GsCtk

Выпуск 4.5.0

сент. 08, 2026

Содержание

1 Введение **3**
- 1.1 Предисловие . . . 3
- 1.2 О системе . . . 3

2 Преимущества решения **4**
- 2.1 Высокая скорость настройки . . . 4
- 2.2 Гибкость настройки . . . 4
- 2.3 Надежность . . . 4
- 2.4 Управляемое копирование . . . 4
- 2.5 Управляемое обновление . . . 4
- 2.6 Работа в изолированной среде . . . 4
- 2.7 Интеграция с внешним CI . . . 5

3 Архитектура решения **5**
- 3.1 Дистрибутив . . . 6
- 3.2 Режим ссылок на образы (Helm) . . . 6
- 3.3 База данных . . . 7
- 3.4 Хранение файлов . . . 7
- 3.5 Алгоритм развертывания . . . 7

4 Группы и книги ресурсов **7**
- 4.1 Классы книг ресурсов . . . 8
- 4.2 Конфигурационные значения (values) группы ресурсов . . . 8

5 Комплект приложения. Комплект группы **8**
- 5.1 Комплект приложений (Appkit) . . . 8
- 5.2 Комплект группы (Groupkit) . . . 10
- 5.3 Управление комплектами . . . 11

6 Классы книг ресурсов **13**
- 6.1 global_server_excl . . . 13
- 6.2 global_server_share . . . 14
- 6.3 global_server_scheduler . . . 15
- 6.4 haproxy . . . 16
- 6.5 grafana . . . 16
- 6.6 rabbitmq . . . 17
- 6.7 gossiprouter . . . 18
- 6.8 jgroups_dns . . . 19

6.9 ctk_restAPI . . . 19
6.10 queuewalker . . . 22

7 Обработка нештатных ситуаций **23**
7.1 Отладка конфигурации служб в контейнерах . . . 23
7.2 Диагностика запуска контейнера . . . 24

8 RabbitMQ **24**
8.1 Настройка кластера . . . 24

9 JGroups **25**

10 ctk_restapi **25**
10.1 Основные функции и назначение . . . 25

11 Приложения **26**
11.1 Nscli . . . 26
11.2 Nsctl . . . 43

12 Справка по параметрам конфигурации **63**
12.1 Параметры группы ресурсов . . . 63
12.2 Параметры книг ресурсов . . . 74
12.3 Вспомогательные структуры . . . 121
12.4 Пример полной конфигурации . . . 136

13 Примечания к версиям **141**
13.1 6.2.3 - 2026-08-21 . . . 141
13.2 6.2.2 - 2026-08-05 . . . 141
13.3 6.2.1 - 2026-07-17 . . . 142
13.4 6.2.0 - 2026-06-18 . . . 142
13.5 6.1.1 - 2026-06-11 . . . 143
13.6 6.1.0 - 2026-05-20 . . . 143
13.7 6.0.8 - 2026-07-17 . . . 143
13.8 6.0.7 - 2026-07-03 . . . 144
13.9 6.0.6 - 2026-06-09 . . . 144
13.106.0.5 - 2026-05-15 . . . 144
13.116.0.4 - 2026-05-12 . . . 144
13.126.0.3 - 2026-05-07 . . . 145
13.136.0.2 - 2026-04-29 . . . 145
13.146.0.1 - 2026-04-28 . . . 145
13.156.0.0 - 2026-04-21 . . . 146
13.165.9.4 - 2026-05-04 . . . 146
13.175.9.3 - 2026-04-20 . . . 147
13.185.9.2 - 2026-04-20 . . . 147
13.195.9.1 - 2026-04-02 . . . 147
13.205.9.0 - 2026-03-30 . . . 147
13.215.8.2 - 2026-03-17 . . . 147
13.225.8.1 - 2026-03-10 . . . 148
13.235.8.0 - 2026-02-27 . . . 148
13.245.7.0 - 2026-01-28 . . . 149
13.255.6.0 - 2026-12-29 . . . 151
13.265.5.1 - 2026-12-29 . . . 151
13.275.5.0 - 2026-12-23 . . . 151
13.285.4.1 - 2026-12-18 . . . 151
13.295.4.0 - 2026-12-15 . . . 152

13.305.3.1 - 2026-12-01 . . . 152
13.315.3.0 - 2026-11-10 . . . 152
13.325.2.5 - 2026-10-31 . . . 153
13.335.2.4 - 2026-10-29 . . . 153
13.345.2.3 - 2026-10-28 . . . 153
13.355.2.2 - 2026-10-24 . . . 153
13.365.2.1 - 2026-10-10 . . . 153
13.375.2.0 - 2026-10-08 . . . 154
13.385.1.6 - 2026-09-24 . . . 154
13.395.1.5 - 2026-09-19 . . . 154
13.405.1.4 - 2026-09-12 . . . 154
13.415.1.3 - 2026-09-01 . . . 155
13.425.1.2 - 2026-08-28 . . . 155
13.435.1.1 - 2026-08-26 . . . 155
13.445.1.0 - 2026-08-18 . . . 155
13.455.0.1 - 2026-08-15 . . . 155
13.465.0.0 - 2026-07-22 . . . 156

1 Введение

1.1 Предисловие

gs-ctk (Global System Cluster Tool Kit) - это комплект для быстрого развертывания отказоустойчивой системы Global ERP в кластере Kubernetes.

Данное руководство описывает внутреннее устройство gs-ctk и предназначено для тех, кто хочет глубже понимать принципы работы комплекта. Оно не рассматривает вопрос развертывания кластера и его эксплуатации, об этом подробнее рассказано в Руководстве системного администратора.

Предполагается, что читатель уже знаком со следующими инструментами:

- Linux
- Docker
- Kubernetes
- Python
- Jinja

1.2 О системе

Давайте представим, что администратору поручили развернуть Global ERP в режиме высокой доступности. Это означает, что ему потребуется не только поднять собственно сервер приложений, но и:

- исследовать принципы работы сервера приложений в кластерном режиме;
- настроить вывод и сбор логов;
- развернуть вспомогательные службы, такие как RabbitMQ;
- создать механизм обновления.

Все это подразумевает изучение и применение специфичных для Global ERP подходов к развертыванию, что может привести к ошибкам и тратам времени.

gs-ctk позволяет упростить этот процесс, предоставляя все необходимы шаблоны и инструменты для развертывания Global ERP на эластичной среде Kubernetes.

2 Преимущества решения

2.1 Высокая скорость настройки

Применение gs-ctk позволяет на несколько порядков увеличить скорость настройки за счет использования готовых шаблонов и отлаженных практик. Удобный командный интерфейс позволяет существенно снизить порог вхождения, необходимый для администрирования Global ERP в кластере Kubernetes

2.2 Гибкость настройки

Комплект позволяет создавать гибкие конфигурации для формирования ландшафтов с контурами для разработки, настройки и отладки, тестирования и промышленной эксплуатации без необходимости размножать репозитории пакетов.

2.3 Надежность

Наличие готовых отлаженных образов контейнеров и шаблонов развертывания позволяет гарантированно получить высокую доступность.

2.4 Управляемое копирование

Одну и ту же конфигурацию можно развернуть в разных рабочих пространствах, получив зеркальные копии, что позволяет минимизировать затраты при поддержке дополнительного плеча высокой доступности.

2.5 Управляемое обновление

Разделение конфигурации и скриптов развертывания позволяет автоматизировать тестирование. А также гарантировать стабильное обновление кластера.

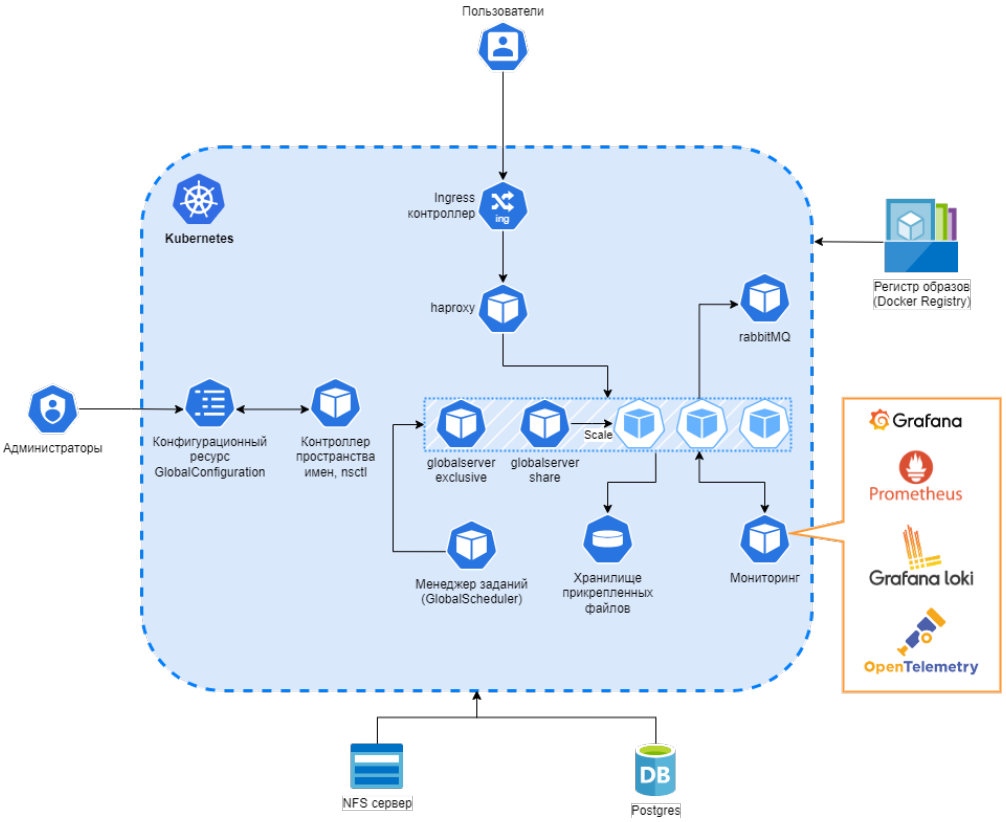
2.6 Работа в изолированной среде

`Gs-ctk` спроектирован для работы в изолированной среде, поэтому набор действий для переноса требуемых артефактов минимален. При этом общая система версий обеспечивает согласованное обновление конфигурации и контейнеров.

2.7 Интеграция с внешним CI

В кластере можно легко развернуть решения собранные на произвольном CI.

3 Архитектура решения



gs-ctk разворачивается в отдельном пространстве имен Kubernetes при помощи Helm-чарта и вспомогательной утилиты **nscli** с **jump-хоста** и состоит из пяти компонентов:

1. Контроллера пространства имен **nsctl**
2. Подов с серверами приложений: эксклюзивного (**global-server-excl**) для задач обслуживания, и реплицируемых (**global-server-share**) для масштабирования и отказоустойчивости
3. Балансировщика нагрузки **HAProxy**
4. Стек мониторинга с доступом через систему визуализации данных **Grafana**
5. Вспомогательных опциональных служб, таких как ***RabbitMQ*** и ***JGroups GossipRouter***

Данные хранятся в *базе данных* и на *файловом хранилище*. Конфигурация gs-ctk хранится в виде ресурсов Kubernetes.

3.1 Дистрибутив

Дистрибутив *gs-ctk* представляет собой:

- набор образов контейнеров:
  - контроллер пространства имен *gs/nsctl*
  - контейнер-болванка со средой для сервера приложений, планировщика задач и GossipRouter *gs/globalserver*
  - вспомогательный (sidecar) контейнер для сбора телеметрии с сервера приложений *gs/systemagent*
  - балансировщик нагрузки *gs/haproxy*
  - вспомогательный (sidecar) контейнер для сбора телеметрии с балансировщика нагрузки *gs/haproxyagent*
  - стек внутреннего мониторинга *gs/grafana*
  - встроенный брокер сообщений *gs/rabbitmq*
- консольную утилиту для работы с кластером с jump-хоста - *nscli*

Контейнеры и утилиту можно скачать с наших репозиториев, однако доступ к ним ограничен. Узнавайте подробнее у контактного лица.

3.2 Режим ссылок на образы (Helm)

При установке Helm-чарта способ формирования ссылок на образы задается в `values.yaml` параметром `imageMode`:

- `tag` - совместимый режим, образа указываются по тегу (`:<tag>`).
- `digestAuto` - digest рабочих образов берутся из lock-файла, встроенного в образ `nsctl` при release-сборке.
- `digestManual` - digest указываются вручную через `imageDigests`.

Параметр `imageDigests` задается списком, например:

```
imageMode: digestManual  # варианты: tag | digestAuto | digestManual
imageDigests:
  - image: nsctl
    digest: sha256:...
  - image: globalserver
    digest: sha256:...
```

В режиме `digestAuto` значение для `image=nsctl` можно задать в `imageDigests` опционально.

Для подготовки такого релиза используется отдельная команда:

```
DEBIAN_MIRROR=https://mirror.example/debian \
DEBIAN_SECURITY_MIRROR=https://mirror.example/debian-security \
./builder/image.sh release --push_to <registry>
```

Она собирает рабочие образы, публикует их, сохраняет digest в lock-файл и только после этого собирает `nsctl` со встроенным списком digest.

3.3 База данных

gs-ctk не разворачивает систему управления баз данных. Вам необходимо самостоятельно развернуть СУБД PostgreSQL с соответствующими требованиями к отказоусточивости. Если вы ищете инструмент автоматического развертывания, посмотрите Patroni.

Рекомендуется использовать систему синхронного или асинхронного реплицирования с автоматическим переключением главной ноды.

Примечание

Синхронная репликация требуется в случае необходимости гарантировать отсутствие потерь выполненных транзакций при сбоях

3.4 Хранение файлов

gs-ctk требует для своей работы несколько файловых хранилищ:

- *системное*, для хранения комплектов приложений и комплектов групп. Указывается для всего gs-ctk при установке.
- *прикладное*, для хранения прикрепленных файлов. Указывается *для группы ресурсов*.

Примечание

Помимо указанных томов, можно примонтировать и другие (пока только NFS). Команды для управления точками монтирования смотрите в справке по *nsctl/resgroup*.

На данный момент поддерживаются только NFS-хранилища.

3.5 Алгоритм развертывания

Совет

Подробнее о развертывании читайте в Руководстве системного администратора.

4 Группы и книги ресурсов

Ключевое понятие gs-ctk - **группа ресурсов**. Группа ресурсов - это единый логический кластер серверов приложений с едиными БД, прикладным решением и вспомогательными службами.

Группа ресурсов в свою очередь делятся на **книги ресурсов**. Книга ресурсов представляет собой набор ресурсов Kubernetes, формируемых по одному шаблону, и реализующих, обычно, одну службу (функциональность).

Мы можем провести параллель между группами ресурсов в gs-ctk и пространствами имен в Kubernetes.

Книги и группы ресурсов хранятся в конфигурационном ресурсе типа GlobalConfiguration, который устанавливается автоматически вместе с Helm-чартом.

4.1 Классы книг ресурсов

Каждая книга ресурсов принадлежит к одному из **классов** книг ресурсов, определяющий ее шаблон и возможные параметры. Например, класс книг ресурсов RabbitMQ. Подробнее о классах книг ресурсов *читайте в отдельном разделе*.

Классы книг ресурсов лежат в директории `resource_library/resbook` в nscli и nsctl.

Класс книги ресурсов состоит из:

- Jinja-шаблона `deploy.yaml`
- модели данных (настроек) в `data.py` для включения в общую модель конфигурационного файла
- представления книги ресурсов в `view.py`
- опционально, скрипта в `migrate.py`, мигрирующего конфигурацию версии 4..

Книга ресурсов автоматически разворачивается, если она и ее группа включены (`enabled`), и автоматически сворачивается, если выключены.

Развертывание заключается в заполнении Jinja-шаблона `deploy.yaml` параметрами из группы и книги ресурсов, хранящимися в ConfigMap, и в последующем применении получившегося манифеста в Kubernetes.

