Задание 4.4.1

Первая часть

1. **Установить Apache и убедиться, что он работает через команду**

   - Установка Apache

```
korolvd@korolvd-VirtualBox:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0
0 upgraded, 9 newly installed, 0 to remove and 40 not upgraded.
Need to get 2 062 kB of archives.
After this operation, 8 234 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8
buntu0.22.04.2 [108 kB]
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.
1-5ubuntu4.22.04.2 [92,8 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlit
3 amd64 1.6.1-5ubuntu4.22.04.2 [11,3 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd6
 1.6.1-5ubuntu4.22.04.2 [9 170 B]
Get:5 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1buil
1 [140 kB]
Get:6 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4
52-1ubuntu4.12 [1 348 kB]
Get:7 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.
2-1ubuntu4.12 [165 kB]
Get:8 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2
4.52-1ubuntu4.12 [89,1 kB]
Get:9 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-
ubuntu4.12 [97,9 kB]
Fetched 2 062 kB in 13s (154 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 181451 files and directories currently installed.)
Preparing to unpack .../0-libapr1_1.7.0-8ubuntu0.22.04.2_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-8ubuntu0.22.04.2) ...
```

   - *Проверка работоспособности через команду*

```
korolvd@korolvd-VirtualBox:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enable>
     Active: active (running) since Sun 2024-12-15 20:16:35 +05; 5min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3042 (apache2)
      Tasks: 55 (limit: 4608)
     Memory: 5.3M
        CPU: 42ms
     CGroup: /system.slice/apache2.service
             ├─3042 /usr/sbin/apache2 -k start
             ├─3044 /usr/sbin/apache2 -k start
             └─3045 /usr/sbin/apache2 -k start

дек 15 20:16:35 korolvd-VirtualBox systemd[1]: Starting The Apache HTTP Server...
дек 15 20:16:35 korolvd-VirtualBox apachectl[3041]: AH00558: apache2: Could not reliably>
дек 15 20:16:35 korolvd-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

- *Проверка работоспособности через браузер*
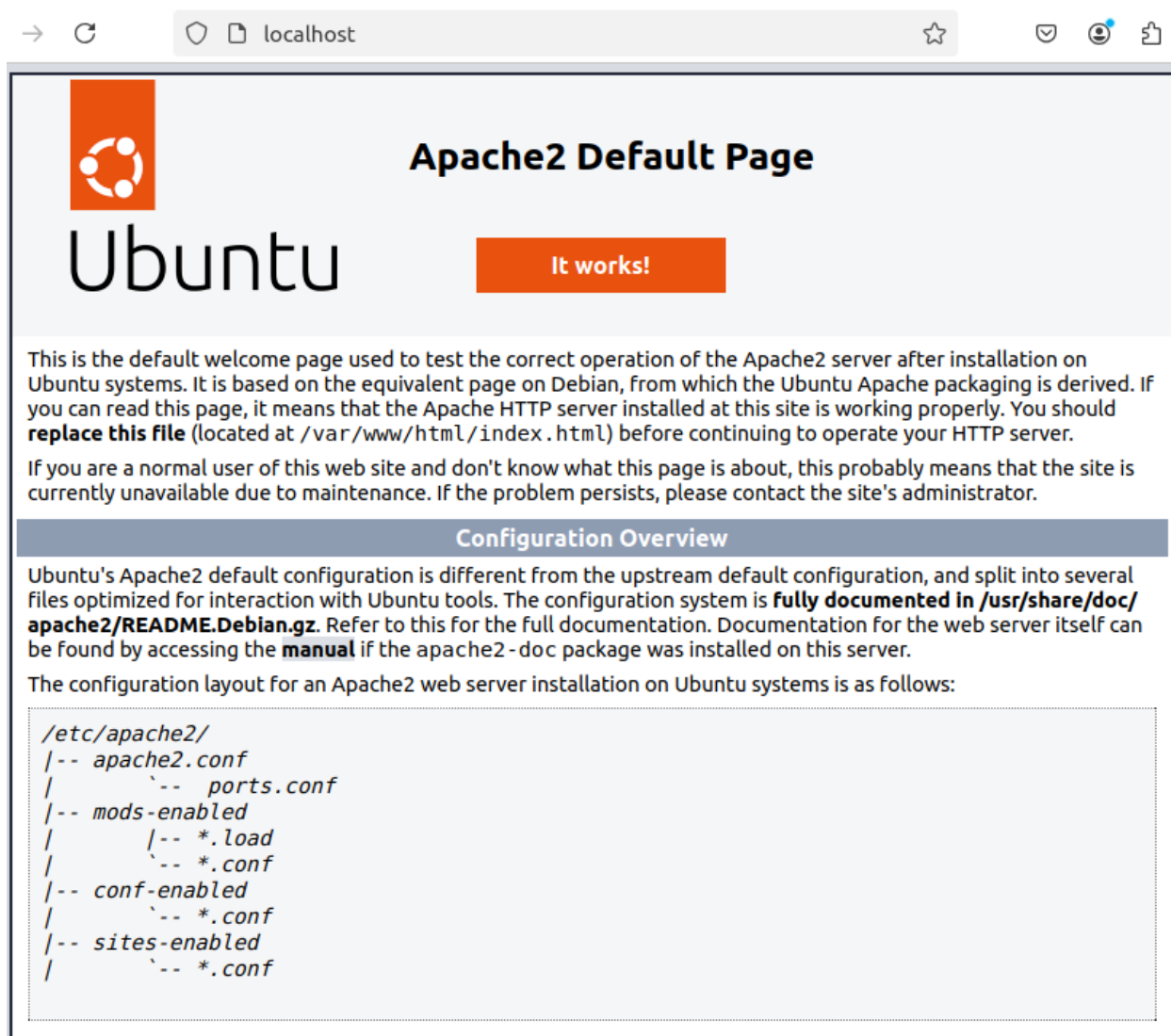
2. Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Apache.

- *Генерация сертификата*

- *Конфигурирование*

