## List Of Changes

### Config Change

In /etc/nginx/sites-available/lemmy.conf

server {

listen 80;

server\_name domain.in www.domain.in;

location /api/ {

proxy\_pass http://127.0.0.1:5000/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_http\_version 1.1;

add\_header 'Access-Control-Allow-Origin' "\*" always;

add\_header 'Access-Control-Allow-Headers' "Content-Type, X-CSRFTOKEN";

add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';

}

location / {

root /var/www/html;

index index.html;

try\_files $uri /index.html;

}

}

### Enable HTTPS using Certbot

sudo apt update

sudo apt install certbot python3-certbot-nginx -y

sudo certbot --nginx -d domain.in -d www.domain.in

### Reload

sudo nginx -t

sudo systemctl reload nginx

### Auto-Renewal Verification

sudo systemctl list-timers | grep certbot

Expected Output

Thu 2025-05-29 ... certbot.timer

## Integration Conflicts

Component - Lemmy

Potential Issue - TLS flag misalignment

Mitigation - Keep tls: false if NGINX handles HTTPS

Component - Phoenix

Potential Issue - Wrong scheme detection

Mitigation - Use rewrite\_on: [:x\_forwarded\_proto]

Component - React

Potential Issue - API request failure

Mitigation - Switch all fetch/axios URLs to https://

Component - NGINX

Potential Issue - Multiple conflicting conf files

Mitigation - Use a single merged config

## Reasoning Behind Changes

### 1. Update Existing NGINX Configuration

What: Extend the existing NGINX config used by Lemmy to include routing for your Phoenix backend and React frontend.

Why:

* Avoids conflicts that arise from having multiple server blocks on the same domain and port.
* Ensures a unified reverse proxy strategy.
* Keeps TLS termination centralized and manageable.

nginx

location /api/ {

proxy\_pass http://127.0.0.1:5000/;

...

}

location / {

root /var/www/html;

try\_files $uri /index.html;

}

### 2. Install and Configure Certbot (Let's Encrypt)

What: Set up HTTPS certificates using Certbot for automatic certificate generation and renewal.

Why:

* Enables HTTPS (TLS) encryption at no cost.
* Secures sensitive data like login credentials and user submissions.
* Reduces browser security warnings and improves SEO and trust.

Command:

sudo certbot --nginx -d domain.in

### 3. Ensure All Services Use HTTPS URLs

What: Update all URLs in:

* React frontend API requests
* Any backend callback URLs
* Lemmy configuration (if needed)

Why:

* Mixed content (HTTP inside HTTPS) causes browsers to block requests.
* Lemmy’s tls flag expects uniform HTTPS across services if enabled.
* Ensures authentication tokens and cookies are securely transmitted.

### 4. Update Phoenix Endpoint Config

What: Ensure Phoenix can correctly process requests forwarded over HTTPS by NGINX.

Why:

* Phoenix must interpret client request URLs as HTTPS, even though NGINX terminates the SSL.
* Prevents URL mismatches and CORS errors.

How:  
In endpoint.ex, add:

config :your\_app, YourAppWeb.Endpoint,

url: [scheme: "https", host: "domain.in", port: 443],

force\_ssl: [rewrite\_on: [:x\_forwarded\_proto]]

### 5. Enable Auto-Renewal for SSL Certificates

What: Use systemd timers to auto-renew Let's Encrypt certificates.

Why:

* Certificates expire every 90 days.
* Manual renewal increases operational overhead and risk of downtime.

Verify:

sudo systemctl list-timers | grep certbot

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_