

Руководство по разворачиванию - BaiMuras Platform

Обзор

Данное руководство описывает процесс разворачивания BaiMuras Platform в различных окружениях: от локальной разработки до продакшн сервера.

Системные требования

Минимальные требования

- **CPU:** 2 ядра
- **RAM:** 4 GB
- **Диск:** 20 GB свободного места
- **ОС:** Ubuntu 20.04+ / CentOS 8+ / Debian 11+

Рекомендуемые требования (продакшн)

- **CPU:** 4 ядра
- **RAM:** 8 GB
- **Диск:** 50 GB SSD
- **ОС:** Ubuntu 22.04 LTS

Программное обеспечение

- **Python:** 3.11+
- **PostgreSQL:** 13+
- **Redis:** 6+
- **Nginx:** 1.18+
- **Docker:** 20+ (опционально)
- **Node.js:** 18+ (для фронтенд сборки)

Локальное разворачивание

1. Подготовка окружения

```
# Обновление системы
sudo apt update && sudo apt upgrade -y

# Установка зависимостей
sudo apt install -y python3.11 python3.11-venv python3-pip postgresql postgresql-contrib redis-server nginx git

# Установка Docker (опционально)
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
```

2. Клонирование проекта

```
git clone https://github.com/ardakchapaev/baimuras.space.git
cd baimuras.space
git checkout fix-setup-env-1750496460
```

3. Настройка Python окружения

```
# Создание виртуального окружения
python3.11 -m venv venv
source venv/bin/activate

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

4. Настройка базы данных

```
# Создание пользователя и базы данных PostgreSQL
sudo -u postgres psql << EOF
CREATE USER baimuras_user WITH PASSWORD 'BaiMuras2025!@#';
CREATE DATABASE baimuras_db OWNER baimuras_user;
GRANT ALL PRIVILEGES ON DATABASE baimuras_db TO baimuras_user;
\q
EOF

# Инициализация миграций
export FLASK_APP=src/main.py
flask db init
flask db migrate -m "Initial migration"
flask db upgrade
```

5. Настройка Redis

```
# Настройка Redis с паролем
sudo tee /etc/redis/redis.conf > /dev/null << EOF
bind 127.0.0.1
port 6379
requirepass baimuras2025
maxmemory 256mb
maxmemory-policy allkeys-lru
EOF

# Перезапуск Redis
sudo systemctl restart redis-server
sudo systemctl enable redis-server
```

6. Конфигурация приложения

```
# Копирование и настройка переменных окружения
cp .env.example .env

# Редактирование .env файла
nano .env
```

Основные переменные для локального развертывания:

```
SECRET_KEY=your-super-secret-key-change-in-production
JWT_SECRET_KEY=your-jwt-secret-key-change-in-production
FLASK_ENV=development

DATABASE_URL=postgresql://baimuras_user:BaiMuras2025!@@localhost:5432/baimuras_db
REDIS_URL=redis://:baimuras2025@localhost:6379/0
CELERY_BROKER_URL=redis://:baimuras2025@localhost:6379/0
CELERY_RESULT_BACKEND=redis://:baimuras2025@localhost:6379/0

MAIL_SERVER=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=your-email@gmail.com
MAIL_PASSWORD=your-app-password
MAIL_DEFAULT_SENDER=info@baimuras.space
```

7. Запуск приложения

```
# Активация виртуального окружения
source venv/bin/activate

# Запуск основного приложения
python wsgi.py

# В отдельном терминале - запуск Celery worker
celery -A automation.celery_app worker --loglevel=info

# В третьем терминале - запуск Celery scheduler
celery -A automation.celery_app beat --loglevel=info
```

Приложение будет доступно по адресу: `http://localhost:5000`

Docker развертывание

1. Быстрый запуск

```
# Клонирование проекта
git clone https://github.com/ardakchapaev/baimuras.space.git
cd baimuras.space
git checkout fix-setup-env-1750496460

# Копирование переменных окружения
cp .env.example .env

# Запуск всех сервисов
docker-compose up -d

# Просмотр логов
docker-compose logs -f
```

2. Конфигурация Docker

docker-compose.yml включает следующие сервисы:

- **flask** - основное приложение
- **postgres** - база данных
- **redis** - кеширование и очереди
- **celery-worker** - обработчик фоновых задач
- **celery-scheduler** - планировщик задач
- **nginx** - reverse proxy

3. Управление Docker сервисами

```
# Запуск сервисов
docker-compose up -d

# Остановка сервисов
docker-compose down

# Перезапуск конкретного сервиса
docker-compose restart flask

# Просмотр логов
docker-compose logs -f flask

# Выполнение команд в контейнере
docker-compose exec flask bash

# Миграции базы данных
docker-compose exec flask flask db upgrade
```

Продакшн развертывание

1. Подготовка сервера

```
# Подключение к серверу
ssh root@95.140.153.181

# Обновление системы
apt update && apt upgrade -y

# Установка необходимого ПО
apt install -y python3.11 python3.11-venv postgresql postgresql-contrib redis-server ng
inx certbot python3-certbot-nginx git htop curl
```

2. Настройка пользователя

```
# Создание пользователя для приложения
adduser baimuras
usermod -aG sudo baimuras
su - baimuras
```

3. Клонирование и настройка проекта

```
# Клонирование в домашнюю директорию
cd /home/baimuras
git clone https://github.com/ardakchapaev/baimuras.space.git
cd baimuras.space
git checkout fix-setup-env-1750496460

# Настройка виртуального окружения
python3.11 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

4. Настройка базы данных (продакшн)

```
# Настройка PostgreSQL
sudo -u postgres psql << EOF
CREATE USER baimuras_user WITH PASSWORD 'BaiMuras2025!@#';
CREATE DATABASE baimuras_db OWNER baimuras_user;
GRANT ALL PRIVILEGES ON DATABASE baimuras_db TO baimuras_user;
ALTER USER baimuras_user CREATEDB;
\q
EOF

# Настройка pg_hba.conf для безопасности
sudo nano /etc/postgresql/14/main/pg_hba.conf
# Добавить: local baimuras_db baimuras_user md5

sudo systemctl restart postgresql
```

5. Конфигурация продакшн переменных

```
# Создание продакшн .env файла
cat > .env << EOF
SECRET_KEY=$(python3 -c 'import secrets; print(secrets.token_hex(32))')
JWT_SECRET_KEY=$(python3 -c 'import secrets; print(secrets.token_hex(32))')
FLASK_ENV=production

DATABASE_URL=postgresql://baimuras_user:BaiMuras2025!@##localhost:5432/baimuras_db
REDIS_URL=redis://:baimuras2025@localhost:6379/0
CELERY_BROKER_URL=redis://:baimuras2025@localhost:6379/0
CELERY_RESULT_BACKEND=redis://:baimuras2025@localhost:6379/0

MAIL_SERVER=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=admin@baimuras.space
MAIL_PASSWORD=your-app-password
MAIL_DEFAULT_SENDER=info@baimuras.space

DOMAIN=baimuras.space
ADMIN_DOMAIN=hub.baimuras.space
AUTOMATION_DOMAIN=automation.baimuras.space

LOG_LEVEL=INFO
LOG_FILE=/var/log/baimuras/app.log
EOF
```

6. Настройка Nginx

```

# Создание конфигурации Nginx
sudo tee /etc/nginx/sites-available/baimuras.space << EOF
server {
    listen 80;
    server_name baimuras.space www.baimuras.space;
    return 301 https://\$server_name\$request_uri;
}

server {
    listen 443 ssl http2;
    server_name baimuras.space www.baimuras.space;

    ssl_certificate /etc/letsencrypt/live/baimuras.space/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/baimuras.space/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-
GCM-SHA384:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;

    location / {
        proxy_pass http://127.0.0.1:5000;
        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;
    }

    location /static {
        alias /home/baimuras/baimuras.space/src/static;
        expires 1y;
        add_header Cache-Control "public, immutable";
    }
}

# Админ-панель
server {
    listen 443 ssl http2;
    server_name hub.baimuras.space;

    ssl_certificate /etc/letsencrypt/live/baimuras.space/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/baimuras.space/privkey.pem;

    location / {
        proxy_pass http://127.0.0.1:5000/dashboard;
        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;
    }
}
EOF

# Активация конфигурации
sudo ln -s /etc/nginx/sites-available/baimuras.space /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```


7. SSL сертификаты

```
# Получение SSL сертификатов Let's Encrypt
sudo certbot --nginx -d baimuras.space -d www.baimuras.space -d hub.baimuras.space

# Автоматическое обновление сертификатов
sudo crontab -e
# Добавить: 0 12 * * * /usr/bin/certbot renew --quiet
```

8. Systemd сервисы

Основное приложение:

