

# УПРАВЛЕНИЕ IOT- ПОДКЛЮЧЕНИЯМИ

Руководство по установке  
Версия 1.3.0

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», 1992–2024

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

# Содержание

<b>1. Подготовка к установке</b> . . . . .	<b>4</b>
1.1. Требования к внешнему окружению . . . . .	4
1.2. Требования к исполняемому окружению инсталлятора . . . . .	5
1.3. Шифрование паролей . . . . .	5
1.4. Настройка окружения Kubernetes . . . . .	5
<b>2. Установка</b> . . . . .	<b>8</b>
2.1. Подготовка дистрибутива . . . . .	8
2.2. Установка COMMON_INSTALLER . . . . .	8
2.3. Установка IOT_CM на хост . . . . .	9
2.4. Установка IOT_CM в Kubernetes . . . . .	11
2.5. Установка IOT_CM вручную . . . . .	13
<b>3. Управление версиями</b> . . . . .	<b>14</b>
3.1. Завершение установки версии . . . . .	14
3.2. Отметка версии как успешной при установке на хост . . . . .	14
3.3. Откат на последнюю успешную версию . . . . .	14
3.4. Удаление незавершенных установок . . . . .	15
<b>4. Запуск и остановка</b> . . . . .	<b>16</b>
<b>5. Обновление</b> . . . . .	<b>17</b>
<b>6. Проверка работоспособности</b> . . . . .	<b>18</b>
<b>Приложение А: Файлы playbook</b> . . . . .	<b>19</b>
<b>Приложение В: Конфигурация продукта</b> . . . . .	<b>20</b>

# 1. Подготовка к установке

Для установки продукта «Управление IoT-подключениями» (IOT\_CM):

1. Выполните требования [к внешнему окружению](#).
2. Выполните требования [к исполняемому окружению](#).
3. При необходимости настройте [шифрование паролей](#).
4. Если продукт устанавливается в Kubernetes, [настройте окружение Kubernetes](#).

## 1.1. Требования к внешнему окружению

### Аппаратное обеспечение

Рекомендуемые характеристики серверов для работы IOT\_CM представлены в [Табл. 1](#).

Табл. 1. Рекомендуемые характеристики серверов для работы IOT\_CM

Компонент	vCPU	RAM (GB)	HDD (GB)	Количество узлов	CPU Site	RAM Site	Схема резервирования	Горизонтальное масштабирование
Frontend Component 1	2	8	70	2	4	16	N+1	Да
DB PostgreSQL	4	8	64	2	8	16	1+1	Нет
Backend Component 1	2	8	50	2	4	16	N+1	Да
Elastic	6	64	200					

### Внешнее программное обеспечение

- РЕД ОС версии не ниже 7.3.1.
- Сервер сообщений RabbitMQ (RABBITMQ) версии 3.6 или более новой.
- Брокер сообщений Kafka версии 2.13-3.1.0 или более новой.
- Apache ZooKeeper версии 3.5 или более новой.
- Kubernetes версии 1.25.0 или более новой.
- SQL-база данных для хранения информации PostgreSQL версии 13 или более новой.

### Продукты Nexign

Для эксплуатации продукта требуется наличие ПО, поставляемого АО «Нэксайн»:

- «Общие роли инсталлятора» (COMMON\_INSTALLER).
- «Open API ESB» (OPENAPI\_ESB).
- «Аутентификация пользователей по технологии единого входа (SSO).
- «Центр обработки требований» (CRAB).
- «Система управления НСИ» (REFDATA).
- «Создание, управление и исполнение групповых операций» (BULK\_OPERATIONS).
- «Машина логического вывода» (EPM).
- «Управление подключением услуг» (SAM).

- «Система управления объектно-реляционной базой данных Nexign» (NORD).

Актуальные версии ПО см. в файле `releasenotes` (в секции `Dependencies`), поставляемом в составе дистрибутива.

## 1.2. Требования к исполняемому окружению инсталлятора

Для установки IOT\_CM требуется продукт COMMON\_INSTALLER.

Требования к исполняемому окружению см. в документации на продукт COMMON\_INSTALLER.

Актуальную версию COMMON\_INSTALLER см. в файле `releasenotes.txt`, поставляемом в составе дистрибутива.

Для установки в Kubernetes дополнительно установите:

- утилиту командной строки [kubectl](#);
- файл конфигурации доступа к кластерам [kubeconfig](#);
- пакетный менеджер [helm](#).

