СИСТЕМА УПРАВЛЕНИЯ КАРТАМИ ОПЛАТЫ

Руководство по установке

Версия 1.2.0

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

Chapter 1. Предварительные условия

Для корректной установки и работы VMS установите следующие приложения:

- SQL-база данных для хранения информации PostgreSQL версии 13 и выше;
- РЕД ОС;
- «Аутентификация пользователей по технологии единого входа» (SSO);
- Apache ZooKeeper версии 3.5 или выше;
- брокер сообщений Kafka версии 2.13-3.1.0 или выше;
- Kubernetes версии 1.19.0-0 или выше;
- Helm (в комплекте с Tiller) версии 3 или выше.

Chapter 2. Установка и настройка с помощью автоинсталлятора

Для установки продукта выполните следующие действия:

1. Скачайте и разархивируйте инсталлятор из Artifactory. Файлы инсталлятора разместите в vms/install
2. Отредактируйте файл requirements.yml. В случае если включена обязательная авторизация для скачивания из Artifactory, файл requirements.yml должен выглядеть так:

```
- src: https://<USER>:<ENCRYPTED_PASSWORD>@<ARTIFACTORY_HOST>/
artifactory/<ARTIFACTORY_REPO>/ps/BIN/COMMON_INSTALLER_1/
<COMMON_INSTALLER_VERSION>/COMMON_INSTALLER/app/
common-installer-<COMMON_INSTALLER_VERSION>-common-installer
-<COMMON_INSTALLER_VERSION>-app.tar.gz
name: ./
```

   Иначе в файле requirements.yml должно быть:

```
- src: https://<ARTIFACTORY_HOST>/artifactory/<ARTIFACTORY_REPO>/
ps/BIN/COMMON_INSTALLER_3/<COMMON_INSTALLER_VERSION>
/COMMON_INSTALLER/app/common-installer-<COMMON_INSTALLER_VERSION>
-common-installer-<COMMON_INSTALLER_VERSION>-app.tar.gz
name: ./
```

   где:

   - `<USER>` – пользователь для входа в 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;
   - `<COMMON_INSTALLER_VERSION>` – требуемая версия `COMMON_INSTALLER`.
3. Создайте схему развертывания.
4. Настройте общие параметры.
5. Настройте хосты.
6. Выполните конфигурацию продукта.
7. Запустите скрипт ansible-prepare.sh. Из Artifactory будет скачан дистрибутив с общими ролями продукта `COMMON_INSTALLER` (подробное описание общих ролей `COMMON_INSTALLER` см. в документе «Руководство по эксплуатации» [COMMON_INSTALLER-DOC_G3]).
8. Запустите автоинсталлятор.

2.1. Создание схемы 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).

2.2. Настройки общих параметров установки

1. Задайте местоположение запускаемых компонентов, файлов журналирования, мониторинга в файле inventory/<схема_развертывания>/group_vars/all.yml (описание параметров см. в разделе «Конфигурация продукта»).
2. При необходимости переопределите общие настройки установки каждого компонента и настройки конфигурации в каталоге inventory/<схема_развертывания>/group_vars/<component_name>.yml (описание параметров см. в разделах «Настройки файла <component_name>.yml»).

Подробнее о настройках inventory см. документацию Ansible.

2.3. Файлы Playbook

Для установки продукта используются файлы playbook:

- vms-db-deploy.yml – установка схем и серверных частей;
- vms-deploy-full.yml – полная установка продукта;
- vms-k8s-deploy.yml – установка компонентов в Kubernetes;
- vms-sso-deploy.yml – установка ролевой части в SSO;
- vms-validate-parameters.yml – проверка заполнения конфигурации, на которой выполняется установка продукта в соответствии со схемой;
- vms-zookeeper-deploy.yml – сохранение настроек backend-модуля в ZooKeeper.

2.4. Настройка хостов

Настройте хосты сервера, на котором устанавливается продукт, и опишите группы серверов схемы развертывания в файле inventory/localhost/1-digital_bss-vms.

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:

```
vms ansible_connection=ssh ansible_host=<адрес удаленной машины>  //
Развертывание на удаленную машину
# Либо
vms ansible_connection=local ansible_host=localhost // Развертывание на
localhost
```

Файл inventory/localhost/1-digital_bss-vms по умолчанию заполнен параметрами для локальной установки и в минимальной конфигурации. Нельзя изменять названия существующих групп. Можно добавлять новые хосты, новые группы и наследования.

2.5. Запуск автоинсталлятора

Общий вид команды запуска:

```
ansible-playbook vms-deploy-full.yml -i inventory/localhost -u <user>
-k
```

где:

- `-i <inventory/localhost>` – inventory, который будет использоваться при выполнении сценария (если указать каталог, то сценарий пройдется по всем хост-файлам, находящимся 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`.

