GsCtk

Выпуск 1.0.3

апр. 01, 2025

Содержание

1 Предисловие

`Gs-ctk` - Global System Cluster Tool Kit. Комплект инструментов для управления кластером

1.1 Назначение

Необходим для работы `Global ERP` в режиме высокой доступности и горизонтального масштабирования.

Основная задача данного комплекта - это отделить параметры конфигурации среды kubernetes, манифестов развертывания, скриптов обновления.

Автоматизация развертывания производится для всех компонентов `Global ERP`:

- Балансировщики нагрузки
- Сервера приложений
- Менеджер заданий
- Средства мониторинга
- Службы логирования
- Механизмы проверки живучести узлов кластера
- Сторожевые таймеры

1.2 Целевая аудитория

Данное руководство предназначено для администраторов серверов `Global ERP`.

1.3 Необходимые навыки

- Базовые навыки работы с Linux
- Понимание принципов работы Kubernetes

Примечание: Для глубокой адаптации потребуются знания:

- python
- jinja
- docker

1.4 Основные характеристики `Gs-ctk`

Высокая скорость настройки

Применения 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

Гибкость настройки

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

Надежность

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

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.

Управляемое обновление

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

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.

Интеграция с внешним CI

В кластере можно легко развернуть решения собранные на произвольном CI

1.5 Программное обеспечение

Linux

Используется для запуска следующих компонентов `gs-ctk`

- `nscli`
- `docker`

Kubernetes

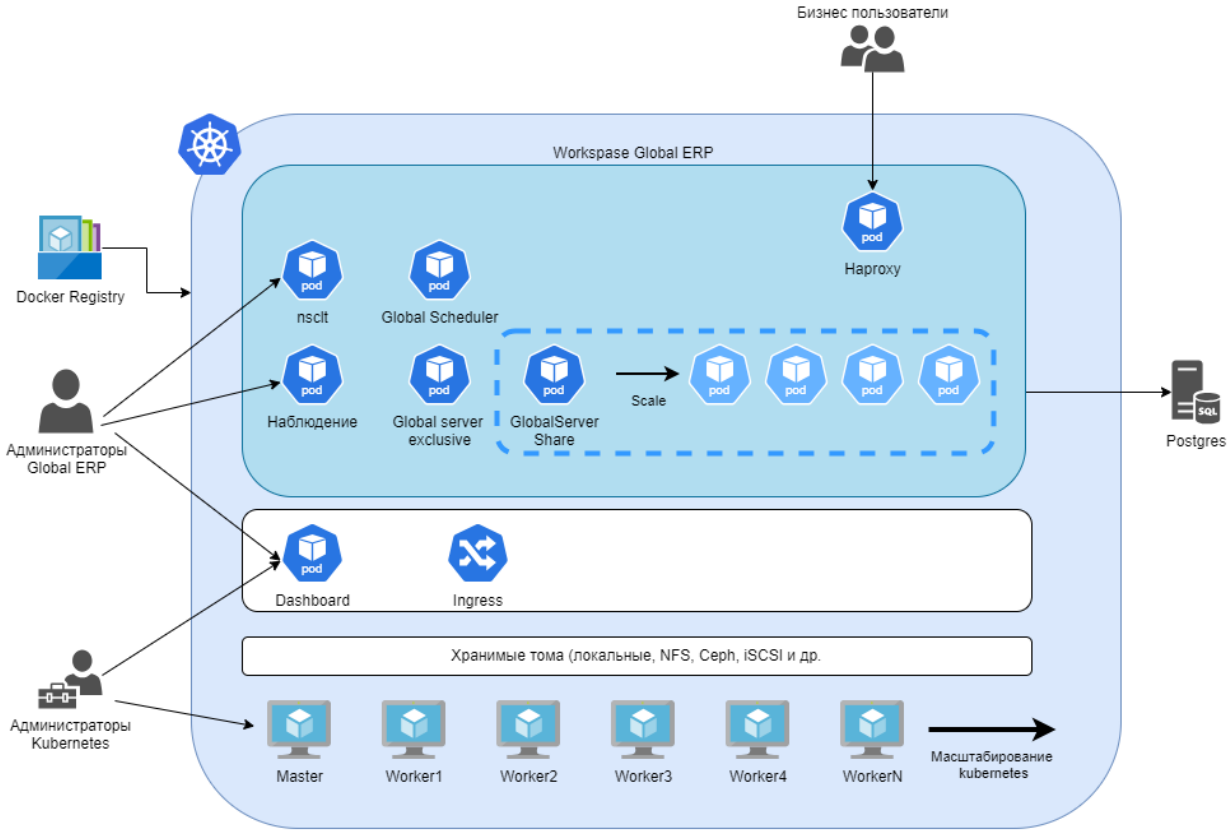
Используется для запуска сервисов системы `Global ERP` в режиме высокой доступности.

