

Правообладатель: Общество с ограниченной ответственностью «Интернет для жизни»  
(ООО «Интернет для жизни»)  
430005, Республика Мордовия, г. Саранск, ул. Большевистская, д. 11, офис 201,  
ОГРН 1081326002724, ИНН 1326207059

**ПЛАТФОРМА ОБРАБОТКИ ДАННЫХ  
НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

**Руководство системного программиста**

**RU.86213171.61897-01 32**

**Листов 52**

## **АННОТАЦИЯ**

В данном программном документе приведено руководство системного программиста по настройке и использованию программы «Платформа обработки данных на основе искусственного интеллекта» (далее – Платформа), предназначенной для автоматизации процессов анализа и детектирования по фотограмметрическим данным объектов, с определением их типа, занимаемой площади и физического расположения.

В разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу.

В разделе «Структура программы» приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

В разделе «Настройка программы» приведено описание действий по настройке программы на условия конкретного применения (настройка на состав технических и программных средств, выбор функций и др.).

В разделе «Проверка программы» приведено описание способов проверки, позволяющих дать общее заключение о работоспособности программы (контрольные примеры, методы прогона, результаты).

В разделе «Сообщения системному программисту» указаны тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

Оформление программного документа «Руководство системного программиста» произведено по требованиям ЕСПД (ГОСТ 19.101-77<sup>1</sup>, ГОСТ 19.103-77<sup>2</sup>, ГОСТ 19.104-78<sup>3</sup>, ГОСТ 19.105-78<sup>4</sup>, ГОСТ 19.503-79<sup>5</sup>).

---

<sup>1</sup> ГОСТ 19.101-77 ЕСПД Виды программ и программных документов

<sup>2</sup> ГОСТ 19.103-77 ЕСПД Обозначение программ и программных документов

<sup>3</sup> ГОСТ 19.104-78 ЕСПД Основные надписи

<sup>4</sup> ГОСТ 19.105-78 ЕСПД Общие требования к программным документам

<sup>5</sup> ГОСТ 19.503-79 ЕСПД Руководство системного программиста. Требования к содержанию и оформлению

## **ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ**

Перечень терминов и сокращений представлен в таблице 1 настоящего программного документа.

Таблица 1 – Перечень терминов и сокращений

|      |                                  |
|------|----------------------------------|
| БД   | База данных                      |
| ОС   | Операционная система             |
| ПКК  | Публичная кадастровая карта      |
| ПО   | Программное обеспечение          |
| СУБД | Система управления базами данных |

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| АННОТАЦИЯ.....  | 2  |
| ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ .....                            | 4  |
| 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ.....                              | 6  |
| 1.1. Назначение программы .....                                 | 6  |
| 1.2. Функции программы.....                                     | 6  |
| 1.3. Минимальный состав технических средств.....                | 9  |
| 1.4. Минимальный состав программных средств.....                | 9  |
| 1.5. Требования к персоналу (системному программисту) .....     | 10 |
| 2. СТРУКТУРА ПРОГРАММЫ .....                                    | 11 |
| 2.1. Сведения о структуре программы.....                        | 11 |
| 2.2. Сведения о составных частях программы.....                 | 11 |
| 2.3. Сведения о связях между составными частями программы ..... | 13 |
| 2.4. Сведения о связях с другими программами.....               | 20 |
| 3. НАСТРОЙКА ПРОГРАММЫ .....                                    | 21 |
| 3.1. Настройка на состав технических средств .....              | 21 |
| 3.2. Настройка на состав программных средств.....               | 21 |
| 3.2.1. Установка программы .....                                | 21 |
| 3.2.1.1. Запуск Платформы.....                                  | 22 |
| 4. ПРОВЕРКА ПРОГРАММЫ.....                                      | 24 |
| 4.1. Описание способов проверки .....                           | 24 |
| 4.2. Методы прогона .....                                       | 25 |
| 4.2.1. Проверка работоспособности программы.....                | 37 |
| 4.2.2. Проверка на сообщение об ошибке .....                    | 42 |
| 5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ.....                       | 51 |
| Лист регистрации изменений.....                                 | 52 |

## **1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**

### **1.1. Назначение программы**

Платформа предназначена для осуществления мониторинга, семантического и пространственного анализа на основе технологии искусственного интеллекта и алгоритмов машинного обучения, с целью ее внедрения в отрасли экономики и народного хозяйства, такие как: земельно-имущественные отношения, сельское хозяйство, лесное хозяйство и другие, для получения необходимых данных, сокращения экономических потерь и защиты прав и законных интересов граждан, бизнеса, государства, в условиях цифровой экономики.

Платформа обеспечивает автоматизацию процессов детектирования объектов по различным видам входных данных с использованием технологии искусственного интеллекта. Результаты детектирования отображаются в геоинформационной системе.

Платформа позволяет формировать аналитическую информацию и выгружать атрибутивную информацию по объектам. Сформированные отчеты с применением сквозных цифровых технологий могут служить основой для повышения эффективности и результативности принимаемых управленческих решений.

### **1.2. Функции программы**

Программа «Платформа обработки данных на основе искусственного интеллекта» реализует следующие функции:

- 1) Загрузка фотограмметрических данных (входные данные для работы с Платформой).
- 2) Управление процессом обработки фотограмметрических данных.
- 3) Возможность создания очереди обработки фотограмметрических данных.

- 4) Создание и настройка конфигурации обработки фотограмметрических данных.
- 5) Детектирование на фотограмметрических данных объектов, согласно классификатору.
- 6) Отображение результатов обработки фотограмметрических данных на геопортале.
- 7) Возможность внесения изменений в результаты обработки фотограмметрических данных.
- 8) Выгрузка результатов обработки фотограмметрических данных в формате \*.geojson (выходные данные).
- 9) Просмотр статистики работы Платформы.
- 10) Возможность организации межсистемного взаимодействия посредством API.
- 11) Просмотр подробной информации о выбранном процессе обработки фотограмметрических данных.
- 12) Импорт и удаление пользовательских векторных данных (слоев) в формате \*.geojson, \*.shp, \*.zip в процесс обработки фотограмметрических данных.
- 13) Указание системы координат для каждого загружаемого векторного слоя в формате \*.geojson, \*.shp, \*.zip в процесс обработки фотограмметрических данных.
- 14) Просмотр импортированных пользовательских векторных данных в процесс обработки фотограмметрических данных на геопортале.
- 15) Просмотр этапов конвейера обработки фотограмметрических данных в процессе обработки фотограмметрических данных.
- 16) Фильтрация процессов обработки фотограмметрических данных по объектам детектирования.
- 17) Фильтрация процессов обработки фотограмметрических данных по статусам обработки.

18) Управление фотограмметрическими данными, используемые при запуске процесса обработки фотограмметрических данных.

19) Просмотр подробной информации об источнике данных.

20) Просмотр подробной информации о конфигурации, используемой при запуске процесса обработки фотограмметрических данных.

21) Загрузка и удаление документов табличного вида в формате \*.xlsx, \*.csv, \*.xml, следующих типов: «Данные по земельным участкам из ФНС», «Права собственности», «Выгрузки из кадастрового плана».

22) Управление анализом данных по выявленным объектам с использованием данных из Публичной кадастровой карты и данных из загруженных документов табличного вида в формате \*.xlsx, \*.csv, \*.xml, следующих типов: «Данные по земельным участкам из ФНС», «Права собственности», «Выгрузки из кадастрового плана».

23) Визуализация результатов анализа данных на тематических дашборд-панелях в информационно-аналитической подсистеме.

24) Выгрузка сформированных аналитических отчетов (результат анализа данных) в формате \*.geojson, \*.xlsx.

25) Выбор одного из следующих типов карты, в качестве картографической основы: «Публичная кадастровая карта», «Google Maps», «Mapbox», «OpenStreet Maps».

26) Выбор одного или нескольких из следующих векторных слоев для отображения административно-территориального деления Российской Федерации: «Федеральные округа», «Субъекты РФ», «Районы».

27) Просмотр атрибутивных данных выявленных объектов, входящих в результат обработки фотограмметрических данных.

28) Управление дополнительными атрибутивными данными выявленных объектов.

29) Внесение изменений в выбранный векторный слой на геопортале.



30) Измерение расстояния между объектами в выбранном векторном слое на геопортале.

31) Поиск объектов по кадастровому номеру в рамках выбранного векторного слоя на геопортале.

32) Скачивание изображений рабочей области интерфейса геопортала, в которой отображены растровые и векторные данные, для последующей печати.

33) Управление ролевой моделью и полномочиями решения.

34) Разграничение прав доступа к модулям и функциям Платформы, реализованное с помощью ролевой модели.

35) Наделение пользователей программы полномочиями, в зависимости от назначенных им ролей.

### **1.3. Минимальный состав технических средств**

Рекомендуемые требования к аппаратной части, на которой планируется развёртывание и функционирование Платформы:

- Процессор: 16x CPU 3.5 GHz.
- Оперативная память: 64 Gb.
- Видео карта: NVIDIA GeForce RTX 3080, 10 Gb.
- SSD накопитель: 1 Tb SSD.
- Жесткий диск: 18 Tb HDD.
- Сетевой адаптер: Gigabit Ethernet 1000 Mbit.

### **1.4. Минимальный состав программных средств**

Системные программные средства, используемые для эксплуатации программы «Платформа обработки данных на основе искусственного интеллекта», должны быть представлены локализованной версией ОС Windows 7/8/10 или Unix-подобным системам.

Эксплуатация программы должна осуществляться в одном из следующих интернет-браузерах:

- Google Chrome (версия 74.x и выше);
- Yandex Browser (версия 19.6.x и выше);
- Mozilla Firefox (версия 67.x и выше);
- Opera (версия 60.x и выше).

Для работы с документами необходимо использовать офисное приложение «Microsoft Excel».

### **1.5. Требования к персоналу (системному программисту)**

Системный программист должен обладать следующими навыками:

- Опыт разработки ПО на языках программирования Python и Java;
- Опыт разработки микросервисных приложений;
- Базовые навыки администрирования Linux;
- Базовые навыки работы с инструментами контейнеризации и оркестрации;
- Базовые навыки работы с геоданными.

## 2. СТРУКТУРА ПРОГРАММЫ

### 2.1. Сведения о структуре программы

Программа «Платформа обработки данных на основе искусственного интеллекта» состоит из одной запускаемой формы.

### 2.2. Сведения о составных частях программы

Платформа содержит в себе составляющие, представленные в таблице 2.

Таблица 2 – Сведения о составных частях программы

| Наименование docker-контейнера                                       | Соотношение docker-контейнера к составляющей Платформы  |
|--|---|
| <b>Набор инфраструктурных микросервисов, компонентов и утилит</b>    |   |
| Camunda EE   | Платформа автоматизации Camunda EE  |
| PostgreSQL 12  | Реляционная СУБД PostgreSQL 12  |
| Directory Cleaner  | Микросервис очистки горячего хранилища  |
| Mongo  | NoSQL СУБД MongoDB  |
| Nginx  | Компонент «Web-сервер Nginx»  |
| Portainer  | Графическая оболочка управления docker-контейнерами   |
| Core   | Компоненты ядра Платформы   |
| Core-versioning  | Сервис обеспечения автоматического сохранения версий сущностей Платформы при их изменениях                    |
| Config   | Сервис обеспечения хранения и оперативного доступа к конфигурационным параметрам docker-контейнеров Платформы |
| Discovery  | Сервис оркестровки  |
| Numerator  | Сервис формирования последовательных номеров (идентификаторов) сущностей Платформы                            |
| Routing  | Сервис агрегации и маршрутизации API  |
| Filestorage  | Сервис хранения основных файлов Платформы   |
| <b>Набор инфраструктурных микросервисов, компонентов и утилит</b>    |   |
| Rabbit-MQ  | Утилита «Брокер сообщений (событий)»  |
| Mongo  | NoSQL СУБД MongoDB  |
| <b>Сервисы, микросервисы, компоненты и утилиты общего назначения</b> |   |
| Att Data Asignation  | Микросервис присвоения атрибутивных данных  |
| Environment Setup  | Микросервис настройки рабочего окружения процесса обработки   |
| File Uploading   | Микросервис выгрузки результатов во внешнюю БД  |
| Geojsonmerging   | Микросервис объединения geojson-файлов  |
| Product Preparing  | Микросервис загрузки тайлов из продактов  |
| Tile Downloader  | Микросервис загрузки тайлов   |

|  |   |
|--|---|
| Gdaltransform  | Сервис преобразований систем координат в WGS 84   |
| Aip-parser   | Сервис обеспечения парсинга файлов табличного вида  |
| <b>Сервис обработки фотограмметрических данных</b>   |   |
| Masks Blending   | Микросервис смешивания масок  |
| Masks Binarization   | Микросервис бинаризации масок   |
| Masks Splitting  | Микросервис разделения масок  |
| Segmentation CPU   | Микросервис сегментации (Mask R-CNN, CPU-версия)  |
| Segmentation GPU   | Микросервис сегментации (Mask R-CNN, GPU-версия)  |
| Segmentation Multiband CPU   | Микросервис сегментации для работы с многоканальными изображениями (Mask R-CNN, CPU-версия)     |
| Segmentation Multiband GPU   | Микросервис сегментации для работы с многоканальными изображениями (Mask R-CNN, GPU-версия)     |
| Segmentation Deeplab CPU   | Микросервис сегментации (DeepLab-V3Plus, CPU-версия)  |
| Segmentation Deeplab GPU   | Микросервис сегментации (DeepLab-V3Plus, GPU-версия)  |
| Segmentation Deeplab Multiband CPU   | Микросервис сегментации для работы с многоканальными изображениями (DeepLab-V3Plus, CPU-версия) |
| Segmentation-Deeplab Multiband GPU   | Микросервис сегментации для работы с многоканальными изображениями (DeepLab-V3Plus, GPU-версия) |
| Segmentation Unet CPU  | Микросервис сегментации и классификации (U-Net, CPU-версия)                                     |
| <b>Сервис обработки фотограмметрических данных</b>   |   |
| Segmentation Unet GPU  | Микросервис сегментации и классификации (U-Net, GPU-версия)                                     |
| Segmentators Manager   | Сервис управления микросервисами сегментации  |
| IC Displacing  | Микросервис генерации тайлов, смещенных по сторонам света                                       |
| IC Filtering   | Микросервис межкардинальной фильтрации сегментированных объектов                                |
| IC Redisplacing  | Микросервис обратного смещения объектов, отфильтрованных алгоритмом межкардинальной фильтрации  |
| <b>Набор микросервисов обработки данных геопространственными алгоритмами</b>   |   |
| QGIS Executor  | Микросервис выполнения QGIS-алгоритмов  |
| Orthogonalization  | Микросервис расчета ортогонализированных полигонов  |
| Vectorization  | Микросервис векторизации тайловых масок   |
| <b>Набор микросервисов семантического и пространственного анализа</b>  |   |
| Fields Calculating   | Микросервис расчета геометрических атрибутов  |
| Pentaho Carte  | Сервис запуска и мониторинга преобразований и заданий Pentaho                                   |
| Pentaho Server   | Сервис Pentaho Data Integration   |
| Analytical Service   | Микросервис подготовки данных для аналитики   |
| <b>Набор микросервисов, компонентов и утилит, организующих внутренний доступ к данным и обеспечивающих их хранение в Платформе</b> |   |
| WideTMS Frontend   | Сервер тайлов   |
| WideTMS Tiler  | Сервис-генератор тайлов   |
| PKK Proxy  | Микросервис проксирования запросов к TMS серверу ПКК  |
| PostgreSQL 12 with PostGIS 3   | Реляционная СУБД PostgreSQL 12 с расширением PostGIS 3  |
| Geoserver 2.17   | Сервер распространения геоданных Geoserver  |
| Mapproxy   | Сервер проксирования внешних геоданных Mapproxy   |
| Product Preparer   | Микросервис подготовки продактов  |

| <b>Набор микросервисов, компонентов и утилит, организующих внутренний доступ к данным и обеспечивающих их хранение в Платформе</b> |   |
|--|---|
| Redis  | СУБД Redis  |
| RabbitMQ   | Брокер сообщения RabbitMQ                                   |
| Result Uploading   | Микросервис выгрузки результатов во внутреннюю БД           |
| PKK Data Provider  | Сервис доступа к данным публичной кадастровой карты         |
| <b>API для организации межсистемного взаимодействия</b>  |   |
| EODAG Server   | Сервис поиска и загрузки EO-продуктов                       |
| Postgis API Layer  | Сервис доступа к PostGIS через API                          |
| Geoserver API Layer  | Сервис доступа к Geoserver через API                        |
| Data Access Layer  | Слой доступа к данным                                       |
| <b>Сервис информационной безопасности</b>  |   |
| KPP  | Сервис информационной безопасности                          |
| <b>Геопортал</b>   |   |
| AIP-static   | Web-клиент Платформы  |
| <b>Система мониторинга</b>   |   |
| Alertmanager   | Система оповещения  |
| Grafana  | Графическое отображение параметров мониторинга              |
| Blackbox-exporter  | Система мониторинга работоспособности компонентов Платформы |
| Cadvisor   |   |
| Node-exporter  |   |
| Prometheus   |   |

### 2.3. Сведения о связях между составными частями программы

Решение реализовано с соблюдением принципов микросервисной архитектуры. Схематичное представление архитектуры Платформы представлено на рисунке 2.3.1.

