu a l'aplicació Monocles Chat

Quan a trucades de veu, sembla que a Pidgin no funcionen i no he pogut fer servir cap altre programa d'escriptori. Però fent servir Monocles Chat amb els dos usuaris sí que funciona.

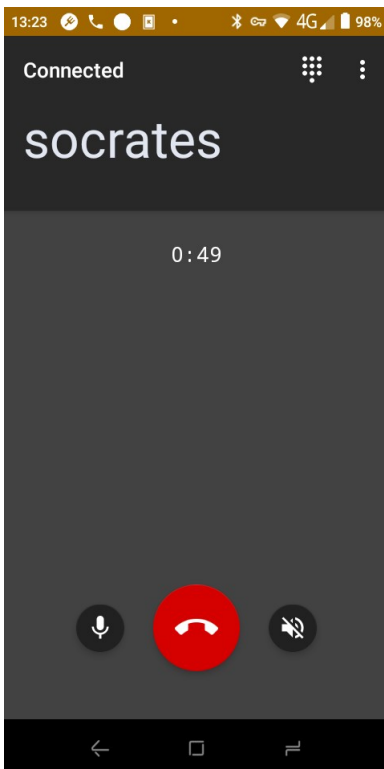


Figura 43: Trucada de veu fent servir Monocles Chat

## 5 Notes

Weküfu Studios utilitza notes com una forma ràpida de definir mecàniques i elements de disseny dels seus videojocs. Per exemple, per definir el comportament de cada enemic.

### Opcions disponibles

#### Joplin

Joplin ens permet crear notes en format md. Les notes estan dintre un 'bloc de notes'. Podem ficar 'blocs de notes' dintre altres blocs de notes, creant una jerarquia, cosa que facilitaria l'ordenació de les notes.

Una característica de Joplin és que permet assignar tags a les notes, de manera que podem trobar totes les notes amb un mateix tag fàcilment.

Les notes es guarden en una base de dades sqlite.

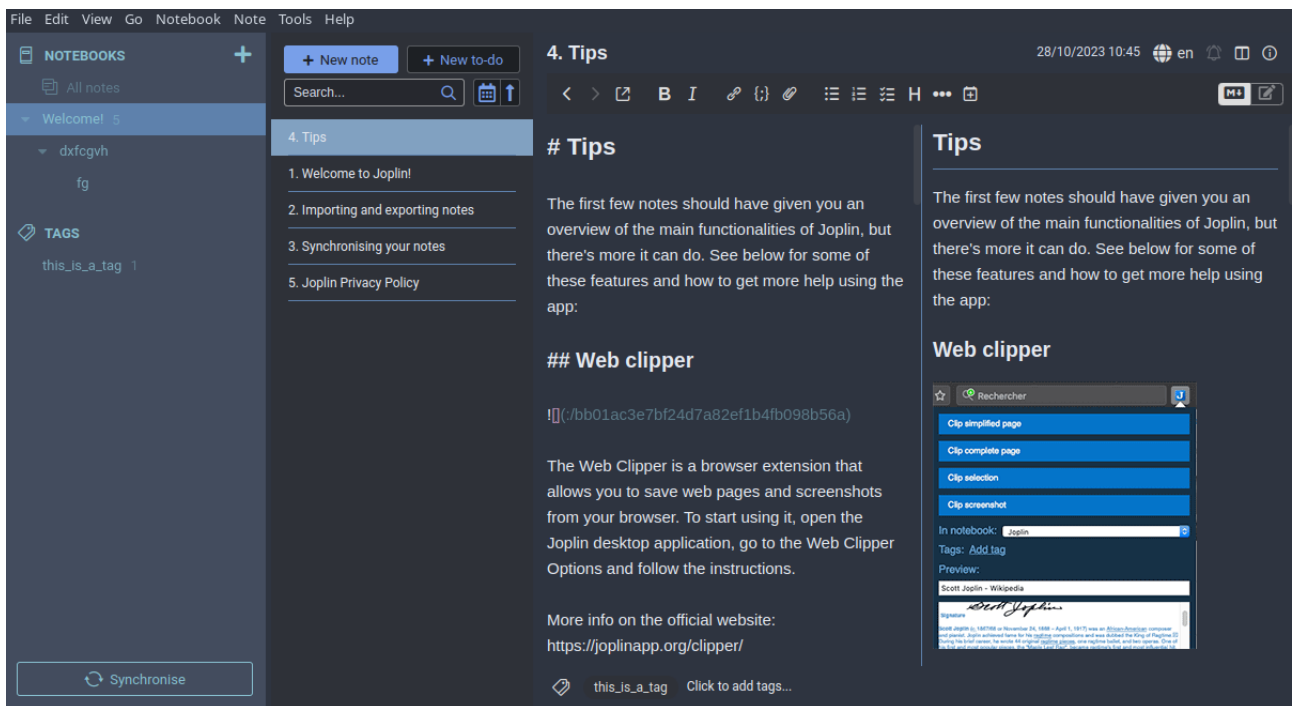


Figura 44: Captura de pantalla Joplin

Podem fer servir l'aplicació d'escriptori per sincronitzar les notes a Nextcloud, però les notes no es guarden a Nextcloud amb el seu nom, si no que es guarden amb una mena d'id, l'interior de la nota conte moltes metadades, i no podríem editar les notes des de Nextcloud, ho hauríem de fer només des de l'aplicació d'escriptori, el que descarta la possibilitat de col·laborar en temps real.

### Nextcloud Notes

Nextcloud té una extensió que afegeix una nova pestanya per escriure notes. Ens guarda les notes en format md, fàcilment editables. Malauradament, aquesta extensió té moltes limitacions:

- Es poden categoritzar les notes en subcarpetes, però per a fer-ho has d'anar a la pestanya de fitxers i crear la carpeta manualment.
- Si creem subcarpetes dintre subcarpetes, l'extensió les reconeix, però les mostra totes dintre la primera subcarpeta. I tampoc ens deixa crear notes dintre la segona subcarpeta si no ho fem manualment des de la pestanya de fitxers o primer creem la nota i després la movem.



