

Инструкция по деплою на Timeweb

Предварительная подготовка

1. Создание сервера на Timeweb

1. Войдите в личный кабинет Timeweb
2. Перейдите в раздел **Облачные серверы** → **Создать сервер**
3. Выберите конфигурацию:
 - **ОС:** Ubuntu 22.04 LTS
 - **Конфигурация:** минимум 1 CPU, 1 GB RAM, 10 GB SSD
 - **Рекомендуется:** 2 CPU, 2 GB RAM, 20 GB SSD
4. Создайте сервер и дождитесь его запуска
5. Сохраните:
 - IP адрес сервера
 - Пароль root (или SSH ключ)

2. Получение настроек SMTP для mail.ru

1. Войдите в почту на mail.ru
2. Перейдите в **Настройки** → **Пароль и безопасность**
3. Найдите раздел **Пароли для внешних приложений**
4. Создайте новый пароль приложения с названием "Telegram Bot"
5. **Сохраните этот пароль** - он понадобится для настройки бота

3. Получение ID вашего Telegram

Для получения вашего Telegram ID:

```
# Запустите бота @userinfobot в Telegram
# Или используйте @getidsbot
# Они покажут ваш User ID
```

4. Получение ID целевого чата

Если бот должен отправлять сообщения в группу:

1. Добавьте бота в группу
2. Отправьте сообщение в группу
3. Перейдите по ссылке: `https://api.telegram.org/bot<BOT_TOKEN>/getUpdates`
4. Найдите `"chat":{"id":-XXXXXXXXX}` - это ID группы (с минусом!)

Деплой на сервер

Шаг 1: Подключение к серверу

```
ssh root@YOUR_SERVER_IP
```

Шаг 2: Обновление системы

```
apt update && apt upgrade -y
```

Шаг 3: Установка Docker

```
# Установка Docker
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh

# Установка Docker Compose
apt install docker-compose -y

# Проверка установки
docker --version
docker-compose --version
```

Шаг 4: Клонирование репозитория

```
# Установка git (если еще не установлен)
apt install git -y

# Клонирование репозитория
git clone https://github.com/sakharchukrv/telegram-travel-bot.git
cd telegram-travel-bot
```

Шаг 5: Настройка переменных окружения

```
# Копируем пример конфигурации
cp .env.example .env

# Редактируем конфигурацию
nano .env
```

Заполните следующие переменные:

```
# Telegram Bot Configuration
TELEGRAM_BOT_TOKEN=6816837353:AAExXsenYuwdl-ywDQywt_D7vMZBqHsYgr8
TELEGRAM_ADMIN_IDS=YOUR_TELEGRAM_ID # Ваш Telegram ID
TARGET_CHAT_ID=-1923544479 # ID чата для отправки заявок

# Database Configuration (для Docker не меняйте!)
DATABASE_URL=postgresql://postgres:postgres@db:5432/travel_bot

# SMTP Configuration (mail.ru)
SMTP_HOST=smtp.mail.ru
SMTP_PORT=465
SMTP_USER=your_email@mail.ru # Ваш email на mail.ru
SMTP_PASSWORD=your_app_password # Пароль приложения (из шага 2)
SMTP_FROM=your_email@mail.ru # Тот же email
SMTP_TLS=true

# Email Configuration
EMAIL_TO_OVERRIDE=srv@cspto.ru # Email получателя заявок

# Application Settings
DEBUG=false
LOG_LEVEL=INFO
```

Сохраните файл: `Ctrl+X`, затем `Y`, затем `Enter`

Шаг 6: Запуск контейнеров

```
# Запуск в фоновом режиме
docker-compose up -d

# Проверка статуса
docker-compose ps
```

Вы должны увидеть два контейнера:

- `travel_bot_db` (PostgreSQL)
- `travel_bot` (сам бот)

Шаг 7: Проверка логов

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

# Просмотр логов базы данных
docker-compose logs -f db

# Выход из логов: Ctrl+C
```

Вы должны увидеть сообщение: “Бот успешно запущен”

Шаг 8: Проверка работы бота

1. Откройте Telegram
2. Найдите бота: `@csp72_bot`
3. Отправьте команду `/start`
4. Если вы админ - увидите главное меню
5. Если нет - получите сообщение об ожидании одобрения

Управление ботом

Остановка бота

```
cd telegram-travel-bot
docker-compose stop
```

Запуск бота

```
cd telegram-travel-bot
docker-compose start
```

Перезапуск бота

```
cd telegram-travel-bot
docker-compose restart
```

Полная остановка и удаление контейнеров

```
cd telegram-travel-bot
docker-compose down
```

Обновление бота (после изменений в коде)

```
cd telegram-travel-bot

# Получить последние изменения
git pull

# Пересобрать и перезапустить
docker-compose down
docker-compose up -d --build
```

Настройка автозапуска

Docker Compose уже настроен на автозапуск (`restart: unless-stopped`).
Бот будет автоматически запускаться после перезагрузки сервера.

Резервное копирование базы данных

Создание бэкапа

```
# Создайте директорию для бэкапов
mkdir -p ~/backups

# Создайте бэкап
docker exec travel_bot_db pg_dump -U postgres travel_bot > ~/backups/backup_$(date +%Y%m%d_%H%M%S).sql
```

Восстановление из бэкапа

```
# Восстановление из файла
cat ~/backups/backup_YYYYMMDD_HHMMSS.sql | docker exec -i travel_bot_db psql -U postgres travel_bot
```

Автоматическое резервное копирование (cron)

```
# Создайте скрипт бэкапа
cat > /root/backup_bot.sh << 'EOF'
#!/bin/bash
BACKUP_DIR="/root/backups"
mkdir -p $BACKUP_DIR
docker exec travel_bot_db pg_dump -U postgres travel_bot > $BACKUP_DIR/backup_$(date +%Y%m%d_%H%M%S).sql
# Удаление бэкапов старше 7 дней
find $BACKUP_DIR -name "backup_*.sql" -mtime +7 -delete
EOF

# Сделайте скрипт исполняемым
chmod +x /root/backup_bot.sh

# Добавьте в cron (ежедневно в 3:00)
crontab -e
# Добавьте строку:
0 3 * * * /root/backup_bot.sh
```

Мониторинг

Просмотр использования ресурсов

```
# Просмотр использования ресурсов контейнерами
docker stats

# Просмотр использования диска
df -h

# Просмотр использования памяти
free -h
```

Проверка состояния контейнеров

```
cd telegram-travel-bot
docker-compose ps
```

Решение проблем

Бот не запускается

1. Проверьте логи:

```
bash
```

```
docker-compose logs bot
```

2. Проверьте переменные окружения в `.env`

3. Проверьте, что PostgreSQL запущен:

```
bash
```

```
docker-compose ps db
```

Бот не отправляет email

1. Проверьте настройки SMTP в `.env`
2. Убедитесь, что используете пароль приложения (не основной пароль mail.ru)
3. Проверьте логи бота на наличие ошибок SMTP

Бот не отправляет сообщения в Telegram группу

1. Убедитесь, что бот добавлен в группу
2. Проверьте, что `TARGET_CHAT_ID` правильный (с минусом для групп)
3. Проверьте, что у бота есть права на отправку сообщений в группе

База данных не доступна

1. Проверьте, что контейнер БД запущен:

```
bash
```

```
docker-compose ps db
```

2. Проверьте логи БД:

```
bash
```

```
docker-compose logs db
```

3. Попробуйте перезапустить контейнеры:

```
bash
```

```
docker-compose restart
```

Обновление до последней версии

```
cd telegram-travel-bot
git pull
docker-compose down
docker-compose up -d --build
```

Безопасность

Настройка файрвола (ufw)

```
# Установка ufw
apt install ufw -y

# Разрешить SSH
ufw allow 22/tcp

# Разрешить PostgreSQL только локально (опционально)
# ufw allow from 127.0.0.1 to any port 5432

# Включить файрвол
ufw enable

# Проверить статус
ufw status
```

Изменение порта SSH (рекомендуется)

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

# Найдите строку #Port 22 и измените на:
Port 2222

# Перезапустите SSH
systemctl restart sshd

# Разрешите новый порт в файрволе
ufw allow 2222/tcp
ufw delete allow 22/tcp
```

После этого подключайтесь так: `ssh -p 2222 root@YOUR_SERVER_IP`

Полезные команды

```
# Просмотр всех контейнеров
docker ps -a

# Просмотр образов
docker images

# Очистка неиспользуемых ресурсов
docker system prune -a

# Просмотр логов последних 100 строк
docker-compose logs --tail=100 bot

# Вход в контейнер бота
docker exec -it travel_bot bash

# Вход в PostgreSQL
docker exec -it travel_bot_db psql -U postgres travel_bot
```

Контакты и поддержка

При возникновении проблем проверьте:

1. Логи бота: `docker-compose logs bot`
2. Переменные окружения в `.env`
3. Статус контейнеров: `docker-compose ps`

Успешного деплоя! 🚀