

ПРОГРАММНЫЙ КОМПЛЕКС
«ДАЛЬПЛАН»

Руководство администратора

ЦВАУ.01185-01 32

Листов 37

Инв.№ подл.	Подл. и дата	Бзм. инв. №	Инв.№ дубл.	Подл. и дата

2025

СОДЕРЖАНИЕ

Аннотация	4
1 Описание и работа	5
1.1 Наименование, обозначение, изготовитель	5
1.2 Назначение и область применения	5
1.3 Перечень эксплуатационной документации.....	6
2 Условия применения	7
2.1 Обеспечивающие технические средства.....	7
2.2 Обеспечивающие программные средства.....	8
3 Структура программного комплекса	10
3.1 Связи между компонентами и другими программами.....	10
3.2 Компоненты и составные части программного комплекса	10
3.3 Структура данных	12
3.4 Тома Docker и их назначение.....	13
4 Описание работы программного комплекса	16
4.1 Этапы работы по заданию	16
4.2 Сценарии использования	16
5 Установка и настройка.....	18
5.1 Установка и конфигурация C#-сервисов и клиентских приложений...	19
5.1.1 Подготовка конфигурационных файлов.....	19
5.1.2 Порядок выполнения операций для первого запуска системы	19
5.1.3 Проверка системы	20
5.2 Установка и конфигурация geo-content-service.....	20
5.2.1 Подготовка конфигурационных файлов.....	21
5.2.2 Compose-файл.....	24
5.2.3 Запуск	25
5.2.4 Проверка системы	25
5.3 Установка и конфигурация Pipeline-гри.....	25
5.3.1 Подготовка конфигурационных файлов.....	26
5.3.2 Порядок выполнения операций для первого запуска системы	26
5.3.3 Проверка системы	26
5.4 Установка и конфигурация фронтенд приложений.....	27
5.4.1 Compose-файл.....	27
5.4.2 Запуск	27
5.4.3 Проверка.....	28
5.5 Установка и конфигурация тайлового сервера	28
5.5.1 Подготовка конфигурационных файлов.....	28
6 Проверка работоспособности	31
7 Обновление и удаление программных компонентов.....	32

8 Аварийные ситуации	33
8.1 Действия при отказах технических средств	33
8.2 Восстановление программ и данных.....	33
9 Перечень принятых сокращений	34
10 Термины и определения.....	35

АННОТАЦИЯ

В данном документе приведены необходимые сведения по установке, настройке и обеспечению работы программного изделия «Программный комплекс «Дальплан» (далее – программный комплекс «Дальплан», программный комплекс).

Документ предназначен для *системных администраторов* – сотрудников организации, обеспечивающих установку и настройку продукта, а также администрирование в ходе его работы.

Настоящий документ не заменяет учебную, справочную литературу, руководства от производителя операционной системы и прочие источники информации, освещающие работу с операционной системой и ее графическим пользовательским интерфейсом.

Эксплуатация технических средств изделия осуществляется в соответствии с комплектом документации, поставляемой изготовителем.

Для выделения важных сведений в документе используется следующее оформление:

 Ссылки на другие документы в основном тексте.

 Примечания; важные сведения; указания на действия, которые необходимо выполнить в обязательном порядке.

 Требования, несоблюдение которых может привести к некорректной работе, повреждению или выходу из строя изделий или программного обеспечения.

ВНИМАНИЕ! Изображения экранов программы, представленные в документе, могут незначительно отличаться от реальных изображений. Данные различия зависят от версии программного обеспечения и настроек персональных свойств экрана операционной системы.

1 ОПИСАНИЕ И РАБОТА

1.1 Наименование, обозначение, изготовитель

Наименование: Программный комплекс «Дальплан»

Обозначение: ЦВАУ.01185-01

Страна- Российская Федерация
изготовитель:

Предприятие- Общество с ограниченной ответственностью
изготовитель: «Центр речевых технологий»

Адрес: 194044, Санкт-Петербург, Выборгская набережная, дом 45,
литера Е, помещение 1-Н, офис 133

Телефон: (812) 325-88-48

Факс: (812) 327-92-97

1.2 Назначение и область применения

Программный комплекс «Дальплан» обеспечивает выполнение следующих функций:

- автоматический поиск и классификация объектов, анализ изображения (вторичная обработка) для выделения и распознавания объекта, определения его координат;
- ретроспективный анализ изменений на местности: выявление изменений в качественном и количественном аспекте за определенные периоды времени, которые зависят от тематики исследования.

Программное обеспечение (далее – ПО) программного комплекса «Дальплан» обеспечивает следующие возможности:

- загрузка снимков в систему в ручном режиме;

- универсальный программный интерфейс для автоматического приема снимков от БПЛА;
- автоматическое обнаружение и классификация объектов;
- получение координат объектов;
- наложение снимка и объектов на картографическую подложку;
- ручная корректировка привязки снимков к геоподложке;
- проверка и редактирование результатов детекции объектов оператором;
- ввод информации оператором по результатам обработки снимков в ручном режиме;
- анализ снимка по зонам интереса;
- измерение линейных размеров объектов;
- нанесение дополнительных геометрических элементов на снимок и карту: линии, полигоны, точки;
- экспорт файлов с координатами объектов в ручном режиме.

