

Front-end Nginx и Back-end Apache2

Задание:

Настроить схему front-end Nginx и Back-end Apache2

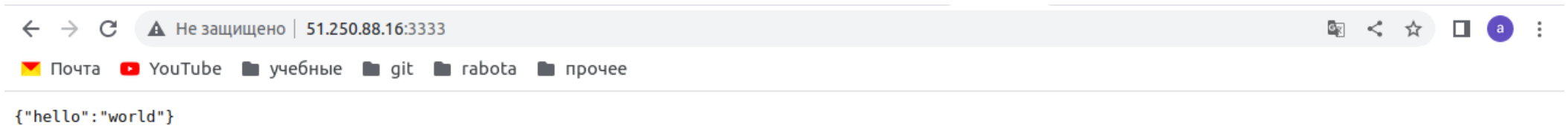
Примечание:

Все будет запущенно в docker и описано в docker-compose.yml.

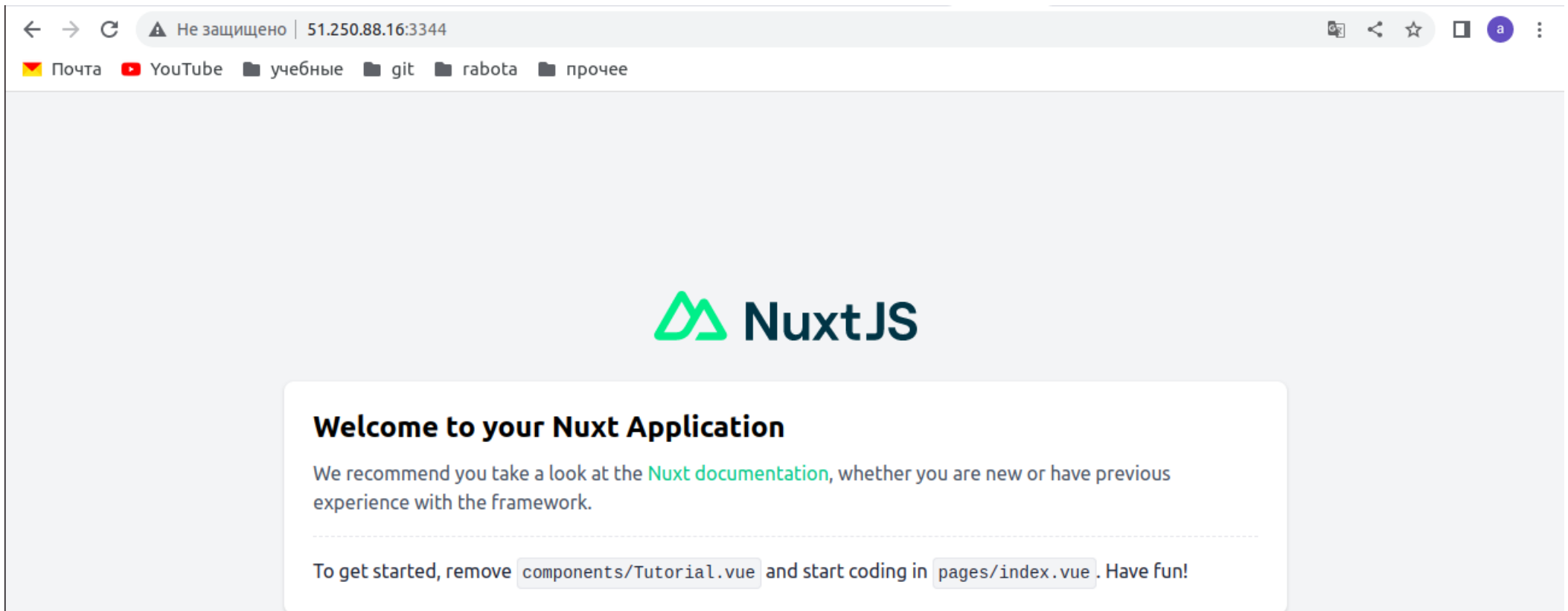
Что бы было видно балансировку нагрузки для примера запущенно 2 приложения (в боевой среде это должны быть абсолютно одинаковые сервера):