Trilium és un altre software que ens permet escriure notes. Permet crear un servidor i sincronitzar els clients amb aquest servidor. Les seves característiques principals són:

- Guarda les notes en format html, fent servir una base de dades.
- Permet crear una jerarquia de notes, fins i tot les carpetes són notes. Cada nota mostra les seves subnotes al final.
- Podem crear enllaços entre notes, facilitant la navegació.
- Permet afegir atributs a les notes i crear mapes de relacions i diagrames, entre altres coses.
- No necessitem cap client, podem editar les notes des del navegador.
- Permet crear els teus propis scripts per afegir funcionalitat.

## Instal·lació i configuració

Instal·larem Trilium, ja que és el servei amb més característiques i es pot utilitzar directament des del navegador, sense necessitat de cap client especialitzat.

Trilium no disposa de binari en arquitectura arm, així que l'instal·larem a través de un contenidor docker.

A docker hub hi ha un [contenidor Trilium](#) que funciona en arquitectures arm. El baixem amb docker pull:

```
[root@switchroot ~]# docker pull zadam/trilium
```

Creem la carpeta trilium-data al directori root. Farem servir aquesta carpeta per guardar les dades de Trilium.

```
[root@switchroot ~]# pwd
/root
[root@switchroot ~]# mkdir trilium-data
```

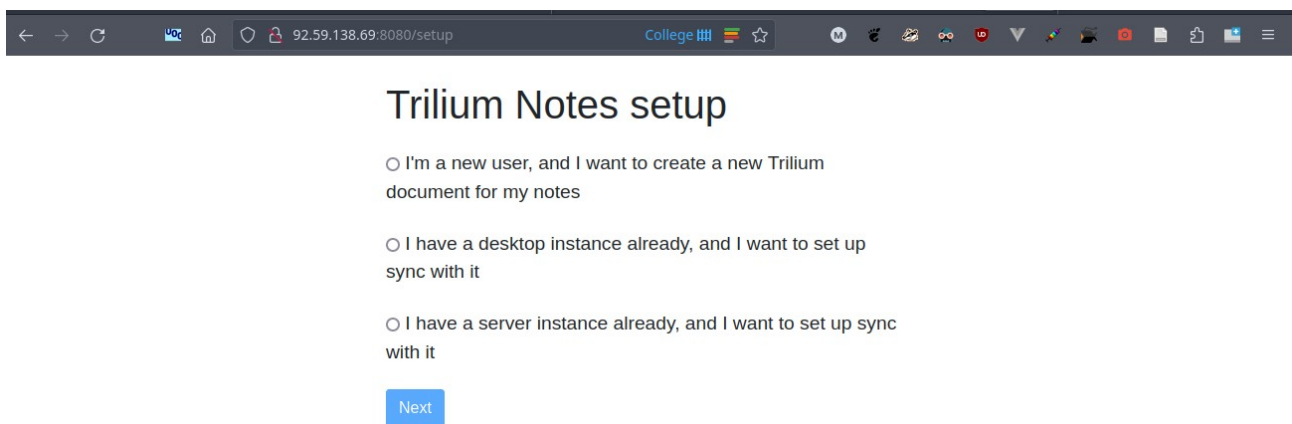
Un cop tenim el contenidor instal·lat, el podem executar amb `docker run`, especificant el nom del contenidor, zadam/trilium. Amb l'opció -p li indiquem a docker que redirigeixi els paquets entrants al port 8080 cap al port 8080 del contenidor docker, 8080 és el port que fa servir Trilium. També li donem un nom amb el paràmetre -name, i amb l'opció -v li diem que passi els fitxers de la

carpeta /home/node/trilium-data del contenidor a la carpeta /root/trilium-data del sistema. D'aquesta manera tindrem accés a les dades de trilium des de fora del contenidor.

```
[root@switchroot ~]# docker run --name=trilium -p 8080:8080 -v /root/trilium-data:/home/node/trilium-data  
...  
App HTTP server starting up at port 8080  
Listening on port 8080
```

Executant aquesta comanda, ja podem accedir a trilium anant al port 8080. Però, igual que els altres serveis, hauríem de configurar un script de systemd perquè iniciï el servei automàticament. L'script és similar al que hem fet abans pel runner de forgejo, es pot consultar a l'annex.

Si ara anem al port 8080, veurem la pantalla de configuració de Trilium:



*Figura 45: Pàgina de configuració inicial Trilium*

Seleccionem la primera opció indicant que som un nou usuari i cliquem 'Next'. Al cap d'uns segons, Trilium ens demana que creem una contrasenya. Un cop l'hem definida, ens redirigeix a la pantalla de login:

# Trilium login

Password

Remember me

Login

Figura 46: Pàgina login trilium

Fiquem la contrasenya que hem creat en el pas anterior i entrem. Ens trobem amb les notes creades per defecte:

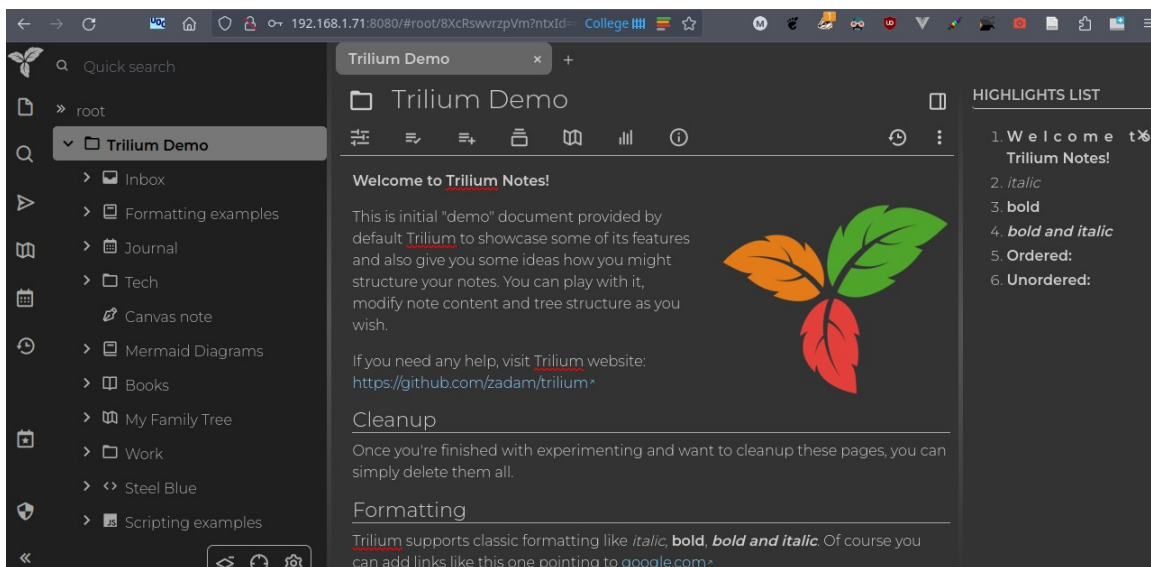


Figura 47: Notes Trilium creades per defecte

## HTTPS

Quan he entrat la contrasenya anteriorment, he fet servir l'adreça IP privada del servidor, de manera que els paquets no han sortit de la LAN, però si volem que el servei sigui accessible des de tot internet, hem d'habilitar https, ja que si no, tothom que interceptes els paquets, podria veure la contrasenya.

Per habilitar https hem de modificar el fitxer config.ini que tenim dintre la carpeta trilium-data. Però primer necessitem un certificat i una clau. Farem servir els mateixos que hem fet servir per Forgejo. Els copiem al directori trilium-data:

```
[root@switchroot ~]# cp /var/lib/forgejo/custom/cert.pem trilium-data/cert.pem
[root@switchroot ~]# cp /var/lib/forgejo/custom/key.pem trilium-data/key.pem
```



Entrem a config.ini i canviem https a 'true' i especifiquem els camins del certificat i la seva clau. Fem servir els camins del contenidor, no els del servidor, ja que si no, no trobaria els fitxers. Com hem vist abans, el directori trilium-data del contenidor està a /home/node/trilium-data:

```
https=true
certPath=/home/node/trilium-data/cert.pem
keyPath=/home/node/trilium-data/key.pem
```

Guardem els canvis del fitxer i reiniciem Trilium:

```
[root@switchroot ~]# systemctl restart trilium
```

Ara ja podem accedir a Trilium a través de https amb un connexió encriptada.

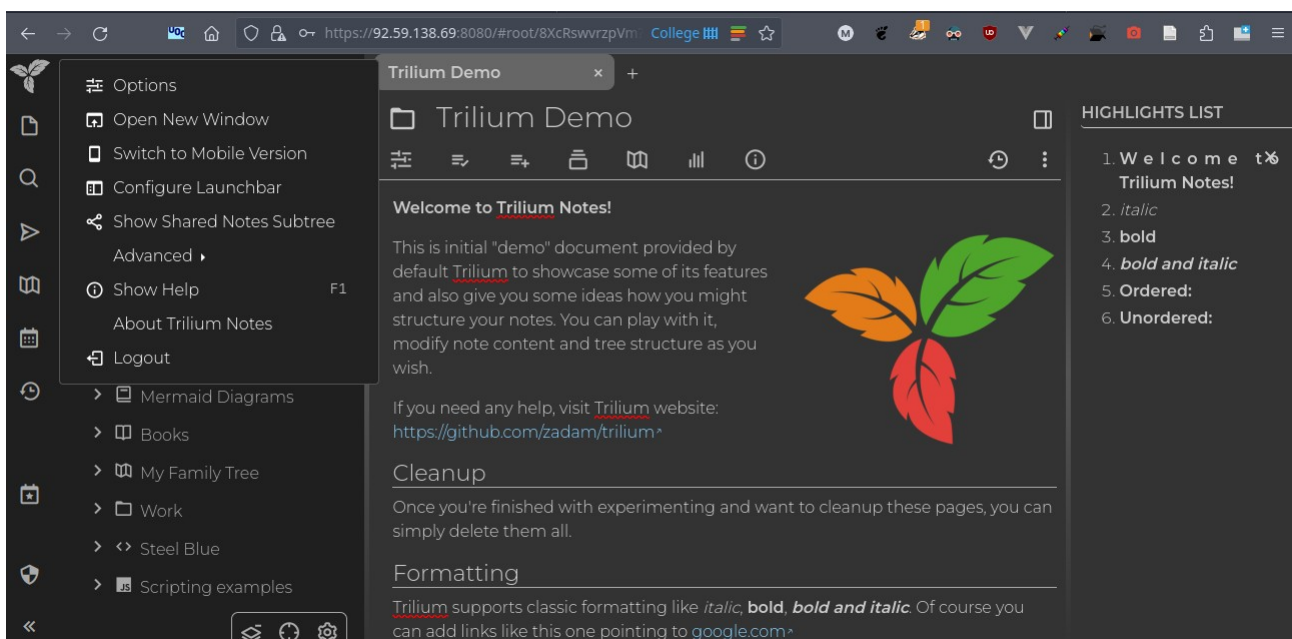


Figura 48: Accés a Trilium fent servir https

## Migració

Per migrar les notes que actualment Weküfu Studios té en format .md, simplement podem importar els fitxers. Primer creem una nota anomenada 'Erebus', el nom d'un dels videojocs de Weküfu Studios, que contindrà les altres notes de carpeta i tot seguit fem clic dret a la nota i seleccionem 'Import into note'. Seleccionem les notes que volem importar i cliquem 'import':