1.3 Перечень эксплуатационной документации

Сотрудники, осуществляющие установку программного комплекса «Дальплан» на серверном оборудовании, должны ознакомиться с руководством администратора программного комплекса «Дальплан».

Сотрудники, осуществляющие непосредственную работу с ПО программного комплекса «Дальплан» на АРМ, должны ознакомиться с руководством пользователя программного комплекса «Дальплан».

2 УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Обеспечивающие технические средства

Работа ПО программного комплекса «Дальплан» может осуществляться на сервере, отвечающем минимальным требованиям, приведенным в таблице 1.

Таблица 1 – Минимальные требования к серверу

Наименование показателя	Значение
Процессор	32-поточный x86_64-совместимый
Оперативная память (RAM):	–
тип памяти	DDR5
объем памяти, Гб	128
Видеоускоритель:	–
объем памяти, Гб	24
тип памяти	GDDR6
разрядность шины памяти, бит	384
производительность, TFlops	80
Системный диск:	–
емкость, Тб	2
шина	pci-e 4
поддержка	nvme
скорость запись/чтение, Мб/с	7000/5000
ресурс записи, Тб	1200
Емкость дисков, Тб	8
Шина:	–
тип	SATA 6Gbps
скорость запись/чтение, Мб/с	560/530
ресурс записи, Тб	2800
Емкость дисков, Тб	8
Порты ввода/вывода	USB версии 3.0 или выше, HDMI версии 2.1а или выше, Ethernet 1Gbps или выше (RJ45)
Операционная система	Ubuntu 24.04, Debian 12.10

Работа с ПО программного комплекса «Дальплан» может осуществляться на любом персональном компьютере (АПМ), отвечающем минимальным требованиям, приведенным в таблице 2.

Таблица 2 – Минимальные требования АРМ пользователя

Компонент	Характеристика	
Системный блок	CPU, количество ядер, не менее	4
	Объем оперативной памяти, ГБ, не менее	8
	Накопитель, ГБ, не менее	500
	Количество сетевых портов RJ45 со скоростью не менее 10/100/1000 Мбит/с, шт., не менее	1
ПО общего назначения	Операционная система, шт.	1
Монитор	ЖК, диагональ не менее 27, шт.	1
	Разрешение, ЭИ	2560 x 1440
	Соотношение сторон	16:9
Наушники	Наушники накладные, импеданс 24 Ом ± 8 Ом, диапазон воспроизводимых частот не хуже 20 – 20000 Гц, шт.	1
Устройства ввода	Клавиатура USB, шт.	1
	Манипулятор мышь оптическая, USB, шт.	1
Сеть	Патч корд RJ45 длиной не менее 3 метров, шт.	1

Допускается применение ноутбука, при условии соответствия его указанным выше требованиям, а также наличия подключения к Интернет.

2.2 Обеспечивающие программные средства

На средства вычислительной техники должно быть предустановлено лицензионное программное обеспечение, обеспечивающее функционирование данных средств, возможность обмена информацией в локальной сети, средства для передачи данных по телекоммуникационным сетям для обеспечения удалённого обмена по стандартным протоколам.

К ПО относятся операционные системы (серверные и клиентские), управляющие, как аппаратными компонентами, так и другими программами.

ПО программного комплекса:

- имеет лицензии на неисключительные права на его использование;
- включают в себя средства, обеспечивающие возможность обмена информацией между компонентами программного комплекса в локальной сети, средства для передачи данных по телекоммуникационным сетям для обеспечения удаленного обмена между компонентами программного комплекса по стандартным протоколам.

ПО программного комплекса имеет пользовательский интерфейс на русском языке.

ПО программного комплекса предустановлено и активировано на всем оборудовании, где требуется его установка.

3 СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА

3.1 Связи между компонентами и другими программами

Общая структура, распределение элементов программного комплекса и их взаимодействие представлены на рисунке 1.

