

Руководство по развертыванию ПО ЭВМ

“Система оценки техники бега”

Содержание

1. [Требования к серверу](#)
2. [Требования к программному обеспечению](#)
3. [Подготовка сервера](#)
4. [Установка приложения](#)
5. [Настройка systemd](#)
6. [Настройка Nginx](#)
7. [Настройка HTTPS](#)
8. [Проверка работоспособности](#)
9. [Обслуживание и мониторинг](#)
10. [Решение проблем](#)

1. Требования к серверу

1.1. Аппаратные требования

Минимальные требования

| Параметр | Значение | Примечание |
|----------|-----------------|-------------------------------|
| CPU | 2 ядра, 2.0 GHz | x86_64 архитектура |
| RAM | 4 ГБ | Для обработки видео до 30 сек |
| Диск | 20 ГБ | SSD рекомендуется |
| Сеть | 100 Мбит/с | Для загрузки видео |

Рекомендуемые требования

| Параметр | Значение | Примечание |
|----------|--------------------|--------------------------------|
| CPU | 4–8 ядер, 3.0+ GHz | Современный Intel/AMD |
| RAM | 8–16 ГБ | Для параллельной обработки |
| Диск | 50–100 ГБ SSD | NVMe для максимальной скорости |
| Сеть | 1 Гбит/с | Для комфортной работы |

Требования для GPU-ускорения (опционально)

| Параметр | Значение |
|----------|--------------------------------|
| GPU | NVIDIA с поддержкой CUDA 11.0+ |
| VRAM | 4+ ГБ |
| Драйвер | NVIDIA 450.0+ |
| CUDA | 11.0+ |
| cuDNN | 8.0+ |

Примечание: GPU значительно ускоряет обработку (в 3–5 раз), но не является обязательным.

1.2. Оценка производительности

| Конфигурация | Время обработки 10 сек видео |
|------------------------|------------------------------|
| 2 CPU, 4 GB RAM | ~60–90 секунд |
| 4 CPU, 8 GB RAM | ~30–45 секунд |
| 8 CPU, 16 GB RAM | ~15–25 секунд |
| 4 CPU + GPU (RTX 3060) | ~5–10 секунд |

1.3. Требования к дисковому пространству

| Компонент | Размер |
|----------------------------------|--------------|
| Операционная система | 5 ГБ |
| Python + виртуальное окружение | 2 ГБ |
| Модели RTMPose (YOLOX + RTMPose) | 500 МБ |
| Временные файлы (uploads) | 10+ ГБ |
| Логи | 1 ГБ |
| Итого минимум | 20 ГБ |

2. Требования к программному обеспечению

2.1. Операционная система

| ОС | Версия | Статус |
|--------------------|---------------|---|
| Ubuntu | 20.04 LTS | <input checked="" type="checkbox"/> Поддерживается |
| Ubuntu | 22.04 LTS | <input checked="" type="checkbox"/> Рекомендуется |
| Ubuntu | 24.04 LTS | <input checked="" type="checkbox"/> Поддерживается |
| Debian | 11 (Bullseye) | <input checked="" type="checkbox"/> Поддерживается |
| Debian | 12 (Bookworm) | <input checked="" type="checkbox"/> Поддерживается |
| Astra Linux CE, SE | 1.2, 2.3 | <input checked="" type="checkbox"/> Поддерживается |
| CentOS/RHEL | 8+ |  Требует адаптации |
| Windows Server | — |  Не поддерживается |

2.2. Системные пакеты

| Пакет | Версия | Назначение |
|--------|----------|---------------|
| Python | 3.8–3.11 | Интерпретатор |

| Пакет | Версия | Назначение |
|---------|--------|----------------------------|
| pip | 21.0+ | Менеджер пакетов Python |
| venv | — | Виртуальные окружения |
| FFmpeg | 4.0+ | Конвертация видео |
| Nginx | 1.18+ | Веб-сервер / reverse proxy |
| systemd | — | Управление сервисами |

2.3. Python-зависимости

| Пакет | Версия | Назначение |
|------------------------|--------|-------------------------|
| flask | ≥2.0 | Веб-фреймворк |
| gunicorn | ≥20.0 | WSGI-сервер |
| werkzeug | ≥2.0 | Утилиты HTTP |
| numpy | ≥1.20 | Математические операции |
| opencv-python-headless | ≥4.5 | Обработка видео |
| rtmllib | ≥0.0.9 | RTMPose inference |
| onnxruntime | ≥1.10 | Запуск ONNX-моделей |
| tqdm | ≥4.0 | Прогресс-бары |
| requests | ≥2.25 | HTTP-клиент |