```
  GNU nano 6.2                    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

        RewriteEngine On
        RewriteCond %{HTTPS} !=on
        RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
</VirtualHost>
<VirtualHost _default_:443>
                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

                #    SSL Engine Switch:
                #    Enable/Disable SSL for this virtual host.
                SSLEngine on

                #    A self-signed (snakeoil) certificate can be created by installing
                #    the ssl-cert package. See
                #    /usr/share/doc/apache2/README.Debian.gz for more info.
                #    If both key and certificate are stored in the same file, only the
                #    SSLCertificateFile directive is needed.
                SSLCertificateFile      /etc/apache2/certs/apache.crt
                SSLCertificateKeyFile /etc/apache2/certs/apache.key

                #    Server Certificate Chain:
                #    Point SSLCertificateChainFile at a file containing the
```
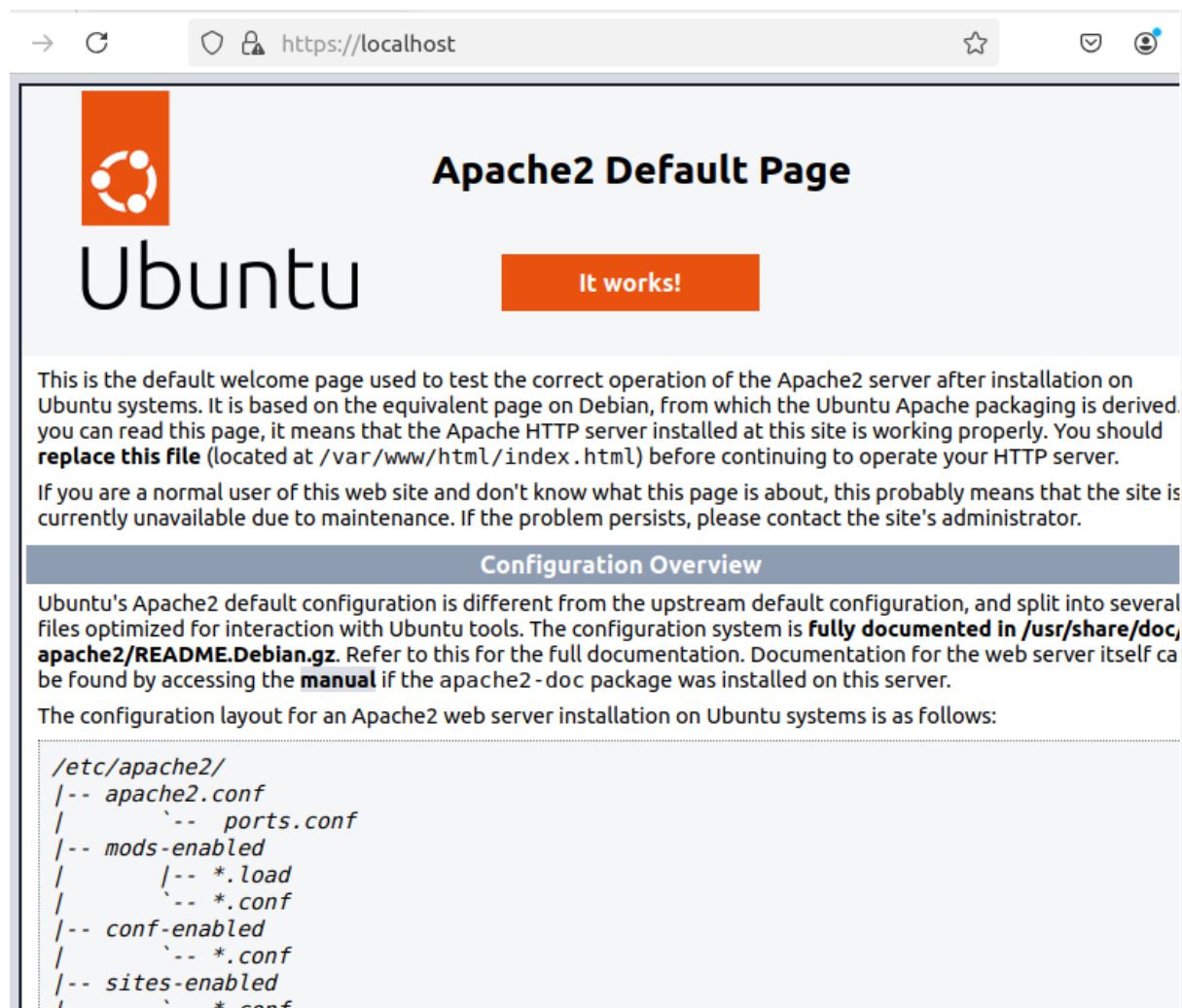
- Проверка работы SSL (https)

Вторая часть

1. **Установить Nginx и убедиться, что он работает**

   - установка nginx

```
korolvd@korolvd-VirtualBox:/etc/apache2$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 nginx-common
  nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 nginx nginx-common
