Picto

Software: Pixelfed
Web oficial: https://pixelfed.org/
Documentación oficial: https://docs.pixelfed.org/
Foro/Matrix de soporte: pixelfed:matrix.org

Ubicaciones importantes: `/usr/share/webapps/pixelfed`
Logs relevantes:

Comandos de gestión

Crear un usuario nuevo

```
cd /usr/share/webapps/pixelfed
sudo php artisan user:create
```

Dar permisos de administrador a una cuenta

```
sudo php artisan user:admin username_here
```

Actualizar

Siempre hay que leer la instrucciones en la documentación oficial y leer el changelog de la nueva versión!!

```
sudo su pixelfed
cd /usr/share/webapps/pixelfed
git pull origin dev
composer install
php artisan config:cache
php artisan route:cache
php artisan migrate --force
```

Diario de instalación

Tras la creación de un usuario propio diferente de root, lo primero que observo es que al usar sudo aparece el siguiente error:

```
sudo: unable to resolve host Pixelfed-Test: Name or service not known
```

Se soluciona editando el siguiente fichero y añadiendo el host:

```
sudo vim /etc/hosts
127.0.1.1       Pixelfed-Test
127.0.1.1       Pixelfed
```

El último lo añado para evitar problemas si cambiamos a Pixelfed el nombre del servidor.

Ahora actualizo todos los paquetes:

```
sudo apt update
sudo apt upgrade
```

Dependencias

Siguiendo la documentación de pixelfed, instalo las dependencias que indican:

```
sudo apt install nginx python3-certbot-nginx mariadb-server php-fpm git jpegoptim optipng pngquant ffmpeg
php-gd
```

Instalamos el Composer:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8cba5b96ac8b64115ac402c8cd292b8a03482574915
d1a8') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo
PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"

sudo mv composer.phar /usr/local/bin/composer
```

Revisamos con el siguiente comando las extensiones de php que tenemos instaladas e instalamos las que nos faltan respecto a la documentación:

```
php-fpm7.4 -m

sudo apt install php-bcmath php-curl php-intl php-mbstring php-mysql php-redis php-xml php-zip redis
```

No he tenido que activara las extensiones de las bases de datos, como dice la documentaci'on de pixelfed.

Finalmente, cambio algunos parámetros del php.ini para que no de problemas al subir ficheros:

```
sudo vim /etc/php/7.4/fpm/php.ini


post_max_size = 10M
upload_max_filesize = 10M
max_execution_time = 600
```

Base de Datos

Procedo a crear la base de datos (obviamente con otra contraseña):

```
sudo mysql -u root -p
```

```
create database pixelfed;
grant all privileges on pixelfed.* to 'pixelfed'@'localhost' identified by 'XXXXXXXXX';
flush privileges;
exit;
```

Configuración del PHP-FPM pool

No creo un usuario dedicado. Usaremos www-data porque el usuario dedicado me ha dado problemas en el pasado.

Para configurar el php-fpm pool:

```
cd /etc/php/7.4/fpm/pool.d/
sudo cp www.conf pixelfed.conf
sudo vim pixelfed.conf
```

Sólo cambio el nombre del pool:

`[www-data]`
Corrijo esta ruta:
`listen = /run/php/php-fpm.sock`
Y decomento la siguiente linea:
`listen.mode = 0660`

Descargar Código

Ya podemos descargar el código fuente des de Git:

```
sudo mkdir /usr/share/webapps
cd /usr/share/webapps/
sudo git clone -b dev https://github.com/pixelfed/pixelfed.git pixelfed
```

Definir los permisos correctos

```
cd pixelfed
sudo chown -R www-data:www-data . # change user/group to http user and http group
sudo find . -type d -exec chmod 755 {} \; # set all directories to rwx by user/group
sudo find . -type f -exec chmod 644 {} \; # set all files to rw by user/group
```

Inicializar las dependencias PHP

`sudo composer install --no-ansi --no-interaction --optimize-autoloader`

Edición del fichero de configuración

Copiamos el fichero de ejemplo y lo rellenamos. En algunos casos he usado de referencia el que teníamos en la instalación previa.

