

**Программный комплекс гарантированной доставки и трансформации
данных «сервисная шина «СинтезПлюс» для организации
взаимодействия различных приложений и сервисов**

Руководство администратора

Листов 40

г. Иннополис, 2022

Лист изменений

Версия	Дата изменения	Описание изменения	Автор редакции
1.0	21.04.2022	Создание документа	

Оглавление

Лист изменений.....	2
Оглавление.....	3
Перечень используемых переменных.....	4
1. Введение	6
1.1. НАЗНАЧЕНИЕ СИСТЕМЫ	6
1.2. ФУНКЦИИ ПРОГРАММЫ.....	6
2. Развертывание сервисов	7
2.1. МИНИМАЛЬНЫЕ ТЕХНИЧЕСКИ ТРЕБОВАНИЯ	7
2.2. РАЗВЕРТЫВАНИЕ СЕРВИСОВ.....	7
2.3. НАСТРОЙКА БАЗЫ ДАННЫХ	8
3. Контактная информация производителя системы.....	11
Приложение А Описание проектов.....	12
GTM-BACK	12
ESB-CONFIG-SERVER	15
ESB-EMAIL-ADAPTER	18
ESB-ENCODE-DB.....	20
ESB-FILE-SERVER-ADAPTER	22
ESB-REST-ADAPTER.....	26
ESB-EXPERIENCE-API	28
ESB-MSSQL-ADAPTER.....	31
ESB-POSTGRES-ADAPTER	33
ESB-PROCESS-APPLICATION	35
Приложение В Перечень таблиц, создаваемых при запуске GTM-BACK	39
Приложение С Перечень таблиц, создаваемых при запуске esb-config-server	40

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ ПЕРЕМЕННЫХ

В данном документе использованы переменные, приведенные в Таблице 1.

Таблица 1 — Используемые переменные

Термин/сокращение	Описание
API_PORT_ENV	Порт на который приложение готово принимать сообщения
CAMEL_SPRING_RUN_CONTROLLER_ENV	Переменная для запуска camel-приложения
CONFIG_CLEAR_SERVER_URL_ENV	Host для подключения к config-server
CONFIG_SERVER_URL	Ссылка (url) для получения настроек приложения, которые хранятся в модуле конфигураций (config-server)
DEFAULT_SCHEMA	Схема базы данных, используемая по умолчанию
ENVIRONMENT	Наименование ветки
GTM_AUDIT_LVL	Уровень заполнения журнала аудита. OFF – аудит отключен BASE – базовая информация о входящих запросах FULL – базовый + сохранение параметров переменных
GTM_LDAP_AUTHTYPE	Тип LDAP аутентификации. Ldap 2.0. поддерживает следующие типы anonymous, simple, Kerberos v4
GTM_LDAP_BASEDN	Базовый dn поиска пользователя в LDAP
GTM_LDAP_POOLED	Разрешить использование пула соединений LDAP
GTM_LDAP_PWD	Пароль пользователя LDAP
GTM_LDAP_URL	Адрес сервера LDAP
GTM_LDAP_USER	Имя учетной записи пользователя LDAP
GTM_PROXY_CONTAINERS	Массив адресов контейнеров для проксирования запросов через gateway
JPA_DATASOURCE_INITIAL	spring.jpa.defer-datasource-initialization. При значении true отложит инициализацию источника данных до тех пор, пока не будут созданы и инициализированы все bean-компоненты EntityManagerFactory.
JPA_DB_PLATFORM	spring.jpa.database-platform используемый диалект базы данных
LOG4J2_CONFIG	Ссылка (Url) для получения log4j конфигурации приложения которые хранятся в config-server'e
LOG_DB_HOST	jdbc-url базы данных в которую происходит логирование
LOG_DB_LOGIN	Логин базы данных в которую происходит логирование
LOG_DB_PASSWORD	Пароль базы данных в которую происходит логирование
LOGGING_GLOG_HOST	Адрес сервера Graylog
LOGGING_GLOG_PORT	Порт сервера Graylog
LOGGING_LVL_ROOT	Уровень логирования. Возможные значения: debug, error, fatal, info, off, trace, warn (https://docs.spring.io/spring-boot/docs/current/reference/html/features.html#features.logging)
LOGGING_PATTERN_CONSOLE	Паттерн вывода данных в консоль логирования

