

# СОЗДАНИЕ, УПРАВЛЕНИЕ И ИСПОЛНЕНИЕ ГРУППОВЫХ ОПЕРАЦИЙ

Руководство по установке

Версия 10.4.0

nexign

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с АО «Нэксайн». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия АО «Нэксайн» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в АО «Нэксайн» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы АО «Нэксайн». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в АО «Нэксайн».

Все имущественные авторские права сохраняются за АО «Нэксайн» в соответствии с действующим законодательством.

© АО «Нэксайн», 1992-2023

АО «Нэксайн»

Россия, 199155, Санкт-Петербург, ул. Уральская, д.4 лит.Б, помещение 22Н

Тел.: + 7 (812) 326-12-99; факс: + 7 (812) 326-12-98.

[office@nexign-systems.com](mailto:office@nexign-systems.com); [www.nexign-systems.com](http://www.nexign-systems.com)

# Содержание

<b>1. Предварительные условия</b>	<b>4</b>
<b>2. Настройка базы данных</b>	<b>5</b>
2.1. База данных Oracle	5
2.2. База данных PostgreSQL	6
<b>3. Установка и настройка с помощью автоинсталлятора</b>	<b>7</b>
3.1. Создание схемы развертывания	7
3.2. Настройки общих параметров установки	7
3.3. Файлы playbook	8
3.4. Настройка хостов	8
3.5. Запуск автоинсталлятора	9
3.6. Установка продуктов из локального репозитория	9
3.7. Принятие/«финализация» (вступление в силу новой версии)	10
3.8. Откат продукта на предыдущую версию	11
3.9. Глобальные настройки продукта	11
3.10. Конфигурация продукта	12
3.10.1. Настройки в файле all.yml	12
3.10.2. Настройки в файле all_backends.yml	12
3.10.3. Настройки в файле bulk_operations_backend_1.yml	13
3.10.4. Настройки в файле bulk_operations_backend_2.yml	13
3.10.5. Настройки в файле bulk_operations_runner_1.yml	13
3.10.6. Настройки в файле bulk_operations_runner_2.yml	14
3.10.7. Настройки в файле bulk_operations_db_oracle_has_bis.yml	14
3.10.8. Настройки в файле bulk_operations_db_oracle_has_crm.yml	14
3.10.9. Настройки в файле tomcat_configs.yml	14
3.10.10. Настройки в файле bulk_runner_zk_config.yml	15
3.10.11. Настройки в файле operations_backend_zk_config.yml	20
3.10.12. Настройки в файле oracle_db.yml	21
3.10.13. Настройки в файле postgresql_db.yml	22
3.10.14. Файл для проверки корректности заполнения inventory	23
3.10.15. Теги	23
3.10.16. Окружение ANSIBLE	23
3.11. Обновление продукта	24
3.12. Настройка конфигурации DLM	26
<b>4. Запуск и остановка продукта</b>	<b>28</b>
<b>5. Проверка работоспособности</b>	<b>29</b>

# 1. Предварительные условия

Перед установкой BULK\_OPERATIONS необходимо проверить, что уже установлены и запущены:

- «OpenAPI ESB» (OPENAPI\_ESB);
- «Аутентификация пользователей по технологии единого входа» (SSO);
- «Система оркестрации» (CRAB);
- «Система обмена сообщениями» (RABBITMQ);
- «Tomcat» (TOMCAT);
- «Централизованный сервис хранения конфигурации и распределенной синхронизации» (ZOOKEEPER);
- "Автоматизированная система расчетов «Nexign BIS»" (BIS);
- «Биллинг, финансы и управление счетами» (BFAM);
- «Управление картотекой клиентов» (CAM);
- «Справочник адресной информации» (AIR);
- «Управление пользовательскими отчетами» (CDM);
- «Управление сервисами обслуживания клиентов» (CCM);
- «Управление лояльностью, маркетинговыми кампаниями и компенсационными программами» (LCCM);
- «Контроль личной информации» (PIC);
- «Высокопроизводительный сервер приложений» (HAS).

Установка продукта выполняется [автоматически](#) при помощи скриптов Ansible.

Требования к окружению:

- сервер Linux RHEL7 x64 с установленным и настроенным Oracle Client;
- OpenJDK 8;
- пакеты Ansible версии 2.4.2 и выше;
- sshpass;
- Python.

## 2. Настройка базы данных

Для работы с продуктом BULK\_OPERATIONS необходимо выполнить ряд операций по настройке базы данных.

### 2.1. База данных Oracle

До запуска установки продукта администратор базы данных должен (требования не относятся к случаю использования `schema_name`):

- создать дополнительного пользователя с привилегиями:
  - `grant create session to <доп.пользователь>;`
  - `grant alter session to <доп.пользователь>;`
  - `grant create synonym to <доп.пользователь>;`
  - `grant alter user <доп.пользователь> quota ... <tablespace данных>;`
  - `grant create table to <доп.пользователь>;`
- создать роль с именем `<владелец_схемы_ГО>_OAPI_BULK_ROLE`;  
Префикс роли берется из параметра `db_liquibase.changelog_params.schema_name` файла `oracle_db.yml`.  
Пример создания: ``CREATE ROLE BULK_OAPI_BULK_ROLE NOT IDENTIFIED.`
- назначить созданную роль дополнительному пользователю: `GRANT <владелец_схемы_продукта>_OAPI_BULK_ROLE TO <доп.пользователь>.`

Oracle-пользователю – владельцу схемы продукта BULK\_OPERATIONS (на промышленном сервере это пользователь BULK) необходимо выдать права:

```
GRANT ALTER SESSION to <USER>;
GRANT CREATE CLUSTER to <USER>;
GRANT CREATE JOB to <USER>;
GRANT CREATE PROCEDURE to <USER>;
GRANT CREATE SEQUENCE to <USER>;
GRANT CREATE SESSION to <USER>;
GRANT CREATE TABLE to <USER>;
GRANT CREATE TRIGGER to <USER>;
GRANT CREATE TYPE to <USER>;
GRANT CREATE VIEW to <USER>;
GRANT CREATE SYNONYM to <USER>;
GRANT QUERY REWRITE to <USER>;
GRANT EXECUTE ON "DBMS_AQ" TO <USER>;
GRANT EXECUTE ON "DBMS_AQADM" TO <USER>;
GRANT EXECUTE ON "DBMS_LOCK" TO <USER>;
GRANT SELECT ON "V_$SESSION" TO <USER>;
GRANT SELECT ON "GV_$SESSION" TO <USER>;
```

Также должна быть создана роль `<USER>_OAPI_BULK_ROLE`.

Для корректной работы продукта DLM по очистке устаревших данных необходимо выдать

Oracle-пользователю – владельцу схемы продукта BULK\_OPERATIONS права:

```
GRANT RESOURCE to BULK;  
GRANT EXECUTE ON DBMS_RLS to BULK;  
GRANT EXECUTE ON DBMS_LOCK to BULK;  
GRANT EXECUTE ON DBMS_CRYPTO to BULK;
```

## 2.2. База данных PostgreSQL

До запуска установки продукта администратор базы данных должен:

1. Обеспечить наличие базы данных.
2. Создать пользователя, от имени которого продукт будет подключаться.  
Минимальный набор системных привилегий: LOGIN, NOSUPERUSER, NOCREATEDB, NOCREATEROLE, INHERIT, NOREPLICATION.
3. Выдать созданному пользователю право на подключение к базе данных:  
`GRANT CONNECT ON DATABASE <db> to <user>;`
4. Создать пользователя-владельца объектов.  
В упрощенном случае можно использовать пользователя из пункта 2.
5. Создать схему для размещения объектов.
6. Назначить владельцем схемы пользователя, созданного в пункте 3.  
Если планируется, что пользователь из п. 2 не будет являться владельцем объектов и доступ к объектам предполагается через роль, то данную роль также необходимо создать.
7. Для работы продукта «Управление жизненным циклом данных» (DLM) необходимо в продукте NORD инициализировать `pg_cron`:
  - в настройках NORD в файле `postgresql.conf` заполнить параметр `cron.database_name`;
  - перезагрузить «Система управления объектно-реляционной базой данных Nexign» (NORD);
  - выполнить через PL/SQL команду `CREATE EXTENSION IF NOT EXISTS pg_cron`;

## 3. Установка и настройка с помощью автоинсталлятора



### Примечание.

В данном документе понятия «инсталлятор» и «автоинсталлятор» идентичны.

