

Apache Httpd SSL Proxy

- [Apache Httpd SSL Proxy](#)
 - [Building the image](#)
 - [Run an SSL Termination Proxy from the CLI](#)
 - [Create DHE Param](#)
 - [Launch a Container](#)
 - [Enable Basic Access authentication](#)
 - [Add additional Apache httpd configuration](#)
 - [Generate SSL Certificates](#)
 - [Verification of CSR](#)

This repository is used to build a Docker image that acts as an HTTP [reverse proxy](#) with optional (but strongly encouraged) support for acting as an [SSL termination proxy](#). The proxy can also be configured to enforce [HTTP basic access authentication](#)

[Apache httpd](#) is the HTTP server, and its SSL configuration is included (and can be modified to suit your needs) in [apache/httpd.conf](#) in this repository.

Building the image

You can build the image yourself by cloning this repository and running :

```
make build
```

Run an SSL Termination Proxy from the CLI

To run an SSL termination proxy you must have an existing SSL certificate and key. These instructions assume they are stored at `/path/to/secrets` and named `myserver.crt` and `myserver.pem`. You'll need to change those values based on your actual file path and names.

Create DHE Param

The Apache SSL configuration for this image also requires that you generate your own DHE parameter. It's easy and takes just a few minutes to complete :

```
openssl dhparam -out /path/to/secrets/dhparam.pem 2048
```

Launch a Container

Modify the below command to include the actual address or host name you want to proxy to, as well as the correct `/path/to/secrets` for your certificate, key and dhparam :

```
make run  
or
```

```
docker run -ti \
-e ENABLE_SSL=true \
-e ENABLE_BASIC_AUTH=true \
-e TARGET_SERVICE=backend.com \
-v $(shell pwd)/myserver.crt:/apache/myserver.crt \
-v $(shell pwd)/myserver.key:/apache/myserver.key \
-v $(shell pwd)/htpasswd:/apache/htpasswd \
allamand/apache-ssl-proxy $(ARGS)
```

Enable Basic Access authentication

Create an htpasswd file :

```
htpasswd -nb you_USERNAME SUPER_SECRET_PASSWORD > /path/to/secrets/htpasswd
```

Launching the container, enabling the feature and mapping in the htpasswd file :

```
docker run -ti \
-e ENABLE_SSL=true \
-e ENABLE_BASIC_AUTH=true \
-e TARGET_SERVICE=backend.com \
-v $(shell pwd)/myserver.crt:/apache/myserver.crt \
-v $(shell pwd)/myserver.key:/apache/myserver.key \
-v $(shell pwd)/htpasswd:/apache/htpasswd \
allamand/apache-ssl-proxy $(ARGS)
```

Add additional Apache httpd configuration

All .conf from apache/extra are added during the build to /etc/apache2/extra and get included on startup of the container. Using volumes you can overwrite them on start of the container:

```
docker run \
-e ENABLE_SSL=true \
-e TARGET_SERVICE=THE_ADDRESS_OR_HOST_YOU_ARE_PROXYING_TO \
-v /path/to/secrets/cert.crt:/etc/secrets/proxycert \
-v /path/to/secrets/key.pem:/etc/secrets/proxykey \
-v /path/to/secrets/dhparam.pem:/etc/secrets/dhparam \
-v /path/to/additional-nginx.conf:/etc/nginx/extra-conf.d/additional_proxy.conf \
nginx-ssl-proxy
```

That way it is possible to setup additional proxies or modifying the apache configuration

Generate SSL Certificates

We can generate Auto-signed SSL certificates using docker image [CertUtil](#)

We give your IP or domain name to the container and it will create a CSR so you can ask a verified certificate, but it will also generate an auto-signed certificate so that you can continue for testing

```
$ make IP=84.39.10.12

Creating 84.39.10.12 repository
mkdir 84.39.10.12
Create conf file
cp openssl.cnf.tpl 84.39.10.12/openssl.cnf
sed -i "s/{ IP }/84.39.10.12/g" 84.39.10.12/openssl.cnf
generate server key
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out
84.39.10.12/84.39.10.12.xip.io.key.pem -outform pem
```

```
openssl req -new -config 84.39.10.12/openssl.cnf -sha256 -key 84.39.10.12/84.39.10.12.xip.io.key.pem -keyform
pem -subj "/C=FR/ST=Alpes\ Maritimes/L=Sophia/O=Oran
ge/OU=Hebex/CN=84.39.10.12.xip.io/emailAddress=dfy.hbx.sec.all@list.orangeportails.net" -out
84.39.10.12/84.39.10.12.xip.io.csr.pem -outform pem -verbose
```

Verification of CSR

```
openssl req -in 84.39.10.12/84.39.10.12.xip.io.csr.pem -noout -text
```