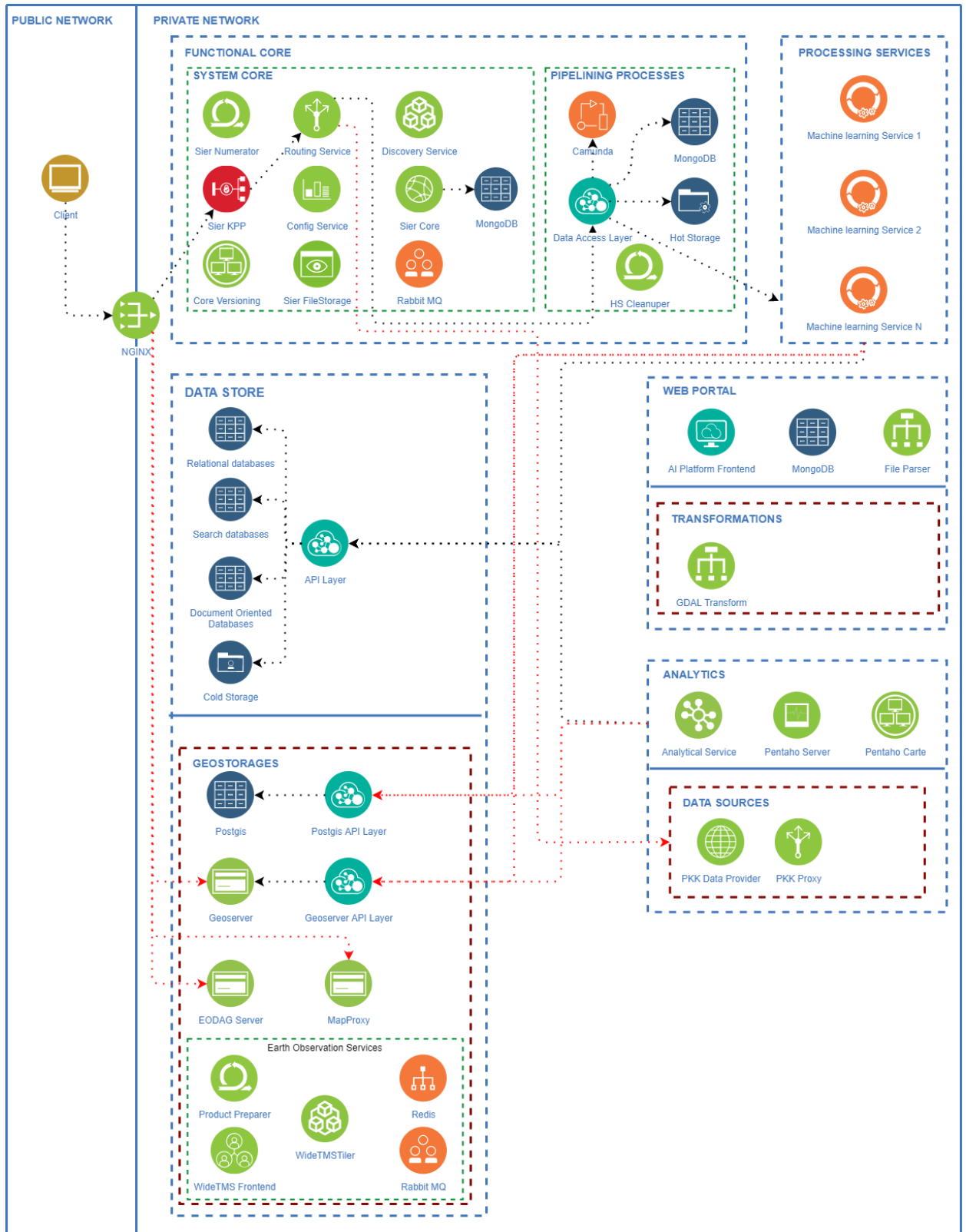


Рисунок 2.3.1 – Схема архитектуры Платформы

В состав Платформы включены следующие функциональные подсистемы, с функционирующими в них соответствующими программными

компонентами:

1. Функциональное ядро (FUNCTIONAL CORE).
2. Вычислительные сервисы (PROCESSING SERVICES).
3. Подсистема хранения данных (DATA STORE).
4. Информационно-аналитическая подсистема (ANALYTICS).
5. Подсистема визуализации данных (WEB PORTAL).

Для обеспечения взаимодействия клиентов Платформы с её программными компонентами используется web-сервер Nginx (NGINX). Организация сервера Платформы на базе web-сервера Nginx позволила обеспечить следующие функциональные возможности:

- Организация обработки http/https запросов, поступающих от клиентов в Платформу;
- Защищенная передача данных по протоколам SSL/TLS;
- Обслуживание статичных запросов в рамках функционирования программных компонентов Платформы;
- Автоматический учет файлов, хранящихся на сервере Платформы;
- Кэширование системных файлов;
- Масштабирование нагрузки при функционировании программных компонентов Платформы;
- Ведение журнала обращений к ресурсам Платформы;
- Обеспечение всех базовых потребностей при администрировании сервера Платформы;
- Гибкая настройка конфигураций сервера Платформы.

Администрирование web-сервера Nginx в рамках функционирования Платформы осуществляется на уровне операционной системы Ubuntu-20.04.

В состав функционального ядра Платформы входят следующие функционально-логические блоки:

- 1) Системное ядро (SYSTEM CORE).

## 2) Модуль управления конвейерной обработки данных (PIPELINING PROCESSES).

Системное ядро предназначено для обеспечения процессов функционирования Платформы и реализации взаимодействия всех программных компонентов решения в рамках интеллектуальной обработки данных.

Модуль управления конвейерной обработки данных предназначен для обеспечения выполнения процессов конвейерной обработки данных, смоделированных в виде схем в нотации BPMN v. 2.0.

Назначение вычислительных сервисов в рамках функционирования Платформы – обеспечение автоматизированных процессов обработки множества типов данных из различных источников с использованием технологии искусственного интеллекта и моделей машинного обучения. Состав вычислительных сервисов (Machine learning Service 1, 2, N) масштабируемый, и разрабатывается под решение частных предметных задач.

Подсистема хранения данных предназначена для обеспечения возможности получения, хранения и управления множества типов данных из различных источников, доступ к которым может быть организован посредством API Layer – сервис (слой) доступа к данным посредством API.

Информационно-аналитическая подсистема предназначена для обеспечения возможности формирования, отображения и выгрузки аналитических данных с использованием BI-инструментов.

Подсистема визуализации данных предназначена для обеспечения возможности инициализации процессов интеллектуальной обработки данных и визуализацию их результатов.

Взаимодействие между модулем управления конвейерной обработки (сервис DAL) и вычислительными сервисами происходит посредством REST API. DAL предоставляет вычислительному сервису следующие методы:



- `fetchAndLock` – метод позволяет получить вычислительную задачу и привязать ее к исполнителю;
- `complete` – метод позволяет уведомить модуль управления конвейерной обработкой об успешном завершении вычислительной задачи;
- `failure` – метод позволяет уведомить модуль управления конвейерной обработкой о завершении вычислительной задачи с ошибкой;
- `telemetry` – метод позволяет вычислительному сервису отправить информацию о ходе выполнения вычислительной задачи (телеметрия) и в ответ получить информацию о работе конвейера или корректирующую команду (пауза, отмена задачи и т.п).

Для управления конвейером модуль управления предоставляет пользователю следующие методы:

- `POST /start` – позволяет пользователю запустить процесс обработки. Позволяет в качестве аргумента передавать конфигурацию конвейера процесса обработки;
- `POST /pause` – позволяет поставить на паузу запущенный процесс обработки;
- `POST /resume` – позволяет снять с паузы запущенный процесс обработки.

Для реализации вычислительного сервиса необходимо:

- 1) Реализовать вычислительную библиотеку, унаследованную от `BaseProcessor`.
- 2) По шаблону создать проект вычислительного сервиса, состоящий из:
  - a) директории `microservice/code`;
  - b) файл `microservice/code/app.py`;
  - c) файл `Dockerfile`;
  - d) файл `VERSION`;
  - e) файл `entrypoint`;
  - f) папка `Jenkins`.

3) В файле `app.py` необходимо реализовать класс `App`, унаследованный от `BaseApp`. В нем должно быть назначено поле `PROCESSOR` – класс, унаследованный от `BaseProcessor` в пункте 1.

4) В директории `microservice/code` необходимо разместить вспомогательные скрипты.

5) В файле `Dockerfile` описать последовательность сборки сервиса.

6) В папку `Jenkins` добавить конфигурацию автосборки.

На рисунке 2.3.2 показано взаимодействие вычислительных сервисов с `Samunda` и `Data Access Layer`. На рисунке 2.3.3 показано взаимодействие компонентов вычислительного сервиса.

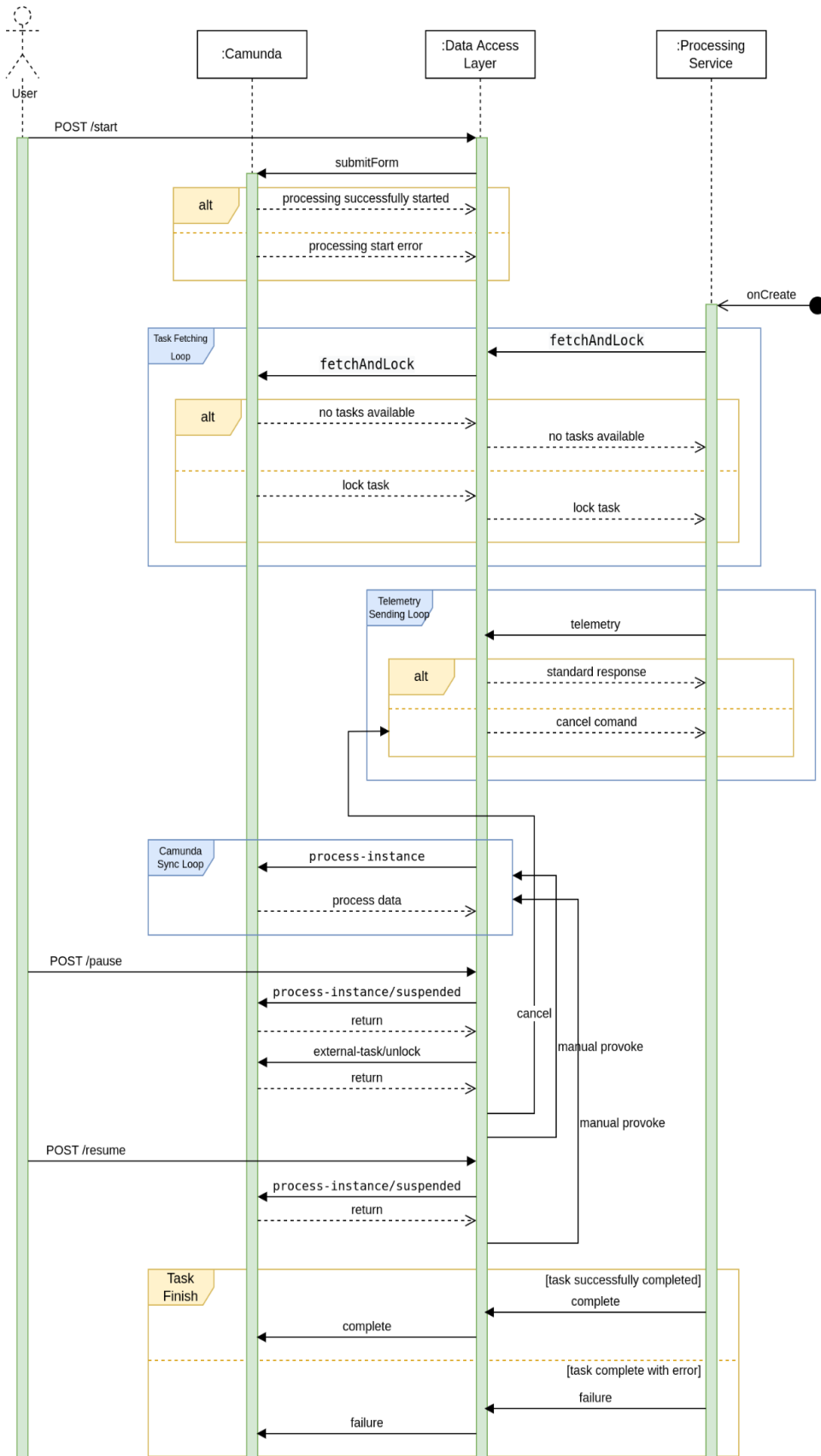


Рисунок 2.3.2 – Схема взаимодействия вычислительных сервисов с Camunda и Data Access Layer

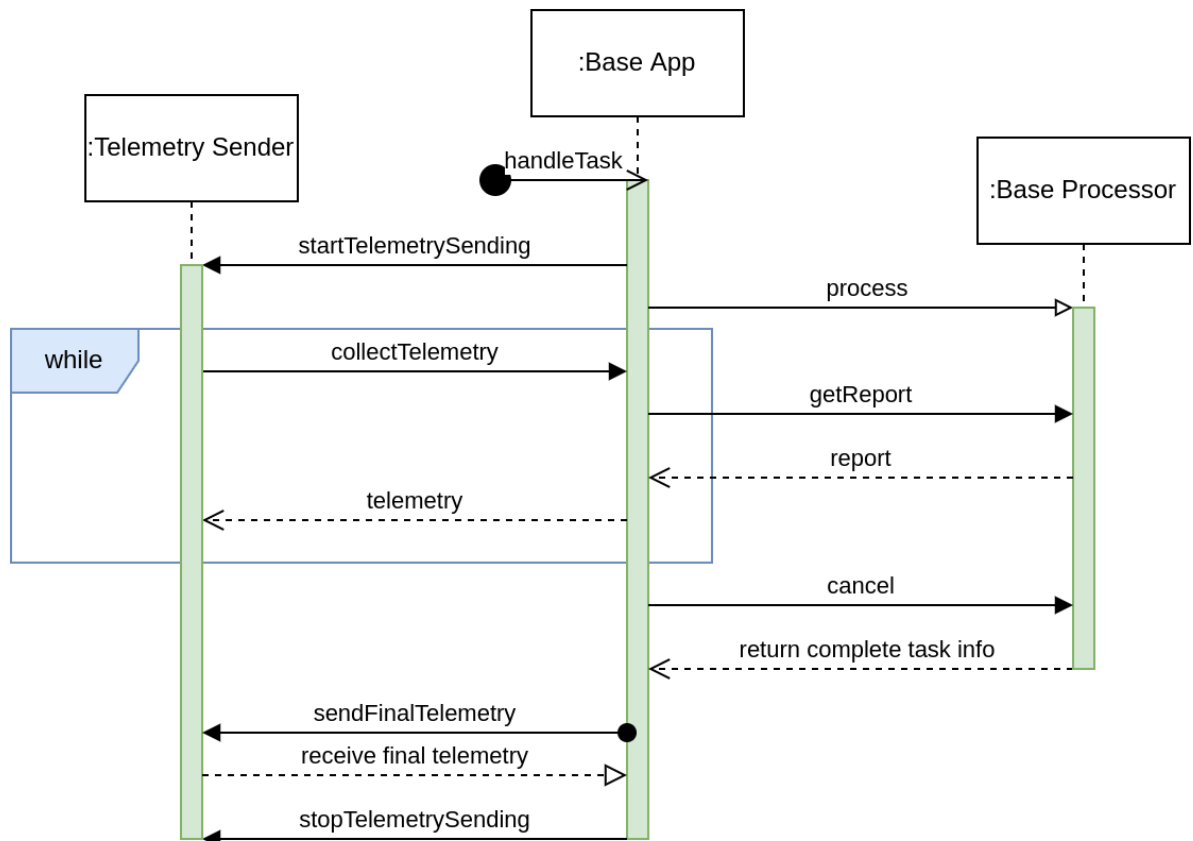


Рисунок 2.3.3 – Схема взаимодействия компонентов вычислительного сервиса

## 2.4. Сведения о связях с другими программами

Для процессов обработки, аналитики и визуализации Платформа использует данные следующих сервисов (внешних систем):

- 1) Публичная кадастровая карта.
- 2) Картографический сервис Mapbox.
- 3) Картографический сервис OpenStreetMap.
- 4) Спутниковые снимки Sentinel-2, полученные безвозмездно у различных провайдеров.

## **3. НАСТРОЙКА ПРОГРАММЫ**

### **3.1. Настройка на состав технических средств**

Платформу или ее программные компоненты возможно развернуть на операционных системах семейства Linux. Гарантируется работоспособность Платформы на серверах, функционирующих под ОС Ubuntu 18.04.

Перед развертыванием Платформы необходимо развернуть на сервере следующее программное обеспечение (ПО):

- 1) PostgreSQL 12.
- 2) MongoDB 4.4.3 Community.
- 3) Docker 20.10.3.
- 4) Nginx 1.18.
- 5) Portainer.

Дальнейшая настройка производится при помощи скриптов запуска и конфигураций развертывания.

В Платформе необходимо настроить директорию для долгосрочного хранения растровых геоданных и директорию для хранения временных файлов процессов обработки.

### **3.2. Настройка на состав программных средств**

Развертывание микросервисов Платформы производится при помощи скриптов запуска и конфигураций развертывания. Конфигурацию Nginx необходимо поместить в стандартную директорию Nginx.

#### **3.2.1. Установка программы**

Перед развертыванием Платформы (в частности, баз данных и микросервисов Платформы) предполагается, что:

- 1) на серверах приложений уже установлен сервис «Docker», и они

объединены в кластер «Docker Swarm»;

2) для скачивания образов с серверов приложений открыт доступ в интернет к ресурсам <https://dockerhub.com> и <https://newharbor.evolenta.ru>.

Выполнить вход в реестр [newharbor.evolenta.ru](https://newharbor.evolenta.ru) можно с помощью команды: `docker login newharbor.evolenta.ru` под соответствующей учетной записью.

Также, в целях обеспечения функционирования Платформы и ее эксплуатации, требуется развернуть базы данных, используя SQL-скрипты под ОС Ubuntu. При запуске SQL-скриптов на других ОС, возможно, потребуются адаптация. Требуется установить необходимые для работы Платформы СУБД – MariaDB, MongoDB, MongoDB (для PDPS стеков), PostgreSQL.

