Website Hosting - Support #856

Setup a Landing Page With Redirection Using Nginx

10/01/2016 02:06 PM - Daniel Curtis

Status:	Closed	**Start date:**	10/02/2016
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:		**Estimated time:**	0.50 hour
Target version:	Nginx	**Spent time:**	2.00 hours

Description

One of the websites I am working on at the moment isn't quite ready for production, but the domain name has already been purchased and I want **something** to show up when visited. To accomplish this I've setup a separate folder with Oleose, an open source landing page template, and set nginx to redirect to the landing page if not connected from localhost.

This is a guide on how to setup a landing page and redirect incoming Internet connections to the landing page with nginx on FreeBSD 10.

NOTE: This setup is done without performance-costing "if" statements; but rather sets the landing page as a custom access denied page, then using nginx's built-in access control list to deny all connections to the main website except for connections from 127.0.0.1.

Install Nginx

- Install nginx:

```
pkg install nginx
```

- Start and enable nginx at boot:

```
echo 'nginx_enable="YES"' >> /etc/rc.conf
service nginx start
```

- Create a configuration directory to make managing individual server blocks easier

```
mkdir /usr/local/etc/nginx/conf.d
```

- Edit the main nginx config file:

```
vi /usr/local/etc/nginx/nginx.conf
```

  - And strip down the config file and add the include statement at the end to make it easier to handle various server blocks:

```
worker_processes  1;
error_log  /var/log/nginx-error.log;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;
    # Load config files from the /etc/nginx/conf.d directory
    include /usr/local/etc/nginx/conf.d/*.conf;
}
```

Basic Website

- Add a **newsite.example.com server block**:

```
vi /usr/local/etc/nginx/conf.d/newsite.example.com.conf
```

  - Add the following:

```
server {
    listen       80;
    server_name  newsite.example.com;
    root         /usr/local/www/newsite.example.com;
    access_log   /var/log/newsite.example.com-access.log;
    error_log    /var/log/newsite.example.com-error.log;

    location / {
      try_files $uri $uri/ /index.html;
    }
}
```

- Restart nginx:

```
service nginx restart
```

Oleose Landing Page

- Clone the oleose landing page from github:

```
cd /usr/local/www
git clone https://github.com/ScoopThemes/Oleose.git landing-page
```

- Edit the **newsite.example.com server block**:

```
vi /usr/local/etc/nginx/conf.d/newsite.example.com.conf
```

  - Add the following:

```
server {
    listen       80;
    server_name  newsite.example.com;
    root         /usr/local/www/newsite.example.com;
    access_log   /var/log/newsite.example.com-access.log;
    error_log    /var/log/newsite.example.com-error.log;

    location ~ ^/(?:CHANGELOG\.md|README.md|.git|.gitignore){
      deny all;
    }

    location / {
      error_page 403 = @internetland;
      allow 127.0.0.1;
      deny all;
      try_files $uri $uri/ /index.html;
    }

    location @internetland {
      return 301 $scheme://newsite.example.com/landing-page;
    }

    location /assets {
      alias /usr/local/www/landing-page/assets;
      index index.html
      allow all;
      try_files $uri $uri/;
    }

    location /landing-page {
      alias /usr/local/www/landing-page;
      index index.html
      allow all;
      try_files $uri $uri/ /index.html;
    }
}
```

- Restart nginx:

```
service nginx restart
```

- Now open a web browser and go to http://newsite.example.com , you will find that all requests from the internet are sent to http://newsite.example.com/landing-page

NOTE: Since the normal website is blocked to all but connections from localhost, I use an SSH tunnel to log into the web server:

```
ssh -L 8080:localhost:80 newsite.example.com
```

Then open a web browser and go to http://localhost:8080 to tunnel all HTTP traffic on port 8080 on the local machine.

Resources

- http://stackoverflow.com/questions/17993011/redirecting-all-requests-that-arent-from-my-ip-with-nginx

History

#1 - 10/01/2016 02:22 PM - Daniel Curtis

- Description updated

- Status changed from New to Resolved

- % Done changed from 0 to 100

#2 - 12/03/2016 12:49 PM - Daniel Curtis

- Status changed from Resolved to Closed

#3 - 01/02/2017 08:35 PM - Daniel Curtis

- Description updated