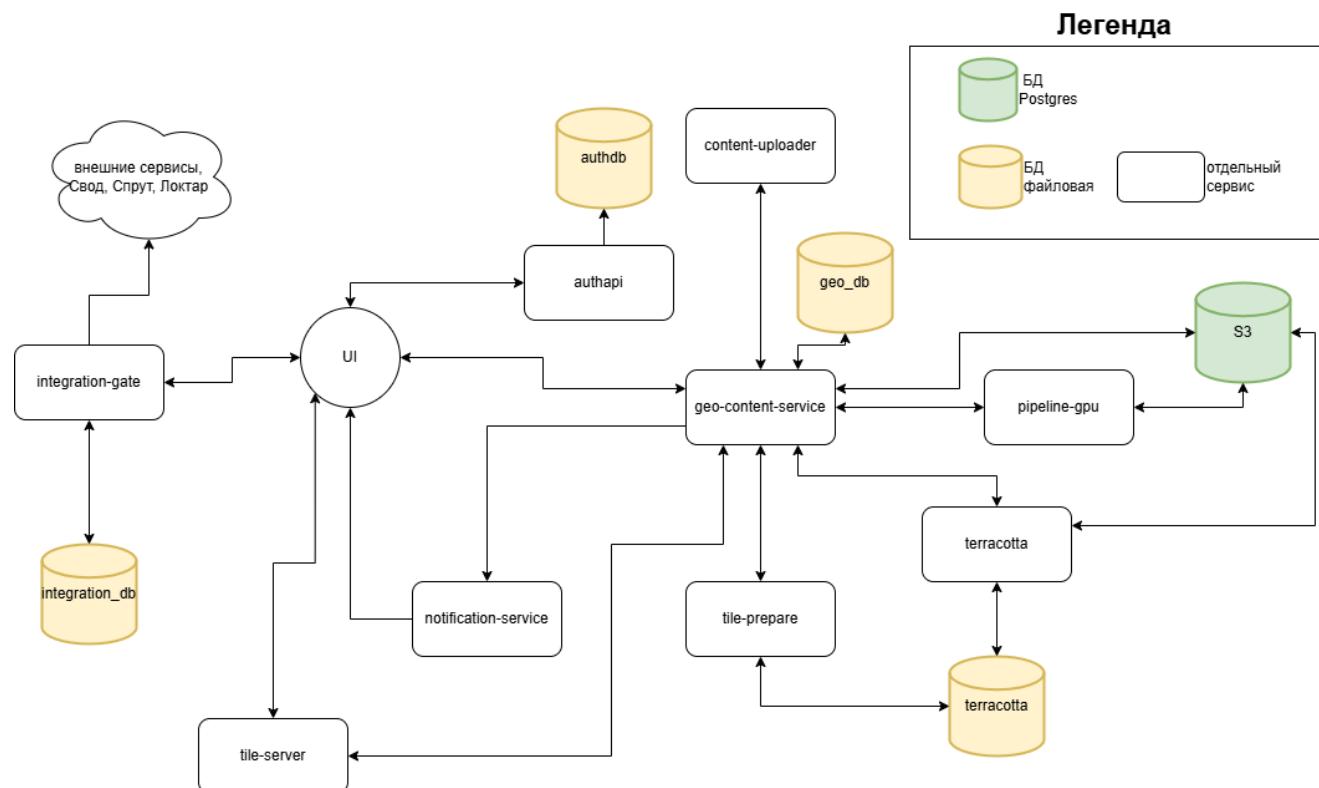


Рисунок 1 – Схема развёртывания программного комплекса «Дальплан»

Программный комплекс «Дальплан» имеет клиент-серверную архитектуру. В соответствии с указанной архитектурой практически все данные хранятся и обрабатываются на стороне сервера.

Клиентские рабочие места взаимодействуют с сервером по сетям Intranet с использованием стандартного стека протоколов TCP/IP.

3.2 Компоненты и составные части программного комплекса

ПО программного комплекса запускается в **docker compose** и состоит из нескольких модулей. Разбивка на модули позволяет устанавливать и перезапускать части сервисов без необходимости перезапуска всего программного комплекса.

Состав модулей:

common-infra-module - общие сервисы, собранные на основе open-source проектов;

PostgreSQL (сервис **postgres-geo-platform**) - СУБД с базами данных, которые используют приложения платформы;

MinIO (сервис s3-server) - S3 сервис для хранения бинарных данных;

RabbitMq (сервис **rabbitmq**) - сервис для пересылки сообщений между новыми компонентами;

proxy-infra-module - сервис-фасад, перенаправляющий внешние запросы к компонентам платформы;

authapi - Хранит данные о пользователях и обеспечивает аутентификацию;

integration-module – сервисы импорта-экспорта данных;

integration-gate – сервис отправки данных во внешние системы;

content-uploader – сервис автоматической загрузки входных данных;

front-module - клиентское веб-приложение;

notifications-module - сервисы пересылки уведомлений между компонентами платформы;

notifications-service - прием сообщений от компонентов;

notifications-processor-platform - общий обработчик сообщений;

geo-content-module - сервисы обработки гео-информации;

geo-content-service - основной бэкенд платформы;

tile-prepare – преобразует исходный tiff файл в формат COG Tiff, обеспечивает извлечение дополнительной информации из графических данных;

terracotta - хранит и отдает данные из COG Tiff для отображения фото на карте;

tile-server — сервис картографических подложек.

3.3 Структура данных

Все хранимые данные представлены в виде томов (volume) докер. Также как и сервисы, данные разбиты на данные и конфигурации, затем на соответствующие модули и сервисы в модулях.

Данные и конфигурации распределены по директориям **data** и **configs** соответственно.

Далее по вложенности следуют модули, где имя директории каждого модуля должно оканчиваться на **-module**, а имя целиком должно соответствовать **-docker-compose.yml** файлу), затем конкретные сервисы.

Имена директорий, соответствующих сервисам, должны совпадать с именами сервисов в compose-файлах.

Общая структура директорий имеет вид, представленный на рисунке 2.



Рисунок 2 – Структура директорий

3.4 Тома Docker и их назначение

1. Postgres

data/common-infra-module/postgres/data:/var/lib/postgresql/data

директория данных СУБД PostgreSQL;

configs/common-infra-module/postgres/postgresql.conf: /etc/postgresql.conf -

основной файл конфигурации PostgreSQL;

configs/common-infra-module/postgres/init-scripts:/docker-entrypoint-initdb.d

- скрипты инициализации, используются при первом запуске PostgreSQL, если директория данных пустая,

2. MinIO(s3-server)

data/common-infra-module/s3/data:/data - данные хранимые в S3.

