

ЦИФРОВАЯ СИСТЕМА ТРАНСФОРМАЦИИ ДАННЫХ

Руководство по установке

Версия 1.1.0

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.com; www.nexign.com

Содержание

1. Предварительные требования	4
2. Установка	6
2.1. Подготовка дистрибутива	6
2.2. Порядок установки	6
2.3. Откат	9
3. Запуск и остановка продукта	10
4. Обновление	11
5. Проверка работоспособности	12
Приложение А: Окружение Ansible	13
Приложение В: Файлы playbook	14
Приложение С: Конфигурация продукта	15
С.1. Параметры в файлах каталога inventory/group_vars	15
С.2. Параметры файла all.yml	16
С.3. Параметры файла mds.yml	17
С.4. Параметры файла dip_dispatcher.yml	22
С.5. Параметры файла dip_web.yml	22

1. Предварительные требования

Для установки продукта «Цифровая система трансформации данных» (MDS) обеспечьте выполнение [требований к компьютеру, с которого выполняется установка](#).

Для эксплуатации продукта выполните требования:

- [к серверам](#);
- [компьютеру пользователя](#).

Требования к компьютеру, с которого выполняется установка

- Ansible версии 2.10.7;
- Java версии 1.8.0_131 или выше;
- ОС Red Hat® Enterprise Linux (RHEL) версий 7.7, 8 или РЕД ОС.

Требования к серверам

Рекомендуемые характеристики сервера:

- CPU 4 ядра;
- RAM 20Гб;
- HDD 80/180 Гб.

Требования к серверу для функционирования скриптов, выполняющих вычисления:

	Минимальные	Рекомендуемые
Процессор	Intel Core 6-ядерный с тактовой частотой 2,5 ГГц или аналогичный	Intel Core 16-ядерный с тактовой частотой 3 ГГц или аналогичный
Оперативная память	16 Гб	32 Гб

Для корректной работы MDS установите на сервер:

- ОС Red Hat® Enterprise Linux (RHEL) версий 7.7, 8 или РЕД ОС;
- SQL-база данных Oracle версии 19 и выше или PostgreSQL 14.

Актуальные версии программного обеспечения см. в файле `releasenotes.txt`, поставляемом в составе дистрибутива.

Требования к компьютеру пользователя

	Минимальные	Рекомендуемые
Процессор	Intel Pentium 4 CPU с тактовой частотой 2,5 ГГц или аналогичный	Intel Core 2-ядерный с тактовой частотой 3 ГГц или аналогичный
Оперативная память	1 Гб	4 Гб
Разрешение экрана монитора	1024x768 px	1680x1050 px
Пропускная способность канала	512 Кбит/с	2 Мбит/с

Продукт поддерживает работу через web-браузеры:

- Mozilla Firefox v.69 и выше;
- Google Chrome v.64 и выше.

2. Установка

Установка выполняется на устройство, которое предоставляет сервисы формата «клиент – сервер» в режиме сервера по каким-либо интерфейсам и уникально определено на этих интерфейсах (хост).

2.1. Подготовка дистрибутива

Перед установкой подготовьте дистрибутив продукта:

- из репозитория Artifactory или Nexus (в зависимости от инструмента, который используется для хранения и управления сборками):
 - скачайте из Artifactory и разархивируйте каталог install из состава дистрибутива продукта; файлы разместите в каталоге mds-install на компьютере, с которого производится установка;
 - если включена обязательная авторизация для скачивания из Artifactory, добавьте в файл requirements.yml строку:

```
src://http://<USER>:<ENCRYPTED_PASSWORD>@<artifactory_url>
```

Параметр	Описание
<USER>	Имя пользователя для входа в Artifactory. Выдается администратором сервера Artifactory
<ENCRYPTED_PASSWORD>	Зашифрованный пароль. Для получения зашифрованного пароля авторизуйтесь в Artifactory, щелкните на свое имя в правом верхнем углу и выберите «Copy encrypted password to clipboard». Подтвердите личность и скопируйте зашифрованный пароль
<ARTIFACTORY_REPO>	URL для скачивания архива с общими ролями

