

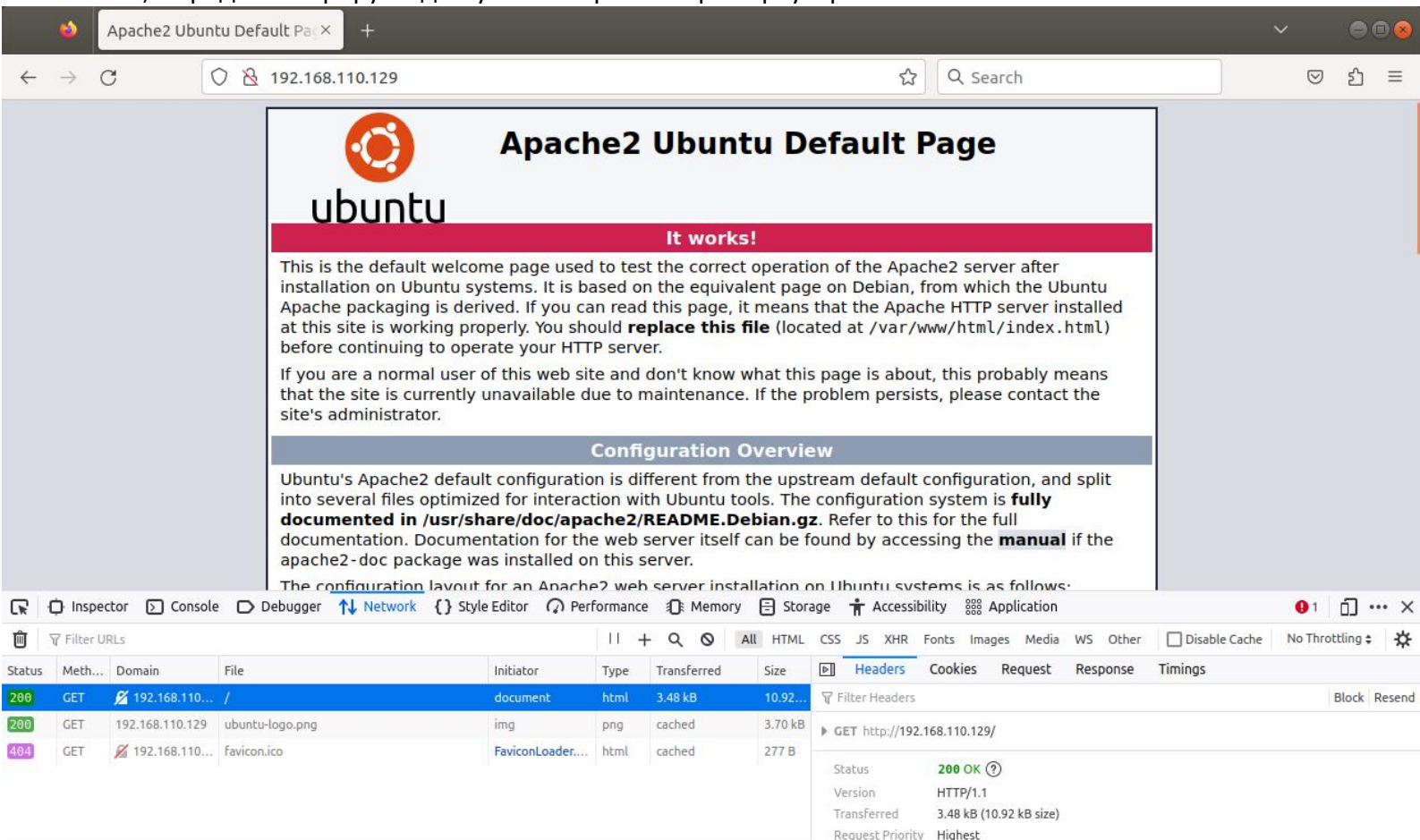
## Первая часть

- 1) Apache Server был мной успешно установлен и запущен. Проверим это и убедимся, что отсутствуют ошибки и предупреждения:

```
user1@ubuntu:~$ sudo systemctl status apache2.service
[sudo] password for user1:
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Thu 2025-10-09 12:02:26 PDT; 19min ago
     Process: 2225 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
     Process: 1011 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 1056 (apache2)
       Tasks: 55 (limit: 4620)
      CGroup: /system.slice/apache2.service
              └─1056 /usr/sbin/apache2 -k start
                2230 /usr/sbin/apache2 -k start
                2231 /usr/sbin/apache2 -k start

Oct 09 12:02:26 ubuntu systemd[1]: Starting The Apache HTTP Server...
Oct 09 12:02:26 ubuntu systemd[1]: Started The Apache HTTP Server.
Oct 09 12:07:24 ubuntu systemd[1]: Reloading The Apache HTTP Server.
Oct 09 12:07:24 ubuntu systemd[1]: Reloaded The Apache HTTP Server.
```

