Nginx

- https://askubuntu.com/questions/766352/multiple-websites-on-nginx-one-ip

The one-domain, multiple subdomains approach.

> This answer section assumes you have one domain and multiple subdomains therein. If you do not have this, please clarify this in your question

With each nginx server {} block in the configuration, you will need to define the server name, and likely set a fourth server block as a 'catch all' for other requests.

> Example: I have three projects, proj1, proj2, proj3. I have a domain called evil-projects.net (NOTE: Doesn't exist really). I want three different subdomains, one for each nginx configuration which will point to one project each. My server resides at 1.2.3.4, and it will serve all sites.

With the above scenario, we have two parts: domains and subdomains, and the server configuration.

(1): DNS Configuration

Set up your DNS at your host such that the following is true with the DNS records:

```
evil-projects.net  IN A  1.2.3.4
proj1.evil-projects.net  IN A  1.2.3.4
proj2.evil-projects.net  IN A  1.2.3.4
proj3.evil-projects.net  IN A  1.2.3.4
```

(2): NGINX configuration at server (1.2.3.4)

Now for your nginx configurations. I am assuming you are going to have the default nginx setups and the packages from the Repositories (I'm going to use 14.04 as a base example). We'll have four configuration files put into /etc/nginx/sites-available, first. **You may need to use sudo when creating these files, as the folder in question is owned by root.**

```
/etc/nginx/sites-available/catch-all
```

this will be the 'catch all' for any nonvalid domains. I like returning http error code 410 (GONE).

```
server {
    listen 80 default_server;

    server_name _;

    return 410;
}
```

Next, we set up the configuration for your sites/projects. I'm going to assume they're all static files, though. Each of these implies that you have different web directories for each project on the server as well (different 'document roots').

/etc/nginx/sites-available/proj1.evil-projects.net:

```
server {
    listen 80;

    server_name proj1.evil-projects.net;

    root /var/www/proj1;
    index index.htm index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

/etc/nginx/sites-available/proj2.evil-projects.net:

```
server {
    listen 80;

    server_name proj2.evil-projects.net;

    root /var/www/proj2;
    index index.htm index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

/etc/nginx/sites-available/proj3.evil-projects.net:

```
server {
    listen 80;

    server_name proj3.evil-projects.net;

    root /var/www/proj3;
```

```
    index index.htm index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

We then have to remove the 'default' configuration from /etc/nginx/sites-enabled, and add our own. Again, sudo is needed here.

```
sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/proj1.evil-projects.net
/etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/proj2.evil-projects.net
/etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/proj3.evil-projects.net
/etc/nginx/sites-enabled/
```

And then we restart the nginx process:

```
sudo systemctl restart nginx
```

Once DNS propagates, the sites will work as they should.

From:
https://www.natrius.eu/dokuwiki/ - **Natrius**

Permanent link:
https://www.natrius.eu/dokuwiki/doku.php?id=digital:server:nginx&rev=1540293384

Last update: **2018/10/23 13:16**