- запустите скрипт ansible-prepare.sh на компьютере, с которого производится установка, от своего имени или от имени технологического пользователя; из Artifactory будет скачан дистрибутив с общими ролями;
- из локального репозитория:
 - разархивируйте дистрибутив продукта;
 - настройте локальную установку – в файле inventory/group_vars/all.yml установите значения параметров:
 - `artifactory.enable = false;`
 - `local_storage.enable = true.`

2.2. Порядок установки

1. Создайте схему развертывания. В каталоге дистрибутива inventory/localhost содержится пример с настройками схемы развёртывания. Чтобы создать собственную схему развёртывания, создайте копию каталога localhost и переименуйте её (например, staging). Создание разных каталогов позволяет хранить конфигурацию для разных площадок, например: inventory/staging, inventory/prom, inventory/testing.
2. Задайте обязательные параметры установки:

Файл	Параметр	Описание
mds.yml	platform	Название платформы, на которую производится установка (i686-redhat-linux(7.7)-64, i686-redhat-linux(8.0)-64, i686-redos-linux(7.0)-64)
	ps.mds.spring.datasource.url	Строка соединения с базой данных продукта
	ps.mds.spring.datasource.username	Логин базы данных продукта
	ps.mds.spring.datasource.password	Пароль базы данных продукта
dip_web.yml	app_name	Наименование компонента
dip_dispatcher.yml	app_name	Наименование компонента

3. Настройте хосты сервера, на котором устанавливается продукт, и опишите группы серверов схемы развертывания в файле или каталоге `inventory/<схема развертывания>/1-md-mds`. В каждой группе укажите серверы, на которые необходимо выполнить установку компонентов продукта, и параметры соединения с ними в формате: `alias`, имя серверной машины `ansible_host` и метод подключения `ansible_connection`.

Пример:

```
mds ansible_connection=ssh ansible_host=<адрес удаленной машины> # deploy на удаленную машину
```

или

```
mds ansible_connection=local ansible_host=<схема развертывания> # deploy на <схема развертывания>
```

Файл `inventory/<схема развертывания>/1-md-mds` по умолчанию заполнен параметрами для локальной установки и в минимальной конфигурации. Нельзя изменять названия существующих групп. Можно добавлять новые хосты, новые группы и наследования.

4. При необходимости переопределите [настройки конфигурации](#) в каталоге `inventory/<схема развертывания>/group_vars` и [параметры конфигурации Ansible в файле `ansible.cfg`](#). Для проверки корректности заполнения параметров `inventory` запустите скрипт:

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

В файле `schemas/mds-schema.json` в формате JSON описаны параметры, которые

необходимо задать в файле `inventory/<схема_развертывания>/group_vars/mds.yml`.

5. Запустите установку, задав необходимые параметры:

```
./auto-deploy.sh -i inventory/<схема развертывания> -u -k
```