```
sudo tee /etc/systemd/system/baimuras.service << EOF
[Unit]
Description=BaiMuras Flask Application
After=network.target postgresql.service redis.service

[Service]
Type=exec
User=baimuras
Group=baimuras
WorkingDirectory=/home/baimuras/baimuras.space
Environment=PATH=/home/baimuras/baimuras.space/venv/bin
ExecStart=/home/baimuras/baimuras.space/venv/bin/gunicorn -c gunicorn.conf.py wsgi:app
ExecReload=/bin/kill -s HUP \${MAINPID}
Restart=always
RestartSec=3

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

Celery Worker:

```
sudo tee /etc/systemd/system/baimuras-worker.service << EOF
[Unit]
Description=BaiMuras Celery Worker
After=network.target redis.service

[Service]
Type=exec
User=baimuras
Group=baimuras
WorkingDirectory=/home/baimuras/baimuras.space
Environment=PATH=/home/baimuras/baimuras.space/venv/bin
ExecStart=/home/baimuras/baimuras.space/venv/bin/celery -A automation.celery_app worker
--loglevel=info
Restart=always
RestartSec=3

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

Celery Scheduler:

```

sudo tee /etc/systemd/system/baimuras-scheduler.service << EOF
[Unit]
Description=BaiMuras Celery Scheduler
After=network.target redis.service

[Service]
Type=exec
User=baimuras
Group=baimuras
WorkingDirectory=/home/baimuras/baimuras.space
Environment=PATH=/home/baimuras/baimuras.space/venv/bin
ExecStart=/home/baimuras/baimuras.space/venv/bin/celery -A automation.celery_app beat
--loglevel=info
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
EOF

```

9. Запуск сервисов

```

# Перезагрузка systemd и запуск сервисов
sudo systemctl daemon-reload
sudo systemctl enable baimuras baimuras-worker baimuras-scheduler
sudo systemctl start baimuras baimuras-worker baimuras-scheduler

# Проверка статуса
sudo systemctl status baimuras
sudo systemctl status baimuras-worker
sudo systemctl status baimuras-scheduler

```

n8n Автоматизация**1. Установка n8n**

```

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

# Установка n8n глобально
sudo npm install -g n8n

# Создание пользователя для n8n
sudo adduser n8n
sudo su - n8n

```

2. Настройка n8n

```
# Создание конфигурационного файла
mkdir -p ~/.n8n
cat > ~/.n8n/.env << EOF
N8N_BASIC_AUTH_ACTIVE=true
N8N_BASIC_AUTH_USER=admin@baimuras.space
N8N_BASIC_AUTH_PASSWORD=Admin123!
N8N_HOST=0.0.0.0
N8N_PORT=5678
N8N_PROTOCOL=http
WEBHOOK_URL=http://95.140.153.181:5678/
EOF
```

3. Systemd сервис для n8n

```
sudo tee /etc/systemd/system/n8n.service << EOF
[Unit]
Description=n8n workflow automation
After=network.target

[Service]
Type=exec
User=n8n
ExecStart=/usr/bin/n8n start
Restart=always
RestartSec=3
Environment=PATH=/usr/bin:/usr/local/bin
Environment=NODE_ENV=production

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable n8n
sudo systemctl start n8n
```

Мониторинг и логирование

1. Настройка логирования

```
# Создание директории для логов
sudo mkdir -p /var/log/baimuras
sudo chown baimuras:baimuras /var/log/baimuras

# Настройка ротации логов
sudo tee /etc/logrotate.d/baimuras << EOF
/var/log/baimuras/*.log {
    daily
    missingok
    rotate 30
    compress
    delaycompress
    notifempty
    create 644 baimuras baimuras
    postrotate
        systemctl reload baimuras
    endscript
}
EOF
```

2. Мониторинг сервисов

```
# Скрипт проверки здоровья системы
cat > /home/baimuras/health_check.sh << 'EOF'
#!/bin/bash

echo "=== BaiMuras Health Check $(date) ==="

# Проверка сервисов
services=("baimuras" "baimuras-worker" "baimuras-scheduler" "postgresql" "redis"
"nginx")
for service in "${services[@]}; do
    if systemctl is-active --quiet $service; then
        echo "  $service: Running"
    else
        echo "  $service: Not running"
    fi
done