  nginx-core
0 upgraded, 9 newly installed, 0 to remove and 40 not upgraded.
Need to get 697 kB of archives.
After this operation, 2 395 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx-common all 1.18.
0-6ubuntu14.5 [40,1 kB]
```

- *Проверка работоспособности*

```
korolvd@korolvd-VirtualBox:/etc/apache2$ sudo systemctl status nginx
● 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 Sun 2024-12-15 21:42:52 +05; 2min 31s ago
       Docs: man:nginx(8)
    Process: 6043 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (c>
    Process: 6044 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exite>
   Main PID: 6133 (nginx)
      Tasks: 2 (limit: 4608)
     Memory: 3.1M
        CPU: 20ms
     CGroup: /system.slice/nginx.service
             ├─6133 "nginx: master process /usr/sbin/nginx -g daemon on; master_process >
             └─6136 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" ""  "">

дек 15 21:42:52 korolvd-VirtualBox systemd[1]: Starting A high performance web server an>
дек 15 21:42:52 korolvd-VirtualBox systemd[1]: Started A high performance web server and>
lines 1-16/16 (END)
```

○ ▢ 127.0.0.1   ☆

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

2. **Настройте прямой и обратный прокси в Nginx для перенаправления запросов**
   - *Конфигурация прямого прокси*

```
  GNU nano 6.2                          sites-available/default *
server {
        listen 8081;
        resolver 8.8.8.8;