Табл. 1. Параметры запуска

Параметр (ключ)	Описание
Основные параметры	
<code>-i inventory/<схема развертывания></code>	Файл <code>inventory</code> , который будет использоваться при выполнении сценария
<code>-u</code>	Пользователь, от имени которого Ansible будет подключаться к серверам, указанным в файле хостов в каталоге <code>inventory</code>
<code>-k</code>	Необходимость ввода пароля пользователя, указанного в ключе <code>-u</code>
Дополнительные параметры	
<code>--become-user</code>	Привилегированный пользователь
<code>-K</code>	Необходимость ввода пароля для привилегированного пользователя, указанного в ключе <code>--become-user</code>
<code>-b</code>	Необходимость выполнения сценария от имени привилегированного пользователя
<code>-t</code>	Перечисление тегов, которые будут выполняться в сценарии (по умолчанию используется <code>all</code> , то есть запуск всех тегов): <ul style="list-style-type: none"> <code>check</code> – проверка <code>inventory</code> на соответствие JSON-схеме; <code>configure</code> – установка конфигурационных файлов; <code>deploy</code> – установка и настройка всего продукта; <code>stop</code> – остановка всех компонентов продукта; <code>start</code> – запуск всех компонентов продукта; <code>restart</code> – перезапуск всех компонентов продукта; <code>rollback</code> – откат на предыдущую версию компонента; <code>success</code> – создание внутри версии компонента файла маркера <code>DEPLOY_SUCCESS</code>; <code>finalize</code> – переключение символической ссылки на последнюю версию, удаление промежуточных «незавершенных» версий, в которых присутствует файл <code>DEPLOY_UNFINISHED</code>, перезапуск компонентов
<code>-l</code>	Перечисление хостов или групп, по которым будет выполнен сценарий (по умолчанию сценарий будет выполнен для всех хостов)
<code>-v</code>	Уровень журналирования; максимальное количество <code>v=4 (-vvvv)</code> , чем больше <code>v</code> , тем выше уровень журналирования

Параметр (ключ)	Описание
-e	Дополнительные переменные задаются в виде key=value или YAML/JSON. Если указывается имя файла, то имя должно начинаться с @

Полный список ключей можно посмотреть в [официальной документации Ansible](#).

6. Если не включена автоматическая финализация (в файле all.yml для параметра `finalize_after_deploy` указано значение `false`), переключитесь на новую версию продукта:

```
./auto-finalize.sh -i inventory/<схема развертывания>
```

7. Зафиксируйте установку версии (статус новой версии продукта – «Успешно установлена»):

```
./auto-success.sh -i inventory/<схема развертывания>
```

Описание параметров запуска см. в [Табл. 1](#).

2.3. Откат

При необходимости отката на предыдущую версию продукта выполните команду:

```
./auto-rollback.sh -i inventory/<схема развертывания>
```

- при наличии на сервере финализированных предыдущих версий продукта происходит откат на предыдущую версию, то есть переключение текущей ссылки (`current`) на предыдущую версию;
- при отсутствии на сервере предыдущих версий продукта откат будет остановлен с ошибкой «rollback_path is defined».

Для удаления артефактов незавершенных установок продукта выполните команду:

```
./auto-remove-unfinished.sh -i inventory/<схема развертывания>
```

Описание параметров запуска см. в [Табл. 1](#).

3. Запуск и остановка продукта

Для запуска продукта выполните команду:

```
./auto-deploy.sh -t start -i inventory/<схема развертывания>
```

Для остановки продукта выполните команду:

```
./auto-deploy.sh -t stop -i inventory/<схема развертывания>
```

4. Обновление

Для обновления продукта установите дистрибутив требуемой версии (см. главу «[Установка](#)»).

На текущий каталог установки указывает символическая ссылка с именем, совпадающим с именем компонента без временной метки.

При установке создается новый каталог с именем компонента, дополненный текущей временной меткой, в который выполняется установка. Если установка выполнена успешно, символическая ссылка сдвигается и начинает указывать на новый каталог.

5. Проверка работоспособности

Для проверки работоспособности:

1. Убедитесь в отсутствии ошибок при установке.
2. В браузере перейдите на сервер, на котором установлен продукт, на указанный в конфигурации порт.
3. Откройте раздел «Ядра» на вкладке «Управление» и убедитесь, что ядра в статусе «Сбой» отсутствуют.

Приложение А: Окружение Ansible

Пример заполнения файла `ansible.cfg`:

```
[defaults]
roles_path = common
hash_behaviour = merge
log_path = ./ansible_mds.log
callback_whitelist = profile_tasks
display_skipped_hosts = False
stdout_callback = debug

[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s -o
UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
```

где:

- `roles_path` – дополнительный путь поиска ролей для Ansible;
- `hash_behaviour` – стратегия слияния переменных Ansible при совпадении имен;
- `log_path` – файл, в который Ansible будет записывать информацию о процессе установки продукта;
- `callback_whitelist` – включение профилирования выполняемых задач (в системный журнал выводится время выполнения задач);
- `display_skipped_hosts` – включение вывода на консоль информации по пропущенным задачам/хостам;
- `stdout_callback` – callback для вывода отладочной информации при установке продукта в удобном для пользователя виде;
- `[ssh_connection]` – настройки SSH-соединений.

