

**Описание технической архитектуры
ПО “Skillometer”**

Описание технической архитектуры ПО “Skillometer”

Оглавление

1. Общие сведения.....	3
2. Описание архитектуры ПО.....	3
3. Заключение.....	4

1. Общие сведения

Настоящий документ содержит описание технической архитектуры программного обеспечения (ПО) «Skillometer», разработанного в рамках служебного задания №2 на разработку Информационной системы “Skillometer” от 22 августа 2022 г.

2. Описание архитектуры ПО

На рис.1 изображена схема архитектуры ПО.

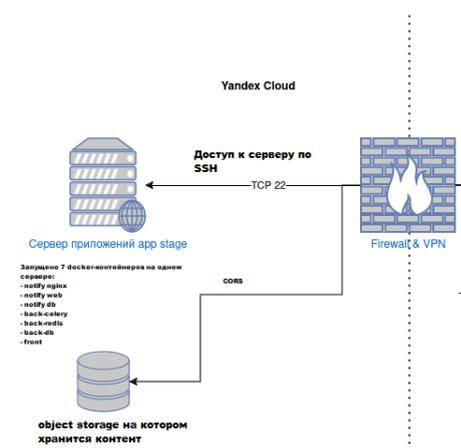


Рис. 1. Схема архитектуры ПО

Архитектура программного ПО представляет собой набор взаимосвязанных сервисов, каждый из которых выполняет определенные функции и взаимодействует с другими компонентами системы. Все сервисы развернуты в контейнерах Docker, что обеспечивает изоляцию и масштабируемость системы.

Ниже приведено описание основных компонентов архитектуры:

- notify_nginx_1

Назначение: Прокси-сервер, обеспечивающий маршрутизацию и балансировку запросов к другим сервисам.

Технология: nginx (лицензия BSD).

Описание: Используется для обработки входящих HTTP-запросов и их перенаправления на соответствующие сервисы.

- notify_web_1

Назначение: Сервис уведомлений, отвечающий за обработку и отправку уведомлений.

Технология: Uvicorn (лицензия BSD).

Описание: Сервис работает на базе Uvicorn, что обеспечивает высокую производительность и поддержку асинхронных запросов.

- notify_db_1

Назначение: База данных для хранения данных, связанных с уведомлениями.

Технология: PostgreSQL версии 14-alpine (лицензия BSD).

Описание: Реляционная база данных, используемая для хранения и управления данными сервиса уведомлений.

- back_celery_1

Назначение: Сервис для выполнения асинхронных задач.

Технология: Celery (лицензия BSD).

Описание: Сервис зависит от базы данных и Redis, использует тома для обмена данными с другими сервисами приложения. Предназначен для выполнения фоновых задач, таких как обработка данных и отправка уведомлений.

- back_redis_1

Назначение: Сервис для кэширования и хранения временных данных.

Технология: Redis (лицензия BSD).

Описание: Использует официальный образ Redis версии 7-alpine. Конфигурационный файл монтируется из локальной директории. Redis обеспечивает быстрое хранение временных данных и минимизацию задержек при работе с системой.

- back_db_1

Назначение: Основная база данных приложения.

Технология: PostgreSQL версии 15-alpine (лицензия BSD).

Описание: Реляционная база данных, используемая для хранения данных приложения. Настроена с проверкой готовности через команду pg_isready, что обеспечивает стабильность работы системы.

- back_web_1

Назначение: Основной сервис веб-приложения.

Технология: Docker, Gunicorn (лицензия BSD).

Описание: Сервис обрабатывает веб-запросы и взаимодействует с базой данных и Redis. Настройки сервиса включают монтирование томов и использование переменных окружения из файла .env.

3. Заключение

Представленная архитектура программного обеспечения (ПО) построена на основе современных технологий и микросервисного подхода, что обеспечивает высокую производительность, масштабируемость и отказоустойчивость системы. Все компоненты архитектуры, включая прокси-сервер nginx, сервисы уведомлений, базы данных PostgreSQL, асинхронные задачи на базе Celery и кэширование с использованием Redis, взаимодействуют между собой, обеспечивая стабильную и эффективную работу приложения.

Использование контейнеризации через Docker позволяет легко развертывать и масштабировать систему в различных средах, а также обеспечивает изоляцию и безопасность каждого сервиса. Все применяемые технологии имеют открытый исходный код и соответствуют современным стандартам разработки, что делает их подходящими для использования в рамках российского программного обеспечения.

Архитектура системы спроектирована с учетом требований к производительности, безопасности и удобству разработки, что позволяет эффективно решать задачи, поставленные перед приложением.

Генеральный директор



(Баишев Сергей Данилович)

