Creating a WebDAV repository server with NGINX

Ergo supports connecting to any standard WebDAV server as a remote repository. This article provides an overview of how to set up NGINX on Ubuntu linux to act as a WebDAV repository for Ergo.

Installing NGINX

Installing NGINX on Ubuntu should be as simple as installing the package via apt-get. You will also want the nginx-extras package:

```
sudo apt-get update
sudo apt-get install nginx nginx-extras
```

at this point, you should be able to start or stop nginx with the comands

```
sudo service nginx stop
sudo service nginx start
```

Optionally set up SSL/TLS

There are various ways to acquire a certificate for secure https access. The easiest and cheapest is to use LetsEcrypt, using the instructions found at https://letsencrypt.readthedocs.org/en/latest/

Configuring WebDAV

Next you will want to create a directory for your webdav root. In this example, we will use /var/dav.

Create the directory, and change the ownership/permissions to be something that your webserver can read/write. In ubuntu this will likely work:

```
sudo mkdir /var/dav
sudo chown www-data /var/dav
```

Next you will want to modify your NGINX configuration files. They will be in /etc/nginx. If there is a file under /etc/nginx/sites-enabled, this would be an appropriate place for the modifications. The full explanation of NGINX configuration is beyond the scope of this document, but the following is a sample, which uses port 443 for https access:

```
#this section redirects from 80 to 443. If you are not using https, use the directives from the 443 section
# below here instead.
server {
  listen 80;
  # always use SSL
  location / {
    if ($request_method = POST) {
      # use temporary to allow for POST to go through
      # 301 will only work for GET/HEAD/OPTIONS
      return 307 https://$host$request_uri;
    }
    return 301 https://$host$request_uri;
  }
}
server {
  listen 443;
  client_max_body_size 0;
  proxy_read_timeout 300;  # answer from server, 5 min
  proxy_send_timeout 300;  # chunks to server, 5 min
  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;
  port_in_redirect  off;
  ssl on;
```

```
  ssl_session_timeout 5m;
  ssl_certificate /etc/ssl/nginx.crt;
  ssl_certificate_key /etc/ssl/nginx.key;
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_prefer_server_ciphers on;
  ssl_ciphers "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384
EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS !RC4";
  root /usr/share/nginx/www;
  index index.html index.htm;
  location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to index.html
    try_files $uri $uri/ /index.html;
  }

  #here you can specify various directories that respond as DAV.
  location /ergo-repo/ {
    root      /var/dav;
    client_body_temp_path /var/dav/temp;
    dav_methods     PUT DELETE MKCOL COPY MOVE;
    dav_ext_methods   PROPFIND OPTIONS;
    create_full_put_path  on;
    dav_access    user:rw group:rw all:rw;
    autoindex     on;
    #below you can specify the access restrictions. In this case, only people on the 141.142 network
        #can write/delete/etc. Everyone else can view.
    limit_except GET PROPFIND OPTIONS{
      allow 141.142.0.0/16;
      deny  all;
    }
    allow all;
  }
  #this is an example of a password restricted repository
  location /password-repo/ {
    root      /var/dav;
    client_body_temp_path /var/dav/temp;
    dav_methods     PUT DELETE MKCOL COPY MOVE;
    dav_ext_methods   PROPFIND OPTIONS;
    create_full_put_path  on;
    dav_access    user:rw group:rw all:rw;
    autoindex     on;
        auth_basic "restricted";
        auth_basic_user_file /etc/nginx/htpasswd;
  }
}
```

Once you modify the configuration file, you must restart your NGINX server before the changes will take effect.

Configuring Security

NGINX has various methods for managing security. WebDAV through NGINX uses the server's standard security mechanisms. The examples above provide examples of how to configure access restrictions for different situations. In general:

- For read-only access, you will want to limit everything except GET PROPFIND OPTIONS. You can do this based on password authentication, network locations, etc.
- To use a username/password to restrict access, follow the instructions from https://www.digitalocean.com/community/tutorials/how-to-set-up-http-authentication-with-nginx-on-ubuntu-12-10