Для GPU-ускорения (опционально)

| Пакет | Версия | Назначение |
|-------------------|--------|-------------|
| onnxruntime-gpu | ≥1.10 | ONNX с CUDA |
| nvidia-cudnn-cu11 | ≥8.0 | cuDNN |

2.4. Модели машинного обучения

| Модель | Размер | URL |
|-----------|---------|---------------------------|
| YOLOX-M | ~190 МБ | OpenMMLab |
| RTMPose-M | ~45 МБ | OpenMMLab |

Модели загружаются автоматически при первом запуске через rtmlib.

3. Подготовка сервера

3.1. Обновление системы

```
sudo apt update && sudo apt upgrade -y
sudo reboot # При обновлении ядра
```

3.2. Установка системных пакетов

```
# Основные пакеты
sudo apt install -y \
    python3 \
    python3-pip \
    python3-venv \
    nginx \
    ffmpeg \
    git \
    curl \
    htop
# Проверка версий
python3 --version # Ожидается: Python 3.8+
ffmpeg -version # Ожидается: ffmpeg version 4.x+
nginx -v      # Ожидается: nginx/1.18+
```

3.3. Настройка файрвола

```
# Если используется UFW
sudo ufw allow 22/tcp # SSH
sudo ufw allow 80/tcp # HTTP
sudo ufw allow 443/tcp # HTTPS
sudo ufw enable
sudo ufw status
```

3.4. Создание системного пользователя

```
# Создаём пользователя без интерактивного входа
sudo useradd -r -s /bin/false -d /var/www/runform runform
# Создаём рабочую директорию
sudo mkdir -p /var/www/runform
sudo chown runform:runform /var/www/runform
```

4. Установка приложения

4.1. Загрузка файлов

Вариант А: Копирование через SCP

```
# С локальной машины
scp -r ./runform_analyzer/* user@server:/tmp/runform/
# На сервере
sudo cp -r /tmp/runform/* /var/www/runform/
sudo chown -R runform:runform /var/www/runform
```

Вариант В: Загрузка архива

```
cd /var/www/runform
sudo -u runform wget https://example.com/runform_analyzer.zip
sudo -u runform unzip runform_analyzer.zip
rm runform_analyzer.zip
```

4.2. Структура директорий

```
/var/www/runform/
├── app.py          # Основное приложение
├── requirements.txt # Зависимости Python
└── templates/
    └── index.html    # HTML-шаблон
└── static/
    ├── uploads/      # Загруженные видео (создаётся автоматически)
    └── demo/          # Демо-материалы
        ├── demo_runner.mp4
        ├── demo_runner_analyzed.mp4
        └── demo_metrics.json
└── venv/           # Виртуальное окружение (создаётся)
```

4.3. Создание виртуального окружения

```
cd /var/www/runform
# Создаём venv от имени пользователя runform
sudo -u runform python3 -m venv venv
```

```
# Активируем и устанавливаем зависимости
sudo -u runform bash << 'EOF'
source venv/bin/activate
pip install --upgrade pip wheel setuptools
# Установка зависимостей
pip install \
    flask>=2.0 \
    gunicorn>=20.0 \
    werkzeug>=2.0 \
    numpy>=1.20 \
    opencv-python-headless>=4.5 \
    rtllib>=0.0.9 \
    onnxruntime>=1.10 \
    tqdm>=4.0 \
    requests>=2.25
# Проверка установки
python -c "import flask, cv2, numpy, rtllib; print('✓ Все зависимости установлены')"
EOF
```

4.4. Установка для GPU (опционально)

```
# Вместо onnxruntime используем onnxruntime-gpu
sudo -u runform bash << 'EOF'
source venv/bin/activate
pip uninstall -y onnxruntime
pip install onnxruntime-gpu>=1.10
EOF
```

4.5. Настройка прав доступа

```
# Права на директории
sudo chown -R runform:runform /var/www/runform
sudo chmod 755 /var/www/runform
sudo chmod 755 /var/www/runform/static
sudo chmod 755 /var/www/runform/static/uploads
sudo chmod 755 /var/www/runform/static/demo
# Права на файлы
sudo chmod 644 /var/www/runform/app.py
sudo chmod 644 /var/www/runform/templates/*.html
```