Для корректной установки продукта выполните следующие действия:

1. Скачайте и разархивируйте инсталлятор из Artifactory.

Файлы инсталлятора разместите в каталог `<product>_install`.

2. Если включена обязательная авторизация для скачивания из Artifactory, необходимо в файл `requirements.yml` добавить строку:  
`src://http://<USER>:<ENCRYPTED_PASSWORD>@<artifactory_url>`, где:
  - `<USER>` – пользователь для входа в Artifactory;
  - `<ENCRYPTED_PASSWORD>` – алгоритм шифрования;
  - `<artifactory_url>` – URL для скачивания архива с общими ролями.
3. Создайте схему развертывания (см. раздел «[Создание схемы развертывания](#)»).
4. Настройте общие параметры (см. раздел «[Настройки общих параметров установки](#)»).
5. Настройте параметры файлов `playbook` (см. раздел «[Файлы playbook](#)»).
6. Настройте хосты (см. раздел «[Настройка хостов](#)»).
7. Выполните конфигурацию продукта (см. раздел «[Конфигурация продукта](#)»).
8. Запустите скрипт `ansible-prepare.sh`. Из Artifactory будет скачан дистрибутив с общими ролями продукта `Common_Installer`. Описание ролей `Common_Installer` приведено в Руководстве по эксплуатации продукта [COMMON\_INSTALLER\_DOC\_G3].
9. Запустите автоинсталлятор (см. раздел «[Запуск автоинсталлятора](#)»).
10. Переключитесь на новую версию вводом команды: `ansible-playbook <product>-deploy-finalize.yml -i inventory/localhost` (см.раздел «[Принятие/«финализация» \(вступление в силу новой версии\)](#)»).
11. Зафиксируйте успешную установку версии (статус новой версии продукта – успешно установлена) вводом команды: `ansible-playbook <product>-deploy-success.yml -i inventory/localhost`.

Подробнее смотрите в документации «Руководство по эксплуатации» [COMMON\_INSTALLER\_DOC\_G3] раздел «Структура директорий продуктов при установке ролями с возможностью бэкап и возврата на предыдущую версию»).

### 3.1. Создание схемы развертывания

В каталоге дистрибутива `install\inventory\localhost` содержится пример с настройками схемы развёртывания.

Чтобы создать собственную схему развёртывания, создайте копию каталога `localhost` и переименуйте её (например, `staging`).

### 3.2. Настройки общих параметров установки

1. Настройте хост-серверы, на которых устанавливается продукт, в файле `inventory/localhost/1-rnd-bulk_operations` (описание параметров приведено в разделе «[Настройка хостов](#)»).

2. Задайте местоположение запускаемых компонентов, файлов журналирования, мониторинга в файле `inventory/localhost/group_vars/all.yml` (описание параметров приведено в разделе «[Конфигурация продукта](#)»).
3. При необходимости переопределите общие настройки установки каждого компонента и настройки конфигурации в каталоге `inventory/localhost/group_vars/<component_name>.yml` (описание параметров приведено в разделе «[Конфигурация продукта](#)»).

Подробнее о настройках inventory см. документацию Ansible [intro\\_inventory.html](#).

### 3.3. Файлы `playbook`

Инсталлятор использует роли продукта `COMMON_INSTALLER` (подробнее об описании общих ролей см. Руководство по эксплуатации [`COMMON_INSTALLER_DOC_G3`]).

Перечень файлов `playbook`:

- `bulk_operations-deploy-full.yml` – общий скрипт установки;
- `bulk_operations-deploy.yml` – установка backend-части на Tomcat и в ZooKeeper;
- `bulk_operations-backend-deploy.yml` – установка `BULK_OPERATIONS_BACKEND`;
- `bulk_operations-runner-deploy.yml` – установка `BULK_RUNNER`;
- `bulk_operations-scenarios-zookeeper-deploy.yml` – конфигурация сценариев ZooKeeper;
- `bulk_operations-deploy-db.yml` – установка серверной части;
- `bulk_operations-validate-parameters.yml` – проверка конфигурационных параметров;
- `bulk_operations-deploy-finalize.yml` – завершение установки и перезапуск продукта;
- `bulk_operations-deploy-success.yml` – отметка текущей версии продукта как успешная;
- `bulk_operations-deploy-rollback.yml` – откат на последнюю успешную версию.

### 3.4. Настройка хостов

В файле `inventory/localhost/1-rnd-bulk_operations` сконфигурируйте группы хостов в зависимости от выбранной СУБД (см. комментарии в файле `localhost/1-rnd-bulk_operations`).

В файле `inventory/localhost/1-rnd-bulk_operations` укажите адреса серверов – `ansible_host`, на которые необходимо установить компоненты продукта, и параметры соединения с ними – `ansible_connection`.

Пример:

```
bulk_operations ansible_connection=ssh ansible_host=&lt;адрес удаленной
машины&gt; # deploy на удаленную машину
# либо
bulk_operations ansible_connection=local ansible_host=localhost #
deploy на localhost
```

Если компоненты продукта устанавливаются на разные Tomcat, то группа `bulk_operations_tomcat` должна включать оба хоста.

Файл `inventory/localhost/1-rnd-bulk_operations` по умолчанию заполнен параметрами в минимальной конфигурации для установки [без простоя](#) (установка не менее чем по два экземпляра для каждого backend'а). Если в целевой схеме развертывания продукта не предусмотрено резервирование backend'ов продукта и устанавливается по одному экземпляру backend'а, необходимо удалить группы `bulk_operations_backend_2` и `bulk_operations_runner_2` и их вхождения в наследование. Удалять другие группы нельзя. Изменять названия существующих групп нельзя. Можно добавлять новые хосты, новые группы

и наследования.

## 3.5. Запуск автоинсталлятора

Автоинсталлятор может запускаться либо вызовом `auto-deploy.sh`, либо вводом команды `ansible-playbook bulk_operations-deploy-full.yml -i inventory/localhost -u <user> -k [-K] [-b] [--become-user <user>] [-t <tags>] [-l <alias_hosts>] [-v]`, где:

- `-i <inventory/localhost>` – inventory, который будет использоваться при выполнении сценария (если указать каталог, то сценарий пройдет по всем файлам хостов, находящихся в указанном каталоге)
- `-u` – пользователь, от имени которого Ansible будет подключаться к серверам, указанным в файле хостов в `inventory`;
- `-k` – необходимость ввода пароля пользователя, указанного в ключе `-u`;
- `-K` – необходимость ввода пароля для привилегированного пользователя, указанного в ключе `-become-user`;
- `-b` – необходимость выполнения сценария от имени привилегированного пользователя;
- `--become-user` – привилегированный пользователь;
- `-t` – перечисление тегов, которые будут выполняться в сценарии (по умолчанию используется `all`, то есть запуск всех тегов);
- `-l` – перечисление хостов или групп, по которым будет выполнен сценарий (по умолчанию сценарий будет выполнен для всех хостов);
- `-v` – уровень журналирования; максимальное количество `v = 4 (-vvvv)`, чем больше `v`, тем выше уровень журналирования.