Термин/сокращение	Описание
	(https://docs.spring.io/spring-boot/docs/current/reference/html/features.html#features.logging)
MINIO_ACCESS_KEY_ENV	Ключ доступа (Access key) в файловое хранилище minio
MINIO_FILE_BUCKET_NAME_ENV	Название бакета, где хранятся файлы в файловом хранилище minio
MINIO_PATH_ENV	Host minio
MINIO_SECRET_KEY_ENV	Secret key в файловое хранилище minio
MINIO_SHORTCUT_BUCKET_NAME_ENV	Название бакета, где хранятся короткие ссылки в файловом хранилище minio
MVC_PATH_MATCH_STRAT	
NAMESPACE_ENV	Наименование пространства имен
NODE_PORT	Внешний порт сервиса
OT_JAEGER_HTTP_SENDER_ENV	Host jaeger
OT_JAEGER_SERVICE_NAME_ENV	Имя сервиса как он будет индексироваться в jaeger
SENTRY_DSN	Адрес сервера приёма логов sentry data source name ассоциируемый с данным проектом
SENTRY_ENVIR	Привязка событий к переменной environment в sentry. (По сути тег поиска)
SERVICENAME	Уникальное имя сервиса
SERVER_PORT	Используемый порт сервера приложений
SERVICE_PORT	Внутренний порт сервиса
SPRING_DATA_DRIVER	Драйвер используемый SpringData
SPRING_DATA_PWD	Пароль пользователя бд SpringData
SPRING_DATA_URL	URL для доступа к бд SpringData
SPRING_DATA_USER	Имя пользователя бд SpringData
SPRING_JRA_HIBERNATE	Свойство spring.jpa.hibernate.ddl-auto
SPRING_JRA_PROPERTIES	Свойство spring.jpa.properties.hibernate.default_schema, схема используемая по умолчанию (может быть указана в рамках url)
SPRING_INDENT_OUT	Определяет необходимость форматирования возвращаемых json-данных
SPRING_LIQ_CHANGE_LOG	Локация мастер-скрипта Liquibase.
SPRING_LIQ_ENABLE	Признак того что Liquibase включен
SPRING_PROFILES_ACTIVE	Активный профиль spring-boot

1. Введение

1.1. Назначение системы

Полное наименование системы: Программный комплекс гарантированной доставки и трансформации данных «сервисная шина «СинтезПлюс» для организации взаимодействия различных приложений и сервисов (далее – Система).

Краткое наименование системы: Сервисная шина СинтезПлюс (СШ СП)

Система предназначена для автоматизированного управления производственными процессами сбора и обработки геолого-промысловых данных, оперативного обеспечения персонала достоверной и полной информацией о режимах работы производственных объектов, учета и анализа производственной деятельности структурных подразделений Заказчика.

1.2. Функции программы

Система содержит следующие основные функциональные компоненты:

- Сервисная шина системы (ESB). Предназначена для гарантированного информационного обмена между системой и сторонними сервисами;
- Бизнес-приложение. Предназначено для выполнения бизнес функциональности системы.

2. Развертывание сервисов

2.1. Минимальные технические требования

Минимальные требования конфигурация сервера баз данных указана в Таблице 2.

Таблица 2 — Конфигурация сервера баз данных

Сервер баз данных	
CPU core	2
RAM, ГБ	8
HDD, ГБ	Тип диска: VMDK Объем: 500ГБ

Минимальные требования конфигурация master-ноды указана в Таблице 3.

Таблица 3 — Конфигурация master-ноды

Сервер Master	
CPU core	4
RAM, ГБ	8
HDD, ГБ	Тип диска: VMDK Объем: 100ГБ

Минимальные требования конфигурация worker-ноды указана в Таблице 3.

Таблица 4 — Конфигурация worker-ноды

Сервер Worker	
CPU core	8
RAM, ГБ	16
HDD, ГБ	Тип диска: VMDK Объем: 200ГБ

2.2. Развертывание сервисов

В данном разделе приведены особенности переноса и развертывания сервисов на кластер k8s.

Официальная документация по работе с docker-контейнерами доступна на сайте <https://docs.docker.com>.

Официальная документация по работе с Kubernetes доступна на сайте <https://kubernetes.io>.

Ниже приведен перечень действий по переносу и развертывания сервисов:

1. Загрузка контейнера сервиса используя команду, указанную в Листинге 1.

Листинг 1 – Загрузка образа

```
# docker login https://harbor.stageogip.ru          ##Подключение к репозиторию
# docker pull harbor.stageogip.ru/gtm/<Имя сервиса>:<ветка>
```

2. Переименование контейнера для последующей загрузки в репозиторий указана в Листинге 2.

Листинг 2 – Переименование образа

```
# docker images -a          ###Вывод всех доступных репозиториев для дальнейшего переименования
*****
      REPOSITORY                TAG                IMAGE ID
harbor.stageogip.ru/gtm/<Имя сервиса>    develop           e51524ebc53c
*****
# docker tag <REPOSITORY>:IMAGE ID> harbor.stageogip.ru/gtm/<Имя сервиса>
```

3. Загрузка контейнера в репозиторий указана в Листинге 3.

Листинг 3 – Загрузка образа

```
#docker pull <Адрес сервера реестра>/<Имя сервиса>:<ветка>
```

4. При недоступности сервера реестра Docker-контейнеров в Листинге 4 приведены шаги по переносу контейнеров на кластер kubernetes.

Листинг 4 – Сохранение контейнера

```
# docker save <REPOSITORY>:IMAGE ID> > /path to save/<REPOSITORY>.tar.gz
```

5. На целевом сервере, после переноса, контейнер необходимо запустить, как указано в Листинге 5. При запуске контейнера namespace создается согласно конфигурационному файлу *kubemanifest.yaml*, которые приведены в Приложении А. Если необходимо поменять namespace, то нужно внести корректировки в переменные файла *kubemanifest.yaml*.

Листинг 5 – Запуск сервиса

```
# kubectl apply -f <kubemanifest>.yaml
```

Для каждого сервиса создан уникальный файл конфигураций *kubemanifest.yaml* и соответствует имени сервиса.

2.3. Настройка базы данных

В данном разделе описываются особенности настройки баз данных сервисов ГТМ использующих СУБД PostgreSQL.