Данная платформа предоставляет:

- Сетевую инфраструктуру
  Выделения ip адресов и их доступность между сетями
- Управление дисковыми пространством
- Гибкую высокую доступность
  Не обязательно иметь двойное дублирование по оборудованию. При сбоях оборудования ресурсы кластера автоматически перемещаются на другой рабочий узел kubernetes

2 Основные понятия

2.1 Принципиальная схема



Администрирование

- Администраторы kubernetes
  Задачи:
  - Создание и сопровождение кластера kubernetes
  - Организация доступа к kubernetes
- Администраторы `Global ERP` Задачи:
  - Конфигурирование ресурсов для работы системы `Global ERP`

Kubernetes

Подробнее о режиме работы kubernetes в высокой доступности смотрите документацию kubernetes

Global ERP Namespace

В данном пространстве имен работают ресурсы системы `Global ERP`.

Смотрите *namespace*

Требования к узлам кластера в минимальной конфигурации отсутствуют.

Database

Подробнее о режиме работы базы данных в режиме высокой доступности смотрите в документации соответствующего решения.

- patroni

Требования к базе данных

- Синхронная репликация или асинхронная

Примечание: Синхронная репликация требуется в случае:

- необходимости гарантировать отсутствие потерь выполненных транзакций при сбоях

- Авто выбор лидера
- адрес доступа к мастеру
- адрес доступа к реплике

Примечание: Требуется в случае необходимости распределения нагрузки

2.2 Nscli

Nscli - инструмент управления рабочим пространством.

Данный инструмент содержит набор вспомогательных команд для управления рабочим пространством из внешнего окружения.

Ключевые функции:

- Формирование скрипта установки `namespace` (рабочего пространства)
- Загрузка артефактов в рабочее пространство
- Копирование изображений контейнеров

Требования к ос

- linux а именно
  - debian 11 и выше
  - astra linux 1.8
- python 3.9

Требование к железу для запуска nscli

- 2 ядра
- 200 мб оперативной памяти
- 60гб свободного места

Команды

Image

Image - инструмент работы с образами контейнера.

Ключевые функции:

- Копирование образов контейнера между регистрами образов
  Используется для массового переноса образов в изолированную среду.

2.3 Namespace

Пространства имен в `kubernetes` в рамках которого разворачивается система `Global ERP`.

Примечание: Скрипт создания данного рабочего пространства может быть сформирован утилитой *nscli/namespace*

Nsctl

Nsctl - компонент отвечающий за управление ресурсами в пространстве имен. Данный компонент устанавливается в момент создания пространства имен для администрирования кластера `Global ERP`.

Ключевые функции:

- Создание конфигураций развертывания `Global ERP`
- Формирование ресурсов `kubernetes` по заданной конфигурации.

Appkit

Комплект приложений определяет перечень артефактов, необходимых для разворачивания системы `Global ERP`:

- Сервер приложений `Global ERP`
- Образ прикладного решения
- Набор конфигураций для сервисов кластера

Sysvolume

Системный хранимый том определяет местоположение системных файлов для рабочего пространства. В данный том загружаются разные версии комплектов приложений для развертывания в кластере. Системный том конфигурируется при создании пространства имен.