Примеры запуска:

- `ansible-playbook bulk_operations-deploy-full.yml -i inventory/localhost -e "bulk_operations_version=8.2.0ansible_ssh_pass=пароль_пользователя ansible_ssh_user=имя_пользователя" bulk_operations-deploy-full.yml`

В ходе запуска `playbook` произойдут:

- валидация конфигурационных параметров;
  - конфигурирование узлов в ZooKeeper;
  - создание версии приложения;
  - установка backend-части BULK\_OPERATIONS-BACKEND;
  - установка backend-части BULK\_OPERATIONS-RUNNER;
  - установка серверной части BULK\_OPERATIONS.
- `ansible-playbook bulk_operations-deploy-full.yml -i inventory/localhost -e "bulk_operations_version=8.2.0ansible_ssh_pass=пароль_пользователя ansible_ssh_user=имя_пользователя" bulk_operations-deploy-full.yml --tags="configure"`

## 3.6. Установка продуктов из локального репозитория

Для установки продукта из локального репозитория измените следующие параметры:

- в `inventory/localhost/group_vars/all.yml`:
  - уберите/закомментируйте переменные `artifactory.*`, либо укажите `artifactory.enable: false`

```
artifactory:  
enable: false # выключение работы с Artifactory  
url: "https://artifactory.billing.ru/artifactory"  
search_url:  
"https://artifactory.billing.ru/artifactory/api/search/aql"  
api_key: "<API_KEY>"  
repo: "QA"
```

- добавьте параметры для локального репозитория

```
local_storage:  
enable: true # включение работы с локальным архивом  
root: "/home/user/product_distrib"
```

- в каталоге `group_vars` во все конфигурационные файлы добавьте:

```
local_storage:  
path: "*PRODUCT_NAME*COMPONENT_NAME"  
artifactory: # можно не удалять для возможности переключения работы  
на Artifactory  
path: "*PRODUCT_GROUP*PRODUCT_NAME*COMPONENT_NAME"
```

При этом необходимо учитывать, что в корневом каталоге должна соблюдаться такая же структура расположения продукта, как в Artifactory. Структура в корневом каталоге должна начинаться с группы продуктов:

`<группа продуктов>/<продукт>/<версия>/<компонент>/<app|conf|mib|info>`.

Возможны три варианта структуры раскладки артефактов:

- `<группа продуктов>/<продукт>/<компонент>/<app|conf|mib|info>`;
- `<группа продуктов>/<продукт>/<версия>/<компонент>/<app|conf|mib|info>`;
- `<компонент>/<app|conf|mib|info>`.

## 3.7. Принятие/«финализация» (вступление в силу новой версии)

После установки нужно выполнить финализацию – запуск новой версии приложения и переключение `current version` в ZooKeeper.

Используйте скрипт `auto-finalize.sh` или команду

```
ansible-playbook bulk_operations-deploy-finalize.yml -i inventory/localhost  
-e "bulk_operations_version=<номер устанавливаемой версии (например,  
9.2.0)> ansible_ssh_pass=пароль_пользователя  
ansible_ssh_user=имя_пользователя"
```

В процессе финализации будут выполнены:

- остановка текущей версии Tomcat;

- замена ссылки `current` на новую версию;
- запуск Tomcat;
- переключение `current_version` в ZooKeeper.

Доступные теги:

- `bulk_operations_backend`, `bulk_runner`;
- `stop` – остановка;
- `restart` – перезапуск;
- `finalize` – только финализация без остановки приложения (использовать осторожно и только при крайней необходимости)
- `current_version` – переключение узла в ZooKeeper.

Если установленная версия работает корректно, ее можно зафиксировать для возможности отката на нее скриптом `auto-success.sh` или командой

```
ansible-playbook bulk_operations-deploy-success.yml -i inventory/localhost
-e "bulk_operations_version=<номер версии (например, 9.2.0)>
ansible_ssh_pass=пароль_пользователя ansible_ssh_user=имя_пользователя"
```

## 3.8. Откат продукта на предыдущую версию

Откат продукта на предыдущую версию можно запустить скриптом `auto-rollback.sh` или командой

```
ansible-playbook bulk_operations-deploy-rollback.yml -i inventory/localhost
-e "bulk_operations_version=<номер версии (например,
9.2.0)>ansible_ssh_pass=пароль_пользователя
ansible_ssh_user=имя_пользователя"
```

Будет выполнен откат ссылки `current` в Tomcat на последнюю сборку с флагом `success`.

### Примечание.



На данный момент при откате продукта выполняется только замена `war`-файлов в Tomcat на предыдущую версию. ZooKeeper-часть продукта при этом не откатывается, а скрипты групповых операций хранятся в ZooKeeper, поэтому если код скриптов текущей версии совместим с предыдущей версией `war`-файлов, то ничего более делать не требуется. Если же совместимости нет, то необходимо дополнительно установить ZooKeeper-часть предыдущей версии продукта.

## 3.9. Глобальные настройки продукта

В глобальных настройках `group_vars` хранятся настройки, которые не изменяются от стенда к стенду. Например, команды запуска и остановки сервиса, путь к продукту в Artifactory.

Параметры глобальных настроек:

- `war_name` – наименование `war`-файла на сервере;
- `zapp_name` – наименования компонента;
- `current_version_node` – узел в ZooKeeper, в который добавляются данные из шаблона конфигурации (по умолчанию – `{{ zapp_name }}`);
- `stop_command` – команда остановки продукта;

- `start_command` – команда запуска продукта;
- `check_running.pid` – способ проверки успешности запуска продукта по наличию `pid`-файла, в этом параметре указывается его расположение.

Файлы в каталоге `group_vars` не нуждаются в конфигурировании.

## 3.10. Конфигурация продукта

Перед началом установки необходимо отредактировать `inventory`-каталог `localhost`, содержащий в себе файлы:

- `inventory/localhost/group_vars/all.yml`;
- `inventory/localhost/group_vars/bulk_operations_backend.yml`;
- `inventory/localhost/group_vars/bulk_operations_runner.yml`;
- `inventory/localhost/group_vars/bulk_operations_db_oracle_has_bis.yml`;
- `inventory/localhost/group_vars/bulk_operations_db_oracle_has_crm.yml`;

Два отдельных файла `bulk_operations_db_oracle_has_bis.yml` и `bulk_operations_db_oracle_has_crm.yml` используются для обеспечения возможности установки продукта на две схемы `HAS`: основную и резервную.

- `inventory/localhost/group_vars/bulk_operations/tomcat_configs.yml`;
- `inventory/localhost/group_vars/bulk_operations/bulk_runner_zk_config.yml`;
- `inventory/localhost/group_vars/bulk_operations/operations_backend_zk_config.yml`;
- `inventory/localhost/group_vars/bulk_operations/oracle_db.yml`;
- `inventory/localhost/group_vars/bulk_operations/postgresql_db.yml`.

