**Документация**

В текущей конфигурации приложение построено по принципу микро сервисной архитектуры и включает в себя три компонента, каждое из которых работает в отдельном docker контейнере. В терминологии файла docker-compose.yaml:

1. department – контейнер, в котором эмулируется работа сервера департамента
2. db – контейнер, в котором развёрнута БД postgresql, а также работает сервис, который периодически скачивает с сервера департамента файлы с датасетами и импортирует данные из них в БД
3. web – контейнер, в котором работает flask приложение, использующее БД на db контейнере

Репозиторий на GitHub доступен по ссылке: <https://github.com/klorman/WorkForecastingService>  
Репозиторий содержит:

1. DB-backup – папка с бэкапом базы данных, которая импортируется при начале работы
2. datasets – папка с датасетами, которые загружаются с сервера департамента department на db
3. src – папка с исходниками и конфигурационными файлами
4. docker-compose.yaml – файл для сборки и запуска контейнеров
5. для каждого контейнера есть соответствующий свой Dockerfile-<название контейнера> и requirements-<название контейнера>.txt

Для сборки и запуска сервиса необходима установка:

1. docker
2. docker-compose
3. Репозиторий с GitHub

Далее в корне репозитория необходимо ввести команду:

* docker-compose up –build

В проекте используются следующие конфигурационные файлы:

* app.config – в этом файле задаётся секретный ключ для flask приложения и директория, в которую скачиваются файлы с датасетами с устройства пользователя. Этот файл должен находиться на контейнере web.
* config.ini – файл для работы postgresql. В нём задаётся ip адрес сервера с БД (db контейнер); название БД; имя пользователя postgresql; пароль пользователя postgresql. Этот файл должен находиться на контейнерах web и db.
* data\_downloader.config – файл для работы сервиса, который периодически скачивает с сервера департамента файлы с датасетами и импортирует их в БД. В нём задаётся url сервера департамента (department контейнер); путь до директории, в которую скачиваются файлы с датасетами; расширение допустимых для скачивания файлов; временный интервал между загрузками и импортом файлов. Этот файл должен находиться на контейнере db.
* emulator.config – в этом файле задаётся путь до директории, в которой лежат подготовленные файлы с датасетами для скачивания. Этот файл должен находиться на контейнере department.

При изменении ip адресов контейнеров web и db необходимо внести соответствующие изменения в Dockerfile-postgres.