- 51.250.88.16:3333 adonis js
- 51.250.88.16:3344 nuxt

Приложение на 51.250.88.16:3333



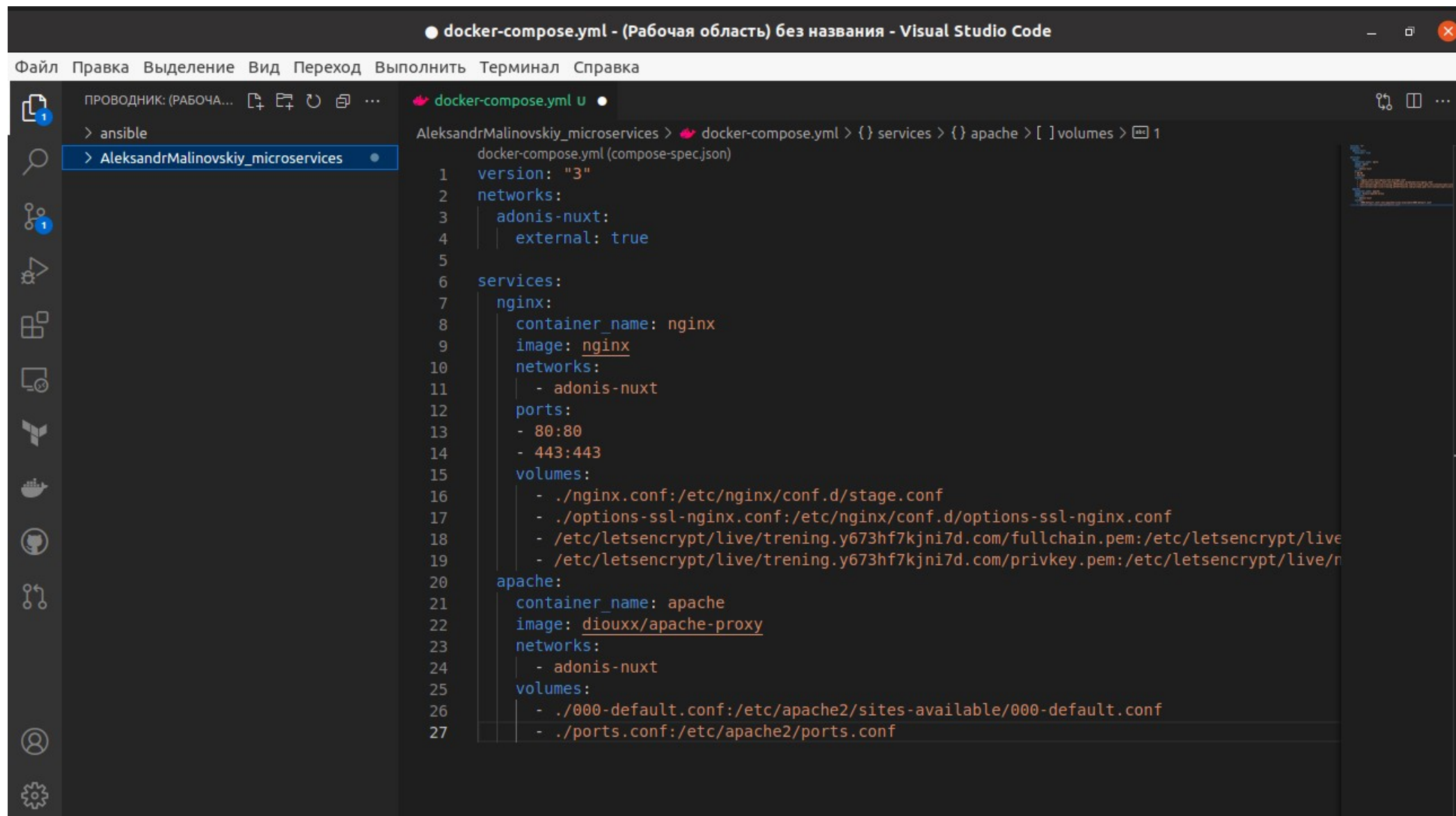
Приложение на 51.250.88.16:3344



Docker-compose файл.

Описано запуск контейнеров с nginx и apache.