4.2 Конфигурационные значения (values) группы ресурсов

Группа ресурсов имеет общие для всех книг ресурсов конфигурационные значения:

- название контура (project_name, по умолчанию равно названию пространства имен, используется лишь для пометки в телеметрии)
- настройки базы данных Postgres:
  - URL базы (database_url)
  - alias базы данных (database_alias, таким будет системное имя БД в сервере приложения)
  - имя секрета Kubernetes с парой логин/пароль от БД (db_user_secret_name)
- основное *прикладное хранилище и дополнительные точки монтирования*
- версия Java (java_version, должна быть равна 21)
- настройки *GossipRouter* (gossip_routers_setting)
- настройки *RabbitMQ* (rabbitmq_settings)

5 Комплект приложения. Комплект группы

5.1 Комплект приложений (Appkit)

Комплект приложений указывается для группы ресурсов и определяет перечень артефактов, необходимых для разворачивания системы Global ERP:

- Дистрибутив сервера приложений Global ERP (globalserver)
- Образ прикладного решения (applib)
- Набор конфигураций для сервисов кластера (profile)

Опционально в комплект приложений также входит:

- Исходный код для отладки прикладного решения (appsrc)

```
appkit
├── applib.zip
├── appsrc.zip
├── globalserver.zip
└── profile
    ├── globalscheduler/...
    └── globalserver/...
```

Комплект приложений хранится *на системном хранилище* в виде zip-архивов (globalserver.zip, applib.zip, profile.zip и, опционально, appsrc.zip) вместе с хэш-суммами, которые могут подготавливаться утилитой nscli.

Путь к комплекту приложений на системном хранилище и хэш-суммы также хранятся в ConfigMap группы ресурсов, что используется подами серверов приложения для определения текущей версии комплекта приложений.

Изменить комплект приложений можно при помощи команд:

nsctl: *./resgroup.sh switch_appkit*

nscli: *./appkit.sh switch*

Дистрибутив сервера приложений (globalserver)

Предоставляется вендором.

Образ прикладного решения (applib)

Формируется во время сборки прикладного проекта.

Набор конфигураций для сервисов кластера (profile)

Профиль (profile) используется контейнерами для конфигурации служб. Внутри архива лежат директории, названия которых соответствуют конфигурируемой службе. Внутри каждой такой директории в свою очередь лежат каталоги статических файлов (`static`) и шаблонов (`template`).

Некоторые конфигурационные файлы можно свободно редактировать, такие обычно поставляются в nscli в папке `default/profile`, которая используется утилитой для создания профиля по умолчанию.

В профиле с шаблонами конфигурационных файлов находятся:

- Конфигурация сервера приложений `globalserver/template/config/global3.config.xml`
- Конфигурация сборщика телеметрии `globalserver/template/config/otel-sdk.config.yaml` и `globalserver/template/config/otel-globalserver.config.yaml`
- Конфигурация менеджера заданий `globalscheduler/template/config/quartz.properties`
- Конфигурация проектных настроек системных логов `globalscheduler/template/config/logback-LoggerContext-ext.xml`
- Конфигурация проектных настроек логов сессии `globalscheduler/template/config/logback-LoggerContext-session-ext.xml`

Исходный код для отладки прикладного решения (appsrc)

Вместе с образом прикладного решения (applib) могут формироваться и исходные коды проекта (appsrc), необходимые для облачной отладки прикладного решения.

Состояния комплекта приложений

Комплект приложений бывает в трех состояних:

- `stopped` - остановлен, серверы приложений не работают
- `started` - запущен, серверы приложений работают в нормальном режиме
- `drained` - осушен, серверы приложений работают в сервисном режиме (то есть пользователи не могут подключиться к серверу)

Переключение состояний возможно при помощи команд:

nscli:

- *`./appkit.sh start`*
- *`./appkit.sh stop`*

5.2 Комплект группы (Groupkit)

Комплект группы указывается для группы ресурсов и определяет перечень вспомогательных артефактов:

- Архив с дополнительными библиотеками (libs)
- Архив с дополнительными шрифтами (fonts)
- Файлы JDK для переопределения (java)

```
groupkit
├── libs.zip
├── fonts.zip
└── java
    └── jre
        └── lib
            └── security
                └── cacerts
```

Наличие всех компонентов не обязательно.

Вы можете добавить собственные шрифты, которые будут затем использоваться при создании документа: добавьте их в комплект группы по пути `fonts`. Вы можете использовать java для добавления хранилища собственных сертификатов cacerts: добавьте их в комплект группы по пути `java/jre/lib/security/cacerts`.

Комплект группы также хранится на *системном хранилище*, а в ConfigMap группы ресурсов сохраняются путь и контрольные суммы компонентов (если они зафиксированы через `switch_remote`/`switch_local`). Изменения в комплекте группы требуют перезапуска всех подов.

Обычно комплект группы не требуется.

Переключить комплект группы можно командой nsctl *`./resgroup.sh switch_groupkit`* или через nscli *`./groupkit.sh`*:

```
./groupkit.sh push --namespace my-namespace --source workspace/groupkit --destination␣
↪groupkit/v1
./groupkit.sh switch_remote --config-path ./config.yaml --resgroup my-resgroup --
↪namespace my-namespace --remote-groupkit groupkit/v1
```

`push` для GroupKit работает по той же модели, что и для AppKit: по умолчанию hash-файлы локально пересчитываются и перезаписываются, а при указании `--validate-local-hashes` существующие `.sha1`/`.sha256` валидируются и не перезаписываются.

5.3 Управление комплектами

Управление комплектами выполняется при помощи команды nscli *./appkit.sh*

Создать комплект с минимальным профилем

Создадим комплект приложений с пользовательским профилем по умолчанию в директории `workspace/appkit/v1`.

```
./appkit.py prepare_profile --appkit-dir workspace/appkit/v1
```

Получаем следующую структуру:

```
workspace/appkit/v1
└── profile
    ├── globalscheduler
    │   └── template
    │       └── config
    │           └── quartz.properties
    └── globalserver
        └── template
            └── config
                ├── global3.config.xml
                ├── logback-LoggerContext-ext.xml
                └── logback-LoggerContext-session-ext.xml
```

Упаковать комплект и пересчитать хэши

Давайте добавим к нашему комплекту приложений в директории `workspace/appkit/v1` дистрибутивы globalserver.zip и applib.zip.

Структура комплекта:

```
workspace/appkit/v1
├── applib.zip
├── globalserver.zip
└── profile
    └── ...
```

Примечание

Как можно заметить, в комплекте лежат дистрибутивы globalserver.zip и applib.zip, а также профиль в отдельной папке. Итоговый комплект можно использовать и как комплект приложения, и как комплект группы.

Выполним:

```
./appkit.py refresh_hash --source workspace/appkit/v1
```

```
workspace/appkit/v1
├── applib.zip
├── applib.zip.sha256
├── globalserver.zip
├── globalserver.zip.sha256
├── profile
│   └── ...
├── profile.zip
└── profile.zip.sha256
```

Все компоненты комплекта, хранящиеся в отдельной папке, теперь упакованы в zip-архивы. Рядом с zip-архивами лежат файлы со значениями хэш-сумм.

Отправить комплект на системное хранилище

Давайте отправим этот комплект на системное хранилище по пути `mykit/v1`.

```
./appkit.py push --namespace my-namespace --source workspace/kit --destination mykit/v1
```

Совет

Предварительно запускать `./appkit.py refresh_hash` необязательно: по умолчанию `./appkit.py push` сам пересчитает и перезапишет `.sha1/.sha256` рядом с архивами.

Если нужно использовать заранее подготовленные hash-файлы из CI/CD и не перезаписывать их, добавьте флаг `--validate-local-hashes`.

Теперь вы должны увидеть этот комплект в контейнере nsctl в папке `~/nsctl/workspace/mnt/sys/mykit/v1`.

Обновить GlobalERP

Сменим комплект приложения и обновим базу данных.

```
./appkit.py switch_and_upgrade --namespace my-namespace --resgroup my-resgroup --remote_
↪appkit mykit/v1
```

Внимание

После выполнения команды, комплект приложений *остается в осушенном состоянии*.

Включить GlobalERP

Сделаем GlobalERP снова доступной для пользователей через вывод комплекта приложений из осушенного состояния.

```
./appkit.py start --namespace my-namespace --resgroup my-resgroup
```

6 Классы книг ресурсов

Что такое книги ресурсов и что такое класс книги, смотрите *здесь*.

6.1 global_server_excl

Книга ресурсов для запуска сервера приложений в эксклюзивном режиме. В таком режиме гарантируется, что только один сервер приложений будет работать в один момент времени в рамках группы.

Это позволяет безопасно выполнять административные задачи.

Описывает Deployment с единственным подом, состоящим из *контейнеров gs/globalserver и gs/systemagent*, а также необходимые ресурсы типа Service.

Конфигурационные значения (values)

external_ip

Внешний адрес. Если задан публикует сервер приложения по данному адресу.

admin_user_secret

Имя секрета администратора. Секрет будет примонтирован к контейнеру, а логин/пароль из него будут подставлены в `global3.config.xml`.

victoriametrics_endpoint, loki_endpoint и tempo_endpoint

Хосты с серверами Prometheus, Loki и Tempo, соответственно. По умолчанию равны путям к grafana, если такая книга есть в группе ресурсов.

external_loki_endpoint, external_tempo_endpoint

Аналогично предыдущему пункту, но предназначены для внешних систем сбора телеметрии (например, общей системы мониторинга заказчика).

Не появляется, если *send_metrics_to_external_system = False*.

Конфигурационные характеристики (spec)

send_metrics_to_external_system

Включить экспорт телеметрии во внешнюю систему

6.2 global_server_share

Сервер приложений, который может быть реплицирован на произвольное количество экземпляров.

Описывает Deployment с подами, состоящими из *контейнеров gs/globalserver и gs/systemagent*.

Конфигурационные значения (values)

replicas

Количество реплик, которое должно быть запущенно в kubernetes.

grace_period_duration

Длительность вежливого отключения в секундах.

admin_user_secret

Имя секрета администратора. Секрет будет примонтирован к контейнеру, а логин/пароль из него будут подставлены в `global3.config.xml`.

victoriametrics_endpoint, loki_endpoint и tempo_endpoint

Хосты с серверами Prometheus, Loki и Tempo, соответственно. По умолчанию равны путям к grafana, если такая книга есть в группе ресурсов.

external_loki_endpoint, external_tempo_endpoint

Аналогично предыдущему пункту, но предназначены для внешних систем сбора телеметрии (например, общей системы мониторинга заказчика).

Не появляется, если *send_metrics_to_external_system = False*.

Конфигурационные характеристики (spec)

send_metrics_to_external_system

Включить экспорт телеметрии во внешнюю систему

6.3 global_server_scheduler

Планировщик задач для кластера Global ERP. Отправляет запросы к серверу приложений, согласно расписанию в базе данных. Запускается в единственном экземпляре.

Описывает Deployment с единственным подом, состоящим из *контейнеров gs/globalserver и gs/systemagent*.

Конфигурационные значения (values)

scheduler_token_secret

Имя секрета с токеном (подписью) планировщика.

victoriametrics_endpoint, loki_endpoint и tempo_endpoint

Хосты с серверами Prometheus, Loki и Tempo, соответственно. По умолчанию равны путям к grafana, если такая книга есть в группе ресурсов.

external_loki_endpoint, external_tempo_endpoint

Аналогично предыдущему пункту, но предназначены для внешних систем сбора телеметрии (например, общей системы мониторинга заказчика).

Не появляется, если *send_metrics_to_external_system = False*.

Конфигурационные характеристики (spec)

send_metrics_to_external_system

Включить экспорт телеметрии во внешнюю систему

6.4 haproxy

Балансировщик нагрузки.

Состоит из трех служб в составе контейнера *gs/haproxy*:

- *Менеджер версий (haproxy_manager).* Отслеживает изменения в комплекте приложений. При необходимости запускает, останавливает и перезапускает менеджер конфигурации и собственно HAProxy.
- *Менеджер конфигурации (config_service).* Ведет список серверов приложений и обновляет конфигурацию HAProxy. Перезагрузка конфигурации HAProxy осуществляется через отправку ему SIGUSR2.
- *Собственно HAProxy (haproxy).* Высокопроизводительный HTTP реверс-прокси, специализированный на балансировке запросов.

Описывает Deployment с подами, состоящими из *контейнеров gs/haproxy и gs/haproxyagent*, а также необходимые ресурсы типа Service и Ingress.

Публикация книги ресурсов

Книга ресурсов может быть опубликована через Ingress или через указание внешнего IP (external_ip).

Конфигурационные значения

stat_secret

Имя секрета из которого берется имя пользователя пароль для просмотра статистики.

Если не задано, статистика не включается.

haproxy_tls_secret

Имя секрета из которого берутся параметры для включения https доступа к kubernetes.

Если не задано, https не включается.

6.5 grafana

Книга ресурсов этого класса описывает под со всем стеком мониторинга (контейнер *gs/grafana*):

- собственно Grafana для визуализации собранной телеметрии
- Prometheus, СУБД для хранения временных рядов
- Tempo, система для хранения данных трассировок
- Loki, хранилище логов

Публикация книги ресурсов

Книга ресурсов может быть опубликована через Ingress или через указание внешнего IP (external_ip).

Конфигурационные значения

storage_class и storage_size

Класс и размер используемого для хранения данных PVC соответственно.

grafana_admin_account

Имя секрета с логином и паролем администратора Grafana по умолчанию

6.6 rabbitmq

Такая книга ресурсов развернет одну ноду RabbitMQ (контейнер *gs/rabbitmq*) для внутренних задач кластера Global ERP.

Одной ноды более чем достаточно для работы Global ERP, поскольку отказ брокера сообщений приведет к неполадкам в очень небольшой части функциональности (например, временно не будут приходить уведомления о новых сообщениях во встроенном мессенджере).

Внимание

gs-ctk *не ограничивает вас в выборе* способа развертывания RabbitMQ. Если вы хотите настроить отказоустойчивый кластер RabbitMQ, например, при помощи оператора, вас ничто не останавливает.

Публикация книги ресурсов

Консоль администратора RabbitMQ, создаваемая этой книгой ресурсов, может быть опубликована через Ingress или через указание внешнего IP (external_ip).

Конфигурационные значения

rabbitmq_secret

Имя секрета, содержащего логин и пароль от виртуального хоста RabbitMQ.

rabbitmq_admin_secret

Имя секрета, содержащего логин и пароль от консоль администратора RabbitMQ.

external_port

Внешний порт - порт, для подключения к консоли администратора.

6.7 gossiprouter

Такая книга ресурсов развернет одну ноду GossipRouter - роутера координационных сообщений *JGroups* между кластером серверов приложений.

Подходит для соединения между серверами приложений через межсетевой экран, например, если Global ERP развернута на нескольких кластерах Kubernetes. Если такой потребности нет, посмотрите на менее ресурсоемкий вариант - *jgroups_dns*.

Вот схема подключения при использовании GossipRouter и двух кластеров:

Как можно увидеть, кластера друг с другом не общаются, а подключаются к GossipRouter'ам по их публичным IP/доменам, что упрощает настройку сети.

Для гарантированной совместимости с Global Server, GossipRouter запускается из библиотеки JGroups, поставляемой с дистрибутивом сервера приложений.

Описывает Deployment с единственным подом, состоящим из *контейнеров gs/globalserver и gs/systemagent*, а также необходимые ресурсы типа Service.

Вы можете создать несколько книг ресурсов GossipRouter и опубликовать на разных портах/адресах, чтобы добавить устойчивости в рамках одного кластера Kubernetes..

Конфигурационные значения (values)

external_ip

Внешний адрес. Если задан публикует GossipRouter по данному адресу.

gossip_router_port

Порт GossipRouter для публикации.

victoriametrics_endpoint, loki_endpoint и tempo_endpoint

Хосты с серверами Prometheus, Loki и Tempo, соответственно. Предполагается, что тут будут пути к поду из книги ресурсов grafana, которые устанавливаются при init_values по умолчанию.

Не появляется, если *is_metric_enabled = False*.

external_loki_endpoint, external_tempo_endpoint

Аналогично предыдущему пункту, но предназначены для внешних систем сбора телеметрии (например, общей системы мониторинга заказчика).

Не появляется, если *send_metrics_to_external_system = False*.