При необходимости можно добавить дополнительные параметры:

- `--become-user` – привилегированный пользователь;
- `-K` – необходимость ввода пароля для привилегированного пользователя, указанного в ключе `--become-user`;
- `-b` – необходимость выполнения сценария от имени привилегированного пользователя;
- `-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;
- `-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.

2.6. Установка продуктов из локального репозитория

1. Скачайте и разложите дистрибутив продукта согласно одной из следующих структур:
   - <группа продуктов>/<продукт>/<компонент>/<app|conf|mib|info>;
   - <группа продуктов>/<продукт>/<версия>/<компонент>/<app|conf|mib|info>;
   - <компонент>/<app|conf|mib|info>.
2. Скачайте и разархивируйте инсталлятор.
3. Создайте в корне инсталлятора каталог common.
4. Скачайте и разархивируйте `COMMON_INSTALLER` в каталог common. Необходимая версия `COMMON_INSTALLER` указывается в файле requirements.yml.
5. Дальнейшие действия совпадают с последовательностью установки продукта, за исключением следующих необходимых изменений:
   - в inventory/group_vars/all.yml:

```
artifactory:
  enable: false ①
  url: "https://artifactory.nexign.com/artifactory"
```

```
  search_url:
"https://artifactory.nexign.com/artifactory/api/search/aql"
  api_key: "<API_KEY>"
  repo: "QA"
local_storage:
  enable: true ②
  root: "/home/user/product_distrib"
```

① Выключение работы с Artifactory
② Включение работы с локальным архивом

- в inventory/group_vars/<component_name>.yml добавить:

```
local_storage:
  path: "*PRODUCT_GROUP*PRODUCT_NAME*"
```

Параметр `local_storage.path` должен быть описан в файле inventory/group_vars/<component_name>.yml. При этом необходимо учитывать, что в `root` должна соблюдаться одна из структур, которые указаны в п.1. Параметр `local_storage.path` должен быть прописан для каждого компонента продукта.

2.7. Установка продуктов в Kubernetes

Существует два типа установки в Kubernetes с помощью инсталлятора:

- установка с помощью k8s (Kubernetes) манифестов;
- установка с помощью helm charts.

В обоих случаях для установки продуктов:

1. На контрольной машине ansible необходимо иметь конфигурационный файл кластера Kubernetes (kubeconfig).
2. В inventory укажите путь к файлу kubeconfig в параметре `env.KUBERCONFIG`:

```
env:
    KUBECONFIG: /path/to/kubeconfig
```

   Либо на контрольной машине ansible определите переменную окружения KUBECONFIG.

3. В inventory для продукта определите параметр `k8s.namespace`. В значении укажите имя namespace, в который будет установлен продукт. Подробнее смотрите документацию Kubernetes.

Для установки продуктов с помощью helm:

- либо добавьте репозиторий, где располагаются helm chart в helm с помощью команды:

```
helm repo add <repo_name> <repo_url> --username <user> --password
<password> && helm repo update
```

- либо указывайте url репозитория каждый раз при установки чарта. В этом случае в inventory:
  - в параметре `chart.source.type` укажите значение `repo`;
  - в параметре `chart.source.location` укажите url репозитория; подробнее см. описание роли `k8s/helm/install` в документе «Руководство по настройке» продукта COMMON_INSTALLER [COMMON_INSTALLER-DOC_CFG].

Для установки используйте файлы Playbook vs-k8s-deploy.yml.

2.8. Конфигурация продукта

Конфигурация продукта задается в файлах:

- inventory/<имя_схемы_развертывания>/group_vars/all.yml;
- inventory/<имя_схемы_развертывания>/group_vars/vms.yml;
- inventory/<имя_схемы_развертывания>/group_vars/k8s_vms.yml.

В Inventory находятся настройки, которые могут изменяться. Эти настройки зависят от окружения, на которое будет устанавливаться продукт.

2.8.1. Настройки файла all.yml

В файле inventory/group_vars/all.yml задайте параметры:

- `artifactory` – настройки Artifactory:
  - `url` – адрес хранилища Artifactory (например: http://example.com/artifactory);
  - `search_url` – адрес API Artifactory (например: http://example.com/artifactory/api/search/aql);
  - `api_key` – ключ доступа пользователя Artifactory;
  - `repo` – репозиторий артефактов (например: "QA, ReleaseCandidate");
  - `enable` – включение работы с Artifactory;
- `local_storage` – настройки локального архива:
  - `enable` – включение работы с локальным архивом;
  - `root` – путь до дистрибутива;
- `sso` – настройки `SSO`:
  - `host` – dns-имя или IP-адрес `SSO`;
  - `port` – порт, на котором работает основной интерфейс `SSO`;
- `sso_api` – настройки sso_api:
  - `enable` – флаг активации регистрации ролевой модели в `SSO` (значение по умолчанию – `false`);
  - `user` – имя пользователя для авторизации на API SSO;
  - `password` – пароль пользователя для авторизации на API SSO;
  - `host` – dns-имя или IP-адрес API SSO;
  - `port` – порт, на котором работает API SSO;
  - `appl_user` – технологический пользователь `SSO`, необходимый для авторизации при получении токена на SSO API;
  - `appl_password` – пароль технологического пользователя `appl_user`;
- `secret_key`: "`secretKey`" – ключ для шифрования пароля;
- `salts_secret_key` – ключ для хеширования модификаторов;

Внимание!

