

# Система генерации ключей Key Manager


## Руководство по установке

Индекс	KeyManager-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

1. Аннотация .....	3
2. Введение .....	4
2.1. Требования к квалификации установщика .....	4
2.2. Системные требования .....	4
2.3. Установка и настройка PostgreSQL .....	4
2.3.1. Настройка PostgreSQL .....	7
3. Установка и настройка системы .....	9
3.1. Процедура установки .....	9
3.2. Как создать новую среду .....	9
3.2.1. Пример .gitlab-ci.yml .....	9
3.2.2. Двухступенчатый деплой .....	9
3.3. Настройка CD для продукта, опубликованного в Releases .....	10
3.4. Настройка и развертывание KeyManager .....	10
3.4.1. Настройка переменных окружения .....	10
3.4.2. Список необходимых переменных окружения для развертывания .....	10
3.4.3. Настройка additional .....	11
3.4.4. Состав репозитория .....	11
3.4.5. Выбор компонентов KeyManager для установки .....	11
3.4.6. Параметры конфигурации .....	11
3.4.6.1. Описание параметров ags_web .....	11
3.4.6.2. Описание параметров db .....	14
3.4.6.3. Описание параметров errormapper .....	15
3.4.6.4. Описание параметров km .....	17
3.4.6.5. Описание параметров km_web .....	19
3.4.7. Динамические параметры в конфигурационных файлах .....	20
3.4.8. Поддержка Canary .....	20
3.4.9. Развертывание km сервиса на нодах с версией Ubuntu 20.04 .....	21
3.4.10. Загрузка лестницы ключей .....	21

## 1. Аннотация

Данный документ содержит руководство по установке и первоначальной настройке системы генерации ключей Key Manager (далее - Система или KeyManager), а также описание системных требований для компонентов. Документ предназначен для сотрудников отдела мониторинга и инсталляции, а также для других технических специалистов, в обязанности которых входит установка и первоначальная настройка системы Key Manager.

 Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиториев ООО "ПЦТ", доступ к которым предоставляется заказчику по запросу.

## 2. Введение

### 2.1. Требования к квалификации установщика

Для установки системы сотрудник обязан:

- иметь базовые представления и практические навыки работы с системой оркестрации Kubernetes (<https://kubernetes.io/docs/tutorials/kubernetes-basics/>) и пакетным менеджером Helm.
- иметь навыки работы с ОС семейства Linux, а именно:
  - установка пакетов;
  - создание и настройка сетевых подключений;
  - запуск служб, настройка автозапуска служб;
  - установка и настройка PostgreSQL;
  - создание и работа с БД под управлением PostgreSQL.
- иметь знания о DNS.
- иметь базовые представления и практические навыки работы с Git.

### 2.2. Системные требования

**Примечание.** Система KeyManager в настоящий момент может работать под ОС Debian 11. Для установки необходимо предварительно выполнить следующие требования:

- Установлен и настроен кластер Kubernetes.
  - Так как развертывание производится в кластере k8s, то необходим config file для доступа к кластеру.
    1. Если пользователь выполнял развертывание Kubernetes самостоятельно, то он сам должен создать config file (см. документацию Kubernetes).
    2. Если Kubernetes был развернут сторонними людьми, то необходимо получить config file у администратора кластера.
- Установлен kubectl (<https://kubernetes.io/docs/tasks/tools/install-kubectl/>).
- Установлен helm.
- Развернут DNS-сервер, преобразование имен dns зоны настроено на мастера k8s (созданы А записи на зону dns).
- Для корректной работы системы KeyManager требуется поднять несколько Redis баз данных;
- Для корректной работы системы KeyManager требуется развернуть кластер высокой доступности PostgreSQL (High-Availability Cluster) (ссылка предоставляется по запросу заказчика).
- Для корректной работы системы KeyManager необходим доступ к следующим ресурсам:
  - chartmuseum (ссылка предоставляется по запросу заказчика);
  - gitlab (ссылка предоставляется по запросу заказчика).
- Необходим доступ к репозиторию продукта (ссылка предоставляется по запросу заказчика), содержащему helmfile для развертывания KeyManager.

