

РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ

(Структура руководства по разворачиванию серверной ИС и веб-сайта)

Название документа: Руководство по разворачиванию [Название Системы]
Версия: 1.0
Дата: [Дата составления]
Ответственные: [ФИО/Отдел]

1. Титульная страница

- Содержание:** Название системы, логотип компании, номер версии документа, дата, гриф конфиденциальности (при необходимости).
- Картинки:** Логотип компании/проекта.

2. История изменений

- Содержание:** Таблица с версией документа, датой изменений, перечнем внесенных правок и автором.

- Пример:**

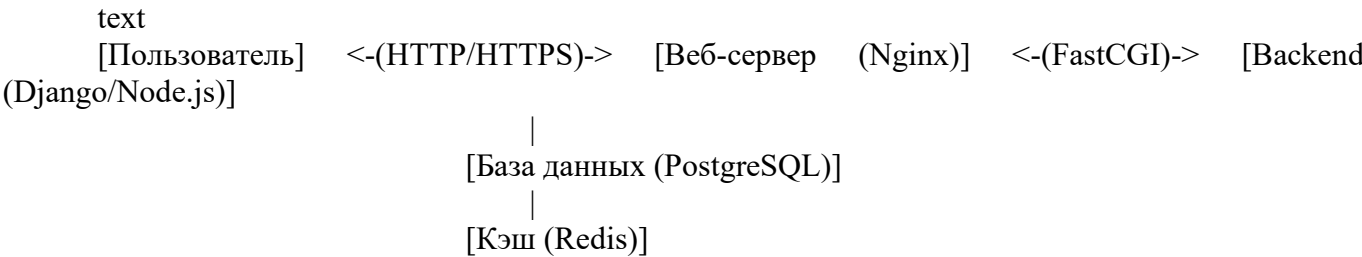
Версия	Дата	Описание изменений	Автор
1.0	2023-10-26	Первоначальная версия документа	Иванов А.С.

3. Введение

- 3.1. Назначение документа:** Краткое описание целей данного руководства (например, "Настоящий документ описывает пошаговый процесс разворачивания системы X в производственной среде").
- 3.2. Область применения:** Для кого предназначен документ (системные администраторы, DevOps-инженеры).
- 3.3. Предварительные требования:** Что должно быть уже готово (например, закуплено оборудование, настроены учетные записи, есть базовые знания о Linux).
- 3.4. Глоссарий:** Список основных терминов и аббревиатур.

4. Обзор архитектуры системы

- Что делать:** дать общее представление о всех компонентах системы, их взаимодействии и окружении.
- Содержание:**
 - Описание серверной части (Backend): API, база данных, кэш, очереди задач, файловое хранилище.
 - Описание клиентской части (Frontend): Веб-сайт, панель администратора.
 - Схема сетевой инфраструктуры (DMZ, внутренняя сеть, порты).
- Где целесообразно вставлять картинки:**
 - Обязательно: Диаграмма архитектуры системы.** Это самая важная картинка в руководстве. Она должна наглядно показывать, как компоненты связаны между собой.
 - Пример:**



5. Подготовка инфраструктуры

- 5.1. Требования к аппаратному и программному обеспечению**

- **Что делать:** Перечислить минимальные и рекомендуемые требования.
 - **Содержание:** CPU, RAM, HDD/SSD, ОС (версия ОС, например, Ubuntu 22.04 LTS), версии ПО (Python 3.11, PostgreSQL 15 и т.д.).
 - **5.2. Подготовка серверов**
 - **Что делать:** Описать шаги по настройке "чистых" серверов.
 - **Содержание:**
 1. Установка и настройка операционной системы.
 2. Создание пользователей, настройка SSH-доступа, ключей.
 3. Базовая настройка безопасности (firewall - ufw/firewalld, обновление системы).
 4. Настройка синхронизации времени (NTP).
 - **Где целесообразно вставлять картинки:**
 - Скриншоты критически важных этапов настройки (например, разделы дисков при установке ОС, настройка правил фаервола).
 - **5.3. Сетевая настройка**
 - **Что делать:** настроить сетевые интерфейсы, DNS-записи, сертификаты.
 - **Содержание:**
 1. Назначение статических IP-адресов серверам.
 2. Настройка DNS A/AAAA записей для доменного имени сайта (например, example.com и www.example.com).
 3. Получение SSL-сертификата (например, с помощью Let's Encrypt certbot).
-

6. Установка и настройка компонентов системы

(Этот раздел часто делят на подразделы по каждому компоненту)