Приложение В: Файлы playbook

Для установки продукта используются файлы playbook:

- mds-deploy.yml – основной сценарий установки продукта;
- mds-deploy-full.yml – установка продукта целиком;
- mds-deploy-rollback.yml – откат на предыдущую версию;
- mds-deploy-success.yml – фиксация успешной установки;
- mds-deploy-finalize.yml – переключение на новую (установленную) версию;
- mds-schema-generate.yml – создание json-файла с содержимым Inventory продукта;
- mds-validate-parameters.yml – проверка конфигурационных параметров;
- mds-remove-unfinished.yml – удаление всех незавершенных установок версий продукта.

Приложение С: Конфигурация продукта

Конфигурация продукта задается в каталогах:

- [inventory/group_vars](#) – настройки, которые не изменяются от стенда к стенду;
- `inventory/<схема разворачивания>/group_vars` – настройки, которые могут изменяться; эти настройки зависят от окружения, на которое будет устанавливаться продукт:
 - [inventory/<схема разворачивания>/group_vars/all.yml](#);
 - [inventory/<схема разворачивания>/group_vars/mds.yml](#);
 - [inventory/<схема разворачивания>/group_vars/dip_dispatcher.yml](#);
 - [inventory/<схема разворачивания>/group_vars/dip_web.yml](#).

По умолчанию параметры обязательны для заполнения, если не указано иное.

С.1. Параметры в файлах каталога `inventory/group_vars`

- `artifactory`:
 - `path` – путь к каталогу продукта в Artifactory; задается через «*» в виде «*<product_group>*<product>»;
 - `start_command` – команда для старта продукта; команда запускается с помощью роли `app/start` на сервере из каталога `app_dir`;
 - `stop_command` – команда для остановки продукта; команда запускается с помощью роли `app/stop` на сервере из каталога `app_dir`;
 - `kill_command` – команда для принудительной остановки процесса; команда запускается с помощью роли `app/kill`;
 - `pre_deploy_scripts` – используется для запуска команд на целевой машине до начала установки продукта;
 - `post_deploy_scripts` – используется для запуска команд на целевой машине после установки продукта;
 - `store_on_host` – параметр влияет на то, останется ли архив с `app` на сервере или нет; необходим для больших, редко меняющихся приложений;

Если значение параметра `false` (или параметр не определен), то архив скачивается, распакуется и удалится, если `true`, то архив останется на сервере и в следующий раз при установке, если архив не менялся, он не будет повторно скачиваться, будет использована скачанная ранее версия.

- `war_name` – наименование `war`-файла на сервере;
- `tomcat_context` – параметр для записи данных в файл `context.xml`;
- `artifacts_deploy_order` – порядок установки `kar`-файла и `jar`-файла для Karaf;
- `artifacts_content` – детальная информация по файлу:
 - `match` – название файла;
 - `features` – функциональные возможности, которые разворачиваются в рамках этого файла;
 - `bundles` – `bundle`, которые разворачиваются в рамках этого файла:
 - `symbolic_name` – `Bundle-SymbolicName`, по которому можно однозначно определить `bundle` в клиенте Karaf (для определения команда `bundle:headers <id>` в клиенте Karaf);
 - `services` – сервисы, от которых зависит `bundle` (для определения команда `bundle:services <id>` в клиенте Karaf); если таковые имеются, все указывать необязательно, достаточно указать один основной, часто им является `org.apache.camel.CamelContext`;