### 2.3. Установка и настройка PostgreSQL

 Для работы системы DRM требуется PostgreSQL версии 12 или выше.

Ниже приведен пример установки PostgreSQL на сервер без развертывания и настройки кластера БД.

1. (Рекомендуется) обновить текущие системные пакеты, если это новый экземпляр сервера:

```
sudo apt update
sudo apt -y install vim bash-completion wget
sudo apt -y upgrade
```

Установите дополнительные пакеты (локаль):

```
locale -a
sudo locale-gen ru_RU.UTF-8
sudo dpkg-reconfigure locales
```

Выполните перезагрузку:

```
sudo reboot
```

2. Добавьте репозиторий PostgreSQL 12:

- a. Перед настройкой репозитория APT импортируйте ключ GPG, используемый для подписи пакетов:

```
sudo apt update
sudo apt -y install gnupg2
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

- b. После импорта ключа GPG добавьте содержимое репозитория в ОС:

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" |sudo tee /etc
/apt/sources.list.d/pgdg.list
```

- c. Добавленный репозиторий содержит много различных пакетов, включая сторонние дополнения. Они включают:

- i. PostgreSQL-клиент
- ii. PostgreSQL
- iii. libpq-DEV
- iv. PostgreSQL-сервер-DEV
- v. пакеты pgadmin

- d. Cat файл, созданный для проверки его содержимого:

```
$ cat /etc/apt/sources.list.d/pgdg.list
deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main
```

3. Установка пакетов PostgreSQL 14:

- a. Обновите список пакетов и установите серверные и клиентские пакеты PostgreSQL 14:

```
sudo apt update
sudo apt -y install postgresql-14 postgresql-client-14 postgresql-14-cron
```

- b. Запустите сервер базы данных, используя следующую команду:

```
sudo pg_ctlcluster 14 main start
```

c. Подтвердите статус службы и используемый файл конфигурации:

```
$ sudo pg_ctlcluster 14 main status
pg_ctl: server is running (PID: 4209)
/usr/lib/postgresql/14/bin/postgres "-D" "/var/lib/postgresql/14/main" "-c" "config_file=/etc/postgresql/14/main/postgresql.conf"
```

d. Можно использовать команду *systemctl* для проверки статуса службы. В случае успешной установки выводится сообщение примерно следующего вида:

```
$ systemctl status postgresql.service
postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2019-10-06 10:23:46 UTC; 6min ago
   Main PID: 8159 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 2362)
   CGroup: /system.slice/postgresql.service
Oct 06 10:23:46 debian systemd[1]: Starting PostgreSQL RDBMS...
Oct 06 10:23:46 debian systemd[1]: Started PostgreSQL RDBMS.

$ systemctl status [email protected]
[email protected] - PostgreSQL Cluster 14-main
   Loaded: loaded (/lib/systemd/system/[email protected]); indirect; vendor preset: enabled)
   Active: active (running) since Sun 2019-10-06 10:23:49 UTC; 5min ago
   Main PID: 9242 (postgres)
   Tasks: 7 (limit: 2362)
   CGroup: /system.slice/system-postgresql.slice/[email protected]
          9242 /usr/lib/postgresql/14/bin/postgres -D /var/lib/postgresql/14/main -c
          config_file=/etc/postgresql/14/main/postgresql.conf
          9254 postgres: 14/main: checkpointer
          9255 postgres: 14/main: background writer
          9256 postgres: 14/main: walwriter
          9257 postgres: 14/main: autovacuum launcher
          9258 postgres: 14/main: stats collector
          9259 postgres: 14/main: logical replication launcher
Oct 06 10:23:47 debian systemd[1]: Starting PostgreSQL Cluster 14-main...
Oct 06 10:23:49 debian systemd[1]: Started PostgreSQL Cluster 14-main.

$ systemctl is-enabled postgresql
enabled
```

e. Во время установки автоматически создаётся пользователь *postgres*. Это суперадминистратор, который имеет полный доступ ко всему PostgreSQL.

