

# Система управления цифровыми правами DREPLUS

## Руководство по установке


Индекс	DRMDREPLUS-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

## Содержание

1. Аннотация .....	3
2. Термины и сокращения .....	4
3. Введение .....	5
3.1. Требования к квалификации установщика .....	5
3.2. Системные требования .....	5
4. Предварительные действия .....	6
4.1. Развёртывание кластера Kubernetes .....	6
4.2. Установка PostgreSQL .....	6
5. Установка и настройка системы .....	8
5.1. Общий порядок установки .....	8
5.2. Состав репозитория .....	8
5.3. Настройка переменных окружения .....	8
5.4. CD для артефактов БД .....	8
5.5. Режим снижения потребления ресурсов .....	9

## 1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой "Системы управления цифровыми правами DREPLUS" (DRM DREPLUS) (далее по тексту - DRM или Система). Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области программного обеспечения.

 **Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиторийев ООО "Цифра", доступ к которым предоставляется заказчику по запросу.**

## 2. Термины и сокращения

Термин	Определение
CI/CD	Комбинация непрерывной интеграции и непрерывного развертывания программного обеспечения в процессе разработки. CI/CD объединяет разработку, тестирования и развёртывания приложения.
DRM	Система управления цифровыми правами DREPLUS (DRM DREPLUS) (далее по тексту - DRM или Система)

Сокращение	Расшифровка
CD	Continuous Deployment
CI	Continuous Integration
БД	База Данных
ОС	Операционная Система


## 3. Введение

### 3.1. Требования к квалификации установщика

Для установки системы сотрудник обязан:

- иметь базовые представления и практические навыки работы с системой оркестрации Kubernetes (<https://kubernetes.io/docs/tutorials/kubernetes-basics/>) и пакетным менеджером Helm.
- иметь навыки работы с ОС семейства Linux, а именно:
  - установка пакетов;
  - создание и настройка сетевых подключений;
  - запуск служб, настройка автозапуска служб;
  - установка и настройка PostgreSQL;
  - создание и работа с БД под управлением PostgreSQL.
- иметь знания о DNS.
- иметь базовые представления и практические навыки работы с Git.

### 3.2. Системные требования

 Система DRM в настоящий момент работает только под ОС Ubuntu 18.04 LTS.

Для установки необходимо предварительно выполнить следующие требования:

- Установлен и настроен кластер Kubernetes.
  - Так как развертывание производится в кластере k8s, то необходим config file для доступа к кластеру.
    1. Если пользователь выполнял развертывание Kubernetes самостоятельно, то он сам должен создать config file (см. документацию Kubernetes).
    2. Если Kubernetes был развернут сторонними людьми, то необходимо получить config file у администратора кластера.
- Установлен kubectl (<https://kubernetes.io/docs/tasks/tools/install-kubectl/>).
- Установлен helm.
- Развернут DNS-сервер, преобразование имен dns зоны настроено на мастера k8s (созданы A записи на зону dns).
- Для корректной работы системы DRM требуется поднять несколько Redis баз данных;
- Для корректной работы системы DRM требуется развернуть кластер высокой доступности PostgreSQL (информацию смотри в документации к PostgreSQL);
- Для корректной работы системы DRM необходим доступ к ряду ресурсов, ссылки на которые предоставляются заказчику по запросу:
  - chartmuseum
  - gitlab
  - репозиторий, содержащий данные для загрузки в БД.
  - репозиторий, содержащий helmfile для развертывания DRM. Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов. Сами компоненты поставляются в виде образов (images), из которых разворачиваются Docker-контейнеры. Данные берутся из gitlab.

## 4. Предварительные действия

### 4.1. Развёртывание кластера Kubernetes

Кластер развёртывается по официальной инструкции (<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/high-availability/>).

#### Требования к кластеру


Кластер Kubernetes может быть развернут на одной виртуальной машине с помощью k3s. Минимальные требования к виртуальной машине, предъявляемые со стороны сервера DRM:

Параметры	Значения
Процессор	4 CPU
ОЗУ	4 Гб
Диск	40 Гб
OS	Ubuntu 18.04 LTS

Конфигурация DRM-full в минимальной конфигурации занимает 24 пода и имеет следующие параметры:

Кол-во подов	CPU Requests	CPU Limits	Memory Requests	Memory Limits	Кол-во виртуальных машин
24	0.72	3.12	1.29 GiB	2.19 GiB	1 шт

### 4.2. Установка PostgreSQL

 Для работы системы DRM требуется PostgreSQL версии 12 или выше.