## 1.3. Шифрование паролей

Для повышения безопасности данных на площадках, где устанавливается продукт IOT\_CM, указывайте в конфигурационных файлах пароли в зашифрованном виде. Для этого получите зашифрованное значение пароля с помощью утилиты PasswordManager.

Пример использования:

```
java -jar PasswordManager-{version}.jar encrypt {SecretKey}
{StringToEncrypt}
```

## 1.4. Настройка окружения Kubernetes

Для установки IOT\_CM в Kubernetes настройте окружение.

1. Создайте Kubernetes Namespace, в который будет установлен продукт.

```
kubectl create namespace "<имя Namespace>"
```

Пример:

```
kubectl create namespace "iot_cm"
```

2. Обеспечьте доступ из Kubernetes Namespace к Docker Registry. Например, это можно сделать путем создания Kubernetes Secret и Service Account.

Создание Secret:

```
kubectl create secret docker-registry "<имя Kubernetes Secret>" \
  --namespace "<имя Namespace>" \
  --docker-server="<Docker registry url>" \
```

```
--docker-username="<Docker registry username>" \  
--docker-password="<Docker registry password>" \  
--docker-email="<Docker registry email>"
```

#### Пример:

```
kubectl create secret docker-registry "artifactory-registry" \  
  --namespace "iot_cm" \  
  --docker-server="http://my-artifactory.com/" \  
  --docker-username="my-name" \  
  --docker-password="my-password" \  
  --docker-email="my-name@my-org.com"
```

#### Создание Service Account с доступом к Docker Registry:

```
cat << EOF | kubectl apply -f -  
apiVersion: v1  
kind: ServiceAccount  
metadata:  
  name: "<Имя Service Account>"  
  namespace: "<Имя Namespace>"  
  annotations:  
    kubernetes.io/description: ServiceAccount with imagePullSecrets  
for access to Docker Registry.  
imagePullSecrets:  
  - name: "<Имя Kubernetes Secret>"  
EOF
```

#### Пример:

```
cat << EOF | kubectl apply -f -  
apiVersion: v1  
kind: ServiceAccount  
metadata:  
  name: "iot_cm"  
  namespace: "iot_cm"  
  annotations:  
    kubernetes.io/description: ServiceAccount with imagePullSecrets  
for access to Docker Registry.  
imagePullSecrets:  
  - name: "artifactory-registry"  
EOF
```

### Используемые параметры:

- `<имя Namespace>` – имя Kubernetes Namespace (как правило, совпадает с именем продукта);
- `<имя Service Account>` – имя Kubernetes Service Account (как правило, совпадает с именем продукта);
- `<Docker registry url>` – URL Docker Registry;
- `<Docker registry username>` – имя пользователя для доступа к Docker Registry;
- `<Docker registry password>` – пароль пользователя для доступа к Docker Registry;
- `<Docker registry email>` – e-mail пользователя для доступа к Docker Registry.

## 2. Установка

Установка IOT\_CM может быть выполнена:

- [на хост](#);
- [в кластер Kubernetes](#);
- [вручную](#).

### 2.1. Подготовка дистрибутива

Подготовьте дистрибутив, используя репозиторий Artifactory или локальный репозиторий.

#### Репозиторий Artifactory

Последовательно скачайте:

- скрипты для скачивания дистрибутива;
- общие роли для установки (с помощью скриптов);
- дистрибутив компонентов продукта.

Скачайте и разархивируйте каталог install из состава дистрибутива продукта. Файлы разместите в каталоге install на сервере, откуда выполняется установка. Если включена обязательная авторизация для скачивания из Artifactory, добавьте в файл requirements.yml строку:

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

где:

- <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\_url> – URL для скачивания архива с общими ролями.

#### Локальный репозиторий

Общие роли для установки и дистрибутив компонентов продукта предоставляются в едином архиве. Разархивируйте дистрибутив продукта. Настройте локальную установку – в файле inventory/group\_vars/all.yml задайте значения параметров:

- artifactory.enable = false;
- local\_storage.enable = true.