4. Проверка соединения с PostgreSQL, настройка пользователя:

a. Во время установки пользователь *postgres* создается автоматически. Этот пользователь имеет полный доступ *superadmin* ко всему экземпляру PostgreSQL.

```
sudo su - postgres
```

b. (Необязательно) замените пароль пользователя на более надежный:

```
psql -c "alter user postgres with password 'NEW_PASSWORD'"
```

c. Запускаем PostgreSQL с помощью команды:

```
$ psql
```

d. Получить информацию о подключении, как показано ниже:

```
$ psql
psql (14.0 (Ubuntu 14.0-1.pgdg18.04+1))
Type "help" for help.

postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql"
at port "5432".
```

e. Убедиться, что сервис PostgreSQL запускается при загрузке системы, можно с помощью команд:

```
$ systemctl status postgresql.service
$ systemctl status postgresql@14-main.service
$ systemctl is-enabled postgresql
```

### 2.3.1. Настройка PostgreSQL

**i** Данный раздел следует использовать только в случае установки БД в режиме Standalone. Следующие действия выполняются на сервере, где будут развернуты базы данных, только после установки пакета postgresql-14.

Открыть конфигурационный файл postgresql.conf для редактирования:

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Изменить значение параметра listen\_addresses, как показано ниже, и раскомментировать соответствующую строку:

```
listen_addresses = '*'           # what IP address(es) to listen on;
```

Для настройки автовакуума рекомендуются приведенные ниже значения (использовались при тестировании)

```
autovacuum = on
#log_autovacuum_min_duration = 0
autovacuum_max_workers = 10
autovacuum_naptime = 1s
autovacuum_vacuum_threshold = 50
autovacuum_analyze_threshold = 50
autovacuum_vacuum_scale_factor = 0.01
autovacuum_analyze_scale_factor = 0.02
```

Открыть конфигурационный файл pg\_hba.conf для редактирования:

```
sudo nano /etc/postgresql/14/main/pg_hba.conf
```

Необходимо, чтобы к postgres могли подсоединиться любые процессы с локальной машины и компьютеры из локальной сети (например, с ip 192.168.x.x). Также необходимо указать настройки IPv6. Таким образом, файл может выглядеть следующим образом (рекомендуется задавать уровень доступа менее открытый, чем trust):

```
# "local" is for Unix domain socket connections only
local  all          all                                trust
# IPv4 local connections:
host   all          all          127.0.0.1/32      md5
host   all          all          172.17.0.0/16     md5
host   all          all          192.168.0.0/16   md5
# IPv6 local connections:
host   all          all          ::1/128          md5
```

При работе DRM требуются подключения к базам данных, приведенным в таблице ниже. Необходимо настроить к ним доступ:

Название БД	Администратор БД
km	kmadmin

После внесения изменений перезапустить PostgreSQL:

```
sudo /etc/init.d/postgresql restart
```



## 3. Установка и настройка системы

### 3.1. Процедура установки

Необходимо выполнить установку системы KeyManager, как описано ниже.

### 3.2. Как создать новую среду

1. Создать отдельный проект в Gitlab
2. Настроить данный проект как подмодуль на основе инструкции, ссылка на которую предоставляется по запросу заказчика.
3. В проекте среды создать helmfile.yaml с содержимым:

```
---
helmfiles:
- path: < >/helmfile.yaml
  values:
  - < >/default.yaml # -
  - production.yaml #
  - versions.yaml # ()
```

#### 3.2.1. Пример .gitlab-ci.yml

```
# (stage)
# ,
# ,
# (, init)
stages:
- init
- compose
- grade

variables:
# GIT_*
# c
GIT_SUBMODULE_STRATEGY: recursive
GIT_STRATEGY: clone
# namespace values//helmfile,
# NAMESPACE
NAMESPACE: sms-web
# NO_PROXY ip kubeconfig, c 3.4
NO_PROXY: 172.28.16.10, 172.28.16.11, 172.28.16.12

include:
- project: 'automation/cd-templates'
  ref: "4.0"
  file: pipeline.yml
```

#### 3.2.2. Двухступенчатый деплой

Для выполнения двухступенчатого деплоя, в случае если часть релизов, описанных в helmfile, следует установить прежде остальных, следует выполнить три условия:

- задать в файле `.gitlab-ci.yml` переменную `STAGED_PIPELINE` в значение `true`;
- в `helmfile.yaml` задать переменные `wait` и `waitForJobs`;
- указать для каждого релиза этап его установки посредством меток `stage: first` или `stage: second`.