При установке выполняются следующие основные действия:

1. Разворачивается кластер высокой доступности PostgreSQL.
2. Настраивается конфигурация PostgreSQL.
3. Настраиваются подключения к базам данных компонентов DRM.

#### Требования к серверу БД

Минимальные требования к серверу БД:

Параметры	Значения
Процессор	4 CPU (виртуальная машина с Ununtu 18.04)

ОЗУ	4 Гб
Диск	40 Гб

В случае ограничения количества ресурсов у Оператора, можно установить базы данных DRM в общий инстанс PostgreSQL. Производительность DRM зависит от скорости обработки запросов на БД.

## 5. Установка и настройка системы

### 5.1. Общий порядок установки

Для установки DRM в имеющийся настроенный кластер Kubernetes используется процесс CI/CD, настраиваемый с помощью GitLab.

Все действия возможно производить на локальной машине или на любом Ubuntu-сервере с доступом через консоль от имени любого пользователя.

1. Настраиваются переменные окружения (см. [Настройка переменных окружения](#)).
2. Настраиваются yaml-файлы, которые определяют состав и настройки развёртываемых сервисов и баз данных (см. [Состав репозитория](#), [Режим снижения потребления ресурсов](#)).
3. С помощью Gitlab CI/CD выполняется развёртывание системы DRM в ДВА этапа (эти два этапа могут быть разнесены по времени):
  - а. Установка Баз Данных, входящих в состав DRM (дополнительно см. [CD для артефактов БД](#)).
  - б. Установка сервисов и служб, входящих в состав DRM.
4. Выполняется начальное наполнение баз данных (инструкция по наполнению баз данных предоставляется заказчику по запросу).

### 5.2. Состав репозитория

Репозиторий имеет следующий состав:

- helmfile.yaml - главный конфигурационный файл утилиты helmfile.
- default.yaml - файл с values окружения утилиты helmfile.
- values - папка с values для каждого чарта; они являются шаблонными и забирают значения из values окружения (файла default.yaml).
- versions.yaml - файл с версиями компонентов; если в версии установлена пустая строка, то берется последняя версия (в соответствии с semver2).
- tariff-plans - папка с values ресурсов подов.


С помощью этих файлов настраиваются компоненты системы DRM, в том числе многочисленные базы данных.

По умолчанию разворачиваются все компоненты продукта DRM, однако при необходимости можно отключать ненужные: для этого в production.yaml, в корне секции соответствующего компонента нужно выставить *enabled: false*.

### 5.3. Настройка переменных окружения

В системе развёртывания DRM требуется указывать переменные окружения, которые используются непосредственно в процессе деплоя DRM в кластер.

Перечень переменных окружения доступен заказчику по запросу.

 **ВАЖНО!** Environment variables имеют более высокий приоритет, чем переменные, заданные в файлах.



## 5.4. CD для артефактов БД

При развертывании DRM происходит установка SCH и API для БД через механизм Kubernetes Jobs. В процессе установки сохраняется лог в контейнере.

Этот режим поддерживают большинство баз данных компонентов DRM.

## 5.5. Режим снижения потребления ресурсов

Поддерживается несколько планов потребления ресурсов, требуемых для работы 100, 1000, 10 000 устройств. Для этого требуется в helmfile.yaml конечного репозитория развертывания (production repository) прописать требуемый план ресурсов:

- drm/tariff-plans/100-devices.yaml
- drm/tariff-plans/1000-devices.yaml
- drm/tariff-plans/10000-devices.yaml

Пример:

```
---
helmDefaults:
  force: false
  tillerless: false
helmfiles:
- path: drm/helmfile.yaml
  values:
    - drm/default.yaml
    - drm/tariff-plans/100-devices.yaml
    - production.yaml
```

Требуемое количество подов выставляется в каждом сервисе в production.yaml.

Таблица количества подов в зависимости от количества устройств для каждого из компонентов DRM предоставляется заказчику по запросу.

© ООО "Цифра", 2018-2024

Документация "Система управления цифровыми правами DREPLUS. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя