Welcome!

While we are waiting to get started, you can:

1. Open a web browser, log in to packetfence & ensure your time is correct.
    1. If your time is not syncing, stop and start NTP
        1. `sudo timedatectl set-ntp False`
        2. `timedatectl status`
        3. `sudo timedatectl set-ntp True`
        4. `timedatectl status`
2. Run updates:
    2. `sudo apt update && sudo apt upgrade -y`
3. If you want to use `vim` instead of `vi`
    1. `sudo apt install vim`
4. If you want to be able to copy & paste during the demo: https://suddenlysixam.club/projects/nginx

NGINX (w/ Etherpad)

Homelab Club at UMD Meeting 2025-04-07

Package installation:

As always, make sure you are up to date:

```
sudo apt update && sudo apt upgrade -y
```

Install Node.js:

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo bash -
```

```
sudo apt install -y nodejs
```

Run as a different user

```
sudo adduser --system --group --shell /bin/bash --home /opt/etherpad etherpad
```

Configure Etherpad Lite

Clone the branch:

```
cd /opt/etherpad

sudo git clone --branch master
https://github.com/ether/etherpad-lite.git .
```

Install dependancies: *sudo ./bin/installDeps.sh*

Note: The dot (.) at the end of the git clone command is important - it will clone the files into the current directory, rather than into a subdirectory.

Configure Etherpad Lite (cont.)

Change ownership of cloned & installed files:

```
sudo chown -vR etherpad:etherpad /opt/etherpad/
```

Create a log directory and change it’s ownership:

```
sudo mkdir /var/log/etherpad
sudo chown -v etherpad:etherpad /var/log/etherpad/
```

Configure Etherpad Lite (cont.)

Edit settings.json: *sudo vi settings.json*

Update the line
"trustProxy": false, to be
"trustProxy": true,

Take note even though we aren't changing it:

```
"dbType": "dirty",
"dbSettings": {
    "filename": "var/dirty.db"
},
```

Create a systemd service

Edit a new file /etc/systemd/system/etherpad.service

sudo vi /etc/systemd/system/etherpad.service

Add the following content:

```
[Unit]
Description=Etherpad-lite, the collaborative editor.
After=syslog.target network.target

[Service]
Type=simple
User=etherpad
Group=etherpad
WorkingDirectory=/opt/etherpad
Environment=NODE_ENV=production
ExecStart=pnpm run prod
Restart=always

StandardOutput=append:/var/log/etherpad/etherpad.log
StandardError=append:/var/log/etherpad/etherpad-error.log

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

Create a systemd service file (cont.)

Reload to get the new configuration

```
sudo systemctl daemon-reload
```

Enable the new service, so that it runs when the Pi starts

```
sudo systemctl enable etherpad
```

Start the service

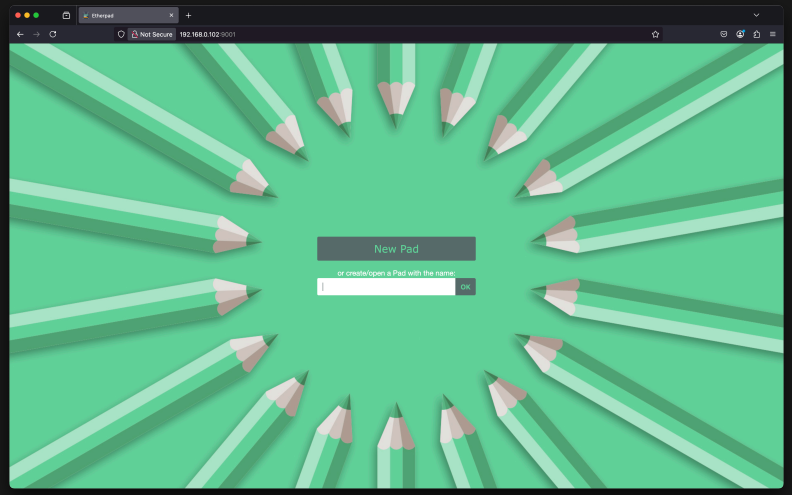
```
sudo systemctl start etherpad
```

Check its status (make sure it is `active (running)`)

```
sudo systemctl status etherpad
```

But does it work?

Navigate to http://localhost:9001 in a web browser. You should now see Etherpad.