5. Настройка systemd

5.1. Создание unit-файла

```
sudo tee /etc/systemd/system/runform.service > /dev/null << 'EOF'
[Unit]
Description=RunForm Analyzer - Running Technique Analysis System
Documentation=https://github.com/your-repo/runform
After=network.target
Wants=network-online.target
[Service]
Type=notify
User=runform
Group=runform
WorkingDirectory=/var/www/runform
Environment="PATH=/var/www/runform/venv/bin:/usr/local/bin:/usr/bin"
Environment="PYTHONUNBUFFERED=1"
# Gunicorn запуск
ExecStart=/var/www/runform/venv/bin/gunicorn \
    --workers 2 \
    --threads 4 \
    --worker-class gthread \
    --bind 127.0.0.1:5000 \
    --timeout 300 \
    --graceful-timeout 30 \
    --keep-alive 5 \
    --max-requests 1000 \
    --max-requests-jitter 50 \
    --access-logfile /var/www/runform/logs/access.log \
    --error-logfile /var/www/runform/logs/error.log \
    --capture-output \
    --log-level info \
    app:app
# Перезапуск при сбоях
Restart=always
RestartSec=5
StartLimitIntervalSec=60
StartLimitBurst=3
# Ограничения ресурсов
MemoryMax=4G
CPUQuota=200%
# Безопасность
NoNewPrivileges=yes
ProtectSystem=strict
```

```

ProtectHome=yes
ReadWritePaths=/var/www/runform/static/uploads /var/www/runform/logs
PrivateTmp=yes
[Install]
WantedBy=multi-user.target
EOF

```

5.2. Создание директории для логов

```

sudo mkdir -p /var/www/runform/logs
sudo chown runform:runform /var/www/runform/logs

```

5.3. Активация сервиса

```

# Перечитываем конфигурацию systemd
sudo systemctl daemon-reload
# Включаем автозапуск
sudo systemctl enable runform
# Запускаем сервис
sudo systemctl start runform
# Проверяем статус
sudo systemctl status runform

```

5.4. Расчёт количества воркеров

Формула: `workers = (2 × CPU_CORES) + 1`

| CPU ядер | Workers | Threads | Примечание |
|----------|---------|---------|----------------------|
| 1 | 2 | 2 | Минимум |
| 2 | 3 | 4 | Базовая конфигурация |
| 4 | 5 | 4 | Рекомендуется |
| 8 | 9 | 4 | Высокая нагрузка |

6. Настройка Nginx

6.1. Создание конфигурации

```

sudo tee /etc/nginx/sites-available/runform > /dev/null << 'EOF'
# Upstream для Gunicorn

```

```
upstream runform_backend {
    server 127.0.0.1:5000 fail_timeout=0;
    keepalive 32;
}

server {
    listen 80;
    listen [::]:80;
    server_name _; # Замените на ваш домен
    # Логирование
    access_log /var/log/nginx/runform_access.log;
    error_log /var/log/nginx/runform_error.log;
    # Максимальный размер загружаемого файла
    client_max_body_size 100M;
    client_body_timeout 300s;

    # Буферы для загрузки
    client_body_buffer_size 128k;

    # Основной location
    location / {
        proxy_pass http://runform_backend;
        proxy_http_version 1.1;

        # Заголовки
        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;
        proxy_set_header Connection "";

        # Таймауты для долгой обработки видео
        proxy_connect_timeout 60s;
        proxy_send_timeout 300s;
        proxy_read_timeout 300s;

        # Буферы
        proxy_buffering on;
        proxy_buffer_size 4k;
        proxy_buffers 8 32k;
        proxy_busy_buffers_size 64k;
    }

    # Статические файлы через Nginx
    location /static/ {
```

```

alias /var/www/runform/static/;
expires 7d;
add_header Cache-Control "public, immutable";

# CORS для видео
add_header Access-Control-Allow-Origin *;

# Gzip для JSON
gzip on;
gzip_types application/json;
}

# Healthcheck endpoint
location /health {
    access_log off;
    return 200 "OK\n";
    add_header Content-Type text/plain;
}
}

EOF