- 2) Продемонстрируем доступность Apache через браузер:



The screenshot shows a web browser window displaying the Apache2 Ubuntu Default Page. The page features the Ubuntu logo and the text "Apache2 Ubuntu Default Page". Below this, a red banner reads "It works!". The main content area contains a message about the default welcome page and a "Configuration Overview" section. The browser's developer tools are open, showing the Network tab with a list of requests and the Headers tab for the selected request.

Status	Meth...	Domain	File	Initiator	Type	Transferred	Size
200	GET	192.168.110...	/	document	html	3.48 kB	10.92...
200	GET	192.168.110.129	ubuntu-logo.png	img	png	cached	3.70 kB
404	GET	192.168.110...	favicon.ico	FaviconLoader...	html	cached	277 B

The Headers tab for the selected request shows the following details:

- Status: 200 OK
- Version: HTTP/1.1
- Transferred: 3.48 kB (10.92 kB size)
- Request Priority: Highest

### 3) Сгенерируем самоподписанные SSL-сертификаты для Apache через openssl:

```
user1@ubuntu:/etc/apache2/certificate$ sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key
Can't load /home/user1/.rnd into RNG
140467580576192:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/home/user1/.rnd
Generating a RSA private key
.....
++++
writing new private key to 'apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:Moscow
Locality Name (eg, city) []:Moscow
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Skillfactory
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:192.168.110.129
Email Address []:a@gmail.com
```

### 4) Настроим Apache для использования SSL-сертификатов (а также проксирования запросов на server.py, запущенном на порту 8080 хоста 192.168.110.128):

```
user1@ubuntu:~$ sudo cat /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

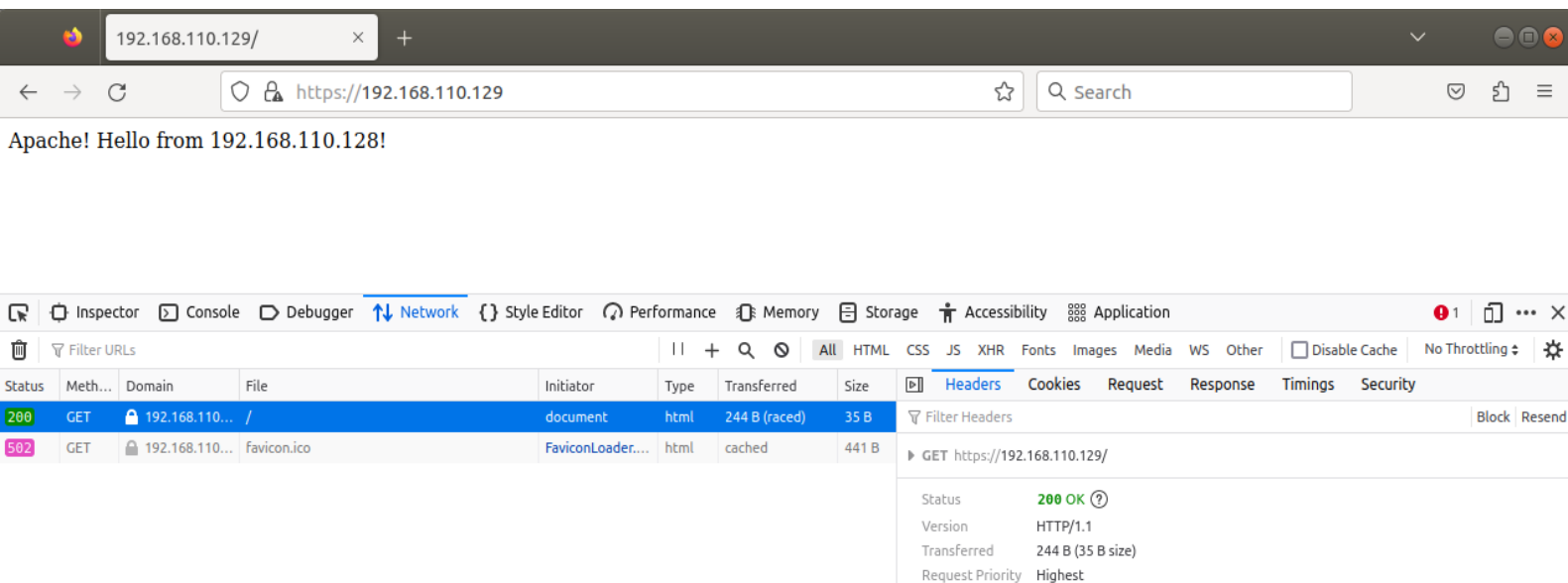
    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/certificate/apache.key
    ProxyPass / http://192.168.110.128:8080
    ProxyPassReverse / http://192.168.110.128:8080
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- 5) Проверим доступности веб-сервера через HTTPS и корректность работы SSL-сертификата. Видим, что запрос выполняется, а также что происходит перенаправление на server.py:



## Вторая часть

- 1) Nginx был мной успешно установлен и запущен. Проверим это и убедимся, что отсутствуют ошибки и предупреждения:

