

Руководство по развертыванию Telegram бота

Это руководство поможет вам развернуть Telegram бота для продажи термопанелей с интеграцией Abacus.AI на различных платформах хостинга для круглосуточной автономной работы.

Содержание

1. [Подготовка к развертыванию](#)
2. [Railway.app](#) (Рекомендуется для начинающих)
3. [Render.com](#)
4. [VPS](#) (DigitalOcean, AWS, и другие)
5. [Проверка работы бота](#)
6. [Мониторинг и отладка](#)
7. [Часто задаваемые вопросы](#)

Подготовка к развертыванию

Необходимые данные

Перед началом убедитесь, что у вас есть:

1. **Telegram Bot Token** - токен вашего бота от @BotFather
2. **Abacus.AI API Key** - ключ API от Abacus.AI

Текущие значения (уже настроены)





```
TELEGRAM_BOT_TOKEN=8063298485:AAHWZ0o3YhtoD_e0vtteXL8x_oqYsjXkYl8
ABACUS_DEPLOYMENT_ID=7c388e8dc
ABACUS_DEPLOYMENT_TOKEN=7ee99cc13aff41c7b00d1b6d7bb45bd8
```

 **Важно:** Вам нужно будет только добавить ваш `ABACUS_API_KEY`

Railway.app (Рекомендуется для начинающих)

Railway.app - это простая платформа для развертывания с бесплатным тарифом (\$5 кредитов в месяц).

Преимущества

-  Простота использования
-  Бесплатный тариф (\$5/месяц кредитов)
-  Автоматическое развертывание из GitHub
-  Встроенный мониторинг и логи

-  Поддержка Docker

Шаг 1: Создание аккаунта

1. Перейдите на railway.app (<https://railway.app>)
2. Нажмите “Start a New Project”
3. Войдите через GitHub (рекомендуется)

Шаг 2: Подготовка GitHub репозитория

Вариант А: Создание нового репозитория (рекомендуется)

1. Перейдите на [GitHub](https://github.com) (<https://github.com>) и войдите в свой аккаунт
2. Создайте новый репозиторий:
 - Нажмите кнопку “+” в правом верхнем углу → “New repository”
 - Введите имя: `telegram-thermopanel-bot`
 - Выберите “Private” для приватного репозитория
 - Не добавляйте README, .gitignore или license (у нас уже есть эти файлы)
 - Нажмите “Create repository”
3. Загрузите код в репозиторий (выполните в терминале):

```
cd ~/telegram_thermopanel_bot

# Если git еще не инициализирован
git init
git add .
git commit -m "Initial commit - production ready bot"

# Добавьте ваш GitHub репозиторий
git remote add origin https://github.com/ваш-username/telegram-thermopanel-bot.git

# Отправьте код
git branch -M main
git push -u origin main
```

Вариант В: Использование существующего репозитория

Если у вас уже есть репозиторий, просто отправьте обновления:

```
cd ~/telegram_thermopanel_bot
git add .
git commit -m "Update for production deployment"
git push
```

Шаг 3: Развертывание на Railway

1. На Railway нажмите “New Project”
2. Выберите “Deploy from GitHub repo”
3. Выберите ваш репозиторий `telegram-thermopanel-bot`
4. Railway автоматически обнаружит Dockerfile и начнет развертывание

Шаг 4: Настройка переменных окружения

1. В панели Railway выберите ваш проект
2. Перейдите на вкладку “Variables”
3. Добавьте следующие переменные:


```
TELEGRAM_BOT_TOKEN=8063298485:AAHWZ0o3YhtoD_e0vtteXL8x_oqYsjXkYl8
ABACUS_API_KEY=ваш_api_key_здесь
ABACUS_DEPLOYMENT_ID=7c388e8dc
ABACUS_DEPLOYMENT_TOKEN=7ee99cc13aff41c7b00d1b6d7bb45bd8
LOG_LEVEL=INFO
HEALTH_CHECK_PORT=8080
```

1. Нажмите “Add” для каждой переменной

Шаг 5: Настройка Health Check (опционально)

1. Перейдите на вкладку “Settings”
2. В разделе “Health Check” включите опцию
3. Установите путь: `/health`
4. Интервал: 30 секунд

Шаг 6: Перезапуск и проверка

1. После добавления переменных Railway автоматически перезапустит бот
2. Перейдите на вкладку “Deployments” → “View Logs” для проверки
3. Вы должны увидеть сообщение: “ Бот успешно запущен и готов к работе!”

Получение доступа к логам





```
# В Railway перейдите на вкладку "Deployments"
# Выберите последнее развертывание
# Нажмите "View Logs"
```





Render.com

Render.com предлагает бесплатный тариф для веб-сервисов с некоторыми ограничениями.

