Apache Server through Proxy

This guide covers setting up a proxied Apache Server. This is an Apache Server which is accessed through a Proxy Server, and itself is not directly visible to the “outside”. We therefore do not need SSL or any other access protection, as this needs to be handled by the Proxy Server.

I'm using such a server behind a proxy for BackupPC and Zabbix services. There are a few particular things to observe for these services, for example that apache should run as user *backuppc*. If you do not need this, e.g. if you do not install BackupPC, then you might want to replace the apache user and group with the default *www-data* instead of *backuppc*.

Apache Installation

1. Install apache

```
$ sudo apt update
$ sudo apt install apache2
```

2. Chose *backuppc* as apache user during the installation. User *backuppc* gets added through installing BackupPC.
3. Setup your virtual hosts
4. Create sub folders in `/var/log/apache2` if you setup log files for the virtual hosts in sub folders
5. Read my guide for the front-end Apache Installation for Apache Settings, PHP Installation, Xdebug, Windows Subsystem for Linux, and SSL for Localhost.

Proxy

- Setup a VirtualHost on your front-end (main) apache server, further referred to as “proxy”.
- Setup a VirtualHost on your proxied apache server (running backuppc or zabbix), further referred to as “host”.
- The SSL certificates are served from the “proxy” through access to https://sub.domain.tld
- The “host” serves an unencrypted site through port 8080, 8081, etc. Using different ports for different services allows you to configure the “host” per service, and you don't need folders to differentiate the service or the served site.
- This assumes your local network is secure to the “outside” world.

VirtualHost on the "proxy"

- sub.domain.tld: external domain name with which you access the “host” behind the “proxy”
- host.yourdomain.tld: internal domain name of your “host”. This could also be your LAN IP address of the “host”.

```
<VirtualHost *:80>
    ServerName sub.domain.tld
```

```
    Redirect 301 / https://sub.domain.tld
<VirtualHost>
<VirtualHost *:443>
    ServerName sub.domain.tld
    ServerAdmin you@domain.tld
    DocumentRoot /var/www/html/yoursite
    SSLEngine on
    RedirectMatch ^/$ /yourapp/ # use this if backuppc is not the default
app, or if you need to access another app on the same server
    # your main service access
    <Location "/">
          ProxyPass "http://host.yourdomain.tld:8080/"
          ProxyPassReverse "http://host.yourdomain.tld:8080/"
          # you should consider to restrict access by ip address or other
means, if the proxied server is not supposed to be accessible by public
          Require all granted
    </Location>
    # special files access which do not belong to the service, for example
access to phpinfo()
    <Location "/yourapp/">
          ProxyPass "http://host.yourdomain.tld:8080/yourapp/"
          ProxyPassReverse "http://host.yourdomain.tld:8080/yourapp/"
          # you should consider to restrict access by ip address or other
means, if //yourapp// is not supposed to be accessible by public
          Require all granted
    </Location>
    # make sure nobody gets the htaccess, README, COPYING or VERSION files
    <Files ~ "^([\._]ht|README$|VERSION$|COPYING$)">
          Require all denied
    </Files>
    # add other options such as Files and Directory permissions
    Include /etc/letsencrypt/options-ssl-apache.conf
    SSLCertificateFile /etc/letsencrypt/live/sub.domain.tld/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/sub.domain.tld/privkey.pem
</VirtualHost>
```

VirtualHost on the "host"

- This is an example for a BackupPC installation on Debian 11.
- Remove the symlink to BackupPC's `apache.conf` file in `/etc/apache2/conf-enabled` and put everything into the VirtualHost file.
- Note that the `/backuppc` alias is necessary for BackupPC to serve the pages correctly.

```
# This is a proxied server, SSL is handled by the proxy
<VirtualHost *:8080>

    ServerAdmin you@domain.tld
    DocumentRoot /usr/share/backuppc/cgi-bin
    Alias /backuppc /usr/share/backuppc/cgi-bin/
    Alias /html/ /home/bco/html/
```

```
    <Directory /usr/share/backuppc/cgi-bin/>
        AllowOverride None

        Options ExecCGI FollowSymlinks
        AddHandler cgi-script .cgi
        DirectoryIndex index.cgi

            AuthUserFile /etc/backuppc/htpasswd
        AuthType basic
        AuthName "BackupPC admin"

            <RequireAll>
            # Comment out this line once you have setup HTTPS and
uncommented SSLRequireSSL
            Require all granted

            # This line ensures that only authenticated users may access
your backups
            Require valid-user
        </RequireAll>
    </Directory>

    <Directory /home/bco/html/>
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/backuppc-error.log
    CustomLog ${APACHE_LOG_DIR}/backuppc-access.log combined
</VirtualHost>
```

Firewall

- If your “proxy” and your “host” are on different networks you will need to set firewall rules for the “proxy” to be able to connect to “host”

Links

- Use apache as a HTTPS to HTTP Proxy
- Reverse Proxy Guide
- Simple Apache reverse proxy example
- Two apache servers on same machine with same port
- How we can run two instance of Apache Http Server on same machine
- Running Multiple Apache Instances

From:
http://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
http://wiki.condrau.com/deb11:apache-proxy

Last update: **2023/01/18 12:21**