При этом возможно так же установить допустимый период ожидания выполнения установки релизов/джейбов посредством переменной `timeout` (по умолчанию - 300).

Версия шаблонов CI должна быть не менее 4.0.

### 3.3. Настройка CD для продукта, опубликованного в Releases

Процедура описана в документе, ссылка на который предоставляется по запросу заказчика.

### 3.4. Настройка и развертывание KeyManager

#### 3.4.1. Настройка переменных окружения

В системе развертывания KeyManager требуется указывать переменные окружения, которые используются непосредственно в самом процессе деплоя KeyManager в кластер.

Настройка переменных осуществляется в gitlab.

В боковом меню выбрать **Settings** (на панели слева) -> **CI/CD** -> **Environment variables**. Отредактировать переменные.

#### 3.4.2. Список необходимых переменных окружения для развертывания

В системе развертывания KMS (Key Management Server - сервер хранения и управления ключами) требуется указывать переменные окружения которые используются непосредственно в самом процессе деплоя ConfigManager в кластер.

Таблица с описанием используемых переменных Gitlab:

ENV	Описание
BBMK_KEY	BBMK-ключ. Ключ является частью лестницы ключей, применяемой в КМ БД
PRIVATE_KMI_KEY	HWRK-ключ. Ключ является частью лестницы ключей, применяемой в КМ БД
KMDB_LOGIN	Имя пользователя для КМ БД
KMDB_PASSWORD	Пароль пользователя для КМ БД
TDEDB_LOGIN	Имя пользователя для TDE БД
TDEDB_PASSWORD	Пароль пользователя для TDE БД
POSTGRES_LOGIN	Имя пользователя для Postgres БД
POSTGRES_PASSWORD	Пароль пользователя для Postgres БД

**i** **ВАЖНО!** Environment variables имеют более высокий приоритет, чем переменные, заданные в файлах.

**i** Параметры `_LOGIN` и `_PASSWORD` задаются пользователем и используются при подключении к соответствующим базам данных.

**i** Большинство приведенных переменных необходимы для работы KeyManager.

### 3.4.3. Настройка **additional**

Папка **additional** содержит файлы, с помощью которых настраиваются `dns`, `ingress`, `probes`, `statsd`. Указанные параметры применяются ко всем сервисам и службам в данном репозитории. **Рекомендуется не менять эти настройки.**

### 3.4.4. Состав репозитория

Конфигурация продукта внутри репозитория (ссылка предоставляется по запросу заказчика) выглядит следующим образом:

Репозиторий имеет следующий состав:

- `helmfile.yaml` - главный конфигурационный файл утилиты `helmfile`.
- `default.yaml` - файл с `values` окружения утилиты `helmfile`.
- `values` - папка с `values` для каждого чарта; они являются шаблонными и забирают значения из `values` окружения (файла `default.yaml`).
- `versions.yaml` - файл с версиями компонентов; если в версии установлена пустая строка, то берется последняя версия (в соответствии с `semver2`).
- `limitation` - папка с `values` ресурсов подов. С помощью этих файлов настраиваются компоненты системы KeyManager, в том числе многочисленные базы данных.

### 3.4.5. Выбор компонентов KeyManager для установки

По умолчанию разворачиваются все компоненты продукта KeyManager, однако при необходимости можно отключать ненужные: для этого в `production.yaml`, в корне секции соответствующего компонента нужно выставить `enabled: false`.

### 3.4.6. Параметры конфигурации

#### 3.4.6.1. Описание параметров `ags_web`

`ags_web` - "проxy" для UI (в деплое обозначается как `web`): перенаправляет запрос в `km` и получает от него ответ:

- в случае успеха - передает ответ в UI.
- в случае возникновения ошибки - на `ags_web` формируется запрос на Error Mapper Server для переопределения внутреннего кода ошибки на внешний числовой код ошибки.

