ИНСТРУКЦИЯ

по установке и запуску программного обеспечения

«КУРС»
(ПО «КУРС»)

Правообладатель: ООО «СофтЭксперт» (ИНН 7107045310)

г. Тула
2026

1. Требования к системе и подготовка.

1.1 Аппаратные требования:

- CPU: 2-4 ядра (рекомендуется 4+)
- RAM: 4-8 ГБ
- SSD: 50-100 ГБ (в зависимости от объема данных)

1.2 Программные требования:

- ОС: Ubuntu 22.04 LTS (с долгосрочной поддержкой)
- Среда выполнения**:** .NET: ASP.NET Core 8.0 LTS
- Веб-сервер: Nginx: последняя стабильная версия
- База данных: PostgreSQL: 15+

2 Подготовка к развёртыванию

Развёртывание выполняется через SSH-подключение к серверу. Для передачи файлов рекомендуется использовать WinSCP или аналогичный инструмент.

Подключимся к серверу по SSH, используя его IP-адрес, логин и пароль

Подключиться к виртуальному серверу по SSH можно одной командой:

ssh <username>@<ip_adress>

где вместо username нужно указать логин пользователя, вместо ip-adress — IP-адрес

сервера, к которому вы подключаетесь. Если на сервере используется нестандартный

порт SSH, команда изменится:

ssh username@ip_adress -p <port>

где port - порт, по которому будет произведено подключение по SSH.

3. После ввода команды нажать Enter.

4. Система запросит пароль пользователя. При вводе пароля символы в командной

строке не отображаются: можно набрать пароль. После ввода нажать клавишу Enter.

2.1 Обновление системы и установка .NET Runtime

Необходимо установить среду выполнения .NET Runtime, для этого в терминале вводим команды

```
# Обновление списка пакетов и установка зависимостей
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install -y curl wget gnupg ca-certificates

# Добавление репозитория Microsoft и установка .NET Runtime
wget https://packages.microsoft.com/config/ubuntu/22.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
```

rm packages-microsoft-prod.deb

Установка ASP.NET Core Runtime 8.0

sudo apt-get update

sudo apt-get install -y aspnetcore-runtime-8.0

Проверка установки

dotnet --info

2.2 Установка и настройка PostgreSQL

Для приложения «КУРС» необходимо развернуть СУБД PostgreSQL

Добавление официального репозитория PostgreSQL

sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee /etc/apt/trusted.gpg.d/pgdg.asc

Установка PostgreSQL 15

sudo apt-get update

sudo apt-get install -y postgresql-15 postgresql-contrib-15

Настройка службы

sudo systemctl start postgresql

sudo systemctl enable postgresql

sudo systemctl status postgresqlНастройка базы данных:

Переключение на пользователя postgres

sudo -i -u postgres

Запуск psql

psql

-- Изменение пароля пользователя postgres

ALTER USER postgres PASSWORD 'your_new_password';

-- Выход из psql

\q

БД будут созданы и инициализированы при запуске приложений

2.3 Установка и настройка Nginx

Установка Nginx

sudo apt-get install -y nginx

Запуск и добавление в автозагрузку

sudo systemctl start nginx

sudo systemctl enable nginx

Пример конфигурации для проксирования запросов к API (файл /etc/nginx/sites-available/courseinfo):

```
server {
    listen 80;
    server_name your_domain_or_ip;

    location /courseapi/ {
        proxy_pass http://localhost:5001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

---

proxy_pass http://localhost:5001; порты указаны в качестве примера. Для каждого приложения укажите собственный порт из диапазона 1024-49151.

Активируйте конфигурацию и перезапустите Nginx:

sudo ln -s /etc/nginx/sites-available/ courseinfo/etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl restart nginx

sudo systemctl enable nginx

3 Развёртывание приложения

3.1 Создание рабочей директории

Определение имени проекта (замените your_project на актуальное название)

projectName="course_app"

Создание корневой директории проекта

sudo mkdir -p /var/www/$projectName