### 3.10.1. Настройки в файле `all.yml`

В файле `inventory/localhost/group_vars/all.yml` задаются параметры:

- `become_user_bulk` – пользователь, от имени которого будут выполняться команды на удаленном хосте;
- `artifactory` – блок настроек Artifactory:
  - `url` – URL-адрес Artifactory;
  - `search_url` – URL к поисковому API Artifactory;
  - `api_key` – ключ для аутентификации пользователя;
  - `repo` – код репозитория, в котором проводится поиск;
- `base_dir` – базовый каталог;
- `logs_dir` – каталог расположения журнальных файлов;
- `mibs_dir` – каталог расположения MIB-файлов;
- `info_dir` – каталог расположения `info.json` файлов;
- `zoo_host` – хост и порт ZooKeeper;
- `secret_key` – ключ для шифрования паролей;
- `tomcat_user` – пользователь GUI Tomcat;
- `tomcat_password` – пароль пользователя GUI Tomcat;
- `tomcat_port` – порт Tomcat.

### 3.10.2. Настройки в файле `all_backends.yml`

В файле `inventory/localhost/group_vars/all_backends.yml` задаются параметры:

- `sso_api` – блок параметров управления установкой ролевой модели в SSO:

- `enable` – признак использования загрузки ролей в SSO;
- `host` – DNS-имя или IP-адрес сервера, где работает API SSO;
- `port` – порт сервера, где работает API SSO;
- `user` – имя специального технологического SSO-пользователя для получения токена на API SSO;

Специальный пользователь, от имени которого происходит регистрация ролей. Пользователя создают администраторы клиента по заявке со стороны продукта.

- `password` – пароль технологического пользователя для получения токена на API SSO;
- `appl_user` – имя технического пользователя для авторизации при получении токена на API SSO;

Специальный пользователь SSO с типом `application` (приложение), который используется для получения токена технологического пользователя. От имени этого пользователя работает `BULK_OPERATIONS`.

- `appl_password` – пароль технического пользователя для авторизации при получении токена на API SSO;
- `nginx` – блок реквизитов доступа к балансировщику Nginx:
  - `nginx_host` – DNS-имя или IP-адрес хоста, на котором работает Nginx;
  - `nginx_port` – порт, на котором Nginx принимает запросы;
  - `user` – логин для аутентификации в Nginx;
  - `password` – пароль для аутентификации в Nginx;
  - `need_zoo_unlock` – флаг выполнения выполнения запроса `{{ nginx }}/upstream/zoo/unlock`.

### 3.10.3. Настройки в файле `bulk_operations_backend_1.yml`

В файле `inventory/localhost/group_vars/bulk_operations_backend_1.yml` задаются базовые конфигурационные параметры установки компонента `bulk_operations_backend`:

- `app.install_dir` – путь до корневого каталога приложения компонента;
- `sso_api` – блок параметров управления установкой ролевой модели в SSO:
  - `enable` – признак использования загрузки ролей в SSO (полная ролевая модель).

Если этот параметр заполнен в файле `all_backends.yml`, значение наследуется оттуда. Если такое значение подходит, параметр в `bulk_operations_backend1.yml` можно не указывать.

### 3.10.4. Настройки в файле `bulk_operations_backend_2.yml`

Файл `bulk_operations_backend_2.yml` создается и используется только при необходимости выполнения [обновления без простоя](#). Параметры совпадают с параметрами в файле `bulk_operations_backend_1.yml`.

### 3.10.5. Настройки в файле `bulk_operations_runner_1.yml`

В файле `inventory/localhost/group_vars/bulk_operations_runner_1.yml` задаются базовые конфигурационные параметры установки компонента `bulk_runner`:

- `app.install_dir` – путь до корневого каталога приложения компонента;
- `need_copy_tomcat_db_libs` – необходимость установки в папку `libs` Tomcat-библиотек для

работы с базой данных (ojdbc6.jar, orai18n.jar);

- `sso_api` – блок параметров управления установкой ролевой модели в SSO:
  - `enable` – признак использования загрузки ролей в SSO (только core-часть ролевой модели).

### 3.10.6. Настройки в файле `bulk_operations_runner_2.yml`

Файл `bulk_operations_runner_2.yml` создается и используется только при необходимости выполнения [обновления без простоя](#). Параметры совпадают с параметрами в файле `bulk_operations_runner_1.yml`.

### 3.10.7. Настройки в файле `bulk_operations_db_oracle_has_bis.yml`

В файле `inventory/localhost/group_vars/bulk_operations_db_oracle_has_bis.yml` задаются параметры подключения к базе данных для установки серверной части продукта на резервную схему HAS. По умолчанию используются данные из файла `inventory/localhost/group_vars/ccm_portal/oracle_db.yml`.

- `db_sqlplus`: – блок конфигурирования параметров подключения к базе данных:
  - `user` – имя схемы HAS;
  - `password` – пароль для схемы HAS;
  - `database` – наименование базы данных со схемой HAS.

### 3.10.8. Настройки в файле `bulk_operations_db_oracle_has_crm.yml`

В файле `inventory/localhost/group_vars/bulk_operations_db_oracle_has_bis.yml` задаются базовые конфигурационные параметры подключения к базе данных для установки серверной части продукта на основную схему HAS. По умолчанию используются данные из файла `inventory/localhost/group_vars/ccm_portal/oracle_db.yml`.

- `db_sqlplus`: – блок конфигурирования параметров подключения к базе данных:
  - `user` – имя схемы HAS;
  - `password` – пароль для схемы HAS;
  - `database` – наименование базы данных со схемой HAS.

### 3.10.9. Настройки в файле `tomcat_configs.yml`

В файле `inventory/localhost/group_vars/bulk_operations/tomcat_configs.yml` задаются базовые конфигурационные параметры приложения Tomcat:

- `zookeeper`: – блок конфигурирования параметров подключения к ZooKeeper:
  - `host` – имя сервера ZooKeeper;
  - `port` – порт подключения к ZooKeeper;
- `bulk_runner`: – настройка пула параллельных потоков для `bulk_runner`:
  - `parallelismFactor` – стандартное число потоков;
  - `parallelismMax` – максимально возможное число потоков;
- `amqp`: – блок конфигурирования параметров очередей в RabbitMQ для получения callback-сообщений с результатами выполнения асинхронных OAPI-функций:
  - `routingKey` – наименование ключа маршрутизации для конфигурирования очереди и публикации сообщения (ключ должен быть уникальным среди всех ключей маршрутизации экземпляров `oapi_bulk_runner`);

Если параметр не задан, по умолчанию наименование будет создано по шаблону

```
ps.bulk_operations.%IP_HOST_TOMCAT_BULK_RUNNER%_%IP_PORT_TOMCAT_BULK_RUNNER%.
```

- `queueName` – наименование очереди (должно быть уникальным среди всех очередей экземпляров `oapi_bulk_runner`);

Если не задано в параметре, то по умолчанию наименование будет создано по шаблону `ps.bulk_operations.notifications.%IP_HOST_TOMCAT_BULK_RUNNER%_%IP_PORT_TOMCAT_BULK_RUNNER%`:

- `IP_HOST_TOMCAT_BULK_RUNNER` – IP-адрес хоста Tomcat, где запущен war-файл `oapi_bulk_runner`;
- `IP_PORT_TOMCAT_BULK_RUNNER` – порт Tomcat, куда установлен war-файл `oapi_bulk_runner`.
- `log`: – блок параметров журналирования:
  - `level` – уровень журналирования;
  - `humanReadable`: – блок параметров для файлов `.log`:
    - `enabled` – признак включения данного типа журналирования;
    - `rollingPolicy`: – блок параметров ротации журналирования:
      - `pattern` – шаблон даты для ротации (от этого параметра зависят частота ротации и имена архивов);
      - `maxHistory` – количество дней для хранения архивов с журналами работы;
      - `maxFileSize` – максимальный объем файла журнала работы, после превышения которого файл будет заархивирован;
- `jsonl`: – блок параметров для файлов `.jsonl`:
  - `enabled` – признак включения данного типа журналирования;
  - `rollingPolicy`: – блок параметров ротации журналирования:
    - `pattern` – шаблон даты для ротации (от этого параметра зависят частота ротации и имена архивов);
    - `maxHistory` – количество дней для хранения архивов с журналами работы;
    - `maxFileSize` – максимальный объем файла журнала работы, после превышения которого файл будет заархивирован.