Параметры конфигурации:

Параметр	Описание
LOGGER_LEVEL	Степень логирования событий. Возможные значения: trace debug info (значение по умолчанию) warning error fatal. Для тестирования рекомендуется trace или debug.
ERRORS_API_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_API_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_API_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_METHOD_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_INTERNAL_CODE	Код ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_TITLE	Заголовок ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_DETAIL	Содержимое блока с ошибкой для обобщенной внутренней ошибки ("internal error").
VERIFICATION_MDS	Аккаунт Менеджер MDS, к которому обращается сервер для определения прав на запрос.

VERIFICATION_SERVICE_NAME	Текущее значение - "smsaccess". Под этим условным обозначением MDS распознает SMS-сервис и ассоциирует с ним набор разрешений, по которым работают роли на MDS.
VERIFICATION_PROTOCOL	Протокол взаимодействия с Аккаунт Менеджером MDS. Значение по умолчанию - http.
ERROR_MAPPER_HOST	Адрес сервиса Error Mapper.
ERROR_MAPPER_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение с error mapper server. Иными словами, Ожидание ответа от error mapper server.
ROUTING_HOST	Адрес сервиса routing_server (без указания протокола), который может иметь следующие значения: IP-адрес:порт либо servicename (в K8s).
ROUTING_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение с routing_server. Иными словами, Ожидание ответа от routing_server.
ROUTING_PROTOCOL	Протокол, по которому осуществляется взаимодействие с routing_server. Значение по умолчанию: http
ROUTING_ENABLED	Флаг получения данных о роутинге: (по умолчанию выключено для кей менеджера): true: данные берутся из Routing DB. Для ags_web для корректной работы с правами нужно выставить в true. false:данные берутся из routing.yml.
ROUTING_HEALTH_CHECK	Флаг проверки доступности routing_server при старте.
ROUTING_SERVICE_NAME	Адрес routing_server, по которому ходит AGS (Api Gateway Server) за получением роутов.
SYSTEM_ROUTING_FILE_PATH	Путь к файлу routing.yml . Файл содержит настройки для динамической конфигурации обработки входящих запросов. В соответствии с настройками, заданными в routing.yml, осуществляется перенаправление запроса в требуемый микросервис.
SYSTEM_TRACER_ENABLED	Флаг включения трассировки. Возможные значения: false - трассировка выключена. true - трассировка включена.
SYSTEM_TRACER_AGENT_HOST_PORT	Адрес и номер порта agent, т.е. сервера, на который идет отправка параметров трассировки. Пример значения: agent: 6831
SYSTEM_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>

SYSTEM_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).
SYSTEM_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_HTTP_IS_USED	Флаг включения взаимодействия по http. Возможные значения: false - Взаимодействие выключено. true - Сервис взаимодействует по http.
SYSTEM_PROMETHEUS_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>
SYSTEM_PROMETHEUS_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).
SYSTEM_PROMETHEUS_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_PPROF_ENABLE	Флаг включения взаимодействия с rprof. Возможные значения: false - rprof выключен. Значение по умолчанию. true - rprof включен.
SYSTEM_PPROF_ACCESS_KEY	Значение ключа доступа, отправляемого в запросе к rprof. Запрос используется для профилирования сервисов (исследования CPU).

#### 3.4.6.2. Описание параметров db

Параметры, которые указываются в production yaml репозитория для установки всех бд, необходимых для работы системы:

Параметр	Описание
pg_db	Общая секция для подключения к машине, где будут устанавливаться базы данных
pg_db.address	указание адреса машины
pg_db.master_port	указание порта для мастер реплики бд
pg_db.sync_port	указание порта для синхронной реплики бд
pg_db.async_port	указание порта для асинхронной реплики бд
km_db_sch	Секция для указания параметров установки схемы бд кей менеджера
km_db_sch.enabled	флаг, который включает и выключает установку компоненты
km_db_sch. km_db_name	наименования базы данных (можно указать свое значение, но если его не указать, то бд будет называться kmdb)