Appvolume

Прикладной хранимый том определяет местоположение хранимых файлов для работы компонентов `Global ERP`.

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

Resgroup

Группа ресурсов задает общие настройки для ресурсов kubernetes в рамках одного комплекта приложений.

Ключевые настройки:

- Комплект приложений
  Задает путь для текущего `appkit`, а так же контрольные суммы для поддержки целостности.
- Настройки прикладного тома
  Данные настройки применяются при создании подов

Resbook

Книга ресурсов является ключевым элементом настройки правил разворачивания ресурсов kubernetes.

На данном уровне задаются:

- `Тип книги ресурсов`
  Определяет общий перечень ресурсов, который может быть развернут
- `Значения`
  Определяют непереносимые при копировании настройки ресурсов. Таких как: реквизиты учетных данных, количество экземпляров.
- `Характеристики`
  Определяют переносимые при копировании настройки ресурсов. Таких как требуемый объем диска, cpu.

- `Образ контейнера`
  Позволяет переопределять образ контейнера при развертывании
- `Шаблон развертывания`
  Позволяет переопределить шаблон формирования ресурсов kubernetes

Подробнее смотрите *перечень книг ресурсов*.

2.4 Алгоритм развертывания

1. Установить `Nscli`
2. Сформировать скрипт установки `namespace`
3. Сформировать комплект приложений
4. Выполнить в kubernetes манифест создания пространства имен
5. Создать необходимый перечень `Resbook` и сконфигурировать его.
6. Включить конфигурацию

При этом произойдет развертывание ресурсов `kubernetes`, которые обеспечат работу комплекса `Global Erp` в необходимой конфигурации.

2.5 Версии

Комплект имеет семантическую систему версионирования. При этом на каждую минорную версию выпускается новый комплект образов.

3 Установка Nscli

1. Скачайте дистрибутив
2. Распакуйте архив
3. Запустите `bin/install.sh`

4 Установка пространства имен

1. Зайдите в папку nscli
2. Запустите `./namespace.sh configure`
3. Ответьте на задаваемые мастером вопросы
4. Получите kubeconfig от администраторов `kubernetes`
5. Запустите `./namespace.sh install` или запустите скрипт `workspace/install_scripts/<namespace>/install.sh`
6. Подключитесь к `nsctl` в `nscli`
   Для этого выполните `./namespace.sh shell`

Внимание: При работе `kubernetes` в изолированном режиме, необходимо передать комплект образов для загрузки в изолированный docker регистр.

4.1 Загрузка образов для работы в изолированном режиме

1. Зайдите в папку `nscli`
2. Запустите `./image.sh pull_all`
   Файлы будут загружены из публичного `docker регистра` в `workspace/images`
3. Передайте полученные файлы `администраторам kubernetes`

5 Установка комплекта приложений в кластер

1. Зайдите в папку `nscli`
2. Загрузите комплект приложений в рабочее пространство

```
./appkit.sh push --namespace [имя_рабочего_пространства_] --source workspace/appkit␣
↪--destination appkits/v1
```

3. Зайдите в консоль nsctl
   В консоль можно зайти на `dashboard kubernets` или выполнив команду `kubectl -n [имя_рабочего_пространства_] exec -it [имя_пода_nsctl] -- /bin/bash`
4. Создайте набор книг ресурсов

```
./resgroup.sh create --name test
./resgroup.sh switch_appkit --name test --path appkits/v1
./resbook.sh create --name global-server-excl --group test --class_name global_
↪server_excl
./resbook.sh create --name global-server-share --group test --class_name global_
↪server_share
./resbook.sh create --name haproxy --group test --class_name haproxy
```

5. Запустите мастер конфигурации

```
./resgroup.sh init --name test
```

6. Ответьте на вопросы мастера
7. Включите конфигурацию

```
./resgroup.sh start_appkit --name test
./resbook.sh enable_all --group test
./resgroup.sh enable --name test
```

6 Экспорт конфигурации

Скрипт `configmgr.sh` в составе nscli предоставляет возможность экспорта и импорта конфигурации в yaml-файл на jump-хосте, что может быть полезно, если по какой-либо причине кластер потребовалось установить заново.

Такой файл содержит лишь значения, указанные пользователем при помощи nsctl, например, объемы вычислительных мощностей, настройки мониторинга и базы данных. Секреты, комплект приложений, постоянные тома не копируются, однако *названия* используемых секретов и постоянных томов экспортируются.

6.1 Экспорт

Команда:

```
cd ~/nscli
./configmgr.sh export \
  --namespace gs-k8s \
  --group gs-cluster \
  --file config.yaml
```

Вывод:

```
Конфигурация сохранена
```

Пример файла:

```
name: gs-cluster-1
spec:
  appkit:
    path: |-
      appkits/v1
    applib_sha1: |-
      a4078dd40eb0f2a3a227049518b919ea86cf2522
    globalserver_sha1: |-
      3d981e8dc5a45861af06c30dc14b0a0660206b58
    profile_sha1: |-
      0fdb74ca8cab0e5f26488586c80492707b3e99a6
    globalserver_instance: |-
      5
    state: |-
      started
values:
  pod_timezone: |-
    Europe/Moscow
  database_url: |-
    jdbc:postgresql://dbms:5432/db
  database_alias: |-
    pgtest
  db_user_secret_name: |-
    db-user-secret
  app_volume:
    type: |-
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      nfs
    spec:
      server: |-
        nfs
      path: |-
        /srv/nfs
resbooks:
  global-scheduler:
    class_name: global_scheduler
    spec:
      is_metric_enabled: |-
        true
      send_metrics_to_external_system: |-
        false
      globalscheduler_xmx: |-
        800M
      containers:
        globalscheduler:
          resources:
            requests:
              memory: |-
                1G
              cpu: |-
                1
            limits:
              memory: |-
                1G
              cpu: |-
                1
        systemagent:
          resources:
            requests:
              memory: |-
                250M
              cpu: |-
                1
            limits:
              memory: |-
                500M
              cpu: |-
                1
    values:
      prometheus_endpoint: |-
        gs-cluster-1-grafana-internal:9090
      loki_endpoint: |-
        gs-cluster-1-grafana-internal:3100
      tempo_endpoint: |-
        gs-cluster-1-grafana-internal:3201
      scheduler_token_secret: |-
        scheduler-token-secret
    is_deploy_templ_overridden: 'false'
  global-server-excl:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    class_name: global_server_excl
    spec:
      is_metric_enabled: |-
        true
      send_metrics_to_external_system: |-
        false
      globalserver_xmx: |-
        3500M
      containers:
        globalserver:
          resources:
            requests:
              memory: |-
                4G
              cpu: |-
                2
            limits:
              memory: |-
                4G
              cpu: |-
                2
        systemagent:
          resources:
            requests:
              memory: |-
                250M
              cpu: |-
                1
            limits:
              memory: |-
                500M
              cpu: |-
                1
      is_debug_enabled: |-
        true
    values:
      prometheus_endpoint: |-
        gs-cluster-1-grafana-internal:9090
      loki_endpoint: |-
        gs-cluster-1-grafana-internal:3100
      tempo_endpoint: |-
        gs-cluster-1-grafana-internal:3201
      external_ip: |-
        10.40.1.100
      admin_user_secret: |-
        gs-admin-auth
      admin_user_secret_type: |-
        kubernetes.io/basic-auth
    is_deploy_templ_overridden: 'false'
global-server-share:
    class_name: global_server_share
    spec:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    is_metric_enabled: |-
      true
    send_metrics_to_external_system: |-
      false
    globalserver_xmx: |-
      3500M
    containers:
      globalserver:
        resources:
          requests:
            memory: |-
              4G
            cpu: |-
              2
          limits:
            memory: |-
              4G
            cpu: |-
              2
      systemagent:
        resources:
          requests:
            memory: |-
              250M
            cpu: |-
              1
          limits:
            memory: |-
              500M
            cpu: |-
              1
  values:
    replicas: |-
      1
    prometheus_endpoint: |-
      gs-cluster-1-grafana-internal:9090
    loki_endpoint: |-
      gs-cluster-1-grafana-internal:3100
    tempo_endpoint: |-
      gs-cluster-1-grafana-internal:3201
    admin_user_secret: |-
      gs-admin-auth
    admin_user_secret_type: |-
      kubernetes.io/basic-auth
  is_deploy_templ_overridden: 'false'
grafana:
  class_name: grafana
  spec:
    containers:
      grafana:
        resources:
          requests:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            memory: |-
              500M
            cpu: |-
              1
          limits:
            memory: |-
              1Gi
            cpu: |-
              2
  values:
    external_ip: |-
      10.40.1.100
    storage_class: |-
      nfs-storage
    storage_size: |-
      1Gi
    cadvisor_service_account: |-
      prometheus-cadvisor
  is_deploy_templ_overridden: 'false'
haproxy:
  class_name: haproxy
  spec:
    containers:
      haproxy:
        resources:
          requests:
            memory: |-
              500M
            cpu: |-
              1
          limits:
            memory: |-
              1G
            cpu: |-
              2
  values:
    external_ip: |-
      10.40.1.100
    external_port: |-
      8080
    stat_secret: |-
      secret-haproxy-auth
    tls_secret: |
  is_deploy_templ_overridden: 'false'
```