```

6.2. Активация конфигурации

```

# Создаём симлинк
sudo ln -sf /etc/nginx/sites-available/runform /etc/nginx/sites-enabled/
# Удаляем дефолтный сайт
sudo rm -f /etc/nginx/sites-enabled/default
# Проверяем конфигурацию
sudo nginx -t
# Перезагружаем Nginx
sudo systemctl reload nginx

```

6.3. Оптимизация Nginx

```

# Редактируем /etc/nginx/nginx.conf
sudo tee -a /etc/nginx/nginx.conf > /dev/null << 'EOF'
# Оптимизации в секции http {}
# worker_connections 4096;
# multi_accept on;
# sendfile on;
# tcp_nopush on;
# tcp_nodelay on;
EOF

```

7. Настройка HTTPS

7.1. Установка Certbot

```
sudo apt install -y certbot python3-certbot-nginx
```

7.2. Получение сертификата

```
# Замените example.com на ваш домен
sudo certbot --nginx -d example.com -d www.example.com
# Следуйте инструкциям:
# 1. Введите email
# 2. Согласитесь с условиями
# 3. Выберите редирект HTTP -> HTTPS (рекомендуется)
```

7.3. Автоматическое обновление

```
# Проверка автообновления
sudo certbot renew --dry-run
# Certbot автоматически добавляет cron-задачу
sudo systemctl status certbot.timer
```

7.4. Итоговая конфигурация с HTTPS

После Certbot конфигурация автоматически обновится:

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name example.com www.example.com;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    # ... остальная конфигурация
}

server {
    listen 80;
    server_name example.com www.example.com;
    return 301 https://$server_name$request_uri;
}
```

8. Проверка работоспособности

8.1. Проверка сервисов

```
# Статус приложения
sudo systemctl status runform
# Статус Nginx
sudo systemctl status nginx
# Прослушиваемые порты
sudo ss -tlnp | grep -E ':(80|443|5000)'
```

8.2. Тестовые запросы

```
# Проверка localhost
curl -I http://127.0.0.1:5000/
# Проверка через Nginx
curl -I http://localhost/
# Проверка healthcheck
curl http://localhost/health
# Проверка demo-ready
curl http://localhost/demo-ready
```

8.3. Тест загрузки видео

```
# Тестовый запрос с файлом
curl -X POST \
  -F "video=@test_video.mp4" \
  -F "smoothing=medium" \
  http://localhost/analyze
```

8.4. Проверка логов

```
# Логи приложения
tail -f /var/www/runform/logs/error.log
# Логи Nginx
tail -f /var/log/nginx/runform_error.log
# Логи systemd
sudo journalctl -u runform -f
```

9. Обслуживание и мониторинг

9.1. Ротация логов

```
sudo tee /etc/logrotate.d/runform > /dev/null << 'EOF'
```

```
/var/www/runform/logs/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 runform runform
    sharedscripts
    postrotate
        systemctl reload runform > /dev/null 2>&1 || true
    endscript
}
EOF
```

9.2. Очистка временных файлов

```
# Создаём скрипт очистки
sudo tee /var/www/runform/cleanup.sh > /dev/null << 'EOF'
#!/bin/bash
# Удаляем файлы старше 24 часов из uploads
find /var/www/runform/static/uploads -type f -mtime +1 -delete
# Удаляем пустые директории
find /var/www/runform/static/uploads -type d -empty -delete
EOF
sudo chmod +x /var/www/runform/cleanup.sh
sudo chown runform:runform /var/www/runform/cleanup.sh
# Добавляем в cron
echo "0 3 * * * runform /var/www/runform/cleanup.sh" | sudo tee /etc/cron.d/runform-cleanup
```

9.3. Мониторинг ресурсов

```
# Использование памяти процессом
ps aux | grep gunicorn
# Использование диска
df -h /var/www/runform
# Размер uploads
du -sh /var/www/runform/static/uploads
```

9.4. Обновление приложения

```
# Остановка сервиса
sudo systemctl stop runform
# Резервное копирование
sudo cp -r /var/www/runform /var/www/runform.backup
```

```
# Обновление файлов
sudo cp -r /path/to/new/files/* /var/www/runform/
sudo chown -R runform:runform /var/www/runform
# Обновление зависимостей
sudo -u runform bash -c 'source /var/www/runform/venv/bin/activate && pip install -r /var/www/runform/requirements.txt'
# Запуск сервиса
sudo systemctl start runform
```

10. Решение проблем

10.1. Сервис не запускается

