NginX Reverse Proxy

Create our Docker Container

Create a nginx reverse proxy by issuing the following command:

```
docker run -d \
--net host \
--restart=always \
-p 80:80 \
--name proxy \
-v $PWD/conf:/etc/nginx/sites-enabled \
-v $PWD/letsencrypt:/etc/letsencrypt \
-v $PWD/conf.d:/etc/nginx/conf.d \
lerenn/nginx-reverse-proxy
```

This will create a reverse proxy running on the host network.

Define our Nginx Configuration Files

In the conf folder(mapped to sites-enabled) we defined in our docker command we will add a configuration like the following:

wiki.conf

```
server {
        listen       80;
        server_name  wiki wiki.jmehan.com;
        location / {
            proxy_pass         http://192.168.1.60:8090/;
        }
}
```

Adding SSL Support (not certbot)

If we want to terminate an SSL connection at our proxy, we can generate an SSL cert and configure it in nginx.

Generate the SSL certificate using the following command:

> openssl req -nodes -new -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 7300

This command will generate a self signed SSL certificate valid for 10 years.

Configure the endpoint to use the certificates. Here we are defining the docker location for the certs.

mysite.conf

```
server {
        listen       8443 ssl;
        #server_name  svn svn.jmehan.com;
        ssl_certificate     /etc/nginx/certificates/svn/cert.pem;
        ssl_certificate_key /etc/nginx/certificates/svn/key.pem;

        location / {
            proxy_pass         http://192.168.1.60:9080/;
            proxy_set_header   Host             $host;
            proxy_set_header   X-Real-IP        $remote_addr;
            proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;

            client_max_body_size       10m;
            client_body_buffer_size    128k;
            proxy_connect_timeout      90;
            proxy_send_timeout         90;
            proxy_read_timeout         90;

            proxy_buffer_size          4k;
            proxy_buffers              4 32k;
            proxy_busy_buffers_size    64k;
            proxy_temp_file_write_size 64k;
        }
}
```

Adding Basic Authentication

You may need to add apache2-utils to your nginx docker container using the following cmd:

> sudo apt-get install apache2-utils

Login to the docker container and create the password file

> docker exec -it <nginx_container> bash

Create a password file

> htpasswd -c /etc/nginx/conf.d/htpasswd <username>

Update the configuration

```
server {
        server_name  kibana kibana.jmehan.com;
        location / {
            proxy_pass         http://192.168.1.60:5601/;
            auth_basic "Administrator's Area";
            auth_basic_user_file /etc/nginx/conf.d/htpasswd;
        }
}
```

Supporting Sites that use websockets

```
server {
        server_name  homebridge homebridge.jmehan.com;
        location / {
            proxy_pass         http://192.168.1.60:8089/;
            proxy_http_version          1.1;
            proxy_buffering             off;
            proxy_set_header            Host $host;
            proxy_set_header            Upgrade $http_upgrade;
            proxy_set_header            Connection "Upgrade";
            proxy_set_header            X-Real-IP $remote_addr;
            proxy_set_header            X-Forward-For $proxy_add_x_forwarded_for;
        }
}
```

Redirecting all traffic to SSL

```
server {
    server_name  www.server.com server.com;
    listen 443 ssl;
    location / {
         proxy_pass         http://192.168.1.60:12345/;
    }
    ssl_certificate /etc/letsencrypt/live/www.server.com-0001/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/www.server.com-0001/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    server_name  www.server.com server.com;
    listen 80;
    return 301 https://www.diabetease.com$request_uri;
}
```

Forwarding Real IP Address

Add **X-Real-IP** and **X-Forwarded-For** headers using the **proxy_set_header** instruction by adding it to the **/etc/nginx/conf.d/proxy.conf** file.

```
proxy_redirect          off;
proxy_set_header        Host            $host;
proxy_set_header        X-Real-IP       $remote_addr;
proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
client_max_body_size    10m;
client_body_buffer_size 500m;
client_header_buffer_size 500m;
proxy_connect_timeout   90;
proxy_send_timeout      90;
proxy_read_timeout      90;
proxy_buffer_size   16k;
proxy_buffers       32   16k;
proxy_busy_buffers_size 64k;
```

Restricting Access to IP Range

In the following example, we restrict access to a login page in confluence to internal ip addresses between: 192.168.1.100-255

See https://www.ipaddressguide.com/cidr for creating ip range.

```
        # restrict access to login to 192.168.1.100-255
        location /login.action {
            allow 192.168.1.100/30;
            allow 192.168.1.104/29;
            allow 192.168.1.112/28;
            allow 192.168.1.128/25;
            deny all;
            proxy_pass         http://192.168.1.50:8090/login.action;
        }
```

Customized Dockerfile

The following Dockerfile adds **certbot** and **apache2-utils** to our nginx-reverse-proxy image.

Dockerfile

```
FROM lerenn/nginx-reverse-proxy

RUN apt-get update
RUN apt-get install -y wget
RUN apt-get install -y apache2-utils
RUN wget https://dl.eff.org/certbot-auto
RUN chmod +x certbot-auto
RUN ./certbot-auto -n --install-only
```

References

Reference	URL
Let's Encrypt	CertBot and Let's Encrypt
Restricting Access with HTTP Basic Authentication	https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/