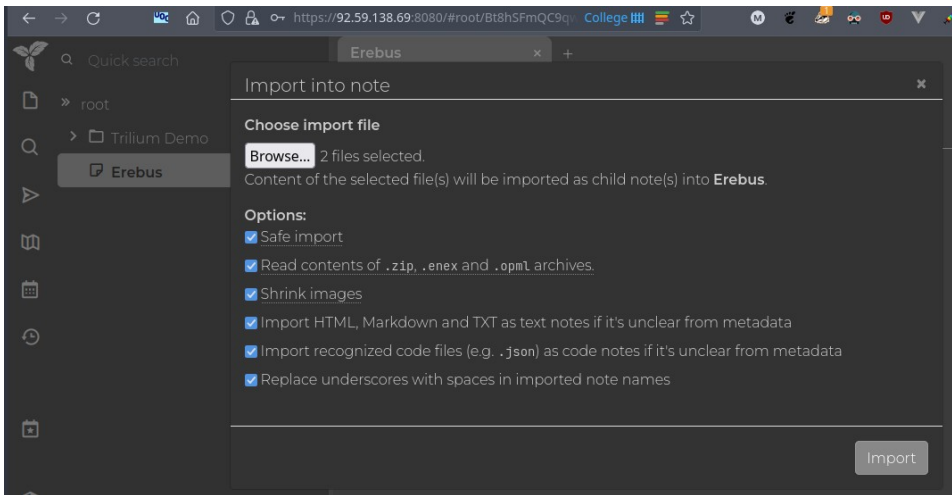


Figura 49: Importar notes a Trilium

Al cap d'uns instants ja tindrem les notes a Trilium. Trilium passa les notes de format md a format html automàticament:

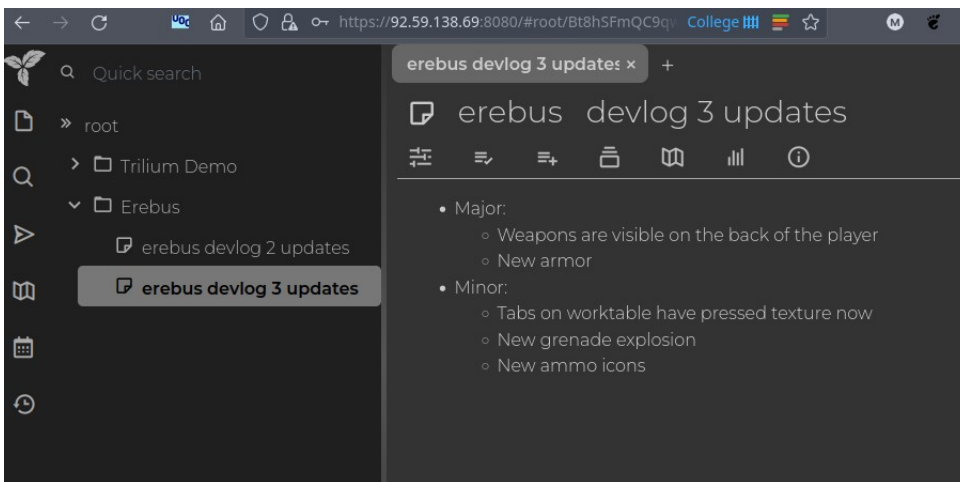


Figura 50: Nota importada a Trilium

Ara fem el mateix per les notes de l'altre videojoc de Weküfu Studios. El resultat final és aquest:

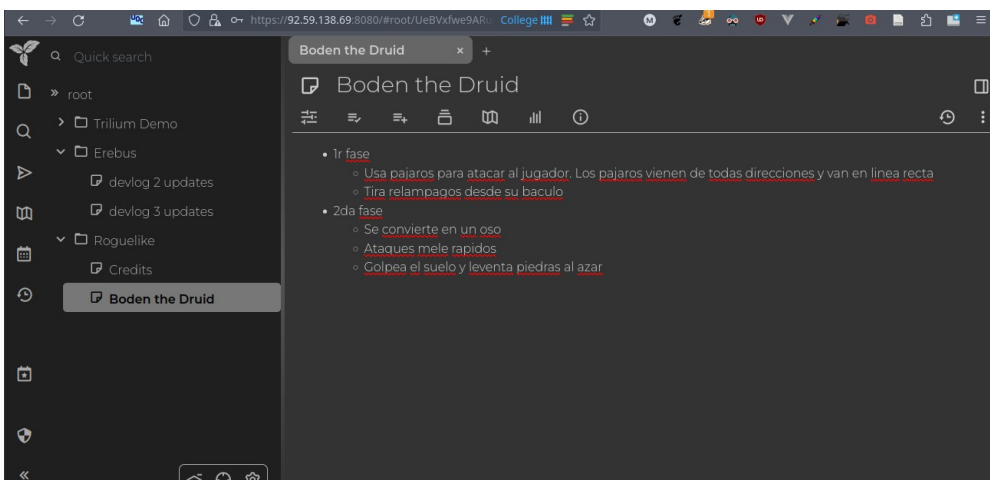


Figura 51: Trilium amb totes les notes importades

## 6 Panell de serveis

El panell de serveis ens permetrà mostrar tots els serveis del nostre servidor i accedir-hi fàcilment a través d'un enllaç. Hi ha moltes opcions quan a panell de serveis, anem a veure'n unes quantes.

### Opcions disponibles



Dashy és molt personalitzable. Com és veu a la figura 52, pots organitzar els diferents serveis en contenidors amb títol. Però hi ha molts altres format: els pots organitzar en columnes, en pestanyes, entre moltes altres opcions. També ve per defecte amb uns quants temes i molts icones.

Una característica bastant útil és que permet afegir un petit indicador a la punta de cada servei que indica si està funcionant bé.

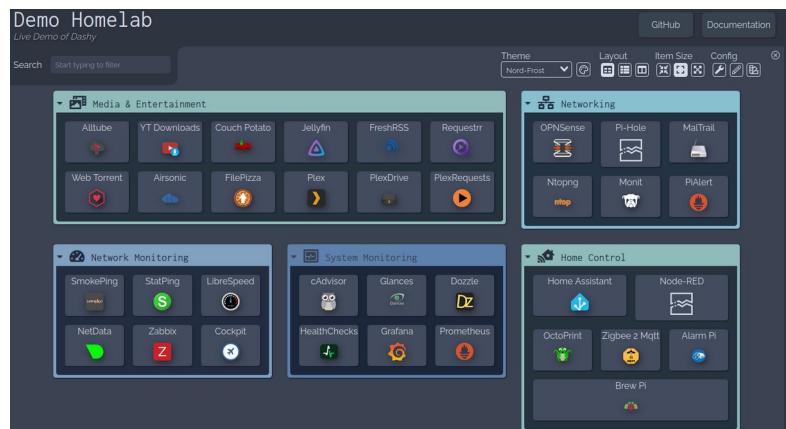


Figura 52: Exemple Panell Dashy

També disposa de gadgets amb funcionalitats molt variades. Per exemple:

- Mostrar l'estat del sistema: RAM, CPU, espai del disc...
- Mostrar el temps atmosfèric.
- Mostrar la IP pública.