# Проверка дискового пространства
df -h / | tail -1 | awk '{print "  Disk usage: " $5 " of " $2}'

# Проверка памяти
free -h | grep Mem | awk '{print "  Memory usage: " $3 " of " $2}'

# Проверка подключения к БД
if sudo -u baimuras psql -d baimuras_db -c "SELECT 1;" > /dev/null 2>&1; then
    echo "  Database: Connected"
else
    echo "  Database: Connection failed"
fi

# Проверка Redis
if redis-cli -a baimuras2025 ping > /dev/null 2>&1; then
    echo "  Redis: Connected"
else
    echo "  Redis: Connection failed"
fi

echo "=====
EOF

chmod +x /home/baimuras/health_check.sh

# Добавление в cron для ежечасной проверки
(crontab -l 2>/dev/null; echo "0 * * * * /home/baimuras/health_check.sh >> /var/log/
baimuras/health.log") | crontab -
```

Резервное копирование

1. Скрипт backup

```
cat > /home/baimuras/backup.sh << 'EOF'
#!/bin/bash

BACKUP_DIR="/home/baimuras/backups"
DATE=$(date +%Y%m%d_%H%M%S)
DB_BACKUP="$BACKUP_DIR/db_backup_$DATE.sql"
FILES_BACKUP="$BACKUP_DIR/files_backup_$DATE.tar.gz"

# Создание директории для backup
mkdir -p $BACKUP_DIR

# Backup базы данных
pg_dump -h localhost -U baimuras_user -d baimuras_db > $DB_BACKUP

# Backup файлов приложения
tar -czf $FILES_BACKUP -C /home/baimuras baimuras.space --exclude=venv --
exclude=__pycache__ --exclude=.git

# Удаление старых backup (старше 30 дней)
find $BACKUP_DIR -name "*.sql" -mtime +30 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +30 -delete

echo "Backup completed: $DATE"
EOF

chmod +x /home/baimuras/backup.sh

# Автоматический backup каждые 6 часов
(crontab -l 2>/dev/null; echo "0 */6 * * * /home/baimuras/backup.sh >> /var/log/
baimuras/backup.log") | crontab -
```

Обновление приложения

1. Скрипт обновления

```
cat > /home/baimuras/update.sh << 'EOF'
#!/bin/bash

echo "Starting BaiMuras update..."

# Переход в директорию проекта
cd /home/baimuras/baimuras.space

# Создание backup перед обновлением
/home/baimuras/backup.sh

# Получение последних изменений
git fetch origin
git checkout fix-setup-env-1750496460
git pull origin fix-setup-env-1750496460

# Активация виртуального окружения
source venv/bin/activate

# Обновление зависимостей
pip install -r requirements.txt

# Миграции базы данных
flask db upgrade

# Перезапуск сервисов
sudo systemctl restart baimuras baimuras-worker baimuras-scheduler

echo "Update completed successfully!"
EOF

chmod +x /home/baimuras/update.sh
```

Устранение неполадок

Частые проблемы

1. Приложение не запускается

```
# Проверка логов
sudo journalctl -u baimuras -f

# Проверка конфигурации
source venv/bin/activate
python -c "from src.config import config; print('Config loaded successfully')"
```

2. Ошибки базы данных

```
# Проверка подключения
sudo -u baimuras psql -d baimuras_db -c "SELECT version();"

# Проверка миграций
flask db current
flask db upgrade
```

3. Проблемы с Redis

```
# Проверка статуса Redis
sudo systemctl status redis-server

# Тест подключения
redis-cli -a baimuras2025 ping
```

4. Проблемы с SSL

```
# Проверка сертификатов
sudo certbot certificates

# Обновление сертификатов
sudo certbot renew --dry-run
```

Полезные команды

```
# Просмотр логов в реальном времени
tail -f /var/log/baimuras/app.log

# Перезапуск всех сервисов
sudo systemctl restart baimuras baimuras-worker baimuras-scheduler nginx

# Проверка портов
sudo netstat -tlnp | grep :5000

# Проверка процессов
ps aux | grep baimuras
```

Контакты поддержки

Техническая поддержка:

- Email: admin@baimuras.space
- GitHub Issues: <https://github.com/ardakchapaev/baimuras.space/issues>

Экстренная поддержка:

- Telegram: [@baimuras_support](https://t.me/baimuras_support)
- Телефон: +7 777 123 45 67

Последнее обновление: 21 июня 2025 г.

Версия руководства: 2.0.0