- `zapp_name` – наименование компонента в ZooKeeper (по умолчанию: `app_name`);
- `znode_path` – узел в ZooKeeper, в который добавляются данные из шаблона конфигурации (по умолчанию: `ps/config/apps/{{ zapp_name }}`);
- `classifier` – каталог в Artifactory в `conf`, из которого загружаются шаблоны (по умолчанию: `it`);
- `product_name` – имя продукта; используется для генерации файла `<product_name>-json.schema`;
- `db_flyway` – параметры для роли `db/flyway`;
- `db_sqlplus` – параметры для роли `db/sqlplus`;
- `db_liquibase` – параметры для роли `db/liquibase`;
- `supervisor_unit_parameters`:
 - `command` – команда запуска продукта в режиме `foreground`; следует учитывать, что супервизор не запускает `shell`, `bash`, `sh`, поэтому при необходимости запуска `sh`-скриптов требуется указывать `/bin/bash` для его принудительного запуска;
 - `directory` – каталог, из которого будет запущена команда;
- `check_running` – словарь, в котором определяются типы проверок при старте и остановке продукта, а также имена параметров для этих приложений; изменяемые части проверок должны быть вынесены в `inventory`, а внутри словаря должны быть использованы их имена через подстановку;
- `ps.product.param` – параметры словаря `ps.product`, изменение которых не допускается.

[Подробное описание Ansible.](#)

C.2. Параметры файла `all.yml`

В файле `inventory/<схема развертывания>/group_vars/all.yml` задаются параметры:

- `artifactory` – настройки Artifactory;
 - `url` – URL-адрес Artifactory;
 - `search_url` – URL к поисковому API Artifactory;
 - `api_key` – ключ для аутентификации пользователя;
 - `repo` – репозиторий, в котором проводится поиск;
- `base_dir` – базовый каталог;
- `logs_dir` – каталог расположения логов;
- `mibs_dir` – каталог расположения `mib`-файлов;
- `info_dir` – каталог расположения `info.json`-файлов;
- `emon_json_dir` – каталог расположения `emon.json`-файлов;
- `supervisor` – настройки `supervisor`;
- `need_check_supervisor` – проверка включения в автозапуске `supervisor`;
- `become_user_<product>` – пользователь, от имени которого будут выполняться команды на удаленном хосте;
- `sso_api` – настройки для взаимодействия с продуктом SSO («Аутентификация пользователей по технологии единого входа»);
- `start_checking_timeout` – период ожидания ответа от хоста.

С.3. Параметры файла `mds.yml`



Примечание.

Параметры, отсутствующие в файле, добавляются вручную при необходимости переопределения значения по умолчанию. В файле `schemas/mds-schema.json` в json-формате описаны параметры, которые необходимо задать в файле `inventory/<схема разворачивания>/group_vars/mds.yml`.

Настройки конфигурации продукта в файле `inventory/<схема разворачивания>/group_vars/mds.yml`:

`ps.mds.install_key_file`

Путь к файлу с ключом шифрования на компьютере, с которого производится установка.

`ps.mds.key_file`

Путь к файлу с ключом шифрования на удалённом сервере; по умолчанию ключ будет создан в корне рабочей папки устанавливаемой версии, например: `/data/mds/versions/******/key_file`.

`ps.mds.passwords_file`

Путь к файлу с зашифрованными паролями; по умолчанию файл будет размещен в корне рабочей папки устанавливаемой версии, например: `/data/mds/versions/******/passwords_file`.

`ps.mds.key`

Ключ шифрования паролей длиной не менее 32 байт; если параметр не задан, ключ формируется случайным образом.

`ps.mds.additional_encrypted_variables`

Список переменных окружения и их значений, которые требуется зашифровать в дополнение к уже шифруемым паролям; результат шифрования будет добавлен в файл, указанный в переменной `ps.mds.passwords_file`; каждый элемент списка содержит поля `name` (название переменной окружения) и `value` (шифруемое значение); значение по умолчанию: `[]`.

```
- name: "TEST_VAR"
  value: "TEST_PASS"
- name: "TEST_VAR2"
  value: "TEST_PASS2"
```



Примечание.