A més a més de modificar la configuració a través dels fitxers yaml, també és pot fer des del panell directament. Un cop canviada la configuració, el panell s'actualitza automàticament, sense necessitat de reiniciar el servei.



Homepage també és molt personalitzable. Permet configurar la interfície de manera similar a Dashy: en contenidors amb serveis. Pots decidir quantes columnes té cada contenidor, el seu títol, i també pots mostrar els contenidors en pestanyes.

També té gadgets que permeten mostrar més informació sobre certs serveis. Per exemple, mostrar l'espai lliure que ens queda a Nextcloud.

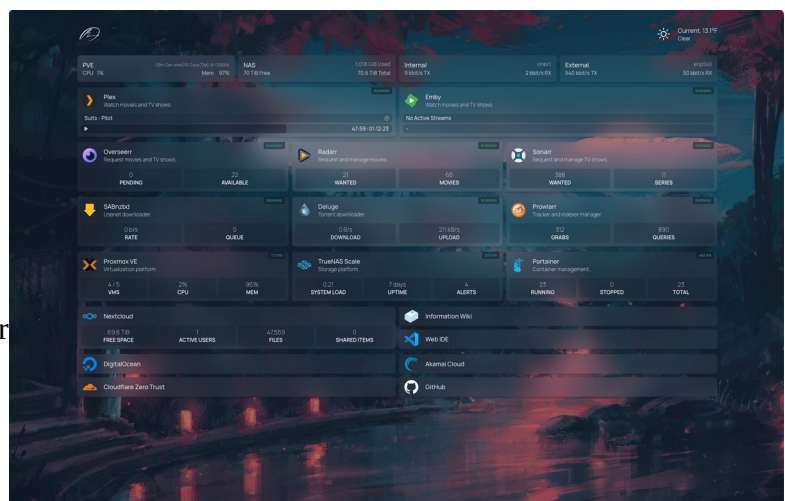


Figura 53: Exemple panell Homepage

Permet mostrar la memòria utilitzada, el tràfic d'internet i el temps atmosfèric, entre altres coses.

Homepage no es pot configurar des del panell mateix, si no que només ho podem fer modificant els fitxers yaml de la configuració.



Similar a Dashy i Homepage ens permet organitzar els serveis en grups i permet personalitzar el tema. Té gadgets, però no tants com Dashy o Homepage.

El que el fa una opció atractiva és la facilitat en que es pot modificar el panell. Tot és pot configurar des del panell mateix, la configuració està en format json, però no es necessari tocar-la, ja que és molt més fàcil moure les coses amb el mode d'edició.

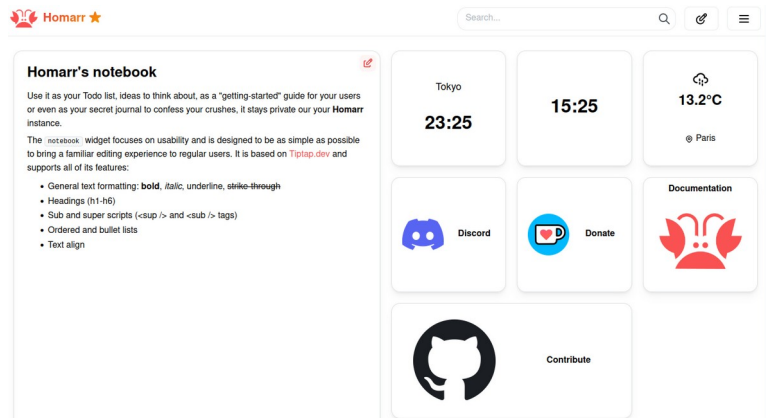


Figura 54: Exemple panell Homarr

## Instal·lació i configuració

Qualsevol de les tres opcions que hem vist són vàlides. La que instal·larem serà Homepage simplement perquè m'agrada l'extensió que mostra els usuaris i espai lliure de Nextcloud de la qual disposa.

Instal·larem Homepage utilitzant el repositori de GitHub. Clonem el repositori:

```
[root@switchroot ~]# git clone https://github.com/gethomepage/homepage.git
Cloning into 'homepage'...
remote: Enumerating objects: 54453, done.
remote: Counting objects: 100% (54452/54452), done.
remote: Compressing objects: 100% (9920/9920), done.
remote: Total 54453 (delta 32865), reused 54025 (delta 32639), pack-reused 1
Receiving objects: 100% (54453/54453), 20.20 MiB | 11.90 MiB/s, done.
Resolving deltas: 100% (32865/32865), done.
```

Per compilar el codi necessitem un gestor de paquets de JavaScript, farem servir npm. L'instal·lem a través de dnf:

```
[root@switchroot ~]# dnf install npm
...
Complete!
```

Un cop tenim npm, entre en el directori de Homepage i compilem el programa:

```
[root@switchroot ~]# cd homepage
[root@switchroot homepage]# npm install
...
[root@switchroot homepage]# npm run build
...
info - Compiled successfully
...
```

Podem inicial Homepage amb la comanda `npm start` estant al directori homepage:

```
[root@switchroot homepage]# npm start

> startpage@0.1.0 start
> next start

Error: listen EADDRINUSE: address already in use 0.0.0.0:3000
  at Server.setupListenHandle [as _listen2] (node:net:1817:16)
  at listenInCluster (node:net:1865:12)
  at doListen (node:net:2014:7)
  at process.processTicksAndRejections (node:internal/process/task_queues:83:21) {
  code: 'EADDRINUSE',
  errno: -98,
  syscall: 'listen',
  address: '0.0.0.0',
  port: 3000
}
```