Конфигурационные характеристики (spec)

is_metric_enabled

Включение сбора метрик.

send_metrics_to_external_system

Включить экспорт телеметрии во внешнюю систему

6.8 jgroups_dns

Книга ресурсов, описывающая всего один ресурс - Service, перечисляющий все поды с серверами приложений. С его помощью сервер приложений по DNS узнает список других членов группы и формирует кластер *JGroups*.

Дополнительных настроек нет.

6.9 ctk_restAPI

Такая книга ресурсов развернет один под ctk-restapi — сервис для безопасного ролевого подключения к API Kubernetes.

ctk-restapi предоставляет REST API с аутентификацией через JWT и управлением доступом на основе разрешений. Он позволяет управлять Kubernetes-сервисами и выполнять операции в кластере с контролем доступа.

Публикация книги ресурсов

Сервис может быть опубликован через Ingress или через указание внешнего IP (external_ip).

Конфигурационные значения (values)

port

Порт, на котором работает REST API (по умолчанию: *8000*).

replicas

Количество реплик пода (по умолчанию: *1*).

env

Окружение приложения: production, development или test (по умолчанию: *production*).

jwt_algorithm

Алгоритм подписи JWT (по умолчанию: *HS256*).

access_token_expire_minutes

Время жизни access-токена в минутах (по умолчанию: *15*).

login_rate_limit_requests

Лимит запросов на аутентификацию в минуту (по умолчанию: *5*).

rate_limit_requests_per_minute

Общий лимит запросов к API в минуту (по умолчанию: *10*).

log_level

Уровень логирования: DEBUG, INFO, WARNING, ERROR, CRITICAL (по умолчанию: *INFO*).

audit_enabled

Включить аудит-логирование (по умолчанию: *true*).

max_reason_length

Максимальная длина причины для операций (например, перезагрузки) в символах (по умолчанию: *500*).

cors_origins

Список разрешённых источников для CORS (по умолчанию: [«*»]).

cors_allow_credentials

Разрешить передачу учётных данных в CORS (по умолчанию: *false*).

cors_allow_methods

Разрешённые HTTP-методы для CORS (по умолчанию: *[«GET», «POST»]*).

cors_allow_headers

Разрешённые HTTP-заголовки для CORS (по умолчанию: *[«Content-Type», «Authorization»]*).

kubernetes_services

Список конфигураций Kubernetes-сервисов, к которым предоставляется доступ через API. Каждый элемент включает:

- *name* — имя сервиса,
- *namespace* — пространство имён,
- *service_name* — имя Kubernetes-сервиса,
- *label_selector* — селектор меток.

users_config

Список пользователей и их разрешений. Каждый пользователь включает:

- *username* — имя пользователя,
- *permissions* — список разрешений в форматах: service:all, service:view, service:restart, service:NAME:view, service:NAME:restart.

Конфигурационные характеристики (spec)

containers.ctk-restapi.resources

Ресурсы контейнера:

- *requests* — запрашиваемые ресурсы (память и CPU),
- *limits* — ограничения ресурсов (память и CPU).

port

Порт сервиса (должен соответствовать значению из values)

6.10 queuewalker

Книга ресурсов описывает *обходчик очередей* - отдельный сервис Global ERP, который отвечает за чтение и выполнение сообщений из очередей.

Конфигурационные значения (values)

resources

Ресурсы queuewalker.

replicas

Количество реплик пода (по умолчанию: *1*).

token_secret

Токен для обходчика очередей (по умолчанию подставляется токен шедулера)

code_worker

Код обходчика очередей, по которому отправляются запросы к БД (по умолчанию: `default`).

reader_timeout_ms

Частота опроса БД (в миллисекундах) для получения новых очередей (по умолчанию: *1000*).

request_timeout_ms

Таймаут запроса (в миллисекундах) (по умолчанию: *10000*).

sender_threads_worker

Количество потоков, отправляющих задачи на сервер (по умолчанию: *10*).

exec_user

Пользователь, от имени которого выполняются JEXL-скрипты (по умолчанию: `admin`).

queue_max_attempts

Количество неуспешных попыток выполнить очередь, по истечении которых очередь будет остановлена (по умолчанию: *5*).

url_path

Путь в URL, на который отправляются задачи (по умолчанию: `app/sys/rest/ss/pkg/Btk_JexlGatePkg/execute`).

queuewalker_xmx

Максимальный размер кучи для обходчика очередей (по умолчанию: `512m`).

log_level

Уровень логирования: DEBUG, INFO, WARNING, ERROR, ALL (по умолчанию: *INFO*).

7 Обработка нештатных ситуаций

7.1 Отладка конфигурации служб в контейнерах

Иногда необходимо быстро проверить изменения тех или иных параметров конфигурации.

Для этого удобно иметь возможность быстро перезапустить тот или инной процесс без перезапуска контейнера.

Для управления процессами:

1. Зайдите в консоль контейнера
2. Запустите `./shared/supervisor/supervisorctl.sh`

Совет

Использовать команду `help` для получения помощи.

3. Выполните команду „status“ для просмотра текущих служб
4. Остановите службу командой `stop group:app`
5. Поправьте файлы конфигурации
6. Запустите службу командой `start group:app`

7.2 Диагностика запуска контейнера

В ситуации когда контейнер не может запустится и постоянно перезапускается, бывает неудобно анализировать ошибку.

Для более удобного анализа можно переключить контейнер в режим отладки. Для этого измените поставьте параметр группы или книги ресурсов `debug_mode` в `true`.

8 RabbitMQ

Некоторая функциональность Global ERP требует брокера сообщений, поддерживающего протокол AMQP. Очевидным кандидатом является RabbitMQ.

Вы можете выбрать один из двух вариантов развертывания RabbitMQ:

- Книга ресурсов *одноименного* класса. Этот вариант позволит вам без дополнительных усилий настроить RabbitMQ на основе уже развернутого кластера gs-ctk с минимальной потребностью в администрировании.
- Самостоятельно развернуть и администрировать RabbitMQ-кластер или использовать сервис вашего облачного провайдера. Этот вариант подойдет, если вы готовы взять на себя больше ответственности в администрировании RabbitMQ.

8.1 Настройка кластера

Если вы решили использовать встроенный под с RabbitMQ, то добавьте, настройте и включите книгу ресурсов в конфигурационном ресурсе. Gs-ctk автоматически настроет группу ресурсов.

Если вы решили развернуть RabbitMQ без использования gs-ctk, то вам потребуется создать в RabbitMQ виртуальный хост и пользователя со всеми привилегиями на этот виртуальный хост. Затем настройте параметр `rabbitmq` группы ресурсов по образцу.

9 JGroups

Сервера приложений связываются друг с другом через библиотеку JGroups. В gs-ctk есть два способа установления связи:

1. ***JGroups DNS***: добавляется книга ресурсов из одного ресурса - сервиса, при помощи которого сервера приложений могут отправлять запросы на раскрутку соединения.
2. ***GossipRouter***: добавляется книга ресурсов, поднимающая дополнительный роутер сообщений между серверами. Прямого соединения между серверами не происходит.

Первый способ не требует поднятия дополнительного пода (запросы обрабатываются любым доступным сервером приложений), но требует поддержки соединения между подами серверов приложений, а значит не подойдет, если нужно создать единый логический кластер Global ERP на нескольких Kubernetes-кластерах с межсетевым экраном между ними.

О настройке JGroups смотрите в Руководстве системного администратора

10 ctk_restapi

Сервис *Cluster Tool Kit Rest API* предоставляет унифицированный программный интерфейс (REST API) для управления компонентами системы и контроля доступа. Он выступает в роли безопасного шлюза, абстрагируя пользователя от специфики среды запуска.

Сервис поддерживает два режима работы в зависимости от архитектуры инсталляции:

- Cluster Mode (ctk_restapi): Работает внутри Kubernetes. Взаимодействует с Kubernetes API для управления подами и сервисами.
- Standalone Mode (scheduler REST API): Работает как системная служба под управлением systemd. Взаимодействует с системными юнитами для управления процессами.

10.1 Основные функции и назначение

1. Единый интерфейс управления (Abstraction Layer)

Вместо прямого взаимодействия с низкоуровневыми инструментами (kubectl или systemctl), пользователь или внешняя система используют стандартные HTTP-запросы. Это позволяет:

- Скрыть сложность инфраструктуры.
- Предоставить только необходимый набор операций (например, только перезапуск конкретных сервисов), не открывая доступ ко всей системе.

2. Ролевой доступ и управление правами (RBAC)

Сервис реализует гибкую модель разграничения прав. Можно настроить разрешения для пользователей на разных уровнях:

- Просмотр (service:view): только получение статуса сервисов.
- Управление (service:restart): право на перезапуск.
- Гранулярный доступ: права на конкретные сервисы (например, service:NAME:all) или группы сервисов.

3. JWT-аутентификация

Для обеспечения безопасности вместо использования системных сертификатов или ключей доступа применяются JWT-токены:

- Разделение прав доступа без передачи системных учетных данных.
- Контроль времени жизни сессии (TTL).
- Возможность оперативного отзыва доступа.

4. Аудит и логирование

Все действия, совершаемые через API, фиксируются в журнале. Для каждой операции сохраняется информация:

- Кто совершил действие.
- Когда и какой запрос был отправлен.
- Причина выполнения операции (необязательное поле).

5. Защита и стабильность (Rate Limiting)

Для предотвращения перегрузки системы и защиты от злоупотреблений реализованы механизмы ограничения запросов:

- Лимиты на количество попыток аутентификации.
- Общие ограничения на количество API-вызовов в единицу времени.

6. Безопасность сетевого взаимодействия

- CORS: Настройка разрешенных доменов для предотвращения запросов с недоверенных ресурсов.
- Контроль заголовков: Строгая проверка состава и содержания HTTP-пакетов.

11 Приложения

11.1 Nscli

Управление комплектом приложения

Содержит общие команды для загрузки\выгрузки комплекта приложения

Commands:

```
usage: appkit.py [-h] cmd ...

positional arguments:
  cmd                 Команды
    full_help         Распечатать справку
    push              Загрузить комплект сборки в общее хранилище
    switch            Переключить комплект
    switch_and_upgrade
                      Переключить комплект и обновить базу
    start             Запустить комплект сборки
    stop              Остановить комплект
    refresh_hash      Пересчитать хэш суммы
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    prepare_profile   Подготовить профиль

optional arguments:
  -h, --help          show this help message and exit
```

Full_help

```
usage: appkit.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Push

```
usage: appkit.py push [-h] --namespace NAMESPACE --source SOURCE --destination
                      DESTINATION [--validate-local-hashes]

Загружает комплект сборки в общее хранилище

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --source SOURCE       Источник
  --destination DESTINATION
                        Назначение
  --validate-local-hashes
                        Не перезаписывать локальные .sha1/.sha256, а сверять
                        архивы с существующими hash-файлами
```

Switch

```
usage: appkit.py switch [-h] --namespace NAMESPACE --resgroup RESGROUP
                        --remote_appkit REMOTE_APPKIT

Переключает комплект

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --resgroup RESGROUP   Группа
  --remote_appkit REMOTE_APPKIT
                        Адрес удаленного комплекта сборки
```

Switch_and_upgrade

```
usage: appkit.py switch_and_upgrade [-h] --namespace NAMESPACE --resgroup
                                    RESGROUP --remote_appkit REMOTE_APPKIT

Переключает комплект и обновляет базу

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --resgroup RESGROUP   Группа
  --remote_appkit REMOTE_APPKIT
                        Адрес удаленного комплекта сборки
```

Start

```
usage: appkit.py start [-h] --namespace NAMESPACE --resgroup RESGROUP

Запускает комплект сборки

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --resgroup RESGROUP   Группа
```

Stop

```
usage: appkit.py stop [-h] --namespace NAMESPACE --resgroup RESGROUP

Останавливает комплект сборки

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --resgroup RESGROUP   Группа
```

Refresh_hash

```
usage: appkit.py refresh_hash [-h] --source SOURCE

Пересчитывает хэш суммы

optional arguments:
  -h, --help       show this help message and exit
  --source SOURCE  Источник
```

Prepare_profile

```
usage: appkit.py prepare_profile [-h] --appkit-dir APPKIT_DIR

Подготавливает профиль

optional arguments:
  -h, --help            show this help message and exit
  --appkit-dir APPKIT_DIR
                        Папка в которую стоит сохранить profile.zip
```

Управление конфигурационными файлами

Содержит команды для экспортирования и импортирования конфигурации

Commands:

```
usage: configmgr.py [-h] cmd ...

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    export    Экспортировать конфигурацию
    import    Импортировать конфигурацию

optional arguments:
  -h, --help  show this help message and exit
```

Full_help

```
usage: configmgr.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Export

```
usage: configmgr.py export [-h] --namespace NAMESPACE [--resgroups RESGROUPS]
                           [--file FILE] [--to_nfs | --no-to_nfs]

Экспорт конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                        Используемое пространство имен
  --resgroups RESGROUPS
                        Группы ресурсов, разделенные лишь запятой, информацию
                        о которых необходимо выгрузить
  --file FILE           Файл YAML, в который следует выгрузить конфигурацию
  --to_nfs, --no-to_nfs
                        Сохранить файл на системное NFS-хранилище
```

Import

```
usage: configmgr.py import [-h] --namespace NAMESPACE [--resgroups RESGROUPS]
                           [--file FILE] [--force | --no-force]
                           [--from-nfs | --no-from-nfs]
                           [--keep-appkit | --no-keep-appkit]

Импорт конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Используемое пространство имен
  --resgroups RESGROUPS
                        Группы ресурсов, разделенные лишь запятой, информацию
                        о которых необходимо загрузить
  --file FILE           Файл YAML, из которого следует загрузить конфигурацию
  --force, --no-force   Сконфигурировать принудительно, даже если книга уже
                        развернута
  --from-nfs, --no-from-nfs
                        Сохранить файл на системное NFS-хранилище
  --keep-appkit, --no-keep-appkit
                        Сохранить значения appkit
```

Управление комплектом группы

`groupkit.py` управляет загрузкой и переключением GroupKit.

Команды

```
usage: groupkit.py [-h] cmd ...

positional arguments:
  cmd
    full_help      Распечатать справку
    push           Загрузить комплект группы в общее хранилище
    switch_local   Переключить комплект по локальным hash-файлам
    switch_remote  Переключить комплект по hash-файлам из NFS
```

Push

```
usage: groupkit.py push --namespace NAMESPACE --source SOURCE --destination DESTINATION␣
↪[--validate-local-hashes]
```

Загружает GroupKit в системное NFS-хранилище через общий механизм `push_basekit` (тот же, что используется в AppKit).

Порядок работы:

- Проверяются архивы/директории в `SOURCE`; директории автоматически упаковываются в `.zip`.
- По умолчанию для каждого архива пересчитываются и перезаписываются оба hash-файла: `.sha1` и `.sha256`.
- При `--validate-local-hashes` уже существующие hash-файлы не перезаписываются, а валидируются; отсутствующие создаются автоматически.
- Локальный хэш сравнивается с хэшем на NFS; если совпадает, повторная загрузка пропускается.
- При несовпадении архив и hash-файлы обновляются на NFS.
- Если компонент отсутствует локально, его архив и hash-файлы удаляются из `DESTINATION`.

Пример:

```
./groupkit.sh push --namespace gs-ctk --source workspace/groupkit --destination groupkit/
↪v1
```

Switch_remote

```
usage: groupkit.py switch_remote --config-path CONFIG_PATH --namespace NAMESPACE --
↪resgroup RESGROUP --remote-groupkit REMOTE_GROUPKIT
```

Читает hash-файлы `libs.zip/java.zip/fonts.zip` из системного NFS (`REMOTE_GROUPKIT`) и записывает их в конфигурацию группы ресурсов.

Пример:

```
./groupkit.sh switch_remote --config-path ./config.yaml --resgroup gs-cluster-1 --
↪namespace gs-ctk --remote-groupkit groupkit/v1
```

Switch_local

```
usage: groupkit.py switch_local --config-path CONFIG_PATH --resgroup RESGROUP --local-
↪groupkit LOCAL_GROUPKIT --remote-groupkit REMOTE_GROUPKIT
```

Читает hash-файлы `libs.zip/java.zip/fonts.zip` из локальной директории `LOCAL_GROUPKIT` и записывает их в конфигурацию группы ресурсов.