Официальная документация по работе с СУБД PostgreSQL доступна на сайте <https://www.postgresql.org>.

Перед настройкой, в терминале управления PostgreSQL, необходимо выполнить следующие действия:

- создать пустую БД командой «*CREATE DATABASE <database_name>;*»;
- создать пользователя БД командой «*CREATE USER <database_username> WITH password 'p@ssword';*»;

После создания БД и пользователя необходимо передать через переменные окружения контейнера ролевого разграничения (gtm-back и esb-config-server) учетные данные пользователя с правами создания схем и объектов (таблицы, представления, сиквенсы, функции).

При запуске контейнера выполнится набор sql-скриптов по созданию необходимых сущностей в БД. Необходимые контейнеры приведены в списке ниже:

- контейнер gtm-back – отвечает за разворачивание необходимых схем и таблиц для бизнес-приложений;
- контейнер esb-config-server – отвечает за создание необходимых схем и приложений для сервисной шины (ESB).

При первом запуске контейнера проверяется наличие сущностей в БД и при отсутствии выполняются sql-скрипты. После запуска контейнера в БД создаются новые схемы с необходимыми таблицами. Перечень объектов, создаваемых в базе данных, приведен в Приложении Б и В.

При повторном запуске контейнеров ранее выполненные sql-скрипты не запускаются.

Все ошибки при запуске контейнеров gtm-back и esb-config-server можно посмотреть в консоли соответствующего контейнера.

Возможные проблемы при запуске контейнера и способы решения указаны в Таблице 5.

Таблица 5 — Проблемы и способы решения

№	Описание проблемы	Способы решения
1	В консоли контейнера ошибки на отсутствие доступа к базе данных	Проверить учетные данные, прописанные в переменных окружения контейнера. Проверить доступность базы данных из контейнера по dns-имени или по ip-адресу сервера базы данных.
2	В консоли контейнера ошибки при выполнении sql-скриптов.	При первом запуске контейнера в базе данных в схемах (gtm-web для gtm-back и esb-repos для config-server) не должно быть сущностей (объектов). Кроме этого, необходимо при наличии удалить таблицы databasechangelog и databasechangeloglock в схеме public.
3	В консоли контейнера ошибки на отсутствие доступа к Active Directory	Проверить учетные данные для AD, прописанные в переменных окружения контейнера.

		Проверить доступность Active Directory из контейнера (gtm-back) по dns-имени или по ip-адресу сервера базы данных.
4	Прочие ошибки в консоли контейнера	Обратиться к вендору

3. Контактная информация производителя системы

3.1 Юридическая информация

Информация о юридическом лице компании:

название компании: АНО ВО «Университет Иннополис»;

юридический адрес: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д.1;

ОГРН: 1121600006142;

ИНН: 1655258235.

3.2 Контактная информация службы технической поддержки

Данные для связи со специалистами службы поддержки программного продукта:

Контакты: 8 (843) 203-92-53, university@innopolis.ru;

Фактический адрес размещения инфраструктуры разработки: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д.1;

Фактический адрес размещения разработчиков: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д.1;

Фактический адрес размещения службы технической поддержки: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д.1;

Фактический адрес размещения серверов: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д.1.

Приложение А Описание проектов

Список *kubemanifest.yaml*, для каждого сервиса, указан ниже:

1. *gtm-back.yaml*;
2. *esb-config-server.yaml*;
3. *esb-email-adapter.yaml*;
4. *esb-file-server-adapter.yaml*;
5. *esb-rest-adapter.yaml*;
6. *esb-experience-api.yaml*;
7. *esb-mssql-adapter.yaml*;
8. *esb-postgres-adapter.yaml*;
9. *esb-process-application.yaml*.

Листинг *kubemanifest.yaml* и описание проектов указаны ниже:

gtm-back

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер модуля ролевого разграничения доступа. Содержит в себе сервисную логику по аутентификации, правовому и ролевому доступу, аудита и логгирования, предоставляет back-сервисы для shell приложения и единой точкой входа для дочерних микрофронт проектов. При первом запуске, контейнер разворачивает необходимые схемы и сущности в базе данных PostgreSQL (хранилище ИЦП) с настройками подключения, переданными через переменные окружения.

Требования: Наличие доступа к БД PostgreSQL (хранилище ИЦП) и к каталогу Active Directory, где координаты подключения задаются через переменные окружения.

Листинг А.4 – gtm-back.yaml

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gtm-back
  namespace: gtm-app-develop
labels:
  app: gtm-back
spec:
```

```
replicas: 1
selector:
  matchLabels:
    app: gtm-back
strategy:
  rollingUpdate:
    maxSurge: 0
    maxUnavailable: 1
  type: RollingUpdate
template:
  metadata:
    labels:
      app: gtm-back
  spec:
    containers:
      - name: gtm-back
        image: harbor.stageogip.ru/gtm/gtm-back:develop
        imagePullPolicy: Always
        ports:
          - name: http
            containerPort: 8080
        env:
          - name: ENVIRONMENT
            value: "develop"
          - name: GTM_AUDIT_LVL
            value: "FULL;"
          - name: GTM_LDAP_AUTHTYPE
            value: "simple"
          - name: GTM_LDAP_BASEDN
            value: "DC=Значение,DC=Значение"
          - name: GTM_LDAP_POOLED
            value: "true"
          - name: GTM_LDAP_PWD
            value: "Pswd"
          - name: GTM_LDAP_URL
            value: "ldap://IP"
          - name: GTM_LDAP_USER
            value: "User"
          - name: GTM_PROXY_CONTAINERS
```