После установки продукта вместо паролей в открытом виде в конфигурационных файлах продукта будут отображаться ссылки на переменные окружения, например: `spring.datasource.password=${MDS_DB_PASS}`. При запуске MDS пароли будут расшифрованы и установлены в переменные окружения процесса.

`ps.mds.logging`

- `level` – уровень журналирования.

ps.mds.ssl

Настройки SSL:

- `enabled` – признак включения режима HTTPS (по умолчанию: `false`);
- `key_alias` – псевдоним (или имя), под которым ключ хранится в хранилище ключей (по умолчанию: `mds`);
- `key_store_type` – тип хранилища ключей (по умолчанию: `PKCS12`);
- `key_store` – путь к файлу хранилища ключей (по умолчанию: `/path/keystore.p12`);
- `key_store_password` – пароль (по умолчанию: `password`).

ps.mds.spring

- `datasource`:
 - `url` – строка соединения с базой данных продукта;
 - `username` – логин базы данных продукта;
 - `owner` – имя пользователя-владельца схемы базы данных;
 - `password` – пароль базы данных продукта;
 - `install_username` – логин пользователя, от имени которого выполняется установка базы данных; по умолчанию совпадает со значением параметра `ps.mds.spring.datasource.username`;
 - `install_password` – пароль пользователя, от имени которого выполняется установка базы данных; по умолчанию совпадает со значением параметра `ps.mds.spring.datasource.password`;
 - `install_ignore` – коды ошибок базы данных, которые должны быть проигнорированы при установке базы данных; данные ошибки расширяют предустановленный набор; формат: "ORA-NNNNN,ORA-MMMMM" (по умолчанию: "");
 - `install_skip` – флаг, определяющий, должна ли быть пропущена установка базы данных (по умолчанию: `false`);
- `multipart_max_file_size` – максимальный размер multipart-файла (по умолчанию: 1MB);
- `multipart_max_request_size` – максимальный размер запроса с multipart-файлом (по умолчанию: 10MB).

ps.mds.audit

- `enabled` – флаг включения/отключения работы с сообщениями аудита; по умолчанию: `false`; возможные значения:
 - `false` – интеграция отключена;
 - `true` – интеграция включена;
- `topic` – название топика Kafka для записи сообщений аудита (по умолчанию: `wpsec-audit-events`);
- `producer_timeout` – максимальное время ожидания подтверждения о записи данных в топик аудита в секундах (по умолчанию: 10);
- `check_interval` – интервал сканирования, в миллисекундах;
- `drop_timeout` – время в миллисекундах, по истечении которого неотправленные сообщения будут удалены;
- `bootstrap_servers` – список пар «хост:порт», разделённый запятыми, с адресами брокеров Kafka; примеры: `localhost:9092, localhost:9092, another.host:9092` (по умолчанию: `localhost:9092`);
- `schema_registry_url` – URL реестра схем Kafka (по умолчанию:

http://localhost:8081).

