

# Руководство по развертыванию Django-приложения на Ubuntu 24.04

---



## Содержание

---

1. [Подготовка сервера](#)
  2. [Установка зависимостей](#)
  3. [Настройка PostgreSQL](#)
  4. [Развертывание Django-приложения](#)
  5. [Настройка Gunicorn](#)
  6. [Настройка Nginx](#)
  7. [Настройка домена и SSL](#)
  8. [Безопасность](#)
  9. [Мониторинг и логи](#)
  10. [Резервное копирование](#)
- 



## Подготовка сервера

---

### 1.1. Обновление системы

```
# Обновление пакетов
sudo apt update
sudo apt upgrade -y

# Установка базовых утилит
sudo apt install -y curl wget git vim htop ufw
```

### 1.2. Настройка времени

```
# Установка временной зоны
sudo timedatectl set-timezone Asia/Almaty

# Проверка времени
timedatectl status
```

## 1.3. Создание пользователя для приложения (опционально)

```
# Создание нового пользователя
sudo adduser django
sudo usermod -aG sudo django

# Переключение на нового пользователя
su - django
```

## Установка зависимостей

### 2.1. Python и системные зависимости

```
# Установка Python 3.11/3.12
sudo apt install -y python3 python3-pip python3-venv python3-dev

# Установка системных зависимостей для PostgreSQL и других компонентов
sudo apt install -y libpq-dev build-essential libssl-dev libffi-dev
```

### 2.2. Node.js (если нужно)

```
# Установка Node.js 18.x
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs
```

## Настройка PostgreSQL

### 3.1. Установка PostgreSQL

```
# Установка PostgreSQL 16
sudo apt install -y postgresql postgresql-contrib

# Проверка статуса
sudo systemctl status postgresql
```

### 3.2. Настройка базы данных

```
# Вход в PostgreSQL
sudo -u postgres psql
```

```
-- Создание базы данных
CREATE DATABASE n3connect;

-- Создание пользователя
CREATE USER n3_user WITH PASSWORD 'StrongPassword123!';
```

```
-- Назначение прав
GRANT ALL PRIVILEGES ON DATABASE n3connect TO n3_user;

-- Расширенные права (для Django)
ALTER DATABASE n3connect OWNER TO n3_user;

-- Выход
\q
```

### 3.3. Настройка аутентификации

```
# Редактирование конфигурации аутентификации
sudo nano /etc/postgresql/16/main/pg_hba.conf
```

Добавить или изменить:

```
# Локальные подключения через Unix socket
local all n3_user md5
local all postgres peer

# Локальные TCP подключения
host all n3_user 127.0.0.1/32 md5
```

```
# Перезапуск PostgreSQL
sudo systemctl restart postgresql
```



## Развертывание Django-приложения

### 4.1. Клонирование проекта

```
# Создание директории для проекта
mkdir -p ~/projects
cd ~/projects

# Клонирование репозитория (замените на ваш репозиторий)
git clone https://github.com/ваш-пользователь/ваш-проект.git
cd ваш-проект
```

## 4.2. Настройка виртуального окружения

```
# Создание виртуального окружения
python3 -m venv venv

# Активация виртуального окружения
source venv/bin/activate

# Установка зависимостей
pip install --upgrade pip
pip install -r requirements.txt

# Если нет requirements.txt, установите основные зависимости
pip install django gunicorn psycpg2-binary python-dotenv
```

## 4.3. Настройка переменных окружения

```
# Создание файла .env
nano .env
```

```
# Django
SECRET_KEY=ваш-секретный-ключ
DEBUG=False
ALLOWED_HOSTS=ваш-домен.ru, localhost, 127.0.0.1

# Database
DB_NAME=n3connect
DB_USER=n3_user
DB_PASSWORD=StrongPassword123!
DB_HOST=localhost
DB_PORT=5432

# Дополнительные настройки
TIME_ZONE=Asia/Almaty
LANGUAGE_CODE=ru-ru
```

## 4.4. Настройка Django settings.py

```
# Добавьте в начало settings.py
import os
from pathlib import Path
from dotenv import load_dotenv

# Загрузка переменных окружения
load_dotenv()

# Базовые настройки
BASE_DIR = Path(__file__).resolve().parent.parent
SECRET_KEY = os.environ.get('SECRET_KEY')
DEBUG = os.environ.get('DEBUG', 'False') == 'True'
```

```

ALLOWED_HOSTS = os.environ.get('ALLOWED_HOSTS', '').split(',')

# База данных
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': os.environ.get('DB_NAME'),
        'USER': os.environ.get('DB_USER'),
        'PASSWORD': os.environ.get('DB_PASSWORD'),
        'HOST': os.environ.get('DB_HOST', 'localhost'),
        'PORT': os.environ.get('DB_PORT', '5432'),
    }
}

# Статические файлы
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static')]

# Медиа файлы
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

# Безопасность для работы за прокси
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')
USE_X_FORWARDED_HOST = True
USE_X_FORWARDED_PORT = True