```
sudo cp .env.example .env
sudo vim .env
```

Configurando servicios

Aquí ejecutamos una serie de tareas que sólo se requieren una vez siguiendo la documentación de forma fiel:

Una vez sólamente, se necesita generar una APP KEY secreta. Aunque la he generado, al final la he tenido que definir yo. Hay que tener cuidado porque la longitud de la llave tiene que ser 32 caracteres. De una configuración anterior arrastraba una más larga y me daba problemas.

`php artisan key:generate`

Enlazar la carpeta storage/ a la aplicación:

`php artisan storage:link`

Se tiene que migrar la base de datos:

`php artisan migrate --force`

Si se quiere activar el soporte para los datos de localización.

`php artisan import:cities`

Si activas la federación ActivityPub:

`php artisan instance:actor`

Si activas OAuth:

`php artisan passport:keys`

Las rutas se tienen que "cachear" cada vez que hay cambios en el código fuente o cuando cambiamos rutas:

```
php artisan route:cache
php artisan view:cache
```

Cada vez que editamos el fichero .env, tenemos que ejecutar este comando para que los cambios tengan efecto:

`php artisan config:cache`

Instalación de Horizon

```
php artisan horizon:install
php artisan horizon:publish
```

Creación de un servicio

Ahora vamos a crear un servicio para que se ejecute horizon. Creamos el siguiente fichero:

`/etc/systemd/system/pixelfed.service`

Con el siguiente contenido:

```
[Unit]
Description=Pixelfed task queueing via Laravel Horizon
After=network.target
Requires=mariadb
Requires=php7.4-fpm
Requires=redis
Requires=nginx

[Service]
Type=simple
```

```
ExecStart=/usr/bin/php7.4 /usr/share/webapps/pixelfed/artisan horizon
User=www-data
Restart=on-failure

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

Y lo activamos:

`sudo systemctl enable --now pixelfed`

Programación de tareas periódicas

Podemos programar un cron para que ejecute tareas regulares (como hacer backups):

`sudo -u www-data crontab -e`

Y incorporamos al cron:

`* * * * * /usr/bin/php7.4 /usr/share/webapps/pixelfed/artisan schedule:run >> /dev/null 2>&1`

Proxy

Usamos el proxy nginx. Copiamos el modelo y lo editamos;

```
sudo cp contrib/nginx.conf /etc/nginx/modules-available/pixelfed.conf
sudo vim /etc/nginx/sites-available/pixelfed.conf
```

Con las siguientes líneas modificadas;

```
server_name picto.anartist.org;                 # change this to your fqdn
root /usr/share/webapps/pixelfed/public;                    # path to repo/public

fastcgi_pass unix:/run/php/php7.4-fpm.sock; # make sure this is correct

#    listen 443 ssl http2;
#    listen [::]:443 ssl http2;
#    ssl_certificate /etc/nginx/ssl/server.crt;       # generate your own
#    ssl_certificate_key /etc/nginx/ssl/server.key;   # or use letsencrypt

#    ssl_protocols TLSv1.2;
#    ssl_ciphers EECDH+AESGCM:EECDH+CHACHA20:EECDH+AES;
```

```
#    ssl_prefer_server_ciphers on;
```

Obtenemos un certificado de Let's Encrypt:

`sudo certbot --nginx -d picto.anartist.org`

Tras correr el certbot he tenido que revisar el fichero de nginx y comentar el server http, porque me lo habia duplicado.

Finalmente activamos la configuración nginx y reiniciamos el servicio

```
sudo ln -s /etc/nginx/sites-available/pixelfed.conf /etc/nginx/sites-enabled/
sudo systemctl restart nginx.service
```

Revision #2
Created 14 June 2024 05:33:57 by marcelcosta
Updated 24 March 2026 19:36:26 by marcelcosta