



Serving a Mercurial Repository with Nginx

Featuring HTTPS and the Mac OS launch daemon

I've recently come to like Eclipse and its Mercurial plug-in for source code maintenance. While the plug-in works fabulously, it currently does have some missing features: at least under my version of Mac OS, the repository must lie in the Eclipse default folder tree, or the commands in the Team context menu simply won't be available.

But I wanted to share the repository locally – an SMB share was out of the question for said restriction.

So then I looked into serving the repository with the Nginx web-server, since I have that installed. This article describes one possible approach that

Configuring NGinx

Firstly, you need an up and running Nginx server. Since my website uses HTTPS for everything, the relevant section of my `nginx.conf` file looks like this:

Portion of `nginx.conf`

```
# HTTPS server
server {
    listen      443;
    server_name ...
    ssl         on;

    location / {
        ... your regular website config ...
    }

    location /my_hg_repo {
        auth_basic          "Restricted";
        auth_basic_user_file /opt/nginx/conf/passwd.txt;
        proxy_pass          http://127.0.0.1:8000;
        proxy_set_header    Host $host;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-HTTPS 0;
    }
}
```

IMPORTANT: Issue `sudo /opt/nginx/sbin/nginx -s reload` to have NGinx update itself.

What the added location `/my_hg_repo` does is that Nginx will proxy incoming HTTPS requests for your repository to the mercurial program running as a server (see next section). Nginx will also ask for password.



worked for me.



Creating a password file for NGinx

To protect your repository, we set up a password file. Go to the website <http://www.htaccessstools.com/htpasswd-generator/> and download the password file. Rename it to passwd.txt, copy it to /opt/nginx/conf/ (adapt to your setup) and change to required permissions.

Modifying your HG config file

In your home folder, there should be a file called .hgrc (remember, I am assuming Mac OS). Edit this file as follows.

Portion of .hgrc

```
[ui]
username = ... your username ...

[paths]
my_hg_repo = /Users/you/path_to_repo

[web]
push_ssl = false
allow_push = *
style = gitweb
```

Starting the mercurial web server as a service

You can now start the mercurial web server by running the following command and you are basically done.

```
hg serve --webdir-conf ...path/to/.hgrc
```

If you are on Mac OS, you can start mercurial as a server automatically after reboot. For this to work, you need to create the following file `/Users/you/Library/LaunchAgents/mercurial.plist`.

mercurial.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Label</key>
    <string>mercurial.plist</string>
```

continued on next page

```
<key>RunAtLoad</key>
<true/>
<key>ProgramArguments</key>
<array>
  <string>/opt/local/bin/hg</string>
  <string>serve</string>
  <string>--web-conf</string>
  <string>/Users/michael/.hgrc</string>
</array>
</dict>
</plist>
```

This script is picked up whenever you login under your user name, if you install it first with the following command line:

```
launchctl load /Users/you/Library/LaunchAgents/mercurial.plist
```

You can unload the script at any time using:

```
launchctl unload /Users/you/Library/LaunchAgents/mercurial.plist
```

Enjoy

You can now access your webserver at https://yourdomain/my_hg_repo.

Nginx will ask for the password (from the `passwd.txt` file) and then show your repository.



While the copyright of this document belongs to the author, the example code can be used for any purpose and should be considered public domain.