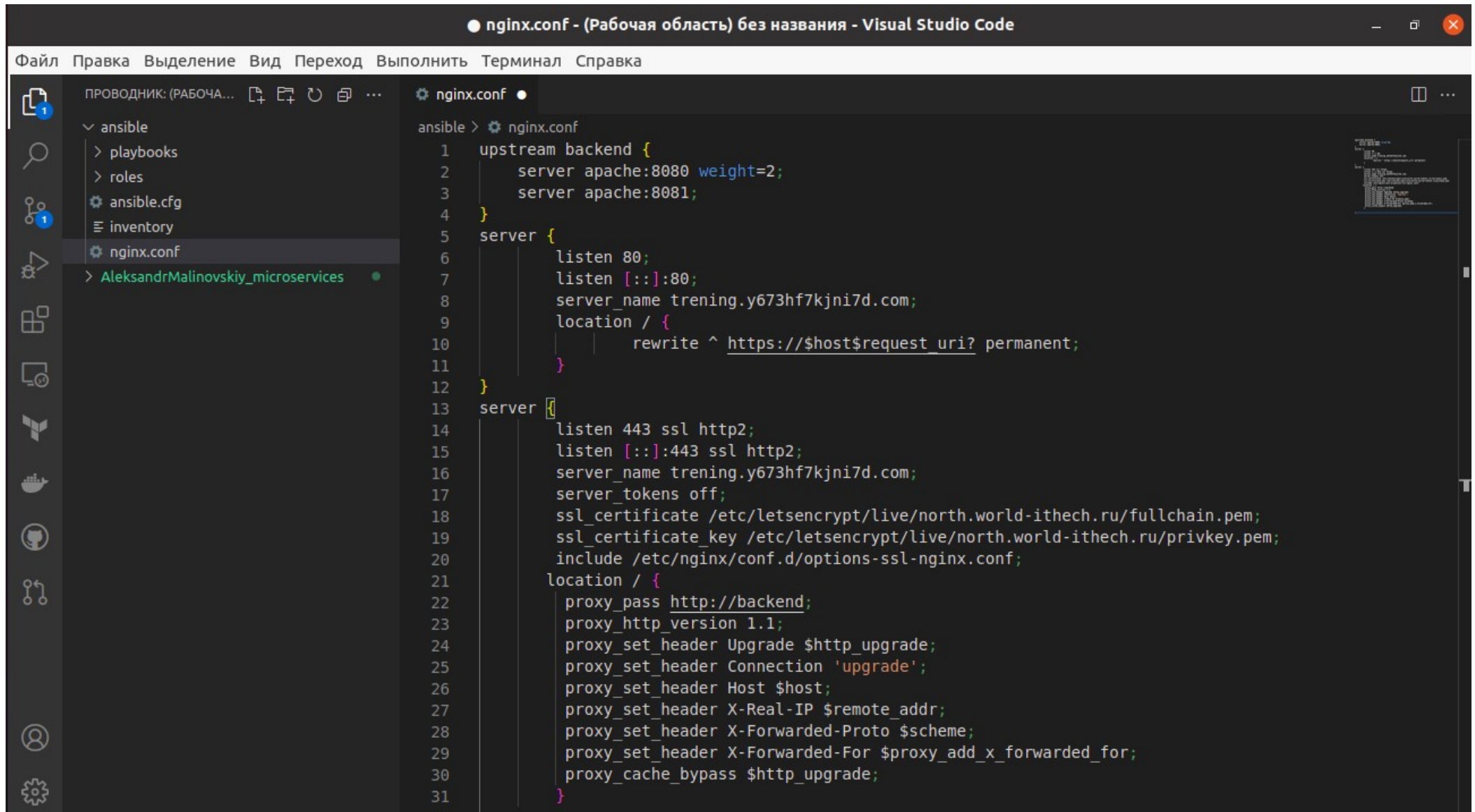
Образ apache уже включает в себя нужные модули для проксирования.

