

# 1 Схема деплоя демонстрационного стенда

На Рисунке 1 представлена схема деплоя демонстрационного стенда.

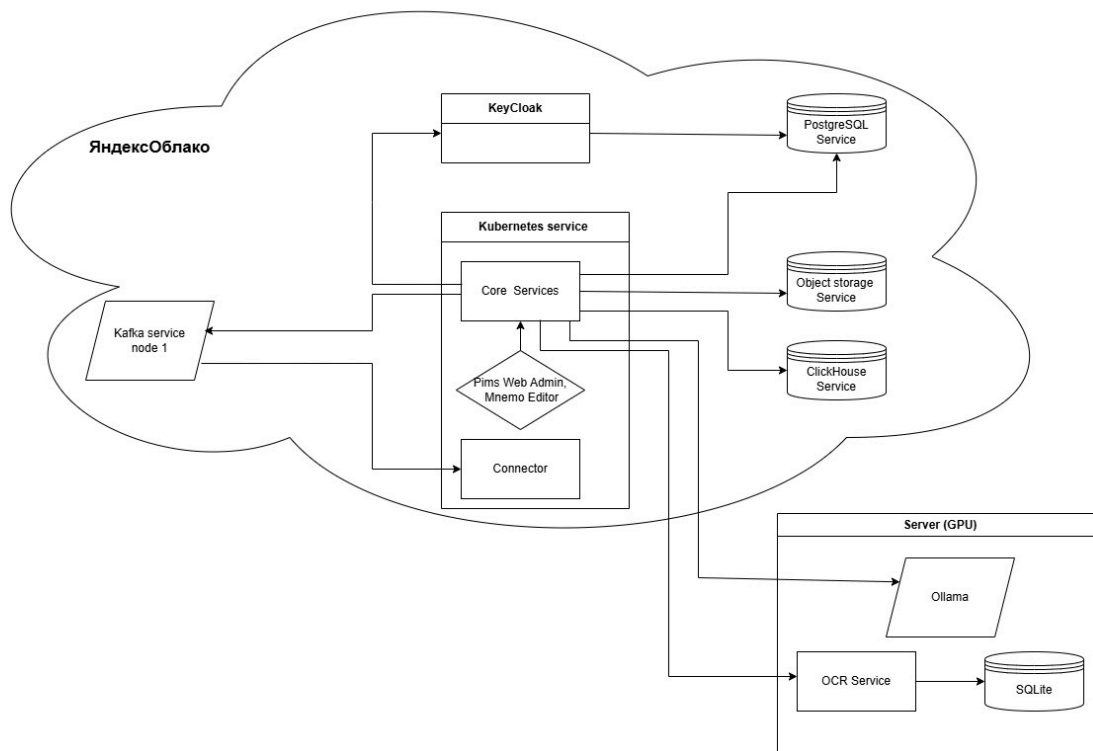


Рис. 1 Схема деплоя демонстрационного стенда

Демонстрационный стенд разворачивается в Яндекс Облаке. Конфигурация стенда:

1. Сервисы системы PIMS, веб-приложения и ПО коннектора разворачиваются в кластере Kubernetes, который является сервисом в Яндекс Облаке;
2. В качестве хранилища данных используется ClickHouse, который является сервисом в Яндекс Облаке;
3. В качестве хранилища конфигураций используется PostgreSQL, которая является сервисом в Яндекс Облаке;
4. Keycloak разворачивается на VM: `keycloakdemos`. База данных для Keycloak содержится в хранилище конфигураций;
5. В качестве брокера сообщений используется Kafka, которая является сервисом в Яндекс Облаке;
6. В качестве хранилища графических объектов используется бакет в Object storage, которое является сервисом в Яндекс Облаке.
7. Ollama, OCR сервис и БД его конфигураций SQLite развернуты на отдельной VM с GPU, расположенной вне Яндекс Облака.

## 2 Предварительные условия для деплоя сервисов PIMS

Для корректной работы платформы PIMS и ее модулей необходимо, чтобы были развернуты и настроены следующие компоненты:

### 1. Кластер K8s/Deckhouse

Официальная документация размещена по ссылке: <https://deckhouse.ru/products/kubernetes-platform/gs/>

### 2. БД Postgres;

Официальная документация размещена по ссылке: <https://www.postgresql.org/docs/current/tutorial-install.html>

### 3. БД Clickhouse

Официальная документация размещена по ссылке: <https://clickhouse.com/docs/getting-started/quick-start>

### 4. Сервис Kafka

Официальная документация размещена по ссылке: <https://kafka.apache.org/quickstart>

Необходимо создать приведенный ниже список топиков.

#### Список топиков:

```
agent_image_topic      -- топик, через который передаются докер-образы
agent_logs_topic       -- топик, в который агенты пишут лог
agent_registry_topic   -- топик для регистрации агентов и их диагностики
agent_request_topic    -- топик запросов от платформы к агентам
agent_response_topic   -- топик ответов от агентов к платформе
archive_value_topic    -- топик, в который коннекторы отправляют исторические данные
ack_alarms_topic       -- топик, в который поступает информация о квитированных тревогах
dead_letter_queue      -- топик, в который записываются неотправленные уведомления
downlink_control_topic -- топик, из которого коннектор забирает обновленную конфигурацию
healthchecks-topic     -- топик, для проверки сервисами связи с Kafka
letter_queue           -- топик, в который сервис уведомлений записывает информацию для отправки
logs_topic             -- топик, в который коннекторы пишут лог
uplink_control_topic   -- топик, в который коннектор отправляет обновления своего состояния
value_topic            -- топик, в который коннекторы отправляют данные
```

### 5. Сервис Keycloak

Официальная документация размещена по ссылке: <https://www.keycloak.org/getting-started/getting-started-docker>

Требуется применить конфигурацию реалмов: `ofg_realm_files.zip`, и в файле `users-0` поменять значения на действительные в разделах `email`, `createdDate`, `secretData`, `credentialData`, `id`.