ps.mds.dispatcher

- **server:**
 - **requestdoctor** – настройки проверки состояния ресурсной модели:
 - **enabled** – включение/выключение проверки состояния ресурсной модели (по умолчанию: `true`);
 - **interval_ms** – периодичность запуска проверки в миллисекундах (по умолчанию: `3600000`);
 - **serverdoctor** – настройки проверки состояния серверов:
 - **enabled** – включение/выключение проверки состояния серверов (по умолчанию: `true`);
 - **interval_ms** – периодичность запуска проверки в миллисекундах (по умолчанию: `60000`);
 - **delay** – настройки периодичности запуска проверки серверов с ошибками:
 - **default_minutes** – периодичность запуска проверки серверов с общими ошибками в минутах (по умолчанию: `20`);
 - **login_minutes** – периодичность запуска проверки серверов с ошибками логина или пароля в минутах (по умолчанию: `20`);
 - **host_minutes** – периодичность запуска проверки серверов с ошибками отсутствия сервера в минутах (по умолчанию: `5`);
 - **additional_start_parameters** – дополнительный параметр для запуска продукта;
 - **port** – HTTP-порт web-части;
 - **filewatcher_enabled** – флаг использования сервиса отслеживания состояния файла с настройкой уровня журналирования (по умолчанию: `false`);
 - **executor:**
 - **options** – управление памятью при выполнении процессов (по умолчанию: `"-Xmx2048M -XX:-UseGCOverheadLimit"`);
 - **maintainer:**
 - **port** – порт для работы постоянных исполнителей;
 - **options** – управление памятью при выполнении процессов постоянными исполнителями;
- **cleaner:**
 - **enabled** – флаг включения/отключения автоматической очистки данных (по умолчанию: `true`); возможные значения:
 - `false` – очистка отключена;
 - `true` – очистка включена;
 - **check_interval** – интервал запуска очистки в миллисекундах (по умолчанию: `1800000`);
 - **depth_storage_minutes** – глубина хранения файлов в хранилищах в минутах (по умолчанию: `7200`);
 - **depth_temp_minutes** – глубина хранения временных файлов в минутах (каталог `spark.local.dir`); по умолчанию: `180`;
 - **depth_logs_minutes** – глубина хранения логов в базе данных в минутах (по умолчанию: `43200`);
 - **depth_statistics_minutes** – глубина хранения статистики в базе данных в минутах (по умолчанию: `43200`);
 - **depth_tasks_minutes** – глубина хранения задач и связанных с ними данных в базе данных в минутах (по умолчанию: `129600`);

- `depth_resources_minutes` – глубина хранения ресурсов в базе данных в минутах (по умолчанию: 43200);
- `depth_requests_minutes` – глубина хранения запросов в базе данных в минутах (по умолчанию: 43200);
- `depth_history_minutes` – глубина хранения архивных версий сценариев в минутах (по умолчанию: 43200);
- `depth_purge_minutes` – глубина хранения информации об очищаемых данных в минутах (по умолчанию: 43200);
- `page_tasks` – количество задач, очищаемых за один раз при запуске системы очистки задач (по умолчанию: 1000);
- `killer`:
 - `enabled` – флаг включения/отключения проставления тегов (по умолчанию: `true`);
 - `interval_ms` – интервал сканирования, в миллисекундах (по умолчанию: 60000);
 - `precision_ms` – задержка в миллисекундах, используемая для сравнения дат (по умолчанию: 60000);
 - `timeout_minutes` – тайм-аут времени выполнения процесса, в минутах (по умолчанию: 300).

ps.mds.web

- `server`:
 - `additional_start_parameters` – дополнительный параметр для запуска продукта;
 - `manager` – настройки суперпользователя:
 - `enabled` – флаг включения/отключения;
 - `login` – логин суперпользователя;
 - `password` – пароль суперпользователя;
 - `autotager` – настройки автоматического проставления тегов сценариям:
 - `enable` – флаг включения/отключения проставления тегов (по умолчанию: `false`);
 - `interval_ms` – интервал, через который стартует проставление тегов, в миллисекундах (по умолчанию: 1000);
 - `synch` – синхронизация сценариев и серверов:
 - `enabled` – синхронизация включена/выключена (по умолчанию: `false`);
 - `interval_ms` – интервал в миллисекундах, через который стартует синхронизация (по умолчанию: 30000);
 - `folder.scenarios` – каталог, из которого рекурсивно (с подкаталогами) считываются файлы сценариев для синхронизации (по умолчанию: `""`);
 - `folder.servers` – каталог, из которого считываются (без подкаталогов) файлы серверов для синхронизации (по умолчанию: `""`);
 - `jolokia.port` – порт для старта Jolokia;
 - `config.animation` – включение/отключение анимации (по умолчанию: `false`);
 - `scenarios_import_folder` – каталог для сценариев, импортированных через OpenAPI (по умолчанию: `/import/oapi`);
- `zookeeper`:
 - `registration_enabled` – флаг регистрации сервиса как OAPI backend (по умолчанию: `false`);
 - `registration_path` – путь регистрации сервиса в ZooKeeper (по умолчанию: `/ps/instances/apps/mds/oapi_mds_backend`);
 - `registration_host` – хост, указываемый при регистрации сервиса в ZooKeeper (по