The image is a screenshot of the Visual Studio Code editor. The title bar at the top reads "docker-compose.yml - (Рабочая область) без названия - Visual Studio Code". The menu bar includes "Файл", "Правка", "Выделение", "Вид", "Переход", "Выполнить", "Терминал", and "Справка". The left sidebar shows a file explorer with a tree view containing "ansible" and "AleksandrMalinovskiy_microservices". The main editor area displays the content of "docker-compose.yml" with the following YAML configuration:

```
1 version: "3"
2 networks:
3   adonis-nuxt:
4     external: true
5
6 services:
7   nginx:
8     container_name: nginx
9     image: nginx
10    networks:
11      - adonis-nuxt
12    ports:
13      - 80:80
14      - 443:443
15    volumes:
16      - ./nginx.conf:/etc/nginx/conf.d/stage.conf
17      - ./options-ssl-nginx.conf:/etc/nginx/conf.d/options-ssl-nginx.conf
18      - /etc/letsencrypt/live/trening.y673hf7kjni7d.com/fullchain.pem:/etc/letsencrypt/live/trening.y673hf7kjni7d.com/fullchain.pem
19      - /etc/letsencrypt/live/trening.y673hf7kjni7d.com/privkey.pem:/etc/letsencrypt/live/trening.y673hf7kjni7d.com/privkey.pem
20   apache:
21     container_name: apache
22     image: diouxx/apache-proxy
23     networks:
24       - adonis-nuxt
25     volumes:
26       - ./000-default.conf:/etc/apache2/sites-available/000-default.conf
27       - ./ports.conf:/etc/apache2/ports.conf
```

Конфигурационный файл nginx.

Описано подключение ключа ssl и применение секции upstream.



The screenshot shows the Visual Studio Code editor with the file `nginx.conf` open. The left sidebar displays the file explorer with the following structure:

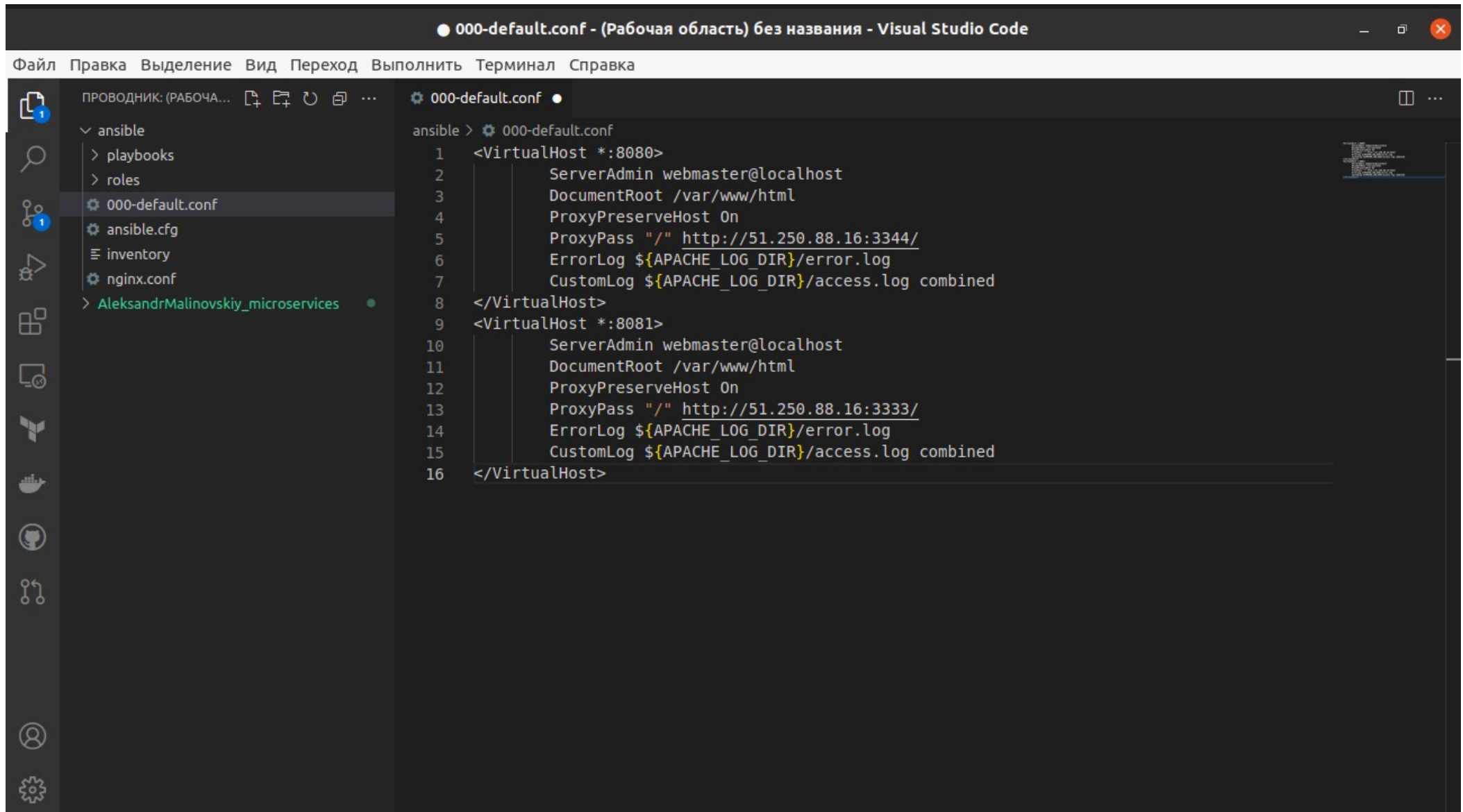
- ansible
 - playbooks
 - roles
 - ansible.cfg
 - inventory
 - nginx.conf
- AleksandrMalinovskiy_microservices