### 6. БД Minio

Официальная документация размещена по ссылке: <https://min.io/docs/minio/linux/operations/installation.html>

### 7. Ollama

Официальная документация размещена по ссылке: <https://docs.ollama.com/>

### 8. БД SQLite

Официальная документация размещена по ссылке: <https://sqlite.org/docs.html>

Для развертки и настройки требуемых сервисов обратитесь к документации официальных разработчиков.

## 3 Деплой сервисов платформы

### Сервисы системы PIMS

Архитектура системы PIMS является сервис ориентированной. Каждый сервис выполняет определенный функционал. Часть сервисов используют БД Postgres для хранения конфигураций.

Файлы сервисов хранятся в репозиториях Gitlab, наименование который формируется по правилу: `git.of-group.ru/pims/[имя сервиса]`.

Например, [git.of-group.ru/pims/alarms-service](https://git.of-group.ru/pims/alarms-service) - репозиторий сервиса тревог, как представлено на Рисунке 2.

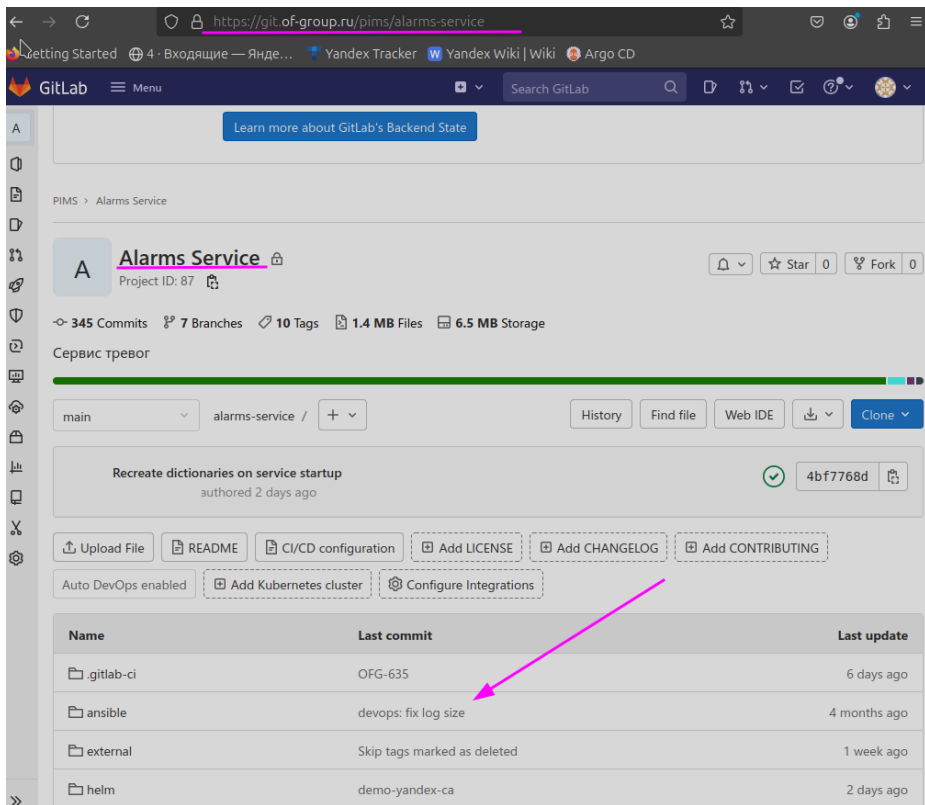


Рис. 2 Пример репозитория сервиса (тревоги) в Gitlab

Конфигурации сервисов хранятся в JSON-формате внутри сервиса `Vault`, развернутого внутри кластера K8s/Deckhouse (см. Рисунок 3). Пример одной из конфигураций представлен на Рисунке 4.

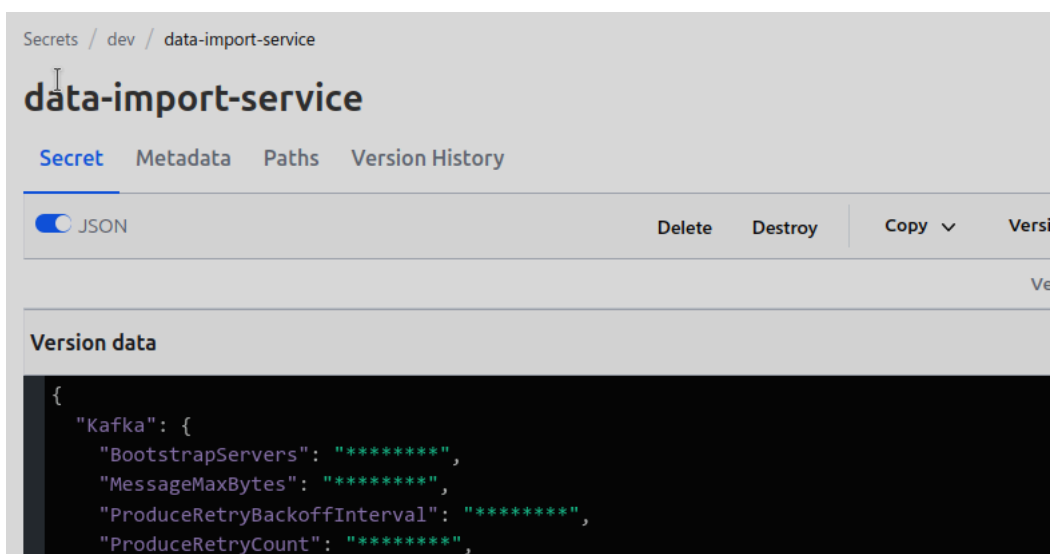


Рис. 3 Хранилище конфигураций сервисов Vault на примере сервиса импорта данных

```

{
  "Kafka": {
    "BootstrapServers": "*****",
    "MessageMaxBytes": "*****",
    "ProduceRetryBackoffInterval": "*****",
    "ProduceRetryCount": "*****",
    "ValueTopic": "*****"
  },
  "Logging": {
    "Console": {
      "FormatterName": "*****",
      "FormatterOptions": {
        "IncludeScopes": "*****",
        "JsonWriterOptions": {
          "Indented": "*****"
        }
      },
      "SingleLine": "*****",
      "TimestampFormat": "*****",
      "UseUtcTimestamp": "*****"
    },
    "LogLevel": {
      "Default": "*****",
      "Microsoft": "*****",
      "Microsoft.Hosting.Lifetime": "*****"
    }
  },
  "LogLevel": {
    "Default": "*****",
    "Microsoft.AspNetCore": "*****"
  }
}

```

Рис. 4 Вид конфигурации на примере сервиса импорта данных

В Таблице 1 приведен общий перечень сервисов системы PIMS, развернутых на демо-стенде, с описанием их назначения, ссылкой на репозиторий и наименованием используемой БД.

Таблица 1. Общий список сервисов

↑↓ Сортировка    🔍 Поиск    ⚙️

	☰ Название ...	☰ Назначение сер...	☰ Ссылка на репозиторий	☰ Назван...	+
1	<code>alarms-service</code> (Сервис тревог)	Сервис отвечает за формирование тревог на основе проверки соответствия значения тега условиям, настроенным пользователем	<a href="https://git.of-group.ru/pims/alarms-service">https://git.of-group.ru/pims/alarms-service</a>	alarms_service_db	
2	<code>api-gateway</code> (Сервис API)	Сервис отвечает за взаимодействие сервисов платформы и сторонних сервисов. Точка входа для сторонних сервисов	<a href="https://git.of-group.ru/pims/api-gateway">https://git.of-group.ru/pims/api-gateway</a>	-	
3	<code>data-import-service</code> (Сервис импорта данных)	Сервис отвечает за внесение в платформу данных, не получаемых с источников	<a href="https://git.of-group.ru/pims/data-import-service">https://git.of-group.ru/pims/data-import-service</a>	-	
4	<code>data-sink</code> (Сервис сбора данных)	Сервис отвечает за вычитывание данных из брокера сообщений и сохранение этих данных в хранилище данных	<a href="https://git.of-group.ru/pims/data-sink">https://git.of-group.ru/pims/data-sink</a>	-	
5	<code>datasource-management</code> (Сервис управления источниками данных)	Сервис отвечает за мониторинг состояния существующих коннекторов и конфигурирование параметров сбора данных и списка тегов	<a href="https://git.of-group.ru/pims/datasource-management-service">https://git.of-group.ru/pims/datasource-management-service</a>	datasource_management_db	
6	<code>handlers-service</code>	Сервис отвечает за	<a href="https://git.of-">https://git.of-</a>	handlers_service	

