


# Сценарий видео-инструкции по развертыванию WOWsilizing Bot

 **Назначение:** Текстовый сценарий для записи видео-инструкции

 **Продолжительность:** Примерно 20-25 минут

 **Цель:** Показать полный процесс развертывания бота от А до Я

## Структура видео

Сегмент	Тема	Время
0:00-1:00	Вступление	1 мин
1:00-3:00	Создание Telegram бота	2 мин
3:00-6:00	Регистрация на GitHub	3 мин
6:00-11:00	Загрузка кода на GitHub	5 мин
11:00-15:00	Регистрация на Railway	4 мин
15:00-20:00	Развертывание и настройка	5 мин
20:00-23:00	Тестирование	3 мин
23:00-25:00	Заключение	2 мин

**Общая продолжительность:** 25 минут

## СЦЕНАРИЙ

### СЕГМЕНТ 1: ВСТУПЛЕНИЕ (0:00-1:00)

**На экране:** Заставка с названием видео

**Текст диктора:**

“Привет! В этом видео я покажу вам, как развернуть Telegram бота для обработки видео на Railway через GitHub.

Это полная инструкция от нуля до рабочего бота. Я покажу:

- Как создать бота в Telegram

- Как загрузить код на GitHub
- Как развернуть на Railway

Все это займет примерно 60-90 минут вашего времени.

Стоимость хостинга - примерно 5-7 долларов в месяц. Railway дает 5 долларов бесплатно для начала.

Начинаем!"

#### **На экране:**

- Плавный переход к рабочему столу
- Показать открытые вкладки: Telegram, GitHub, Railway



## **СЕГМЕНТ 2: СОЗДАНИЕ TELEGRAM БОТА (1:00-3:00)**

**На экране:** Открыть Telegram

#### **Текст диктора:**

"Первый шаг - создать бота в Telegram.

Открываем Telegram и в поиске набираем 'BotFather'.

Это официальный бот Telegram для создания других ботов. У него есть синяя галочка верификации.

Открываем чат с BotFather."

#### **Действия на экране:**

1. Открыть Telegram
2. Набрать в поиске "BotFather"
3. Нажать на первый результат
4. Показать синюю галочку крупным планом

#### **Текст диктора:**

"Теперь отправляем команду /newbot

BotFather спросит название бота. Я назову его 'Мой Видео Бот'.

Затем он попросит username - это должно быть на английском и заканчиваться на 'bot'.

Я ввожу 'my\_video\_editor\_bot'."

#### **Действия на экране:**

1. Набрать "/newbot"
2. Отправить
3. Набрать "Мой Видео Бот"
4. Отправить
5. Набрать "my\_video\_editor\_bot"
6. Отправить

#### **Текст диктора:**

“Отлично! BotFather создал бота и отправил нам токен.

Это очень важно - токен нужно скопировать и сохранить в безопасном месте.

НИКОМУ НЕ ПОКАЗЫВАЙТЕ ЕГО!

Я скопирую его и сохраню в текстовый файл.”

**Действия на экране:**

1. Выделить токен (1234567890:ABCdef...)
2. Скопировать (Ctrl+C)
3. Открыть блокнот
4. Вставить токен
5. Сохранить файл

**Текст диктора:**

“Первый шаг готов! Бот создан, токен получен.”



### **СЕГМЕНТ 3: РЕГИСТРАЦИЯ НА GITHUB (3:00-6:00)**

**На экране:** Открыть браузер

**Текст диктора:**

“Теперь нам нужен GitHub - это сервис для хранения кода.

Переходим на [github.com](https://github.com)

Если у вас уже есть аккаунт - просто войдите. Если нет - сейчас создадим.

Нажимаем ‘Sign up’.”

**Действия на экране:**

1. Открыть браузер
2. Перейти на <https://github.com>
3. Нажать “Sign up”

**Текст диктора:**

“Вводим email - используйте реальный, туда придет письмо для подтверждения.

Придумываем пароль - минимум 8 символов с буквами и цифрами.

Придумываем username - это будет ваш адрес на GitHub. Например, ‘ivan-video-2024’.

Проходим капчу и нажимаем ‘Create account’.”

**Действия на экране:**

1. Ввести email
2. Ввести пароль
3. Ввести username
4. Пройти капчу
5. Нажать “Create account”

**Текст диктора:**

“Теперь нужно подтвердить email.

Открываем почту, находим письмо от GitHub и нажимаем ‘Verify email address’.

Отлично! Мы вошли в GitHub.

Выбираем бесплатный план - Free. Его более чем достаточно.”

