Apache HTTP Server

Для работы серверных компонентов системы на ОС Linux требуется веб-сервер, работающий в режиме работы обратного прокси-сервера.
Apache — самый популярный в мире HTTP-сервер, имеет возможность работы в режиме обратного прокси-сервера, продукт позиционируется производителем как функциональный, надежный и гибкий в настройке.

Установка apache

Чтобы установить apache, выполните следующие команды в зависимости от используемого дистрибутива или выбрав сборку из исходного кода:

RHEL и производные дистрибутивы

```
sudo yum install httpd
sudo systemctl enable httpd
sudo systemctl start httpd
```

Debian и производные дистрибутивы

```
sudo apt install apache2
sudo systemctl enable apache2
sudo service apache2 start
```

Установка из исходного кода

Скачайте последний релиз с портала производителя.

Из папки со скачанным дистрибутивом выполните:
(при выполнении команд замените "NN" на версию скачанного дистрибутива, например, 2.4.57)

```
gzip -d httpd-NN.tar.gz
tar xvf httpd-NN.tar
cd httpd-NN
./configure --prefix=/usr/local/apache2
make
make install
/usr/local/apache2/bin/apachectl -k start
```

Требования к установке, а также подробнее процесс установки из исходного кода описан на портале производителя.

Выпуск SSL/TLS сертификата

Для настройки защищенного соединения необходимо выпустить SSL/TLS сертификат на имя рабочей станции с установленным apache.
Возможно выпустить самоподписанный сертификат или сертификат с УЦ.

Самоподписанный сертификат

1. Создайте самоподписанный сертификат утилитой openssl (вместо "*SERVER_FQDN*" подставьте DNS-имя рабочей станции с apache):

   ```
   sudo openssl req -x509 -nodes -addext "subjectAltName=DNS:SERVER_FQDN,DNS:www.SERVER_FQDN" -days 730 -newkey rsa:2048 -keyout /etc/apache2/ssl/SSL.key -out /etc/apache2/ssl/SSL.crt
   ```

2. Добавьте сертификат в доверенные на локальной рабочей станции в соответствии с выбранной для настройки ОС.

   Для RHEL и производных дистрибутивов:

   ```
   sudo cp /etc/apache2/ssl/SSL.crt /etc/pki/ca-trust/source/anchors/SSL.crt
   sudo update-ca-trust extract
   ```

   Для Debian и производных дистрибутивов:

   ```
   sudo cp /etc/apache2/ssl/SSL.crt /usr/local/share/ca-certificates/
   sudo update-ca-certificates -f
   ```

3. Сделайте сертификат доверенным в домене, например, с помощью групповых политик.

Выпуск сертификата на УЦ

1. Выпустите сертификат на УЦ, например на Microsoft CA, экспортируйте данный сертификат в формате *.pfx* (с закрытым ключом, с цепочкой корневых /промежуточных УЦ) на рабочую станцию с установленным apache.

   ⚠ **Субъект** (Subject) сертификата должен содержать FQDN сервера Apache.
   Дополнительное имя субъекта (Subject Alternative Name) сертификата должно содержать атрибут **DNS-имя** (DNS Name) (FQDN сервера Apache). Например: *astra.demo.local* или соответствующую запись с подстановочными знаками, например: **. demo.local* (Wildcard certificate).
   Улучшенный ключ (Enhanced Key Usage) сертификата должен содержать значение **Проверка подлинности сервера** (Server Authentication).

2. Разделите *.pfx* сертификат на файл цепочки сертификатов и ключ, сделайте файл закрытого ключа без пароля (необходимо подставить имя импортированного файла вместо PFXFILE):

```
openssl pkcs12 -in PFXFILE.pfx -chain -nokeys | sed -ne '/-BEGIN CERTIFICATE/,/END CERTIFICATE/p' > SSL.crt
openssl pkcs12 -in PFXFILE.pfx -cacerts -nokeys | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > root-ca.crt
openssl pkcs12 -in PFXFILE.pfx -nocerts -out SSLencrypted.key
openssl rsa -in SSLencrypted.key -out SSL.key
rm SSLencrypted.key
```

   Файл цепочки сертификатов *SSL.crt* должен быть следующего вида:

```
-----BEGIN CERTIFICATE-----
#Ваш сертификат#
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
#Промежуточный сертификат#
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
#Корневой сертификат#
-----END CERTIFICATE-----
```

3. Добавьте на рабочую станцию с установленным apache сертификат корневого УЦ в список доверенных.

   Для RHEL и производных дистрибутивов:

```
sudo cp root-ca.crt /etc/pki/ca-trust/source/anchors/
sudo update-ca-trust extract
```

   Для Debian и производных дистрибутивов:

```
sudo cp root-ca.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates -f
```

4. Скопируйте файлы цепочки сертификатов и ключа в папку, которая будет указана в найстройках сайта apache:

```
sudo cp SSL.crt /etc/apache2/ssl/
sudo cp SSL.key /etc/apache2/ssl/
```

Настройка модулей и конфигурации

Apache реализован в виде ядра и модулей, которые подключаются по необходимости использования дополнительной функциональности.