km_db_api	Секция для установки апи базы данных кей менеджера
km_db_api.enabled	флаг, который включает и выключает установку компоненты
tde_db_sch	Секция для установки схемы базы данных tde
tde_db_sch.enabled	флаг, который включает и выключает установку компоненты
tde_db_sch. tde_db_name	наименования базы данных (можно указать свое значение, но если его не указать, то бд будет называться tde)
tde_db_api	Секция для установки апи базы данных tde
tde_db_api.enabled	флаг, который включает и выключает установку компоненты
ems_db_sch	Секция для установки схемы базы данных error mapper
ems_db_sch.enabled	флаг, который включает и выключает установку компоненты
ems_db_sch. errmap_db_name	наименования базы данных (можно указать свое значение, но если его не указать, то бд будет называться km_errmap)
ems_db_api	Секция для установки апи базы данных error mapper
ems_db_api.enabled	флаг, который включает и выключает установку компоненты
km_error_maps_i18n	Секция для начального наполнения бд error mapper
km_error_maps_i18n. enabled	флаг, который включает и выключает установку компоненты
km_error_maps_i18n. table_trunc	флаг, который регулирует очищение таблицы перед установкой начального наполнения

### 3.4.6.3. Описание параметров errormapper

Настройка EMS (Error Mapper)

Параметр	Описание
LOGGER_LEVEL	Степень логирования событий. Возможные значения: 0 - trace 1 - debug 2 - info (значение по умолчанию) 3 - warning 4 - error 5 - fatal. Для тестирования рекомендуется trace или debug.
ERRORS_API_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).

ERRORS_API_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_API_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_METHOD_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_INTERNAL_CODE	Код ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_TITLE	Заголовок ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_DETAIL	Содержимое блока с ошибкой для обобщенной внутренней ошибки ("internal error").
SYSTEM_TRACER_ENABLED	Флаг включения трассировки. Возможные значения: false - трассировка выключена. true - трассировка включена.
SYSTEM_TRACER_AGENT_HOST_PORT	Адрес и номер порта agent, т.е. сервера, на который идет отправка параметров трассировки. Пример значения: agent: 6831
SYSTEM_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>
SYSTEM_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).
SYSTEM_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_PROMETHEUS_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>



SYSTEM_PROMETHEUS_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).
SYSTEM_PROMETHEUS_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_DATABASE_HOST	IP-адрес Errmap DB
SYSTEM_DATABASE_PORT	Номер порта для подключения к Errmap DB
SYSTEM_DATABASE_DB_NAME	Имя Errmap DB в PostgreSQL
SYSTEM_DATABASE_USER	Логин пользователя, под которым осуществляется подключение к Errmap DB
SYSTEM_DATABASE_PASSWORD	Пароль для подключения к Errmap DB
SYSTEM_PPROF_ENABLE	Флаг включения взаимодействия с rprof. Возможные значения: false - rprof выключен. Значение по умолчанию. true - rprof включен.
SYSTEM_PPROF_ACCESS_KEY	Значение ключа доступа, отправляемого в запросе к rprof. Запрос используется для профилирования сервисов (исследования CPU).

#### 3.4.6.4. Описание параметров km

Параметр	Описание
LOGGER_LEVEL	Степень логирования событий. Возможные значения: trace debug info (значение по умолчанию) warning error fatal Для тестирования рекомендуется trace или debug.
ERRORS_API_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_API_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).
ERRORS_API_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "api not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер неизвестный URL request).