6.2 Импорт

Команда:

```
./namespace.sh create_install_scripts
chmod +x ~/nscli/workspace/install_scripts/gs-cluster-k8s/install.sh
~/nscli/workspace/install_scripts/gs-cluster-k8s/install.sh
```

Вывод:

```
Конфигурация импортирована
```

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 пересборки и перегрузки 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.

1. Зайдите в контейнер `nsctl`
2. Остановите книгу ресурсов

   ```
   ./resbook.sh disable --name $name --group $group
   ```

3. Переключите книгу в режим отладки

   ```
   ./resbook.sh enable_debug --name $name --group $group
   ```

4. Включите книгу

```
./resbook.sh enable --name $name --group $group
```

Примечание: В режиме отладки контейнер не останавливается в случаи ошибок

5. Зайдите в консоль контейнера и произведите анализ ситуации
6. Остановите книгу ресурсов и отключите режим отладки

```
./resbook.sh disable --name $name --group $group
./resbook.sh disable_debug --name $name --group $group
```

8 Приложения

8.1 Перечень книг ресурсов

Global server exclusive

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.

Конфигурационные значения

external_ip

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

Global server share

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

Конфигурационные значения

replicas

Количество реплик, которе должно быть запущенно в kubernetes.

HAProxy

Балансировщик нагрузки.

Конфигурационные значения

external_ip

Если задан, по данному ip включается внешний доступ к haproxy.

stat_secret

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

Если не задано, статистика не включается.

tls_secret

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

Если не задано, https не включается.

8.2 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
    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

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
```

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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        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
```

(continues on next page)

(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]

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
```

Управление комплектом приложения

Содержит общие команды для загрузки\выгрузки комплекта приложения

Commands:

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

positional arguments:
  cmd         Команды
    full_help
                Распечатать справку
    push      Загрузить комплект сборки в общее хранилище

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

Full_help

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

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

Push

```
usage: groupkit.py push [-h] --namespace NAMESPACE --source SOURCE
                        --destination DESTINATION

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
```

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
  --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
    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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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]

Создает управляемое пространство

optional arguments:
  -h, --help  show this help message and exit
  -y          Автоподтверждение установленной конфигурации
```

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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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
```

8.3 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]

Импорт конфигурации

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
```

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
    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:
  -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
  --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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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

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 [--force | --no-force]

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
  --force, --no-force  Сконфигурировать принудительно, даже если группа уже
                       развернута
```

Init_spec

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

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
  --force, --no-force  Сконфигурировать принудительно, даже если группа уже
                       развернута
```

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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        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
```

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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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
```