1. Для работы системы включите модули:

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod ssl
sudo a2enmod headers
sudo a2enmod rewrite
sudo systemctl restart apache2
```

2. Добавьте следующие директивы в конфигурационный файл Apache (*apache2.conf*), по умолчанию расположенный по пути */etc/apache2/apache2.conf*.
(в данном месте и далее замените *"SERVER_FQDN"* на имя используемого сервера)

```
Listen 3003
LimitRequestLine 16384
LimitRequestFieldSize 16384
ServerName SERVER_FQDN
Header append X-FRAME-OPTIONS "SAMEORIGIN"
Header set X-Content-Type-Options "nosniff"
```

При установке Apache в **ОС Astra Linux** в файле *apache2.conf* может потребоваться отключение параметра AstraMode, подробнее на портале Astra Linux.

Настройка сайта

Для работы Indeed CM требуется создать сайт в Apache, чтобы он обслуживал запросы и отправлял их на проксируемый адрес (сервис Indeed CM).

1. Создайте файл сайта */etc/apache2/sites-available/SERVER_FQDN.conf*

```
sudo touch /etc/apache2/sites-available/SERVER_FQDN.conf
```

2. Заполните файл рекомендуемым содержимым:

В параметрах **SSLCertificateFile** и **SSLCertificateKeyFile** указаны пути к созданным/импортированным в предыдущих шагах файлам сертификата и закрытого ключа, требуется проверить указанные пути и имена файлов.

```
<VirtualHost *:80>
```

```
    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI}/$1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
    Protocols h2 http/1.1
    SSLCertificateFile /etc/apache2/ssl/SSL.crt
    SSLCertificateKeyFile /etc/apache2/ssl/SSL.key
    SSLCipherSuite @SECLEVEL=1:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-
GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-
GCM-SHA256:DHE-RSA-AES256-GCM-SHA384

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on
    SSLProtocol -all +TLSv1.2
    SSLHonorCipherOrder off
    SSLCompression off
    SSLSessionTickets on
    SSLUseStapling off
    SSLProxyEngine on
    SetEnv nokeepalive ssl-unclean-shutdown
    RequestHeader set X-Forwarded-Proto https
    Header always set Strict-Transport-Security "max-age=63072000"

    ProxyPreserveHost On

        ProxyPass /cm/mc http://localhost:5001/cm/mc
        ProxyPassReverse /cm/mc http://localhost:5001/cm/mc

        ProxyPass /cm/ss http://localhost:5002/cm/ss
        ProxyPassReverse /cm/ss http://localhost:5002/cm/ss

        ProxyPass /cm/rss http://localhost:5003/cm/rss
        ProxyPassReverse /cm/rss http://localhost:5003/cm/rss

        ProxyPass /cm/api http://localhost:5004/cm/api
        ProxyPassReverse /cm/api http://localhost:5004/cm/api

        ProxyPass /cm/credprovapi http://localhost:5005/cm/credprovapi
        ProxyPassReverse /cm/credprovapi http://localhost:5005/cm/credprovapi

        ProxyPass /cm/oidc http://localhost:5008/cm/oidc
        ProxyPassReverse /cm/oidc http://localhost:5008/cm/oidc

        ProxyPass /cm/wizard http://localhost:5009/cm/wizard
        ProxyPassReverse /cm/wizard http://localhost:5009/cm/wizard

</VirtualHost>

<VirtualHost *:3003>
```

```
    protocols h2 http/1.1

    SSLCertificateFile /etc/apache2/ssl/SSL.crt
    SSLCertificateKeyFile /etc/apache2/ssl/SSL.key
    SSLCipherSuite @SECLEVEL=1:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-
GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-
GCM-SHA256:DHE-RSA-AES256-GCM-SHA384

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on
    SSLProtocol -all +TLSv1.2
    SSLHonorCipherOrder off
    SSLCompression off
    SSLSessionTickets on
    SSLUseStapling off
    SSLProxyEngine on
    RequestHeader set X-Forwarded-Proto https
    Header always set Strict-Transport-Security "max-age=63072000"

        ProxyPass /agentregistrationapi http://localhost:5006/agentregistrationapi
        ProxyPassReverse /agentregistrationapi http://localhost:5006/agentregistrationapi

    <Location "/agentserviceapi">
          SSLVerifyClient optional_no_ca
          SSLOptions +ExportCertData
          RequestHeader unset x-ssl-client-cert
          RequestHeader set x-ssl-client-cert "expr=%{escape:%{SSL_CLIENT_CERT}}"
        #RequestHeader set x-ssl-client-cert "expr=%{escape:%{SSL_CLIENT_S_DN}}"

         ProxyPass http://localhost:5007/agentserviceapi
          ProxyPassReverse http://localhost:5007/agentserviceapi
    </Location>
</VirtualHost>
```

3. Перечитайте файл конфигурации и включите файл сайта:

```
sudo a2ensite SERVER_FQDN
sudo apachectl configtest
sudo systemctl restart apache2
```