Преимущества

-  Бесплатный тариф
-  Автоматическое развертывание из GitHub
-  Простая настройка
-  Встроенный SSL

Ограничения бесплатного тарифа

-  Сервис засыпает после 15 минут неактивности
-  750 часов работы в месяц

Шаг 1: Создание аккаунта

1. Перейдите на render.com (<https://render.com>)
2. Нажмите “Get Started”
3. Войдите через GitHub

Шаг 2: Подготовка репозитория

Используйте тот же GitHub репозиторий, что и для Railway (см. выше).

Шаг 3: Создание веб-сервиса

1. В панели Render нажмите “New +”
2. Выберите “Web Service”
3. Подключите ваш GitHub аккаунт (если еще не подключен)
4. Выберите репозиторий `telegram-thermopanel-bot`
5. Заполните настройки:
 - **Name:** `thermopanel-bot` (или любое другое имя)
 - **Environment:** `Docker`
 - **Region:** выберите ближайший регион
 - **Branch:** `main`
 - **Plan:** выберите “Free”

Шаг 4: Настройка переменных окружения

В разделе “Environment Variables” добавьте:


```
TELEGRAM_BOT_TOKEN=8063298485:AAHWZ0o3YhtoD_e0vtteXL8x_oqYsjXkYl8
ABACUS_API_KEY=ваш_api_key_здесь
ABACUS_DEPLOYMENT_ID=7c388e8dc
ABACUS_DEPLOYMENT_TOKEN=7ee99cc13aff41c7b00d1b6d7bb45bd8
LOG_LEVEL=INFO
HEALTH_CHECK_PORT=8080
```

Шаг 5: Настройка Health Check

В разделе “Health Check”:

- **Health Check Path:** `/health`
- **Health Check Interval:** 30 seconds

Шаг 6: Развертывание

1. Нажмите “Create Web Service”
2. Render начнет автоматическое развертывание
3. Дождитесь завершения (обычно 2-5 минут)
4. Проверьте логи на наличие сообщения: “ Бот успешно запущен и готов к работе!”

Важно для бесплатного тарифа

На бесплатном тарифе Render сервис засыпает после 15 минут неактивности. Чтобы это решить:

Вариант 1: Использовать платный тариф (\$7/месяц)

- Бот будет работать 24/7 без ограничений

Вариант 2: Настроить внешний ping сервис (для бесплатного тарифа)

- Используйте сервис вроде [UptimeRobot](https://uptimerobot.com) (<https://uptimerobot.com>) для пинга вашего health check endpoint каждые 5 минут
- URL для пинга: `https://ваш-сервис.onrender.com/health`

VPS (DigitalOcean, AWS, и другие)

Развертывание на VPS дает полный контроль над сервером и подходит для продакшн-окружения.

Рекомендуемые провайдеры

- **DigitalOcean** - от \$4/месяц (рекомендуется для начинающих)
- **AWS EC2** - гибкие тарифы
- **Hetzner** - от €3/месяц
- **Vultr** - от \$2.50/месяц
- **Linode** - от \$5/месяц

Минимальные требования

- **OS:** Ubuntu 20.04+ или Debian 11+
- **RAM:** 512 MB (рекомендуется 1 GB)
- **CPU:** 1 vCore
- **Диск:** 10 GB

Вариант А: Развертывание с Docker (Рекомендуется)

Шаг 1: Подключение к серверу

```
# Замените IP_адрес на IP вашего сервера
ssh root@IP_адрес
```

Шаг 2: Установка Docker

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

# Установка необходимых пакетов
sudo apt install -y apt-transport-https ca-certificates curl software-properties-
common

# Добавление Docker GPG ключа
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

# Добавление Docker репозитория
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(l
sb_release -cs) stable"

# Установка Docker
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io

# Установка Docker Compose
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-com-
pose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

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

Шаг 3: Клонирование репозитория

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

# Создание директории для бота
mkdir -p /opt/bots
cd /opt/bots

# Клонирование репозитория
git clone https://github.com/ваш-username/telegram-thermopanel-bot.git
cd telegram-thermopanel-bot
```

Шаг 4: Настройка переменных окружения

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

Вставьте следующее содержимое:

```
TELEGRAM_BOT_TOKEN=8063298485:AAHWZ0o3YhtoD_e0vtteXL8x_oqYsjXkYl8
ABACUS_API_KEY=ваш_api_key_здесь
ABACUS_DEPLOYMENT_ID=7c388e8dc
ABACUS_DEPLOYMENT_TOKEN=7ee99cc13aff41c7b00d1b6d7bb45bd8
LOG_LEVEL=INFO
HEALTH_CHECK_PORT=8080
```

Сохраните файл (Ctrl+X, затем Y, затем Enter)

Шаг 5: Запуск бота

```
# Сборка и запуск контейнера
docker-compose up -d

# Проверка статуса
docker-compose ps

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

Шаг 6: Настройка автозапуска

Docker Compose с опцией `restart: unless-stopped` автоматически перезапустит бота при перезагрузке сервера.

