RaspberryPi5

- Jellyfin Setup
- Bookstack Setup
    - DuckDNS Setup

Jellyfin Setup

Bookstack Setup

DuckDNS Setup

Here is a complete, step-by-step guide compiling all the actions we took to get your BookStack instance securely online. This guide assumes you have a running BookStack Docker container (on a local IP like 192.168.0.76:6875) and Nginx installed on the same Raspberry Pi.

1. Set Up DuckDNS First, we need to get a domain name and set up a script to keep your home's public IP address in sync with it.

- Go to https://www.duckdns.org/ and log in.
- Create a new subdomain (e.g., choofpibookstack.duckdns.org).
- Copy your token from the top of the page.
- On your Raspberry Pi, find your duckdns folder (we found it at /duckdns). cd /duckdns
- Create or edit the duck.sh script: nano duck.sh
- Paste the following line into the file, replacing the domains and token values with your own. If you have more than one domain, separate them with a comma (no spaces). echo url="https://www.duckdns.org/update?domains=choofpibookstack,my-jellyfin&token=YOUR-TOKEN-HERE&ip=" | curl -k -o /duckdns/duck.log -K -
- Save and exit (Ctrl+X, Y, Enter).
- Make the script executable (this only needs to be done once): chmod +x duck.sh
- Run the script to update your IP for the first time: ./duck.sh
- Check the log to ensure it worked. It must say OK. cat duck.log

2. Configure Router Port Forwarding Nginx acts as a "receptionist" for all your web traffic. We must open the "main doors" (ports 80 and 443) on your router and forward them to your Nginx server (your Pi).

- Log in to your router's administration page.
- Find the Port Forwarding section.
- Add the following two TCP rules. The "Internal IP" must be your Pi's static IP. | Rule Name | External Port | Internal Port | Internal IP (Your Pi) | Protocol | |---|---|---|---|---| | HTTP (Certbot) | 80 | 80 | 192.168.0.76 | TCP | | HTTPS (Site) | 443 | 443 | 192.168.0.76 | TCP | Verification: After saving these rules, go to an external tool like canyouseeme.org and check that Port 80 is "Open".

3. Configure Nginx Next, we'll tell Nginx (your receptionist) what to do when someone asks for choofpibookstack.duckdns.org.

- Create a new config file in your sites-available directory: sudo nano /etc/nginx/sites-available/bookstack.duckdns.conf
- Paste in this simple HTTP-only configuration. This is just for Certbot to use for validation. (Remember to change the server_name and proxy_pass IP/port). server { listen 80;

server_name choofpibookstack.duckdns.org;
location / { proxy_pass http://192.168.0.76:6875; 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_set_header X-Forwarded-Proto $scheme; } }

- Save and exit (Ctrl+X, Y, Enter).
- Enable this new site by creating a symbolic link (a "shortcut") to it: sudo ln -s /etc/nginx/sites-available/bookstack.duckdns.conf /etc/nginx/sites-enabled/
- Test your Nginx configuration to make sure there are no typos: sudo nginx -t
- If the test is successful, reload Nginx: sudo systemctl reload nginx

4. Obtain SSL Certificate with Certbot Now we'll use Certbot to automatically convert your site to secure HTTPS.

- Run the Certbot command: sudo certbot --nginx
- Certbot will read your Nginx files and find your new domain. Select it from the list.
- When asked, choose the Redirect option. This forces all traffic to use HTTPS.
- Certbot will now validate your domain (by connecting on Port 80, which you opened) and automatically edit your bookstack.duckdns.conf file to add all the SSL settings.

> Troubleshooting Note: If this step failed with a SERVFAIL error, it meant your DuckDNS record hadn't updated worldwide yet. The solution was to wait 10-15 minutes (checking progress on whatsmydns.net) and then run sudo certbot --nginx again.

5. Re-configure BookStack This is the final, critical step. We must tell the BookStack application its new, public HTTPS address. This fixed your 502 Bad Gateway and APP_KEY errors.

- Find your BookStack docker-compose.yml file. We used this command: sudo find / -type f (-name "docker-compose.yml" -o -name "docker-compose.yaml") -exec grep -l "bookstack" {} +
- Navigate to that directory (e.g., /home/pi/docker/bookstack). cd /path/to/your/bookstack-folder
- Generate a new APP_KEY. Your container was crashing because this was missing. docker run -it --rm --entrypoint /bin/bash lscr.io/linuxserver/bookstack:latest appkey
- Copy the output key (e.g., base64:xxxxxxxx...).
- Edit your docker-compose.yml: nano docker-compose.yml
- Find the environment: section for the bookstack service and make two changes:
  - Change APP_URL to your new HTTPS address.
  - Paste your new APP_KEY.

environment:

- PUID=1000
- PGID=1000

- TZ=Australia/Brisbane
- APP_URL=https://choofpibookstack.duckdns.org # <-- CHANGED
- APP_KEY=base64:PASTE_YOUR_NEW_KEY_HERE # <-- ADDED/FIXED
- DB_HOST=bookstack_db
- DB_PORT=3306
- DB_USERNAME=bookstack
- DB_PASSWORD=_7mK3iXh-@AveEi
- DB_DATABASE=bookstackapp

- Save and exit (Ctrl+X, Y, Enter).
- Restart your BookStack containers to apply all changes: docker-compose down docker-compose up -d

6. Final Test That's it! Your site is now fully configured. You can access it securely from anywhere in the world by going to: https://choofpibookstack.duckdns.org