```
user1@user1-virtual-machine:~$ sudo systemctl status nginx
[sudo] password for user1:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-10-08 22:09:57 MSK; 24h ago
     Docs: man:nginx(8)
  Process: 24316 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 24317 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 24318 (nginx)
    Tasks: 3 (limit: 4539)
   Memory: 23.4M
      CPU: 571ms
   CGroup: /system.slice/nginx.service
           └─24318 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─24319 "nginx: worker process"
               └─24320 "nginx: worker process"

окт 08 22:09:57 user1-virtual-machine systemd[1]: Starting A high performance web server and a reverse proxy server...
окт 08 22:09:57 user1-virtual-machine systemd[1]: Started A high performance web server and a reverse proxy server.
```

- 2) Настроим nginx в качестве прямого прокси (8.8.8.8 – прокси-сервер от Google):

```
user1@user1-virtual-machine:~$ sudo cat /etc/nginx/sites-available/config2
server {
    listen 8080;
    server_name localhost;

    resolver 8.8.8.8;

    location / {
        proxy_pass http://$http_host$uri$is_args$args;
    }
}
```

- 3) Проверим работу прямого прокси через curl --proxy. Видим, что ответ возвращается:

```
user1@user1-virtual-machine:~/ModSecurity$ curl --proxy http://localhost:8080/ http://www.faqs.org/faqs/
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <title>Internet FAQ Archives - Online Education - faqs.org</title>

  <meta name="Description" content="" />
  <meta name="robots" content="index, follow" />
```

#### 4) Сгенерируем самоподписанные SSL-сертификаты для nginx через openssl:

[illegible]

5) Настроим nginx для использования SSL-сертификатов И в качестве обратного прокси (немного обрезано справа):

```

user1@user1-virtual-machine:~$ sudo cat /etc/nginx/sites-available/config1
[sudo] password for user1:
# upstream мурроху блок определяет бэкенд-сервер, на который будет выполняться проксирование трафика.
# Здесь указан бэкенд-сервер с локальным адресом 127.0.0.1 и портом 81.
upstream мурроху {
    least_conn;
    server 127.0.0.1:81;
    server 127.0.0.1:82;
}

# server блок, слушающий HTTP-трафик на порту 80 и перенаправляющий его на HTTPS.
server {
    listen          80; # Прослушивание HTTP-трафика на порту 80.
    server_name     localhost; # Указание доменного имени сервера.

    # Возврат 301 (перманентное перенаправление) на HTTPS-версию сайта.
    return 301 https://localhost/;
}

# server блок, слушающий HTTPS-трафик на порту 443 и обрабатывающий его с использованием SSL/TLS.
server {
    listen 443 ssl; # Прослушивание HTTPS-трафика на порту 443.
    server_name localhost; # Указание доменного имени сервера.