### 2.2. Установка COMMON\_INSTALLER

Установите COMMON\_INSTALLER с помощью команды:

```
./ansible-prepare.sh
```

## 2.3. Установка IOT\_CM на хост

Для установки продукта выполните действия (параметры для команд запуска приведены в [Табл. 2](#)):

1. Создайте схему развертывания. В каталоге дистрибутива `inventory/localhost` содержится пример с настройками схемы развертывания. Чтобы создать собственную схему развертывания, создайте копию каталога `localhost` и переименуйте её (например, `staging`).
2. Настройте хосты и опишите группы серверов схемы развертывания в файле `inventory/<схема_развертывания>/1-iotp-iot_cm`. В каждой группе укажите серверы, на которые необходимо установить компоненты продукта, и параметры соединения с ними в формате: `alias`, имя сервера `ansible_host` и метод подключения `ansible_connection`.

Примеры:

```
iot_cm ansible_connection=ssh ansible_host=<адрес удаленного сервера>
# развертывание на удаленный сервер
# Либо
iot_cm ansible_connection=local ansible_host=localhost #
развертывание на localhost
```

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

3. При необходимости переопределите настройки конфигурации в каталоге `inventory/<схема_развертывания>/group_vars` и файле `ansible.cfg`. Для проверки корректности заполнения параметров `inventory` запустите скрипт:

```
ansible-playbook -i inventory/<схема_развертывания> -e
"ansible_ssh_pass=пароль_пользователя
ansible_ssh_user=имя_пользователя" iot_cm-validate-parameters.yml
```

В файле `schemas/iot_cm-schema.json` в JSON-формате описаны параметры, которые необходимо задать в файле `inventory/<схема_развертывания>/group_vars/iot_cm/iot_cm.yml`.

4. Запустите установку, задав необходимые параметры:

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

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

Для установки определенной версии продукта или компонента при запуске установки задайте в параметре `--extra-vars (-e)` одну из следующих переменных:

- `version;`

- path\_version;
- <component>\_version;
- <component>\_path\_version,

где <component> – имя группы в файле хостов и соответствующий этому имени файл в каталоге group\_vars. В этом случае будет установлена указанная версия. Пример:

```
./auto-deploy.sh -i inventory/<схема_развертывания> -u -k --extra
-args="path_version=3.1.0"
```

## 5. Завершите установку.

Табл. 2. Параметры запуска

Параметр (ключ)	Описание
-i inventory/<схема_развертывания>	Каталог inventory, который будет использоваться при выполнении сценария (сценарий обрабатывает все файлы хостов данного каталога)
-u	Пользователь, от имени которого Ansible будет подключаться к серверам, указанным в файлах хостов в каталоге inventory
-k	Необходимость ввода пароля пользователя, указанного в ключе -u
--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): <ul style="list-style-type: none"> <li>• start – запуск;</li> <li>• stop – остановка;</li> <li>• finalize – переключение на новую версию;</li> <li>• restart – перезапуск;</li> <li>• db – установка серверной части;</li> <li>• rollback – откат на предыдущую версию;</li> <li>• success – добавление в текущую версию метки success (признак успешной установки версии);</li> <li>• configure – изменение конфигурации продуктов;</li> <li>• zookeeper – изменение конфигурации в ZooKeeper;</li> <li>• deploy – установка компонентов IOT_CM;</li> <li>• check – проверка inventory на соответствие JSON-схеме;</li> <li>• sso_acl – развертывание ролей продукта «Аутентификация пользователей по технологии единого входа» (SSO);</li> <li>• never – предотвращение выполнения задания;</li> <li>• helm – тег для ролей, которые устанавливают компоненты продукта в Kubernetes с помощью Helm</li> </ul>

## 2.4. Установка IOT\_CM в Kubernetes

Для установки продукта выполните действия (параметры для команд запуска приведены в [Табл. 2](#)):

1. Создайте схему развертывания. В каталоге дистрибутива `inventory/localhost` содержится пример с настройками схемы развертывания. Чтобы создать собственную схему развертывания, создайте копию каталога `localhost` и переименуйте её (например, `staging`).
2. Настройте хосты и опишите группы серверов схемы развертывания в файле `inventory/<схема_развертывания>/1-iotp-iot_cm-k8s`. Убедитесь, что в файле присутствует группа хостов `k8s_iot_cm`. В каждой группе укажите серверы, на которые необходимо установить компоненты продукта, и параметры соединения с ними в формате: `alias`, имя сервера `ansible_host` и метод подключения `ansible_connection`.