```
value: '{"bc-planning-dev': 'http://gtm-planning-back:8081', 'esb': 'http://poc-experience-api:30131', 'bc-dictionary-dev': 'http://gtm-dictionaries-back:8082', 'bc-gis-dev': 'http://gtm-gis-back:8083', 'gtm-plan-back': 'http://gtm-plan-back:8080', 'gtm-request-back': 'http://gtm-request-back:8080', 'camunda-poc': 'http://camunda-embd-poc:8083'}'
```

```
- name: LOGGING_GLOG_HOST
```

```
value: "IP"
```

```
- name: LOGGING_GLOG_PORT
```

```
value: "12201"
```

```
- name: LOGGING_LVL_ROOT
```

```
value: "INFO"
```

```
- name: LOGGING_PATTERN_CONSOLE
```

```
value: ""
```

```
- name: SENTRY_DSN
```

```
value: "http://[SENTRY_DNS_VALUE]@[GRAYLOG_HOST]:9001/7"
```

```
- name: SENTRY_ENVIR
```

```
value: "develop"
```

```
- name: SPRING_BANNER_LOC
```

```
value: "classpath:/static/banner.txt"
```

```
- name: SPRING_DATA_PWD
```

```
value: "Pswd"
```

```
- name: SPRING_DATA_URL
```

```
value: "jdbc:postgresql://[postgres_host]: [postgres_port]/[postgres_database]"
```

```
- name: SPRING_DATA_USER
```

```
value: "User"
```

```
- name: SPRING_JRA_HIBERNATE
```

```
value: "none"
```

```
- name: SPRING_JRA_PROPERTIES
```

```
value: "gtm_web"
```

```
- name: SPRING_LIQ_CHANGE_LOG
```

```
value: "db/changelog/db.master-changelog.yaml"
```

```
- name: SPRING_LIQ_ENABLE
```

```
value: "true"
```

```
---
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
name: gtm-back
```

```
namespace: gtm-app-develop
```

```
spec:
  selector:
    app: gtm-back
  ports:
    - name: http
      protocol: TCP
      port: 8080
      targetPort: http

---
apiVersion: v1
kind: Service
metadata:
  name: gtm-back-expose
  namespace: gtm-app-develop
spec:
  selector:
    app: gtm-back
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 30031
```

esb-config-server

Тип проекта: серверный SpringBoot-проект.

Описание: Контейнер сервиса модуля конфигураций, предоставляющего Rest API для получения и управления конфигурациями маршрутов сервисной шины. При первом запуске, контейнер разворачивает необходимые схемы и сущности в базе данных postgres (внутреннее хранилище сервисной шины) с настройками подключения, переданными через переменные окружения. Основные функции представлены в списке ниже:

- хранение/управление конфигурациями маршрутов интеграции;
- хранение/управление конфигурациями приложений;
- хранение/управление скриптами трансформаций.

Требования: Наличие доступа до БД PostgreSQL (внутреннее хранилище сервисной шины).

```
---
apiVersion: v1
kind: Service
metadata:
  name: esb-config-server
  namespace: gtm-esb-develop
spec:
  selector:
    app: esb-config-server
  ports:
    - name: http
      protocol: TCP
      port: 9000
      targetPort: http

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: esb-config-server
  namespace: gtm-esb-develop
  labels:
    app: esb-config-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: esb-config-server
  strategy:
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: esb-config-server
    spec:
```

containers:

- name: esb-config-server

image: harbor.stageogip.ru/gtm/ esb-config-server:develop

imagePullPolicy: Always

ports:

- name: http

containerPort: 9000

env:

- name: ENVIRONMENT

value: "develop"

- name: GRAYLOG_HOST

value: "[GRAYLOG_HOST]"

- name: GRAYLOG_PORT

value: "12201"

- name: SPRING_PROFILES_ACTIVE

value: "develop"

- name: LOG_DB_HOST

value: "jdbc:postgresql://[esb_postgres_host]: [postgres_port]/[esb_postgres_database] "

- name: LOG_DB_LOGIN

value: "User"

- name: LOG_DB_PASSWORD

value: "Pswd"

- name: API_PORT

value: "9000"

- name: CAMEL_SPRING_RUN_CONTROLLER

value: "true"

- name: CAMEL_DISCOVER_OBJECT_MAPPER

value: "true"

- name: OT_JAEGER_HTTP_SENDER

value: "http://[JAEGER_HOST]:[JAEGER_PORT]/api/traces"

- name: OT_JAEGER_SERVICE_NAME

value: "CONFIG_SERVER_DEVELOP"

- name: SENTRY_DSN

value: "http://[SENTRY_DNS_VALUE]@[GRAYLOG_HOST]:9001/5"

- name: SPRING_DATA_DRIVER

value: "org.postgresql.Driver"

- name: SPRING_LIQ_DROP_FIRST

value: "false"

- name: SPRING_LIQ_ENABLE

```
    value: "true"
  - name: GELF_PORT
    value: "12201"
  - name: GELF_HOST
    value: "udp:[GRAYLOG_HOST]"
  - name: SPRING_LIQ_CONTEXT
    value: "'base', 'develop'"
```