7	(Сервис обработчиков)  <code>history-service</code>  (Сервис истории)	преобразование значения тега, получаемого от источника, в соответствии с заданной логикой  Сервис отвечает за обращение в хранилище данных для получения истории и актуальных значений тегов	<a href="https://git.ofgroup.ru/pims/handlers-service">group.ru/pims/handlers-service</a>  <a href="https://git.ofgroup.ru/pims/history-service">https://git.ofgroup.ru/pims/history-service</a>	_db  -
8	<code>messaging-service</code>  (Сервис отправки сообщений)	Сервис отвечает за формирование и отправку сообщения на email	<a href="https://git.ofgroup.ru/pims/messaging-service">https://git.ofgroup.ru/pims/messaging-service</a>	messaging_service_db
9	<code>mnemo-backend</code>  (Сервис бэкенда мнемосхем)	Сервис отвечает за серверную часть редактора мнемосхем и хранение библиотеки графических элементов, мнемосхем и их иерархии	<a href="https://git.ofgroup.ru/pims/mnemo-backend">https://git.ofgroup.ru/pims/mnemo-backend</a>	mnemo_backend_db
10	<code>mnemoeditor</code>  (Редактор мнемосхем)	Веб-приложение "Автоматизированное рабочее место"	<a href="https://git.ofgroup.ru/pims/mnemoeditor">https://git.ofgroup.ru/pims/mnemoeditor</a>	-
11	<code>object-model-service</code>  (Сервис объектной модели)	Сервис, который представляет собой иерархическое хранилище объектов, их атрибутов и сопоставление имен переменных объекта идентификаторам тэгов. Позволяет связать теги с источника с цифровой моделью предприятия	<a href="https://git.ofgroup.ru/pims/object-model-service">https://git.ofgroup.ru/pims/object-model-service</a>	object_model_db
12	<code>pims-web-admin</code>	Веб-приложение "Графический интерфейс системного администратора"	<a href="https://git.ofgroup.ru/pims/pims-web-admin">https://git.ofgroup.ru/pims/pims-web-admin</a>	-
13	<code>reporting-service</code>  (Сервис отчетности)	Сервис отвечает за формирование файлов, содержащих данные из хранилища данных, преобразованные и отформатированные в соответствии с пользовательской логикой	<a href="https://git.ofgroup.ru/pims/reporting-service">https://git.ofgroup.ru/pims/reporting-service</a>	reporting_service_db
14	<code>predictive-analytics-service</code>  (Сервис предиктивной аналитики)	Сервис отвечает за прогнозирование вероятности возникновения отказов оборудования и времени отказа на основе исторических данных с оборудования	<a href="https://git.ofgroup.ru/pims/predictive-analytics-service">https://git.ofgroup.ru/pims/predictive-analytics-service</a>	predictive_analytics_db
15	<code>triggers-service</code>  (Сервис уведомлений)	Сервис отвечает за создание уведомления для рассылки	<a href="https://git.ofgroup.ru/pims/triggers-service">https://git.ofgroup.ru/pims/triggers-service</a>	triggers_service_db

16	<code>window-functions-service</code> (Сервис оконных функций)	получателям, в случае срабатывания условия - триггера Сервис отвечает за расчет показателей на основе истории тега за интервал - окно и пользовательского скрипта	<a href="https://git.of-group.ru/pims/window-functions-service">https://git.of-group.ru/pims/window-functions-service</a>	window_functions_db	
17	<code>sap-integration-service</code> (Сервис интеграции с SAP)	Сервис отвечает за получение информации по ТОиР из SAP и отправку значений измерений в SAP	<a href="https://git.of-group.ru/pims/sap-integration-service">https://git.of-group.ru/pims/sap-integration-service</a>	sap_integration_service_db	
18	<code>operator-journal-service</code> (Сервис журнала диспетчера)	Сервис отвечает за возможность внесения в платформу текстовых заметок пользователями системы	<a href="https://git.of-group.ru/pims/operator-journal-service">https://git.of-group.ru/pims/operator-journal-service</a>	operator_journal_service_db	
19	<code>ai-service</code> (Сервис искусственного интеллекта)	Сервис отвечает за построение базы знаний, поиск аномалий и корреляций в данных, формирование пояснений к корреляциям, функционал ИИ-помощника	<a href="https://git.of-group.ru/pims/ai-service">https://git.of-group.ru/pims/ai-service</a>	ai_service_db	
20	<code>ocr-service</code> (Сервис OCR)	Сервис отвечает за возможность распознавания текста из загруженных PDF-файлов	<a href="https://git.of-group.ru/ai/ocr">https://git.of-group.ru/ai/ocr</a>	ocr_service_db	
+					

## Установка сервиса PIMS

1. Необходимо удостовериться, что создана база данных сервиса в PostgreSQL;
2. Перед деплоем сервиса необходимо создать таблицы для сервисов при помощи скриптов SQL. Скрипт необходимо запустить в БД, соответствующей сервису. Скрипты уникальны для каждого сервиса и хранятся в папке `sql`, файл `latest.sql` (см. Рисунок 5).