### 3.10.10. Настройки в файле `bulk_runner_zk_config.yml`

В файле `inventory/localhost/group_vars/bulk_operations/bulk_runner_zk_config.yml` задаются конфигурационные параметры `BULK_RUNNER` для ZooKeeper:

- `common`: – блок стандартных параметров:
  - `asyncScenarioRetryCheckIntervalSeconds` – интервал сканирования асинхронных сценариев для их перезапуска;
  - `asyncScenarioRetryTimeSeconds` – время (с), через которое асинхронный шаг сценария будет перезапущен, если за это время не поступит оповещение о его выполнении;
  - `defaultLocale` – язык локализации сообщений движка (`ru` – русский, `en` – английский);
  - `executeReadSize` – размер порции чтения данных выполнения;
  - `loadDataReadSize` – размер порции чтения данных загрузки для формирования данных выполнения;
  - `recoveryCallbackMaxWaitTimeSeconds` – время в секундах ожидания обратного вызова во время продолжения работы, после останова которого сценарий выполняется заново;

- `scenarioBatchPackTime` – время (мс) сбора пачки сценариев;
- `scenarioBatchSize` – размер пачки выполнения сценариев;
- `defaultExchange` – имя обменника по умолчанию для нотификации;
- `enableRedistributeOperations` – включить/отключить перераспределение задач с отказавших экземпляров `BULK_RUNNER`;
- `retryTimeoutBeforeRedistribute` – интервал (с) между проверками, не вернулся ли в строй отключившийся экземпляр `OAPI_BULK_RUNNER`;
- `retryCountBeforeRedistribute` – количество проверок, не вернулся ли в строй отключившийся экземпляр `OAPI_BULK_RUNNER`, перед перераспределением задач;
- `actorSystemCtoptTimeout` – время ожидания (с) приостановки обработки задач или их завершения движком групповых операций при остановке приложения;
- `dbTimeZone` – временная зона для работы с датами в базе данных;
- `requestCheckingBatchSize` – размер порции данных для выполнения проверок сущностей;
- `requestFinalProcessingBatchSize` – размер порции данных для выполнения процесса финализации задачи;
- `crab` – блок параметров работы с продуктом CRAB:
  - `server` – сервер и корневой каталог развертывания CRAB;
  - `connectionTimeout` – максимальное время в мс установки соединения с CRAB;
  - `executeTimeout` – максимальное время в мс выполнения запроса в CRAB;
- `bus` – блок параметров работы с RabbitMQ:
  - `host` – имя хоста RabbitMQ-нотификаций;
  - `port` – порт RabbitMQ;
  - `user` – имя пользователя;
  - `password` – пароль;
  - `connectionTimeout` – время (мс) для соединения с RabbitMQ;
  - `vhost` – имя виртуального хоста (должен быть прописан виртуальный хост CRAB);
  - `exchange` – имя обменника для получения callback-ов о выполнении асинхронных oapi-функций (по умолчанию – `ps.Mcg_om_notify`);
  - `handshakeTimeout` – время ожидания ответа;
  - `prefetchCount` – число сообщений, выбираемых из RabbitMQ за одно обращение;
  - `reconnectWaitTime` – время между попытками подключения к RabbitMQ (мс);
- `db` – блок параметров базы данных:
  - `driverClassName` – драйвер базы данных;
  - `initSQL` – запрос инициализации и проверки наличия соединения с базой данных;
  - `initialSize` – начальный размер пула соединений;
  - `maxActive` – максимальный размер пула соединений;
  - `minIdle` – минимальное количество неиспользуемых соединений;
  - `maxIdle` – максимальное количество неиспользуемых соединений;
  - `maxWait` – максимальное время в мс ожидания свободного соединения из пула;
  - `queryTimeout` – максимальное время выполнения запроса в мс или 0;
  - `removeAbandonedTimeout` – время в секундах, через которое зависшие запросы будут удалены из пула;
  - `password` – зашифрованный пароль для схемы `BULK`;
  - `url` – JDBC URL соединения со схемой `BULK`;
  - `username` – имя схемы `BULK`;

- `logAbandoned` – признак журналирования зависших соединений при их получении;
- `removeAbandoned` – признак удаления зависших соединений из пула
- `validationInterval` – интервал между запросами на валидацию соединения (мс);
- `validationQuery` – запрос для проверки соединения;
- `testOnBorrow` – признак валидации соединения при взятии из пула;
- `additional:` – блок настроек пользовательских соединений к базе данных:
  - `driverClassName` – драйвер базы данных;
  - `url` – JDBC URL соединения с базой данных;
  - `username` – имя пользователя базы данных;
  - `password` – зашифрованный пароль доступа к базе данных;
  - `minIdle` – минимальное количество неиспользуемых соединений;
  - `maxIdle` – максимальное количество неиспользуемых соединений;
  - `maxActive` – максимальный размер пула соединений;
  - `queryTimeout` – максимальное время выполнения запроса (мс);
  - `connectionProperties` – дополнительные настройки соединения;
  - `logAbandoned` – признак журналирования зависших соединений при их получении;
  - `removeAbandoned` – признак удаления зависших соединений из пула
  - `validationInterval` – интервал между запросами на валидацию соединения (мс);
  - `validationQuery` – запрос для проверки соединения;
  - `testOnBorrow` – признак валидации соединения при взятии из пула;
- `logging:` – блок настроек для отладочной информации:
  - `logAkkaMessages` – добавление сообщений в журнальные файлы;
  - `storeWaitingMessages` – журналировать сообщения ожидания;
  - `oapi:` – блок настроек взаимодействия с OAPI:
    - `oapiRequestBodyLogLength` – количество символов тела запроса к OAPI, которые записываются в журнальный файл;
    - `oapiResponseBodyLogLength` – количество символов тела ответа от OAPI, которые записываются в журнальный файл;
  - `notification:` – блок настроек уведомлений:
    - `collect` – добавлять в журналы;
    - `maxSize` – максимальный размер сообщения;
- `extensionPoints:` – настройки для задания точек расширения:
  - `beforeExecutionScenariosScript:` – настройки точки расширения для автоматического задания размера пачки выполнения (запускается перед выполнением скриптов операции – один раз на каждую операцию):
    - `enabled` – признак включения запуска точки расширения в рамках выполнения групповой операции;
    - `script` – код скрипта точки расширения;
- `find:` – блок настроек поиска клиентов:
  - `batchSize` – размер пачки поиска клиентов/абонентов (для запросов, поддерживающих пакетный режим);
  - `maxCollectBatchDuration` – максимальное время (мс) объединения единичных запросов в пакет;
- `notification:` – блок параметров для нотификации о завершении выполнения задачи:
  - `defaultExchange` – имя обменника по умолчанию для нотификации о завершении выполнения задачи;
  - `bus:` – блок параметров работы с RabbitMQ;