Заполнить значением, которым захешированы модификаторы в базе данных (получить текущий ключ из защищенного хранилища Kubernetes (`secret`) штатными средствами Kubernetes).

Если модификаторы еще не созданы, заполнить ключом, сгенерированным с помощью команды

```
./voucher-crypt-util.sh  generateKey --file key.txt
```

утилиты `voucher-crypt-util`.

- `zoo_host` – адрес ZooKeeper (`{{ zookeeper.host }}:{{ zookeeper.port }}`);
- `zookeeper`:
  - `host` – dns-имя или IP-адрес ZooKeeper;
  - `port` – порт ZooKeeper;
- `kafka` – настройки Kafka;
  - `host` – dns-имя или IP-адрес Kafka;
  - `broker_port` – порт Kafka;
- `docker` – настройки docker:
  - `registry` – адрес хранилища docker-образов;
- `k8s` – настройки Kubernetes:
  - `namespace` – пространство имен k8s, в которое будет установлен продукт;
  - `serviceaccountname` – наименование `ServiceAccount` (объект разграничения прав доступа);
  - `kubeconfig_path` – путь до файла конфигурации admin.conf;
  - `image_pull_secrets` – ключ для аутентификации пользователя;
- `chart` – настройки Helm chart:
  - `version` – версия chart (по умолчанию совпадает с версией продукта);
  - `source` – настройки Helm chart:
    - `type` – тип chart (значение по умолчанию – `alias`);
    - `location` – адрес расположения хранилища chart;
    - `username` – имя пользователя для авторизации в хранилище chart;
    - `password` – пароль пользователя для авторизации в хранилище chart;
- `sftp` – настройки соединения с SFTP-сервером:
  - `host` – dns-имя или IP-адрес SFTP-сервера;
  - `password` – пароль пользователя для авторизации на SFTP-сервере;
  - `port` – порт SFTP-сервера;
  - `username` – имя пользователя для авторизации на SFTP-сервере;
  - `rootFilePath` – путь до корневого каталога для формирования файлов с ваучерами.

2.8.2. Настройки файла vms.yml

Настройки конфигурации продукта в файле inventory/group_vars/vms.yml:

- `db` – настройки базы данных;
  - `username` – имя пользователя для подключения к базе данных;
  - `password` – пароль пользователя для подключения к базе данных;

- enc_password – закодированный пароль пользователя для подключения к базе данных;
- host – dns-имя или IP-адрес базы данных;
- port – порт базы данных;
- default_database – название базы данных;
- schema_name – название схемы данных;

- ps:
  - vms:
    - db – настройки доступа к базе данных для установки объектов базы данных (таблицы, процедуры, и пр.)
      - username – имя пользователя для подключения к базе данных;
      - password – пароль пользователя для подключения к базе данных;
      - driver – обработчик соединений;
      - url – адрес для подключения к базе данных;
      - schema_name – название схемы данных;
      - changelog_params – прочие параметры журнала записей событий;
        - author – автор записи;
      - log_dir – репозиторий для хранения записей журнала;
      - log_level – уровень журналирования соединений;
      - salts:
        - init_salts – указывает на необходимость заполнить таблицу модификаторов с ключом из параметра salts_secret_key из файла all.yml, значение по умолчанию – true;
        - length – длина генерируемых модификаторов; значение по умолчанию – 128;
        - count – количество генерируемых модификаторов; значение по умолчанию – 512;
    - zk:
      - activation_service – настройки сервиса activation-service;
        - db – настройки соединения к базе данных;
          - idleTimeout – тайм-аут сессии в пуле к базе данных для микросервиса activation-service;
          - jdbcUrl – строка соединения с базой данных для микросервиса activation-service;
          - maximumPoolSize – максимальное количество соединений в пуле базы данных для микросервиса activation-service;
          - minimumIdle – минимальное количество сессий в пуле для базы данных микросервиса activation-service;
          - password – пароль пользователя базы данных микросервиса activation-service;
          - username – имя пользователя базы данных микросервиса activation-service;
      - common – общие настройки продукта:
        - activation – настройки активации по неполному коду;
          - activateCodeUnknownSymbol – символ, который при вводе неполного кода активации означает неизвестный символ кода;
          - activateCodeUnknownSymbolsQuantity – максимальное разрешенное количество неизвестных символов кода активации в случае активации по части кода;
        - crab:
          - password – пароль технологического пользователя CRAB;
          - techUser – имя технологического пользователя CRAB;