Ens trobem amb un error. El problema és que per defecte, Homepage utilitza el port 3000, el mateix port que està fent servir el servidor web de Forgejo. No hi poden haver dos serveis amb el mateix port, n'hem de canviar un dels dos. Canviarem el port de Forgejo a 3001, ja que sembla més simple que canviar el port de Homepage. Canviem el valor de la propietat HTTP\_PORT de la configuració de Forgejo de 3000 a 3001:

```
[root@switchroot homepage]# vi /etc/forgejo/app.ini
...
HTTP_PORT = 3001
...
```

També actualitzem els ports del proxy apache perquè redirigeixin les connexions al nou port:

```
ProxyPass "/" "http://127.0.0.1:3001/"
ProxyPassReverse "/" "http://127.0.0.1:3001/"
```

Reiniciem Forgejo i Apache perquè els canvis tinguin efecte:

```
[root@switchroot homepage]# systemctl restart forgejo
[root@switchroot homepage]# systemctl restart httpd
```

Si tornem a provar d'executar Homepage, ara ja no ens surt cap error:

```
[root@switchroot homepage]# npm start
> startpage@0.1.0 start
> next start

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
```

En aquest moment, Homepage ja és funcional. Però, abans de configurar-lo, podem crear un script per iniciar Homepage automàticament quan el servidor s'inicia, com hem fet amb tots els altres serveis. L'script en qüestió és troba a l'annex.

Si anem al port 3000 del servidor, veurem el panell per defecte:

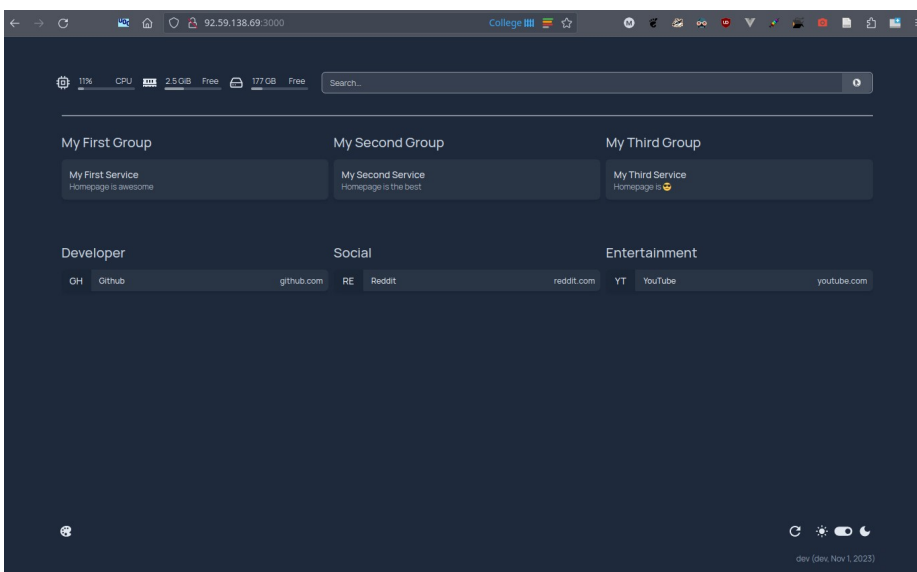


Figura 55: Panell per defecte de Homepage

El fitxer de configuració es troben a `/usr/local/bin/homepage/config/`. A continuació afegirem els serveis modificant el fitxer `services.yml`.

## Serveis

### Forgejo

Canviem el nom del grup a 'Fitxers' i el nom del primer servei a 'Forgejo'. Canviem href a <https://92.59.138.69:3001>, l'adreça del servidor web Forgejo. També escrivim una descripció apropiada. Homepage treu les icones del [repositori dashboard-icons](#), aquest repositori té una icona per Forgejo amb el nom 'forgejo.png', però podem escriure només 'forgejo' i Homepage ja buscarà el fitxer adequat.

```
- Fitxers:
- Forgejo:
  href: https://92.59.138.69:3001
  description: Repositoris Forgejo
  icon: forgejo
```

Reiniciem Homepage i ja veiem el nou enllaç al panell de serveis:

```
[root@switchroot ~]# systemctl restart homepage
```

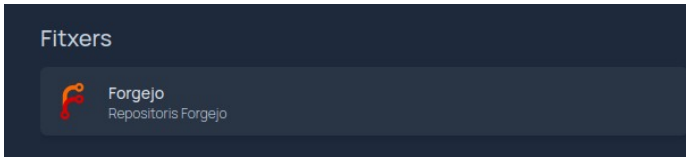


Figura 56: Enllaç Forgejo a Homepage

## Nextcloud

Fiquem el servei Nextcloud dintre el grup Fitxers, juntament amb Forgejo. Configurem Nextcloud de la mateixa manera que hem fet amb Forgejo, amb l'addició d'un widget, el qual ens permetrà mostrar informació del servidor Nextcloud. Perquè el widget funcioni, hem d'indicar el seu tipus, 'nextcloud', la seva url '<https://92.59.138.69/nextcloud>' i un usuari i contrasenya:

```
- Fitxers:
- Forgejo:
  ...
- Nextcloud:
  href: https://92.59.138.69/nextcloud
  icon: nextcloud
  widget:
    type: nextcloud
    url: https://92.59.138.69/nextcloud
    username: mateu
    password: contrassenya
```

El widget ens mostra l'espai lliure, el nombre d'usuaris actius, el nombre de fitxers i el nombre de fitxers compartits:

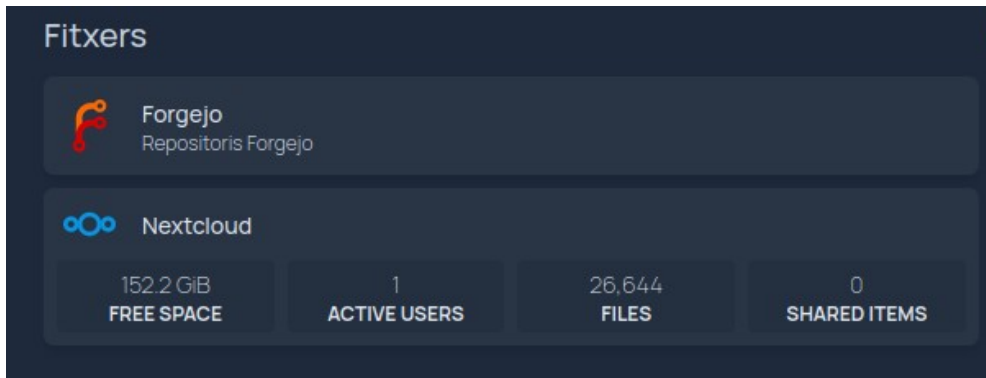


Figura 57: Nextcloud a Homepage

## Altres

He afegit enllaços per a Trilium i per a la pàgina web. Per la pàgina web, utilitzo la icona de la web que està a [http://92.59.138.69/wekufu\\_logo.png](http://92.59.138.69/wekufu_logo.png). No he afegit el servidor xmpp perquè aquest no té interfície web.