```
# Подробные логи
sudo journalctl -u runform -n 100 --no-pager
# Проверка синтаксиса Python
sudo -u runform /var/www/runform/venv/bin/python -c "from app import app"
# Проверка прав
ls -la /var/www/runform/
```

10.2. Ошибка 502 Bad Gateway

```
# Gunicorn не запущен
sudo systemctl status runform
# Неверный адрес upstream
curl http://127.0.0.1:5000/
# Проверка SELinux (CentOS/RHEL)
sudo setsebool -P httpd_can_network_connect 1
```

10.3. Ошибка загрузки файла

```
# Проверка размера в Nginx
grep client_max_body_size /etc/nginx/sites-enabled/runform
# Проверка прав uploads
ls -la /var/www/runform/static/uploads/
sudo chown runform:runform /var/www/runform/static/uploads
sudo chmod 755 /var/www/runform/static/uploads
```

10.4. Таймаут при обработке

```
# Увеличение таймаутов Gunicorn
# В runform.service: --timeout 600
# Увеличение таймаутов Nginx
# В nginx конфиге: proxy_read_timeout 600s;
```

```
# Перезапуск
sudo systemctl daemon-reload
sudo systemctl restart runform
sudo systemctl reload nginx
```

10.5. Нехватка памяти

```
# Проверка памяти
free -h
# Уменьшение воркеров
# В runform.service: --workers 1
# Добавление swap (если нет)
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

10.6. Модели не загружаются

```
# Проверка интернета
curl -I https://download.openmmlab.com
# Ручная загрузка моделей
sudo -u runform bash << 'EOF'
source /var/www/runform/venv/bin/activate
python -c "
from rtmllib import PoseTracker
tracker = PoseTracker(mode='lightweight')
print('Models loaded successfully')
"
EOF
```

10.7. FFmpeg не найден

```
# Проверка установки
which ffmpeg
ffmpeg -version
# Установка если отсутствует
sudo apt install -y ffmpeg
# Проверка пути в приложении
grep -n "ffmpeg" /var/www/runform/app.py
```

Приложение А. Быстрая установка (скрипт)

```
#!/bin/bash
# deploy.sh - Автоматическая установка RunForm Analyzer
set -e
echo "==== RunForm Analyzer Deployment ===="
# 1. Системные пакеты
sudo apt update
sudo apt install -y python3-pip python3-venv nginx ffmpeg
# 2. Пользователь и директория
sudo useradd -r -s /bin/false -d /var/www/runform runform 2>/dev/null || true
sudo mkdir -p /var/www/runform
sudo chown runform:runform /var/www/runform
# 3. Копирование файлов
sudo cp -r ./ /var/www/runform/
sudo chown -R runform:runform /var/www/runform
# 4. Python окружение
sudo -u runform bash -c '
cd /var/www/runform
python3 -m venv venv
source venv/bin/activate
pip install --upgrade pip
pip install flask gunicorn numpy opencv-python-headless rtmllib onnxruntime tqdm requests
'

# 5. Директория логов
sudo mkdir -p /var/www/runform/logs
sudo chown runform:runform /var/www/runform/logs
# 6. Systemd сервис
sudo cp /var/www/runform/runform.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable runform
sudo systemctl start runform
# 7. Nginx
sudo cp /var/www/runform/nginx.conf /etc/nginx/sites-available/runform
sudo ln -sf /etc/nginx/sites-available/runform /etc/nginx/sites-enabled/
sudo rm -f /etc/nginx/sites-enabled/default
sudo nginx -t
sudo systemctl reload nginx
echo "==== Deployment Complete ===="
echo "Access: http://$(hostname -I | awk '{print $1}')/"
```

Приложение В. Чек-лист разёртывания

- Сервер соответствует минимальным требованиям
- Установлены системные пакеты (python3, nginx, ffmpeg)
- Создан пользователь runform
- Скопированы файлы приложения
- Создано виртуальное окружение Python
- Установлены Python-зависимости
- Настроен systemd сервис
- Сервис runform запущен и работает
- Настроен Nginx
- Nginx перезагружен без ошибок
- Приложение доступно по HTTP
- (Опционально) Настроен HTTPS
- Настроена ротация логов
- Настроена очистка временных файлов

Версия документа: 1.0

Дата: декабрь 2025