- `host` – имя хоста RabbitMQ нотификаций;
- `port` – порт RabbitMQ;
- `user` – имя пользователя;
- `password` – пароль;
- `connectionTimeout` – время в миллисекундах для соединения с RabbitMQ;
- `vhost` – имя виртуального хоста (должен быть прописан виртуальный хост CRAB);
- `handshakeTimeout` – время ожидания ответа;
- `oapi`: – блок параметров OAPI:
  - `connectionTimeout` – максимальное время в Мс установки соединения с OAPI;
  - `executeTimeout` – максимальное время в Мс выполнения запроса в OAPI;
  - `server` – сервер и порт OAPI;
  - `oapiService` – Groovy-скрипт, позволяющий переопределить логику работы с входным списком клиентов/абонентов;
  - `searchLimit` – ограничение на количество абонентов клиента, которых возвращает функция `/openapi/v1/customers/{customerId}/subscribers`;
  - `auth`: – блок параметров авторизации:
    - `appl_user` – пользователь в SSO с типом «Приложение»;
    - `appl_password` – пароль пользователя в SSO с типом «Приложение».
    - `password` – пароль доступа к OAPI;
    - `user` – имя пользователя для доступа к OAPI;
    - `additionalTechUsers` – список дополнительных технологических пользователей;
    - `useTokenStubForAuth` – выбор функции авторизации;
- `retry`: – настройка количества повторов:
  - `oapi` – число повторных операций к OAPI (в настоящий момент – повторов поисковых запросов);
  - `scenario` – число повторов выполнения сценариев;
  - `scenarioForCallback` – число повторов выполнения сценариев с целью получения нотификации (в терминах OAPI – `callback`);
- `schedulers`: – настройки планировщиков:
  - `newBulkRequestsGetter`: – настройки выборки новых задач для передачи на исполнение:
    - `scanInterval` – время в Мс сканирования новых задач;
  - `standHealthCheck`: – настройки мониторинга состояния площадки с backend'ами:
    - `enabled` – флаг включения функциональности мониторинга;
    - `script` – код скрипта, реализующий логику получения общего состояния площадки;
    - `execInterval` – интервал времени (с) между вызовом скрипта;
    - `standIsOk` – результат выполнения скрипта мониторинга состояния площадки;
    - `maxBulkRequestWeight` – порог веса задачи;
- `worker`: – блок параметров выполнения операций:
  - `findActorCount` – количество обработчиков поиска;
  - `readActorCount` – количество обработчиков чтения данных обработки;
  - `runDataActorCount` – количество обработчиков чтения данных запуска;
  - `runInformationActorCount` – количество обработчиков получения данных о запусках;
  - `runUpdateDatesActorCount` – количество обработчиков изменения состояния запуска;
  - `scenarioWorkerCount` – количество обработчиков выполнения внутреннего сценария;
  - `scenarioActorForCallbackCount` – количество акторов для обработки `callback`-

сообщений от асинхронных функций;

- `dedicatedScenarioWorkerPools`: – параметры для задания размера выделенных пулов `ScenarioActor`’ов для типов операций;

Пример:

```
CloseClients:
"\{\{p.bulk_runner.worker.dedicatedScenarioWorkerPools.CloseClients
| default('15')\}\}"

PersonalInformationControlCheck:
"\{\{p.bulk_runner.worker.dedicatedScenarioWorkerPools.PersonalInfo
rmationControlCheck | default('20')\}\}"
```

- `switches`: – блок функциональных переключателей:
  - `commonDistributeChargesRulesEnabled` – рычаг включения возможности назначения общего набора лимитов на абонентов;
  - `productOfferingStage2Enabled` – рычаг включения доработок по продуктовой трансформации (управление приоритетами, подключение дополнительных продуктов при смене тарифного плана);
  - `productOfferingRecurringChargesEnabled` – рычаг включения возможности редактирования абонентской платы в операциях «Смена ТП (новая)» и «Подключение дополнительных продуктов»;
  - `individualizationOfRecurringChargesEnabled` – рычаг включения возможности индивидуализировать полный атрибутивный состав абонентской платы (АП) и объемы в групповых операциях (ГО) «Смена ТП (новая)», «Подключение дополнительных продуктов» и «Индивидуализация продуктов»;
- `settings`:
  - `defaultRussiaAirId` – идентификатор страны «Россия» по умолчанию в продукте AIR;
  - `recurringChargesRatingProfiles` – перечень идентификаторов тарификационных профилей, которые не должны отключаться при выполнении групповых операций «Отключение всех тарифных предложений (кроме АП) у абонентов», «Отключение всех тарифных предложений (кроме АП) у клиентов»;
- `monitoring`: – блок параметров мониторинга:
  - `jmx`:
    - `work` – флаг включения мониторинга (`true` – мониторинг включен или `false` – мониторинг выключен);
    - `BulkRequests`: – метрики в разрезе задач на групповые операции
      - `minEntitiesCountInBulkRequest` – количество сущностей в задаче на ГО, при достижении которого для этой задачи начинает собираться метрика по количеству обработанных записей;
    - `BulkRequestsSummary`: – общие метрики по задачам на групповые операции
      - `scheduler`:
        - `execInterval` – периодичность (с), с которой выполняется запрос в базе данных для получения метрик (если значение меньше 1, сбор метрик отключается).
- `utils`: – блок с утилитными скриптами предпроверок:
  - `preChecks`:
    - `common` – Groovy-скрипт, включающий в себя 6 предпроверок.

### 3.10.11. Настройки в файле `operations_backend_zk_config.yml`

В файле `inventory/localhost/group_vars/bulk_operations/operations_backend_zk_config.yml` задаются конфигурационные параметры `OPERATIONS_BACKEND` для ZooKeeper:

- `db`: – блок параметров соединения с базой данных:
  - `driverClassName` – драйвер базы данных;
  - `url` – JDBC URL соединения с базой данных;
  - `username` – имя пользователя базы данных;
  - `password` – зашифрованный пароль базы данных;
  - `initSQL` – запрос инициализации и проверки наличия соединения с базой данных;
  - `initialSize` – начальный размер пула соединений;
  - `maxActive` – максимальный размер пула соединений;
  - `maxIdle` – максимальное количество неиспользуемых соединений;
  - `minIdle` – минимальное количество неиспользуемых соединений;
  - `removeAbandonedTimeout` – время в секундах, через которое зависшие запросы будут удалены из пула;
  - `logAbandoned` – признак журналирования зависших соединений при их получении;
  - `maxWait` – максимальное время ожидания свободного соединения из пула (мс);
  - `queryTimeout` – максимальное время выполнения запроса (мс);
  - `removeAbandoned` – признак удаления зависших соединений из пула
  - `validationQueryTimeout` – таймаут для запроса на проверку соединения (с);
  - `validationInterval` – интервал между запросами на валидацию соединения (мс);
  - `validationQuery` – запрос для проверки соединения;
  - `testOnBorrow` – признак валидации соединения при взятии из пула;
- `excel_inputItemMc`: – блок параметров Excel-файлов с результатами проверки входных параметров операции:
  - `error` – заголовок столбца, содержащего текст ошибки;
  - `status` – заголовок столбца, содержащего статус проверки;
- `excel_resultItemMc`: – блок параметров Excel-файла с результатами операции:
  - `customer` – заголовок столбца, содержащего идентификатор клиента;
  - `error` – заголовок столбца, содержащего текст ошибки;
  - `status` – заголовок столбца, содержащего статус операции по клиенту/абоненту;
  - `subscriber` – заголовок столбца, содержащего идентификатор абонента;
- `excel_upload`: – блок параметров парсинга входящего Excel-файла:
  - `AccountNumber` – заголовок столбца, содержащего AccountNumber абонента в Excel-файле;
  - `McISDN` – заголовок столбца, содержащего MSISDN абонента в Excel-файле;
  - `batchSize` – размер пачки обработки элементов файла при парсинге входящего файла.
- `monitoring`: – блок параметров мониторинга:
  - `jmx`:
    - `work` – флаг включения мониторинга (`true` – мониторинг включен или `false` – мониторинг выключен);
- `console`: – блок параметров работы с административными консолями продукта:
  - `auth`: – блок параметров аутентификации для доступа к административным консолям продукта:
    - `type` – тип аутентификации для доступа к административным консолям продукта;