- `db` – настройки по умолчанию для соединения с базой данных, при отсутствии настроек на уровне самого микросервиса;
  - `idleTimeout` – тайм-аут сессии в пуле к базе данных, общая для всех микросервисов, в мс;
  - `jdbcUrl` – строка соединения с базой данных для всех микросервисов;
  - `maximumPoolSize` – максимальное количество соединений в пуле базы данных, общее для всех микросервисов;
  - `minimumIdle` – минимальное количество сессий в пуле для базы данных, общее для всех микросервисов;
  - `password` – пароль пользователя в пуле для базы данных, общий для всех микросервисов;
  - `username` – имя пользователя в пуле для базы данных, общее для всех микросервисов;
- `externalDictionaries`:
  - `host` – dns-имя или IP-адрес OAPI-сервера внешних справочников;
  - `manufacturers` – относительный путь в URL OAPI для внешних справочников производителей;
  - `password` – пароль технологического пользователя для получения данных внешних справочников;
  - `port` – порт OAPI-сервера внешних справочников;
  - `productOffers` – относительный путь в URL OAPI для внешних справочников продуктовых предложений;
  - `storePoints` – относительный путь в URL OAPI для внешних справочников точек хранения;
  - `techUser` – имя технологического пользователя для получения данных внешних справочников;
- `grpc` – порт приложения для взаимодействия по gRPC интерфейсу:
  - `idleTimeoutSec` – тайм-аут (в мс) ожидания ответа для gRPC функций (значение по умолчанию – `3600`);
- kafka:
  - `bootstrap_servers` – строка соединения с кластером Kafka вида <host>:<port> (значение по умолчанию – `<kafka.host>:<kafka.broker_port>`); в случае установки Kafka внутри кластера k8s, допускается указание в виде `<service-name>.<namespace-name>.svc:<port>`;
  - `maxPollIntervalMs` – максимальное время (в мс) в течение которого consumer в Kafka может бездействовать, прежде чем извлекать дополнительные записи;
- `oapi`:
  - `host` – dns имя или IP-адрес OAPI;
  - `port` – порт OAPI;
  - `sso`:
    - `adminPassword` – пароль пользователя, от имени которого вызывается сервис для получения токена доступа;
    - `adminUsername` – пользователь, от имени которого вызывается сервис для получения токена доступа;
    - `host` – dns имя или IP-адрес SSO (по умолчанию `<sso.host>` из файла all.yml);
    - `port` – порт, на котором работает основной интерфейс SSO (значение по умолчанию – `<sso.port>` из файла all.yml);
    - `tokenExpirePeriod` – срок действия токена SSO (в мс);

- paymentManagement:
  - techUser – имя технологического пользователя PM;
  - password – пароль технологического пользователя PM;
- pin:
  - hash:
    - algorithm – код алгоритма хэширования кода активации;
  - length – длина кода активации;
  - pattern – шаблон для кода активации ваучера;
- secretKey – ключ шифрования паролей (значение по умолчанию – <secretKey> из файда all.yml);
- serialNumber:
  - numberLength – длина номера для серийного номера;
  - seriesLength – длина серии для серийного номера;
- serverTimeZone – часовой пояс (time zone) сервера;

- filemaker_service – настройки сервиса filemaker-service:
  - file:
    - csv:
      - fieldSeparator – разделитель записей в файле;
      - writeHeader – параметр, определяющий наличие заголовка в файле;
    - dirForTmpFiles – каталог, где будут создаваться временные файлы с ваучерами для выгрузки; если filemaker-service развернут как микросервис (в кластере k8s), значение невозможно изменить – /vs/files;
    - grpcRecordsBatchSize – количество записей в пачке ваучеров при формировании файлов для выгрузки;
    - maxRecordsInFile – максимальное количество записей в выгружаемом файле с ваучерами;
  - sftp:
    - host – dns имя или IP-адрес SFTP-сервера (значение по умолчанию – <sftp.host> из файла all.yml);
    - password – пароль пользователя для авторизации на SFTP-сервере (значение по умолчанию – <sftp.password> из файла all.yml);
    - port – порт SFTP-сервера (значение по умолчанию – <sftp.port> из файла all.yml);
    - rootFilePath – путь до корневого каталога для формирования файлов с ваучерами (значение по умолчанию – <sftp.rootFilePath> из файла all.yml);
    - username – имя пользователя для авторизации на SFTP-сервере (значение по умолчанию – <sftp.username> из файла all.yml);
- management_service – настройки сервиса management-service:
  - db – настройки соединения к базе данных;
    - idleTimeout – тайм-аут сессии в пуле к базе данных для микросервиса management-service;
    - jdbcUrl – строка соединения с базой данных для микросервиса management-service;
    - maximumPoolSize – максимальное количество соединений в пуле базы данных для микросервиса management-service;
    - minimumIdle – минимальное количество сессий в пуле для базы данных микросервиса management-service;