Примеры:

```
iot_cm ansible_connection=ssh ansible_host=<адрес удаленного сервера>
# развертывание на удаленный сервер
# либо
iot_cm ansible_connection=local ansible_host=localhost #
развертывание на localhost
```

Файл `inventory/<схема_развертывания>/1-iotp-iot_cm-k8s` по умолчанию заполнен параметрами для локальной установки и в минимальной конфигурации. Нельзя изменять названия существующих групп. Можно добавлять новые хосты, новые группы и наследования.

3. При необходимости переопределите настройки конфигурации в каталоге `inventory/<схема_развертывания>/group_vars` и файле `ansible.cfg`. Для проверки корректности заполнения параметров `inventory` запустите скрипт:

```
ansible-playbook -i inventory/<схема_развертывания> -e
"ansible_ssh_pass=пароль_пользователя
ansible_ssh_user=имя_пользователя" iot_cm-validate-parameters.yml
```

В файле `schemas/iot_cm-schema.json` в JSON-формате описаны параметры, которые необходимо задать в файле `inventory/<схема_развертывания>/group_vars/iot_cm/iot_cm.yml`.

4. Укажите путь к файлу `kubeconfig` одним из способов:
  - определите переменную окружения `KUBECONFIG`;
  - задайте путь к `kubeconfig` в `inventory` в параметре `env.KUBECONFIG`.

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

5. Задайте в `inventory` в параметре `k8s.namespace` название Kubernetes Namespace, в которое будет установлен продукт.
6. Настройте доступ к репозиториям Helm Charts одним из способов:
  - добавьте репозиторий, где располагаются Helm Charts, с помощью команды:

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

В этом случае в `inventory` в параметре `chart.source.type` задайте значение `alias` и укажите `location` в виде `<repo_name>`.

```
chart:  
  source:  
    type: alias  
    location: <repo_name>
```

- указывайте URL Helm-репозитория в параметрах `inventory`:

```
chart:  
  source:  
    type: repo  
    location: "<URL Helm-репозитория>"  
    username: "<Имя пользователя для доступа к Helm-репозиторию>"  
    password: "<Пароль пользователя для доступа к Helm-  
репозиторию>"
```

Подробнее см. описание роли `k8s/helm/install` в документе «Руководство по настройке» продукта COMMON\_INSTALLER [COMMON\_INSTALLER-DOC\_CFG].

7. Запустите установку, задав необходимые параметры:

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

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

Для установки определенной версии продукта или компонента при запуске установки задайте в параметре `--extra-vars (-e)` одну из следующих переменных:

- `version`;
- `path_version`;
- `<component>_version`;
- `<component>_path_version`,

где `<component>` – имя группы в файле хостов и соответствующий этому имени файл в каталоге `group_vars`.

В этом случае будет установлена указанная версия.

Пример:

```
./auto-deploy.sh -i inventory/<схема_развертывания> -u -k --extra-  
-vars="path_version=3.1.0"
```

8. [Завершите установку.](#)

## 2.5. Установка IOT\_CM вручную

Для установки продукта «Управление IoT-подключениями» (IOT\_CM):

- скопируйте и разархивируйте дистрибутив продукта;
- установите схему данных продукта;
- настройте брокер сообщений RabbitMQ для взаимодействия с функциональными модулями продукта;
- настройте брокер сообщений Kafka для взаимодействия с функциональными модулями продукта;
- настройте кластер elasticsearch для работы с продуктом;
- задайте конфигурационные параметры продукта на сервере ZooKeeper;
- установите функциональные модули IOT\_CM с помощью Helm-чартов для Kubernetes на сервер с РЕД ОС.

## 3. Управление версиями

Доступные операции:

- завершение установки версии;
- отметка версии как успешной;
- откат на последнюю успешную версию;
- удаление незавершенных установок.

Описание параметров см. в [Табл. 2](#).

### 3.1. Завершение установки версии

Завершение установки может выполняться автоматически и вручную.

Для включения автоматического завершения установки (по умолчанию выключено) добавьте параметр `finalize_after_deploy = true` в файл `inventory/<схема развертывания>/iot_cm/iot_cm.yml`.

Для завершения установки вручную запустите:

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

Новая версия установится в новый каталог, а по окончании установки произойдет переключение на нее.

### 3.2. Отметка версии как успешной при установке на хост

Отметьте версию как успешную, чтобы в будущем иметь возможность выполнить откат на нее:

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

Статус установленной версии продукта изменится на «Успешно установлена».