- `simple`: – блок параметров простой (статической) аутентификации:
  - `username` – имя пользователя для простой аутентификации;
  - `password` – зашифрованный пароль пользователя для простой аутентификации;
- `ldap`: – блок параметров LDAP-аутентификации:
  - `url` – адрес LDAP-сервера, по маске: `'ldap://host:port/dc=domain component,dc=domain component... etc'`;
  - `managerUser` – имя пользователя - «менеджера» для доступа к LDAP-серверу;  
Если оставить значение пустым, доступ к серверу будет анонимным.
  - `managerPassword` – зашифрованный пароль пользователя - «менеджера»;
  - `user`:
    - `dnPattern` – паттерн поиска пользователя (например, `'cn={0},ou=people'`);
    - `searchBase` – тип пользователя, который будет использоваться (например, `'ou=people'`);  
Может быть пустым, используется вместе с `searchFilter`.
    - `searchFilter` – фильтр поиска (например, `'uid={0}'`);  
Может быть пустым, если не задан `searchBase`.
  - `group`:
    - `searchBase` – тип группы пользователя (например, `'ou=groups'`);  
Может быть пустым, тогда поиск будет осуществляться во всех группах.
    - `searchFilter` – фильтрация пользователей в группе (например, `'uniqueMember={0}'`).
- `settings`: – блок общих параметров:
  - `maxRequestDateInFuture` – максимально разрешенное смещение даты запуска задачи относительно `sysdate` в будущем;
  - `dbTimeZone` – временная зона для работы с датами в базе данных.
- `result_reports`: – пользовательские классы для отчетов о результатах выполнения операции:
  - `ExampleCustomResultReportBuilder` – код примера пользовательского класса для тестовой ГО.

### 3.10.12. Настройки в файле `oracle_db.yml`

В файле `inventory/localhost/group_vars/bulk_operations/oracle_db` задаются параметры базы данных:

- `bulk_operations_tech_user` – имя технологического пользователя для выдачи прав для выполнения операций;
- `bulk_additional_user` – имя дополнительного пользователя, которому необходимо выдать права на объекты схемы `BULK`;
- `db`: – блок параметров базы данных:
  - `has`: – блок параметров базы данных со схемой `HAS`:
    - `user` – имя пользователя схемы `HAS`;
    - `password` – пароль пользователя схемы `HAS`;
    - `database` – наименование базы данных со схемой `HAS`;

- **bulk:** – блок параметров базы данных со схемой BULK:
  - **user** – имя пользователя схемы BULK;
  - **password** – зашифрованный пароль пользователя схемы BULK;
  - **database** – наименование базы данных со схемой BULK;
- **bulk\_operations\_tech\_user:** "OAPI\_CRAB";
- **d1m:** – блок параметров конфигурирования D1M для очистки устаревших данных:
  - **d1mMinStoragePeriod** – время хранения данных о запуске задачи (дни);
  - **d1mCleanPeriod** – периодичность запуска клинера;
- **db:** – блок параметров базы данных;
  - **dbtype** – тип базы данных ("oracle");
  - **schema** – имя схемы базы данных;
- **db\_liquibase:** – блок параметров новой схемы данных ГО:
  - **username** – имя нового пользователя схемы данных ГО;
  - **password** – незашифрованный пароль нового пользователя схемы данных ГО;
  - **url** – JDBC URL соединения с новой схемы данных ГО;
  - **changelog\_paramc:** – блок параметров для отслеживания изменений при миграции:
    - **tailored** – клиент, для которого устанавливается серверная часть;
    - **schema\_name** – имя нового пользователя схемы данных ГО;
    - **user\_name** – имя пользователя ГО (имя из параметра **schema\_name** или же имя дополнительного пользователя, которому необходимо выдать права на объекты схемы BULK);
  - **user\_password** – пароль пользователя ГО.

### 3.10.13. Настройки в файле `postgresql_db.yml`

В файле `inventory/localhost/group_vars/postgresql_db.yml` задаются параметры:

- **db:** – блок параметров базы данных:
  - **bulk:** – блок параметров базы данных со схемой BULK:
    - **user** – имя пользователя схемы BULK;
    - **password** – зашифрованный пароль пользователя схемы BULK;
    - **database** – наименование базы данных со схемой BULK;
- **bulk\_operations\_tech\_user** – имя технологического пользователя, которому необходимо выдать права на объекты схемы BULK;
- **d1m:** – блок параметров конфигурирования D1M для очистки устаревших данных:
  - **db:** – блок параметров базы данных;
    - **dbtype** – тип базы данных ("postgresql");
    - **schema** – имя схемы базы данных;
  - **d1mMinStoragePeriod** – время хранения данных о запуске задачи в днях;
- **db\_liquibase:** – блок параметров новой схемы данных ГО:
  - **username** – имя нового пользователя схемы данных ГО;
  - **password** – незашифрованный пароль нового пользователя схемы данных ГО;
  - **driver** – название драйвера базы данных ("org.postgresql.Driver");
  - **url** – JDBC URL соединения с новой схемой данных ГО;
  - **jvm\_file\_encoding** – кодировка файлов ("UTF-8");
  - **defaultSchemaName** – имя схемы данных для служебных таблиц Liquibase;
  - **changelog\_paramc:** – блок параметров для отслеживания изменений при миграции:

- `tailored` – клиент, для которого устанавливается серверная часть;
- `schema_name` – имя нового пользователя схемы данных ГО;
- `app_user_name` – имя пользователя для подключения продукта к базе данных;
- `owner_name` – логин пользователя - владельца объектов;
- `emptyDictionaries` – флаг, который указывает, следует ли наполнять справочники данными (`true/false`);
- `dmlMinStoragePeriod` – время хранения данных о запуске задачи (дни);
- `maxRequestDateInFuture` – максимальное разрешенное смещение даты запуска задачи относительно `sysdate` в будущее (дни).

### 3.10.14. Файл для проверки корректности заполнения `inventory`

Схемы валидации для конфигурационных файлов находятся в каталоге `schemas`.

В файле `schemas/bulk_operations-schema.json` в json-формате описаны параметры, которые необходимо задать в файле `inventory/localhost/group_vars/<product>.yaml`.

```
ansible-playbook -i inventory/localhost -e  
"ansible_ssh_pass=пароль_пользователя ansible_ssh_user=имя_пользователя"  
bulk_operations-validate-parameters.yml
```

### 3.10.15. Теги

Доступные теги:

- `bulk` – установка на схему BULK;
- `bulk_operations_backend` – установка и конфигурирование компонента `bulk_operations_backend`;
- `bulk_runner` – установка и конфигурирование компонента `bulk_runner`;
- `bulk_fill_dictionaries` – заполнение справочников (операции, типы, статусы) на схеме BULK;
- `check` – проверка конфигурационных параметров;
- `configure` – конфигурирование узлов в ZooKeeper и каталога `conf` в Tomcat;
- `db` – установка серверной части;
- `dml_configure` – установка конфигурации DLM;
- `deploy` – установка `war`-архивов компонентов;
- `has` – установка на схему HAS;
- `setup_tech_user` – выдача ролей для выполнения операций технологическому пользователю;
- `mapping` – установка маппинга входных параметров операций в ZooKeeper;
- `roles` – установка ролевой модели;
- `scripts` – установка сценариев групповых операций в ZooKeeper;
- `war` – работа с Backend-частью на Tomcat;
- `zookeeper` – установка параметров в ZooKeeper для всех компонентов.