- умолчанию: значение переменной `ansible_host`);
- `connection_string` – строка соединения с ZooKeeper (по умолчанию: `localhost:2181`);
- `retry_interval_ms` – тайм-аут на соединение с ZooKeeper (по умолчанию: 5000);
- `ldap`:
 - `enabled` – флаг включения/отключения функциональности LDAP;
 - `url` – URL LDAP-сервера;
 - `login_suffix` – постоянная часть логина;
 - `base_dn` – основной DN для поиска сотрудников;
 - `search_filter` – фильтр для определения сотрудников (по умолчанию: `"(&(objectClass=user)(sAMAccountName=%s))"`);
 - `attribute_firstname` – имя пользователя (по умолчанию: `"givenName"`);
 - `attribute_secondname` – фамилия пользователя (по умолчанию: `"sn"`);
 - `attribute_mail` – адрес электронной почты (по умолчанию: `"mail"`);
 - `attribute_department` – отдел, где работает пользователь (по умолчанию: `"department"`);
 - `attribute_company` – компания, где работает пользователь (по умолчанию: `"company"`);
 - `attribute_title` – должность пользователя (по умолчанию: `"title"`);
 - `attribute_membership` – группы, в которых состоит пользователь (по умолчанию: `"memberOf"`);
 - `attribute_photo` – данные фотографии пользователя (по умолчанию: `""`);
- `sso`:
 - `enabled` – включение/отключение авторизации пользователей (по умолчанию: `false`);
 - `server_login` – сервер SSO, который отвечает на запрос токена пользователя (по умолчанию: `"localhost:47141"`);
 - `server_grants` – сервер SSO, который по токену пользователя возвращает его права (по умолчанию: `"localhost:47180"`);
 - `server_attributes` – сервер SSO, который по логину пользователя возвращает его атрибуты (по умолчанию: `"localhost:47141"`);
 - `trusted_admin` – имя пользователя, которое используется при запросе атрибутов в режиме `sso_trusted_channel` (по умолчанию: `"Admin_super"`);
 - `login` – логин пользователя SSO (по умолчанию: `"Admin"`);
 - `password` – пароль пользователя SSO (по умолчанию: `"1111"`).

ps.mds.lnx

- `lnx.host` – адрес сервера лицензий (по умолчанию: `"localhost"`);
- `lnx.port` – порт для соединения с сервером лицензий (по умолчанию: `"10443"`);
- `lnx.product` – название лицензируемого продукта;
- `lnx.application` – название лицензируемого компонента продукта; может быть пустым, если у продукта один компонент;
- `lnx.version` – версия продукта;
- `lnx.licensePath` – относительный или абсолютный путь к файлу лицензий (по умолчанию: `"./lnx/license"`).

ps.mds.encryptor

- `encryption_key` – ключ для шифрования паролей.

С.4. Параметры файла `dip_dispatcher.yml`

Настройки конфигурации в файле `inventory/<схема развертывания>/group_vars/dip_dispatcher.yml`:

- `app` – блок конфигурирования каталогов компонента:
 - `install_dir` – путь к корневому каталогу приложения компонента;
 - `external_dirs.logs` – каталог, в котором будут располагаться журнальные файлы компонента.

С.5. Параметры файла `dip_web.yml`

Настройки конфигурации в файле `inventory/<схема развертывания>/group_vars/dip_web.yml`:

- `app` – блок конфигурирования каталогов компонента:
 - `install_dir` – путь к корневому каталогу приложения компонента;
 - `external_dirs.logs` – каталог, в котором будут располагаться журнальные файлы компонента.