### 3.3. Откат на последнюю успешную версию

Для отката на версию продукта, [отмеченную как успешная](#), выполните команду:

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

При откате на предыдущую версию продукта происходит переключение символической ссылки (каталог `current`) на каталог с предыдущей версией.

При отсутствии на сервере предыдущих версий продукта откат будет остановлен с ошибкой «`rollback_path is defined`».

## 3.4. Удаление незавершенных установок

Для удаления артефактов незавершенных установок продукта выполните команду:

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

## 4. Запуск и остановка

Продукт запускается автоматически после установки.

Для ручного управления запуском и остановкой компонентов используйте графический интерфейс Kubernetes (см. [официальную документацию Kubernetes](#)).

Последовательность запуска и остановки компонентов не имеет значения.

## 5. Обновление

Для обновления продукта установите требуемую версию (см. главу [«Установка»](#)).



**Примечание.**

При каждом запуске установки происходит повторная загрузка данных в ZooKeeper. При этом внесенные в конфигурацию изменения не сохраняются.

При работе продукта в Kubernetes обновление отдельных компонентов не поддерживается.

### Обновление продукта без простоя

Обновление IOT\_CM выполняется без простоя при установке в Kubernetes либо при последовательном обновлении нескольких экземпляров на разных хостах.

### Обновление компонентов продукта без простоя

Обновление компонентов без простоя доступно для компонентов одной версии продукта.

## 6. Проверка работоспособности

Для проверки работоспособности продукта IOT\_CM убедитесь, что в log-файлах установочных скриптов нет сообщений об ошибках.

Для проверки работоспособности продукта в Kubernetes выполните команду:

```
kubectl get pods --namespace "<имя Namespace>" \  
  --field-selector=status.phase!=Running,status.phase!=Succeeded \  
  -l app.kubernetes.io/part-of="<Имя продукта>"
```

В случае успешной установки выводится сообщение: «No resources found».

Пример:

```
kubectl get pods --namespace "iot_cm" \  
  --field-selector=status.phase!=Running,status.phase!=Succeeded \  
  -l app.kubernetes.io/part-of="iot_cm"  
  
No resources found
```

## Приложение А: Файлы playbook

Для установки продукта используются файлы playbook:

- `iot_cm-deploy-db.yml` – установка серверной части продукта;
- `iot_cm-deploy-finalize.yml` – переключение на новую (установленную) версию;
- `iot_cm-deploy-full.yml` – установка продукта целиком;
- `iot_cm-deploy-rollback.yml` – откат на предыдущую версию;
- `iot_cm-deploy-sso-roles.yml` – установка ролей SSO;
- `iot_cm-deploy-success.yml` – фиксация успешной установки;
- `iot_cm-deploy.yml` – установка и настройка клиентской части продукта;
- `iot_cm-k8s-deploy.yml` – установка компонентов продукта в Kubernetes;
- `iot_cm-remove-unfinished.yml` – удаление всех незавершенных установок версий продукта;
- `iot_cm-validate-parameters.yml` – проверка конфигурационных параметров;
- `iot_cm-zookeeper-deploy.yml` – установка параметров в ZooKeeper;
- `requirements.yml` – скачивание COMMON\_INSTALLER из Artifactory.

## Приложение В: Конфигурация продукта

Конфигурация продукта задается в каталогах:

- `install/group_vars` – настройки, которые не изменяются от стенда к стенду;
- `install/inventory/default/group_vars` – параметры установки продукта по умолчанию;
- `install/inventory/<схема_развертывания>/group_vars` – настройки, которые могут изменяться в зависимости от окружения, и настройки, которые переопределяют параметры по умолчанию, указанные в файлах каталога `install/inventory/default/group_vars`.

### Внимание!

Изменение значений параметров в файлах `install/inventory/default/group_vars` запрещено.

Если нужно изменить параметр, скопируйте его в соответствующий файл каталога `install/inventory/<схема_развертывания>/group_vars` и задайте ему новое значение.



Приоритет `inventory` зависит от порядка передачи их в командной строке.

Рекомендуется порядок, при котором параметры `install/inventory/<схема_развертывания>/group_vars` переопределяют параметры по умолчанию `install/inventory/default/group_vars`:

```
/auto-deploy.sh -i inventory/<схема_развертывания>
```