    # Включение поддержки SSL/TLS.

    # Пути к SSL-сертификату и приватному ключу.
    ssl_certificate      /etc/ssl/private/nginx.crt;
    ssl_certificate_key  /etc/ssl/private/nginx.key;

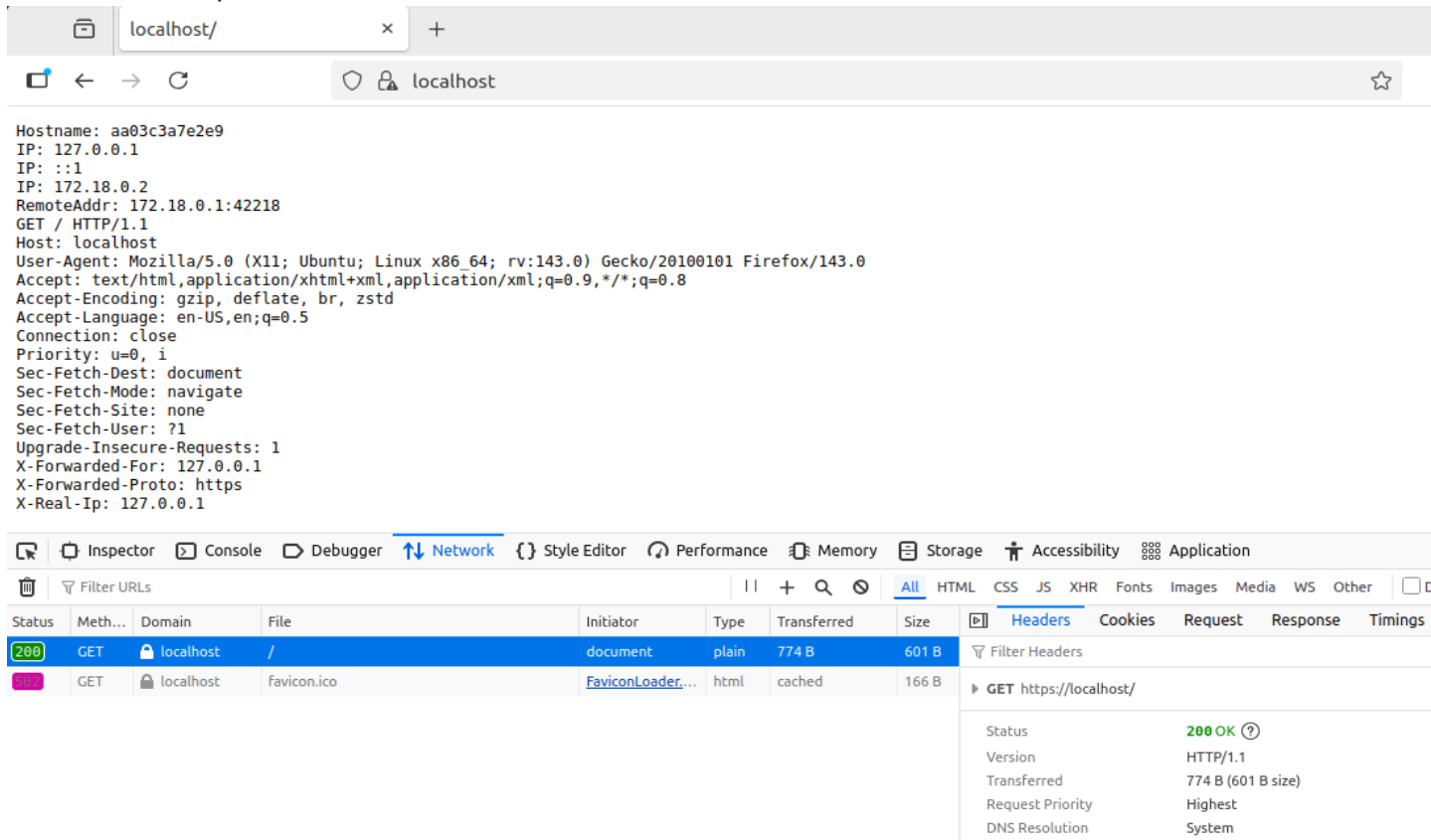
    # Опции безопасности SSL/TLS (протоколы, шифры, предпочтительные шифры).
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:ECDHE-RSA-AES128_GCM_SHA256:
    ssl_prefer_server_ciphers off;

    location / {
        # Настройки проксирования трафика на бэкенд-сервер (мурроху).
        proxy_pass http://мурроху;
        proxy_ssl_server_name on;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```