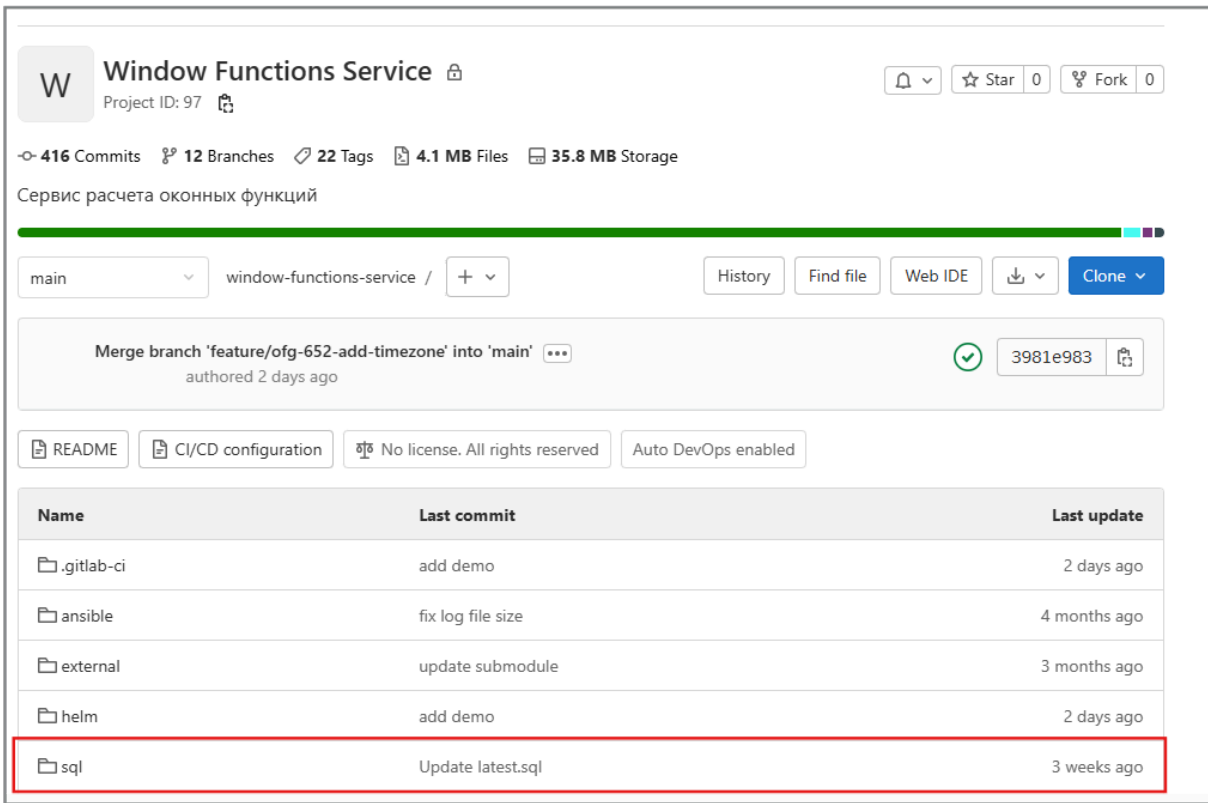


Рис. 5 Пример расположения директории для хранения скриптов создания таблиц

3. Для сервисов предиктивной аналитики ( `predictive-analytics-service` ) и триггеров ( `triggers-service` ) требуется использовать скрипт `quartznet` : [https://github.com/quartznet/quartznet/blob/main/database/tables/tables\\_postgres.sql](https://github.com/quartznet/quartznet/blob/main/database/tables/tables_postgres.sql) .

4. Деплой сервиса PIMS при помощи Gitlab CI/CD:

1. Необходимо открыть репозиторий сервиса и выбрать вкладку CI/CD - Pipelines, как представлено на Рисунке 6. Во вкладке находится список начатых и завершенных процессов сборки (Pipelines).

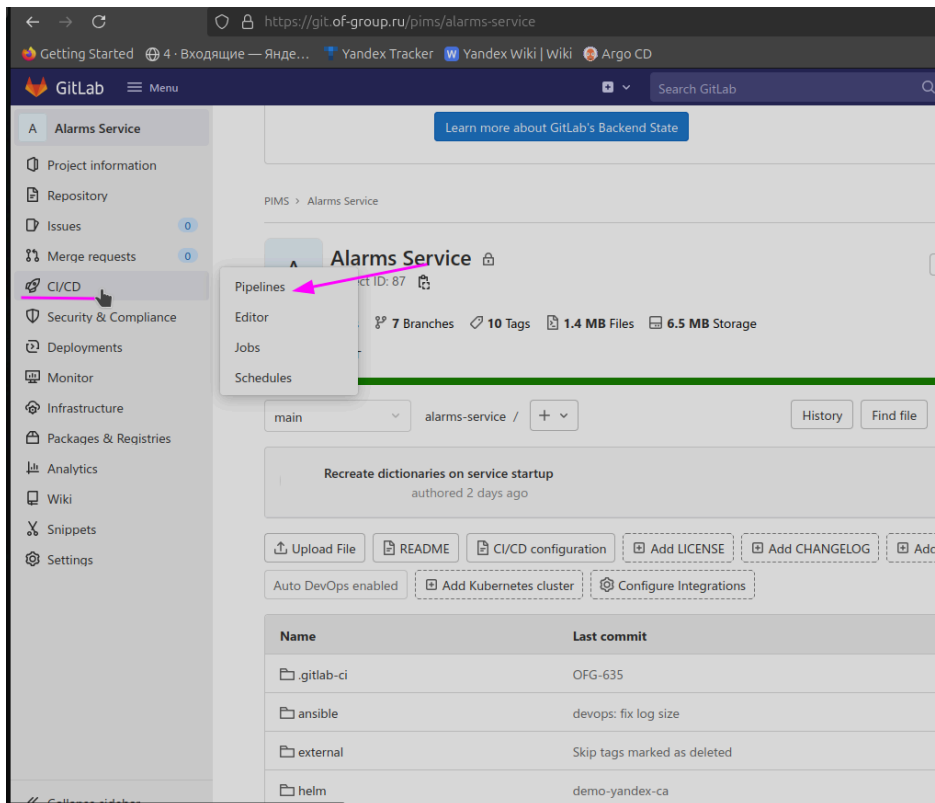


Рис. 6 Вкладка "CI/CD - Pipelines" в репозитории сервиса

2. Для формирования сборки необходимо нажать на кнопку "Run Pipeline", представленную на Рисунке 7.