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

- *Проверка прямого прокси*

```
korolvd@korolvd-VirtualBox:/etc/nginx$ curl -v -L --proxy http://localhost:8081 http://httpforever.com
*   Trying 127.0.0.1:8081...
* Connected to (nil) (127.0.0.1) port 8081 (#0)
> GET http://httpforever.com/ HTTP/1.1
> Host: httpforever.com
> User-Agent: curl/7.81.0
> Accept: */*
> Proxy-Connection: Keep-Alive
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx/1.18.0 (Ubuntu)
< Date: Sun, 15 Dec 2024 19:06:20 GMT
< Content-Type: text/html
< Content-Length: 5124
< Connection: keep-alive
< Last-Modified: Wed, 22 Mar 2023 14:54:48 GMT
< ETag: "641b16b8-1404"
< Referrer-Policy: strict-origin-when-cross-origin
< X-Content-Type-Options: nosniff
< Feature-Policy: accelerometer 'none'; camera 'none'; geolocation 'none'; gyroscope 'none'; magnetomete
r 'none'; microphone 'none'; payment 'none'; usb 'none'
< Content-Security-Policy: default-src 'self'; script-src cdnjs.cloudflare.com 'self'; style-src cdnjs.c
loudflare.com 'self' fonts.googleapis.com 'unsafe-inline'; font-src fonts.googleapis.com fonts.gstatic.c
om cdnjs.cloudflare.com; frame-ancestors 'none'; report-uri https://scotthelme.report-uri.com/r/d/csp/en
force
< Accept-Ranges: bytes
<
<!DOCTYPE HTML>
<html>
        <head>
                <title>HTTP Forever</title>
                <meta http-equiv="content-type" content="text/html; charset=utf-8" />
                <meta name="description" content="A site that will always be available over HTTP!" />
                <meta name="keywords" content="HTTP WiFi Captive Portal" />
                <!--[if lte IE 8]><script src="css/ie/html5shiv.js"></script><![endif]-->
                <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js" integrit
y="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8=" crossorigin="anonymous"></script>
                <script src="https://cdnjs.cloudflare.com/ajax/libs/skel/3.0.1/skel.min.js" integrity="s
```

- *Конфигурация обратного прокси*

```
server {
        listen 8080;
        server_name localhost;

        location / {
                proxy_pass http://localhost:8082;
        }
}
```

- *Проверка обратного прокси (запуск сервиса на порту 8082 и отправка запроса на порт 8080)*

```
korolvd@korolvd-VirtualBox:~$ python3 -m http.server 8082
Serving HTTP on 0.0.0.0 port 8082 (http://0.0.0.0:8082/) ...
127.0.0.1 - - [16/Dec/2024 00:28:08] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [16/Dec/2024 00:28:08] code 404, message File not found
127.0.0.1 - - [16/Dec/2024 00:28:08] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [16/Dec/2024 00:28:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [16/Dec/2024 00:38:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [16/Dec/2024 00:38:58] "GET /Documents/ HTTP/1.1" 200 -
127.0.0.1 - - [16/Dec/2024 00:40:33] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [16/Dec/2024 00:41:00] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [16/Dec/2024 00:46:25] "GET / HTTP/1.0" 200 -
```

```
korolvd@korolvd-VirtualBox:/etc/nginx$ curl -v -L localhost:8080
*   Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET / HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx/1.18.0 (Ubuntu)
< Date: Sun, 15 Dec 2024 19:46:25 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 1326
< Connection: keep-alive
<
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a></li>
<li><a href=".bash_logout">.bash_logout</a></li>
<li><a href=".bashrc">.bashrc</a></li>
<li><a href=".cache/">.cache/</a></li>
```

3. **Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Nginx.**

- генерация сертификата

```
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]:korolvd
Organizational Unit Name (eg, section) []:vitaly
Common Name (e.g. server FQDN or YOUR name) []:korolvd
Email Address []:korol.vd@mail.ru
korolvd@korolvd-VirtualBox:/etc/nginx$ cd certs/
korolvd@korolvd-VirtualBox:/etc/nginx/certs$ ls
nginx.cert   nginx.key
korolvd@korolvd-VirtualBox:/etc/nginx/certs$
```

- *Конфигурирование*

```
  GNU nano 6.2                     sites-available/default
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        return 301 https://localhost/;
}

server {
        listen 443;
        server_name localhost;

        ssl on;

        ssl_certificate          /etc/nginx/certs/nginx.cert;
        ssl_certificate_key      /etc/nginx/certs/nginx.key;
        ssl_ciphers 'TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_>
        ssl_prefer_server_ciphers off;

        location / {
                root /var/www/html;
                index index.html index.htm index.nginx-debian.html;
        }
```

- *Проверка ssl (https)*

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.