

1. Для установки пакетов необходимо переключиться в терминале на суперпользователя

```
su -  
root
```

2. Необходимо отредактировать файл sudoers

```
visudo /etc/sudoers
```

Снять комментирование следующих строк

```
User_Alias WHEEL_USERS = %wheel  
User_Alias XGRP_USERS = %xgrp  
WHEEL_USERS ALL=(ALL:ALL) ALL
```

Выходим из редактора sudoers с сохранением изменений

```
:wq
```

3. Установка необходимых пакетов ПО из репозиториев. Дальнейшие действия выполняются пользователями входящим в группу wheel.

```
sudo apt-get update  
sudo apt-get dist-upgrade  
sudo apt-get install unzip curl postgresql14-server postgresql14-contrib nginx nano  
mc
```

4. Перед запуском службы PostgreSQL необходимо создать системные базы данных

```
sudo /etc/init.d/postgresql initdb
```

5. Запускаем службу PostgreSQL и включаем ее по умолчанию

```
sudo systemctl start postgresql  
sudo systemctl enable postgresql
```

6. Редактируем конфигурационные файлы PostgreSQL

Проверяем версию установленной PostgreSQL

```
postgres --version
```

Открываем редактором файл postgresql.conf

```
sudo nano /var/lib/pgsql/data/postgresql.conf
```

Вносим в файл изменения в секции # - Connection Settings - значение параметра listen\_addresses с localhost на \*

должно получиться:

```
listen_addresses = '*'
```

Открываем редактором файл pg\_hba.conf

```
sudo nano /var/lib/pgsql/data/pg_hba.conf
```

Вносим в файл изменения в секции

# IPv4 Local connections: значение параметра ADDRESS с 127.0.0.1/32 на 0.0.0.0/0 должно получиться

```
# IPv4 local connections:  
host all all 0.0.0.0/0 scram-sha-256
```

#### 7. Перезапускаем службу

```
sudo systemctl restart postgresql
```

#### 8. Проверяем, что выполняется прослушивание на порту 5432 для всех активных адресов

```
sudo netstat -ltnp
```

```
tcp 0 0 0.0.0.0:5432 0.0.0.0:* LISTEN 14569/postgres
```

#### 9. Добавляем пользователя postgres в группу владельца директории, из которой выполняются действия:

Проверяем владельца текущей папки (обозначена «.»)

```
ls -la
```

Проверяем текущий рабочий каталог

```
pwd
```

Пример каталога:

```
/home/vitro
```

Добавляем пользователя postgres в группу владельца

```
sudo usermod -aG vitro postgres
```

#### 10. Войти в cli postgresql от имени пользователя postgres

```
sudo psql -U postgres
```

#### 11. Создать пользователя с ролью суперпользователя

```
postgres-#CREATE USER vitrodbuser WITH PASSWORD 'vitrodbpassword' SUPERUSER;
```

12. Создать новую базу данных с именем vitrodb

```
postgres-#CREATE DATABASE vitrodb with owner = vitrodbuser encoding = 'UTF8'  
connection limit = -1;
```

13. Отключиться от cli postgresql

```
postgres-# \q
```

14. Скачать и распаковать дистрибутив

```
curl "https://linkto/Vitro.Server.MultiPlatform.Linux-24.1.20.zip" --output  
'Core.Web.Linux.zip'  
unzip Core.Web.Linux.zip
```

15. Скопировать в доступное для пользователей расположение (к примеру tmp) файл резервной копии базы данных

```
cp ~/Vitro.Server.MultiPlatform.Linux-24.1.20/DB/vitrodb.sql.bak /tmp
```

16. Запустить восстановление БД из бэкапа

```
sudo -u postgres pg_restore -v -U vitrodbuser -h localhost -d vitrodb  
/tmp/vitrodb.sql.bak
```

17. Скопировать папку файлового хранилища в расположение, в которое смонтирован раздел для хранения

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-24.1.20/DB/VitroFileStorage  
/mnt/storage
```

18. Создаем пользователя в группе www-data

```
sudo useradd -U www-data  
groups www-data
```

19. Сменить владельца для папки файлового хранилища

```
sudo chown -R www-data:www-data /mnt/storage/VitroFileStorage
```

20. Скопировать папку приложения в желаемое расположение

```
sudo cp -r ~/Vitro.Server.MultiPlatform.Linux-24.1.20/Vitro.Server.Core.Web  
/usr/local/
```

21. Сменить владельца для папки приложения