#### **Действия на экране:**

1. Открыть почту (в новой вкладке)
2. Найти письмо от GitHub
3. Нажать “Verify email address”
4. Вернуться на вкладку с GitHub
5. Выбрать “Free”
6. Нажать “Continue”

#### **Текст диктора:**

“Второй шаг готов! GitHub аккаунт создан.”



## **СЕГМЕНТ 4: ЗАГРУЗКА КОДА НА GITHUB (6:00-11:00)**

**На экране:** GitHub главная страница

#### **Текст диктора:**

“Теперь нужно создать репозиторий - это место, где будет храниться код нашего бота.

Нажимаем зеленую кнопку ‘New’ слева.

Вводим название репозитория - я назову ‘wowsilizing-bot’.

Можно добавить описание - ‘Telegram bot for video processing’.

Важно: выбираем ‘Public’ - это проще для Railway.

И самое главное - НЕ ставим галочки ‘Add README’, ‘.gitignore’ или ‘license’. У нас уже есть эти файлы.

Нажимаем ‘Create repository’.”

#### **Действия на экране:**

1. Нажать “New”
2. Ввести “wowsilizing-bot”
3. Ввести описание
4. Выбрать “Public”
5. Показать, что галочки НЕ стоят
6. Нажать “Create repository”

#### **Текст диктора:**

“Отлично! Репозиторий создан.

Теперь нужно загрузить файлы бота.

Я покажу самый простой способ - через веб-интерфейс.

Находим текст 'uploading an existing file' и нажимаем на него.

Теперь открываем папку с файлами бота на компьютере.

Выделяем все файлы - нажимаем Ctrl+A."

#### **Действия на экране:**

1. Нажать на "uploading an existing file"
2. Открыть проводник (в новом окне)
3. Перейти в папку с ботом
4. Нажать Ctrl+A
5. Показать выделенные файлы

#### **Текст диктора:**

"Теперь перетаскиваем их в окно браузера с GitHub.

Ждем, пока все файлы загрузятся. Это может занять 1-2 минуты.

Видим прогресс-бар внизу.

Отлично! Все файлы загрузились.

Теперь внизу в поле 'Commit changes' пишем: 'Первая загрузка бота'.

И нажимаем зеленую кнопку 'Commit changes'."

#### **Действия на экране:**

1. Перетащить файлы в браузер
2. Показать прогресс загрузки (ускоренно)
3. Прокрутить вниз
4. Ввести "Первая загрузка бота"
5. Нажать "Commit changes"

#### **Текст диктора:**

"Отлично! Код загружен на GitHub.

Видим все наши файлы: bot.py, config.py, requirements.txt и другие.

Важно: проверьте, что файла .env здесь НЕТ. Должен быть только .env.example.

Третий шаг готов!"

#### **Действия на экране:**

1. Показать список файлов на GitHub
2. Прокрутить, показать .env.example
3. Показать, что .env отсутствует



## **СЕГМЕНТ 5: РЕГИСТРАЦИЯ НА RAILWAY (11:00-15:00)**

**На экране:** Открыть новую вкладку

#### **Текст диктора:**

"Теперь переходим к Railway - это платформа, где будет работать наш бот.

Переходим на railway.app

Нажимаем 'Login'.

Выбираем 'Login with GitHub' - это самый простой способ.

GitHub спросит, разрешить ли Railway доступ. Нажимаем 'Authorize Railway'."

**Действия на экране:**

1. Открыть новую вкладку
2. Перейти на <https://railway.app>
3. Нажать "Login"
4. Нажать "Login with GitHub"
5. Нажать "Authorize Railway"

**Текст диктора:**

"Отлично! Мы вошли в Railway.

Теперь Railway попросит добавить способ оплаты. Это нужно для верификации, но вы получите 5 долларов бесплатно.

Нажимаем 'Start Trial' или 'Add Payment Method'.

Выбираем 'Credit/Debit Card'.

Вводим данные карты: номер, срок действия, CVV код, имя держателя.

Также страну и почтовый индекс."

**Действия на экране:**

1. Нажать "Start Trial"
2. Выбрать "Credit/Debit Card"
3. Ввести данные карты (заблорить номер)
4. Ввести адрес
5. Нажать "Add Card"

**Текст диктора:**

"Отлично! Карта добавлена.

Railway проверит карту - спишется и вернется один доллар.

И вы получите 5 долларов бесплатных кредитов!

Четвертый шаг готов!"



## **СЕГМЕНТ 6: РАЗВЕРТЫВАНИЕ И НАСТРОЙКА (15:00-20:00)**

**На экране:** Railway Dashboard

**Текст диктора:**

"Теперь самая важная часть - развертывание бота.

Нажимаем большую фиолетовую кнопку 'New Project'.

Выбираем 'Deploy from GitHub repo'.