També he afegit la propietat ping per cada un d'ells que ens indica si estan actius o no i el seu ping. A la següent captura podem veure tots els serveis:

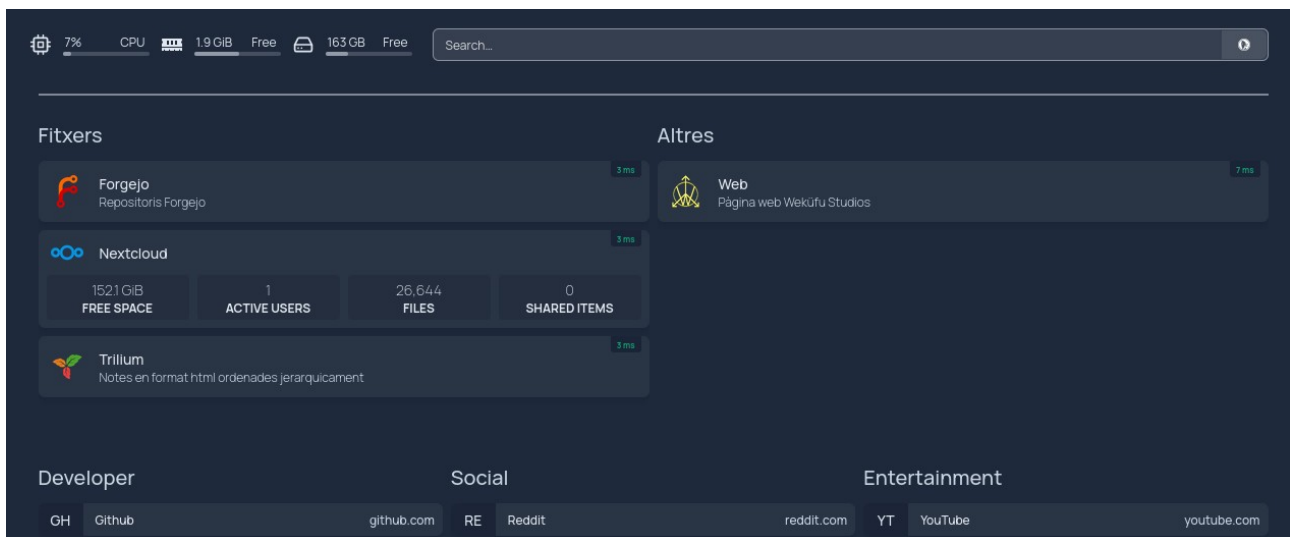


Figura 58: Panell Homepage amb tots els serveis del nostre servidor

## Preferits

A part dels serveis que ofereix el nostre servidor, també podem afegir altres enllaços de pàgines externes. Ho podem fer a través del fitxer `bookmarks.yaml`. La sintaxi del fitxer és la mateixa que la del fitxer de serveis: definim grups i fem els enllaços dintre d'aquests grups.

Per exemple, per crear un grup amb nom 'Traduir' i amb enllaços a Word Reference i DeepL:



```

- Traduir:
  - Word Reference:
    - abbr: WR
    href: https://www.wordreference.com/
  - Deepl:
    - abbr: D
    href: https://www.deepl.com/translator
    description: Traductor

```

Com que no dispo de les icones, he fet servir una abreviació en el seu lloc. També he afegit enllaços al Twitter, Mastodon, YouTube, i itchio de Weküfu Studios. A la següent captura es poden veure tots els preferits:

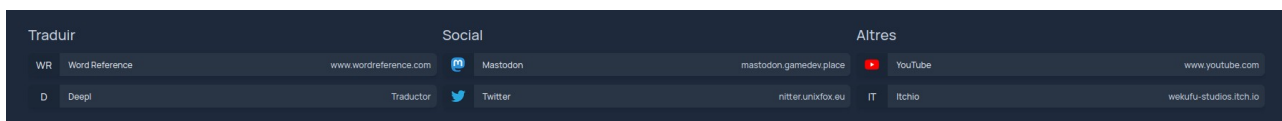


Figura 59: Preferits Homepage

## Altres

A config.yaml podem configurar moltes altres coses com l'estil de la capçalera, el fons, el tema...

A continuació canviarem el fons. Farem servir la url <https://w.wallhaven.cc/full/wy/wallhaven-wypo96.jpg>. Al fitxer de configuració fem un apartat pel fons i li indiquem la url amb la propietat 'image'. Les formes no es veuen gaire bé amb la imatge a sota, per solucionar-ho podem reduir la transparència de la imatge amb la propietat 'opacity'

```

background:
  image: https://w.wallhaven.cc/full/wy/wallhaven-wypo96.jpg
  opacity: 20

```

Aquest és el resultat final:



Figura 60: Panell de serveis final

## 7 Conclusió

La gran majoria dels objectius proposats han estat assolits. Tenim un servidor funcional el qual Weküfu Studios pot utilitzar per cobrir les seves necessitats.

Amb Apache, la seva pàgina web és accessible per tothom qui hi vulgui accedir.

Amb Nextcloud, Weküfu Studios té un lloc on guardar tots els fitxers importants. A més, amb l'aplicació d'escriptori de Nextcloud, gestionar els fitxers és molt fàcil, ni tan sols cal accedir a la pàgina web, només ficar els fitxers a la carpeta correcta i es pujaran sols.