```
sudo chown -R www-data:www-data /usr/local/Vitro.Server.Core.Web
```

22. Добавляем пользователя www-data в группу \_nginx

```
sudo usermod -aG _nginx www-data
```

23. Установить атрибут "Исполняемый" для файла Vitro.Server.Core.Web

```
sudo chmod +x /usr/local/Vitro.Server.Core.Web/Vitro.Server.Core.Web
```

24. Внести изменения в конфигурационный файл db.json

```
sudo nano /usr/local/Vitro.Server.Core.Web/Vitro/Server/Conf/db.json
```

изменить строку соединения, указав в качестве значения для атрибута "server" имя или IP сервера БД, а в качестве значения для атрибута "database" название созданной в п.13 базы данных:

```
"ConnectionString": "server=localhost;database=vitrodb;user  
id=vitrodbuser;password=vitrodbpassword;Include Error Detail=true",
```

изменить параметр "FileStoragePath", указав корректный путь к папке файлового хранилища:

```
"FileStoragePath": "/mnt/storage/VitroFileStorage",  
"TempPath": "/tmp",
```

25. Создать и отредактировать файл модуля systemd для запуска службы

```
sudo nano /lib/systemd/system/vitro-server.service
```

содержимое файла модуля:

```
[Unit]  
Description=Vitro .NET Web Application  
  
[Service]  
WorkingDirectory=/usr/local/Vitro.Server.Core.Web  
ExecStart=/usr/local/Vitro.Server.Core.Web/Vitro.Server.Core.Web --urls  
http://localhost:4615  
Restart=always  
# Restart service after 10 seconds if the dotnet service crashes:  
RestartSec=10  
KillSignal=SIGINT  
SyslogIdentifier=dotnet-example  
User=www-data  
SyslogIdentifier=Vitro-server  
  
[Install]  
WantedBy=multi-user.target
```

## 26. Применить изменения и запустить службу

```
sudo systemctl enable vitro-server
sudo systemctl start vitro-server
```

## 27. Проверить выполняется ли прослушивание на порту, заданном в файле модуля (здесь 4615) для адреса Lo интерфейса

```
netstat -ltnp
...
tcp        0      0 127.0.0.1:4615      0.0.0.0:*           LISTEN -
...
```

## 28. Настроить обратное проксирование при помощи nginx, отредактировав конфигурационный файл

```
sudo nano /etc/nginx/sites-available/default.conf
```

Итоговое содержимое конфигурационного файла:

```
# Default server configuration
#

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # Only allow access if the host is correct

    server_name *.*. *.*; #указывается ip-адрес или домен площадки
    без http://

    client_max_body_size 100000M;

    proxy_buffer_size 1M;
    proxy_buffers 4 1M;

    location / {
        #proxying traffic to upstream application (Vitro.Server.Core.Web)
        proxy_pass http://127.0.0.1:4615;
        proxy_buffering on;
    }
}
```

```
}
```

29. Проверить наличие символической ссылки по пути `/etc/nginx/sites-enabled.d/` и при необходимости сформировать ее

```
sudo ln -s /etc/nginx/sites-available.d/default.conf /etc/nginx/sites-enabled.d/
```

30. Настроить конфигурационный файл NGINX

```
sudo nano /etc/nginx/nginx.conf
```

Итоговое содержимое конфигурационного файла:

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
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;
    ##
    # SSL Settings
    ##
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;
    ##
    # Logging Settings
    ##
    access_log /var/log/nginx/access.log;
```

```
error_log /var/log/nginx/error.log;
##
# Gzip Settings
##
gzip on;
    proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=STATIC:10m
    inactive=24h max_size=1g;
# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml
application/xml application/xml+rss text/javascript;
##
# Virtual Host Configs
##
include /etc/nginx/conf-enabled/*.conf;
include /etc/nginx/sites-enabled/*.conf;
}
```

### 31. Выполнить перезагрузку службы nginx

```
sudo systemctl reload nginx
```

### 32. Дополнительно проверить статусы запущенных служб и при необходимости запустить

```
sudo systemctl status nginx
sudo systemctl status postgresql
sudo systemctl status vitro-server
```

### 33. Проверить статус службы Apache

```
sudo systemctl status httpd2
```

Если служба запущена, то останавливаем ее

```
sudo systemctl stop httpd2
```