        - password – пароль пользователя базы данных микросервиса management-service;
        - username – имя пользователя базы данных микросервиса management-service;
    - oapi_service – настройки сервиса voucher-server-oapi:
        - current_version – текущая версия устанавливаемого продукта (значение по умолчанию – <artifactory.version> из group_vars/vms.yml);
        - defaultLanguage – язык по умолчанию;
        - defaultRecordsForList – количество записей по умолчанию, возвращаемое списковыми OAPI-функциями;
        - maxRecordsForList – максимальное количество записей по умолчанию, возвращаемое списковыми OAPI-функциями;
    - storage_service – настройки сервиса storage-service:
        - selectOfPinAddedPercent – буфер дополнительных хэш-кодов; параметр задает величину в процентах на сколько больше надо сгенерировать дополнительных хэш-кодов на случай возникновения коллизий; значение по умолчанию – 10;
        - selectOfPinMinStep – минимальное количество хэш-кодов, которое будет сгенерировано на случай возникновения коллизий. Таким образом, будет сгенерировано количество хэш-кодов по параметру selectOfPinAddedPercent, но не меньше, чем в параметре selectOfPinMinStep; значение по умолчанию – 1;
        - selectOfPinMaxAttempts – максимальное число попыток получения дополнительных хэш-кодов; значение по умолчанию – 3;
        - db - настройки соединения к базе данных;
            - idleTimeout – тайм-аут сессии в пуле к базе данных для микросервиса storage-service;
            - jdbcUrl – строка соединения с базой данных для микросервиса storage-service;
            - maximumPoolSize – максимальное количество соединений в пуле базы данных для микросервиса storage-service;
            - minimumIdle – минимальное количество сессий в пуле для базы данных микросервиса storage-service;
            - password – пароль пользователя базы данных микросервиса storage-service;
            - username – имя пользователя базы данных микросервиса storage-service.

2.8.3. Настройки файла k8s_vms.yml

Настройки конфигурации продукта в файле inventory/group_vars/k8s_vms.yml:

- ps:
    - vms:
        - k8s – настройки установки сервисов в k8s кластер;
            - global:
                - image:
                    - repository – адрес хранилища docker-образов (значение по умолчанию – <docker.registry> из файла all.yml);
                - zookeeper – адрес ZooKeeper;
                    - host – dns имя или ip-адрес ZooKeeper (по умолчанию <zookeeper.host> из all.yml); в случае установки zookeeper внутри кластера k8s допускается указание в виде <service-name>.<namespace-name>.svc;

- `port` – порт ZooKeeper (значение по умолчанию – `<zookeeper.port>` из файла all.yml);

- `generic`:
  - `serviceAccountName` – serviceAccount с которым будут ассоциированы все разворачиваемые микросервисы (совпадает с именем продукта); значение по умолчанию – `<k8s.serviceaccountname>` из all.yml;
- `voucher_activation` – настройки микросервиса voucher-activation:
  - `secrets`:
    - `servicesecretenvvars`:
      - `data`:
        - `secretKey:{{ secret_key }}` – используется для дешифрования пароля для базы данных в поде, который лежит в zk – `{{ db.enc_password }}`;
    - `deployments`:
      - `resources` – ограничения, накладываемые на использование Pod CPU и памяти (RAM). CPU измеряется в единицах ядра, а память – в байтах;
        - `limits` – определяет максимальные ресурсы, которые может запросить docker container в ходе своей работы:
          - `cpu` – `800m`;
          - `memory` – `1024Mi`;
          - `ephemeral_storage` – `256Mi`;
        - `requests` – определяет ограничение ресурсов, выдаваемых для docker container при его установке:
          - `cpu` – `400m`;
          - `memory` – `512Mi`;
          - `ephemeral_storage` – `128Mi`;
      - `livenessProbe`:
        - `initialDelaySeconds` – задержка перед выполнением первой проверки, значение по умолчанию `20`;
        - `periodSeconds` – период между проверками; значение по умолчанию – `10`;
        - `failureThreshold` – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – `5`;
      - `readinessProbe` – параметры запроса, определяющего готовность приложения принимать трафик на основном порту:
        - `initialDelaySeconds` – задержка перед выполнением первой проверки; значение по умолчанию – `10`;
        - `periodSeconds` – период между проверками; значение по умолчанию – `5`;
        - `failureThreshold` – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – `3`;
      - `service`:
        - `type` – тип сервиса; значение по умолчанию – `ClusterIP`;
      - `ports` – настройка портов микросервиса:
        - `internal` – настройка портов приложения в рамках кластера k8s;
          - `srv` – порт приложения для взаимодействия по http интерфейсу; значение по умолчанию – `28091`;
          - `mgmt` – порт приложения для выполнения служебных запросов; значение по умолчанию – `8051`;