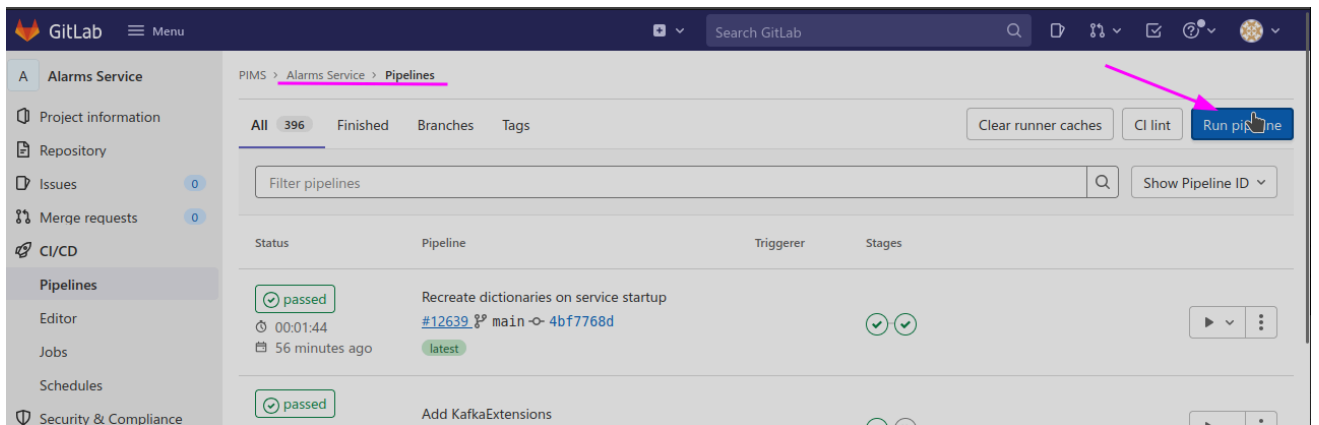


Рис. 7 Кнопка "Run Pipeline" во вкладке "CI/CD - Pipelines" в репозитории сервиса

3. Необходимо выбрать требуемую версию сервиса в поле "Run for branch name or tag" (см. Рисунок 8).

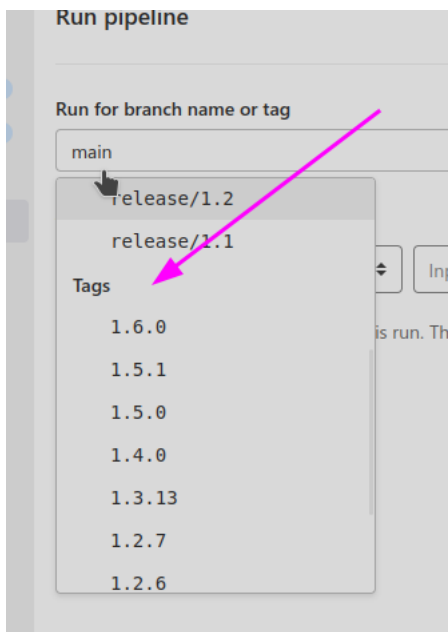


Рис. 8 Выпадающий список выбора версий в поле "Run for branch name or tag"

4. После выбора необходимой версии сервиса необходимо нажать на кнопку "Run Pipeline", представленную на Рисунке 9.

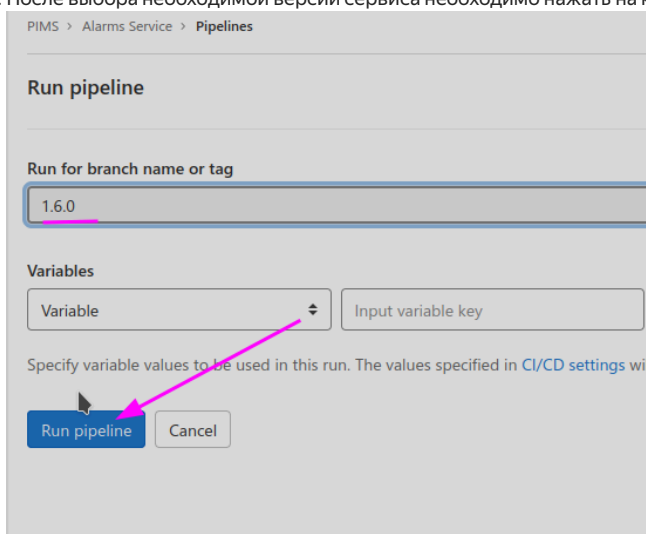


Рис. 9 Кнопка "Run Pipeline" запуска процесса деплоя

Запустится процесс сборки образа docker для сервиса. После выполнения процесса кнопка "build-release" окрасится в зеленый цвет, как представлено на Рисунке 10.