Для дополнительной надежности настройте systemd service:

```
# Создание systemd service
sudo nano /etc/systemd/system/thermopanel-bot.service
```

Вставьте:

```
[Unit]
Description=Thermopanel Telegram Bot
Requires=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/bots/telegram-thermopanel-bot
ExecStart=/usr/local/bin/docker-compose up -d
ExecStop=/usr/local/bin/docker-compose down
TimeoutStartSec=0

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

Сохраните и активируйте:

```
sudo systemctl daemon-reload
sudo systemctl enable thermopanel-bot
sudo systemctl start thermopanel-bot
sudo systemctl status thermopanel-bot
```

Вариант B: Развертывание без Docker

Если вы предпочитаете не использовать Docker:

Шаг 1: Установка Python и зависимостей

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

# Установка Python 3.11 и pip
sudo apt install -y python3.11 python3.11-venv python3-pip git

# Создание пользователя для бота
sudo useradd -r -s /bin/bash -m botuser
```

Шаг 2: Установка бота

```
# Переключение на пользователя botuser
sudo su - botuser

# Клонирование репозитория
git clone https://github.com/ваш-username/telegram-thermopanel-bot.git
cd telegram-thermopanel-bot

# Создание виртуального окружения
python3.11 -m venv venv

# Активация виртуального окружения
source venv/bin/activate

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

Шаг 3: Настройка переменных окружения

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

Вставьте переменные (как в Варианте А)

Шаг 4: Создание systemd service

Выход из пользователя botuser:

```
exit
```

Создание service файла:

```
sudo nano /etc/systemd/system/thermopanel-bot.service
```

Вставьте:


```
[Unit]
Description=Thermopanel Telegram Bot
After=network.target

[Service]
Type=simple
User=botuser
WorkingDirectory=/home/botuser/telegram-thermopanel-bot
Environment="PATH=/home/botuser/telegram-thermopanel-bot/venv/bin"
ExecStart=/home/botuser/telegram-thermopanel-bot/venv/bin/python bot.py
Restart=always
RestartSec=10

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

Сохраните и активируйте:

```
sudo systemctl daemon-reload
sudo systemctl enable thermopanel-bot
sudo systemctl start thermopanel-bot
sudo systemctl status thermopanel-bot
```

Шаг 5: Просмотр логов

```
# Просмотр системных логов
sudo journalctl -u thermopanel-bot -f

# Или просмотр файла логов бота
sudo tail -f /home/botuser/telegram-thermopanel-bot/bot.log
```

✓ Проверка работы бота

1. Проверка через Telegram

1. Откройте Telegram
2. Найдите вашего бота по username
3. Отправьте команду `/start`
4. Бот должен ответить приветственным сообщением
5. Задайте любой вопрос о термопанелях

2. Проверка Health Check

Проверьте health check endpoint:

```
# Для Railway/Render
curl https://ваш-сервис-url.railway.app/health

# Для VPS
curl http://IP_адрес:8080/health
```

Ожидаемый ответ:

```
{
  "status": "healthy",
  "uptime_seconds": 3600.5,
  "total_messages": 42,
  "errors_count": 0,
  "last_message_time": "2025-10-13T10:30:00.123456"
}
```



Мониторинг и отладка

Railway.app

Просмотр логов:

1. Перейдите в проект на Railway
2. Выберите "Deployments"
3. Нажмите "View Logs"

Перезапуск:

1. Settings → "Restart"

Render.com

Просмотр логов:

1. Перейдите в сервис
2. Вкладка "Logs"

Перезапуск:

1. Manual Deploy → "Deploy latest commit"

VPS

Просмотр логов (Docker):

```
# Логи контейнера
docker-compose logs -f

# Последние 100 строк
docker-compose logs --tail=100

# Логи с определенного времени
docker-compose logs --since 30m
```

Просмотр логов (без Docker):

```
# Системные логи
sudo journalctl -u thermopanel-bot -f

# Файл логов бота
sudo tail -f /home/botuser/telegram-thermopanel-bot/bot.log
```

Перезапуск бота:

```
# С Docker
docker-compose restart

# Без Docker
sudo systemctl restart thermopanel-bot
```