nexign

        - `grpc` – порт приложения для взаимодействия по gRPC интерфейсу; значение по умолчанию – `9091`;
    - `log_lvl` – уровень формирования логов приложения; поддерживаются уровни ERROR, WARN, INFO, DEBUG, TRACE; значение по умолчанию – `INFO`;
    - `diagnosticMode`:
        - `enabled` – режим отладки; доступные значения: `true`, `false`;значение по умолчанию – `false`; при значении `true` контейнер запускается с командой «sleep» и аргументом «infinity»;
- `voucher_filemaker`:
  - `secrets`:
    - `servicesecretenvvars`:
      - `data`:
        - `secretKey:{{ secret_key }}` – используется для дешифрования пароля для базы данных в поде, который лежит в zk – `{{ db.enc_password }}`;
    - `deployments`:
      - `resources` – ограничения, накладываемые на использование Pod CPU и памяти (RAM). CPU измеряется в единицах ядра, а память – в байтах;
        - `limits` – определяет максимальные ресурсы, которые может запросить docker container в ходе своей работы:
          - `cpu` – `800m`;
          - `memory` – `1024Mi`;
          - `ephemeral_storage` – `256Mi`;
        - `requests` – определяет ограничение ресурсов, выдаваемых для docker container при его установке:
          - `cpu` – `400m`;
          - `memory` – `512Mi`;
          - `ephemeral_storage` – `128Mi`;
      - `livenessProbe`:
        - `initialDelaySeconds` – задержка перед выполнением первой проверки; значение по умолчанию – `20`;
        - `periodSeconds` – период между проверками; значение по умолчанию – `10`;
        - `failureThreshold` – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – `5`;
      - `readinessProbe` – параметры запроса, определяющего готовность приложения принимать трафик на основном порту:
        - `initialDelaySeconds` – задержка перед выполнением первой проверки; значение по умолчанию – `10`;
        - `periodSeconds` – период между проверками; значение по умолчанию – `5`;
        - `failureThreshold` – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – `3`;
      - `service`:
        - `type` – тип сервиса; значение по умолчанию – `ClusterIP`;
      - `ports` – настройка портов микросервиса;
        - `internal` – настройка портов приложения в рамках кластера k8s;
          - `srv` – порт приложения для взаимодействия по http интерфейсу;

значение по умолчанию – 28093;
- mgmt – порт приложения для выполнения служебных запросов; значение по умолчанию – 8053;

- log_lvl – уровень формирования логов приложения; поддерживаются уровни ERROR, WARN, INFO, DEBUG, TRACE; значение по умолчанию – INFO;
- diagnosticMode:
  - enabled – режим отладки; доступные значения: true, false; значение по умолчанию false); при значении true контейнер запускается с командой «sleep» и аргументом «infinity»;

- voucher_management:
  - secrets:
    - servicesecretenvvars:
      - data:
        - secretKey:{{ secret_key }} – используется для дешифрования пароля для базы данных в поде, который лежит в zk – {{ db.enc_password }};
    - deployments:
      - resources – ограничения, накладываемые на использование Pod CPU и памяти (RAM). CPU измеряется в единицах ядра, а память – в байтах;
        - limits – определяет максимальные ресурсы, которые может запросить docker container в ходе своей работы:
          - cpu – 800m;
          - memory – 1024Mi;
          - ephemeral_storage – 256Mi;
        - requests – определяет ограничение ресурсов, выдаваемых для docker container при его установке:
          - cpu – 400m;
          - memory – 512Mi;
          - ephemeral_storage – 128Mi;
      - livenessProbe:
        - initialDelaySeconds – задержка перед выполнением первой проверки; значение по умолчанию – 20;
        - periodSeconds – период между проверками; значение по умолчанию – 10;
        - failureThreshold – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – 5;
      - readinessProbe – параметры запроса, определяющего готовность приложения принимать трафик на основном порту:
        - initialDelaySeconds – задержка перед выполнением первой проверки; значение по умолчанию – 10;
        - periodSeconds – период между проверками; значение по умолчанию – 5;
        - failureThreshold – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – 3;
      - service:
        - type – тип сервиса; значение по умолчанию – ClusterIP;
      - ports – настройка портов микросервиса:
        - internal – настройка портов приложения в рамках кластера k8s;