```

## 4.5. Применение миграций и создание суперпользователя

```

# Применение миграций
python manage.py migrate

# Сбор статических файлов
python manage.py collectstatic --noinput

# Создание суперпользователя
python manage.py createsuperuser

```

## Настройка Gunicorn

### 5.1. Создание systemd службы

```
sudo nano /etc/systemd/system/gunicorn.service
```

```

[Unit]
Description=gunicorn daemon for Django
After=network.target postgresql.service

```

```
[Service]
Type=simple
User=django
Group=www-data
WorkingDirectory=/home/django/projects/ваш-проект
Environment="PATH=/home/django/projects/ваш-проект/venv/bin"
Environment="DJANGO_SETTINGS_MODULE=ваш_проект.settings"
ExecStart=/home/django/projects/ваш-проект/venv/bin/gunicorn \
    --workers 3 \
    --bind 127.0.0.1:8000 \
    --access-logfile - \
    --error-logfile - \
    ваш_проект.wsgi:application
Restart=always
RestartSec=3

# Security
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

## 5.2. Запуск и настройка Gunicorn

```
# Перезагрузка systemd
sudo systemctl daemon-reload

# Запуск Gunicorn
sudo systemctl start gunicorn

# Включение автозагрузки
sudo systemctl enable gunicorn

# Проверка статуса
sudo systemctl status gunicorn

# Просмотр логов
sudo journalctl -u gunicorn -f
```

---

## Настройка Nginx

### 6.1. Установка Nginx

```
sudo apt install -y nginx
```

## 6.2. Создание конфигурации сайта

```
sudo nano /etc/nginx/sites-available/ваш-сайт
```

```
server {
    listen 80;
    listen [::]:80;
    server_name ваш-домен.ru www.ваш-домен.ru;
    server_tokens off;

    # Статические файлы
    location /static/ {
        alias /home/django/projects/ваш-проект/staticfiles/;
        expires 1y;
        add_header Cache-Control "public, immutable";
    }

    # Медиа файлы
    location /media/ {
        alias /home/django/projects/ваш-проект/media/;
        expires 1y;
        add_header Cache-Control "public";
    }

    # Основное приложение
    location / {
        proxy_pass http://127.0.0.1:8000;
        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;

        # WebSocket поддержка
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        # Таймауты
        proxy_connect_timeout 60s;
        proxy_send_timeout 60s;
        proxy_read_timeout 60s;
    }

    # Запрет доступа к скрытым файлам
    location ~ /\. {
        deny all;
    }

    # Фавикон и robots.txt
    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }
```

```
# Кэширование статики
location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg)$ {
    expires 1y;
    add_header Cache-Control "public, immutable";
}
}
```

## 6.3. Активация сайта

```
# Создание символической ссылки
sudo ln -s /etc/nginx/sites-available/ваш-сайт /etc/nginx/sites-enabled/

# Удаление дефолтного конфига
sudo rm /etc/nginx/sites-enabled/default

# Проверка конфигурации
sudo nginx -t

# Перезапуск Nginx
sudo systemctl reload nginx
```

---

## Настройка домена и SSL

### 7.1. Настройка DNS

Добавьте А-записи в настройках домена:

```
@      -> IP вашего сервера
www    -> IP вашего сервера
```

### 7.2. Установка SSL (Let's Encrypt)

```
# Установка Certbot
sudo apt install -y certbot python3-certbot-nginx

# Получение SSL сертификата
sudo certbot --nginx -d ваш-домен.ru -d www.ваш-домен.ru

# Автоматическое обновление сертификата
sudo certbot renew --dry-run
```

### 7.3. Автоматическое перенаправление HTTP → HTTPS

Nginx автоматически обновит конфигурацию, добавив:



```
server {  
    listen 80;  
    server_name ваш-домен.ru www.ваш-домен.ru;  
    return 301 https://$server_name$request_uri;  
}
```



## Безопасность

### 8.1. Настройка фаервола (UFW)

```
# Включение UFW  
sudo ufw enable  
  
# Разрешение SSH  
sudo ufw allow OpenSSH  
  
# Разрешение HTTP/HTTPS  
sudo ufw allow 'Nginx Full'  
  
# Проверка правил  
sudo ufw status verbose
```

### 8.2. Безопасность SSH

```
# Редактирование конфигурации SSH  
sudo nano /etc/ssh/sshd_config
```

Измените:

```
Port 2222 # Изменение стандартного порта  
PermitRootLogin no  
PasswordAuthentication no # Только по ключу  
PubkeyAuthentication yes
```

```
# Перезапуск SSH  
sudo systemctl restart sshd
```

# Не забудьте добавить свой SSH ключ перед отключением пароля!

### 8.3. Настройка fail2ban (защита от брутфорса)

```
sudo apt install -y fail2ban  
sudo systemctl enable fail2ban  
sudo systemctl start fail2ban
```



# Мониторинг и логи

## 9.1. Просмотр логов в реальном времени