ERRORS_METHOD_NOT_ALLOWED_CODE	Код ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_TITLE	Заголовок ошибки для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_METHOD_NOT_ALLOWED_DETAIL	Содержимое блока с ошибкой для сценария ошибки "method not allowed" (ошибки, отправляемой в случае, если клиент пытается отправить на сервер верный URL request, но при этом использует неподдерживаемый HTTP method).
ERRORS_INTERNAL_CODE	Код ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_TITLE	Заголовок ошибки для обобщенной внутренней ошибки ("internal error").
ERRORS_INTERNAL_DETAIL	Содержимое блока с ошибкой для обобщенной внутренней ошибки ("internal error").
SYSTEM_DB_XXX_HOST	IP-адрес базы XXX.
SYSTEM_DB_XXX_PORT	Номер порта базы XXX.
SYSTEM_DB_XXX_USER	Логин пользователя, под которым осуществляется подключение к базе XXX.
SYSTEM_DB_XXX_PASSWORD	Пароль для подключения к базе XXX.
SYSTEM_DB_XXX_DB_NAME	Имя базы XXX в PostgreSQL.
SYSTEM_DB_XXX_SCHEME	Название схемы в базе XXX, из которой берутся ключи.
SYSTEM_DB_XXX_TIMEOUT	Время ожидания ответа (в сек.) от базы XXX.
SYSTEM_DB_XXX_MAX_CONNS	Максимальное количество соединений, создаваемых сервером с базой XXX.
SYSTEM_GRPC_PORT	Номер порта для работы с gRPC (системой удаленного вызова процедур). Примечание. С gRPC работают на одном хосту, поэтому другие параметры не требуются.
SYSTEM_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>
SYSTEM_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).

SYSTEM_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_HTTP_IS_USED	Флаг включения взаимодействия по http. Возможные значения: false - Взаимодействие выключено. true - Сервис взаимодействует по http.
SYSTEM_KEY_ROTATOR_DB_REQUEST_INTERVAL	Время через которое необходимо выполнить ротацию ключей
SYSTEM_KEY_ROTATOR_DB_REQUEST_LIMIT	Максимальное число запросов на ротацию
SYSTEM_KEY_ROTATOR_WORKERS	Количество параллельных потоков
SYSTEM_PROMETHEUS_HTTP	Параметры взаимодействия по prometheus_http:
SYSTEM_PROMETHEUS_HTTP_ADDRESS	Адрес для запуска сервера в контейнере. Формат записи: <IP-адрес>:<номер порта>
SYSTEM_PROMETHEUS_HTTP_WRITE_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на запись данных).
SYSTEM_PROMETHEUS_HTTP_READ_TIMEOUT	Таймаут (в сек.), по которому будет разорвано соединение (при обращении на чтение данных).
SYSTEM_PPROF_ENABLE	Флаг включения взаимодействия с rprof. Возможные значения: false - rprof выключен. Значение по умолчанию. true - rprof включен.
SYSTEM_PPROF_ACCESS_KEY	Значение ключа доступа, отправляемого в запросе к rprof. Запрос используется для профилирования сервисов (исследования CPU).
SYSTEM_TDE_DBMK_FILE_NAME	Полный путь и название ВВМК-ключа, шифрованного public HWRK-ключом. Файл с ключом экспортируется из системы КМІ. Примечание. Ключ является частью лестницы ключей, применяемой при шифровании данных в TDE DB.
SYSTEM_TDE_DSN	DSN базы данных (TDE DB), которую использует tde (механизм шифрования данных по лестнице ключей).
SYSTEM_TDE_HWRK_FILE_NAME	Полный путь и название private HWRK-ключа, сгенерированного на сервере. Примечание. Ключ является частью лестницы ключей, применяемой при шифровании данных в TDE DB.
SYSTEM_TDE_USE_MLOCK	Флаг. Если установлен в true, то блокируется возможность сброса на диск данных, хранящихся в оперативной памяти в модуле по работе с tde.


### 3.4.6.5. Описание параметров km\_web

Параметр	Описание
API_HOST	Адрес, на который далее будут отправлены запросы с web-интерфейса. Указывается адрес (с указанием протокола) сервиса ags_web. (Например " <a href="http://ags.testdrp.tz.cas">http://ags.testdrp.tz.cas</a> ")
SSO_HOST	Ссылка на сервис SSO авторизации. Примечание. Использовать значение 'http://'
PRODUCT_CODE	Код продукта для общего экрана авторизации. Примечание. Использовать значение "KeyManager"
SSO_ACTIVE	Флаг для регулирования использования SSO авторизации. Если стоит значение false, то она не используется, а работа происходит при помощи локальной авторизации. Примечание. Использовать значение "false"
TOKEN_NOTIFY_TIME	Параметр отвечает за время, когда начнут выводиться уведомления о скором истечении времени жизни токена.