Railway попросит доступ к репозиториям GitHub.

Нажимаем 'Configure GitHub App'."

**Действия на экране:**

1. Нажать "+ New Project"
2. Нажать "Deploy from GitHub repo"
3. Нажать "Configure GitHub App"

**Текст диктора:**

"Мы перешли на страницу GitHub.

Здесь нужно выбрать, к каким репозиториям дать доступ.

Я рекомендую выбрать 'Only select repositories' и выбрать только наш репозиторий 'wowsilizing-bot'.

Это безопаснее.

Нажимаем на выпадающее меню, выбираем наш репозиторий, и нажимаем 'Save'."

**Действия на экране:**

1. Выбрать "Only select repositories"
2. Нажать на выпадающее меню
3. Выбрать "wowsilizing-bot"
4. Нажать "Save"

**Текст диктора:**

"Мы вернулись в Railway.

Теперь видим наш репозиторий в списке.

Нажимаем на него.

Railway покажет информацию о репозитории.

Нажимаем большую фиолетовую кнопку 'Deploy Now'."

**Действия на экране:**

1. Найти "wowsilizing-bot" в списке
2. Нажать на него
3. Нажать "Deploy Now"

**Текст диктора:**

"Отлично! Railway создал проект.

Но бот еще не запустится - нужно добавить переменные окружения.

Это САМЫЙ ВАЖНЫЙ ШАГ!

Нажимаем на карточку сервиса.

Переходим на вкладку 'Variables' вверху.

Нажимаем '+ New Variable'."

#### **Действия на экране:**

1. Нажать на карточку сервиса
2. Нажать на вкладку "Variables"
3. Нажать "+ New Variable"

#### **Текст диктора:**

"Сначала добавляем BOT\_TOKEN.

В поле 'Variable Name' вводим: BOT\_TOKEN

В поле 'Value' вставляем тот токен, который мы получили от BotFather.

Я открываю файл, где сохранил токен, копирую его и вставляю.

Нажимаем 'Add'."

#### **Действия на экране:**

1. Ввести "BOT\_TOKEN"
2. Открыть блокнот с токеном
3. Скопировать токен
4. Вставить в поле Value (заблюрить)
5. Нажать "Add"

#### **Текст диктора:**

"Теперь добавляем вторую обязательную переменную - PREMIUM\_USERNAME.

Снова '+ New Variable'.

Variable Name: PREMIUM\_USERNAME

Value: ваш username в Telegram, БЕЗ символа @.

Я ввожу свой username.

Нажимаем 'Add'.

Этот пользователь получит доступ к премиум функциям, если они настроены."

#### **Действия на экране:**

1. Нажать "+ New Variable"
2. Ввести "PREMIUM\_USERNAME"
3. Ввести свой username (без @)
4. Нажать "Add"

#### **Текст диктора:**

"Отлично! Минимальные переменные добавлены.

Если вы хотите использовать AI функции - можно добавить OPENAI\_API\_KEY, GOOGLE\_AI\_API\_KEY и другие. Но это опционально.

После добавления переменных Railway автоматически начнет развертывание.

Переходим на вкладку 'Deployments'."



**Действия на экране:**

1. Показать добавленные переменные
2. Нажать на вкладку "Deployments"

**Текст диктора:**

"Видим активное развертывание с крутящимся индикатором.

Нажимаем на него, чтобы увидеть логи.

Railway сейчас:

- Клонировать код с GitHub
- Собирает Docker образ
- Устанавливает ffmpeg и yt-dlp
- Устанавливает Python библиотеки

Это займет примерно 3-5 минут. Я ускорю видео."

**Действия на экране:**

1. Нажать на активное deployment
2. Показать логи (ускоренное воспроизведение)
3. Показать основные этапы

**Текст диктора:**

"И вот, мы видим сообщения:

- 'Bot started successfully'
- 'Started polling'

Это значит, что бот успешно запустился!

Статус изменился на зеленую точку - Active.

Пятый шаг готов!"

**СЕГМЕНТ 7: ТЕСТИРОВАНИЕ (20:00-23:00)**

**На экране:** Переключиться на Telegram

**Текст диктора:**

"Теперь самое интересное - проверим, работает ли бот!

Открываем Telegram.

Находим нашего бота - я ищу по имени 'My Video Bot'.

Видим, что рядом с именем есть зеленая точка - это означает, что бот онлайн!

Открываем чат с ботом."

**Действия на экране:**

1. Переключиться на Telegram
2. Найти бота в поиске

3. Показать зеленую точку (укрупненно)

4. Открыть чат

**Текст диктора:**

“Отправляем команду /start

И... Бот сразу отвечает!

Мы видим приветственное сообщение, меню с кнопками и инструкции.