### 3.10.16. Окружение ANSIBLE

В файле `ansible.cfg` задаются параметры настройки Ansible:

```
[defaults]
host_key_checking = False # Проверка ключей при подключении по SSH
accept_hostkey = True # Проверка наличия ssh опции -o
StrictHostKeyChecking=no
hash_behaviour = merge # Тип слияния переменных ansible
roles_path = common # Путь до ansible ролей
[ssh_connection] # Параметры ssh подключения
ssh_args = -o ControlMaster=auto -o ControlPersist=60s -o
UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
```

## 3.11. Обновление продукта

Обновление продукта происходит так же, как и установка.

### Обновление без простоя

Если продукт установлен на несколько серверов Tomcat (есть минимум два экземпляра `oapi-operations-backend` и минимум два экземпляра `oapi_bulk_runner`), то возможна установка "без простоя", т.е. без предварительной остановки backend'ов и/или серверов Tomcat на время наката новой версии.

Для этого необходимо корректно настроить хосты в файле [inventory/localhost/1-rnd-bulk\\_operations](#).

Для обеспечения гарантированной последовательной обработки экземпляров backend'ов в файле `1-rnd-bulk_operations` для каждого backend'a (`bulk_operations_backend` и `bulk_operations_runner`) создано две группы хостов - по последовательности остановки/пуска экземпляров backend'ов. В каждой группе прописываются несколько целевых хостов, на которые происходит накатка backend'ов.

Например,

```
[bulk_operations_backend_1]
bulk_operations_backend-host1 ansible_host=host1

[bulk_operations_backend_2]
bulk_operations_backend-host2 ansible_host=host2

[bulk_operations_runner_1]
bulk_operations_runner-host1 ansible_host=host1

[bulk_operations_runner_2]
bulk_operations_runner-host2 ansible_host=host2

[bulk_operations_backend:children]
bulk_operations_backend_1
bulk_operations_backend_2

[bulk_operations_runner:children]
bulk_operations_runner_1
bulk_operations_runner_2

[bulk_operations_tomcat:children]
bulk_operations_backend_1
bulk_operations_backend_2
bulk_operations_runner_1
bulk_operations_runner_2
```



**Внимание!**

Backend'ы одной группы должны быть установлены на одном и том же сервере Tomcat.

Целевая последовательность накатки:

1. Обновление схемы.  
В случае серьезных изменений, требующих большого времени для установки, эту часть допустимо установить предварительно, отдельно от остальных части продукта.
2. Обновление серверной части.
3. Обновление ZooKeeper.
4. Обновление клиентской части.

При обновлении клиентской части экземпляры backend'ов, находящиеся на хостах в группах `bulk_operations_backend_1` и `bulk_operations_runner_1` будут обработаны на первом шаге:

- выведены из-под нагрузки (через балансировщик Nginx);
- остановлены;
- обновлены;

- запущены;
- введены в работу (через балансировщик Nginx).

Во время этой обработки работа продукта будет обеспечиваться экземплярами, которые находятся на хостах из групп `bulk_operations_backend_2` и `bulk_operations_runner_2`. После окончания обновления backend'ов на первую группу хостов производится переключение версии (`current-version`) продукта в ZooKeeper, нагрузка переводится на обновленные экземпляры и описанные шаги по обновлению продукта применяются к экземплярам backend'ов из второй группы хостов.

Описания хостов необходимо внести в соответствующие файлы:

`bulk_operations_backend_1.yml`, `bulk_operations_backend_2.yml` и `bulk_operations_runner_1.yml`, `bulk_operations_runner_2.yml` (см. описание в разделе «[Конфигурация продукта](#)»).

## 3.12. Настройка конфигурации DLM

Реализован скрипт `d1m_config.sql`, который задает конфигурацию продукта DLM для обеспечения очистки устаревших данных продукта `BULK_OPERATIONS`.

Скрипт представляет собой PL/SQL-блок с вызовом PL/SQL API продукта DLM и автоматически запускается инсталлятором `BULK_OPERATIONS` при установке продукта (подробное описание DLM приведено в документации на этот продукт).

### Конфигурирование для базы данных Oracle

Скрипт передает параметры согласно значениям, заданным в конфигурационном файле `inventory/localhost/group_vars/bulk_operations/operations_backend_zk_config.yml`:

- `d1mMinStoragePeriod` – время хранения данных о запуске задачи (в днях; по умолчанию – 365);
- `maxRequestDateInFuture` – максимальное разрешенное смещение даты запуска задачи относительно `sysdate` в будущее (в днях; по умолчанию – 365);
- `d1mCleanPeriod` – периодичность запуска клинера (по умолчанию – `FREQ=MONTHLY;BYMONTHDAY = 15`).

Скрипт создает конфигурацию очистки схемы базы данных и после этого запускает задание (job), которое будет выполняться с периодичностью, указанной в параметре `d1mCleanPeriod`.

Если необходимо изменить конфигурацию во время работы продукта:

- установите новые значения для параметров `inventory` и запустите автоинсталлятор с тегом `d1m_configure`

или

- установите новые значения для параметров в файле `SCR/d1m_configure/d1m_config_def.sql`, в файле `SCR/d1m_configure/d1m_config_setup.bat` отредактируйте строку соединения с базой данных и запустите скрипт вручную.

### Конфигурирование для базы данных PostgreSQL

Скрипт передает параметры согласно значениям, заданным в конфигурационном файле `inventory/localhost/group_vars/bulk_operations/postgresql_db.yml`:

- `d1mMinStoragePeriod` – время хранения данных о запуске задачи (в днях; по умолчанию – 183);
- `maxRequestDateInFuture` – максимальное разрешенное смещение даты запуска задачи относительно `sysdate` в будущее (в днях; по умолчанию – 365);
- `d1mCleanPeriod` – периодичность запуска клинера (по умолчанию (cron-style) – `* 1 * *`

\*).

Скрипт создает конфигурацию очистки схемы базы данных и после этого запускает cron-задачу, которая будет выполняться с периодичностью, указанной в параметре `d1mCleanPeriod`.

Если необходимо изменить конфигурацию во время работы продукта, запустите автоинсталлятор только с тегом `d1m_configure` с новыми значениями указанных выше параметров или выполните изменения в схеме базы данных согласно документации DLM.

## 4. Запуск и остановка продукта

Запуск и остановка продукта выполняются посредством последовательного запуска/остановки каждого компонента (OAPI\_OPERATIONS\_BACKEND и OAPI\_BULK\_RUNNER) в административной панели Tomcat.

## 5. Проверка работоспособности

После установки продукта убедитесь, что:

- нет ошибок при выполнении установки с помощью Ansible;
- журнальные файлы установки серверной части продукта не содержат сообщений об ошибках;
- компоненты продукта зарегистрированы в схеме HAS и схеме BIS (либо в той, куда устанавливалась bulk-часть продукта).

Проверьте работоспособность компонентов `OAPI_OPERATIONS_BACKEND` и `OAPI_BULK_RUNNER` после установки на Tomcat:

- модуль стартовал;
- журнальные файлы не содержат сообщений об ошибках.