Пример:

```
./groupkit.sh switch_local --config-path ./config.yaml --resgroup gs-cluster-1 --local-
↪groupkit workspace/groupkit --remote-groupkit groupkit/v1
```

Совместимость

- Команда `switch` для GroupKit удалена.
- Совместимый путь использования: `push` (подготовка/сверка хэшей и загрузка в NFS) и затем `switch_remote` или `switch_local` для фиксации hash-значений в конфигурации.

Секреты Helm-чарта

Helm-чарт `gs-ctk` может создавать Kubernetes Secret'ы при включении параметра `addSecrets`.

Для новых конфигураций scheduler private key используйте `stringData`. Поле `data` сохраняется для обратной совместимости.

Для `scheduler-token-secret` в `stringData.private.key` указывается исходное содержимое ключа. При использовании legacy-поля `data.private.key` значение должно быть в формате base64, как это принято в Kubernetes Secret.

Пример в `values.yaml`

```yaml
# Включить добавление секретов через values.yaml
addSecrets: true

# Настройка секретов по умолчанию
secrets:
  gs-admin-auth:
    type: kubernetes.io/basic-auth
    stringData:
      password: admin
      username: admin
  db-user-secret:
    type: kubernetes.io/basic-auth
    stringData:
      password: admin
      username: admin
  scheduler-token-secret:
    stringData:
      private.key: change_me_scheduler_private_key
  secret-haproxy-auth:
    type: kubernetes.io/basic-auth
    stringData:
      username: admin
      password: t0p-Secret
  grafana-admin:
    type: kubernetes.io/basic-auth
    stringData:
      password: admin
      username: admin
  rabbitmq-global:
    type: kubernetes.io/basic-auth
    stringData:
      password: globalrabbitmq
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    username: globalrabbitmq
  rabbitmq-admin:
    type: kubernetes.io/basic-auth
    stringData:
      password: admin
      username: admin
  ctk-restapi-secret:
    type: Opaque
    stringData:
      jwt-secret-key: change_me_in_production_min_32_char
      admin-password: admin
      viewer-password: viewer
  prometheus-admin:
    type: kubernetes.io/basic-auth
    stringData:
      password: admin
      username: admin
```

Применить такой файл можно командой:

```
helm upgrade gs-ctk ~/nscli/helm_chart -f ./my-values.yaml --set addSecrets=true
```

Обновление через `helm upgrade`

```
helm upgrade gs-ctk . \
  --set addSecrets=true \
  --set-string secrets.gs-admin-auth.stringData.password="admin" \
  --set-string secrets.db-user-secret.stringData.username="dbuser" \
  --set-string secrets.db-user-secret.stringData.password="dbpass" \
  --set-json 'secrets.scheduler-token-secret.stringData={"private.key":"token..."}' \
  --set-string secrets.ctk-restapi-secret.stringData.jwt-secret-key="replace_me_with_32_
↪or_more_chars" \
  --set-string secrets.ctk-restapi-secret.stringData.admin-password="admin" \
  --set-string secrets.ctk-restapi-secret.stringData.viewer-password="viewer"
```

Для `scheduler-token-secret` рекомендуется `--set-json`, так как имя ключа `private.key` содержит точку.

`_helpers.tpl`

Для всех секретов, кроме `scheduler-token-secret`, Helm-чарт работает по прежней схеме и просто вставляет их описание из `values.yaml` в итоговый `Secret`.

Отдельный helper `gs-ctk.secretSpec` используется только для `scheduler-token-secret`.

Его задача одна: если одновременно заданы `data` и `stringData` с одинаковым ключом `private.key`, chart сохраняет значение из `data`. Это нужно, чтобы старые сценарии с `secrets.scheduler-token-secret.data` продолжали работать после появления дефолтного `stringData`.

Обратная совместимость

Если в уже существующих сценариях используется `data`, Helm-чарт продолжит принимать такой формат:

```
helm upgrade gs-ctk . \
  --set addSecrets=true \
  --set-json 'secrets.scheduler-token-secret.data={"private.key":"token..."}'
```

Если для одного и того же ключа заданы и `data`, и `stringData`, Helm-чарт сохраняет значение из `data`, чтобы не ломать legacy-сценарии.

Работа с образами

Содержит утилиты по работе с образами

Commands:

```
usage: image.py [-h] cmd ...

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    push      Выгрузить образ
    pull      Загрузить образ

optional arguments:
  -h, --help  show this help message and exit
```

Full_help

```
usage: image.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Push

```
usage: image.py push [-h] [--repo REPO] --name NAME [--version VERSION]

Выгружает образ в регистр докера

optional arguments:
  -h, --help         show this help message and exit
  --repo REPO        Репозиторий
  --name NAME        Имя образа
  --version VERSION  Версия
```

Pull

```
usage: image.py pull [-h] [--repo REPO] --name NAME [--version VERSION]

Загружает образ из регистра докера

optional arguments:
  -h, --help         show this help message and exit
  --repo REPO        Репозиторий
  --name NAME        Имя образа
  --version VERSION  Версия
```

Запускает команды на подах

Запускает команды на подах

Commands:

```
usage: invoker.py [-h] cmd ...

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    run       Запустить команду

optional arguments:
  -h, --help  show this help message and exit
```

Full_help

```
usage: invoker.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Run

```
usage: invoker.py run [-h] --namespace NAMESPACE --app APP --cmd RUN_CMD

Запускает команды на подах

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  --app APP             Источник
  --cmd RUN_CMD         Команда
```

Управление пространством имен

Содержит команды для создания пространства имен

Commands:

```
usage: namespace.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    create_namespace (create_install_scripts)
                        Создать управляемое пространство
    setup_settings      Установить настройки в CI-режиме
    configure_unmanaged
                        Создать (отредактировать) конфигурацию для
                        неуправляемого пакета
    create_unmanaged_resources
                        Создать ресурсы неуправляемого пакета по конфигурации
    create_unmanaged_namespace
                        Создать пространство для неуправляемого пакета по
                        конфигурации
    install_namespace   Развернуть пространство имен
    diagnose            Диагностировать кластер на готовность к развертыванию
                        пространства имен
    upgrade_namespace   Диагностировать кластер на готовность к развертыванию
                        пространства имен
    shell               Открыть командную оболочку
    set_unmanaged_hook  Добавить хук
    unset_unmanaged_hook
                        Удалить хук
    list_unmanaged_hooks
                        Вывести список хуков

optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: namespace.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Create_namespace

```
usage: namespace.py create_namespace [-h] [-y]
                                     [--ignore-checks | --no-ignore-checks]

Создает управляемое пространство

optional arguments:
  -h, --help            show this help message and exit
  -y                    Автоподтверждение установленной конфигурации
  --ignore-checks, --no-ignore-checks
                        Проверки не останавливают выполнение
```

Setup_settings

```
usage: namespace.py setup_settings [-h] --namespace NAMESPACE --nfs-server
                                   NFS_SERVER --nfs-path NFS_PATH
                                   --docker-repository DOCKER_REPOSITORY
                                   [--master-key MASTER_KEY]
                                   [--docker-username DOCKER_USERNAME]
                                   [--docker-password DOCKER_PASSWORD]
                                   [--ignore-checks | --no-ignore-checks]

Устанавливает настройки в CI-режиме

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Название пространства имен
  --nfs-server NFS_SERVER
                        Адрес NFS-сервера
  --nfs-path NFS_PATH   Путь к экспортируемой NFS-сервером папке
  --docker-repository DOCKER_REPOSITORY
                        Адрес Docker-репозитория
  --master-key MASTER_KEY
                        Путь к мастер-ключу (по умолчанию `~/.gs-ctk.priv`)
  --docker-username DOCKER_USERNAME
                        Имя пользователя Docker-репозитория
  --docker-password DOCKER_PASSWORD
                        Пароль пользователя Docker-репозитория (можно также
                        передать через переменную окружения
                        NSCLI_DOCKER_PASSWORD)
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  --ignore-checks, --no-ignore-checks
                        Проверки не останавливают выполнение
```

Configure_unmanaged

```
usage: namespace.py configure_unmanaged [-h] --config CONFIG
                                        [--cryptor | --no-cryptor]

Создает (редактирует) конфигурацию для неуправляемого пакета

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG       Путь к изменяемому файлу конфигурации
  --cryptor, --no-cryptor
                        Попытаться расшифровать данные учетной записи Docker
```

Create_unmanaged_resources

```
usage: namespace.py create_unmanaged_resources [-h] --config CONFIG
                                               --resources RESOURCES
                                               [--cryptor | --no-cryptor]

Создает ресурсы неуправляемого пакета по конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG       Путь к файлу конфигурации
  --resources RESOURCES
                        Путь по которому следует сохранить ресурсы
  --cryptor, --no-cryptor
                        Попытаться расшифровать данные учетной записи Docker
```

Create_unmanaged_namespace

```
usage: namespace.py create_unmanaged_namespace [-h] --config CONFIG
                                               --namespace-deploy
                                               NAMESPACE_DEPLOY

Создает пространство для неуправляемого пакета по конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG       Путь к файлу конфигурации
  --namespace-deploy NAMESPACE_DEPLOY
                        Путь по которому следует сохранить ресурс пространства
                        имен
```

Install_namespace

```
usage: namespace.py install_namespace [-h] [--namespace NAMESPACE]

Разворачивает пространство имен в K8s

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
```

Diagnose

```
usage: namespace.py diagnose [-h]

Диагностирует кластер на готовность к развертыванию пространства имен

optional arguments:
  -h, --help  show this help message and exit
```

Upgrade_namespace

```
usage: namespace.py upgrade_namespace [-h]
                                      [--force | --no-force | -y | --yes | --no-yes]

Диагностирует кластер на готовность к развертыванию пространства имен

optional arguments:
  -h, --help            show this help message and exit
  --force, --no-force, -y, --yes, --no-yes
                        Не требовать подтверждения у пользователя
```

Shell

```
usage: namespace.py shell [-h] [--namespace NAMESPACE] [--app APP]
                          [--container CONTAINER]

Открывает командную оболочку на поде

optional arguments:
  -h, --help            show this help message and exit
  --namespace NAMESPACE
                        Пространство имен
  --app APP             Под (по умолчанию nsctl)
  --container CONTAINER
                        Контейнер внутри пода
```

Set_unmanaged_hook

```
usage: namespace.py set_unmanaged_hook [-h] --name {tolerations} --group GROUP
                                       --book BOOK --hook_file HOOK_FILE
                                       --config CONFIG

Добавляет хук

optional arguments:
  -h, --help            show this help message and exit
  --name {tolerations}  Название хука
  --group GROUP         Название группы
  --book BOOK           Название книги
  --hook_file HOOK_FILE
                        Файл с содержимым хука
  --config CONFIG       Путь к изменяемому файлу конфигурации
```

Unset_unmanaged_hook

```
usage: namespace.py unset_unmanaged_hook [-h] --name {tolerations} --group
                                         GROUP --book BOOK --config CONFIG

Удаляет хук

optional arguments:
  -h, --help            show this help message and exit
  --name {tolerations}  Название хука
  --group GROUP         Название группы
  --book BOOK           Название книги
  --config CONFIG       Путь к изменяемому файлу конфигурации
```

List_unmanaged_hooks

```
usage: namespace.py list_unmanaged_hooks [-h] --group GROUP --book BOOK
                                         --config CONFIG

Выводит список хуков для книги

optional arguments:
  -h, --help       show this help message and exit
  --group GROUP    Название группы
  --book BOOK      Название книги
  --config CONFIG  Путь к изменяемому файлу конфигурации
```

Утилита создания секретов

Содержит мастера создания секретов

Commands:

```
usage: secret.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    register_basic_auth
                        Создать секрет
    create_tls          Создать tls секрет

optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: secret.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Register_basic_auth

```
usage: secret.py register_basic_auth [-h]

Создать секрет для basic авторизации

optional arguments:
  -h, --help  show this help message and exit
```

Create_tls

```
usage: secret.py create_tls [-h]

Создать tls сертификат для защиты коммуникации между серверами

optional arguments:
  -h, --help  show this help message and exit
```

Управление томами

Содержит команды для управления томами

Commands:

```
usage: volume.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    create_local_persistent_volume
                        Создать том

optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: volume.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Create_local_persistent_volume

```
usage: volume.py create_local_persistent_volume [-h] --name NAME --hostname
                                                HOSTNAME --local_path
                                                LOCAL_PATH [--size SIZE]

Создает локальных хранимый том

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя
  --hostname HOSTNAME   Хост
  --local_path LOCAL_PATH
                        Локальный путь
  --size SIZE           размер
```

11.2 Nsctl

Справка по командам в консоли рабочего пространства.

Менеджер облачного отладчика

Содержит команды для запуска облачного отладчика.

Commands:

```
usage: cloud_debugger.py [-h] cmd ...

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    start     Запустить облачный отладчик

optional arguments:
  -h, --help  show this help message and exit
```

Full_help

```
usage: cloud_debugger.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Start

```
usage: cloud_debugger.py start [-h] --group GROUP [--book BOOK]
                               [--timer TIMER] [--force | --no-force | -f]

Запускает облачный отладчик

optional arguments:
  -h, --help            show this help message and exit
  --group GROUP, --name GROUP, -g GROUP, -n GROUP
                        Имя группы ресурсов
  --book BOOK, -b BOOK  Имя книги ресурсов - шаблона для отладчика (должна
                        быть класса global_server_share)
  --timer TIMER, -t TIMER
                        Время существования отладчика, в секундах
  --force, --no-force, -f
                        Не требовать подтверждения пользователя
```

Управление конфигурационными файлами

Содержит команды для экспортирования и импортирования конфигурации

Commands:

```
usage: configmgr.py [-h] cmd ...

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    export    Экспортировать конфигурацию
    import    Импортировать конфигурацию

optional arguments:
  -h, --help  show this help message and exit
```

Full_help

```
usage: configmgr.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Export

```
usage: configmgr.py export [-h] [--resgroups RESGROUPS] [--file FILE]

Экспорт конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --resgroups RESGROUPS
                        Группы ресурсов, разделенные лишь запятой, информацию
                        о которых необходимо выгрузить
  --file FILE           Файл YAML, в который следует выгрузить конфигурацию
```

Import

```
usage: configmgr.py import [-h] [--resgroups RESGROUPS] --file FILE
                           [--force | --no-force]
                           [--keep-appkit | --no-keep-appkit]

Импорт конфигурации

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
optional arguments:
  -h, --help            show this help message and exit
  --resgroups RESGROUPS
                        Группы ресурсов, разделенные лишь запятой, информацию
                        о которых необходимо загрузить
  --file FILE           Файл YAML, из которого следует загрузить конфигурацию
  --force, --no-force   Сконфигурировать принудительно, даже если книга уже
                        развернута
  --keep-appkit, --no-keep-appkit
                        Сохранить значения appkit
```

Команды для работы с файлами

Содержит комадны для работы с файлами через потоки

Commands:

```
usage: file.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    create              создать файл из потока ввода вывода
    create_from_base64
                        создать файл из потока ввода вывода в кодеровки base64
    print               распечатать файл в формате base64

optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: file.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Create

```
usage: file.py create [-h] -f F

Создает файл из потока ввода выода

optional arguments:
  -h, --help  show this help message and exit
  -f F        файл
```

Create_from_base64

```
usage: file.py create_from_base64 [-h] -f F

Создает файл из потока ввода выода в кодеровки base64

optional arguments:
  -h, --help  show this help message and exit
  -f F        файл
```

Print

```
usage: file.py print [-h] -f F

печатает файл в формате base64

optional arguments:
  -h, --help  show this help message and exit
  -f F        файл
```

Команды для управление книгами ресурсов

Содержит команды для создания и конфигурирование книг ресурсов

Commands:

```
usage: resbook.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    create              Создать книгу ресурсов
    init_values         Сконфигурировать значения
    init_spec           Сконфигурировать характеристики
    merge_node_selector
                        Добавить условие
    delete_node_selector
                        Удалить условия
    enable_debug        Включить режим отладки
    disable_debug       Выключить режим отладки
    export_values       Экспортировать значения
    import_values       Импортировать значения
    export_deploy_template
                        Экспортирует шаблон
    override_deploy_template
                        Переопределить шаблон
    create_install_scripts
                        Сгенерировать установочные скрипты
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    enable              Включить книгу ресурсов
    enable_all          Включить все книги ресурсов
    disable             Выключить книгу ресурсов
    disable_all         Выключить все книги ресурсов
    list                Вывести список книг ресурсов
    delete              Удалить книгу ресурсов
    set_hook            Добавить хук
    unset_hook          Удалить хук
    list_hooks          Вывести список хуков

optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: resbook.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Create

```
usage: resbook.py create [-h] --name NAME --group GROUP --class_name
                         CLASS_NAME

Создает книгу ресурсов

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя экземпляра
  --group GROUP         Группа книги
  --class_name CLASS_NAME
                        Имя книги ресурсов
```

Init_values

```
usage: resbook.py init_values [-h] --name NAME --group GROUP
                              [--force | --no-force]

Конфигурирует значения для книги ресурсов

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя экземпляра
  --group GROUP         Группа книги
  --force, --no-force   Сконфигурировать принудительно, даже если книга уже
                        развернута
```

Init_spec

```
usage: resbook.py init_spec [-h] --name NAME --group GROUP
                            [--force | --no-force]

Конфигурирует характеристики для книги ресурсов

optional arguments:
  -h, --help           show this help message and exit
  --name NAME          Имя экземпляра
  --group GROUP        Группа книги
  --force, --no-force  Сконфигурировать принудительно, даже если книга уже
                       развернута
```

Merge_node_selector

```
usage: resbook.py merge_node_selector [-h] --name NAME --group GROUP
                                      --criteria CRITERIA [CRITERIA ...]