```
apiVersion: v1
kind: Service
metadata:
  name: config-server-expose
  namespace: gtm-esb-develop
spec:
  selector:
    app: config-server
  type: NodePort
  ports:
    - protocol: TCP
      port: 9000
      targetPort: 9000
      nodePort: 30138
```

esb-email-adapter

Тип проекта: серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API для отправки сообщений на электронную почту. Основные функции представлены в списке ниже:

- отправка email по REST-вызову
- отправка email, полученного через JMS-очередь.

Требования: Наличие доступа до модуля конфигураций Config-server.

Листинг A.25 – esb-email-adapter.yaml

```
apiVersion: v1
kind: Service
```

```
metadata:
  name: email-adapter
  namespace: gtm-esb-develop
spec:
  selector:
    app: email-adapter
  ports:
    - name: port
      protocol: TCP
      port: 9092
      targetPort: 9092

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: email-adapter
  namespace: gtm-esb-develop
  labels:
    app: email-adapter
spec:
  replicas: 1
  selector:
    matchLabels:
      app: email-adapter
  template:
    metadata:
      labels:
        app: email-adapter
    spec:
      containers:
        - name: email-adapter
          image: harbor.stageogip.ru/gtm/email-adapter:develop
          imagePullPolicy: Always
          ports:
            - name: port
              containerPort: 9092
      envFrom:
        - configMapRef:
```

```
name: email-adapter-config
```

```
---
```

```
apiVersion: v1
```

```
kind: ConfigMap
```

```
metadata:
```

```
name: email-adapter-config
```

```
namespace: gtm-esb-develop
```

```
labels:
```

```
app: email-adapter
```

```
data:
```

```
ENVIRONMENT: develop
```

```
CONFIG_SERVER_URL: http://config-server:9000/api/config/develop/resource/EMAIL_APP/conf
```

```
LOG4J2_CONFIG: http://config-server:9000/api/config/develop/resource/EMAIL_APP/log4j2.xml
```

esb-encode-db

Тип проекта: серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API по взаимодействию с СУБД PostgreSQL, содержащей информацию по сопоставлениям (перекодировкам) справочников систем.

Требования: Наличие доступа до модуля конфигураций Config-server.

Листинг А.26 – encode-db.yaml

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
name: encode-db
```

```
namespace: gtm-dev
```

```
selfLink: /apis/apps/v1/namespaces/gtm-dev/deployments/encode-db
```

```
spec:
```

```
replicas: 1
```

```
selector:
```

```
matchLabels:
```

```
app: encode-db
```

```
template:
```

```
metadata:
```

creationTimestamp: null

labels:

app: encode-db

spec:

containers:

- name: encode-db

image: 'harbor.stageogip.ru/gtm/poc-postgres-adapter:latest'

ports:

- containerPort: 9095

protocol: TCP

envFrom:

- configMapRef:

name: encode-db-config

resources:

limits:

cpu: 800m

memory: 4Gi

requests:

cpu: 200m

memory: 400Mi

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

imagePullPolicy: Always

restartPolicy: Always

terminationGracePeriodSeconds: 30

dnsPolicy: ClusterFirst

securityContext: {}

schedulerName: default-scheduler

strategy:

type: RollingUpdate

rollingUpdate:

maxUnavailable: 25%

maxSurge: 25%

revisionHistoryLimit: 10

progressDeadlineSeconds: 600

apiVersion: v1

```
kind: ConfigMap
metadata:
  name: encode-db-config
  namespace: gtm-dev
  selfLink: /api/v1/namespaces/gtm-dev/configmaps/encode-db-config
data:
  CONFIG_SERVER_URL: 'http://config-server:9000/api/config/dev/resource/ENCODE_DB/conf'
  ENVIRONMENT: dev
  LOG4J2_CONFIG: 'http://config-server:9000/api/config/dev/resource/ENCODE_DB/log4j2.xml'
---
apiVersion: v1
kind: Service
metadata:
  name: encode-db
  namespace: gtm-dev
  selfLink: /api/v1/namespaces/gtm-dev/services/encode-db
status:
  loadBalancer: {}
spec:
  ports:
    - protocol: TCP
      port: 9095
      targetPort: 9095
      nodePort: 31514
  selector:
    app: encode-db
  type: NodePort
  sessionAffinity: None
  externalTrafficPolicy: Cluster
```

esb-file-server-adapter

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API по взаимодействию с файловыми хранилищами типа S3. Основные функции представлены в списке ниже:

- сохранение файлов в S3-хранилище;
- скачивание файлов из S3-хранилища;
- скачивание части файла из S3-хранилища;
- получение списка файлов, которые хранятся в S3-хранилище;
- генерация коротких ссылок для скачивания файлов из S3-хранилища.

Требования: Наличие доступа до модуля конфигураций Config-server и до файлового хранилища.

