/ prowlarr / installation / reverse-proxy

# Prowlarr Reverse Proxy

Configuring reverse proxy setup for Prowlarr with nginx, Apache, and other web servers

## Reverse Proxy Configuration

Sample config examples for configuring Prowlarr to be accessible through a reverse proxy.

> These examples assumes the default port of `9696` and that you set a baseurl of `prowlarr`. It also assumes your web server i.e nginx and Prowlarr running on the same server accessible at `localhost`. If not, use the host IP address or a FQDN instead.

### NGINX

Add the following configuration to `nginx.conf` located in the root of your Nginx configuration. The code block should be added inside the `server context`. Full example of a typical Nginx configuration

> If you're using a non-standard http/https server port, make sure your Host header also includes it, i.e.: `proxy_set_header Host $host:$server_port` or `proxy_set_header Host $http_host` as well as `proxy_set_header X-Forwarded-Host $host:$server_port` or `proxy_set_header X-Forwarded-Host $http_host`

```
location /prowlarr {
    proxy_pass http://127.0.0.1:9696;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_redirect off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $http_connection;
}
# Allow the API/Indexer External Access via NGINX
location ~ /prowlarr(/[0-9]+)?/api {
    auth_basic off;

    proxy_pass http://127.0.0.1:9696;
}
```

A better way to organize your configuration files for Nginx would be to store the configuration for each site in a separate file.

To achieve this it is required to modify `nginx.conf` and add `include subfolders-enabled/*.conf` in the `server` context. So it will look something like this.

```
server {
  listen 80;
  server_name _;

  # more configuration

  include subfolders-enabled/*.conf
}
```

Adding this line will include all files that end with `.conf` to the Nginx configuration. Make a new directory called `subfolders-enabled` in the same folder as your `nginx.conf` file is located. In that folder create a file with a recognizable name that ends with .conf. Add the configuration from above from the file and restart or reload Nginx. You should be able to visit Prowlarr at `yourdomain.tld/prowlarr`. tld is short for Top Level Domain

### Subdomain

Alternatively you can use a subdomain for prowlarr. In this case you would visit `prowlarr.yourdomain.tld`. For this you would need to configure a `A record` or `CNAME record` in your DNS.

> Many free DNS providers do not support this

By default Nginx includes the `sites-enabled` folder. You can check this in `nginx.conf`, if not you can add it using the include directive . And really important, it has to be inside the `http context`. Now create a config file inside the sites-enabled folder and enter the following configuration.

> For this configuration it is recommended to set baseurl to '' (empty). This configuration assumes you are using the default `9696` and Prowlarr is accessible on the localhost (127.0.0.1). For this configuration the subdomain `prowlarr` is chosen (line 5).

⚠ If you're using a non-standard http/https server port, make sure your Host header also includes it, i.e.: `proxy_set_header Host $host:$server_port`

```
server {
  listen      80;
  listen [::]:80;
  server_name prowlarr.*;
  location / {
    proxy_set_header   Host $host;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header   X-Forwarded-Host $host;
    proxy_set_header   X-Forwarded-Proto $scheme;
    proxy_set_header   Upgrade $http_upgrade;
    proxy_set_header   Connection $http_connection;
    proxy_redirect     off;
    proxy_http_version 1.1;

    proxy_pass http://127.0.0.1:9696;
  }
}
```

Now restart Nginx and Prowlarr should be available at your selected subdomain.

## Apache

This should be added within an existing VirtualHost site. If you wish to use the root of a domain or subdomain, remove `prowlarr` from the `Location` block and simply use `/` as the location.

Note: Do not remove the baseurl from ProxyPass and ProxyPassReverse if you want to use `/` as the location.

```
<Location /prowlarr>
  ProxyPreserveHost on
    ProxyPass http://127.0.0.1:9696/prowlarr connectiontimeout=5 timeout=300
    ProxyPassReverse http://127.0.0.1:9696/prowlarr
</Location>
```

`ProxyPreserveHost on` prevents apache2 from redirecting to localhost when using a reverse proxy.

Or for making an entire VirtualHost for Prowlarr:

```
ProxyPass / http://127.0.0.1:9696/prowlarr/
ProxyPassReverse / http://127.0.0.1:9696/prowlarr/
```

If you implement any additional authentication through Apache, you should exclude the following paths:

▶ `/prowlarr/api/`

## Using SSL on the Apache reverse proxy

If the reverse proxy does SSL termination (i.e. the URL to access the reverse proxy is using the `https://` protocol), then you need to tell Prowlarr that it should use `https://` for its API responses by setting the `X-Forwarded-Proto` correctly. The common way is to add the following lines under the `ProxyPassReverse` configuration:

```
RequestHeader set "X-Forwarded-Proto" expr=%{REQUEST_SCHEME}
RequestHeader set "X-Forwarded-SSL" expr=%{HTTPS}
```

Note that this configuration requires enabling the `mod_header` Apache module, which is often not enabled by default.

Powered by Wiki.js