Note: Alternatively, from a different host on the same network, http://<ip-address>:9001

Configure NGINX

Install packages:

```
sudo apt install nginx
```

NGINX: Basics

```
server {

    location <prefix> {
        ...
     }

    ...
 }
```

```
location / {
    root /data/www;
}
```

```
location / {
    proxy_pass http://localhost:8080;
}
```

NGINX: Our configuration

sudo vi /etc/nginx/sites-available/etherpad.conf

```
server {
    listen       80;
    listen       [::]:80;
    server_name  ;

    access_log  /var/log/nginx/etherpad.access.log;
    error_log   /var/log/nginx/etherpad.error.log;

    location / {
        proxy_pass         http://127.0.0.1:9001;
        proxy_buffering    off;
        proxy_set_header   Host $host;
        proxy_pass_header  Server;

        # proxy headers
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $remote_addr;
        proxy_set_header    X-Forwarded-Proto $scheme;
        proxy_http_version  1.1;

        # websocket proxying
        proxy_set_header  Upgrade $http_upgrade;
        proxy_set_header  Connection "upgrade";
    }
}
```

Note: Port 80 instead of port 443. We aren’t doing HTTPS just yet.

NGINX: Our configuration (cont.)

Enable the configuration by creating a link:

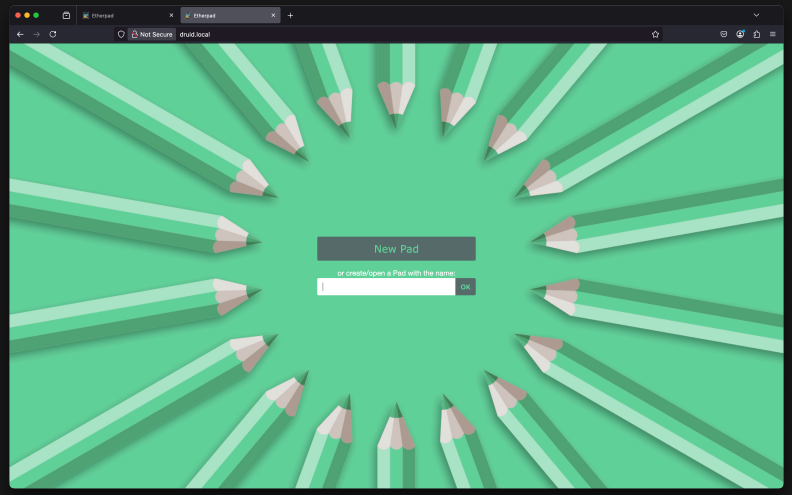
```
sudo ln -s /etc/nginx/sites-available/etherpad.conf
/etc/nginx/sites-enabled/
```

Check the configuration file syntax, and reload the NGINX service

```
sudo nginx -t
sudo systemctl reload nginx
```

But does it work? (pt. 2)

Navigate to the domain that you configured in your web browser (e.g. druid.local). You should now again see Etherpad.



Note: If it does not load in the web browser, did you set the domain to <hostname>.local?

Note: We can also test the collaboration aspect, by opening up the site on two devices, and editing a ‘pad’ of the same name. We can edit in both places, and see each others edits.

Self Signed Certificate

Generate a self signed cert:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Maryland
Locality Name (eg, city) []:College Park
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Homelab Club at UMD
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:druid.local
Email Address []:admin@druid.local
```

Create a strong Diffie-Hellman (DH) group

sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048

Self Signed Cert: Configure NGINX

sudo vi /etc/nginx/snippets/self-signed.conf

Add the content:

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

Self Signed Cert: Configure NGINX (cont.)

sudo vi /etc/nginx/snippets/ssl-params.conf

Add the content:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
ssl_ecdh_curve secp384r1;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
# Disable preloading HSTS for now.  You can use the commented out header line that includes
# the "preload" directive if you understand the implications.
#add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;

ssl_dhparam /etc/ssl/certs/dhparam.pem;
```

Self Signed Cert: Configure NGINX (cont.)

Edit the NGINX configuration file that we made previously

sudo vi /etc/nginx/sites-available/etherpad.conf

```
server {
    listen       80;
    listen       [::]:80;
    server_name  ;
    return 302 https://`$server_name$`request_uri;
}

server {
    listen       443 ssl http2;
    listen       [::]:443 ssl http2;
    server_name  ;

    include snippets/self-signed.conf;
    include snippets/ssl-params.conf;

    access_log  /var/log/nginx/etherpad.access.log;
    error_log   /var/log/nginx/etherpad.error.log;

    location / {
        proxy_pass         http://127.0.0.1:9001;
        proxy_buffering    off;
        proxy_set_header   Host $host;
        proxy_pass_header  Server;

        # proxy headers
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $remote_addr;
        proxy_set_header    X-Forwarded-Proto $scheme;
        proxy_http_version  1.1;

        # websocket proxying
        proxy_set_header  Upgrade $http_upgrade;
        proxy_set_header  Connection "upgrade";

        add_header X-Frame-Options ALLOW always;
    }
}
```

Self Signed Cert: Configure NGINX (cont.)

Check the NGINX configuration

sudo nginx -t

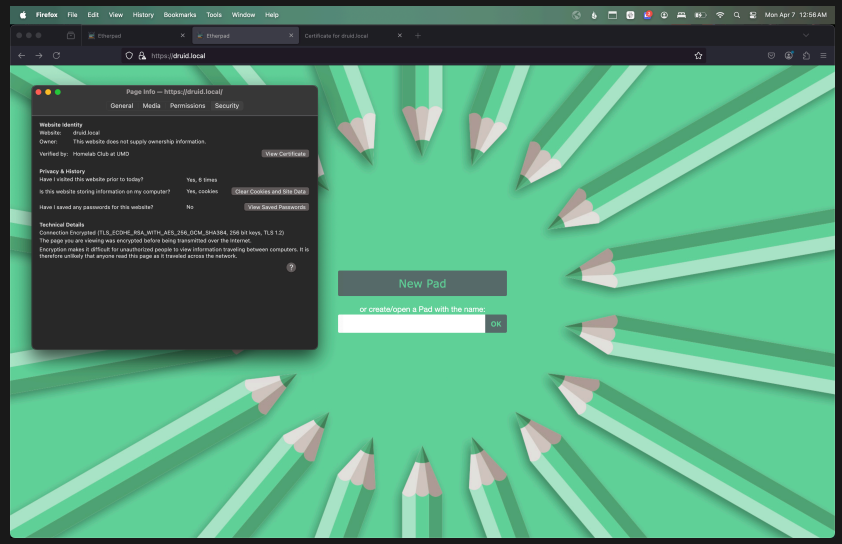
Restart NGINX

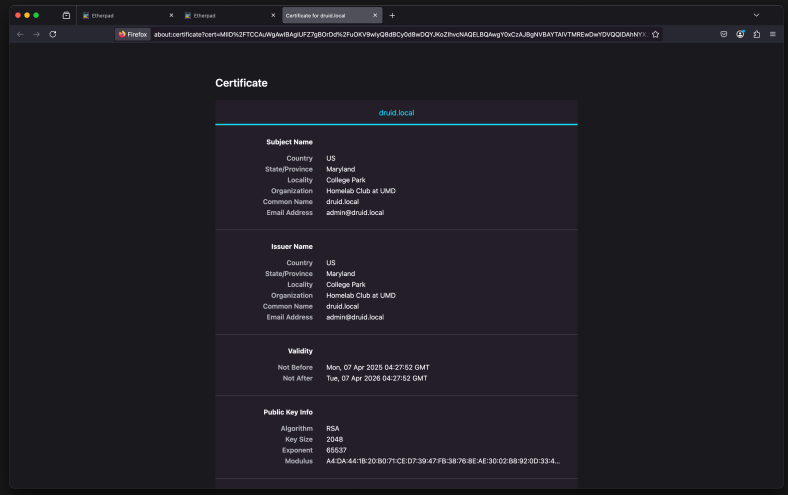
sudo systemctl restart nginx

Note: "ssl_stapling" ignored warning is expected/

But does it work? (pt. 3)

Navigate to the domain that you configured in your web browser (e.g. druid.local). If promted, accept the risk & continue - we know its self signed. You should now again see Etherpad, but this time, with a cert!





Note: Regardless of if you go to http://, https://, or don't specify, it should redirect you to https://.

Recommended next steps

1. Configure a firewall (e.g. UFW)
2. Configure a non-self signed cert (e.g. certbot)

Thank you!

Don’t forget to join the Discord!

https://suddenlysixam.club/discord