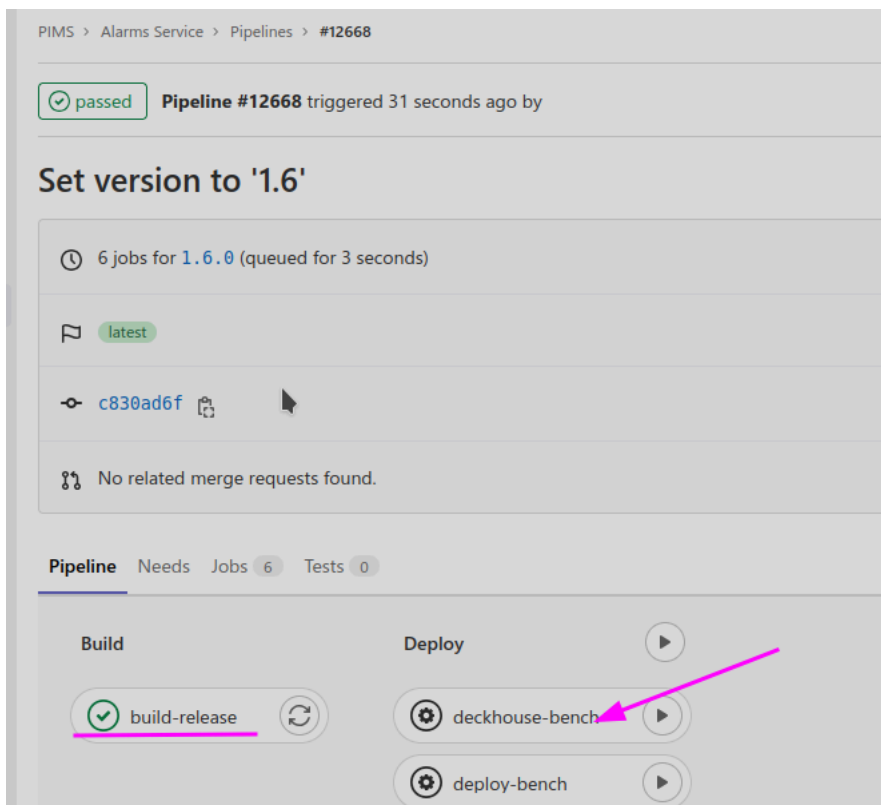


Рис. 10 Индикация кнопки "build-release" при завершении процесс сборки

- Необходимо нажать на кнопку деплоя в требуемом контуре, что запустит процесс переноса сборки в контур. В случае демо-стенда кнопка "ус-demos". Прогресс можно отследить в ссылке на перенос, как представлено на Рисунке 11.

```

69 Namespace:      pims
70 URL:            https://argocd.infra.of-group.ru/applications/bench_alarms-service
71 Sources:
72 - Repo:         https://git.of-group.ru/infra/manifests/pims/alarms-service.git
73 Target:        bench
74 Path:          ./
75 SyncWindow:    Sync Allowed
76 Sync Policy:   Automated (Prune)
77 Sync Status:   Synced to bench
78 Health Status: Healthy
79 GROUP          KIND          NAMESPACE NAME          STATUS HEALTH H
80 OOK MESSAGE
81 Service        pims      alarms-service Synced
82 service/alarms-service unchanged
83 apps           Deployment pims      alarms-service Synced
84 deployment.apps/alarms-service configured
85 monitoring.coreos.com ServiceMonitor pims      alarms-service Synced
86 servicemonitor.monitoring.coreos.com/alarms-service unchanged
87 $ if [ ${exit_code} -ne 0 ]; then git revert $MANIFESTS_COMMIT_SHA --no-edit && git
88 commit --amend -m "Rollback to previous commit $MANIFESTS_PREVIOUS_COMMIT_SHA" && git
89 push && argocd --server $ARGO_SRV --auth-token $ARGO_AUTHTOKEN --grpc-web app get --
90 refresh $APP_NAME > /dev/null && argocd --server $ARGO_SRV --auth-token $ARGO_AUTHTO
91 KEN --grpc-web app sync --prune $APP_NAME --force --prune > /dev/null && argocd --se
92 rver $ARGO_SRV --auth-token $ARGO_AUTHTOKEN --grpc-web app wait $APP_NAME --sync --h
93 ealth --timeout $ARGO_TIMEOUT || exit_code_rollback=$?; else echo "Deployed $CI_PROJ
94 ECT_NAME $CI_ENVIRONMENT_NAME $MANIFESTS_COMMIT_SHA"; fi
95 Deployed alarms-service bench 512372c
96 $ if [ ${exit_code_rollback} -ne 0 ]; then exit 1; elif [ ${exit_code} -eq 0 ]; the
97 n exit 0; elif [ ${exit_code_rollback} -eq 0 ]; then echo "rollback successfull to p
98 revious deploy" && exit 1; else exit 1; fi
99
100 Cleaning up project directory and file based variables
101
102 Job succeeded

```

Рис.11 Мониторинг процесс переноса сборки в контур

- Для проверки рабочего состояния сервиса в кластере необходимо задать в командной строке `kubectl get pods -n pims` и убедиться, что сервис в статусе `Running`, как представлено на Рисунке 12.

```

largid@Largid-Pinux: $ kubectl get pods -n pims
NAME                                READY   STATUS    RESTARTS   AGE
alarms-service-57dd8894db-4rr9q    1/1    Running   0           6m20s
api-gateway-service-5b5f4b6fb7-wq7cz 1/1    Running   0           19d
data-import-service-56d79f86b4-vz1qr 1/1    Running   0           3d8h
data-sink-service-7874f8d7d4-f2rb5   1/1    Running   0           3d8h
datasource-management-service-5bd496cc4d-mtc4w 1/1    Running   0           3d8h
handlers-service-64c6b5cb6d-nlxx9   1/1    Running   0           3d8h
history-service-8557f748cd-49vw2    1/1    Running   0           3d8h
messaging-service-fb65f6fc6-sbs2q   1/1    Running   0           3d8h
mnemo-backend-service-84b7c8c769-v5dxb 1/1    Running   0           3d8h
mnemoeditor-7bcfc6f4dc-941q5        1/1    Running   0           2d
object-model-service-76766b456-mbjzv 1/1    Running   0           3d8h
operator-journal-service-54c47b795b-6bjnv 1/1    Running   0           19d
pims-web-admin-658bf5f65-hxkrn      1/1    Running   0           19d
predictive-analytics-service-56984d44cb-vk6tk 1/1    Running   0           19d
reporting-service-6f7cc76f98-ntgxf   1/1    Running   0           3d8h
reports-web-view-7fb6b58dfb-8sdb5    1/1    Running   0           19d
sap-integration-service-bb8c5c5b8-fltnf 1/1    Running   0           19d
triggers-service-6994896fd4-hgn2d    1/1    Running   0           19d
window-functions-service-57696f6fc-n5m62 1/1    Running   0           3d8h