```
# Логи Nginx
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log

# Логи Gunicorn
sudo journalctl -u gunicorn -f

# Логи PostgreSQL
sudo tail -f /var/log/postgresql/postgresql-16-main.log
```

## 9.2. Установка мониторинга

```
# Установка htop для мониторинга ресурсов
sudo apt install -y htop

# Установка netdata (опционально)
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

## 9.3. Создание скрипта для проверки здоровья

```
nano ~/health_check.sh
```

```
#!/bin/bash

# Проверка сервисов
services=("nginx" "postgresql" "gunicorn")
for service in "${services[@]}; do
    if systemctl is-active --quiet $service; then
        echo "✅ $service is running"
    else
        echo "❌ $service is not running"
    fi
done

# Проверка дискового пространства
df -h /

# Проверка нагрузки
uptime

# Проверка памяти
free -h
```

```
chmod +x ~/health_check.sh
```

# Резервное копирование

## 10.1. Скрипт резервного копирования

```
nano ~/backup.sh
```

```
#!/bin/bash

# Настройки
BACKUP_DIR="/home/django/backups"
DATE=$(date +%Y%m%d_%H%M%S)
DB_NAME="n3connect"

# Создание директории для бэкапов
mkdir -p $BACKUP_DIR

# Бэкап базы данных
sudo -u postgres pg_dump $DB_NAME > $BACKUP_DIR/db_${DATE}.sql

# Бэкап медиа файлов
tar -czf $BACKUP_DIR/media_${DATE}.tar.gz /home/django/projects/ваш-проект/media/

# Бэкап кода
tar -czf $BACKUP_DIR/code_${DATE}.tar.gz /home/django/projects/ваш-проект/ --exclude=venv --exclude=*.pyc

# Удаление старых бэкапов (старше 7 дней)
find $BACKUP_DIR -name "*.sql" -mtime +7 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +7 -delete

echo "Backup completed: $BACKUP_DIR"
```

```
chmod +x ~/backup.sh
```

## 10.2. Автоматическое резервное копирование (cron)

```
# Редактирование cron задач
crontab -e
```

Добавьте:

```
# Ежедневный бэкап в 2:00 ночи
0 2 * * * /home/django/backup.sh

# Проверка состояния сервисов каждый час
0 * * * * /home/django/health_check.sh >> /home/django/health.log
```

# Быстрый чек-лист развертывания

## Проверка после установки:

```
# 1. Проверка сервисов
sudo systemctl status nginx
sudo systemctl status postgresql
sudo systemctl status gunicorn

# 2. Проверка доступности сайта
curl -I http://localhost
curl -I https://ваш-домен.ru

# 3. Проверка базы данных
sudo -u postgres psql -d n3connect -c "\dt"

# 4. Проверка статических файлов
ls -la /home/django/projects/ваш-проект/staticfiles/

# 5. Проверка логов на ошибки
sudo tail -f /var/log/nginx/error.log
sudo journalctl -u gunicorn -n 50
```

## Команды для обслуживания:

```
# Перезапуск всех сервисов
sudo systemctl restart nginx postgresql gunicorn

# Проверка конфигурации Nginx
sudo nginx -t

# Просмотр потребления ресурсов
htop

# Очистка кэша Django
python manage.py clear_cache

# Перезапуск Gunicorn (после изменений кода)
sudo systemctl restart gunicorn
```

## Устранение неполадок

## Распространенные проблемы и решения:

### 1. Ошибка 502 Bad Gateway

```
# Проверьте, работает ли Gunicorn
sudo systemctl status gunicorn

# Проверьте логи
sudo journalctl -u gunicorn -n 50

# Проверьте, слушает ли порт 8000
sudo netstat -tulpn | grep :8000
```

## 2. Ошибки с базой данных

```
# Проверьте подключение к PostgreSQL
sudo -u postgres psql -d n3connect

# Проверьте миграции
python manage.py migrate

# Создайте миграции заново
python manage.py makemigrations
python manage.py migrate
```

## 3. Проблемы со статическими файлами

```
# Пересоберите статические файлы
python manage.py collectstatic --noinput

# Проверьте права доступа
sudo chown -R django:www-data /home/django/projects/ваш-проект/staticfiles
sudo chmod -R 755 /home/django/projects/ваш-проект/staticfiles
```

## 4. Проблемы с SSL

```
# Обновите сертификат
sudo certbot renew --force-renewal

# Проверьте конфигурацию Nginx
sudo nginx -t

# Перезапустите Nginx
sudo systemctl reload nginx
```



## Контакты и дополнительная помощь

- Официальная документация Django: <https://docs.djangoproject.com/>
- Документация Ubuntu: <https://ubuntu.com/server/docs>
- Документация Nginx: <https://nginx.org/en/docs/>
- Документация PostgreSQL: <https://www.postgresql.org/docs/>

Для получения дополнительной помощи вы можете обратиться к:

- Логам сервера
- Сообществу Django
- Системным администраторам