The main editor area shows the content of `nginx.conf` with line numbers 1 through 31. The configuration includes an upstream block for backend servers and two server blocks: one for HTTP on port 80 and another for HTTPS on port 443.

```
1 upstream backend {
2     server apache:8080 weight=2;
3     server apache:8081;
4 }
5 server {
6     listen 80;
7     listen [::]:80;
8     server_name trening.y673hf7kjni7d.com;
9     location / {
10         rewrite ^ https://$host$request_uri? permanent;
11     }
12 }
13 server {
14     listen 443 ssl http2;
15     listen [::]:443 ssl http2;
16     server_name trening.y673hf7kjni7d.com;
17     server_tokens off;
18     ssl_certificate /etc/letsencrypt/live/north.world-ithech.ru/fullchain.pem;
19     ssl_certificate_key /etc/letsencrypt/live/north.world-ithech.ru/privkey.pem;
20     include /etc/nginx/conf.d/options-ssl-nginx.conf;
21     location / {
22         proxy_pass http://backend;
23         proxy_http_version 1.1;
24         proxy_set_header Upgrade $http_upgrade;
25         proxy_set_header Connection 'upgrade';
26         proxy_set_header Host $host;
27         proxy_set_header X-Real-IP $remote_addr;
28         proxy_set_header X-Forwarded-Proto $scheme;
29         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
30         proxy_cache_bypass $http_upgrade;
31     }
32 }
```

Конфигурационный файл для apache.