- **6.1. Установка и настройка сервера баз данных (например, PostgreSQL)**
 - **Что делать:** Установить СУБД, создать базу данных, пользователя, настроить доступ.
 - **Содержание:**
 - `apt install postgresql`
 - `sudo -u postgres psql -c "CREATE USER myuser WITH PASSWORD 'mypassword';"`
 - `sudo -u postgres psql -c "CREATE DATABASE mydb OWNER myuser;"`
 - Правка `pg_hba.conf` для настройки доступа.
- **6.2. Установка и настройка Backend-приложения**
 - **Что делать:** Развернуть код приложения, настроить виртуальное окружение, зависимости, переменные окружения.
 - **Содержание:**
 1. Клонирование репозитория с кодом (`git clone`).
 2. Создание виртуального окружения (`python -m venv venv`).
 3. Установка зависимостей (`pip install -r requirements.txt`).
 4. Создание файла с переменными окружения (`.env`) с секретами (ключи БД, API-ключи).
 5. Запуск миграций базы данных (`python manage.py migrate`).
 6. Создание суперпользователя (`python manage.py createsuperuser`).
 7. Сбор статических файлов (`python manage.py collectstatic`).
- **6.3. Установка и настройка Frontend-приложения (веб-сайта)**
 - **Что делать:** развернуть клиентский код, собрать его для production.
 - **Содержание:**
 1. Клонирование репозитория.
 2. Установка зависимостей (`npm install`).
 3. Сборка проекта (`npm run build`).
 4. Настройка веб-сервера (Nginx) на раздачу собранных статических файлов (папка `dist` или `build`).
- **6.4. Настройка веб-сервера (например, Nginx) и проксирование**
 - **Что делать:** настроить Nginx как обратный прокси и для раздачи статики.
 - **Содержание:**

- Создание конфигурационного файла виртуального хоста (/etc/nginx/sites-available/my_site).
- Пример конфига:

```

nginx
server {
    listen 80;
    server_name example.com www.example.com;
    return 301 https://$server_name$request_uri; # Редирект на HTTPS
}

server {
    listen 443 ssl;
    server_name example.com www.example.com;

    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/private.key;

    location /static/ {
        alias /path/to/backend/static/;
    }

    location /media/ {
        alias /path/to/backend/media/;
    }

    location / {
        proxy_pass http://127.0.0.1:8000; # Прокси на бэкенд-приложение
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

- Активация конфига и перезагрузка Nginx.

• 6.5. Настройка сервис-менеджера (например, systemd)

- **Что делать:** создать systemd-юнит для автоматического запуска и управления бэкенд-приложением.
- **Содержание:**
 - Создание файла /etc/systemd/system/my_backend.service.
 - Пример содержимого:

```

ini
[Unit]
Description=My Backend Service
After=network.target

[Service]
Type=simple
User=myuser
Group=mygroup
WorkingDirectory=/path/to/backend/code
Environment="PATH=/path/to/venv/bin"
ExecStart=/path/to/venv/bin/gunicorn myproject.wsgi:application --bind 127.0.0.1:8000
Restart=on-failure

```

```

[Install]
WantedBy=multi-user.target

```

- Запуск и активация сервиса: systemctl enable --now my_backend.service.

- **Где целесообразно вставлять картинки:**

- Скриншоты ключевых конфигурационных файлов с поясняющими аннотациями (что и зачем меняется).
- Скриншот вывода команд `systemctl status my_backend.service` и `nginx -t` для демонстрации успешности выполнения.
- Скриншот структуры каталогов проекта.

7. Настройка безопасности

- **Что делать:** Применить дополнительные меры безопасности.

- **Содержание:**

- Настройка брандмауэра (открыть только порты 80, 443, 22).
- Настройка Fail2Ban для защиты от подбора паролей.
- Проверка прав доступа к файлам и папкам.
- Настройка регулярного резервного копирования.

8. Завершение развертывания и проверка

- **8.1. Финальные шаги**

- **Что делать:** Выполнить заключительные действия перед сдачей системы.
- **Содержание:**
 1. Провести первичную настройку системы через веб-интерфейс (если есть).
 2. Наполнить систему начальными данными (справочники, административные пользователи).

- **8.2. Чек-лист проверки работоспособности**

- **Что делать:** Систематически проверить, что все работает корректно.
- **Содержание:**
 - Сайт открывается по HTTPS.
 - Все страницы и стили загружаются правильно.
 - API-эндпоинты отвечают.
 - Подключение к базе данных работает.
 - Загрузка и отображение файлов (медиа) работает.
 - Фоновые задачи (если есть) выполняются.
 - Логи не содержат критических ошибок.

- **Где целесообразно вставлять картинки:**

- **Обязательно:** Скриншот главной страницы работающего сайта в браузере.
- Скриншоты работающего API (например, ответ от `/api/health` в Postman или браузере).
- Скриншоты панели администратора с созданным суперпользователем.

9. Резервное копирование и аварийное восстановление

- **Что делать:** Описать процедуры регулярного бэкапа и восстановления.

- **Содержание:**

- Скрипт для резервного копирования базы данных (`pg_dump`) и медиафайлов.
- Расписание запуска скрипта (через `cron`).
- Инструкция по восстановлению системы из бэкапа.

10. Приложения

- **А. Конфигурационные файлы:** Полные листинги ключевых конфигов (Nginx, systemd, .env).
- **В. Полезные команды:** Шпаргалка с командами для мониторинга и обслуживания (проверка логов, перезапуск сервисов).
- **С. Ссылки и ресурсы:** Ссылки на официальную документацию используемого ПО, внутренние Wiki-страницы.