- 6) Проверим доступность веб-сервера через HTTPS, корректность работы SSL-сертификата И работу обратного прокси. Видим, что запрос выполняется, а также что происходит перенаправление на один из Docker-контейнеров:



Hostname: aa03c3a7e2e9  
IP: 127.0.0.1  
IP: ::1  
IP: 172.18.0.2  
RemoteAddr: 172.18.0.1:42218  
GET / HTTP/1.1  
Host: localhost  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:143.0) Gecko/20100101 Firefox/143.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, br, zstd  
Accept-Language: en-US,en;q=0.5  
Connection: close  
Priority: u=0, i  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: none  
Sec-Fetch-User: ?1  
Upgrade-Insecure-Requests: 1  
X-Forwarded-For: 127.0.0.1  
X-Forwarded-Proto: https  
X-Real-IP: 127.0.0.1

Status	Meth...	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost	/	document	plain	774 B	601 B
	GET	localhost	favicon.ico	FaviconLoader...	html	cached	166 B

GET https://localhost/

Status: 200 OK  
Version: HTTP/1.1  
Transferred: 774 B (601 B size)  
Request Priority: Highest  
DNS Resolution: System

- 7) Убедимся, что модуль ModSecurity установлен, и включим его в конфигурационном файле nginx.conf:

```
user1@user1-virtual-machine:~$ ls -l /usr/lib/nginx/modules
total 632
-rw-r--r-- 1 root root 19024 abr 22 15:46 ngx_http_geoip2_module.so
-rw-r--r-- 1 root root 31872 abr 22 15:46 ngx_http_image_filter_module.so
-rwxr-xr-x 1 root root 244672 окт 8 01:31 ngx_http_modsecurity_module.so
-rw-r--r-- 1 root root 27672 abr 22 15:46 ngx_http_xslt_filter_module.so
-rw-r--r-- 1 root root 108168 abr 22 15:46 ngx_mail_module.so
-rw-r--r-- 1 root root 18896 abr 22 15:46 ngx_stream_geoip2_module.so
-rw-r--r-- 1 root root 184904 abr 22 15:46 ngx_stream_module.so
```

```
GNU nano 6.2 /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
load_module modules/ngx_http_modsecurity_module.so;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    modsecurity on;
    modsecurity_rules_file /etc/nginx/modsec/main.conf;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;
```

- 8) Настроим правила фильтрации ModSecurity ([базовые правила](#) и [Core Rules Set](#)):

```
user1@user1-virtual-machine:~$ sudo cat /etc/nginx/modsec/main.conf
Include /etc/nginx/modsec/modsecurity.conf
Include /usr/local/owasp-modsecurity-crs-3.0.0/crs-setup.conf
Include /usr/local/owasp-modsecurity-crs-3.0.0/rules/*.conf
```

Примечание: SecRuleEngine поставлен как on в modsecurity.conf. Правила для защиты от SQL-инъекций расположены в rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf

- 9) Проверим, что фильтрация SQLI работает. Возвращает 403 статус:

```
user1@user1-virtual-machine:~$ curl -I 'https://localhost/?param="";DROP+TABLE+users"'
HTTP/1.1 403 Forbidden
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 09 Oct 2025 22:15:32 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
```

- 10) Переименуем файл, отвечающий за защиту от SQLI (я посчитал это более целесообразным, чем комментирование каждой строчки):

```
user1@user1-virtual-machine:~$ cd /usr/local/owasp-modsecurity-crs-3.0.0/rules
user1@user1-virtual-machine:/usr/local/owasp-modsecurity-crs-3.0.0/rules$ sudo mv REQUEST-942-APPLICATION-ATTACK-SQLI.conf REQUEST-942-APPLICATION-ATTACK-SQLI.conf.disabled
```

- 11) Убедимся, что фильтрация SQLI больше не работает. Возвращает 200 статус:

```
user1@user1-virtual-machine:~$ curl -I 'https://localhost/?param="";DROP+TABLE+users"'
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 09 Oct 2025 22:21:23 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 284
Connection: keep-alive
```