### 3.2.1.1. Запуск Платформы

Развертывание осуществляется посредством yaml-файлов docker-compose. В них прописаны инструкции по настройке и запуску микросервисов, а именно имена микросервиса и образа; перенаправление портов для доступа к контейнерам из внешней сети; монтируемые внутрь контейнеров папки и конфигурационные файлы; количество запускаемых реплик; ресурсы, выделяемые контейнеру при старте; переменные окружения и т.п. По выполняемым функциям микросервисы разделены на группы, составляющие стеки, в частности:

#### 1. Стек AIR.

Файл инструкций [docker-compose.air\\_stack.yml](#) – системное ядро, предназначено для обеспечения процессов функционирования Платформы и реализации взаимодействия всех программных компонентов решения в рамках интеллектуальной обработки данных.

#### 2. Стек Geostorages.

Файл инструкций [docker-compose.geostorages\\_stack.yml](#) – подсистема

хранения геоданных данных, предназначена для обеспечения возможности получения, хранения и управления множеством типов геоданных из различных источников.

### 3. Стеки PDPS-management и PDPS-processing.

Файлы инструкций docker-compose.pdps-management\_stack.yml и docker-compose.pdps-processing\_stack.yml – модуль управления конвейерной обработки данных, предназначен для обеспечения выполнения процессов конвейерной обработки данных, смоделированных в виде схем в нотации BPMN v. 2.0.

### 4. Стек Pentaho.

docker-compose.pentaho\_stack.yml – информационно-аналитическая подсистема, предназначена для обеспечения возможности формирования, отображения и выгрузки аналитических данных с использованием BI-инструментов.

5. Стек Monitoring – система мониторинга, состоящая из подсистемы сбора метрик Prometheus и системы визуализации Grafana.

Запуск стеков производится с помощью bash-скриптов:

1. Стек AIP – run\_aip\_stack.sh, запускается в первую очередь.
2. Стек Geostorages – run\_geostorages\_stack.sh.
3. Стек PDPS-management – run\_pdps-management\_stack.sh.
4. Стек PDPS-processing – run\_pdps-processing\_stack.sh.
5. Стек Pentaho – run\_pentaho\_stack.sh.
6. Стек Monitoring – run\_monitoring\_stack.sh.

Для проверки успешности запуска Платформы используются следующие команды:

1. docker stack ls – проверка запуска стеков.

Примерный вывод команды представлен на рисунке 3.2.1.5.1

| NAME            | SERVICES | ORCHESTRATOR |
|-----------------|----------|--------------|
| air             | 14       | Swarm        |
| file-server     | 1        | Swarm        |
| geostorages     | 13       | Swarm        |
| pdps-management | 2        | Swarm        |
| pdps-processing | 34       | Swarm        |
| pentaho         | 2        | Swarm        |

Рисунок 3.2.1.5.1 – Примерный вывод команды проверки запуска стеков  
docker stack ls

В выводе должны быть перечислены все вышеуказанные стеки с соответствующим количеством микросервисов.

2. docker stack services <имя стека> – проверка запущенных микросервисов и их реплик.

Например, docker stack services air.

Примерный вывод команды представлен на рисунке 3.2.1.5.2

| ID            | NAME                        | MODE       | REPLICAS | IMAGE  | PORTS                                |
|---------------|-----------------------------|------------|----------|--|--------------------------------------|
| c4m21mmopdc   | air-air-parser              | replicated | 1/1      | newharbor.evoluta.ru/air/air-parser:36           |                                      |
| d03frz5zqkfu  | air-air-static              | replicated | 1/1      | newharbor.evoluta.ru/air/air-static:132          |                                      |
| fvf7mopna1wd  | air-analiticalservice       | replicated | 1/1      | newharbor.evoluta.ru/air/analyticalservice:0.4.0 |                                      |
| ra0zgilj1v18  | air-cloud-config-service    | replicated | 1/1      | newharbor.evoluta.ru/air/config:v9               |                                      |
| p5cspb1oqk5c  | air-cloud-discovery-service | replicated | 1/1      | newharbor.evoluta.ru/air/discovery:v9            |                                      |
| mds16kmzysgq  | air-cloud-routing-service   | replicated | 1/1      | newharbor.evoluta.ru/air/routing:v8              | *:8080->8080/tcp, *:9080->9080/tcp   |
| qk7cmbrs1f1fm | air-gdaltransform           | replicated | 1/1      | newharbor.evoluta.ru/air/gdaltransform:v2        | *:8082->8080/tcp                     |
| zfr6464dmam3  | air-pkkprovider             | replicated | 1/1      | newharbor.evoluta.ru/air/pkkprovider:0.4.9       |                                      |
| t26bqy18lhka  | air-sier-core               | replicated | 1/1      | newharbor.evoluta.ru/air/core:v679               |                                      |
| g139zwrxosx   | air-sier-core-versioning    | replicated | 1/1      | newharbor.evoluta.ru/air/core-versioning:v386    |                                      |
| q5q1xpe21czz  | air-sier-filestorage        | replicated | 1/1      | newharbor.evoluta.ru/air/filestorage:v43         |                                      |
| x3j33eqe419j  | air-sier-kpp                | replicated | 1/1      | newharbor.evoluta.ru/air/kpp:v48-mysql           |                                      |
| m35sx6dqd71v  | air-sier-numerator          | replicated | 1/1      | newharbor.evoluta.ru/air/numerator:v8            | *:81->80/tcp, *:9192->9192/tcp       |
| kctwuakpqd9h  | air-sier-rabbitmq           | replicated | 1/1      | rabbitmq:3-management                            | *:5672->5672/tcp, *:15672->15672/tcp |

Рисунок 3.2.1.5.2 – Примерный вывод команды проверки запуска стеков  
docker stack services <имя стека>

В выводе должны быть перечислены все микросервисы, а количество запущенных реплик должно равняться количеству заявленных (например, 1/1).

## 4. ПРОВЕРКА ПРОГРАММЫ

### 4.1. Описание способов проверки

Работоспособность Платформы проверяется ручным тестированием всех компонентов Платформы.



## 4.2. Методы прогона

В Платформе можно применять два метода прогона ПО:

1) Пробный прогон (dry running) – предварительная проверка программы, машинных средств и/или тестовых данных до их рабочего использования.

2) Контрольный прогон (benchmark run, benchmark running) – выполнение программы для определения рабочих характеристик ЭВМ или проверки самой программы.

Для выполнения пробного прогона необходимо:

1) Авторизоваться в Платформе.

Для авторизации в Платформе необходимо в браузере открыть интерфейс Платформы, по доменному имени Платформы или по ip-адресу, если доменное имя не настроено. Платформа автоматически проверит статус авторизации и, если она отсутствует, то перенаправит в интерфейс авторизации (рисунок 4.2.1).

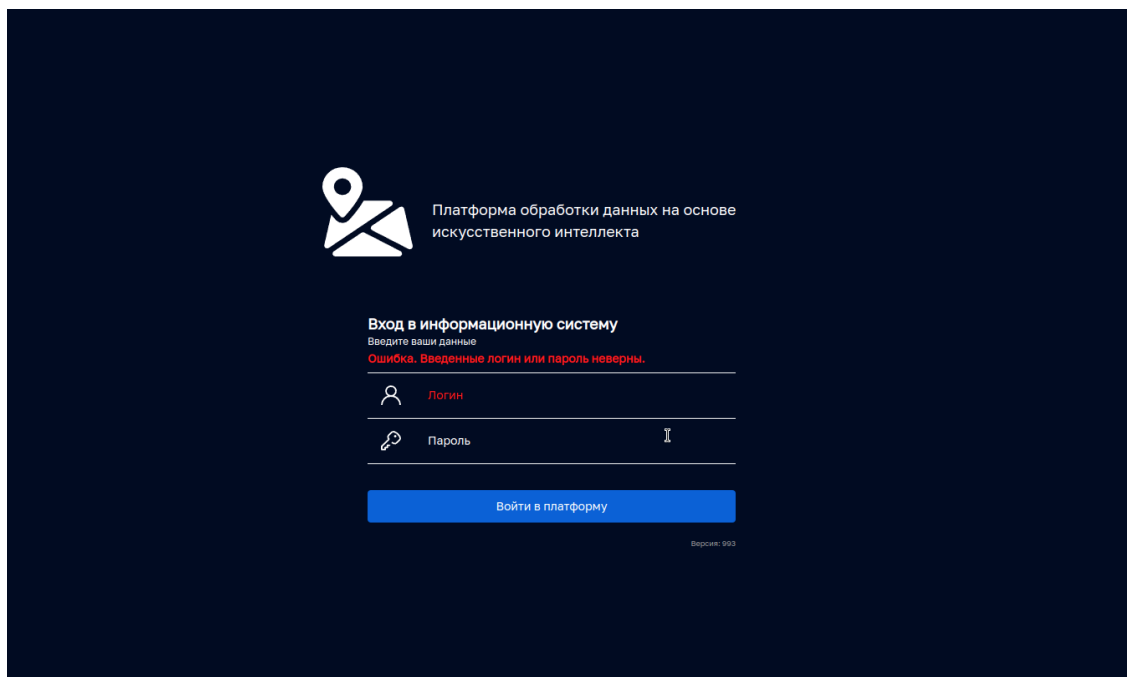


Рисунок 4.2.1 – Интерфейс авторизации Платформы

Если при авторизации возникают ошибки, при этом данные авторизации корректны, необходимо произвести проверку следующих сервисов:

- 1) MongoDB.
- 2) KPP.

Если авторизация прошла успешно, то в интерфейсе Платформы отобразится интерфейс, содержащий список организаций, доступных пользователю, для работы в Платформе (рисунок 4.2.2).

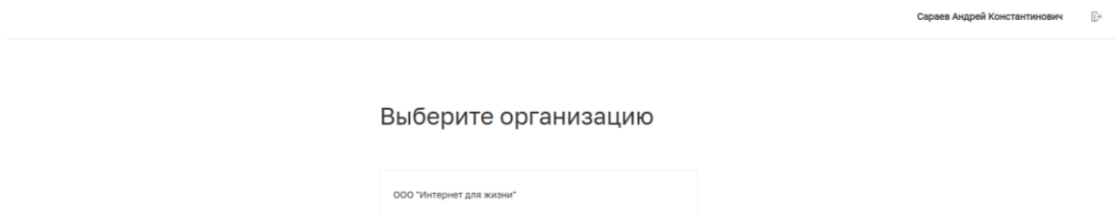
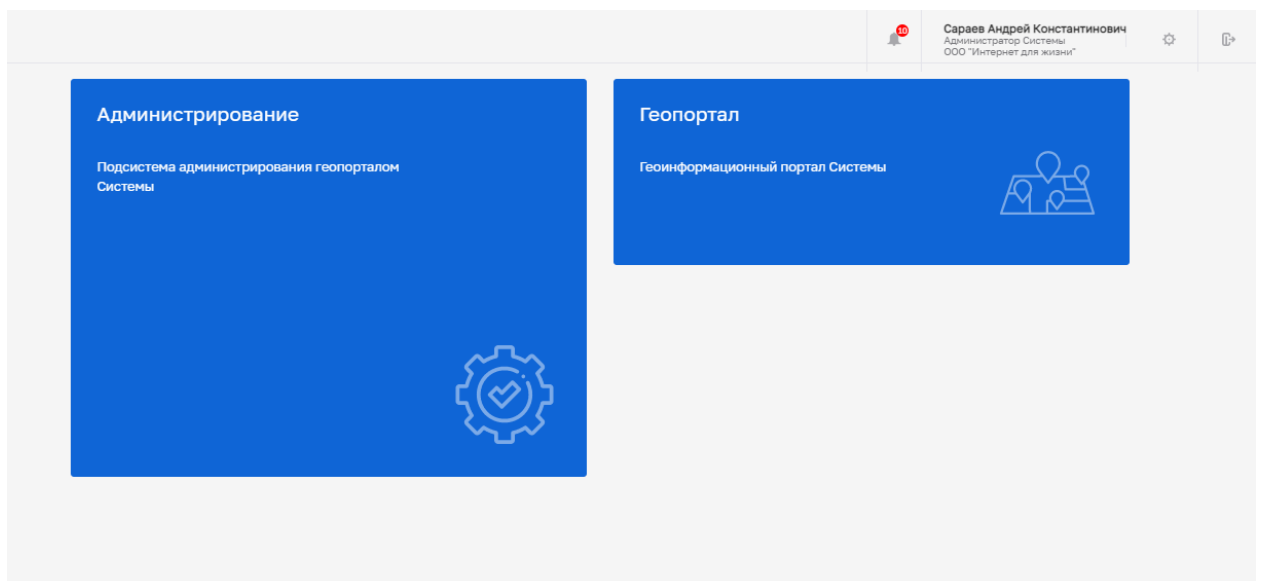


Рисунок 4.2.2 – Интерфейс Платформы с выбором организации

После выбора организации в интерфейсе Платформы отобразится интерфейс главного меню (рисунок 4.2.3).



## Рисунок 4.2.3 – Интерфейс главного меню

## 2) Загрузить источник данных.

Для этого необходимо в интерфейсе главного меню выбрать пункт «Геопортал», перейти в раздел «Интеллектуальная обработка», выбрать пункт «Источники данных» и нажать кнопку «Добавить источник» (рисунок 4.2.4).

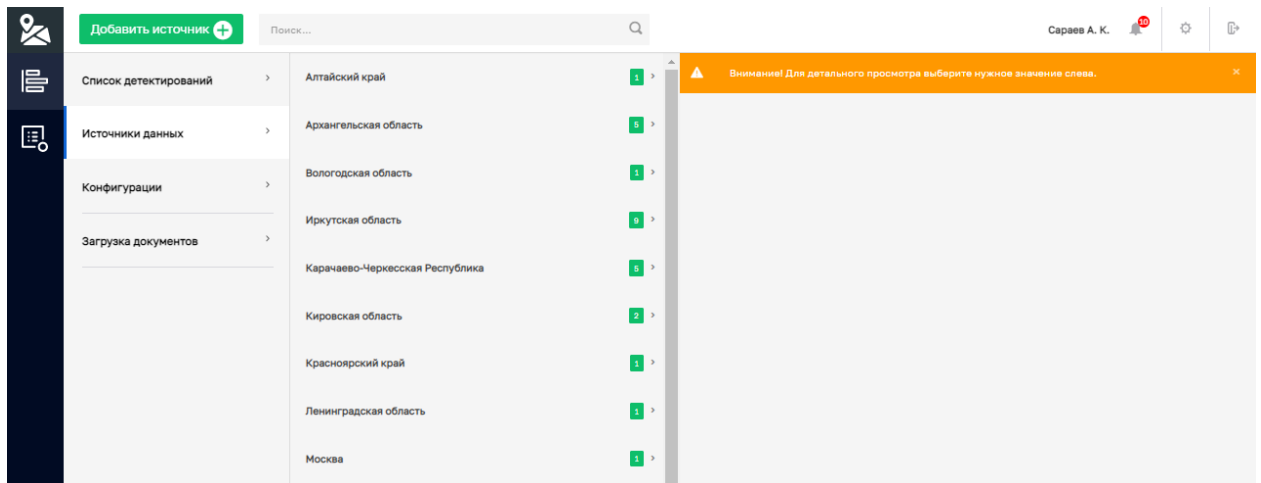


Рисунок 4.2.4 – Интерфейс работы с источниками пространственных данных

В интерфейсе загрузки пространственных данных (рисунок 4.2.5) необходимо сконфигурировать источник и загрузить tiff-файлы. Для проверки работоспособности Платформы желательно использовать tiff-файлы, которые уже обрабатывались на других экземплярах Платформы и результат обработки которых уже известен. При загрузке tiff-файлов, возможно, потребуется указать систему координат (рисунок 4.2.6).

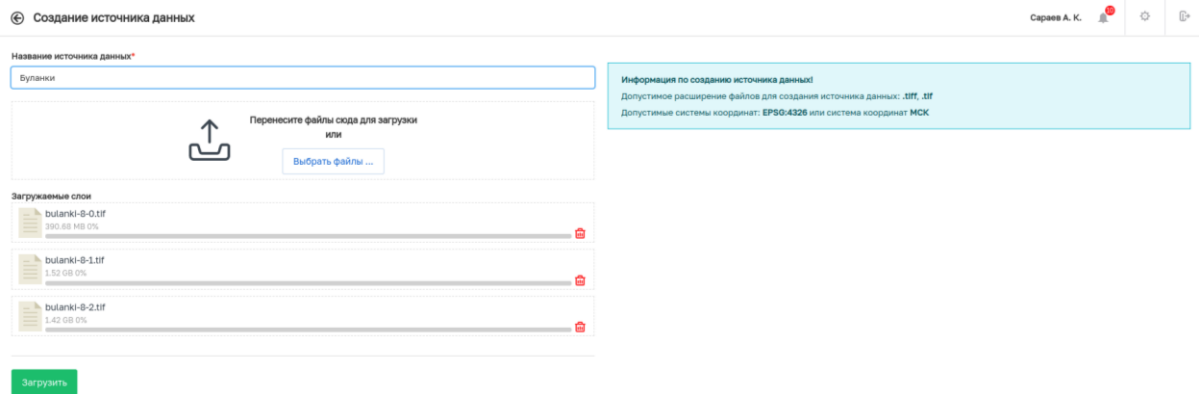


Рисунок 4.2.5 – Интерфейс загрузки пространственных данных

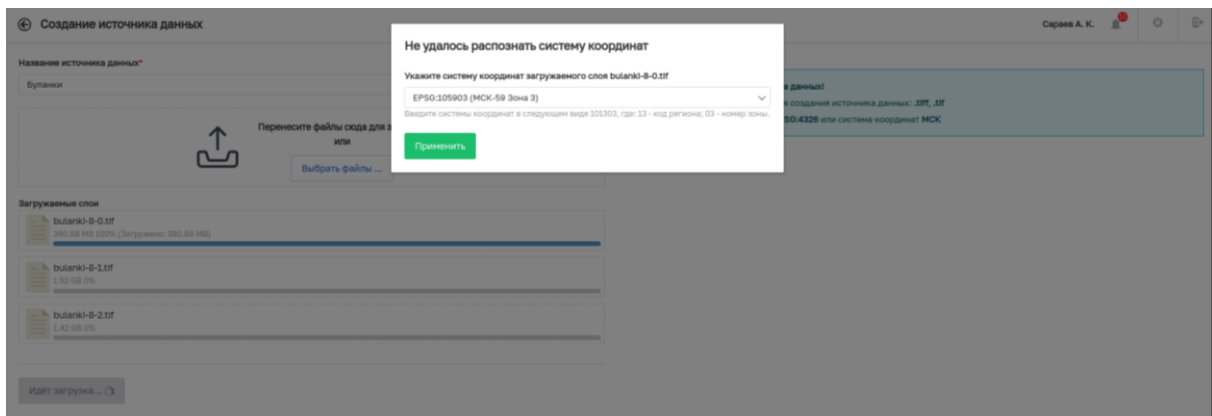


Рисунок 4.2.6 – Интерфейс установки системы координат

В случае успешной загрузки пространственных данных (рисунок 4.2.7), Платформа покажет соответствующее сообщение и предоставит ссылку на интерфейс просмотра загруженных пространственных данных (рисунок 4.2.8). Если при загрузке возникают ошибки, необходимо произвести проверку следующих сервисов:

- 1) Geoserver API Layer.
- 2) Geoserver.
- 3) KPP.