Добавляет условие по выбору узла

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя экземпляра
  --group GROUP         Группа книги
  --criteria CRITERIA [CRITERIA ...]
                        Критерии фильтрации(key=value)
```

Delete_node_selector

```
usage: resbook.py delete_node_selector [-h] --name NAME --group GROUP

Удаляет условия по выбору узла

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Enable_debug

```
usage: resbook.py enable_debug [-h] --name NAME --group GROUP

Включает режим отладки.
В режиме отладки pod не останавливается при сбоях

optional arguments:
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Disable_debug

```
usage: resbook.py disable_debug [-h] --name NAME --group GROUP

Выключает режим отладки.
В режиме отладки pod не останавливается при сбоях

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Export_values

```
usage: resbook.py export_values [-h] --name NAME --group GROUP

Конфигурирует значения для книги ресурсов

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Import_values

```
usage: resbook.py import_values [-h] --name NAME --group GROUP

Конфигурирует значения для книги ресурсов

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Export_deploy_template

```
usage: resbook.py export_deploy_template [-h] --name NAME --group GROUP

Экспортировать шаблон генерации ресурсов.
Выводит шаблон генерации ресурсов в поток вывода.

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Override_deploy_template

```
usage: resbook.py override_deploy_template [-h] --name NAME --group GROUP

Переопределяет шаблон генерации ресурсов.
Переопределение шаблона идет из потока ввода. Если шаблон пустой переопределение␣
↪снимается

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Create_install_scripts

```
usage: resbook.py create_install_scripts [-h] --name NAME --group GROUP

Генерирует скрипт для ручной установки в случаи необходимости отладки

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Enable

```
usage: resbook.py enable [-h] --name NAME --group GROUP
                         [--force-reload | --no-force-reload]

Включает книгу ресурсов.
Если включена книга ресурсов и ее группа происходит развертывание ресурсов kubernetes

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя экземпляра
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  --group GROUP         Группа книги
  --force-reload, --no-force-reload
                        Обновить ресурсы K8s принудительно
```

Enable_all

```
usage: resbook.py enable_all [-h] --group GROUP
                             [--force-reload | --no-force-reload]

Включает все книги ресурсов

optional arguments:
  -h, --help            show this help message and exit
  --group GROUP         Группа книги
  --force-reload, --no-force-reload
                        Обновить ресурсы K8s принудительно
```

Disable

```
usage: resbook.py disable [-h] --name NAME --group GROUP

Выключает книгу ресурсов

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Disable_all

```
usage: resbook.py disable_all [-h] --group GROUP

Выключает все книги ресурсов

optional arguments:
  -h, --help     show this help message and exit
  --group GROUP  Группа книги
```

List

```
usage: resbook.py list [-h]

Выводит список книг ресурсов

optional arguments:
  -h, --help  show this help message and exit
```

Delete

```
usage: resbook.py delete [-h] --name NAME --group GROUP

Удаляет книгу ресурсов

optional arguments:
  -h, --help     show this help message and exit
  --name NAME    Имя экземпляра
  --group GROUP  Группа книги
```

Set_hook

```
usage: resbook.py set_hook [-h] --name {tolerations} --group GROUP --book BOOK
                           --hook_file HOOK_FILE

Добавляет хук

optional arguments:
  -h, --help            show this help message and exit
  --name {tolerations}  Название хука
  --group GROUP         Название группы
  --book BOOK           Название книги
  --hook_file HOOK_FILE
                        Файл с содержимым хука
```

Unset_hook

```
usage: resbook.py unset_hook [-h] --name {tolerations} --group GROUP --book
                             BOOK

Удаляет хук

optional arguments:
  -h, --help            show this help message and exit
  --name {tolerations}  Название хука
  --group GROUP         Название группы
  --book BOOK           Название книги
```

List_hooks

```
usage: resbook.py list_hooks [-h] --group GROUP --book BOOK

Выводит список хуков для книги

optional arguments:
  -h, --help     show this help message and exit
  --group GROUP  Название группы
  --book BOOK    Название книги
```

Управление группой ресурсов

Содержит команды управления группой ресурсов

Commands:

```
usage: resgroup.py [-h] cmd ...

positional arguments:
  cmd                   Команды
    full_help           Распечатать справку
    create              Создать группу
    delete              Удалить группу
    enable              Включить группу
    disable             Выключить группу
    full_reload         Обновить все группы с перезапуском подов при
                        необходимости
    init_values         Сконфигурировать значения группы
    init_spec           Сконфигурировать характеристики группы
    extra_mountpoint_ls
                        Вывести список дополнительных точек монтирования"
    add_extra_mountpoint
                        Создать дополнительную точку монтирования"
    rm_extra_mountpoint
                        Удалить дополнительную точку монтирования
    edit_extra_mountpoint
                        Редактировать дополнительную точку монтирования
    export              Экспортирует конфигурацию
    import              Импортирует конфигурацию файла в NFS-хранилище или из
                        стандартного ввода
    switch_appkit       Переключить комплект приложения
    switch_groupkit     Переключить комплект группы
    start_appkit        Стартовать комплект приложения
    stop_appkit         Остановить комплект приложения
    drain_appkit        Осушить комплект приложения
    wait_lease          Ожидать остановки лизов
    create_pvc          Создать PVC для группы ресурсов
    delete_pvc          Удалить PVC для группы ресурсов
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
optional arguments:
  -h, --help            show this help message and exit
```

Full_help

```
usage: resgroup.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Create

```
usage: resgroup.py create [-h] --name NAME

Создает группу

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Delete

```
usage: resgroup.py delete [-h] --name NAME

Удаляет группу

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Enable

```
usage: resgroup.py enable [-h] --name NAME
                          [--force-reload | --no-force-reload]

Включает группу

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя группы
  --force-reload, --no-force-reload
                        Обновить ресурсы K8s принудительно
```

Disable

```
usage: resgroup.py disable [-h] --name NAME

Выключает группу

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Full_reload

```
usage: resgroup.py full_reload [-h]

Обновляет все группы с перезапуском подов при необходимости

optional arguments:
  -h, --help  show this help message and exit
```

Init_values

```
usage: resgroup.py init_values [-h] [--name NAME] [--all | --no-all]
                               [--force | --no-force] [--yes | --no-yes | -y]

Мастер конфигурации значений группы.
Конфигурирует как саму группу так и входящие в нее книги ресурсов

optional arguments:
  -h, --help           show this help message and exit
  --name NAME          Имя группы
  --all, --no-all      Обойти все группы
  --force, --no-force  Сконфигурировать принудительно, даже если группа уже
                       развернута
  --yes, --no-yes, -y  Выполнить в неинтерактивном режиме
```

Init_spec

```
usage: resgroup.py init_spec [-h] [--name NAME] [--all | --no-all]
                             [--force | --no-force] [--yes | --no-yes | -y]

Мастер конфигурации характеристик группы.
Конфигурирует как саму группу так и входящие в нее книги ресурсов

optional arguments:
  -h, --help           show this help message and exit
  --name NAME          Имя группы
  --all, --no-all      Обойти все группы
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  --force, --no-force  Сконфигурировать принудительно, даже если группа уже
                       развернута
  --yes, --no-yes, -y  Выполнить в неинтерактивном режиме
```

Extra_mountpoint_ls

```
usage: resgroup.py extra_mountpoint_ls [-h] --group-name GROUP_NAME

Выводит список дополнительных точек монтирования

optional arguments:
  -h, --help            show this help message and exit
  --group-name GROUP_NAME
                        Имя группы
```

Add_extra_mountpoint

```
usage: resgroup.py add_extra_mountpoint [-h] --group-name GROUP_NAME
                                        [--vol-name VOL_NAME]
                                        [--vol-type VOL_TYPE]
                                        [--vol-mnt-path VOL_MNT_PATH]
                                        [--vol-srv-path VOL_SRV_PATH]
                                        [--vol-server VOL_SERVER]
                                        [--quiet | --no-quiet]

Создает дополнительную точку монтирования

optional arguments:
  -h, --help            show this help message and exit
  --group-name GROUP_NAME
                        Имя группы
  --vol-name VOL_NAME   Имя дополнительной точки монтирования
  --vol-type VOL_TYPE   Тип дополнительного раздела (по умолчанию 'nfs')
  --vol-mnt-path VOL_MNT_PATH
                        Путь монтирования раздела в поде относительно
                        `/root/globalserver/workspace/mnt/`
  --vol-srv-path VOL_SRV_PATH
                        Путь к папке на NFS-сервере
  --vol-server VOL_SERVER
                        Адрес NFS-сервера
  --quiet, --no-quiet   Не спрашивать подтверждения у пользователя
```

Rm_extra_mountpoint

```
usage: resgroup.py rm_extra_mountpoint [-h] --group-name GROUP_NAME --vol-name
                                       VOL_NAME

Удаляет дополнительную точку монтирования

optional arguments:
  -h, --help            show this help message and exit
  --group-name GROUP_NAME
                        Имя группы
  --vol-name VOL_NAME   Имя дополнительной точки монтирования
```

Edit_extra_mountpoint

```
usage: resgroup.py edit_extra_mountpoint [-h] --group-name GROUP_NAME
                                         --vol-name VOL_NAME

Редактирует дополнительную точку монтирования

optional arguments:
  -h, --help            show this help message and exit
  --group-name GROUP_NAME
                        Имя группы
  --vol-name VOL_NAME   Имя дополнительной точки монтирования
```

Export

```
usage: resgroup.py export [-h] --name NAME [--with_values | --no-with_values]
                          [--to_nfs | --no-to_nfs] [--path PATH]

Экспорт конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Имя группы
  --with_values, --no-with_values
                        Обрабатывать значения (default: False)
  --to_nfs, --no-to_nfs
                        Сохраняет конфигурацию в NFS-хранилище
  --path PATH           Путь, по которому следует сохранить конфигурацию в
                        NFS-хранилище. Если не указано, то конфгурация
                        сохраняется в корень nfs-раздела.
```

Import

```
usage: resgroup.py import [-h] [--with_values | --no-with_values]
                          [--from_nfs | --no-from_nfs] [--path PATH]

Импорт конфигурации

optional arguments:
  -h, --help            show this help message and exit
  --with_values, --no-with_values
                        Обрабатывать значения (default: False)
  --from_nfs, --no-from_nfs
                        Импортировать ли конфигурацию из NFS-хранилища. Если
                        не указано, то конфгурация берется из стандартного
                        ввода.
  --path PATH           Путь к импортируемой конфигурации в NFS-хранилище.
```

Switch_appkit

```
usage: resgroup.py switch_appkit [-h] --name NAME --path PATH

Переключает комплект приложения

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
  --path PATH  Путь к комплекту приложения
```

Switch_groupkit

```
usage: resgroup.py switch_groupkit [-h] --name NAME --path PATH

Переключает комплект группы

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
  --path PATH  Путь к комплекту группы
```

Start_appkit

```
usage: resgroup.py start_appkit [-h] --name NAME

Стартует комплект приложений

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Stop_appkit

```
usage: resgroup.py stop_appkit [-h] --name NAME

Останавливает комплект приложений

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Drain_appkit

```
usage: resgroup.py drain_appkit [-h] --name NAME

Осушает комплект приложений.
При этом контейнеры и службы перестают работать без возможности доступа к ним␣
↪пользователей

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Wait_lease

```
usage: resgroup.py wait_lease [-h] --name NAME

Ожидает остановки лизов.
Используется в скриптах обновления для ожидания отсоединения всех пользователей

optional arguments:
  -h, --help   show this help message and exit
  --name NAME  Имя группы
```

Create_pvc

```
usage: resgroup.py create_pvc [-h] --pvc-name PVC_NAME --group GROUP
                              --classname CLASSNAME
                              [--requested-size REQUESTED_SIZE] --single-node
                              | --no-single-node | --rwo | --no-rwo

Создает PersistentVolumeClaim для подключения постоянного тома к группе ресурсов

optional arguments:
  -h, --help            show this help message and exit
  --pvc-name PVC_NAME   Имя создаваемого PVC
  --group GROUP         Имя группы
  --classname CLASSNAME
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                        Класс хранилища
  --requested-size REQUESTED_SIZE
                        Размер хранилища
  --single-node, --no-single-node, --rwo, --no-rwo
                        Включить ли режим ReadWriteOnce вместо ReadWriteMany
```

Delete_pvc

```
usage: resgroup.py delete_pvc [-h] --pvc-name PVC_NAME --group GROUP

Удаляет PersistentVolumeClaim для подключения постоянного тома к группе ресурсов

optional arguments:
  -h, --help           show this help message and exit
  --pvc-name PVC_NAME  Имя создаваемого PVC
  --group GROUP        Имя группы
```

Утилита тестирования

Содержит команды для тестирования кластера

Commands:

```
usage: tester.py [-h] cmd ...

positional arguments:
  cmd           Команды
    full_help   Распечатать справку
    test-network
                тестировать подключение к сети
    test-write  тестировать возможность писать в NFS-хранилище

optional arguments:
  -h, --help    show this help message and exit
```

Full_help

```
usage: tester.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Test-network

```
usage: tester.py test-network [-h]

Тестирует подключение к сети

optional arguments:
  -h, --help  show this help message and exit
```

Test-write

```
usage: tester.py test-write [-h] [--resgroup RESGROUP] [--test | --no-test]

Тестирует возможность писать в NFS-хранилище

optional arguments:
  -h, --help           show this help message and exit
  --resgroup RESGROUP  название группы
  --test, --no-test    выдать исключение в случае ошибки
```

Утилиты

Содержит прочие утилиты для упрощения работы администратора

Commands:

```
usage: utils.py [-h] cmd ...

positional arguments:
  cmd              Команды
    full_help      Распечатать справку
    clear-grafana  очистить раздел для метрик
    backup-grafana
                   копировать раздел для метрик на NFS-хранилище
    restore-grafana
                   восстановить постоянный раздел для Grafana, Tempo,
                   Prometheus и Loki из копии на диске

optional arguments:
  -h, --help       show this help message and exit
```

Full_help

```
usage: utils.py full_help [-h]

optional arguments:
  -h, --help  show this help message and exit
```

Clear-grafana

```
usage: utils.py clear-grafana [-h] --resgroup RESGROUP --resbook RESBOOK
                              [--force | --no-force]

Удаляет содержимое постоянного раздела для Grafana, Tempo, Prometheus и Loki.
Не выполняется, если не отключен под Grafana (используйте "./resbook.sh disable").

optional arguments:
  -h, --help           show this help message and exit
  --resgroup RESGROUP  название группы
  --resbook RESBOOK    название книги ресурсов Grafana
  --force, --no-force  не подтверждать намерение пользователя
```

Backup-grafana

```
usage: utils.py backup-grafana [-h] --resgroup RESGROUP --resbook RESBOOK
                               [--rel-path REL_PATH] [--nfs-server NFS_SERVER]
                               [--nfs-path NFS_PATH]

Копирует содержимое постоянного раздела для Grafana, Tempo, Prometheus и Loki на NFS-
↪хранилище.
Под Grafana следует отключить перед выполнением операции (используйте "./resbook.sh␣
↪disable").

optional arguments:
  -h, --help            show this help message and exit
  --resgroup RESGROUP   название группы
  --resbook RESBOOK     название книги ресурсов Grafana
  --rel-path REL_PATH   путь на NFS-хранилище, по которому будет сохранена
                        копия (по умолчанию сохраняется в папку grafana-
                        YYYYMMDDhhmmss)
  --nfs-server NFS_SERVER
                        адрес сервера NFS (если не указано, бекап сохраняется
                        на системный отдел)
  --nfs-path NFS_PATH   адрес пути к монтируемой папки на сервере NFS
```

Restore-grafana

```
usage: utils.py restore-grafana [-h] --resgroup RESGROUP --resbook RESBOOK
                                --rel-path REL_PATH [--nfs-server NFS_SERVER]
                                [--nfs-path NFS_PATH] [--force | --no-force]

Восстанавливает раздел для метрик из копии на NFS-хранилище.
Не выполняется, если не отключен под Grafana (используйте "./resbook.sh disable").

optional arguments:
  -h, --help            show this help message and exit
  --resgroup RESGROUP   название группы
  --resbook RESBOOK     название книги ресурсов Grafana
  --rel-path REL_PATH   путь на NFS-хранилище, по которому хранится копия
  --nfs-server NFS_SERVER
                        адрес сервера NFS (если не указано, бекап сохраняется
                        на системный отдел)
  --nfs-path NFS_PATH   адрес пути к монтируемой папки на сервере NFS
  --force, --no-force   не подтверждать намерение пользователя
```

12 Справка по параметрам конфигурации

12.1 Параметры группы ресурсов

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя группы ресурсов

namespace

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Название пространства имен, в котором необходимо развернуть группу.