Проверка статуса:

```
# С Docker
docker-compose ps

# Без Docker
sudo systemctl status thermopanel-bot
```

Обновление бота:

```
# Переход в директорию
cd /opt/bots/telegram-thermopanel-bot # для Docker
# или
cd /home/botuser/telegram-thermopanel-bot # без Docker

# Получение обновлений
git pull

# С Docker
docker-compose down
docker-compose build
docker-compose up -d

# Без Docker
sudo systemctl restart thermopanel-bot
```

? Часто задаваемые вопросы

1. Бот не отвечает на сообщения

Проверьте:

- Правильность токена бота (`TELEGRAM_BOT_TOKEN`)
- Правильность API ключа Abacus.AI (`ABACUS_API_KEY`)
- Логи на наличие ошибок
- Health check endpoint возвращает status "healthy"

Решение:

```
# Проверьте логи
docker-compose logs --tail=50 # для Docker
# или
sudo journalctl -u thermopanel-bot --tail=50 # без Docker

# Проверьте переменные окружения
docker-compose exec telegram-bot env | grep TELEGRAM # для Docker
```

2. Ошибка “TELEGRAM_BOT_TOKEN не установлен”

Причина: Переменная окружения не настроена

Решение:

- Проверьте файл `.env` на наличие правильного токена
- Убедитесь, что в `docker-compose.yml` или настройках платформы переменные указаны правильно
- Перезапустите бот после изменения переменных

3. Бот работает, но не отвечает корректно

Причина: Проблема с Abacus.AI API

Решение:

- Проверьте правильность `ABACUS_API_KEY`
- Проверьте логи на ошибки связи с Abacus.AI
- Убедитесь, что deployment ID и token правильные

4. Контейнер постоянно перезапускается (Docker)

Проверка:

```
docker-compose ps
docker-compose logs --tail=100
```

Частые причины:

- Неправильные переменные окружения
- Недостаточно памяти на сервере
- Ошибка в коде (проверьте логи)

5. Как изменить токен бота?

Railway/Render:

1. Перейдите в настройки проекта
2. Измените переменную `TELEGRAM_BOT_TOKEN`
3. Сервис автоматически перезапустится

VPS:

```
# Отредактируйте .env файл
nano .env

# Перезапустите бота
docker-compose restart # для Docker
# или
sudo systemctl restart thermopanel-bot # без Docker
```

6. Как посмотреть использование ресурсов?

Docker:

```
docker stats
```

Системные ресурсы:

```
htop # или top
```

7. Бот работает медленно

Возможные причины:

- Недостаточно ресурсов сервера (увеличьте RAM/CPU)
- Медленный ответ от Abacus.AI API
- Проблемы с сетью

Решение:

- Проверьте использование ресурсов
- Проверьте логи на медленные запросы
- Рассмотрите обновление плана хостинга

8. Как сделать backup данных?

Логи:

```
# С Docker
docker-compose logs > backup-logs-$(date +%Y%m%d).txt

# Без Docker
sudo journalctl -u thermopanel-bot > backup-logs-$(date +%Y%m%d).txt
```

Конфигурация:

```
# Сохранение .env и конфигурации
tar -czf bot-backup-$(date +%Y%m%d).tar.gz .env docker-compose.yml
```

9. Как настроить несколько ботов?

Для каждого бота:

1. Создайте отдельную директорию
2. Используйте свой `TELEGRAM_BOT_TOKEN`
3. Измените `HEALTH_CHECK_PORT` для каждого бота (8080, 8081, 8082, и т.д.)

10. Бот не запускается на Render (бесплатный план)

Причина: На бесплатном плане Render сервис засыпает после 15 минут неактивности

Решения:

1. Перейдите на платный план (\$7/месяц)
2. Используйте UptimeRobot для пинга health check endpoint каждые 5 минут
3. Используйте Railway или VPS вместо Render



Получение помощи

Если у вас возникли проблемы:

1. **Проверьте логи** - большинство проблем видны в логах
2. **Проверьте health check** - он покажет статус бота

3. Проверьте документацию платформы:

- [Railway Docs](https://docs.railway.app/) (<https://docs.railway.app/>)
- [Render Docs](https://render.com/docs) (<https://render.com/docs>)

4. Проверьте переменные окружения - частая причина проблем

Готово!

Ваш бот теперь работает 24/7 автономно!

Следующие шаги:

- Настройте мониторинг для отслеживания работы бота
- Регулярно проверяйте логи на наличие ошибок
- Сделайте backup конфигурации
- Протестируйте все функции бота

Удачи с вашим ботом! 🚀