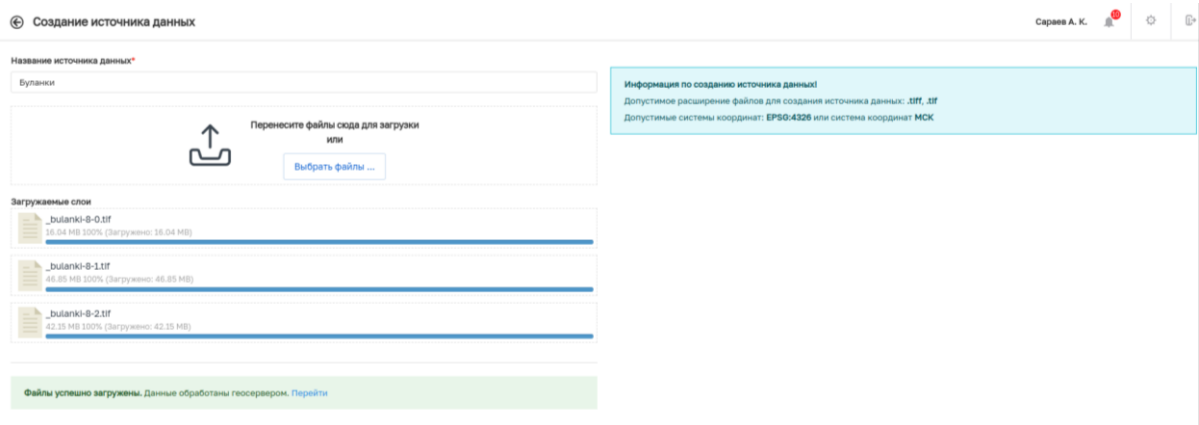


Рисунок 4.2.7 – Интерфейс загрузки пространственных данных в случае успешной загрузки

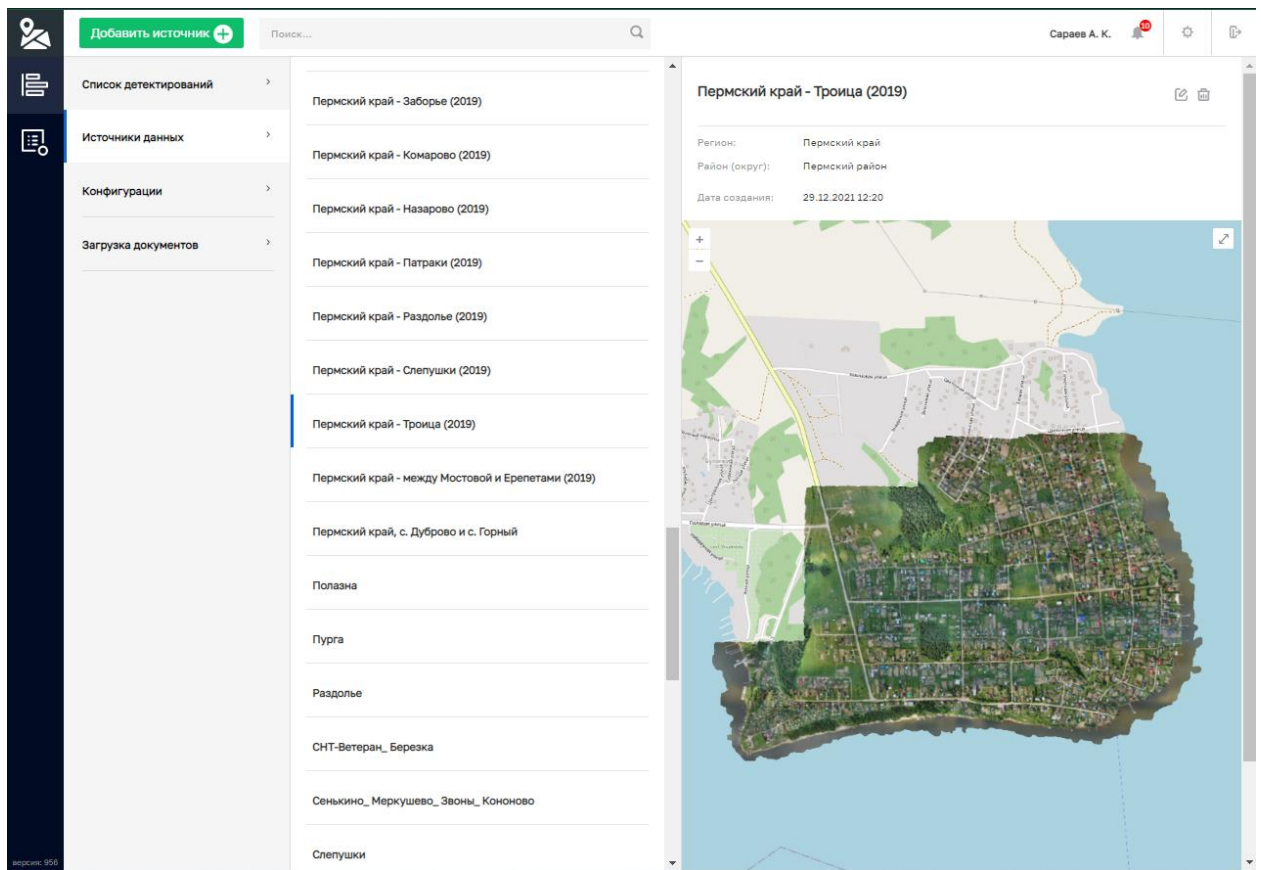


Рисунок 4.2.8 – Интерфейс просмотра загруженных пространственных данных

3) Запустить обработку загруженного источника.

Для этого необходимо в главном меню перейти в раздел «Интеллектуальная обработка», выбрать пункт «Источники данных».

Откроется интерфейс со списком обработок (рисунок 4.2.9), в котором необходимо нажать кнопку «Добавить процесс».

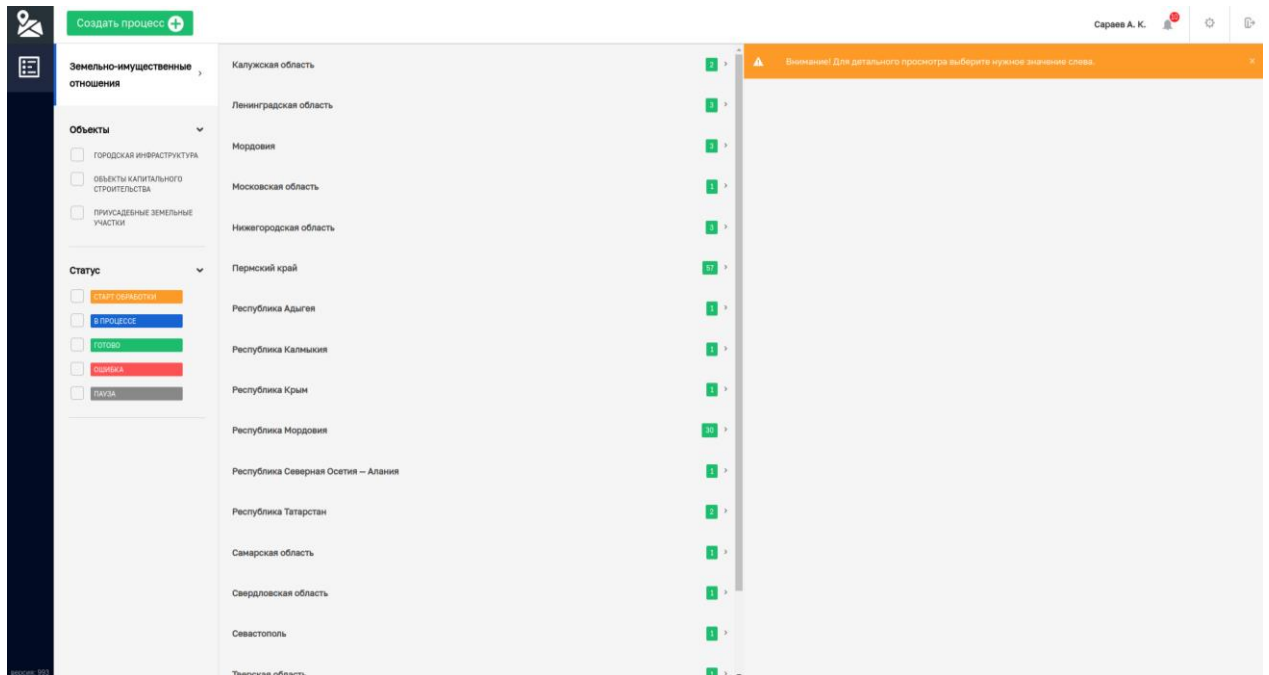


Рисунок 4.2.9 – Интерфейс со списком интеллектуальных обработок

В интерфейсе конфигурирования обработки (рисунок 4.2.10) необходимо указать конфигурацию обработки, источник данных и зону обработки. Далее необходимо нажать кнопку «Запустить процесс». После запуска процесса имеется возможность перейти в интерфейс мониторинга процессов интеллектуальной обработки (рисунок 4.2.11), в котором можно отследить ее статус.

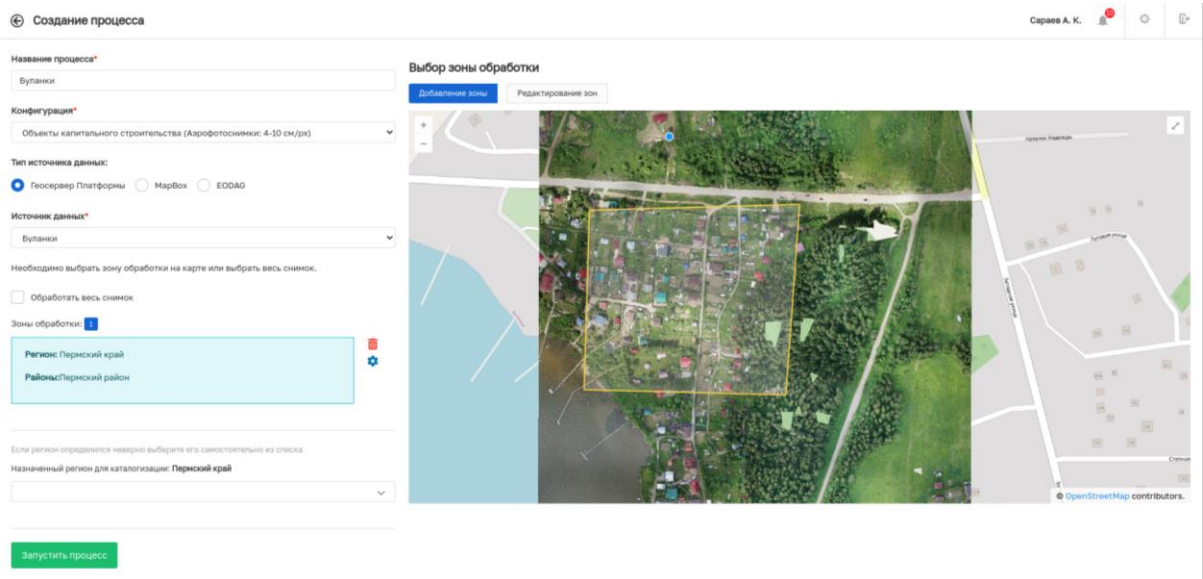


Рисунок 4.2.10 – Интерфейс конфигурирования интеллектуальной обработки

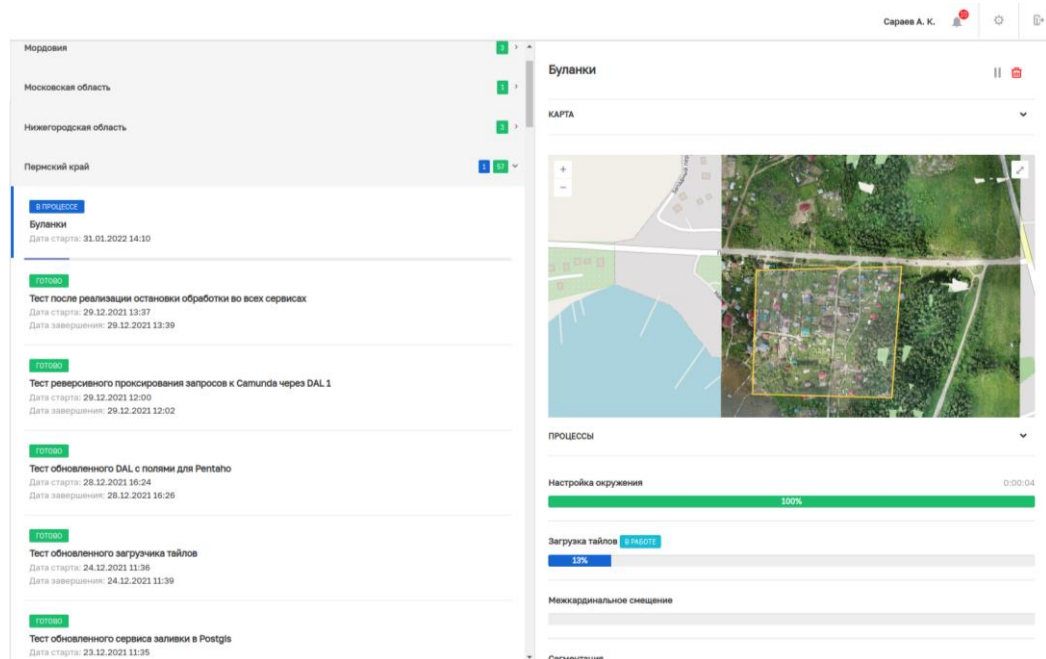


Рисунок 4.2.11 – Интерфейс мониторинга процессов интеллектуальной обработки

По завершению, Платформа предложит перейти к просмотру результатов обработки (рисунок 4.2.12). Если при загрузке возникают ошибки, необходимо произвести проверку следующих сервисов:

- 1) Camunda EE.
- 2) Data Access Layer.



3) Сервисы, в которых произошла ошибка на какой-либо из стадии конвейера.

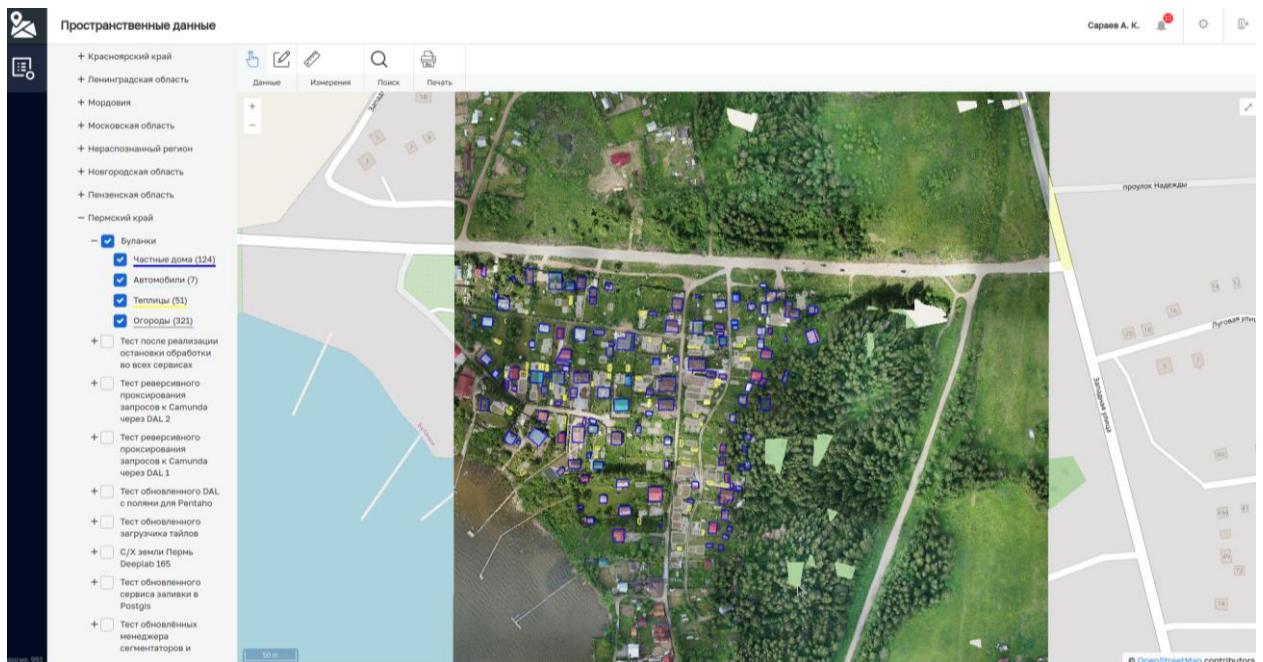


Рисунок 4.2.12 – Интерфейс просмотра результатов интеллектуальной обработки

4) Запустить запрос данных из внешнего источника «Публичная кадастровая карта» (ПКК).