        - srv – порт приложения для взаимодействия по http интерфейсу; значение по умолчанию – 28092;
        - mgmt – порт приложения для выполнения служебных запросов; значение по умолчанию – 8052;
        - grpc – порт приложения для взаимодействия по gRPC интерфейсу; значение по умолчанию – 9092;
      - log_lvl – уровень формирования логов приложения; поддерживаются уровни ERROR, WARN, INFO, DEBUG, TRACE; значение по умолчанию – INFO;
      - diagnosticMode:
        - enabled – режим отладки; доступные значения: true, false; значение по умолчанию – false; при значении true контейнер запускается с командой «sleep» и аргументом «infinity»;
- voucher_server_oapi:
  - secrets:
    - servicesecretenvvars:
      - data:
        - secretKey:{{ secret_key }} – используется для дешифрования пароля для базы данных в поде, который лежит в zk – {{ db.enc_password }};
    - deployments:
      - resources – ограничения, накладываемые на использование Pod CPU и памяти (RAM). CPU измеряется в единицах ядра, а память – в байтах;
        - limits – определяет максимальные ресурсы, которые может запросить docker container в ходе своей работы:
          - cpu – 800m;
          - memory – 1024Mi;
          - ephemeral_storage – 256Mi;
        - requests – определяет ограничение ресурсов, выдаваемых для docker container при его установке:
          - cpu – 400m;
          - memory – 512Mi;
          - ephemeral_storage – 128Mi;
      - livenessProbe:
        - initialDelaySeconds – задержка перед выполнением первой проверки, значение по умолчанию 20;
        - periodSeconds – период между проверками; значение по умолчанию – 10;
        - failureThreshold – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – 5;
      - readinessProbe – параметры запроса, определяющего готовность приложения принимать трафик на основном порту:
        - initialDelaySeconds – задержка перед выполнением первой проверки; значение по умолчанию – 10;
        - periodSeconds – период между проверками; значение по умолчанию – 5;
        - failureThreshold – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – 3;
      - service:

- type – тип сервиса; значение по умолчанию – ClusterIP;
- ports – настройка портов микросервиса:
  - internal – настройка портов приложения в рамках кластера k8s;
    - srv – порт приложения для взаимодействия по http интерфейсу; значение по умолчанию – 28090;
    - mgmt – порт приложения для выполнения служебных запросов; значение по умолчанию – 8050;
  - log_lvl – уровень формирования логов приложения; поддерживаются уровни ERROR, WARN, INFO, DEBUG, TRACE; значение по умолчанию – INFO;
  - diagnosticMode:
    - enabled – режим отладки; доступные значения: true, false; значение по умолчанию – false; при значении true контейнер запускается с командой «sleep» и аргументом «infinity»;

- voucher_storage:
  - secrets:
    - servicesecretenvvars:
      - data:
        - secretKey:{{ secret_key }} – используется для дешифрования пароля для базы данных в поде, который лежит в zk – {{ db.enc_password }};
        - env_salt_secretkey:"{{ salts_secret_key }}" – ключ для хеширования модификаторов по умолчанию берется значение из параметра salts_secret_key в файле all.yml;
    - deployments:
      - resources – ограничения, накладываемые на использование Pod CPU и памяти (RAM). CPU измеряется в единицах ядра, а память – в байтах;
        - limits – определяет максимальные ресурсы, которые может запросить docker container в ходе своей работы:
          - cpu – 800m;
          - memory – 1024Mi;
          - ephemeral_storage – 256Mi;
        - requests – определяет ограничение ресурсов, выдаваемых для docker container при его установке:
          - cpu – 400m;
          - memory – 512Mi;
          - ephemeral_storage – 128Mi;
      - livenessProbe:
        - initialDelaySeconds – задержка перед выполнением первой проверки; значение по умолчанию – 20;
        - periodSeconds – период между проверками; значение по умолчанию – 10;
        - failureThreshold – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – 5;
      - readinessProbe – параметры запроса, определяющего готовность приложения принимать трафик на основном порту:
        - initialDelaySeconds – задержка перед выполнением первой проверки; значение по умолчанию – 10;

    - `periodSeconds` – период между проверками; значение по умолчанию – `5`;
    - `failureThreshold` – количество неудачных попыток, после чего контейнер будет перезапущен; значение по умолчанию – `3`;
  - `service`:
    - `type` – тип сервиса; значение по умолчанию – `ClusterIP`;
  - `ports` – настройка портов микросервиса:
    - `internal` – настройка портов приложения в рамках кластера k8s;
      - `srv` – порт приложения для взаимодействия по http интерфейсу; значение по умолчанию – `28094`;
      - `mgmt` – порт приложения для выполнения служебных запросов; значение по умолчанию – `8054`;
      - `grpc` – порт приложения для взаимодействия по gRPC интерфейсу; значение по умолчанию – `9090`;
  - `log_lvl` – уровень формирования логов приложения; поддерживаются уровни ERROR, WARN, INFO, DEBUG, TRACE; значение по умолчанию – `INFO`;
  - `diagnosticMode`:
    - `enabled` – режим отладки; доступные значения: `true`, `false`; значение по умолчанию – `false`; при значении `true` контейнер запускается с командой «sleep» и аргументом «infinity».