3. GeoContentService

configs/geo-content-module/geo-content-

service/appsettings.json:/app/appsettings.json - основной файл конфигурации сервиса.

4. Integration-gate

configs/integration-module/content-uploader/appsettings.json – основной файл конфигурации сервиса

5. ContentUploader

configs/integration-module/content-uploader/appsettings.json – основной файл конфигурации сервиса

6. IntegrationGate

configs/integration-module/integration-gate/appsettings.json – основной файл конфигурации сервиса

7. NotificationsService

configs/notifications-module/notifications-service-
api/appsettings.json:/app/appsettings.json - основной файл конфигурации сервиса.

8. NginxApiGate

configs/proxy-infra-module/nginx-api-gate/nginx-
conf/nginx.conf:/etc/nginx/nginx.conf - основной файл конфигурации nginx;

configs/proxy-infra-module/nginx-api-gate/nginx-
conf/conf.d:/etc/nginx/conf.d - директория с конфигами nginx;

configs/proxy-infra-module/nginx-api-gate/nginx-
conf/include:/etc/nginx/include - include-директория с конфигами nginx;

configs/proxy-infra-module/nginx-api-gate/local:/local - локальные файлы nginx.

9. AuthApi

configs//users-module/authapi/appsettings.json:/app/appsettings.json -
основной файл конфигурации сервиса

Порядок установки и настройки описан в разделе 5 «Установка и настройка».

4 ОПИСАНИЕ РАБОТЫ ПРОГРАММНОГО КОМПЛЕКСА

4.1 Этапы работы по заданию

Описание этапов работы по заданию в рамках ПО программного комплекса «Дальплан» представлено в таблице 3



Подробное описание функций, выполняемых программным комплексом, представлено в документе «Программный комплекс «Дальплан». Руководство пользователя».

Таблица 3 – Описание этапов работы по заданию

Этап	Описание
Создание задания	Создание задания в программном комплексе. Заполнение сопроводительной информации по заданию.
Обеспечение поступления в задание данных	Подключение источника данных или загрузка данных в задание.
Запуск задания	Запуск выполнения задания
Работа по заданию	Обработка данных, выполнение идентификации объектов, вычисление координат объектов.
Предоставление сведений	Обеспечение обратной связи по поставленному заданию. Контроль хранения данных.

Этапы «Работы по заданию» и «Предоставление сведений» циклично повторяются по мере поступления данных, если не установлен запрет на пополнение задания новыми данными.

4.2 Сценарии использования

Основной сценарий:

- пользователь создает задание на детекцию объектов на фото;
- пользователь загружает в задание снимок или несколько снимков в формате .TIFF или .GeoTIFF;
- ПО программного комплекса выполняет проверку загруженных данных, проверяет расширения файла/файлов на соответствие поддерживаемым форматам;

- пользователь инициирует процесс обработки файла/файлов;
- ПО программного комплекса проводит поиск известных классов на снимках объектов;
- ПО программного комплекса определяет координаты обнаруженных на снимках объектов;
- ПО программного комплекса выделяет цветным полигоном найденные объекты в соответствии с классом объекта;
- пользователь выгружает информацию об обнаруженных объектах и их координатах в *.kml файл;
- ПО программного комплекса выполняет отправку результатов детекции объектов в стороннюю систему в соответствии с настройками интеграции.

Дополнительный сценарий:

- пользователь создает задание пользователь на импорт файлов форматов *.kml (или *.csv), содержащих координаты объектов;
- пользователь загружает в задание файл или несколько файлов формата *.kml (или *.csv);
- ПО программного комплекса выполняет проверку загруженных данных, проверяет расширения файла/файлов на соответствие поддерживаемым форматам;
- пользователь инициирует процесс обработки файла/файлов;
- ПО программного комплекса регистрирует объекты, описанные в импортируемых файлах;
- ПО программного комплекса обозначает результаты обработки в виде полигонов на картографической подложке;
- пользователь выгружает информацию об обнаруженных объектах и их координатах в *.kml файл;
- ПО программного комплекса выполняет отправку результатов детекции объектов в стороннюю систему в соответствии с настройками интеграции.

5 УСТАНОВКА И НАСТРОЙКА

Перед установкой ПО программного комплекса «Дальплан» необходимо выполнить перечисленные ниже действия.

- 1) Установить на компьютер, выполняющий роль сервера, одну из операционных систем:, Ubuntu 24.04, Debian 12.10.
- 2) Установить на все компьютеры, выполняющие роль клиентских рабочих мест, произвольную операционную систему, обеспечивающую наличие web-браузера.
- 3) Произвести обновление ОС на компьютере, выполняющем роль сервера.
- 4) Настроить локальную сеть. Убедиться, что в программном комплексе задан статический IP-адрес.
- 5) Убедиться, что все вычислительные средства, работающие с программным комплексом «Дальплан», находятся в одной временной зоне и на них установлено одинаковое время.
- 6) Установить общее программное обеспечение (пререквизиты), актуальные драйвера для оборудования и произвести настройку программного комплекса.
- 7) Установить на компьютер, выполняющий роль сервера, docker и docker compose.
- 8) Импортировать docker-образы из поставки.
- 9) Убедиться, что на компьютере, выполняющем роль сервера, установлены и запущены приложения для исполнения моделей и необходимые модели загружены на S3.
- 10) Убедиться, что на компьютере, выполняющем роль сервера, платформа находится в рабочем состоянии, т.е. можно выполнить обработку изображения через соответствующее фронтенд-приложение.

5.1 Установка и конфигурация C#-сервисов и клиентских приложений

5.1.1 Подготовка конфигурационных файлов

Подготовить каталог *data* согласно структуре, указанной в *Readme.md*-файле репозитория конфигов локальных стендов.

Поменять значения секретов в *appsettings.json*-файлах на актуальные.

Для каждого docker-compose-файла сгенерирован *.env*-файл с его переменными окружения.

```
common-infra-module-docker-compose.yml geometry-docker-compose.yml ✓ Bakplatform laydin 16.1
down.sh integration-module-docker-compose.yml ↵ proxy-infra-module-docker-compose.yml
front-module-docker-compose.yml logging.json tiles-module-docker-compose.yml 9 июля 20
geo-content-module-docker-compose.yml ns-appsettings.json up.sh
geometry-dependencies.yml pipeline-gpu-module-docker-compose.yml ↵ users-module-docker-compose.yml
```

В общем виде схема развертывания состоит из выполнения запуска скрипта *up.sh*.

5.1.2 Порядок выполнения операций для первого запуска системы

- 1) создать в настройках S3 платформы Сбербанка нового пользователя (для minio связку Access Key / Secret Key) с правами на создание бакетов. И прописать туда Access Key / Secret Key из конфигов для minio Визиря.

Пример конфига ключа Minio на создание бакетов:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "admin:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:)"
      ]
    },
    {
      "Effect": "Allow",
```

```
"Action": [  
    "s3:*"  
,  
    "Resource": [  
        "arn:aws:s3:::*"  
    ]  
}  
]  
}
```

2) Запустить data infra

```
docker compose -f ./data-infra-docker-compose.yml --env-file  
./data-infra.env up -d
```

3) Создать пользователей и базы в PostgreSQL

4) Запустить сервис аутентификации

```
docker compose -f ./users-module-docker-compose.yml --env-file  
./users-module.env up -d
```

5) Запустить прокси инфраструктуру

```
docker compose -f ./proxy-infra-docker-compose.yml --env-file  
./proxy-infra.env
```

5.1.3 Проверка системы

Через приложение администратора, запущенное по умолчанию на порту 9908, авторизоваться под ролью Суперпользователя, настроить рабочие области по умолчанию и создать обычного пользователя (права ролей admin+operator).

Через пользовательское приложение, запущенное на порту, настроенном выше, зайти от имени созданного пользователя и запустить на выполнение задание.

Если задание выполнилось успешно, значит система готова к работе

5.2 Установка и конфигурация geo-content-service

Перед началом установки убедиться, что на компьютере, выполняющем роль сервера, установлены и успешно запущены: инфраструктурный модуль (PG, S3, RabbitMQ, NATS).

5.2.1 Подготовка конфигурационных файлов

Необходимо заполнить файл переменных окружения **geo-content-module.env.env**:

```
#geo-content-service
GEO_CONTENT_SERVICE_TAG=latest
GEO_CONTENT_SERVICE_PORT=8089
GEO_CONTENT_SERVICE_APPSETTINGS_PATH=/var/app/compose_files/geo-
content-service/appsettings.json

#tile-prepare
TILE_PREPARE_TAG=latest
AWS_S3_REGION=ru-central-1
AWS_S3_ENDPOINT=http://s3-server:9000
AWS_S3_ACCESSKEYID=<value>
AWS_S3_SECRETACCESSKEY=<value>
AWS_S3_BUCKET=andr
AWS_S3_FORCEPATHSTYLE=true
AMQP_HOST=rabbitmq
AMQP_USERNAME=guest
AMQP_PASSWORD=guest
DB_URL=postgresql://user:password@postgres-geo-
platform:5432/terracotta
FILE_STORAGE_TILEDIR=/tiles/

#terracotta
TC_DRIVER_PATH=postgresql://user:password@postgres-geo-
platform:5432/terracotta
TC_DRIVER_PROVIDER=postgresql
TC_AWS_DEFAULT_REGION=ru-central-1
TC_AWS_ACCESS_KEY_ID=<value>
TC_AWS_SECRET_ACCESS_KEY=<value>
TC_AWS_STORAGE_BUCKET_NAME=andr
TC_AWS_S3_ENDPOINT=s3-server:9000
TC_AWS_HTTPS=false
TC_AWS_VIRTUAL_HOSTING=false
```

И заполнить **appsettings.json** для geo-content-service:

```
{
  "Serilog": {
    "Using": [ "Serilog.Sinks.Console",
      "Serilog.Sinks.ElasticSearch" ],
    "MinimumLevel": {
```