Описано проксирование с порта 8080 и 8081 на ip 51.250.88.16 порт 3344 и порт 3333.



```
000-default.conf - (Рабочая область) без названия - Visual Studio Code

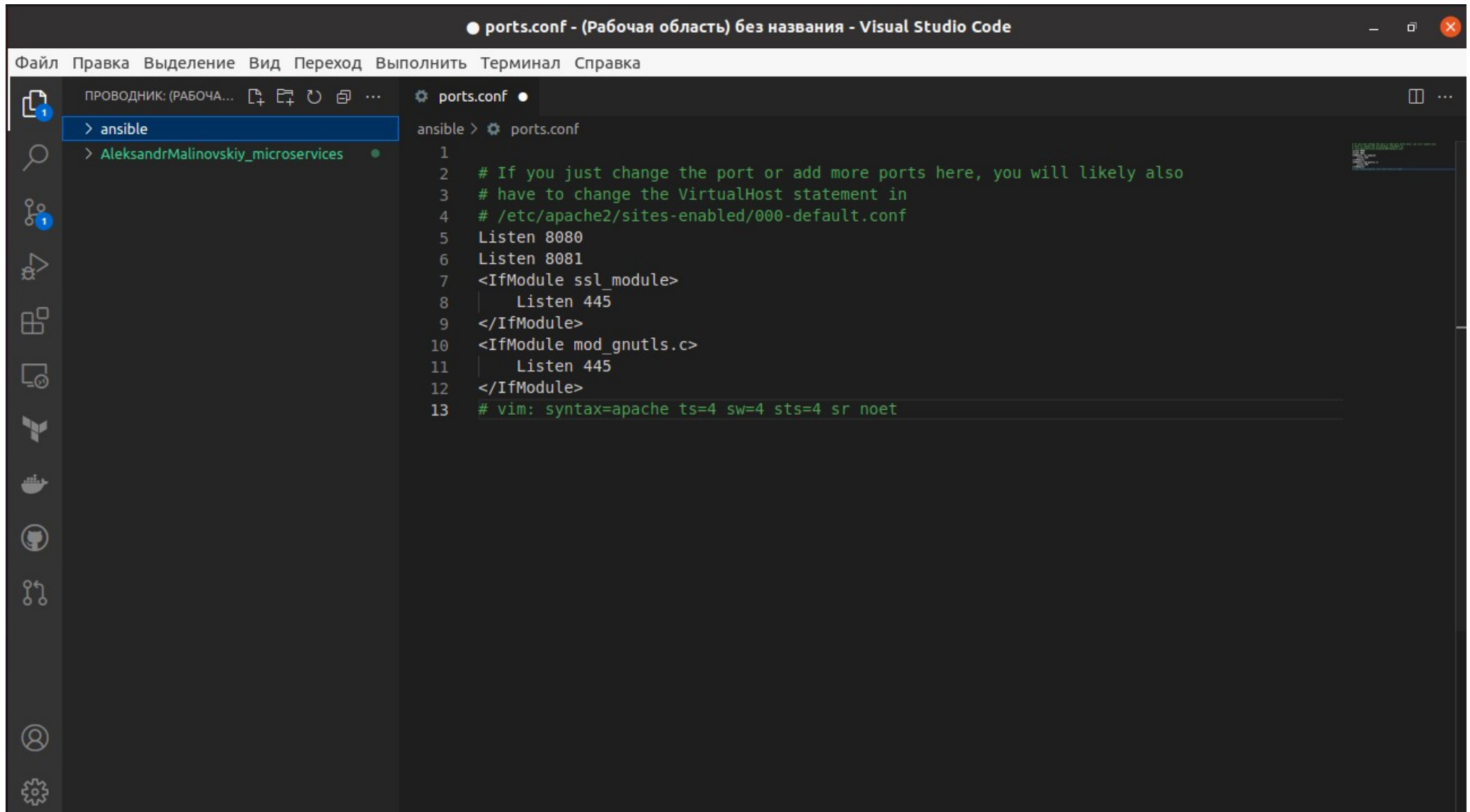
Файл  Правка  Выделение  Вид  Переход  Выполнить  Терминал  Справка

PROVODNIK: (РАБОЧА...  000-default.conf

ansible
├── playbooks
├── roles
├── 000-default.conf
├── ansible.cfg
├── inventory
├── nginx.conf
└── AleksandrMalinovskiy_microservices

ansible > 000-default.conf
1  <VirtualHost *:8080>
2      ServerAdmin webmaster@localhost
3      DocumentRoot /var/www/html
4      ProxyPreserveHost On
5      ProxyPass "/" http://51.250.88.16:3344/
6      ErrorLog ${APACHE_LOG_DIR}/error.log
7      CustomLog ${APACHE_LOG_DIR}/access.log combined
8  </VirtualHost>
9  <VirtualHost *:8081>
10     ServerAdmin webmaster@localhost
11     DocumentRoot /var/www/html
12     ProxyPreserveHost On
13     ProxyPass "/" http://51.250.88.16:3333/
14     ErrorLog ${APACHE_LOG_DIR}/error.log
15     CustomLog ${APACHE_LOG_DIR}/access.log combined
16 </VirtualHost>
```

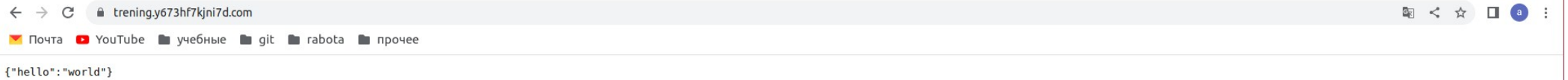
Конфигурационный файл для apache с указанием открытия портов 8080 и 8081.