2.9. Установка на базу данных PostgreSQL

Для установки `VMS` на базу данных PostgreSQL:

1. Поставьте базу данных PostgreSQL (рекомендуемая версия PostgreSQL – 13.1).
2. Создайте базу данных `VMS`.
3. Создайте пользователя с правами администратора и паролем.
4. От имени пользователя создайте схему базы данных для `VMS`, используя команды:

```
SET client_encoding = 'UTF8';
CREATE SCHEMA IF NOT EXISTS <schema_name>;
GRANT ALL ON SCHEMA <schema_name> TO vs;
ALTER ROLE vs SET SEARCH_PATH TO <schema_name>;
```

1. Установите значение `org.postgresql.Driver` в параметре driver.
2. Укажите хост и порт базы данных в параметре ps.<имя_компонента>.db.url (например: db.url=jdbc:postgresql://%хост базы данных%:5432/VMS).
3. Запустите установку.

2.9.1. Playbook group_vars

В playbook group_vars (каталог group_vars) хранятся настройки, которые не изменяются от стенда к стенду. Например, это могут быть команды запуска и остановки сервиса, путь к продукту в Artifactory.

Параметры playbook group_vars

- `artifactory.path` – путь к каталогу продукта в Artifactory; задавать через «*» в виде «*<product_group>*<product>*»;
- `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 }}`);
- `classifiers` – каталог в Artifactory в conf, из которого загружаются шаблоны (по умолчанию: `it`);
- `product_name` – имя продукта (используется для генерации файла <product_name>-json.schema);
- `ps.product.param` – параметры словаря ps.product, изменение которых не допускается.

Подробное описание Ansible.

2.9.2. Файл для проверки корректности заполнения inventory

В файле schemas/vs-schema.json в json-формате описаны параметры, которые необходимо задать в файле inventory/localhost/group_vars/vs.yml.

2.9.3. Теги

Доступные теги:

- <компонент> – установка и конфигурация одного компонента;
- check – проверка inventory на соответствие JSON-схеме;
- configure – установка конфигурационных файлов;
- zk – действия с ZooKeeper;
- helm – установка продуктов через Helm;
- sso_acl – загрузка ролей, ресурсов, иерархий в API SSO.
- deploy – установка всего продукта;
- stop – остановка всех компонентов продукта;
- start – запуск всех компонентов продукта;
- restart – перезапуск всех компонентов продукта;
- rollback – откат на предыдущую версию компонента;
- success – создание внутри версии компонента файла маркера DEPLOY_SUCCESS;
- finalize – переключение symlink current на последнюю версию, удаление промежуточных незавершенных версий, в которых присутствует файл DEPLOY_UNFINISHED;
- db – установка всей серверной части продукта на базу данных;
- <компонент_db> – установка серверной части одного компонента;
- install_packs – установка пакетов в схему данных.

2.9.4. Окружение Ansible

Пример заполнения файла ansible.cfg:

```
[defaults]
hash_behaviour = merge
roles_path = common
log_path = ./ansible_vs.log
stdout_callback = debug
```

```
jinja2_native = true

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

где:

- `hash_behaviour` – использовать или нет переопределение параметра;
- `roles_path` – путь до ролей Ansible, может быть перечислено несколько путей через ':';
- `log_path` – путь для записи логов Ansible (изменяемый параметр);
- `stdout_callback` – задает callback для вывода отладочной информации в удобном для пользователя виде при установке продукта;
- `ssh_args` – требование `COMMON_INSTALLER`.

Chapter 3. Ручная установка продукта

Для установки `VMS`:

- скопируйте и разархивируйте дистрибутив продукта;
- установите схему данных продукта;
- настройте брокер сообщений Kafka для взаимодействия с функциональными модулями продукта;
- задайте конфигурационные параметры продукта на сервере ZooKeeper;
- установите функциональные модули `VMS` с помощью Helm-чартов для Kubernetes на сервер с РЕД ОС.

Chapter 4. Запуск и остановка продукта

Продукт запускается автоматически после развертывания автоинсталлятором.

При необходимости запуска компонентов вручную используйте графический интерфейс Kubernetes (см. официальную документацию Kubernetes).

Последовательность запуска и остановки компонентов не имеет значения.

Chapter 5. Обновление

Чтобы обновить продукт, выполните установку новой версии.

Chapter 6. Проверка работоспособности

Для проверки установки продукта `VMS` убедитесь, что в log-файлах установочных скриптов нет сообщений об ошибках.

Работоспособность компонентов проверьте с помощью метрик мониторинга.