```
"Default": "Information",
"Override": {
    "Microsoft": "Information",
    "System": "Information",
    "Microsoft.AspNetCore": "Warning",
    "Microsoft.EntityFrameworkCore": "Warning",
    "Microsoft.EntityFrameworkCore.Infrastructure": "Information",
    "Microsoft.EntityFrameworkCore.Migrations": "Information"
},
"WriteTo": [
{
    "Name": "Console",
    "Args": {
        "restrictedToMinimumLevel": "Information",
        "formatter": "Serilog.Formatting.Compact.RenderedCompactJsonFormatter",
        "Serilog.Formatting.Compact"
    }
},
{
    "Enrich": [ "FromLogContext", "WithMachineName",
    "WithThreadId" ],
    "Properties": {
        "Application": "tasks-service",
        "Environment": "Production"
    }
},
{
    "geo-content-service": {
        "DbSettings": {
            "Provider": "PostgreSql",
            "ConnectionString": "User
Id=user; Password=password; Host=postgres-geo-
platform; Port=5432; Database=geo_db",
            "CommandTimeout": 30
        },
        "TerracottaDbSettings": {
            "Provider": "PostgreSql",
            "ConnectionString": "User
Id=user; Password=password; Host=postgres-geo-
platform; Port=5432; Database=geo_db"
        }
    }
}
]
```

```
"ConnectionString": "User
Id=user;Password=password;Host=postgres-geo-
platform;Port=5432;Database=terracotta",
"CommandTimeout": 30
},
"configuration-service": {
"ServiceUrl": "http://configuration-service/",
"NumberOfConnectionAttempts": 5,
"DelayInMillisecondsBetweenAttempts": 5000
},
"terracotta-service": {
"ServiceUrl": "http://terracotta:8086"
},
"storage-settings": {
"AllowedFileTypes": [
{
"Name": "Tiff",
"Extensions": [
"tiff",
"tif"
],
"UploadBucket": "andr"
},
{
"Name": "Jpeg",
"Extensions": [
"jpg"
],
"UploadBucket": "jpg"
},
{
"Name": "GeoContentFile",
"Extensions": [
"kml",
"csv"
],
"UploadBucket": "geo-content"
}
]
},
"Logging": {
"LogLevel": {
"Default": "Debug",
"File": "Info"
}
}
```

```
        "Microsoft.AspNetCore": "Debug"
    }
},
"AllowedHosts": "*"
}
```

5.2.2 Compose-файл

Модуль состоит из трех компонентов: непосредственно сам geo-content-service, а также tile-prepare (нарезка тайлов и анализ разрешения фото) и terracotta (хранение тайлов). Пример compose-файла **geo-content-module-docker-compose.yml**:

```
networks:
  geometrynet:
    external: true

services:
  geo-content-service:
    image: it-harbor.speechpro.com/geometrynet-box-dev/stc/geo-
content-service:${GEO_CONTENT_SERVICE_TAG}
    restart: always
    volumes:
      -
        ${GEO_CONTENT_SERVICE_APPSETTINGS_PATH}:/app/appsettings.json"
    networks:
      - "geometrynet"
    ports:
      - "${GEO_CONTENT_SERVICE_PORT}:80"
    extra_hosts:
      - "host.docker.internal:host-gateway"

  tile-prepare:
    image: it-harbor.speechpro.com/geometrynet-box-
dev/stc/tile_prepare:${TILE_PREPARE_TAG}
    restart: always
    env_file:
      - ./geo-content-module.env
    networks:
      - "geometrynet"
    extra_hosts:
      - "host.docker.internal:host-gateway"
```

```
terracotta:  
    image: it-harbor.speechpro.com/geometrynet-  
box/terracotta:2.0-SNAPSHOT  
    restart: always  
    environment:  
        - TC_DRIVER_PATH=${TC_DRIVER_PATH}  
        - TC_DRIVER_PROVIDER=${TC_DRIVER_PROVIDER}  
        - AWS_DEFAULT_REGION=${TC_AWS_DEFAULT_REGION}  
        - AWS_ACCESS_KEY_ID=${TC_AWS_ACCESS_KEY_ID}  
        - AWS_SECRET_ACCESS_KEY=${TC_AWS_SECRET_ACCESS_KEY}  
        - AWS_STORAGE_BUCKET_NAME=${TC_AWS_STORAGE_BUCKET_NAME}  
        - AWS_S3_ENDPOINT=${TC_AWS_S3_ENDPOINT}  
        - AWS_HTTPS=${TC_AWS_HTTPS}  
        - AWS_VIRTUAL_HOSTING=${TC_AWS_VIRTUAL_HOSTING}  
    networks:  
        - "geometrynet"  
    ports:  
        - "8086:8086"  
    privileged: true
```

5.2.3 Запуск

Запуск модуля осуществляется командой

```
docker compose --env-file geo-content-module.env -f geo-content-  
module-docker-compose.yml up -d
```

5.2.4 Проверка системы

С помощью браузера зайти на хост и порт, указанный в GEO_CONTENT_SERVICE_PORT, по пути /swagger - должен открыться swagger сервиса.

5.3 Установка и конфигурация Pipeline-gpu

Перед началом установки убедиться, что на компьютере, выполняющем роль сервера, установлен и успешно запущен инфраструктурный модуль (PG, S3, RabbitMQ, NATS).

5.3.1 Подготовка конфигурационных файлов

Необходимо заполнить файл переменных окружения *pipeline-gpu-module.env*:

```
export pipeline_gpu_image_tag=2.2.0
export models_aws_secret_access_key=<value>
export to_aws_access_key_id=<value>
export to_aws_secret_access_key=<value>
export from_aws_secret_access_key=<value>
export models_aws_access_key_id=<value>
export from_aws_access_key_id=<value>
export to_endpoint_url=http://s3-server:9000
export from_endpoint_url=http://s3-server:9000
export models_endpoint_url=http://s3-server:9000
export to_bucket=andr
export models_bucket=andr
export from_bucket=andr
export CUDA_VISIBLE_DEVICE=0
export CACHE_GEOMETRY_VOLUME=/var/app/compose_files/pipeline-gpu/.cache_geometry
export GPU_CONFIG_VOLUME=/var/app/compose_files/pipeline-gpu/gpuconfig/
```

5.3.2 Порядок выполнения операций для первого запуска системы

- 1) Создать в настройках S3 нового пользователя (для minio связку Accesss Key / Secret Key) с правами на создание бакетов, и прописать туда Accesss Key / Secret Key из вышеприведенного списка переменных
- 2) Запустить common-infra:

```
docker compose -f ./common-infra-docker-compose.yml --env-file
./common-infra.env up -d
```

- 3) Загрузить пакет моделей в S3
- 4) Запустить сервис pipeline-gpu:

```
docker compose -f ./pipeline-gpu-module-docker-compose.yml --
env-file ./pipeline-gpu-module.env up -d
```

5.3.3 Проверка системы

С помощью браузера зайти хост на порт 8090, по пути /docs. При этом должен открыться swagger сервиса.

Внимание, первый запуск (или после очистки CACHE_GEOMETRY_VOLUME) первый запуск может занимать десятки минут.

5.4 Установка и конфигурация фронтенд приложений

Перед началом установки убедиться, что на компьютере, выполняющем роль сервера, установлены и успешно запущены: бэкенд сервисы и инфраструктурный модуль (PG, S3, RabbitMQ, NATS).

5.4.1 Compose-файл

Пример compose-файла:

```
networks:  
  geometrynet:  
    external: true  
  
services:  
  
  angular-front-arm:  
    image: it-harbor.speechpro.com/geometrynet-box-dev/stc/arm-app:latest  
    restart: always  
    environment:  
      - JAVA_BACK_URL=geo-content-service/api/v1  
    networks:  
      - "geometrynet"  
    ports:  
      - "8888:80"  
    extra_hosts:  
      - "host.docker.internal:host-gateway"
```

Внимание! Переменная окружения JAVA_BACK_URL указывает на основной сервис модуля geo-content-service. Название переменной сохранилось по историческим причинам.

5.4.2 Запуск

```
docker compose -f docker-compose-front.yml up -d
```

5.4.3 Проверка

Через пользовательское приложение, запущенное на порту, настроенном выше нужно зайти от имени созданного пользователя, и запустить на выполнение задание.

Если задание выполнилось успешно, значит система готова к работе.

5.5 Установка и конфигурация тайлового сервера

5.5.1 Подготовка конфигурационных файлов

Конфигурационный файл представляет собой указание расположения шрифтов, стилей и источника данных (файл mbtiles)

Пример конфигурационного файла:

```
{
  "options": {
    "paths": {
      "fonts": "fonts",
      "styles": "styles"
    }
  },
  "styles": {
    "tiles": {
      "style": "osm-bright/style.json",
      "tilejson": {
        "type": "overlay",
        "bounds": [8.529446, 47.364758, 8.55232, 47.380539]
      }
    },
    "klokantech-basic": {
      "style": "klokantech-basic/style.json",
      "tilejson": {
        "type": "overlay",
        "bounds": [8.529446, 47.364758, 8.55232, 47.380539]
      }
    }
  },
  "data": {
    "openmaptiles": {
      "mbtiles": "maptiler-osm-2020-02-10-v3.11-planet.mbtiles"
    }
  }
}
```

```
    }  
}  
}
```

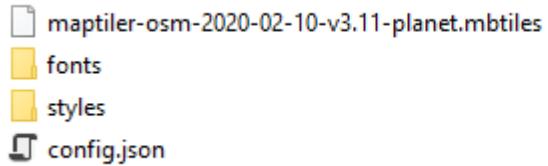
Блок **styles** влияет на пути, по которым доступны тайлы: в приведенном примере конфигурационного файла два стиля и соответственно шаблоны путей (для png-тайлов) для tiles будут /styles/tiles/{z}/{x}/{y}.png, а для klokantech-basic - /styles/klokantech-basic/{z}/{x}/{y}.png

Блок **data** указывает на расположение файла с векторными данными (путь указывается относительно текущего расположения конфига)

Примерный docker-compose.yml файл для двух экземпляров серверов, обслуживающих разные наборы карт:

```
services:  
  tile-server-osm:  
    image: it-harbor.speechpro.com/geometrynet-box-  
dev/maptiler:v4.7.0  
    restart: always  
    ports:  
      - "0.0.0.0:8621:8080"  
    volumes:  
      - "./data:/data"  
    networks:  
      - "geometrynet"  
    extra_hosts:  
      - "host.docker.internal:host-gateway"  
  
  tile-server-satellite:  
    image: it-harbor.speechpro.com/geometrynet-box-  
dev/maptiler:v4.7.0  
    restart: always  
    ports:  
      - "0.0.0.0:8622:8080"  
    volumes:  
      - "./data-satellite:/data"  
    networks:  
      - "geometrynet"  
    extra_hosts:  
      - "host.docker.internal:host-gateway"
```

Структура подключаемого тома /data:



Для доступа из приложения необходимо настроить ***nginx proxy*** для доступа к разным экземплярам карт:

```
location /tile-server/global-map/satellite/ {
    include /etc/nginx/conf.d/cors.conf;
    include /etc/nginx/conf.d/proxy.conf;
    proxy_pass
http://host.docker.internal:8622/data/openmaptiles/;

}

location /tile-server/global-map/osm/ {
    include /etc/nginx/conf.d/cors.conf;
    include /etc/nginx/conf.d/proxy.conf;
    proxy_pass
http://host.docker.internal:8621/styles/tiles/;
```

6 ПРОВЕРКА РАБОТОСПОСОБНОСТИ

Отслеживание состояния работы сервисов производится в docker compose.

Во вкладке **Containers** представлены сервисы программного комплекса (с указанием названием контейнеров).

Для каждого сервиса отображается:

- идентификатор контейнера,
- порт, на который установлен сервис,
- процент загрузки процессора,
- время, прошедшее с момента старта сервиса.

Для каждого сервиса доступны запуск, приостановка работы, рестарт, просмотр дополнительной информации и удаление.

7 ОБНОВЛЕНИЕ И УДАЛЕНИЕ ПРОГРАММНЫХ КОМПОНЕНТОВ

Обновление программных компонентов осуществляется путём копирования новых докер образов новых версий сервисов и указанием реквизитов этих образов в docker compose.

Удаление программных компонентов выполняется стандартными средствами и командами для работы с docker-образами.

8 АВАРИЙНЫЕ СИТУАЦИИ

8.1 Действия при отказах технических средств

Восстановление работы программного комплекса «Дальплан» при отказах технических средств осуществляется в соответствии с рекомендациями эксплуатационной документации на эти технические средства.

8.2 Восстановление программ и данных

В случае нештатного отключения программного комплекса «Дальплан» для возобновления работы всех компонентов выполните следующие действия:

1. Включить компьютер, выполняющий роль сервера программного комплекса «Дальплан»;
2. Запустите все сервисы программного комплекса «Дальплан».
3. Проверьте факт запуска всех сервисов, если какие-либо из сервисов недоступны, запустите соответствующие сервисы (см. п. 6 Проверка работоспособности).

9 ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

Сокращение	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
БПЛА	Беспилотный летательный аппарат
ПО	Программное обеспечение
ОС	Операционная система

10 ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Вектор – графический элемент, наносимый на снимок программным комплексом или пользователем для обозначения объекта; вектор может быть различного типа:

- точка,
- прямая,
- ломаная линия,
- квадрат,
- прямоугольник,
- полигон.

Вероятность – метрика, характеризующая уверенность программного комплекса в принятии решения о принадлежности объекта к одному из известных ей классов.

Дерево объектов – форма представления результатов детекции в UI программного комплекса.

Детекция – процедура автоматического поиска и классификации программным комплексом объектов на снимке.

Задание (таска/ джоба/ Jobs) – Задания на обработку и распознавание объектов на снимках, на внесение в систему KML-файлов с уже готовой информацией об объектах.

Картографические данные – любые данные с географической привязкой.

Картографическая подложка – конкретная реализация картографического слоя; примеры подложек:

- Open street map,
- Google map.

Картографический слой – базовый слой визуализации на котором размещаются картографические данные.

Класс – группа объекта одного типа или вида, детектируемых программным комплексом; класс объекта и класс вектора тождественны.

Модель – модель машинного обучения, решающая задачи детекции и классификации объектов на снимке.

Область интереса – территория на снимке или карте, выделяемая пользователем для анализа распределения детектируемых объектов по классам.

Объект – условный знак, который отображается на карте; может как принадлежать реальному миру, так и быть виртуальной сущностью.

Полигон – графический элемент, принятый в программном комплексе по умолчанию для обозначения детектируемых объектов.

Слой – набор картографических данных, объединенных по типу источника и/или функциональному назначению; примеры слоев:

картографический слой,

слой высот,

слой векторов.

Снимок – фотографическое изображение на котором программный комплекс производит поиск объектов.

GeoTIFF – Формат изображений с привязкой к местности (.tif, .tiff).

GeoJson – Формат для хранения географических структур данных, основанный на языке JavaScript Object Notation (JSON).

GPU (graphics processing unit) – Графический процессор. Это специализированный процессор, который работает с графикой, обрабатывает данные, связанные с отображением изображений, видео и 3D-графики.

NATS – Брокер сообщений.

RabbitMQ – Брокер сообщений.

UI (фронт) – Web-клиент.

Лист регистрации изменений

002-030925