Листинг А.27 – file-server-adapter.yaml

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: file-server-adapter
  namespace: gtm-esb-develop
labels:
  app: file-server-adapter
spec:
  replicas: 1
  selector:
    matchLabels:
      app: file-server-adapter
  strategy:
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: file-server-adapter
    spec:
      containers:
        - name: file-server-adapter
```

image: harbor.stageogip.ru/gtm/file-server-adapter:develop

imagePullPolicy: Always

ports:

- name: http

containerPort: 9006

env:

- name: ENVIRONMENT

value: "develop"

- name: API_PORT

value: "9006"

- name: CAMEL_DISCOVER_OBJECT_MAPPER

value: "CAMEL_DISCOVER_OBJECT_MAPPER_ENV"

- name: CAMEL_SPRING_RUN_CONTROLLER

value: "true"

- name: CONFIG_CLEAR_SERVER_URL

value: "http://config-server:9000/api/config/develop/"

- name: MINIO_ACCESS_KEY

value: "gtmaccesskey"

- name: MINIO_FILE_BUCKET_NAME

value: "camelfilebn"

- name: MINIO_PATH

value: "http://IP:9002"

- name: MINIO_SECRET_KEY

value: "gtmsecretkey"

- name: MINIO_SHORTCUT_BUCKET_NAME

value: "camelshortcutbn"

- name: OT_JAEGER_HTTP_SENDER

value: "http://[JAEGER_HOST]:[JAEGER_PORT]/api/traces"

- name: OT_JAEGER_SERVICE_NAME

value: "FILE_SERVER_ADAPTER_DEVELOP"

- name: SENTRY_DSN

value: "http://[SENTRY_DNS_VALUE]@[GRAYLOG_HOST]:9001/5"

- name: SPRING_DATA_PWD

value: "SPRING_DATA_PWD_ENV"

- name: SPRING_DATA_URL

value: "SPRING_DATA_URL_ENV"

- name: SPRING_DATA_USER

value: "SPRING_DATA_USER_ENV"

- name: SPRING_DATA_DRIVER

```
  value: "SPRING_DATA_DRIVER_ENV"
- name: SPRING_LIQ_DROP_FIRST
  value: "SPRING_LIQ_DROP_FIRST_ENV"
- name: SPRING_LIQ_ENABLE
  value: "SPRING_LIQ_ENABLE_ENV"
- name: SPRING_LIQ_CONTEXT
  value: "SPRING_LIQ_CONTEXT_ENV"
- name: GELF_HOST
  value: "udp:IP.100"
- name: GELF_PORT
  value: "12201"
```

```
apiVersion: v1
kind: Service
metadata:
  name: file-server-adapter
  namespace: gtm-esb-develop
spec:
  selector:
    app: file-server-adapter
  ports:
    - name: http
      protocol: TCP
      port: 9006
      targetPort: http
```

```
apiVersion: v1
kind: Service
metadata:
  name: file-server-adapter-expose
  namespace: gtm-esb-develop
spec:
  selector:
    app: file-server-adapter
  type: NodePort
  ports:
    - protocol: TCP
```

```
port: 9006
targetPort: 9006
nodePort: 31404
```

esb-rest-adapter

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API по взаимодействию с ресурсами (информационными системами, программными комплексами) по Rest API.

Требования: Наличие доступа до модуля конфигураций Config-server и до Rest-сервиса.

Листинг A.29 – gtm-rest-adapter.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gtm-rest-adapter
  namespace: gtm-dev
  selfLink: /apis/apps/v1/namespaces/gtm-dev/deployments/gtm-rest-adapter
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gtm-rest-adapter
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: gtm-rest-adapter
  spec:
    containers:
      - name: gtm-rest-adapter
        image: 'harbor.stageogip.ru/gtm/poc-rest-adapter:latest'
        ports:
          - containerPort: 9095
            protocol: TCP
        envFrom:
          - configMapRef:
              name: gtm-rest-adapter-config
```

```
resources:
  limits:
    cpu: 800m
    memory: 4Gi
  requests:
    cpu: 200m
    memory: 400Mi
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
  imagePullPolicy: Always
restartPolicy: Always
terminationGracePeriodSeconds: 30
dnsPolicy: ClusterFirst
securityContext: {}
schedulerName: default-scheduler
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 25%
    maxSurge: 25%
  revisionHistoryLimit: 10
  progressDeadlineSeconds: 600
---

apiVersion: v1
kind: ConfigMap
metadata:
  name: gtm-rest-adapter-config
  namespace: gtm-dev
  selfLink: /api/v1/namespaces/gtm-dev/configmaps/gtm-rest-adapter-config
data:
  CONFIG_SERVER_URL: 'http://config-server:9000/api/config/dev/resource/GTM_REST/conf'
  ENVIRONMENT: dev
  LOG4J2_CONFIG: 'http://config-server:9000/api/config/dev/resource/GTM_REST/log4j2.xml'
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: gtm-rest-adapter
  namespace: gtm-dev
  selfLink: /api/v1/namespaces/gtm-dev/services/gtm-rest-adapter
spec:
  ports:
    - protocol: TCP
      port: 9095
      targetPort: 9095
  selector:
    app: gtm-rest-adapter
  type: ClusterIP
  sessionAffinity: None
```

esb-experience-api

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер сервиса, предоставляющего Rest API для взаимодействия с сервисной шиной. Является единой точкой по взаимодействию с сервисной шиной (ESB). Основные функции представлены в списке ниже:

- предоставление доступа к ресурсам по протоколу REST;
- запуск маршрутов интеграции по протоколу REST;
- доступ к модулю конфигураций (Config-server);
- доступ к модулю передачи данных (Process-application).

Требования: Наличие доступа до всех модулей и адаптеров сервисной шины, а именно:

- email-adapter;
- encode-db;
- file-server-adapter;
- gtm-adapter;
- gtm-rest-adapter;
- poc-experience-api;
- poc-mssql-adapter;
- poc-postgres-adapter.

```
---
apiVersion: v1
kind: Service
metadata:
  name: poc-experience-api
  namespace: gtm-esb-develop
spec:
  selector:
    app: poc-experience-api
  ports:
    - name: port1
      protocol: TCP
      port: 7092
      targetPort: 7092
    - name: port2
      protocol: TCP
      port: 8088
      targetPort: 8088
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: poc-experience-api
  namespace: gtm-esb-develop
labels:
  app: poc-experience-api
spec:
  replicas: 1
  selector:
    matchLabels:
      app: poc-experience-api
  strategy:
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
```

labels:

app: poc-experience-api

spec:

containers:

- name: poc-experience-api

image: harbor.stageogip.ru/gtm/poc-experience-api:develop

imagePullPolicy: Always

ports:

- name: port1

containerPort: 7092

- name: port2

containerPort: 8088

env:

- name: ENVIRONMENT

value: "develop"

- name: CONFIG_SERVER_URL

value: "http://config-server:9000/api/config/develop/resource/EXPERIENCE_API/conf"

- name: LOG4J2_CONFIG

value: "http://config-server:9000/api/config/develop/resource/EXPERIENCE_API/log4j2.xml"

- name: SPRING_PROFILES_ACTIVE

value: "develop"

- name: LOG_DB_HOST

value: "jdbc:postgresql://[esb_postgres_host]: [postgres_port]/[esb_postgres_database]"

- name: LOG_DB_LOGIN

value: "User"

- name: LOG_DB_PASSWORD

value: "Pswd"

apiVersion: v1

kind: Service

metadata:

name: poc-experience-api-expose

namespace: gtm-esb-develop

spec:

selector:

app: poc-experience-api

type: NodePort

ports:

```
- name: port1
  protocol: TCP
  port: 7092
  targetPort: 7092
  nodePort: 30131
- name: port2
  protocol: TCP
  port: 8088
  targetPort: 8088
  nodePort: 30132
```

esb-mssql-adapter

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API по взаимодействию (загрузка, выгрузка данных) с базой данных MS SQL. Основные функции представлены в списке ниже:

- предоставление универсального (конфигурируемого) доступа к базам данных типа MSSQL;
- исполнение SQL-скриптов;
- вызов хранимых процедур.

Требования: Наличие доступа до модуля конфигураций Config-server и до базы данных MS SQL.

Листинг А.31 – esb-mssql-adapter.yaml

```
---
apiVersion: v1
kind: Service
metadata:
  name: poc-mssql-adapter
  namespace: gtm-esb-develop
spec:
  selector:
    app: poc-mssql-adapter
  ports:
    - name: http
      protocol: TCP
```

port: 9093

targetPort: http

apiVersion: apps/v1

kind: Deployment

metadata:

name: poc-mssql-adapter

namespace: gtm-esb-develop

labels:

app: poc-mssql-adapter

spec:

replicas: 1

selector:

matchLabels:

app: poc-mssql-adapter

strategy:

rollingUpdate:

maxSurge: 0

maxUnavailable: 1

type: RollingUpdate

template:

metadata:

labels:

app: poc-mssql-adapter

spec:

containers:

- name: poc-mssql-adapter

image: harbor.stageogip.ru/gtm/poc-mssql-adapter:develop

imagePullPolicy: Always

ports:

- name: http

containerPort: 9093

env:

- name: ENVIRONMENT

value: "develop"

- name: CONFIG_SERVER_URL

value: "http://config-server:9000/api/config/develop/resource/ARMITS/conf"

- name: LOG4J2_CONFIG

```
value: "http://config-server:9000/api/config/develop/resource/ARMITS/log4j2.xml"
```

```
---
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: poc-mssql-adapter-expose
```

```
  namespace: gtm-esb-develop
```

```
spec:
```

```
  selector:
```

```
    app: poc-mssql-adapter
```

```
  type: NodePort
```

```
  ports:
```

```
    - protocol: TCP
```

```
      port: 9093
```

```
      targetPort: 9093
```

```
      nodePort: 30133
```

esb-postgres-adapter

Тип проекта: Серверный SpringBoot-проект.

Описание: Контейнер адаптера сервисной шины, предоставляющего Rest API по взаимодействию (загрузка, выгрузка данных) с СУБД PostgreSQL. Основные функции представлены в списке ниже:

- предоставление универсального (конфигурируемого) доступа к БД типа PostgreSQL;
- исполнение SQL-скриптов;
- вызов хранимых процедур.

Требования: Наличие доступа до модуля конфигураций Config-server и до БД PostgreSQL.

Листинг А.32 – esb-postgres-adapter.yaml

```
---
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: poc-postgres-adapter
```

```
  namespace: gtm-esb-develop
```

```
spec:
```

```
selector:
  app: poc-postgres-adapter
ports:
  - name: http
    protocol: TCP
    port: 9092
    targetPort: http

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: poc-postgres-adapter
  namespace: gtm-esb-develop
  labels:
    app: poc-postgres-adapter
spec:
  replicas: 1
  selector:
    matchLabels:
      app: poc-postgres-adapter
  strategy:
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: poc-postgres-adapter
    spec:
      containers:
        - name: poc-postgres-adapter
          image: harbor.stageogip.ru/gtm/poc-postgres-adapter:develop
          imagePullPolicy: Always
          ports:
            - name: http
              containerPort: 9092
      env:
```

```
- name: ENVIRONMENT
  value: "develop"
- name: CONFIG_SERVER_URL
  value: "http://config-server:9000/api/config/develop/resource/POSTGRES/conf"
- name: LOG4J2_CONFIG
  value: "http://config-server:9000/api/config/develop/resource/POSTGRES/log4j2.xml"
- name: GRAYLOG_HOST
  value: "[GRAYLOG_HOST]"
- name: GRAYLOG_PORT
  value: "12201"
- name: SPRING_PROFILES_ACTIVE
  value: "develop"
- name: LOG_DB_HOST
  value: "jdbc:postgresql://[esb_postgres_host]: [postgres_port]/[esb_postgres_database]"
- name: LOG_DB_LOGIN
  value: "User"
- name: LOG_DB_PASSWORD
  value: "Pswd"
```

apiVersion: v1

kind: Service

metadata:

name: poc-postgres-adapter-expose

namespace: gtm-esb-develop

spec:

selector:

app: poc-postgres-adapter

type: NodePort

ports:

- protocol: TCP

port: 9092

targetPort: 9092

nodePort: 30134

esb-process-application

Тип проекта: серверный SpringBoot-проект.

Описание: Контейнер сервиса, предоставляющего Rest API для управления маршрутами сервисной шины на основе bpmn-описаний бизнес-процессов. Основные функции представлены в списке ниже:

- исполнение маршрутов интеграции (передача данных от ресурса-источника до ресурса-приемника с различными вариантами обогащения/преобразования исходных данных);
- гарантированная доставка (при критической ошибке "участника" маршрута, данные сохраняются в очереди, и, после восстановления, маршрут продолжит выполнение с этапа прерывания);
- трансформация данных с использованием скриптового языка Datasounet.

Требования: Наличие доступа до модуля конфигураций Config-server и до брокера сообщений Active MQ.

Листинг А.33 – esb-process-application.yaml

```
---
apiVersion: v1
kind: Service
metadata:
  name: poc-process-application
  namespace: gtm-esb-develop
spec:
  selector:
    app: poc-process-application
  ports:
    - name: http
      protocol: TCP
      port: 8092
      targetPort: http

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: poc-process-application
  namespace: gtm-esb-develop
  labels:
    app: poc-process-application
spec:
```

```
replicas: 1
selector:
  matchLabels:
    app: poc-process-application
strategy:
  rollingUpdate:
    maxSurge: 0
    maxUnavailable: 1
  type: RollingUpdate
template:
  metadata:
    labels:
      app: poc-process-application
  spec:
    containers:
      - name: poc-process-application
        image: harbor.stageogip.ru/gtm/poc-process-application:develop
        imagePullPolicy: Always
        ports:
          - name: http
            containerPort: 8092
        env:
          - name: ENVIRONMENT
            value: "develop"
          - name: CONFIG_SERVER_URL
            value: "http://config-server:9000/api/config/develop/resource/PROCESS_APP/conf"
          - name: LOG4J2_CONFIG
            value: "http://config-server:9000/api/config/develop/resource/PROCESS_APP/log4j2.xml"
          - name: GRAYLOG_HOST
            value: "[GRAYLOG_HOST]"
          - name: GRAYLOG_PORT
            value: "12201"
          - name: SPRING_PROFILES_ACTIVE
            value: "develop"
          - name: LOG_DB_HOST
            value: "jdbc:postgresql://[esb_postgres_host]: [postgres_port]/[esb_postgres_database]"
          - name: LOG_DB_LOGIN
            value: "User"
          - name: LOG_DB_PASSWORD
```

value: "Pswd"

- name: CONFIG_SERVER_CLEAR_URL

value: "http://config-server:9000/api/config/develop"

apiVersion: v1

kind: Service

metadata:

name: poc-process-application-expose

namespace: gtm-esb-develop

spec:

selector:

app: poc-process-application

type: NodePort

ports:

- protocol: TCP

port: 8092

targetPort: 8092

nodePort: 30135

Приложение В Перечень таблиц, создаваемых при запуске gtm-back

Таблица В.1 — Перечень таблиц, создаваемых при запуске gtm-back

Наименование схемы	Наименование таблицы
gtm_web	audit
gtm_web	audit_detail
gtm_web	endpoints
gtm_web	mutually_ex_roles
gtm_web	right_types
gtm_web	rights
gtm_web	role_right
gtm_web	roles
public	databasechangelog
public	databasechangeloglock

Приложение С Перечень таблиц, создаваемых при запуске esb-config-server

Таблица С.1 — Перечень таблиц, создаваемых при запуске config-server

Наименование схемы	Наименование таблицы
esb_repos	application_log
esb_repos	dictionary
esb_repos	email
esb_repos	endpoint
esb_repos	job
esb_repos	mapping_result
esb_repos	resource
esb_repos	route
esb_repos	script
esb_repos	transfer_log
esb_repos	transfer_log_info
esb_repos	vendpoint
esb_repos	vroutes
public	databasechangelog
public	databasechangelock