### 3.4.7. Динамические параметры в конфигурационных файлах

В конфигурационных файлах key\_manager.cfg.dft (ссылка предоставляется по запросу заказчика) параметры разделены на две группы:

1. Все параметры, лежащие вне секции "system". Эти параметры можно менять динамически, т.е. без перезапуска соответствующей службы. При изменении значений этих параметров в конфигурационном файле, по прошествии некоторого времени, новые значения будут автоматически применены к службе.

 **Обратите внимание!** Параметры, изменяемые динамически, нельзя задать через переменные окружения (см. [выше](#)), они меняются только в конфигурационном файле.

2. Некоторые из динамически изменяемых параметров нельзя применять со значениями "по умолчанию", они должны быть настроены на production.

Параметры в секции "system". Эти параметры нельзя изменить динамически: чтобы изменения этих параметров вступили в силу, соответствующая служба должна быть перезапущена.

### 3.4.8. Поддержка Canary

Canary-релиз - это стратегия развертывания, в рамках которой изменения сначала выпускаются для небольшой группы пользователей. Далее за системой тщательно следят, выявляя признаки проблем. При этом используются как KPI, так и операционные метрики.

Теперь можно поднять параллельно два стенда: один основной, другой - канареечный, - и часть трафика будет уходить на канареечный стенд.

Для работы данной функциональности необходимо, чтобы у сервисов fas-entry обоих стендов были **одинаковые ингерессы**.

Есть три варианта работы:

1. *byWeight* - по весу. На канареечный стенд будет уходить определенный процент от общего количества запросов.
2. *byHeader* - по хедеру. Если в запросе присутствует определенный header с определенным значением - такой запрос уходит на канареечный стенд.
3. *bySubnetIp* - по ip. На канареечный стенд уходят все запросы с определенного диапазона ip адресов. Ip определяется по header'y *X-Forwarded-For*.

Соответствующие настройки задаются в следующих файлах:

1. `default.yaml`:

```
canary:
enabled: false
byHeader:
  enabled: false
  headerKey: key
  headerValue: value
bySubnetIp:
  enabled: false
  bySubnetIpHeader: X-Forwarded-For
  subnetIp: "102.222.11.0 192.168.0.0"
byWeight:
  enabled: true
  weight: 30
```

### 3.4.9. Развертывание km сервиса на нодах с версией Ubuntu 20.04

При развертывании сервиса km происходит инициализация TDE библиотеки, которая требует повышенного количества памяти. Для успешного развертывания на машине с Ubuntu 20.04 необходимо задать в файле `etc /security/limits.conf` следующие настройки:

```
*          soft  nproc      65000
*          hard  nproc      1000000
*          -    nofile     1048576
root      -    memlock    unlimited
```

Так же необходимо задать `LimitMEMLOCK=infinity` в `k3s.service`.



**Обратите внимание!** Для применения настроек необходимо перезагрузить машину.

### 3.4.10. Загрузка лестницы ключей

Для функционирования системы необходимо выполнить загрузку лестницы ключей с помощью скрипта `tde_load_keys.sh` (ссылка предоставляется по запросу заказчика).

Формат команды запуска скрипта с параметрами:

```
bash tde_load_keys.sh [-help] -f input_file [-h host] [-p port] [-pass pg_user_password] [-d database] [-u pg_user]
```

Параметры запуска:

1. `-help` - вызов справки.

2. -f input\_file - путь к файлу с загружаемыми данными (файл с конфигурацией для part type / лестницей ключей). Обязательный параметр.
3. -h host - имя хоста TDE DB. Значение по умолчанию (задается в файле скрипта) - "localhost".
4. -p port - номер порта TDE DB. Значение по умолчанию (задается в файле скрипта) - "5432".
5. -pass pg\_user\_password - пароль для доступа к TDE DB. Значение по умолчанию (задается в файле скрипта) - "tdeadadmin".
6. -d database - имя базы TDE DB. Значение по умолчанию (задается в файле скрипта) - "tde".
7. -u pg\_user - имя пользователя TDE DB. Значение по умолчанию (задается в файле скрипта) - "tdeadadmin".

© ООО "ПЦТ", 2024

Документация "Система генерации ключей Key Manager. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя