

Инструкция по установке Зов Буревестника

Инструкция по установке экземпляра программного обеспечения, предоставленного для проведения экспертной проверки

Данная инструкция по установке разработана для проведения экспертной проверки экземпляра Программа для ЭВМ «Зов Буревестника».

Рекомендуемая ОС для проверки – Ubuntu, но Программа написана с использованием кроссплатформенных инструментов, поэтому описанная инструкция применима и для установки программы на других операционных системах.

СОДЕРЖАНИЕ

1. Используемые технические средства

Программный продукт состоит из трех компонентов, каждый из которых представлен в виде Docker-контейнеров, что обеспечивает гибкость и переносимость при развертывании. Каждый компонент требует определенного набора аппаратного и программного обеспечения для корректной работы:

Аппаратные требования к серверу:

- Архитектура процессора: x86, x86_64
- Минимальный объем оперативной памяти: 4 ГБ
- Минимальный объем свободного пространства на жестком диске: 10 ГБ
- Доступ в Интернет

Программные требования к серверу:

- Docker – для создания и управления контейнерами
- Docker Compose — это инструмент, который позволяет управлять многоконтейнерными приложениями.
- Операционная система: Linux (например, на основе Debian или Alpine)

1. Основное приложение с веб-интерфейсом на Django (myosotis-app)

Состав программного комплекса:

- Docker Image: содержит следующие компоненты:

- Интерпретатор Python версии 3.8
- Основные библиотеки и фреймворки:
 - Django версии 4.2.5 – основной веб-фреймворк
 - psycopg2 версии 2.9.8 – для работы с базой данных PostgreSQL
 - aiohttp версии 3.8.5 – для асинхронной работы с HTTP
 - requests версии 2.31.0 – для выполнения HTTP-запросов
 - loguru версии 0.6.0 – для логирования

2. Приложение ast (myosotis-ast)

Состав программного комплекса:

- Docker Image: содержит следующие компоненты:
 - Asterisk – система IP-телефонии, версии 18 или выше
 - Набор конфигурационных файлов, необходимых для настройки Asterisk в рамках проекта

3. Приложение vrs (myosotis-vrs)

Состав программного комплекса:

- Docker Image: содержит следующие компоненты:
 - Интерпретатор Python версии 3.8
 - Основные библиотеки и фреймворки:
 - FastAPI версии 0.109.0 – для создания REST API
 - vosk версии 0.3.44 – для перевода речи в текст
 - uvicorn версии 0.25.0 – для развертывания и обслуживания сервера
 - pydantic версии 2.5.3 – для валидации данных
 - orjson версии 3.9.10 – для работы с JSON
 - pydub версии 0.25.1 – для обработки аудио
 - Дополнительные зависимости:
 - ffmpeg – для обработки аудио и видео файлов
 - gcc – для компиляции зависимостей
 - libffi-dev и libssl-dev – для поддержки криптографических функций
 - vosk-model – загружаемая модель для распознавания речи

Программный продукт использует вышеуказанные контейнеры, что обеспечивает удобство развертывания и поддержки, а также изоляцию каждого компонента для повышения стабильности и безопасности системы.

2. Развертывание и вызов основных программных модулей

1. Подготовка окружения

Перед запуском приложения, необходимо подготовить ваше окружение. Для этого потребуется установить Docker и Docker Compose. Следуйте инструкциям ниже для корректной установки (инструкция для Ubuntu). Если Docker и Docker Compose уже установлены, можно переходить к пункту 2.

1.1. Установка Docker

Docker позволяет вам изолировать приложение в контейнере, что облегчает его развертывание и управление.

1. Обновите систему:

```
sudo apt-get update
```

2. Установите необходимые зависимости:

```
sudo apt-get install ca-certificates curl gnupg lsb-release
```

3. Добавьте официальный GPG ключ Docker:

```
sudo mkdir -m 0755 -p /etc/apt/keyrings curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

4. Добавьте Docker репозиторий в ваш список источников:

```
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" |  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

5. Установите Docker:

```
sudo apt-get update sudo apt-get install docker-ce docker-ce-cli  
containerd.io docker-buildx-plugin docker-compose-plugin
```

6. Убедитесь, что Docker установлен корректно, выполнив команду:

```
sudo docker --version
```

1.2. Установка Docker Compose

Docker Compose — это инструмент, который позволяет управлять многоконтейнерными приложениями.

1. Установите Docker Compose:

```
sudo apt-get install docker-compose-plugin
```

2. Проверьте установку:

```
docker compose version
```

Теперь ваше окружение готово для работы с Docker и Docker Compose.

2. Разворачивание приложения

Шаг 1: Создание рабочей папки

Создайте новую папку для размещения файла `docker-compose.yml`.

1. Создайте папку:

```
mkdir myosotis-deploy
```

2. Перейдите в созданную папку:

```
cd myosotis-deploy
```

Шаг 2: Создание файла `docker-compose.yml`

Теперь необходимо создать файл `docker-compose.yml`, который будет содержать конфигурацию для запуска приложения. Вместе с доступами для Шага 3 по запросу предоставляются и данные для доступа лицензионному серверу (переменные `LICENSE_BASE_URL`, `LICENSE_USERNAME`, `LICENSE_PASSWORD`).

1. Создайте файл:

```
nano docker-compose.yml
```

2. Вставьте следующий код:

```
version: "3.6"

services:
  app:
    image: cr.selcloud.ru/myosotis/myosotis-app:latest
    container_name: app
```

```
ports:
  - "8888:80"
restart: always
volumes:
  - ./myosotis/audio:/app/audio # Убедитесь, что эти пути
существуют на локальной машине
  - ./myosotis/records:/app/records
  - ./myosotis/media:/app/media
environment:
  - IS_DEMO=1
  - PROD=1
  - PROXY=
  - POSTGRES_HOST=db
  - POSTGRES_PORT=5432
  - POSTGRES_USER=user
  - POSTGRES_PASSWORD=password
  - POSTGRES_DB_NAME=db
  - POSTGRES_DB=db
  - LICENSE_BASE_URL= # Локальный URL
  - LICENSE_USERNAME=local_license_user # Заменить для
локального использования
  - LICENSE_PASSWORD=local_license_password # Заменить для
локального использования
depends_on:
  db:
    condition: service_healthy

ast:
  image: cr.selcloud.ru/myosotis/myosotis-ast:latest
  hostname: ast
  container_name: ast
  restart: always
  environment:
    - ASTER_IP=127.0.0.1 # Локальный IP
  volumes:
    - ./myosotis/audio:/app/audio # Убедитесь, что эти пути
существуют на локальной машине
    - ./myosotis/media:/app/media
    - ./myosotis/records:/app/records

vrs:
  image: cr.selcloud.ru/myosotis/myosotis-vrs:latest
  hostname: vrs
  container_name: vrs
  restart: always

db:
```

```
image: postgres:14
container_name: db
restart: always
ports:
  - "5454:5432"
environment:
  - PGDATA=/var/lib/postgresql/data/pgdata
  - POSTGRES_USER=user
  - POSTGRES_PASSWORD=password
  - POSTGRES_DB_NAME=db
  - POSTGRES_DB=db
volumes:
  - ./postgres_data:/var/lib/postgresql/data/pgdata # Убедитесь,
что путь существует на локальной машине
healthcheck:
  test: ["CMD-SHELL", "pg_isready", "-U", "${POSTGRES_USER}"]
  interval: 10s
  retries: 5
  timeout: 5s

volumes:
  postgres_data: # Локальный volume
```

3. Сохраните и закройте файл (в редакторе Nano нажмите `Ctrl + X`, затем `Y`, и нажмите `Enter`).

Шаг 3: Авторизация в Docker-хранилище

Для скачивания Docker-образов выполните авторизацию в приватном Docker-хранилище:

1. Выполните команду:

```
echo "XXXXX" | docker login cr.selcloud.ru -u "XXXXX" --password-stdin
```

- Замените `"XXXXX"` на ваш логин и `"XXXXXX"` на пароль или токен.