Отлично! Бот работает!”

**Действия на экране:**

1. Набрать “/start”

2. Отправить

3. Показать ответ бота

4. Показать меню и кнопки

**Текст диктора:**

“Давайте проверим основную функцию - обработку видео.

Я отправляю короткое тестовое видео.

Бот принял видео, показал его длительность и предложил варианты обработки.

Нажимаю ‘Вырезать фрагмент’.

Бот просит ввести тайм-коды. Я ввожу ‘00:00:03-00:00:08’ - это значит вырезать с 3 по 8 секунду.

Отправляю.

Бот начинает обработку, показывает прогресс.”

**Действия на экране:**

1. Отправить видео

2. Показать ответ бота

3. Нажать “Вырезать фрагмент”

4. Ввести “00:00:03-00:00:08”

5. Отправить

6. Показать процесс обработки

**Текст диктора:**

“И вот, бот отправляет обработанное видео!

Проверяем - видео длится именно 5 секунд, как мы и хотели.

Воспроизводится корректно.

Отлично! Бот полностью работает!”

**Действия на экране:**

1. Показать отправленное видео

2. Показать длительность

3. Воспроизвести часть видео

---

## СЕГМЕНТ 8: ЗАКЛЮЧЕНИЕ (23:00-25:00)

**На экране:** Вернуться к Railway, показать статус Active

**Текст диктора:**

“Отлично! Мы успешно развернули Telegram бота на Railway!

Давайте подведем итоги.

Мы прошли следующие шаги:

1. Создали бота через BotFather
2. Зарегистрировались на GitHub
3. Загрузили код на GitHub
4. Зарегистрировались на Railway
5. Развернули бота и настроили переменные
6. Протестировали работу”

**На экране:** Показать слайд со списком шагов и галочками

**Текст диктора:**

“Теперь ваш бот:

- Работает 24/7 в облаке
- Автоматически обновляется при изменениях на GitHub
- Обработывает видео для пользователей

Стоимость - примерно 5-7 долларов в месяц.”

**На экране:** Показать слайд с преимуществами

**Текст диктора:**

“Что дальше?

1. Поделитесь ботом с друзьями
2. Мониторьте работу через Railway
3. Улучшайте и добавляйте новые функции

Все подробные инструкции вы найдете в документации:

- COMPLETE\_GUIDE.md - полное руководство
- QUICK\_REFERENCE.md - быстрая справка
- DEPLOYMENT\_CHECKLIST.md - чек-лист

Ссылки на все файлы - в описании под видео.”

**На экране:** Показать список файлов документации

**Текст диктора:**

“Спасибо за просмотр!

Если видео было полезным - поставьте лайк и подпишитесь на канал.

Если есть вопросы - пишите в комментариях!

Удачи с вашим ботом!”

**На экране:** Заключительная заставка с названием канала и ссылками

---



## Дополнительные заметки для монтажа

---

### Музыка:

- Фоновая музыка: Спокойная, ненавязчивая
- Громкость: 20-30% от голоса

### Текстовые вставки на экране:

- При упоминании команд - показывать их в рамке
- При упоминании ссылок - показывать их внизу экрана
- Важные моменты - выделять красным или желтым

### Переходы:

- Между сегментами - плавное затухание (0.5-1 сек)
- При смене окон - использовать zoom-in/zoom-out

### Выделения:

- Кнопки, на которые нужно нажать - обводить красным
- Важные поля - подсвечивать
- Курсор - увеличить или выделить

### Тайминг:

- Позволить зрителю прочитать на экране
  - Не торопиться с переключением экранов
  - Делать паузы после важных моментов
- 



## Ключевые моменты для внимания

---

1. Подчеркнуть важность безопасности BOT\_TOKEN
  2. Показать, что .env не должен быть на GitHub
  3. Объяснить, что Railway требует карту, но дает \$5 бесплатно
  4. Показать, что первое развертывание занимает время
  5. Демонстрировать реальную работу бота в конце
- 



## Ссылки для описания видео

---

### Документация:


- COMPLETE\_GUIDE.md
- GITHUB\_SETUP.md
- RAILWAY\_GITHUB\_DEPLOY.md

- DEPLOYMENT\_CHECKLIST.md
- QUICK\_REFERENCE.md

### Внешние ссылки:

- <https://github.com> - GitHub
- <https://railway.app> - Railway
- <https://t.me/BotFather> - BotFather
- <https://docs.railway.app> - Railway Docs
- <https://core.telegram.org/bots/api> - Telegram Bot API

---

 **Примечание:** Этот сценарий можно адаптировать под ваш стиль и темп. Основная цель - показать полный процесс от нуля до рабочего бота максимально просто и понятно.