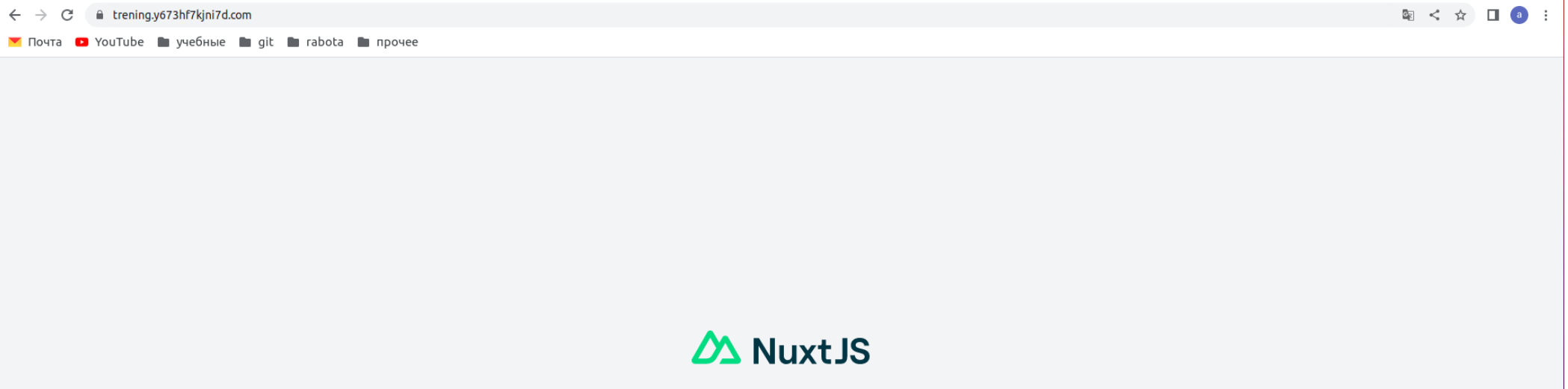
The screenshot shows the Visual Studio Code interface with the file `ports.conf` open. The left sidebar displays the file explorer with the project structure: `ansible` and `AleksandrMalinovskiy_microservices`. The main editor area shows the content of `ports.conf`, which configures Apache to listen on ports 8080 and 8081. The configuration includes conditional blocks for `ssl_module` and `mod_gnutls.c`, both listening on port 445. A vim syntax file is also specified at the bottom.

```
1
2 # If you just change the port or add more ports here, you will likely also
3 # have to change the VirtualHost statement in
4 # /etc/apache2/sites-enabled/000-default.conf
5 Listen 8080
6 Listen 8081
7 <IfModule ssl_module>
8     Listen 445
9 </IfModule>
10 <IfModule mod_gnutls.c>
11     Listen 445
12 </IfModule>
13 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Переходим по домену и видим стартовую страницу adonis.



Обновляем страницу и видим стартовую страницу nuxt, что означает что балансировка работает.



Так же для теста дергаем с помощью Curl домен. Видим первый раз ответ adonis второй раз ответ nuxt.

```
alex@apache: ~  
root@apache:~# curl https://trening.y673hf7kjni7d.com/  
{"hello":"world"}root@apache:~#  
root@apache:~#  
root@apache:~#  
root@apache:~#  
root@apache:~#  
root@apache:~#  
root@apache:~#  
root@apache:~# curl https://trening.y673hf7kjni7d.com/  
<!doctype html>  
<html data-n-head-ssr lang="en" data-n-head="%7B%22lang%22:%7B%22ssr%22:%22en%22%7D%7D">  
  <head >  
    <title>nuxt</title>  
<meta data-n-head="ssr" charset="utf-8">  
<meta data-n-head="ssr" name="viewport" content="width=device-width, initial-scale=1">  
<meta data-n-head="ssr" data-hid="description" name="description" content="">  
<meta data-n-head="ssr" name="format-detection" content="telephone=no">  
<link data-n-head="ssr" rel="icon" type="image/x-icon" href="/favicon.ico">  
<link rel="preload" href="/_nuxt/runtime.js" as="script"><link rel="preload" href="/_nuxt/commons/app.js" as="script"><link rel="preload" href="/_nuxt/vendors/app.js" as="script"><link rel="preload" href="/_nuxt/app.js" as="script"><link rel="preload" href="/_nuxt/pages/index.js" as="script"><style data-vue-ssr-id="6c15cca0:0">  
.nuxt-progress {  
  position: fixed;  
  top: 0px;  
  left: 0px;  
  right: 0px;  
  height: 2px;  
  width: 0%;  
  opacity: 1;  
  transition: width 0.1s, opacity 0.4s;  
  background-color: black;  
  z-index: 999999;  
}  
.nuxt-progress.nuxt-progress-notransition {  
  transition: none;
```