```
# Создание директорий для файлового хранилища
sudo mkdir -p /var/www/$projectName/files/
# Копирование файлов приложения
# Предполагается, что файлы находятся в локальной папке ./publish/
sudo cp -r ./publish/ /var/www/$projectName/api/
# Настройка прав доступа
sudo chown -R www-data:www-data /var/www/$projectName
sudo chmod -R 755 /var/www/$projectName
# Особые права для директорий с файлами (разрешение записи)
sudo chmod -R 775 /var/www/$projectName/wwwroot/Videos
```

3.2 Конфигурация приложения

Для корректной работы необходимо настроить следующие параметры в конфигурационном файле appsettings.json.

Сохранить файл и скопировать каталог приложения на сервер в ранее созданную папку

Настройка приложения

```
" ConnectionStrings: ": {
  "PostgresConnection": "" <ПАРАМЕТРЫ_ПОДКЛЮЧЕНИЯ_К_ОСНОВНОЙ_БД >"",
 },
 "EmailSender": {
  "Host": "<ПОЧТОВЫЙ_СЕРВЕР>",
  "Port": "<ПОРТ_ПОЧТОВОГО_СЕРВЕРА>",
  "NameFrom": "<ИМЯ_ОТПРАВИТЕЛЯ>",
  "EmailFrom": "<АДРЕС_ОТПРАВИТЕЛЯ>",
  "Password": "<ПАРОЛЬ_ПОЧТЫ>"
 },
 "Jwt": {
  "SecretKey": "<СЕКРЕТНЫЙ КЛЮЧ ДЛЯ ТОКЕНА МОБ. ПРИЛОЖЕНИЯ>",
  "ValidIssuer": "<ИЗДАТЕЛЬ КЛЮЧА>",
  "ValidAudience": "<АУДИТОРИЯ ТОКЕНА>"
 }
```

3.4 Создание systemd сервисов для приложений

```
# Обновление демона systemd
sudo systemctl daemon-reload
```

Для управления запуском и восстановлением приложений установим их в качестве служб

Для каждого приложения создадим отдельную службу

Создание файла сервиса

sudo nano /etc/systemd/system/kestrel-$<projectName.service>

Содержимое файла сервиса:

[Unit]

Description=ASP.NET Core Web Application - $<projectName.service>

[Service]

WorkingDirectory=/var/www/$<projectName.service>

ExecStart=/usr/bin/dotnet /var/www/$<projectName.service>/$<projectName.service>.Web.dll

Restart=always

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=dotnet-$<projectName.service>

Environment=ASPNETCORE_ENVIRONMENT=Production

Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

Environment=ASPNETCORE_HTTPS_PORT=5001

Environment=ASPNETCORE_URLS=http://*:5090

[Install]

WantedBy=multi-user.target

Где:

WorkingDirectory – путь к каталогу приложения

ExecStart - Команда запуска приложения.

/var/www/$<projectName.service>/$<projectName.service>.Web.dll - основная библиотека приложения

SyslogIdentifier - Идентификатор в системных логах

Environment=ASPNETCORE_URLS=http://*:5090 - URL и порты, которые слушает Kestrel

Порт для приложения должен совпадать с портом указанным для приложения в nginx(пункт 3.1)

Проверяем и перезапускаем Nginx

sudo nginx -t

sudo systemctl reload nginx

Для управления службами есть следующие команды:

Включение автозапуска

sudo systemctl enable kestrel-$projectName.service

Запуск сервиса

sudo systemctl start kestrel-$projectName.service

Проверка статуса

sudo systemctl status kestrel-$projectName.service

Остановка сервиса

sudo systemctl stop kestrel-$projectName.service

Перезапуск сервиса

sudo systemctl restart kestrel-$projectName.service

для приложения нужно включить автозапуск и запустить сервис, проверить статус. Система готова к работе.

4. **Запуск ПО**

Ввести в браузере адрес http:// your_domain_or_ip

Если все шаги были выполнены корректно, откроется стартовая страница