Gràcies a Forgejo, l'organització pot guardar i gestionar els seus repositoris. I no només això, també pot organitzar les tasques amb 'Projects', una funcionalitat de Forgejo que permet crear i organitzar tasques, ja sigui amb el mètode Kanban, o d'una altra manera.

Després d'instal·lar Prosody, els membres de l'organització es poden comunicar sense dependre de software propietari i les preocupacions de privadesa que aquest genera, però només per a Android. Tot i que els clients xmpp per a Android que he provat funcionen a la perfecció, m'he trobat que els clients d'escriptori deixen molt que desitjar. La majoria no m'han deixat fer servir el certificat signat per mi mateix, i l'únic que m'ha deixat, Pidgin, és un projecte que ja no rep actualitzacions i on les trucades de veu no funcionen. Per això, els objectius d'aquest servei no han estat del tot assolits, ja que lo ideal seria poder fer-lo servir des d'un ordinador d'escriptori.

Trilium permet a l'organització prendre notes fàcilment des del navegador i tenir-les ben organitzades jeràrquicament.

Homepage ens proporciona una pàgina des d'on accedir a tots els altres serveis. Gràcies al panell de serveis, els usuaris no han de memoritzar l'adreça de cada servei, només la del panell.

Durant la realització del treball he après un munt de coses. Alguns serveis han estat molt fàcils d'instal·lar i configurar, com Apache, instal·les el paquet i ja està, ja funciona. En canvi, hi ha hagut altres serveis els quals ha estat una odissea fer-los funcionar, com les accions de Forgejo i Prosody.

També he descobert varies funcionalitats de LibreOffice Writer que desconeixia, com les capçaleres i els subtítols de les imatges.

Finalment, he après a crear scripts per fer les còpies de seguretat dels diferents serveis. Per falta d'espai, aquests scripts han quedat a l'annex. Addicionalment, com que surten noves versions de Forgejo constantment, he creat un script per actualitzar-lo el qual també està inclòs en l'annex.

El fet que el servidor que he fet servir té un processador arm64, en comptes de amd64, m'ha causat alguna dificultat en l'instal·lació de Trilium, ja que no té binari per a arm64, però res greu, ja que al final he vist que té un contenidor de docker compatible amb arquitectura arm64.

El principal problema ha estat el certificat del servidor. Degut a que he signat el certificat jo mateix, molts programes m'han donat problemes. Sobretot, els client de xmpp per escriptori. Penso que el fet que no tinc un domini també hi ha contribuït, ja que tot afegir el certificat a la llista de certificats de confiança de l'ordinador, em diu que el certificat està assignat a un altre domini.

De tota manera, obtenir un domini no requereix cap coneixement informàtic, simplement pagar. I abans d'obtenir un domini, necessitem una adreça IP estàtica (l'adreça que tinc a casa meua és dinàmica) i probablement pagar, si es que és possible tenir una adreça estàtica en un habitatge.

Així doncs, l'obtenció de una IP estàtica i un domini quedaran pendents d'implementar en el futur, quan Weküfu Studios s'ho pugui permetre. Ja que com he dit, és més una qüestió econòmica, que no pas de coneixements informàtics.

## Glossari

**Git:** sistema de control de versions creat per Linus Torvalds. És el més utilitzat en l'actualitat.

**GitHub:** servidor git propietat de Microsoft. És el servidor git més popular, hi ha projectes molt importants com Linux, Godot, Rust, vsodium, node, entre molts altres.

**Push:** acció d'enviar el codi al servidor git.

**CI/CD:** Continuous Integration / Continuous delivery. Ens permet executar certes instruccions quan fem un push al repositori per exemple.

**QUIC:** HTTP a través del protocol de transport UDP en comptes de TCP.

**Md:** format de fitxer que permet fer servir alguns dels 'tags' del format html però d'una manera més simple.

**Yaml:** 'YAML Ain't Markup Language', un format de fitxer que normalment és fa servir pels fitxers de configuració.

# Bibliografia

## Web

- [Pàgina principal Algernon](#)
- [Pàgina principal Apache](#)
- [Pàgina principal Nginx](#)
- [Comparació Apache i Nginx](#)
- [Instal·lació i configuració Apache Fedora](#)
- [Descarregar només una branca git](#)
- [Ignora certificat al fer git clone](#)
- [Permisos fitxers web](#)

## Fitxers

- [Pàgina principal Nextcloud](#)
- [Pàgina principal OwnCloud](#)
- [Pàgina principal Seafile](#)
- [Documentació Fedora per instal·lar Nextcloud](#)
- [Tutorial per instal·lar Nextcloud a Fedora](#)
- [Port MariaDB](#)
- [Fòrum Nextcloud que explica com solucionar l'error 'Data directory accessible from the internet'](#)
- [Php error limit memòria](#)
- [How to install Collabora Online in Nextcloud Hub](#)

## Git

- [Pàgina principal Gitea](#)
- [Pàgina principal Gogs](#)
- [Pàgina principal Forgejo](#)
- [Pàgina principal OneDev](#)
- [Documentació instal·lació Forgejo](#)
- [Documentació gitea https](#)
- [Documentació Apache proxy](#)

- [Documentació Apache Virtual Host](#)
- [Donar permis a un usuari perquè accedeixi a docker](#)
- [git push especificant url i branca del repositori](#)

## Comunicació

- [Pàgina principal xmpp](#)
- [Pàgina principal Matrix](#)
- [Matrix vs XMPP](#)
- [Documentació Prosody](#)

## Notes

- [Pàgina principal Joplin](#)
- [Vídeo que ensenya com sincronitzar notes amb Nextcloud](#)
- [Extensió notes Nextcloud](#)
- [Pàgina GitHub Trilium](#)
- [Documentació Trilium](#)
- [Tutorial docker](#)
- [Iniciar contenidor docker a través de systemd](#)

## Panell de serveis

- [Pàgina GitHub Dashy](#)
- [Pàgina principal Homepage](#)
- [Pàgina principal Homarr](#)
- [Instruccions d'instal·lació de Homepage](#)
- [Especificar directori del projecte a npm](#)
- [Crear script systemd](#)