Шаг 4: Скачивание Docker-образов

После успешной авторизации скачайте образы приложения.

1. Выполните команду:

```
docker-compose pull
```

Эта команда скачает все необходимые образы, указанные в `docker-compose.yml`.

Шаг 5: Запуск приложения

Теперь можно запустить контейнеры.

1. Запустите приложение в фоновом режиме:

```
docker-compose up -d
```

Контейнеры начнут запускаться, и приложение станет доступно.

Шаг 6: Проверка статуса контейнеров

Чтобы проверить, что все контейнеры запущены корректно:

1. Выполните команду:

```
docker ps
```

Вы увидите список всех запущенных контейнеров, их статусы и назначенные порты.

Приложение развернуто, и демонстрационные данные загружены, так как значение переменной `IS_LOAD_DEMO=1`. Интеграции игнорируются, так как их поля оставлены пустыми.

3. Вызов тестового экземпляра ПО

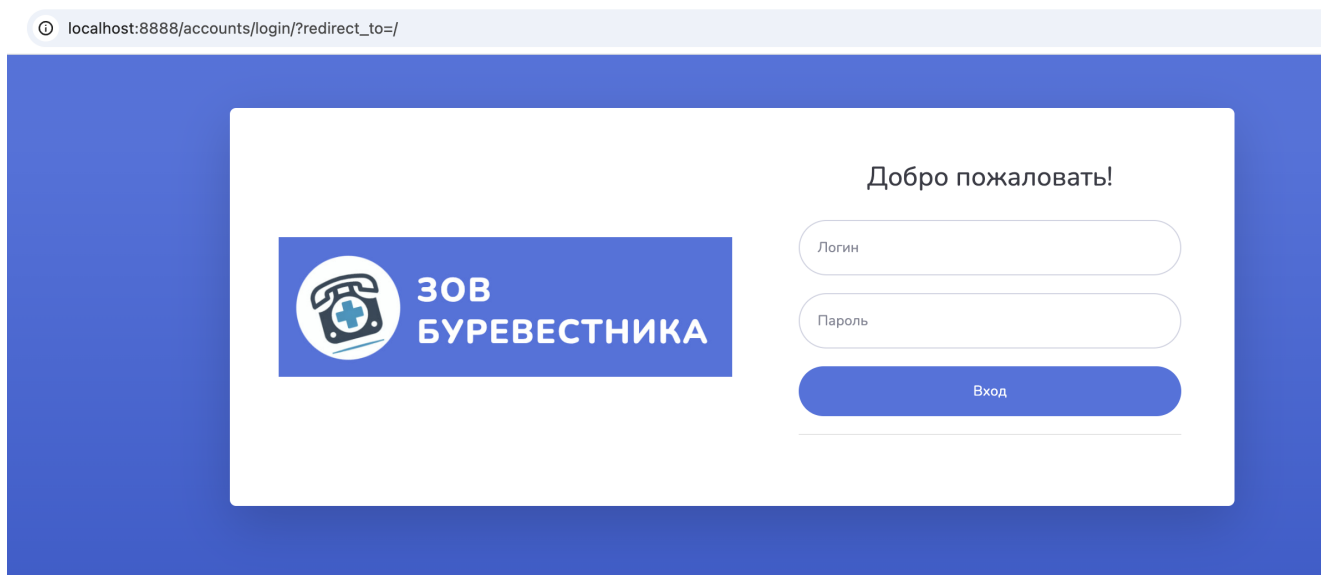
Для доступа к программному комплексу необходимо открыть интернет-браузер и ввести в адресную строку адрес его размещения: [HTTP://LOCALHOST:8888/](http://localhost:8888/) (локальная версия).

В окне веб-браузера появится окно авторизации пользователя. Для продолжения работы необходимо выполнить следующие действия:

- Ввести имя пользователя: **admin**;

- Ввести пароль: **admin**;
- Нажать на кнопку «Вход» для входа.

Этот процесс позволит авторизоваться и получить доступ к программному комплексу.



Доступ к основной функциональности сайта осуществляется через веб-интерфейс.