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли группа ресурсов. Если указать `false`, ресурсы группы будут свернуты.

appkit

- **Опциональный**
- **Допустимые типы:**
  - *AppkitData*

Комплект приложения. Подробнее о комплекте приложений читайте *здесь*.

groupkit

- **Опциональный**
- **Допустимые типы:**
  - Строковый
  - *GroupKitData*

Комплект группы. Подробнее о комплекте группы читайте *здесь*.

Примеры:

-
```
groupkits/v1
```

timezone

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
Europe/Moscow
```

Часовой пояс.

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

drain_cluster_before_db_upgrade

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Определяет, требует ли обновление БД осушения кластера.

database_url

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Путь к базе данных в формате JDBC.

Примеры:

- `jdbc:postgresql://host:5432/database`

database_write_pool_parameters

- **Опциональный**
- **Допустимые типы:**
- **Значение по умолчанию:**

```
accept_tx_session: true
accept_no_tx_session: true
```

Параметры основного пула соединений сервера приложений к СУБД. Читайте подробнее здесь.

database_readonly_replica_url

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Путь к читающей реплике базы данных в формате JDBC.

Примеры:

- `jdbc:postgresql://ro-replica:5432/database`

database_read_only_pool_parameters

- **Опциональный**
- **Допустимые типы:**
- **Значение по умолчанию:**

```
accept_tx_session: false
accept_no_tx_session: true
```

Параметры читающего пула соединений сервера приложений к СУБД. Читайте подробнее здесь.

database_extra_pools

- **Опциональный**
- **Допустимые типы:**

Дополнительные пулы соединений сервера приложений к СУБД. Читайте подробнее здесь.

Примеры:

- ```
  first:
    minPoolSize: 1
    max_pool_size: 700
    acceptTxSession: true
    accept_no_tx_session: false
  second:
    enabled: false
    accept_primary: true
    acceptSecondary: true
    acceptTxSession: false
    accept_no_tx_session: true
  ```
  ```

database_special_pools

- **Опциональный**
- **Допустимые типы:**

Специальные пулы соединений сервера приложений к СУБД. Читайте подробнее здесь.

Примеры:

- ```
  first:
    minPoolSize: 1
    max_pool_size: 700
    acceptTxSession: true
    accept_no_tx_session: false
  second:
    enabled: false
    accept_primary: true
    acceptSecondary: true
    acceptTxSession: false
    accept_no_tx_session: true
  ```

database_alias

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя БД в Global ERP. Отображается на странице входа.

Примеры:

- ```
  DEMOERP
  ```

database_secret

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Секрет с логином и паролем к БД.

upgrade_job

- **Опциональный**
- **Допустимые типы:**
  - *UpgradeJobSpec*
- **Значение по умолчанию:**

```yaml
type:
name:
enabled: false
debug_mode:
node_selector:
hooks:
java_xmx: 3500M
victoriametrics_endpoint:
loki_endpoint:
tempo_endpoint:
send_metrics_to_external_system:
external_loki_endpoint:
resources:
  requests:
    memory: 4G
    cpu: '2'
  limits:
    memory: 4G
    cpu: '2'
agent_resources:
  requests:
    memory: 500M
    cpu: '1'
  limits:
    memory: 1G
    cpu: '1'
admin_secret:
admin_secret_type:
```

Настройки Kubernetes Job обновления БД.

hot_reload

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Разрешены ли горячие обновления.

hot_reload_grace_period

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
900
```

Время в секундах, предоставленное пользователям с устаревшим прикладным решением на переподключение. Значение меньше 120 (например, -1) позволяет находиться таким пользователям не ограниченное время.

Примеры:

- ```
  -1
  ```
- ```
  3600
  ```

database_schema_version

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Версия схемы БД. При изменении целевой версии схемы базы данных - автоматически будет запущено обновление БД. Информация о текущей версии хранится в конфигурационном ресурсе, в поле `status`.

search_domains

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Поисковые домены DNS

Примеры:

- ```
  - mycompany.ru
  ```
- ```
  - universe-1.ru
  - universe-a.ru
  ```

gossiprouters

- **Опциональный**
- **Допустимые типы:**
  - *GossipRouterData*

Список хостов/портов gossiprouter

Примеры:

- ```
  - host: gr.mycompany.ru
    port: 1100
  ```
  ```

- ```
  - host: 10.0.1.2
    port: 1200
  - host: 10.0.2.2
    port: 1200
  ```

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя единого (для всех контейнеров по умолчанию) секрета администратора

admin_secret_type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Тип единого (для всех контейнеров по умолчанию) секрета администратора

Примеры:

- ```
  kubernetes.io/basic-auth
  ```

victoriametrics_secret

- **Опциональный**
- **Другие названия:**
  - prometheus_secret
- **Допустимые типы:**
  - Строковый

Имя секрета basic-auth для настройки обязательной авторизации в Prometheus. Пустое поле означает отсутствие авторизации.

prometheus_secret

То же самое, что и `victoriametrics_secret`.

java_version

- **Опциональный**
- **Допустимые типы:**
  - Один из: `'8'`, `'21'`
- **Значение по умолчанию:**

```
'21'
```

Версия Java

appvol

- **Обязательный**
- **Допустимые типы:**
  - *NFSVolume*
  - *PVCVolume*

Прикладное хранилище для пользовательских файлов

Примеры:

- ```
  type: nfs
  server: nfs.mycompany.ru
  path: /export
  ```

extra_vols

- **Опциональный**
- **Допустимые типы:**
  - *ExtraVolume*

Дополнительные точки монтирования

Примеры:

- ```
  name: mp1
  path: extras/mp1
  volume:
    type: nfs
    server: mfs.mycompany.ru
    path: /export2
  ```

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

rabbitmq

- **Опциональный**
- **Допустимые типы:**
  - Булевый
  - *RabbitMQSettingsData*

Настройки RabbitMQ. По умолчанию - берутся из настроек книги ресурсов rabbitmq, если такой нет - RabbitMQ отключен. Поставьте `rabbitmq: false`, чтобы выключить. Используйте этот параметр, чтобы настроить внешний RabbitMQ.

Примеры:

- ```
  false
  ```
- ```
  enabled: true
  host: rabbitmq.mycompany.com
  port: 5672
  vhost: globalerp
  secret: rabbitmq-secret
  ```

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи.

Примеры:

- `external.loki.domain:3100`

external_tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

OTLPGRPC-эндпоинт внешнего инстанса Tempo, в который будут отправляться трейсы пользовательских операций.

Примеры:

- `external.tempo.domain:4317`

external_prometheus_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Prometheus (либо VictoriaMetrics), в который будут отправляться метрики.

Примеры:

- `external.prometheus.domain:9090`

external_prometheus_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Секрет от внешнего Prometheus

project_name

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Название контура для фильтрации во внешней Grafana (по умолчанию - название пространства имен).

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes.

12.2 Параметры книг ресурсов

global_scheduler

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `global_scheduler`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
800M
```

Предельный объем кучи виртуальной машины Java

Примеры:

- `4000M`
- `32G`

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 4G
  cpu: '2'
limits:
  memory: 4G
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с приложением.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

scheduler_token_secret

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Секрет с токеном шедулера

global_server_excl

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `global_server_excl`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
3500M
```

Предельный объем кучи виртуальной машины Java

Примеры:

-
```
4000M
```
-
```
32G
```

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 4G
  cpu: '2'
limits:
  memory: 4G
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с приложением.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя секрета администратора Global ERP

admin_secret_type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Тип секрета администратора Global ERP

Примеры:

- `kubernetes.io/basic-auth`

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*

Способ публикации книги ресурсов. Книга ресурсов `global_server_excl` поддерживает публикацию только через IP.

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```

global_server_share

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `global_server_share`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов
  ```

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
3500M
```

Предельный объем кучи виртуальной машины Java

Примеры:

- `4000M`
- `32G`

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 4G
  cpu: '2'
limits:
  memory: 4G
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с приложением.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя секрета администратора Global ERP

admin_secret_type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Тип секрета администратора Global ERP

Примеры:

- `kubernetes.io/basic-auth`

replicas

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

  `2`

Количество реплик.

grace_period_duration

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный

Период ожидания выхода пользователя при удалении пода.

debugger

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `debugger`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
3500M
```

Предельный объем кучи виртуальной машины Java

Примеры:

- ```
  4000M
  ```
- ```
  32G
  ```

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```yaml
requests:
  memory: 4G
  cpu: '2'
limits:
  memory: 4G
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с приложением.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя секрета администратора Global ERP

admin_secret_type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Тип секрета администратора Global ERP

Примеры:

- `kubernetes.io/basic-auth`

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

debugger_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 2G
  cpu: '2'
limits:
  memory: 12G
  cpu: '4'
```

Запросы и лимиты ресурсов контейнера с отладчиком.

available_until

- **Обязательный**
- **Допустимые типы:**
  - Целочисленный

Точка во времени в виде эпохи Unix, до которой отладчик должен оставаться доступен.

gossiprouter

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `gossiprouter`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
800M
```

Предельный объем кучи виртуальной машины Java

Примеры:

- ```
  4000M
  ```
- ```
  32G
  ```

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 1G
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с роутером сообщений.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

gossip_router_port

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

  ```
  12001
  ```

Порт GossipRouter

grafana

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `grafana`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 2Gi
  cpu: '1'
limits:
  memory: 4Gi
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера

storage_class

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Класс хранилища для постоянного тома Grafana

storage_size

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
1Gi
```

Запрашиваемое пространство

cadvisor_service_account

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Устаревший параметр

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя секрета администратора

prometheus_stack_service

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
prometheus-k8s.monitoring.svc.cluster.local
```

Адрес сервиса kube-prometheus-stack.

prometheus_stack_scrape_interval

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
'30'
```

Интервал запроса метрик из kube-prometheus-stack.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

external_tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

OTLPGRPC-эндпоинт внешнего инстанса Tempo, в который будут отправляться трейсы пользовательских операций. По умолчанию указывается эндпоинт для группы ресурсов.

external_prometheus_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Prometheus (либо VictoriaMetrics), в который будут отправляться метрики. По умолчанию указывается эндпоинт для группы ресурсов.

haproxy

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `haproxy`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1Gi
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с балансировщиком.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```yaml
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

statistics_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Секрет для сбора статистики.

haproxy_tls_secret

- **Опциональный**
- **Другие названия:**
  - tls_secret
- **Допустимые типы:**
  - Строковый

Секрет для ключа и сертификата TLS

tls_secret

То же самое, что и `haproxy_tls_secret`.

replicas

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
1
```

Количество реплики при работе через Ingress

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

jgroups_dns

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `jgroups_dns`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

rabbitmq

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `rabbitmq`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 2Gi
  cpu: '1'
limits:
  memory: 4Gi
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера.

apmq_secret

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Секрет для соединения между RabbitMQ и GlobalERP

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Секрет администратора

vhost

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
globalrabbitmq
```

Название виртуального хоста

admin_port

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
15672
```

Порт консоли администратора

img2doc

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `img2doc`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 1G
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера.

port

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
7000
```

Порт `img2doc`.

replicas

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
1
```

Количество реплик `img2doc`.

queuewalker

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `queuewalker`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 512M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера.

replicas

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
1
```

Количество реплик обходчика очередей

token_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Токен для обходчика очередей (при пустом значении подставляется токен планировщика задач)

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

code_worker

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
default
```

Код обходчика очередей, по которому отправляются запросы к БД

reader_timeout_ms

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
1000
```

Частота опроса БД (в миллисекундах) для получения новых очередей

request_timeout_ms

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
10000
```

Таймаут запроса (в миллисекундах)

sender_threads_worker

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
10
```

Количество потоков, отправляющих задачи на сервер

exec_user

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
admin
```

Пользователь, от имени которого выполняются JEXL-скрипты

queue_max_attempts

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
5
```

Количество неуспешных попыток выполнить очередь (по истечении которых очередь будет остановлена)

url_path

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
app/sys/rest/ss/pkg/Btk_JexlGatePkg/execute
```

Путь в URL, на который отправляются задачи

queuewalker_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
512m
```

Максимальный размер кучи для обходчика очередей

log_level

- **Опциональный**
- **Допустимые типы:**
  - Один из: `DEBUG`, `INFO`, `WARN`, `ERROR`, `ALL`
- **Значение по умолчанию:**

```
INFO
```

Уровень подробности логов

globalserver_host

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хост глобал-сервера (`https://globalerp.mycompany.ru`), по умолчанию будет подставлен внутренний сервис кубернетес

ctk-restapi

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `ctk-restapi`

Тип книги ресурсов.