```

Рис. 12 Мониторинг состояния сервиса в кластере

5. Необходимо повторить шаги 4.1 - 4.6 для всех сервисов из таблицы 1, после чего стенд считается развернутым.
6. Для использования системы необходимо перейти к веб приложениям:
  - o Приложение системного администратора доступно по ссылке: <https://pims-web-admin.demo.of-group.ru/>
  - o Автоматизированное рабочее место доступно по ссылке: <http://mnemoeditor.demo.of-group.ru>
7. Необходимо авторизовать в приложении, заполнив одну из форм, представленных на Рисунке 13. Для авторизации потребуется логин и пароль.

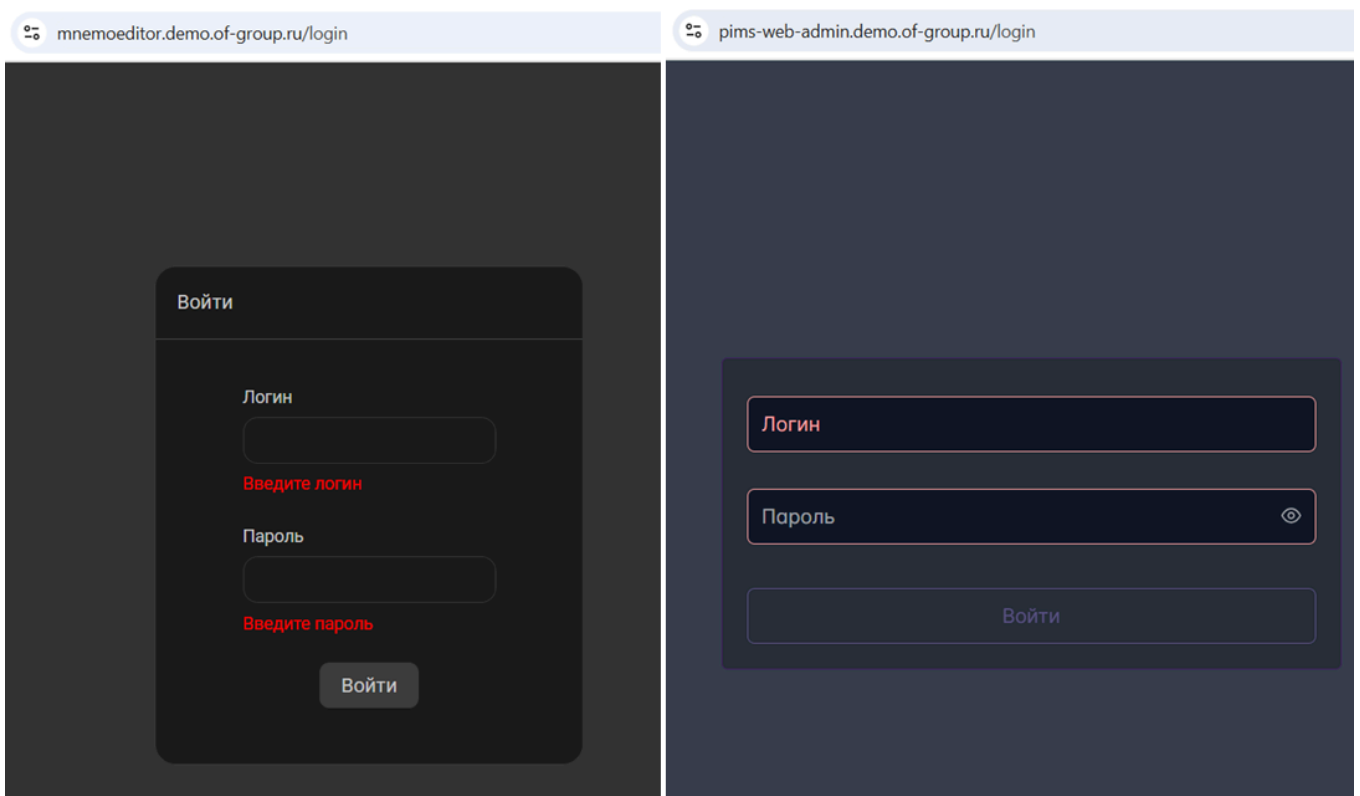


Рис. 13 Формы авторизации в веб приложения (слева - Автоматизированное рабочее место, справа - Графический интерфейс администратора)

В случае успешной авторизации откроется окно выбранного веб-приложения, после чего можно продолжить работу в нем. Руководства пользователя веб-приложениями не входят в описание данной инструкции.