Для этого необходимо в главном меню выбрать пункт «Геопортал», перейти в раздел «Интеллектуальная обработка», выбрать пункт «Список детектированных», выбрать процесс, и нажать кнопку «Получить данные» или «Обновить данные» в блоке «Данные из публичной кадастровой карты» (рисунок 4.2.13). Прогресс получения/обновления данных из ПКК можно отслеживать в соответствующем интерфейсе (рисунок 4.2.14).



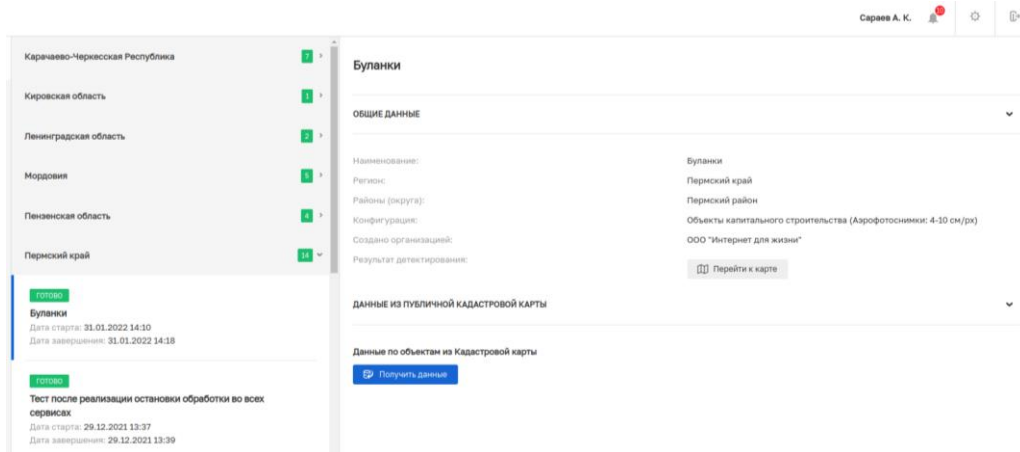


Рисунок 4.2.13 – Интерфейс запуска процесса получения данных из ПКК

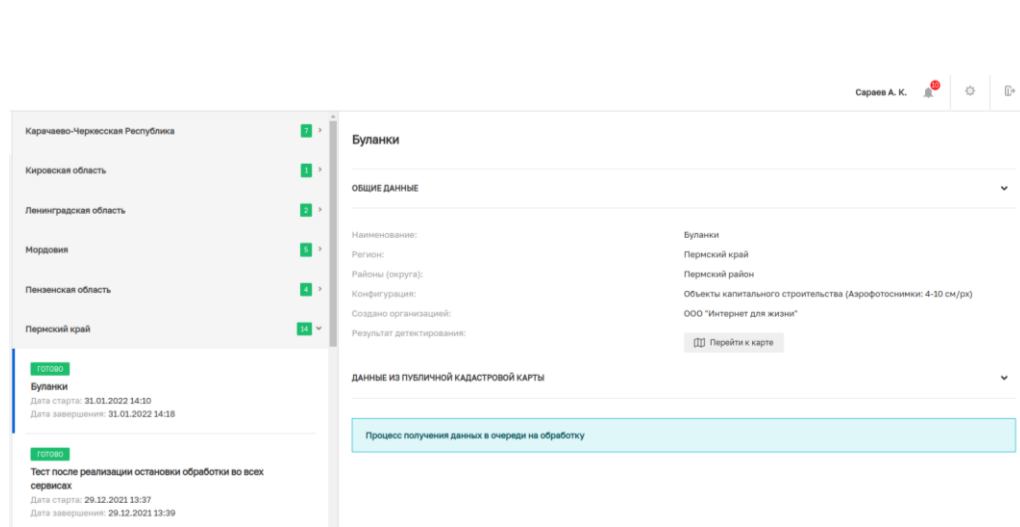


Рисунок 4.2.14 – Интерфейс отслеживания прогресса получения данных из ПКК

По окончании получения данных будут доступны 2 кнопки: обновить данные и удалить данные (рисунок 4.2.15). Если при получении данных возникают ошибки, необходимо произвести проверку следующих сервисов:

- 1) РКК Data Provider.

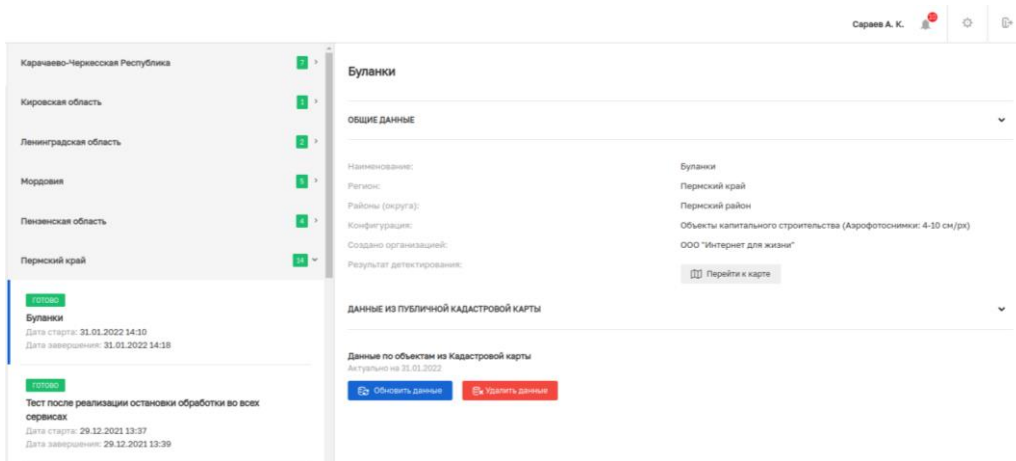


Рисунок 4.2.15 – Интерфейс результата получения данных

5) Запустить анализ данных.

Для этого необходимо в главном меню выбрать пункт «Геопортал», перейти в раздел «Интеллектуальная обработка», выбрать пункт «Список детектирований», выбрать процесс, выбрать набор формируемых отчетов, и нажать кнопку «Анализ данных» в блоке «Аналитический сервис» (рисунки 4.2.16 – 4.2.17).

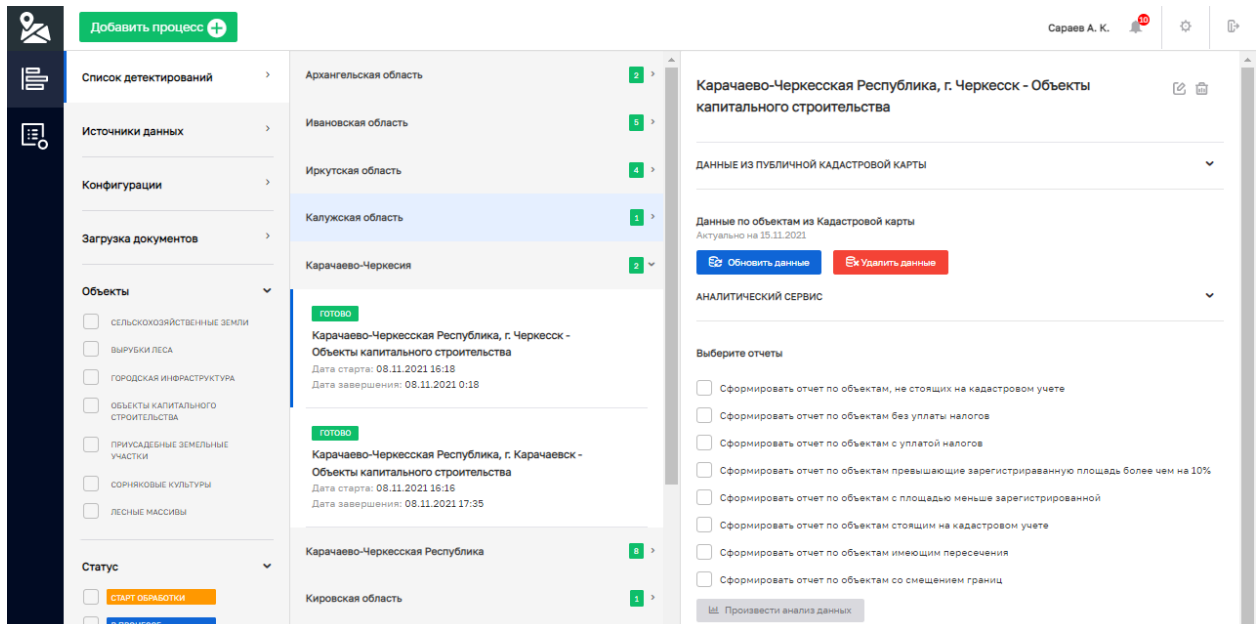


Рисунок 4.2.16 – Интерфейс процесса обработки в подразделе «Формирование реестров»

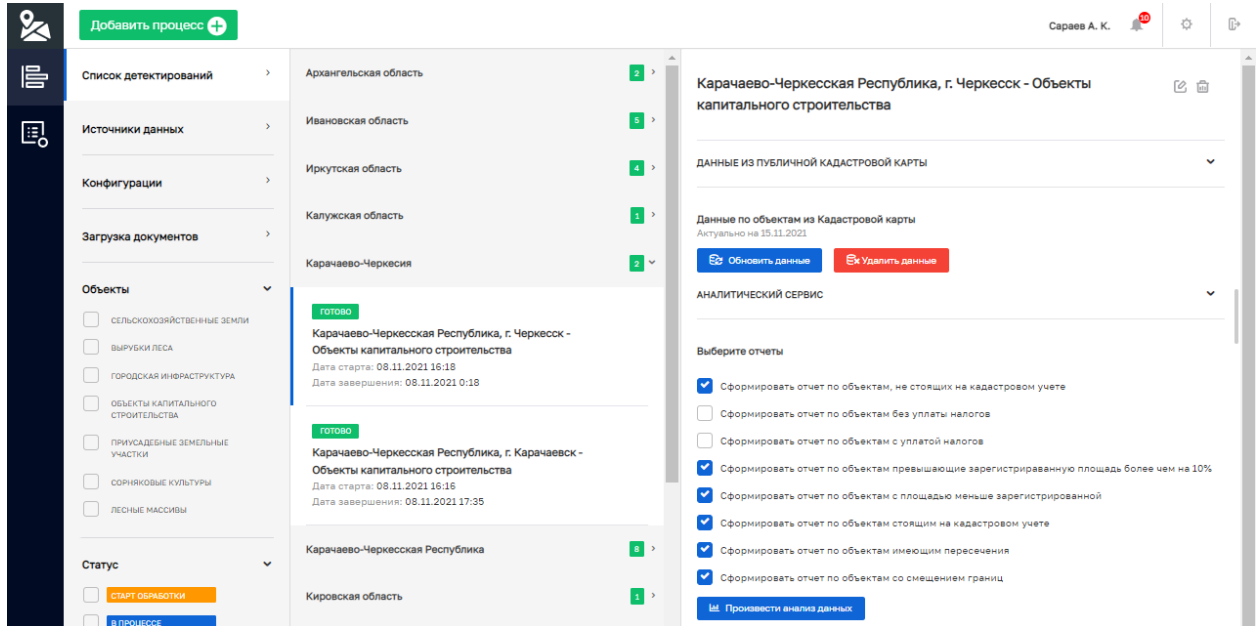


Рисунок 4.2.17 – Интерфейс выбора отчетов

Прогресс анализа данных можно отслеживать в соответствующем интерфейсе (рисунок 4.2.18).

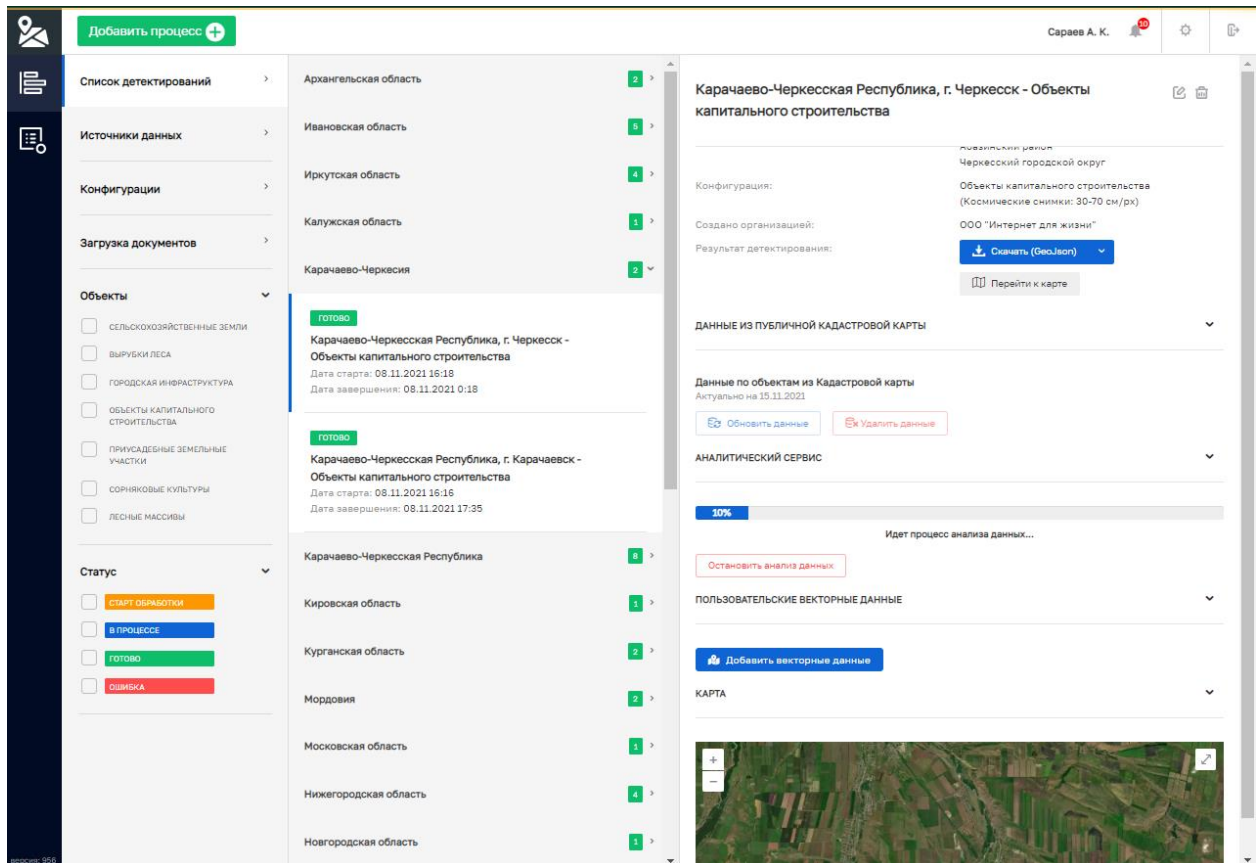


Рисунок 4.2.18 – Интерфейс отслеживания прогресса формирования отчетов

Если при анализе данных возникают ошибки, необходимо произвести проверку следующих сервисов:

- 1) Analytical Service.
- б) Ознакомиться с аналитикой.

Для этого необходимо в главном меню выбрать пункт «Геопортал», перейти в раздел «Интеллектуальная обработка», выбрать пункт «Список детектированных», выбрать процесс. Если анализ данных прошел успешно, то в блоке «Аналитический сервис» станут доступны для скачивания набор аналитических ответов (рисунок 4.2.19), и доступна кнопка «Дашборд аналитики», при нажатии на которую откроется дашборд аналитики (рисунок 4.2.20).