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя книги ресурсов

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включена ли книга ресурсов. Если указать `false`, ресурсы книги будут свернуты.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Включить debug

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

service

- **Опциональный**
- **Допустимые типы:**
  - *IPServiceDescData*
  - *IngressServiceDescData*

Способ публикации книги ресурсов

Примеры:

- ```
  type: ip
  external_ip: 10.20.0.5
  ```
- ```
  type: ingress
  ingress_class: nginx
  host: globalerp.mycompany.ru
  annotations: {}
  ingress_tls_secret: ''
  ```

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 256Mi
  cpu: 500m
limits:
  memory: 512Mi
  cpu: 1000m
```

Запросы и лимиты ресурсов контейнера

replicas

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
1
```

Количество реплик

port

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
8000
```

Порт сервиса API

jwt_algorithm

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
HS256
```

Алгоритм JWT

access_token_expire_minutes

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
15
```

Время истечения токена (минут)

login_rate_limit_requests

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
5
```

Лимит запросов входа

rate_limit_requests_per_minute

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
10
```

Лимит запросов в минуту

log_level

- **Опциональный**
- **Допустимые типы:**
  - Один из: `DEBUG`, `INFO`, `WARNING`, `ERROR`, `CRITICAL`
- **Значение по умолчанию:**

```
INFO
```

Уровень логирования

audit_enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Включить аудит

max_reason_length

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

```
500
```

Максимальная длина причины

cors_origins

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
- '*'
```

CORS источники

cors_allow_credentials

- **Опциональный**
- **Допустимые типы:**
  - Булевый

CORS учетные данные

cors_allow_methods

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
- GET
- POST
```

CORS методы

cors_allow_headers

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
- Content-Type
- Authorization
```

CORS заголовки

kubernetes_services

- **Опциональный**
- **Допустимые типы:**
  - *KubernetesServiceData*

Kubernetes сервисы

users

- **Опциональный**
- **Допустимые типы:**
  - *UserData*

Пользователи

env

- **Опциональный**
- **Допустимые типы:**
  - Один из: `production`, `development`, `test`
- **Значение по умолчанию:**

```
production
```

Окружение приложения

secret_name

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
ctk-restapi-secret
```

Имя секрета

12.3 Вспомогательные структуры

AppkitData

applib_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма образа прикладного решения по алгоритму SHA-1. Устарел.

appsrc_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма исходного кода прикладного решения по алгоритму SHA-1. Устарел.

applib_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма образа прикладного решения по алгоритму SHA-256.

appsrc_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма исходного кода прикладного решения по алгоритму SHA-256.

globalserver_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма сервера приложений по алгоритму SHA-1. Устарел.

profile_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма образа конфигурационного профиля по алгоритму SHA-1. Устарел.

globalserver_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма сервера приложений по алгоритму SHA-256.

profile_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма образа конфигурационного профиля по алгоритму SHA-256. Устарел.

globalserver_instance

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Версия сервера приложений, желательно обновлять при каждом изменении комплекта приложения.

path

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Путь к комплекту приложений на системном NFS-хранилище.

state

- **Опциональный**
- **Допустимые типы:**
  - Один из: `started`, `drained`, `stopped`
- **Значение по умолчанию:**

```
stopped
```

Состояние комплекта приложений.

GroupKitData

path

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Путь к комплекту группы на системном NFS-хранилище.

libs_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма дополнительных библиотек по алгоритму SHA-1. Устарел.

java_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма файлов JDK для переопределения по алгоритму SHA-1. Устарел.

fonts_sha1

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма архива с дополнительными шрифтами по алгоритму SHA-1. Устарел.

libs_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма дополнительных библиотек по алгоритму SHA-256.

java_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма файлов JDK для переопределения по алгоритму SHA-256.

fonts_sha256

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хеш-сумма архива с дополнительными шрифтами по алгоритму SHA-256.

HooksData

tolerations

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Хук на Tolerations.

ContainerResourcesSectionData

memory

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Ограничение по памяти

cpu

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Органичение по процессору

ContainerResourcesData

requests

- **Обязательный**
- **Допустимые типы:**
  - *ContainerResourcesSectionData*

Запросы контейнера. Столько нода кластера будет резервировать из доступных ей ресурсов перед запуском пода.

limits

- **Обязательный**
- **Допустимые типы:**
  - *ContainerResourcesSectionData*

Лимиты контейнера. Определяют максимальное потребление ресурсов. Контейнер не сможет получить больше процессорного времени, а при превышении лимита памяти будет остановлен по OOM.

UpgradeJobSpec

type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Не используется.

name

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Не используется.

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Включено ли обновление через Kubernetes Job.

debug_mode

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Выключить перезагрузку подов при ошибках. Помогает при отладке.

node_selector

- **Опциональный**
- **Допустимые типы:**
  - Словарь

Селектор нод, как в Kubernetes

hooks

- **Опциональный**
- **Допустимые типы:**
  - *HooksData*

Хуки. Подроблнее о хуках читайте здесь.

java_xmx

- **Опциональный**
- **Допустимые типы:**
  - Строковый
- **Значение по умолчанию:**

```
3500M
```

Предельный объем кучи виртуальной машины Java

Примеры:

- ```
  4000M
  ```
- ```
  32G
  ```

victoriametrics_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса VictoriaMetrics, не требует ручного определения.

loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Loki, не требует ручного определения.

tempo_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внутреннего инстанса Tempo, не требует ручного определения.

send_metrics_to_external_system

- **Опциональный**
- **Допустимые типы:**
  - Булевый

Определяет, отправлять ли данные во внешнюю систему. По умолчанию используется аналогичное значение для группы ресурсов.

external_loki_endpoint

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Эндпоинт внешнего инстанса Loki, в который будут отправляться логи. По умолчанию указывается эндпоинт для группы ресурсов.

resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 4G
  cpu: '2'
limits:
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  memory: 4G
  cpu: '2'
```

Запросы и лимиты ресурсов контейнера с приложением.

agent_resources

- **Опциональный**
- **Допустимые типы:**
  - *ContainerResourcesData*
- **Значение по умолчанию:**

```
requests:
  memory: 500M
  cpu: '1'
limits:
  memory: 1G
  cpu: '1'
```

Запросы и лимиты ресурсов контейнера с агентом сбора телеметрии.

admin_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Имя секрета администратора Global ERP

admin_secret_type

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Тип секрета администратора Global ERP

Примеры:

- `kubernetes.io/basic-auth`

GossipRouterData

host

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Адрес к GossipRouter

Примеры:

- `gr.mycompany.ru`
- `10.0.1.2`

port

- **Обязательный**
- **Допустимые типы:**
  - Целочисленный

Порт GossipRouter

Примеры:

- `12001`
- `1101`

NFSVolume

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `nfs`

Тип раздела. Значение `nfs` означает, что должен использоваться NFS.

server

- **Обязательный**
- **Допустимые типы:**
  - Строковый

NFS-сервер

path

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Путь на NFS

PVCVolume

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `pvc`

Не используется

name

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Не используется

classname

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Не используется

label_selectors

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Не используется

ExtraVolume

name

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Определяет название точки монтирования при просмотре пода в `kubectl describe`.

path

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Путь к точке монтирования в файловой системе контейнера относительно `/root/globalserver/workspace/mnt/`.

volume

- **Обязательный**
- **Допустимые типы:**
  - *NFSVolume*
  - *PVCVolume*

Описание монтируемого раздела.

RabbitMQSettingsData

enabled

- **Опциональный**
- **Допустимые типы:**
  - Булевый
- **Значение по умолчанию:**

```
true
```

Должен ли быть доступен серверам приложений RabbitMQ

host

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Адрес к серверу RabbitMQ

Примеры:

- ```
  rabbitmq.mycompany.com
  ```

port

- **Опциональный**
- **Допустимые типы:**
  - Целочисленный
- **Значение по умолчанию:**

  ```
  5672
  ```

Порт сервера RabbitMQ

vhost

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Виртуальный хост RabbitMQ

secret

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Секрет для подключения к RabbitMQ
  ```

IPServiceDescData

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `ip`

Тип публикации

external_ip

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Внешний IP кластера, на который будет публиковаться сервис

IngressServiceDescData

type

- **Обязательный**
- **Допустимые типы:**
  - Один из: `ingress`

Тип публикации

ingress_class

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Класс Ingress

host

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Поле `Host` HTTP-запроса, по которому он будет маршрутизироваться на данный сервис

annotations

- **Опциональный**
- **Допустимые типы:**

Дополнительные аннотация. Читайте подробнее здесь.

ingress_tls_secret

- **Опциональный**
- **Допустимые типы:**
  - Строковый

TLS-секрет для Ingress-прокси

KubernetesServiceData

name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя сервиса

service_name

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя Kubernetes сервиса

label_selector

- **Обязательный**
- **Допустимые типы:**
  - Строковый

K8s селектор меток

UserData

username

- **Обязательный**
- **Допустимые типы:**
  - Строковый

Имя пользователя

permissions

- **Опциональный**
- **Допустимые типы:**
  - Строковый

Список разрешений

12.4 Пример полной конфигурации

```
apiVersion: global-system.ru/v1
kind: GlobalConfiguration
metadata:
  name: config
spec:
  type: advanced
  resgroups:
  - enabled: true
    timezone: Europe/Moscow
    drain_cluster_before_db_upgrade: true
    database_write_pool_parameters:
      accept_tx_session: true
      accept_no_tx_session: true
    database_readonly_replica_url: ''
    database_read_only_pool_parameters:
      accept_tx_session: false
      accept_no_tx_session: true
    database_extra_pools: {}
    database_special_pools: {}
    database_secret: db-user-secret
    hot_reload: false
    hot_reload_grace_period: 900
    search_domains: []
    gossiprouters: []
    admin_secret: gs-admin-auth
    victoriametrics_secret: victoria-metrics-auth
    java_version: '21'
    extra_vols: []
    debug_mode: false
    external_prometheus_secret: ''
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
  node_selector: {}
  upgrade_job:
    java_xmx: 3500M
    resources:
      requests:
        memory: 4G
        cpu: '2'
      limits:
        memory: 4G
        cpu: '2'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
  resbooks:
  - type: global_scheduler
    name: global-scheduler
    enabled: true
    java_xmx: 800M
    resources:
      requests:
        memory: 4G
        cpu: '2'
      limits:
        memory: 4G
        cpu: '2'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    scheduler_token_secret: scheduler-token-secret
  - type: global_server_excl
    name: global-server-excl
    enabled: true
    java_xmx: 3500M
    resources:
      requests:
        memory: 4G
        cpu: '2'
      limits:
        memory: 4G
        cpu: '2'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
      limits:
        memory: 1G
        cpu: '1'
  - type: global_server_share
    name: global-server-share
    enabled: true
    java_xmx: 3500M
    resources:
      requests:
        memory: 4G
        cpu: '2'
      limits:
        memory: 4G
        cpu: '2'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    replicas: 2
    grace_period_duration: 0
  - type: gossiprouter
    name: gossiprouter
    enabled: true
    java_xmx: 800M
    resources:
      requests:
        memory: 1G
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    gossip_router_port: 12001
  - type: grafana
    name: grafana
    enabled: true
    resources:
      requests:
        memory: 2Gi
        cpu: '1'
      limits:
        memory: 4Gi
        cpu: '2'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    storage_class: ''
    storage_size: 1Gi
    admin_secret: grafana-admin
    prometheus_stack_service: prometheus-k8s.monitoring.svc.cluster.local
    prometheus_stack_scrape_interval: '30'
  - type: haproxy
    name: haproxy
    enabled: true
    resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1Gi
        cpu: '2'
    agent_resources:
      requests:
        memory: 500M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    statistics_secret: secret-haproxy-auth
    replicas: 1
  - type: jgroups_dns
    name: jgroups-dns
    enabled: true
  - type: rabbitmq
    name: rabbitmq
    enabled: true
    resources:
      requests:
        memory: 2Gi
        cpu: '1'
      limits:
        memory: 4Gi
        cpu: '2'
    apmq_secret: rabbitmq-global
    admin_secret: rabbitmq-admin
    vhost: globalrabbitmq
    admin_port: 15672
  - type: img2doc
    name: img2doc
    enabled: true
    resources:
      requests:
        memory: 1G
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    port: 7000
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    replicas: 1
  - type: queuewalker
    name: queuewalker
    enabled: true
    resources:
      requests:
        memory: 512M
        cpu: '1'
      limits:
        memory: 1G
        cpu: '1'
    replicas: 1
    token_secret: ''
    code_worker: default
    reader_timeout_ms: 1000
    request_timeout_ms: 10000
    sender_threads_worker: 10
    exec_user: admin
    queue_max_attempts: 5
    url_path: app/sys/rest/ss/pkg/Btk_JexlGatePkg/execute
    queuewalker_xmx: 512m
    log_level: INFO
    globalserver_host: ''
  - type: ctk-restapi
    name: ctk-restapi
    enabled: true
    debug_mode: false
    resources:
      requests:
        memory: 256Mi
        cpu: 500m
      limits:
        memory: 512Mi
        cpu: 1000m
    replicas: 1
    port: 8000
    jwt_algorithm: HS256
    access_token_expire_minutes: 15
    login_rate_limit_requests: 5
    rate_limit_requests_per_minute: 10
    log_level: INFO
    audit_enabled: true
    max_reason_length: 500
    cors_origins:
    - '*'
    cors_allow_credentials: false
    cors_allow_methods:
    - GET
    - POST
    cors_allow_headers:
    - Content-Type
    - Authorization
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    kubernetes_services:
    - name: global-scheduler
      service_name: gs-cluster-1-global-scheduler
      label_selector: app=gs-cluster-1-global-scheduler
    users:
    - username: admin
      permissions:
      - service:all
    env: production
    secret_name: ctk-restapi-secret
```

13 Примечания к версиям

13.1 6.2.3 - 2026-08-21

Артефакты: nscli, helm.

Исправления

- Исправлена работа Pyroscope на некоторых кластерах.
- Оптимизирован расчет хешей в nsctl, благодаря чему работа контроллера должна быть более отзывчивой.
- Устранена ошибка «Не получилось освободить лиз» при незапланированном завершении работы контейнера.

Другие изменения

- Увеличен запрос ОЗУ для отладчика с 2 до 6 гигабайт.

13.2 6.2.2 - 2026-08-05

Артефакты: nscli, helm.

Исправления

- Устранена ошибка при запуске облачного отладчика.
- Исправлен выбор режима обновления (горячее/полное) комплекта приложений.
- Увеличен период ожидания запуска сервера приложений до 6 минут. При превышении времени ожидания, выбрасывается исключение.

13.3 6.2.1 - 2026-07-17

Артефакты: nscli, helm.

Исправления

- Исправлена работа облачного отладчика.
- Устранена ошибка «Не удалось загрузить файл ... Не удалось переместить временный файл», уменьшена вероятность OOM при `./appkit.sh push`.
- Устранена ошибка в `push` при попытке прочитать контрольную сумму из файла.

Другие изменения

- Переработано логирование nsctl.
- Теперь можно указать количество попыток загрузки комплекта приложений/группы при помощи параметра `./appkit.sh push --remote-copy-max-attempts` (аналогично для `./groupkit.sh`) или через переменную окружения `NSCLI_REMOTE_COPY_MAX_ATTEMPTS`.
- В `nsctl` добавлен инструмент для снятия отладочной информации `./debug_info.sh`.

13.4 6.2.0 - 2026-06-18

Артефакты: nscli, helm.

Новые функции

- **Теперь возможно создавать конфигурацию gs-ctk при помощи Helm-чарта.**

  Для этого в `values.yaml` чарта добавлено поле `helmResgroup`. Подробнее читайте здесь.

  Обратите также внимание на обновленную инструкцию о развертывании одним чартом, где данная функция активно используется.

- **GossipRouter теперь доступен для серверов приложений внутри своей группы ресурсов автоматически, без необходимости указывать отдельный адрес.**

Другие изменения

- Добавлена поддержка ApplicationServer версий 1.27.0-rc11, 1.28.0-ms8 и выше.
- Удалены лишние лейблы из метрик, добавлена фильтрация метрик kube-prometheus-stack

13.5 6.1.1 - 2026-06-11

Артефакты: nscli, helm.

Исправления

- Исправлен запуск пода gossiprouter
- Доступ к Pyroscope из Grafana теперь работает при отсутствии доступа к интернета.

13.6 6.1.0 - 2026-05-20

Артефакты: nscli, helm.

Новые функции

- **Теперь можно указать произвольное количество пулов соединений и их параметры.**

  Для этого добавлены параметры группы ресурсов *`database_extra_pools`*, *`database_write_pool_parameters`*, *`database_read_only_pool_parameters`* и *`database_special_pools`*. Обратите внимание, что для применения настроек *требуется пересоздать profile.zip*. Подробнее читайте здесь.
- Обновлены доски «Database client connections», «HTTP Server» и «System overview».

Исправления

- Устранена ошибка агента телеметрии `'GsServiceAgent' object has no attribute 'cur_gc_workload'`, мешающая выгрузке метрики о времени, потраченном на работу сборщика мусора.

13.7 6.0.8 - 2026-07-17

Артефакты: nscli, helm.

Исправления

- Устранена ошибка «Не удалось загрузить файл ... Не удалось переместить временный файл», уменьшена вероятность OOM при `./appkit.sh push`. *(порт. из 6.2.1)*
- Устранена ошибка в `push` при попытке прочитать контрольную сумму из файла. *(порт. из 6.2.1)*

Другие изменения

- Переработано логирование nsctl. *(порт. из 6.2.1)*

13.8 6.0.7 - 2026-07-03

Артефакты: nscli, helm.

Другие изменения

- В `nsctl` добавлен инструмент для снятия отладочной информации `./debug_info.sh`. *(порт. из 6.2.1)*

13.9 6.0.6 - 2026-06-09

Артефакты: nscli, helm.

Исправления

- Исправлен запуск пода gossiprouter *(порт. из 6.1.1)*

13.10 6.0.5 - 2026-05-15

Артефакты: nscli, helm.

Исправления

- Исправлен запуск nscli на Alt Linux p10 и других дистрибутивах с Python 3.9.

Другие изменения

- Повышение стабильности и скорости загрузки appkit в nsctl pod. Вывод прогресса загрузки appkit при push to nsctl pod в одну строку

13.11 6.0.4 - 2026-05-12

Артефакты: nscli, helm.

Другие изменения

- Добавлена поддержка Python 3.12

13.12 6.0.3 - 2026-05-07

Артефакты: nscli, helm.

Исправления

- Исправлена отправка метрик во внешнюю систему.
- Исправлена логика сбора метрики `systemagent_health_check`.

Другие изменения

- Helm-чарт теперь публикуется в нативном формате с расширением tgz, а не zip.
- Утилита `./make_dump.sh` снимает дампы с queuewalker.
- Обновлен queuewalker.
- Внимание! Для стабильной работы обновленного стэка телеметрии, рекомендуем повысить ресурсы системных агентов до 500Мб request, 1Гб limit.

13.13 6.0.2 - 2026-04-29

Исправления

- Под queuewalker исправно загружает переопределения из комплекта группы.
- Устранена ошибка при запуске VictoriaMetrics, связанная с наличием в пароле авторизации спецсимволов.

13.14 6.0.1 - 2026-04-28

Другие изменения

- Добавлена возможность указать пул соединений только для чтения через параметр группы ресурсов `database_readonly_replica_url`. (Для работы параметра требуется пересоздать профиль)
- В дашборды интеграций добавлено значение по умолчанию.
- Обновлен queuewalker.

13.15 6.0.0 - 2026-04-21

Новые функции

- **Добавлено встроенное профилирование серверов приложений при помощи Pyroscope.**
- **При сборке gs-ctk в nsctl фиксируются хеши других образов gs-ctk, которые он может использовать при развертывании других контейнеров.**

  Для включения развертывания при помощи хешей необходимо в values.yaml указать imageMode: «digestAuto».

- **Ключ планировщика задач теперь можно применить одной командой.**

  При помощи одной команды можно сохранить ключ из секрета Kubernetes в БД и перезагрузить планировщик задач и сервера приложений. В nscli команда доступна, как `./scheduler_keys.sh push_scheduler_keys --namespace gs-ctk --group gs-cluster-1`, в nsctl: `./utils.sh push_scheduler_keys --group gs-cluster-1`.

Другие изменения

- Вместо Prometheus в качестве встроенной СУБД для метрик теперь используется VictoriaMetrics.
- Вместо Otelcol и Promtail для сбора телеметрии теперь используется Grafana Alloy.
- Загрузка приватного ключа scheduler в кластер без обертки в Base64 (`stringData`) с сохранением обратной совместимости (`data`)
- В дашборде Grafana «Checklist» теперь отслеживается соответствие ключа планировщика в секрете Kubernetes и в базе данных (метрика «Scheduler key convergence»).
- Переработано создание ключей планировщика задач. Теперь ключи создаются командой nscli `./scheduler_keys.sh generate_new_keys [--destination workspace/keys]`.

Исправления

- HAProxy теперь стартует без ошибок после перезапуска при наличии в пространстве имен подов-сирот.

Примечания к развертыванию

Версия обратно совместима с gs-ctk 5. Никаких изменений конфигурации для развертывания новой версии gs-ctk не требуется.

13.16 5.9.4 - 2026-05-04

Другие изменения

- Обновлен queuewalker.

13.17 5.9.3 - 2026-04-20

Другие изменения

- Обновлен queuewalker.

13.18 5.9.2 - 2026-04-20

Исправления

- Исправлена балансировка запросов обходчика очередей

13.19 5.9.1 - 2026-04-02

Исправления

- Исправлена ошибка в работе `./appkit.sh push` на этапе загрузки profile.zip.

13.20 5.9.0 - 2026-03-30

Новые функции

- **Реализована отправка трейсов и метрик во внешние системы.**
- **Добавлена команда для снятия дампа.**

  Можно снимать дампы сервера приложений, шедулера и GossipRouter при помощи команды nscli `./make_dump.sh`, а также на самом поде при помощи той же команды.

Другие изменения

- Теперь при запуске шедулера в лог контейнера выводится публичный ключ, связанный с приватным ключем шедулера.

Исправления

- Исправлена генерация профиля.

13.21 5.8.2 - 2026-03-17

Исправления

- Исправлена работа nscli с версией Python 3.9.
- Исправлена работа queuewalker

Другие изменения

- Возвращена возможность по умолчанию перезаписывать файлы хешей для appkit/groupkit.

  По умолчанию push работает, как было до 5.8.0: локальные .sha1 и .sha256 рядом с архивом всегда пересчитываются и перезаписываются. Новый режим контроля целостности сохранён под флагом –validate-local-hashes: тогда существующие hash-файлы не затираются, а сверяются с архивом или создаются при отсутствии.

13.22 5.8.1 - 2026-03-10

Исправления

- Устранена ошибка `TypeError: object of type 'NoneType' has no len()`, которая иногда возникает при обновлении БД.

Другие изменения

- Секрет администратора для задачи (K8s Job) обновления может браться из книг ресурсов `global_server_excl` и `global_server_share`, если такой секрет не указан явно для задачи обновления или группы ресурсов в целом.
- Теперь возможно указать секрет TLS для ресурсов Ingress. Подробнее в документации.
- Теперь переменная KUBECONFIG учитывается при загрузке конфигурационного файла клиента Kubernetes.
- Добавлены более явные сообщения о критических ошибках сервера приложений и прикладного решения при обновлении БД.

13.23 5.8.0 - 2026-02-27

Новые функции

- **Добавлен ctk_restapi.**

  Сервис для упрощенного мониторинга и перезагрузки Kubernetes-подов, через Rest API с ролевой моделью доступа и аудитом событий.

  Подробная документация о структуре и целях сервиса: https://help.globalerp.ru/books/G3SysAdminDoc/SNAPSHOT/html/070_addition/043_ctk_restapi_manual.html

- **Добавлен обходчик очередей.**

  Сервис для Global ERP, отвечающий за чтение и выполнение сообщений из очередей интеграции.

  Добавить обходчик очередей можно через команду `./configmgr.sh configure_resbook --config-path config.yaml --resgroup gs-cluster-1 --resbook queuewalker --classname queuewalker`. Подробнее читайте в https://help.globalerp.ru/books/GsCtk/SNAPSHOT/html/resbooks/100_queuewalker.html.

- **Для комплекта группы унифицирован контроль целостности; добавлен режим фиксации контейнерных образов по digest.**

  Подробное описание команд и параметров (генерация/проверка `.sha1/.sha256`, сверка локальных и NFS-хэшей, использование `switch_remote/switch_local` для фиксации хэшей в конфигурации) - https://help.globalerp.ru/books/G3SysAdminDoc/SNAPSHOT/html/040_global-k8s/020_

install.html#id9, а также (режимы `imageMode`/`imageDigests` для Helm) - https://help.globalerp.ru/books/G3SysAdminDoc/SNAPSHOT/html/040_global-k8s/020_install.html#id3.

- **Добавлена возможность указать запросы и пределы ресурсов для nsctl через Helm-чарт.**

  Настроить можно через параметр `nsctlResources` в `values.yaml`.

Исправления

- Исправлено создание K8s Job обновления при использовании с версиями, содержащими знак `#`.
- Статистика и телеметрия с HAProxy доступна, как в состоянии комплекта приложений `started`, так и `stopped`, и `drained`.

Другие изменения

- Команды `./appkit.sh switch_local` и `switch_remote` теперь выдают более полные сообщения об ошибках.
- Увеличен объем эфемерного хранилища для временных файлов с 5 до 10 ГБ.
- Увеличено максимальное количество потоков в JGroups DNS по умолчанию с 8 до 32.

13.24 5.7.0 - 2026-01-28

Новые функции

- **Добавлена возможность запускать обновление БД с помощью ресурса Kubernetes типа Job.**

  Запуск обновления при помощи Job позволяет выполнять обновление на изолированном свежем поде, что увеличивает стабильность и управляемость процесса. Позволяет контролировать процесс обновления БД без использования nscli.

  Вы можете начать использовать Job обновления, настроив его в мастере конфигурации группы ресурсов:

```
nscli $ ./configmgr.sh configure_resgroup --config-path config.yaml --resgroup gs-
↪cluster-1
```

```
Настройка группы ресурсов: gs-cluster-2
=======================================

...
Включить обновление при помощи ресурса Kubernetes типа Job?[да,нет]:да
Укажите размер кучи Java (Xmx) для Job обновления:3500M
Введите запрос CPU для globalserver:2
Введите запрос MEMORY для globalserver:4G
Введите лимиты CPU для globalserver:2
Введите лимиты MEMORY для globalserver:4G
Введите запрос CPU для systemagent:1
Введите запрос MEMORY для systemagent:250M
Введите лимиты CPU для systemagent:1
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
Введите лимиты MEMORY для systemagent:500M
...
```

Вы также можете настроить Job обновления без мастера конфигурации, добавив секцию `upgrade_job` в настройки группы ресурсов

```
apiVersion: global-system.ru/v1
kind: GlobalConfiguration
metadata:
  name: config
spec:
  type: advanced
  resgroups:
  - name: gs-cluster-1
    upgrade_job:
      enabled: true
      java_xmx: 3500M
      resources:
        requests:
          memory: 4G
          cpu: '2'
        limits:
          memory: 4G
          cpu: '2'
      agent_resources:
        requests:
          memory: 250M
          cpu: '1'
        limits:
          memory: 500M
          cpu: '1'
    ...
```

Переключайте старый и новый механизм обновления при помощи параметра `upgrade_job.enabled` группы ресурсов.

После выполнения `./appkit.sh upgrade` (или изменения параметра группы ресурсов `database_schema_version` другим способом) и применения конфигурационного ресурса, nsctl должен удалить старый Job вместе с созданным им подом, и создать новый.

Job и под обновления можно определить по лейблу `global-system.ru/is-upgrade-job-pod-for`, значение которого соответствует названию группы ресурсов. Название Job состоит из префикса `upgrade-job-` и названия группы ресурсов, например, `upgrade-job-gs-cluster-1`.

Другие изменения

- Команда `./appkit.sh upgrade` теперь по умолчанию добавляет к `database_schema_version` случайный суффикс формата `#xxxx`. Это гарантирует запуск обновления в случае потери инкремента версии (например, при запуске команды в эфемерном контейнере). Добавьте флаг `--do-not-add-random-part`, чтобы не добавлять суффикс.

13.25 5.6.0 - 2026-12-29

Новые функции

- Добавлена авторизация в Prometheus.

13.26 5.5.1 - 2026-12-29

Исправления

- Исправлен баг в команде `./appkit.sh wait_upgrade`, связанный с появлением сообщения о неустановленном CRD в больших количествах.

13.27 5.5.0 - 2026-12-23

Новые функции

- В конфигурационные ресурсы указываются настройки по умолчанию.
- Добавлены команды `./configmgr.sh enable_resgroup`, `disable_resgroup`, `enable_resbook`, `disable_resbook`.
- Переработана доска VM Memory.
- Добавлены дополнительные фильтры метрик.
- Обновлена Grafana.

13.28 5.4.1 - 2026-12-18

Исправления

- Исправлен импорт настроек из файлов профиля logback-LoggerContext-ext.xml и logback-LoggerContext-session-ext.xml.

13.29 5.4.0 - 2026-12-15

Другие изменения

- Из метрики занятой памяти в чеклисте убран объем файлового кеша.

Новые функции

- Добавлены горячие обновления.
- Теперь Helm-чарт публикуется не только с nscli, но и отдельно.
- Добавлен конфигурационный ресурс пространства имен (`NamespacedGlobalConfiguration`).
- Секреты теперь можно создавать через Helm–чарт.
- Добавлен сбор телеметрии шедулера.
- Реализована возможность указать интервал запроса метрик из kube-prometheus-stack.
- Добавлена метрика доступности внешнего Loki.
- Добавлена доска с телеметрией интеграции.

Исправления

- Исправлен баг в лизах, возникающий когда счетчик микросекунд равен 0.
- Исправлено развертывание GossipRouter при наличии дополнительных точек монтирования.

13.30 5.3.1 - 2026-12-01

Другие изменения

- Теперь хранится меньше поколений ReplicaSet (3 вместо 10).

Исправления

- Исправлена ошибка при изменении конфигурации/обновлении gs-ctk, когда управляемые ресурсы сливались, а не изменялись, сохраняя устаревшие параметры.
- Исправлено поведение лизов.

13.31 5.3.0 - 2026-11-10

Другие изменения

- HAProxy показывает заглушку в осушенном и остановленном состоянии комплекта приложений.
- мастер включает JGroups DNS по умолчанию.

Новые функции

- Добавлен контейнер img2doc.
- Переработано обновление БД.

13.32 5.2.5 - 2026-10-31

Другие изменения

- Актуализирован конфиг планировщика.

Исправления

- Исправлен статус HAProxy на дашборде System overview.
- Исправлена работа JGroups DNS.

13.33 5.2.4 - 2026-10-29

Исправления

- Исправлен шаблон gossiprouter.

13.34 5.2.3 - 2026-10-28

Исправления

- Исправлена миграция gossiprouter с версии 4.*.*.

13.35 5.2.2 - 2026-10-24

Исправления

- Исправлен статус NFS в Grafana.

13.36 5.2.1 - 2026-10-10

Исправления

- Исправление работы с VK Cloud Containers.

13.37 5.2.0 - 2026-10-08

Другие изменения

- Обновление зависимостей.

Новые функции

- Второй пакет доработок по телеметрии и чек-листу.

13.38 5.1.6 - 2026-09-24

Исправления

- Исправлен баг, из-за которого из грушкита не подгружались JAR-библиотеки.

13.39 5.1.5 - 2026-09-19

Другие изменения

- В appkit.sh добавлены команды просмотра версии базы данных, получения логов и ожидания завершения обновления.

Исправления

- Исправлена ошибка в Helm-чарте, которая не позволяла развернуть чарт в прямо указанном пространстве имен.

13.40 5.1.4 - 2026-09-12

Новые функции

- Облачный отладчик запускается в закрытой среде.

Исправления

- Исправить миграцию gossiprouter и grafana с версии 4.*.*.
- Исправить дефолтные значения для Helm-чарта.

13.41 5.1.3 - 2026-09-01

Исправления

- Порт HAProxy TLS для Grafana теперь определяется корректно.

13.42 5.1.2 - 2026-08-28

Другие изменения

- Увеличены дефолтные запросы и лимиты Grafana.
- Настроено развертывание с ограниченными правами.

13.43 5.1.1 - 2026-08-26

Другие изменения

- Добавлены флаги `--skip-checks`, `--skip-profile-creation` для `./appkit.sh push`.

13.44 5.1.0 - 2026-08-18

Другие изменения

- Обновлен RabbitMQ.

Новые функции

- В Grafana добавлен чеклист.
- Теперь используется kube-prometheus-stack, а не собственная реализация.

13.45 5.0.1 - 2026-08-15

Другие изменения

- Grafana больше не требует PV, как и в 4.*.

Исправления

- Исправлены неправильные подсказки и ошибки в командах мастера конфигурации `nscli/configmgr.sh`.
- Исправлена ошибка с экспортом телеметрии во внешнюю систему.

13.46 5.0.0 - 2026-07-22

Новые функции

- Возможно развернуть через Helm.
- Использует шаблон оператора.
- Единый конфигурационный ресурс.
- Облачный отладчик (пока только при наличии интернета).