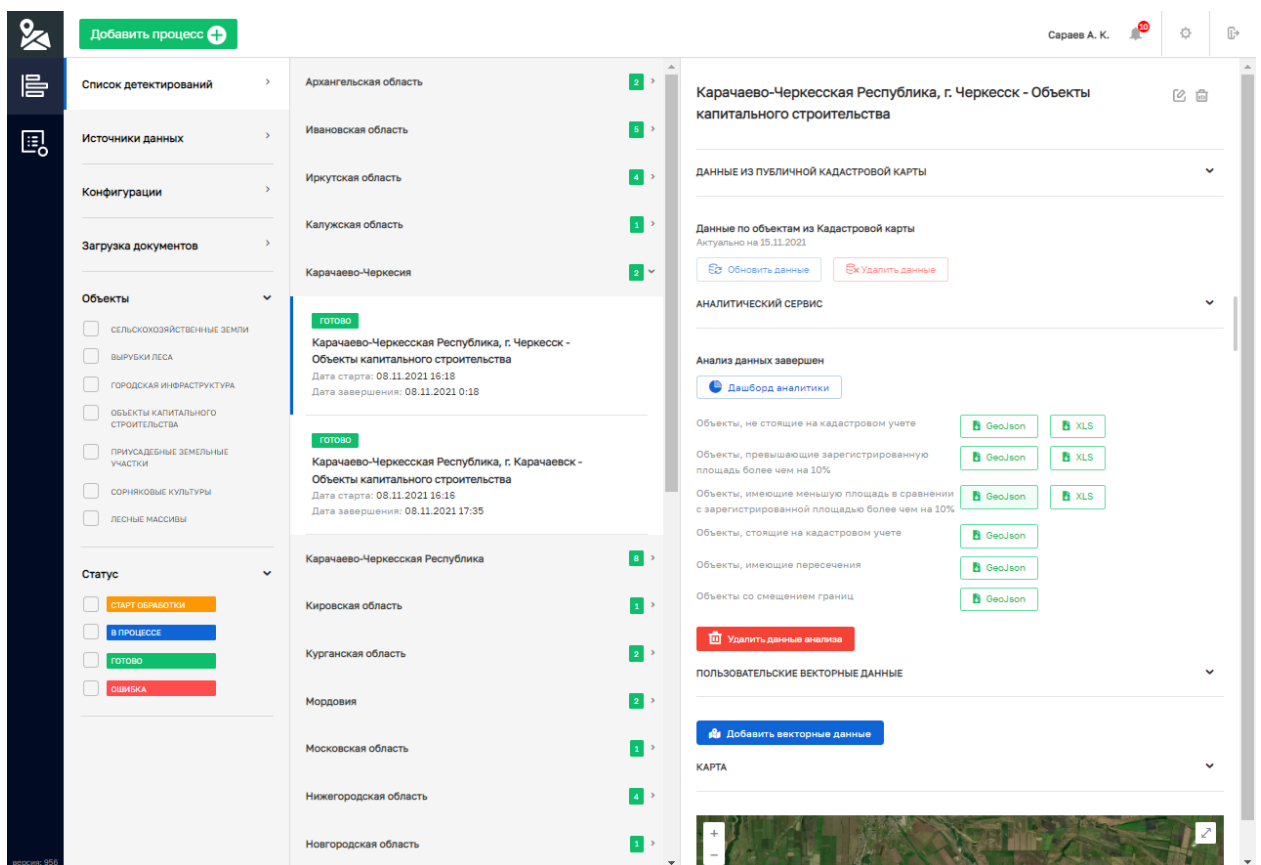


Рисунок 4.2.19 – Интерфейс с набором сформированных аналитических отчетов

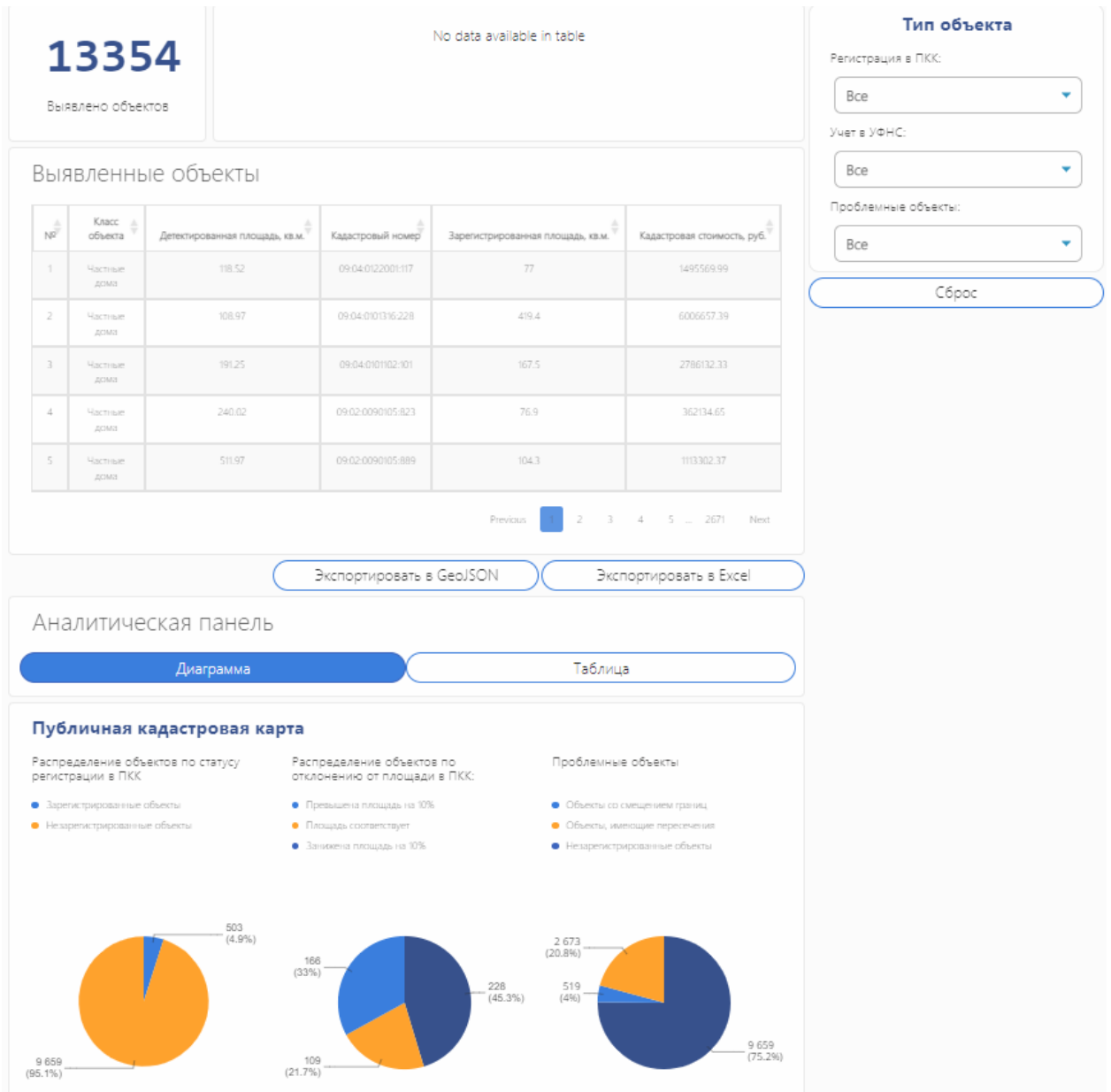


Рисунок 4.2.20 – Интерфейс дашборда аналитики

#### 4.2.1. Проверка работоспособности программы

Для проверки работоспособности программы необходимо проверить работоспособность всех подсистем отдельно:

- 1) Инфраструктурная подсистема.

Для проверки работоспособности инфраструктурной подсистемы необходимо открыть portainer Платформы и убедиться, что все сервисы из стека **air** развернуты и не произошло никаких ошибок при разворачивании.

Если при разворачивании произошли ошибки, необходимо проанализировать логи сервисов, выявить и исправить причину, приведшую к ошибкам при развертывании и повторить развертывание. На рисунок 4.2.1.1 представлен корректно развернутый стэк **aip**.

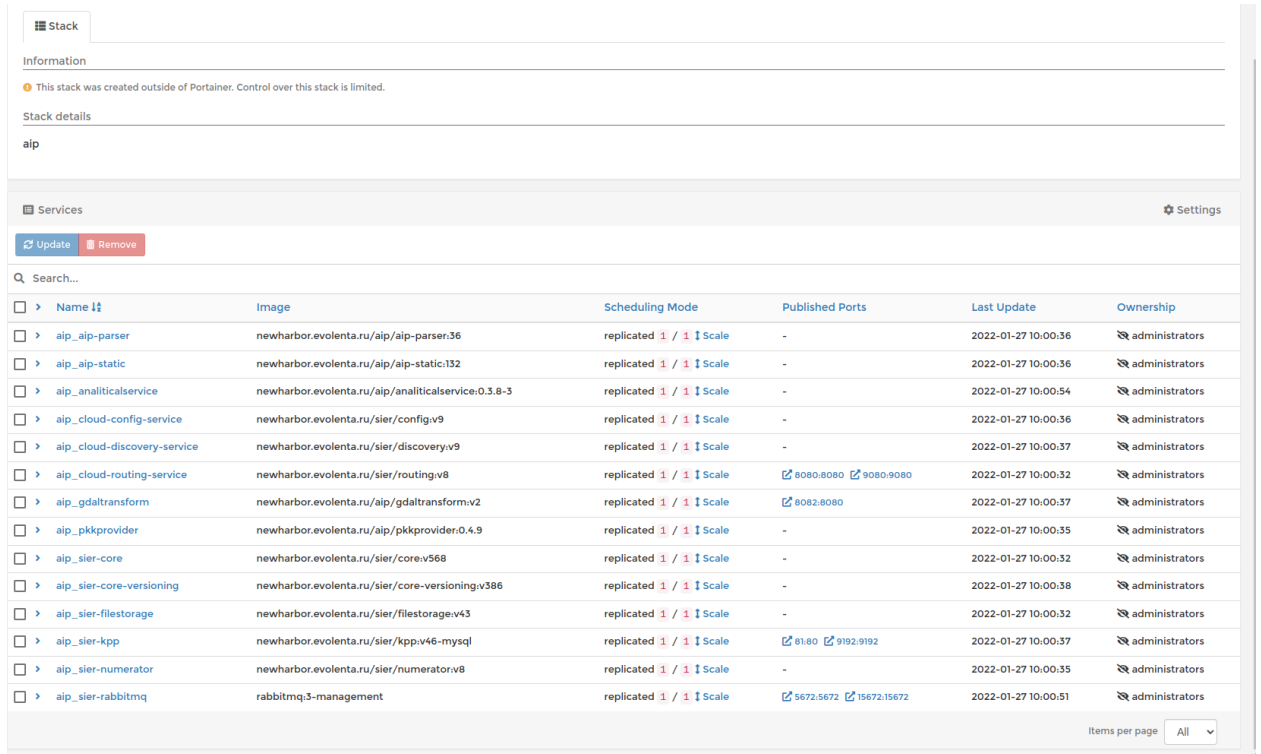


Рисунок 4.2.1.1 – Интерфейс portainer Платформы с отображением стэка **aip**

2) Подсистема управления процессами обработки и вычислительные сервисы.

Для проверки работоспособности подсистемы управления процессами обработки необходимо убедиться, что развернут сервис Samunda и развернуты все необходимые конфигурации обработок.

Для этого необходимо произвести авторизацию в Samunda (рисунок 4.2.1.2) и в разделе processes проверить наличие конфигураций (рисунок 4.2.1.3). Если при разворачивании произошли ошибки, необходимо проанализировать логи сервиса, выявить и исправить причину, приведшую к ошибкам при развертывании, и повторить развертывание.

**CAMUNDA**

Cockpit

Username

Password

**Log in**

Рисунок 4.2.1.2 – Интерфейс авторизации в Camunda

The screenshot shows the Camunda Cockpit interface. At the top, there is a navigation bar with 'Camunda Cockpit' and a menu with 'Processes', 'Decisions', 'Cases', 'Human Tasks', and 'More'. On the right, there is a 'Demo Demo' user indicator. Below the navigation bar, there is a search bar for process instances with the text 'Search process instances' and a sub-label 'Add criteria'. To the right of the search bar is an 'Execute batch' button. Below the search bar, there is a section titled '15 process definitions deployed'. This section contains a table with columns for 'State', 'Incidents', 'Running Instances', 'Name', 'Tenant ID', 'History view', 'Report', and 'Action'. The table lists 15 process definitions, each with a green checkmark in the 'State' column and '0' in the 'Incidents' and 'Running Instances' columns. The 'Name' column contains various process names related to land registry and agricultural monitoring. The 'History view', 'Report', and 'Action' columns contain icons for each process definition.

| State | Incidents | Running Instances | Name   | Tenant ID | History view | Report | Action |
|-------|-----------|-------------------|--|-----------|--------------|--------|--------|
| ✓     | 0         | 0                 | Реестр земель лесного фонда (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px – с фильтрацией облаков)                    |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр земель лесного фонда (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px)  |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр земель сельскохозяйственного назначения (Космические снимки со спутника Sentinel-2: 10 м/px)  |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр земель сельскохозяйственного назначения (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px – с фильтрацией облаков) |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр земель сельскохозяйственного назначения (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px)                         |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр зон распространения сорняковых культур: Борщевик Сосновского (Аэрофотоснимки: 4-10 см/px – AI: Unet)                                |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр зон распространения сорняковых культур: Борщевик Сосновского (Аэрофотоснимки: 4-10 см/px – AI: Deeplab)                             |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр многоквартирных домов (Космические снимки: 30-70 см/px)   |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр объектов капитального строительства (Аэрофотоснимки: 4-10 см/px)  |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр объектов капитального строительства (Космические снимки: 30-70 см/px)   |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр объектов капитального строительства без уплаты налогов (Налоговый потенциал)  |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр приусадебных земельных участков (Аэрофотоснимки: 4-10 см/px)  |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр рубок леса (Космические снимки со спутника Sentinel-2: 10 м/px)   |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр рубок леса (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px – с фильтрацией облаков)                              |           | History view | 📄      | 🗑️     |
| ✓     | 0         | 0                 | Реестр рубок леса (Многоканальные космические снимки со спутника Sentinel-2: 10 м/px)  |           | History view | 📄      | 🗑️     |

Рисунок 4.2.1.3 – Интерфейс просмотра конфигураций обработок в Camunda

3) Подсистема хранения геоданных.

Для проверки работоспособности подсистемы хранения геоданных необходимо открыть portainer Платформы и убедиться, что все сервисы из

стека **geostorages** развернуты и не произошло никаких ошибок при разворачивании.

Если при разворачивании произошли ошибки, необходимо проанализировать логи сервисов, выявить и исправить причину, приведшую к ошибкам при развертывании и повторить развертывание. На рисунок 4.2.1.4 представлен корректно развернутый стек **geostorages**.

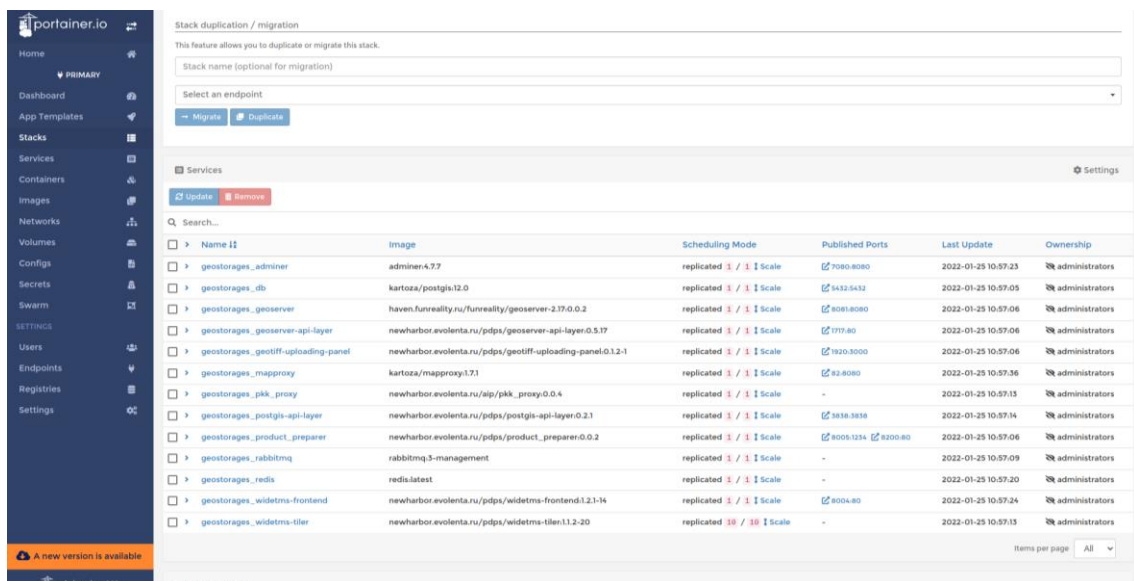


Рисунок 4.2.1.4 – Интерфейс portainer Платформы с отображением стека **geostorages**

Также необходимо убедиться, что геосервер развернулся. Для этого необходимо пройти авторизацию в нем (рисунок 4.2.1.5).

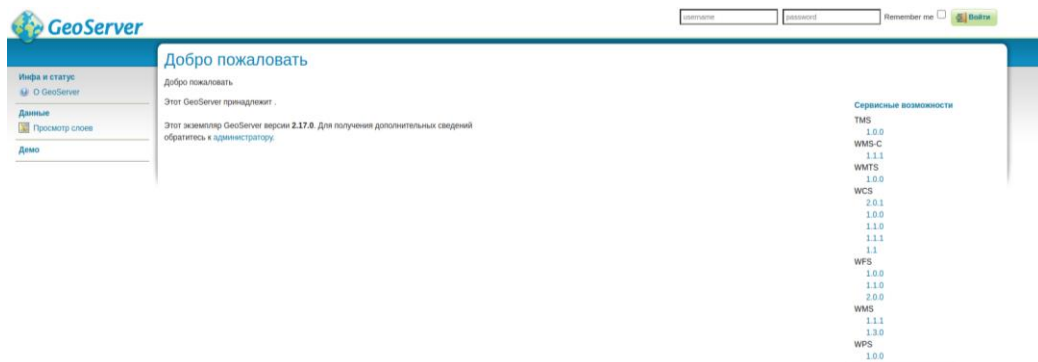


Рисунок 4.2.1.5 – Интерфейс авторизации в geoserver



#### 4) Информационно-аналитическая подсистема.

Для проверки работоспособности информационно-аналитической подсистемы необходимо открыть portainer Платформы и убедиться, что все сервисы из стека **pentaho** развернуты и не произошло никаких ошибок при разворачивании.

Если при разворачивании произошли ошибки, необходимо проанализировать логи сервисов, выявить и исправить причину, приведшую к ошибкам при развертывании и повторить развертывание. На рисунок 4.2.1.6 представлен корректно развернутый стек **pentaho**.

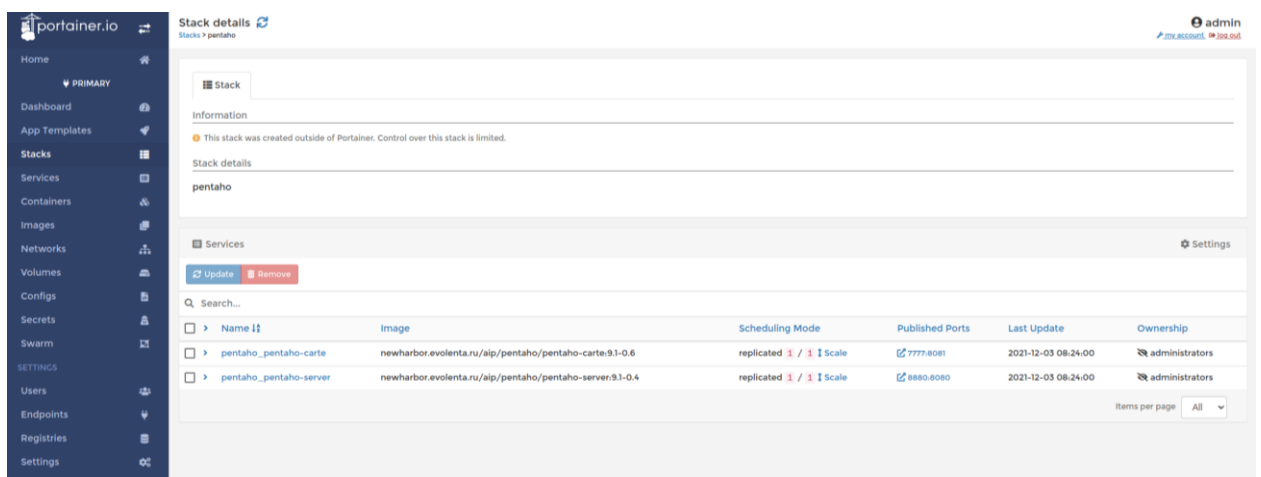


Рисунок 4.2.1.6 – Интерфейс portainer Платформы с отображением стека **pentaho**

Также необходимо убедиться, что сервис **pentaho** развернулся и все необходимые конфигурации также развернулись. Для этого необходимо пройти авторизацию в нем и через браузер файлов проверить наличие требуемых конфигураций (рисунок 4.2.1.7).

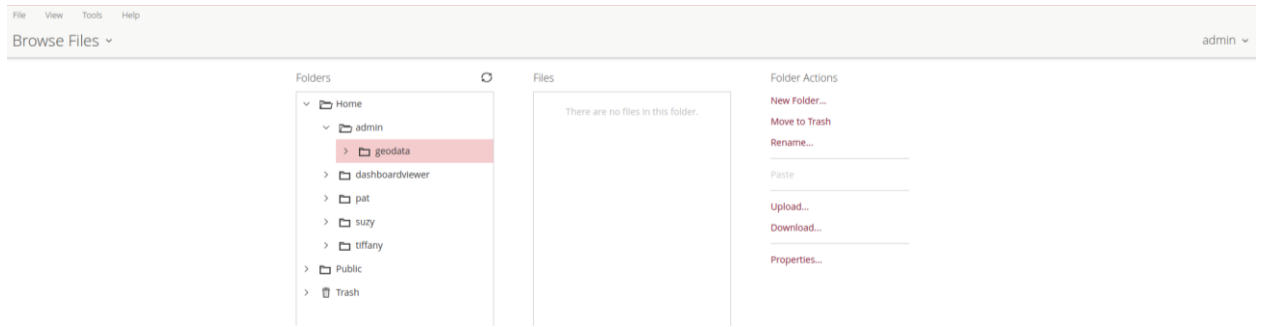


Рисунок 4.2.1.7 – Интерфейс renato Платформы с отображением конфигураций

#### 4.2.2. Проверка на сообщение об ошибке

Логи компонентов Платформы можно просмотреть, используя следующие инструменты:

- 1) Portainer Платформы.
- 2) Командную строку docker.
- 3) Файловую систему сервера.

Рассмотрим случаи на различных примерах.

##### Пример 1 – Ошибка авторизации.

Для этого попытаемся авторизоваться в Платформе с неверным паролем (рисунок 4.2.2.1).

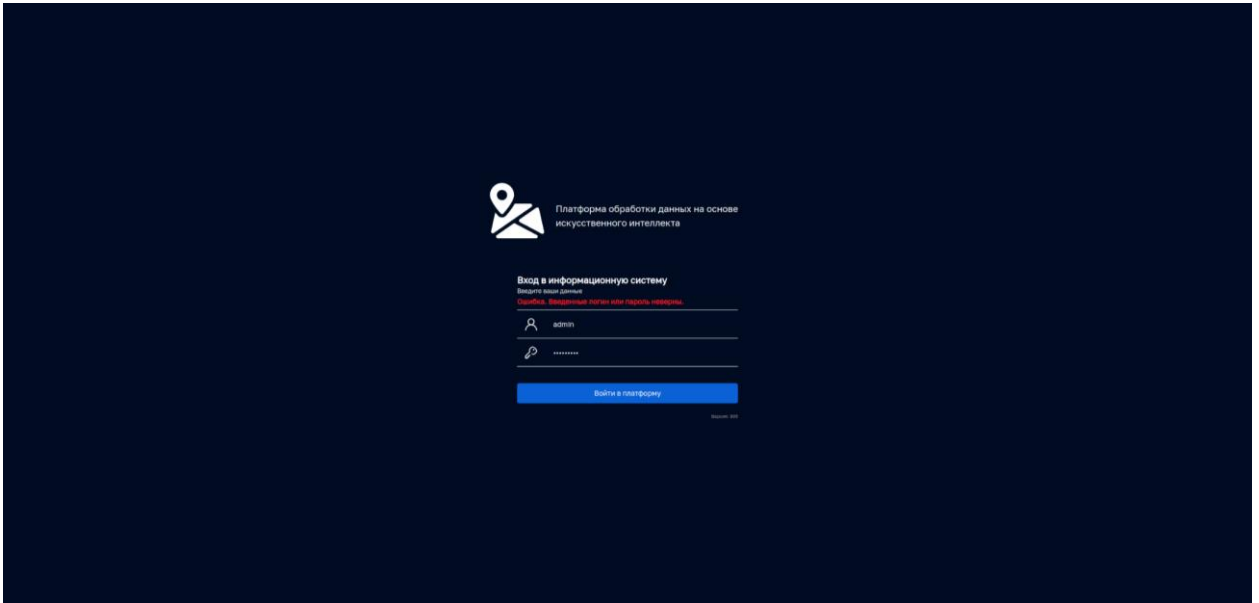


Рисунок 4.2.2.1 – Интерфейс авторизации с сообщением об ошибке авторизации

Проверим логи в portainer. Для этого откроем раздел containers и введем в поисковую строку «kpp» (рисунок 4.2.2.2). Откроем логи найденного сервиса и убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.3).

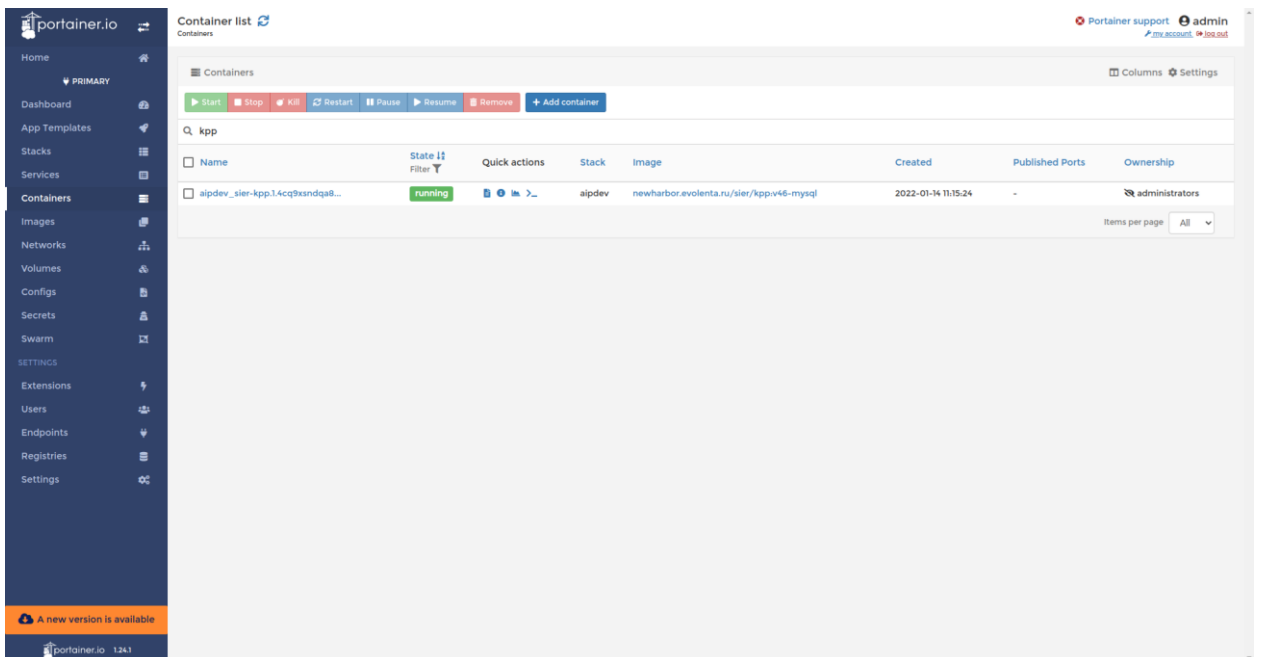


Рисунок 4.2.2.2 – Интерфейс portainer Платформы с отображением результата поиска контейнера КПП

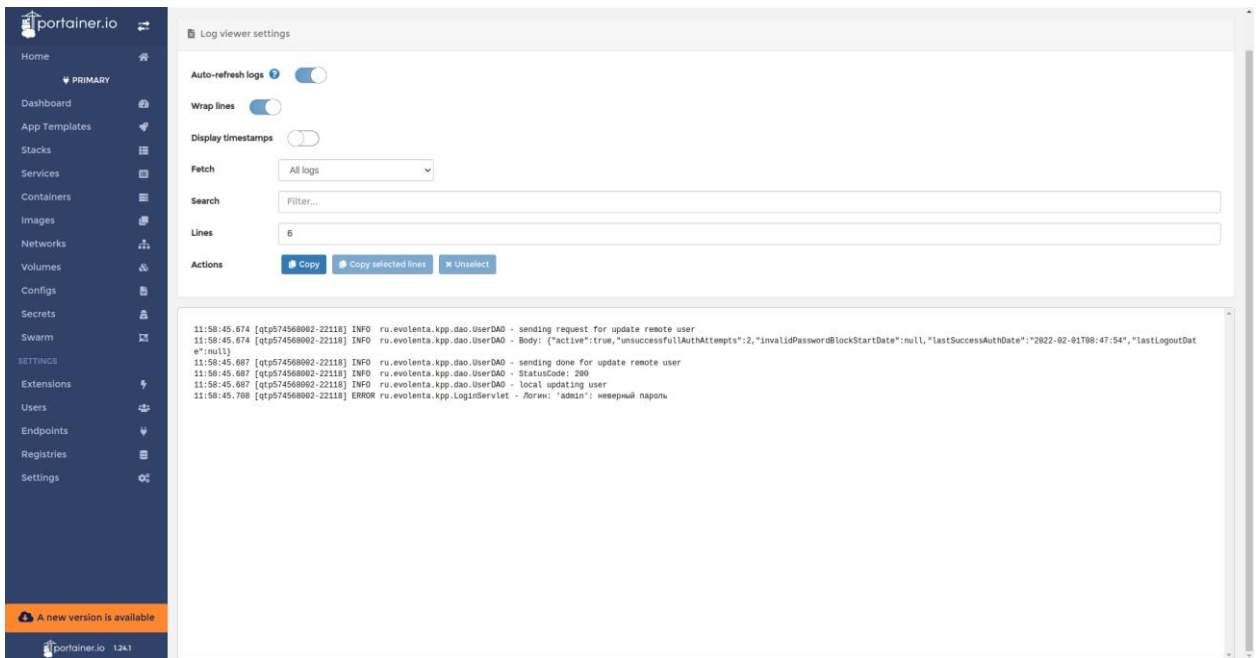


Рисунок 4.2.2.3 – Интерфейс portainer Платформы с отображением логов контейнера КПП

Проверим логи через командную строку docker. С помощью команды `sudo docker ps | grep 'kpp'` получим id контейнера КПП. Далее откроем логи найденного сервиса и убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.4).

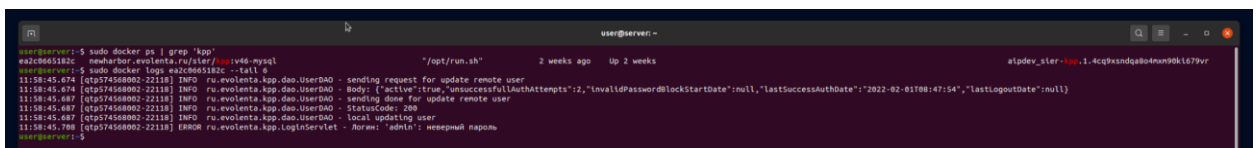


Рисунок 4.2.2.4 – Отображение логов контейнера КПП через командную строку Docker

Проверим логи через файловую систему сервера. С помощью команды `sudo docker ps | grep 'kpp'` получим id контейнера КПП. Далее в директории `/var/lib/docker/containers/{id}` найдем файл логов и выведем последние строки файла через команду `sudo tail -n`. Убедимся, что в логах присутствует



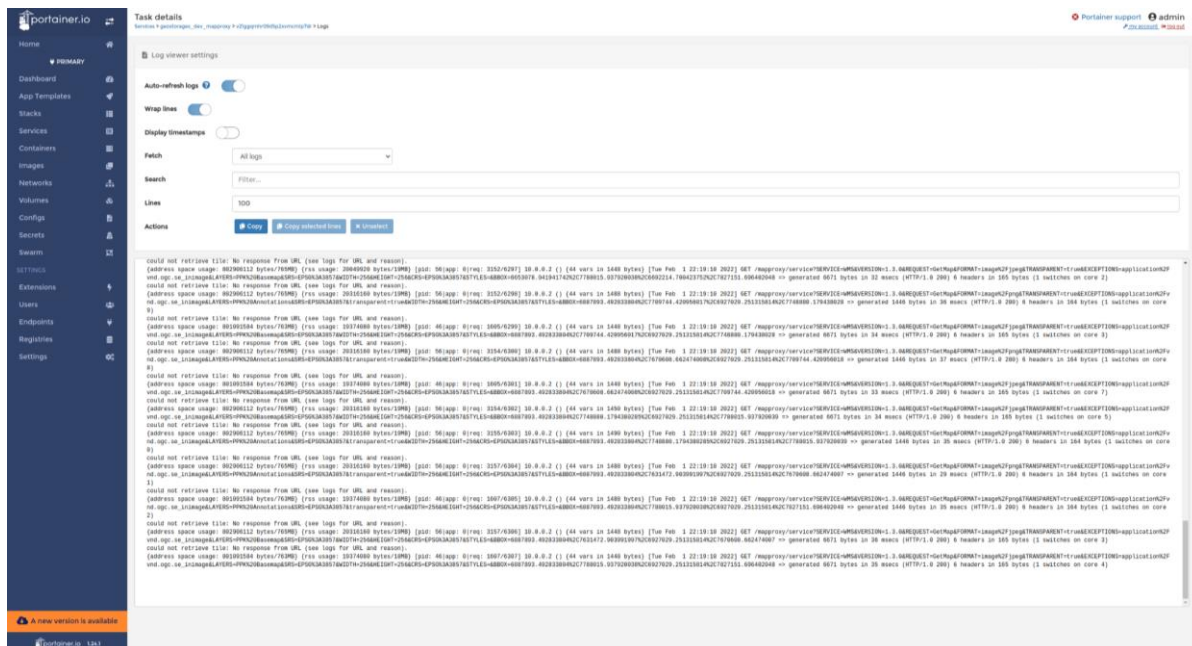


Рисунок 4.2.2.7 – Интерфейс portainer Платформы с выводом логов контейнера Марроху

Проверим логи через командную строку docker. С помощью команды *sudo docker ps | grep 'mapproxy'* получим id контейнера Марроху. Далее откроем логи найденного сервиса и убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.8).



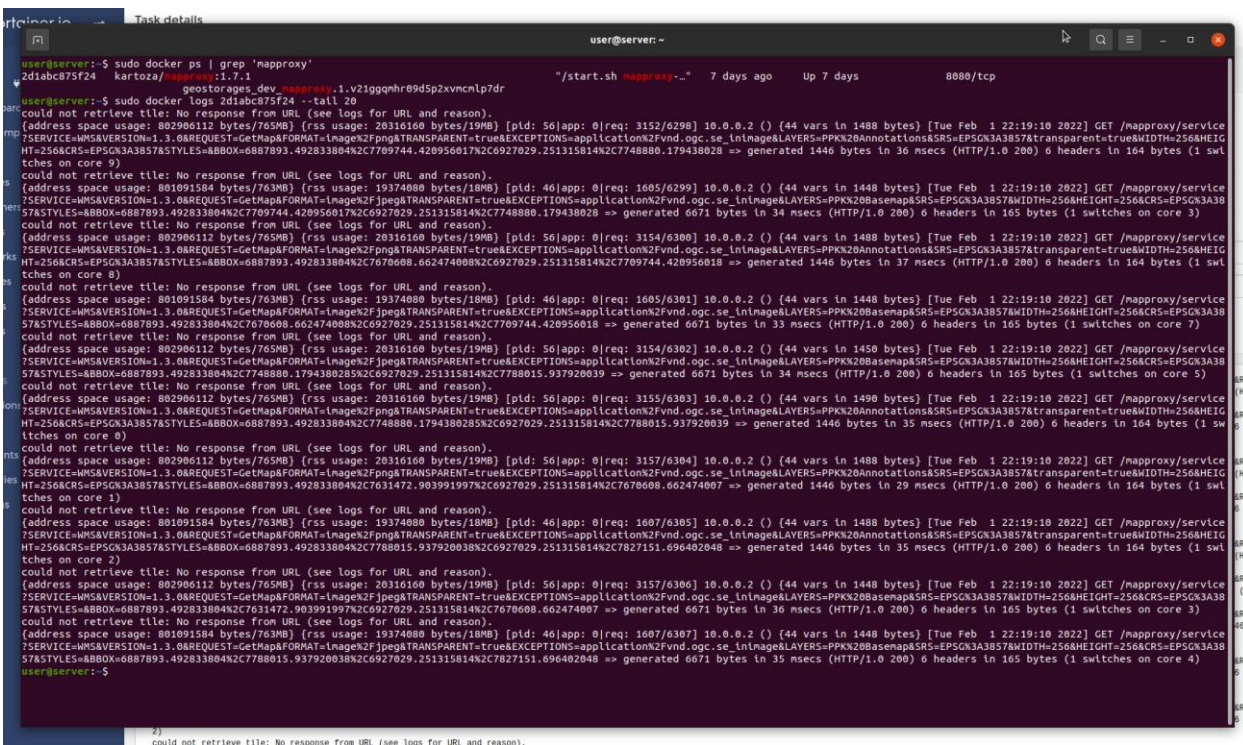
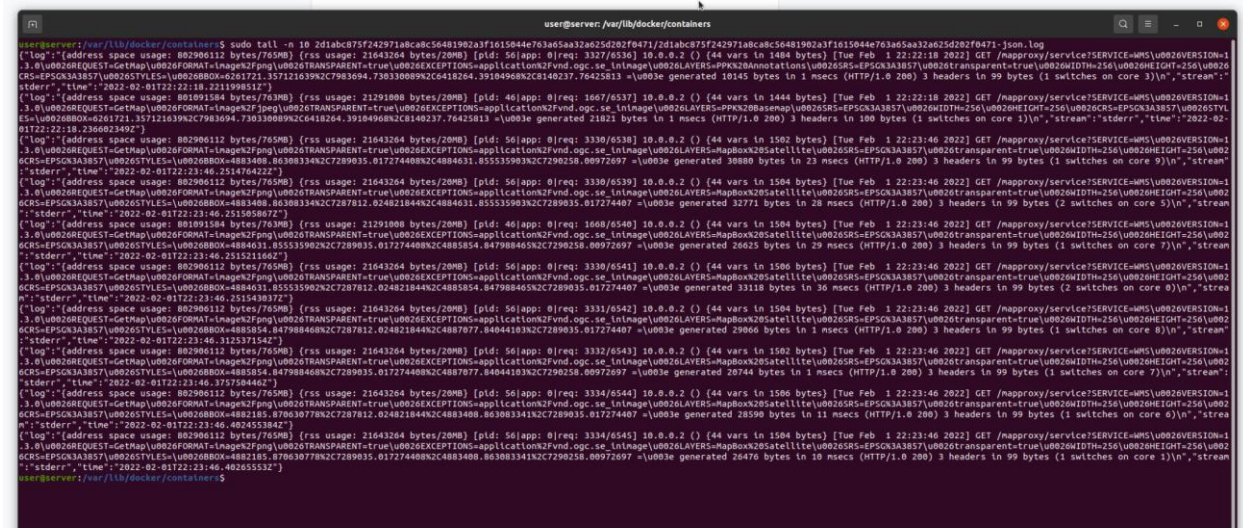


Рисунок 4.2.2.8 – Вывод логов контейнера Марроху через командную строку Docker

Проверим логи через файловую систему сервера. С помощью команды `sudo docker ps | grep 'mapproxy'` получим id контейнера Марроху. Далее в директории `/var/lib/docker/containers/{id}` найдем файл логов и выведем последние строки файла через команду `sudo tail -n`. Убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.9).



## Рисунок 4.2.2.9 – Вывод логов контейнера Marproху из файла логов

Пример 3 – Ошибка загрузки векторных данных.

Для этого попытаемся загрузить в Платформу невалидный слой векторных данных. Платформа выдает ошибку в процессе загрузки (рисунок 4.2.2.10).

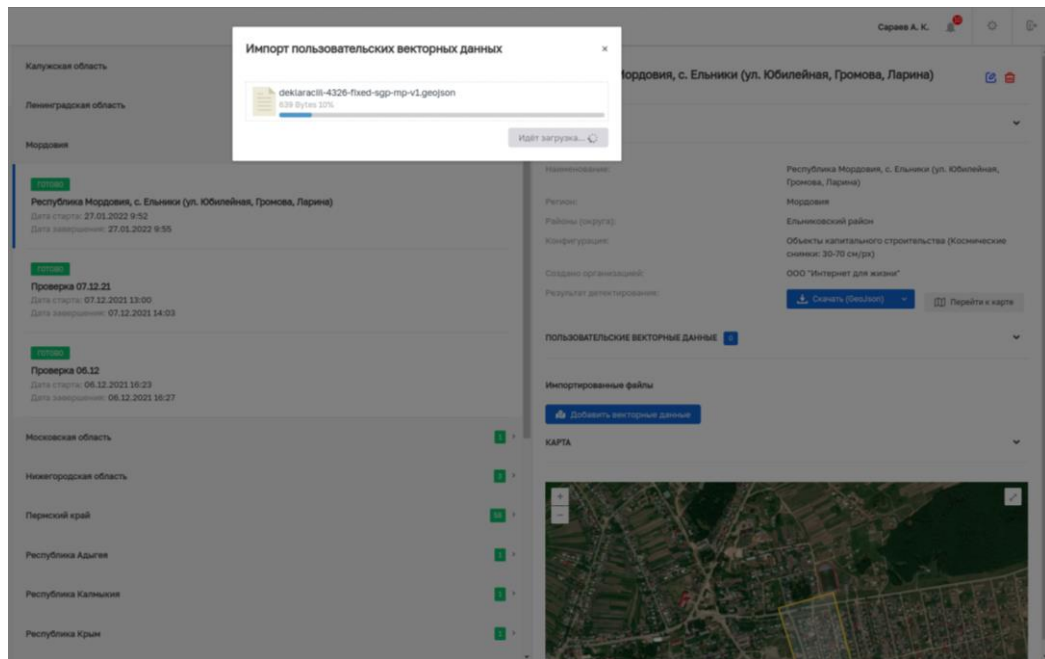


Рисунок 4.2.2.10 – Интерфейс с отображением ошибки при загрузке невалидных векторных данных

Проверим логи в portainer. Для этого откроем раздел containers и введем в поисковую строку «geoserver-api-layer». Откроем логи найденного сервиса и убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.11).



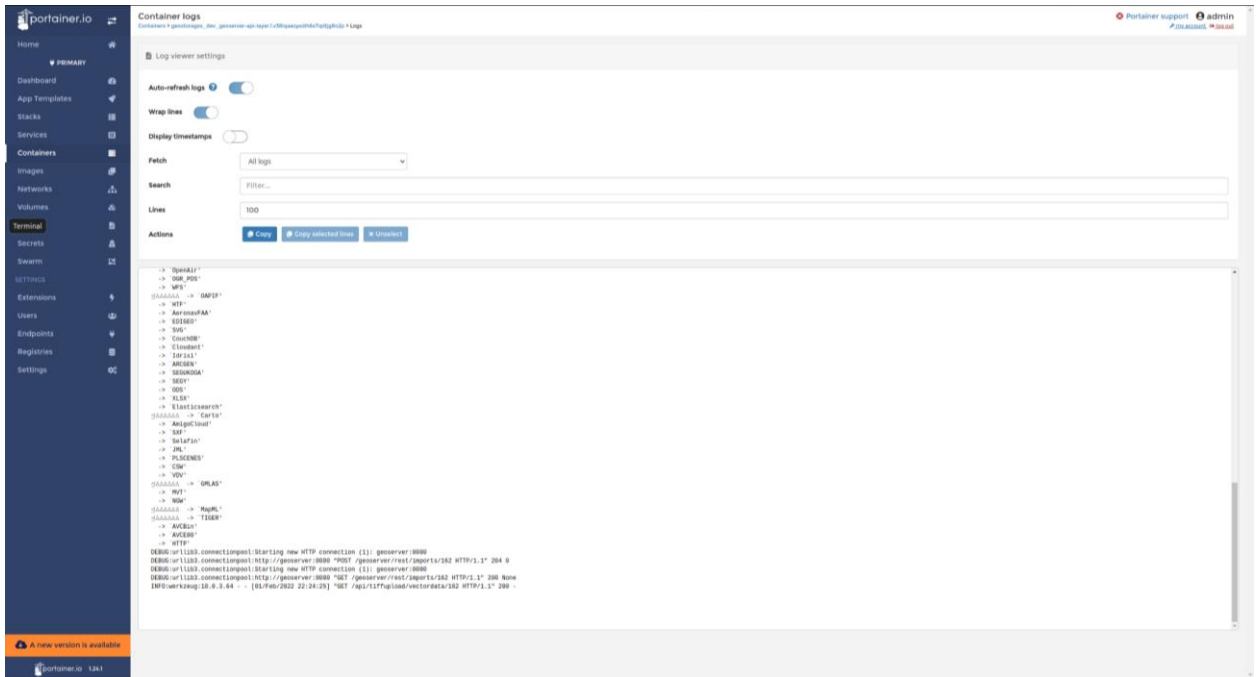


Рисунок 4.2.2.11 – Интерфейс portainer Платформы с выводом логов контейнера Geoserver API Layer

Проверим логи через командную строку docker. С помощью команды `sudo docker ps | grep 'geoserver-api-layer'` получим id контейнера Geoserver API Layer. Далее откроем логи найденного сервиса и убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.12).

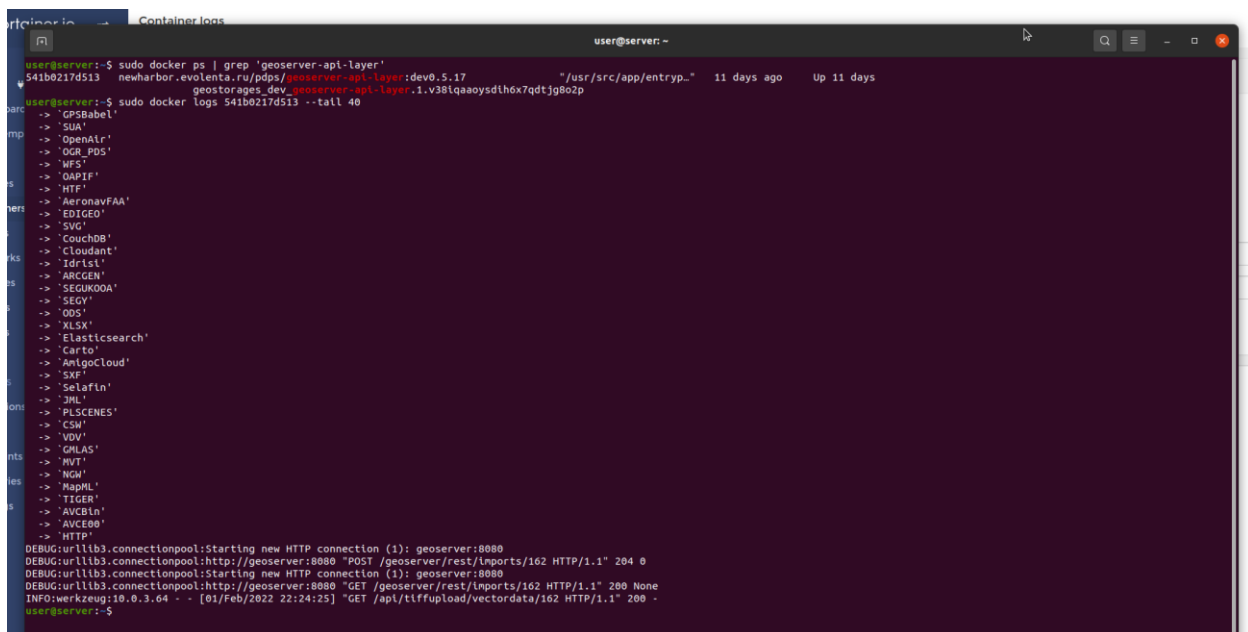


Рисунок 4.2.2.12 – Вывод логов контейнера Geoserver API Layer через

## командную строку Docker

Проверим логи через файловую систему сервера. С помощью команды `sudo docker ps | grep 'geoserver-api-layer'` получим id контейнера Geoserver API Layer. Далее в директории `/var/lib/docker/containers/{id}` найдем файл логов и выведем последние строки файла через команду `sudo tail -n`. Убедимся, что в логах присутствует сообщение об ошибке (рисунок 4.2.2.13).

```

user@server: /var/lib/docker/containers
user@server: /var/lib/docker/containers$ sudo tail -n 40 541b0217d513b883d252e059e9045a675557a50eb1b409aa440ae10b3c59f40/541b0217d513b883d252e059e9045a675557a50eb1b409aa440ae10b3c59f40-json.log
{"log": "\u003e GPSBabel\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174931267Z"}
{"log": "\u003e SUR\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174912162Z"}
{"log": "\u003e OpenAIr\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174917121Z"}
{"log": "\u003e OGR_PDS\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174920037Z"}
{"log": "\u003e WFS\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174922942Z"}
{"log": "\u003e OAPIF\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174925848Z"}
{"log": "\u003e HTF\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174928763Z"}
{"log": "\u003e AeronavFAA\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174931678Z"}
{"log": "\u003e EDIGO\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174934594Z"}
{"log": "\u003e SVC\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174937509Z"}
{"log": "\u003e CouchDB\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174940425Z"}
{"log": "\u003e Cloudant\n", "stream": "stderr", "time": "2022-02-01T22:24:21.17494333Z"}
{"log": "\u003e Idrisi\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174946296Z"}
{"log": "\u003e ARCGEN\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174949191Z"}
{"log": "\u003e SEGUOOA\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174952127Z"}
{"log": "\u003e SEGV\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174955032Z"}
{"log": "\u003e ODS\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174957947Z"}
{"log": "\u003e XLSX\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174960862Z"}
{"log": "\u003e Elasticsearch\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174963777Z"}
{"log": "\u003e Carto\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174966692Z"}
{"log": "\u003e AmigoCloud\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174969607Z"}
{"log": "\u003e SXP\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174972522Z"}
{"log": "\u003e Setaf\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174975437Z"}
{"log": "\u003e PLSCEMES\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174978352Z"}
{"log": "\u003e CSH\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174981267Z"}
{"log": "\u003e VDV\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174984182Z"}
{"log": "\u003e GMLAS\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174987097Z"}
{"log": "\u003e MVT\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174989912Z"}
{"log": "\u003e NGW\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174992827Z"}
{"log": "\u003e MapML\n", "stream": "stderr", "time": "2022-02-01T22:24:21.174995742Z"}
{"log": "\u003e TIGER\n", "stream": "stderr", "time": "2022-02-01T22:24:21.175000157Z"}
{"log": "\u003e AVCE00\n", "stream": "stderr", "time": "2022-02-01T22:24:21.175004572Z"}
{"log": "\u003e HTTP\n", "stream": "stderr", "time": "2022-02-01T22:24:21.175008987Z"}
{"log": "DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): geoserver:8080\n", "stream": "stderr", "time": "2022-02-01T22:24:21.183285823Z"}
{"log": "DEBUG:urllib3.connectionpool:http://geoserver:8080 \u005cPOST /geoserver/rest/Inports/162 HTTP/1.1\u005c 204 0\n", "stream": "stderr", "time": "2022-02-01T22:24:21.186687963Z"}
{"log": "DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): geoserver:8080\n", "stream": "stderr", "time": "2022-02-01T22:24:25.090230611Z"}
{"log": "DEBUG:urllib3.connectionpool:http://geoserver:8080 \u005cGET /geoserver/rest/Inports/162 HTTP/1.1\u005c 200 None\n", "stream": "stderr", "time": "2022-02-01T22:24:25.095982981Z"}
}
{"log": "INFO:werkzeug:10.0.3.64 - - [01/Feb/2022 22:24:25] \"GET /api/tiffupload/vectordata/162 HTTP/1.1\" 200 -\n", "stream": "stderr", "time": "2022-02-01T22:24:25.097004329Z"}
user@server: /var/lib/docker/containers

```

Рисунок 4.2.2.13 – Вывод логов контейнера Geoserver API Layer из файла

ЛОГОВ

## 5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Логи компонентов Платформы (рисунок 5.1) можно просмотреть как через portainer Платформы, так и через командную строку docker.

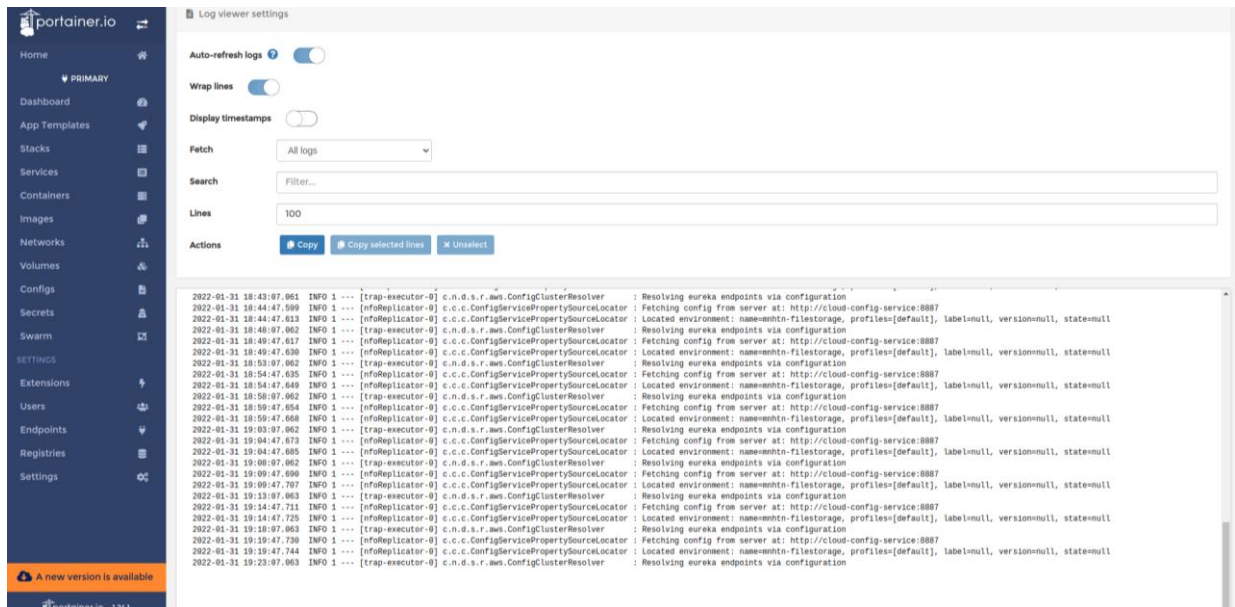


Рисунок 5.1 – Интерфейс portainer Платформы с отображением логов

