

DRE Advanced Media Platform UI

Руководство по установке

Индекс	PlatformUI-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. Установка и настройка системы	6
4.1. Создание новой среды	6
4.2. Настройка параметров установки в кластер	6
4.3. Наполнение веб-интерфейса	6
4.4. Настройка services.json	7
4.5. Настройка production.yaml	7
4.6. Настройка для работы папки со статичными файлами	9
4.7. Настройка доступа пользователя к сервисам	9
4.8. Развертывание системы	9

1. Аннотация

Данный документ содержит руководство по установке и первоначальной настройке системы DRE Advanced Media Platform UI (далее по тексту - PlatformUI или Система), а также описание системных требований для компонентов.

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой системы. Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области инсталляции 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 репозиториях ООО "Цифра", доступ к которым предоставляется заказчику по запросу.

2. Термины и сокращения

Термин	Определение
Account Manager	Условное обозначение продукта "DRE Account Manager". Сервис авторизации и распределения прав в системе DREAMP. Через Account Manager в DREAMP авторизуются сервисы (например, Система управления цифровыми правами DREPLUS, DRE Config Manager, DRE Advanced Media Platform Profile Data Guide и другие), в WEB UI Account Manager прописываются их права. Также в Account Manager хранятся, создаются новые, редактируются учетные записи пользователей DRE Advanced Media Platform META DATA SERVER, через WEB UI назначаются права, создаются роли и группы прав для учетных записей пользователей
DREAMPlatform	DRE Advanced Media Platform. Комплексное решение для телесмотрения, состоящее из взаимосвязанных компонентов, обеспечивающих генерацию, хранение и доставку контента (OTT и VOD) до телезрителя
PlatformUI	Условное обозначение продукта "DRE Advanced Media Platform UI". Программа предназначена для реализации единой точки входа с систему DRE Advanced Media Platform. Предоставляет администраторам платформы доступ к полному списку всех продуктов, доступных на ней в зависимости от комплекта поставки, а также возможности для централизованной настройки этих продуктов
SSO	Единый сервис авторизации и аутентификации

Сокращение	Расшифровка
ACM	Account Manager
DREAMP	DRE Advanced Media Platform
SSO	Single Sign-On
UI	User Interface

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.
- иметь базовые представления и практические навыки работы с Git.

3.2. Системные требования

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 (<https://helm.sh/ru/docs/intro/install/>).



Helm необходим в случае ручной установки элементов системы в кластер k8s. При стандартной установке с использованием gitlab-ci использование helm не предполагается.

- Для корректной работы Системы необходим доступ к следующим ресурсам:
 - chartmuseum (ссылка и права доступа предоставляются по запросу заказчика)
 - gitlab (ссылка и права доступа предоставляются по запросу заказчика)
- Необходим доступ к репозиторию (ссылка и права доступа предоставляются по запросу заказчика), содержащему helmfile для развертывания PlatformUI. Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов. Сами компоненты поставляются в виде чартов (charts), из которых развертываются Kubernetes Pods. Данные берутся из gitlab (ссылка и права доступа предоставляются по запросу заказчика).

4. Установка и настройка системы

Конфигурация и утилиты helm для разворачивания PlatformUI находятся в репозитории (ссылка и права доступа предоставляются по запросу заказчика).

4.1. Создание новой среды

Основные этапы создания среды следующие:

1. Создать отдельный проект в Gitlab. Необходимо использовать собственное название репозитория для продукта PlatformUI.
2. Клонировать имеющийся проект для его локального редактирования.
3. Включить в скопированный репозиторий релизную версию конфигурации продукта PlatformUI путем добавления submodule.
4. Создать файл `production.yaml` для последующего переопределения значений из файла `default.yaml`.
5. Создать `helmfile.yaml`.
6. Создать файл `gitlab-ci.yml` со стандартным содержанием.
7. Проверить конфигурацию продукта.
8. Синхронизировать локальный и удаленный репозитории.

4.2. Настройка параметров установки в кластер

Основные этапы создания среды следующие:

1. Подключить Gitlab runner.
2. Указать необходимое значение `kube_config`.


4.3. Наполнение веб-интерфейса

Наполнение веб-интерфейса данными осуществляется через прокидывание статичных файлов в контейнер с веб-интерфейсом.

Для этого нужно создать папку `project-static-data`, скопировав содержимое из папки `platform-ui-template-example` (входит в состав релизного репозитория).

В папке хранятся:


- json файлы с переводами (`ru.json.example`, `en.json.example`);
- `services.json.example` - файл с описанием и настройками продуктов на странице;
- изображения и иконки, которые могут быть прописаны в файле `services.json`.

 Путь к файлу `services.json` впоследствии должен быть указан при настройке `production.yaml` (секция `dirConfig`, параметр `mountPath`).

При этом должен быть выставлен флаг `dirConfig.enabled: true`.

4.4. Настройка services.json


В созданной папке project-static-data можно (при необходимости) настроить файл services.json.

 services.json.example (в папке platform-ui-template-example) это **правильно настроенный** набор конфигов для конкретного решения, с правильно выставленными настройками (**за исключением полей link**).

services.json представляет собой массив объектов (карточек продуктов).

Каждый продукт может содержать поля:

- name - ключ названия продукта
- serviceVersion - цифра версии продукта
- operatingMode - режим работы (например, SCRAMBLER - DREMARK)
- title - ключ заголовка продукта
- description - ключ описания продукта
- img - название изображения заднего фона карточки продукта, которое лежит в project-static-data
- permissions - строка с названием доступа (permission) для отображения данного продукта текущему пользователю
- link - полный url для перехода с разводящей страницы на страницу продукта
- isToken - флаг (true/false), указывает, нужно ли добавить токен к url для перехода с разводящей страницы на страницу продукта
- subsections - поле хранит данные разделов продукта. Массив разделов, каждый раздел может содержать поля:
 - name - ключ названия раздела
 - link - полный url для перехода с разводящей страницы на продукт в нужный раздел
 - permissions - строка с доступом для отображения данного раздела текущему пользователю
 - img - название иконки раздела, которая лежит в project-static-data

 **Обратите внимание!** После копирования services.json надо **обязательно** исправить для каждого сервиса поле link - установить актуальную ссылку на веб-интерфейс сервиса. По умолчанию редактирование services.json, **кроме полей link**, не требуется.

4.5. Настройка production.yaml

В файле production.yaml нужно настроить параметры, которые используются в промышленной эксплуатации в соответствии с требуемой конфигурацией. Файл production.yaml переопределяет переменные в файле default.yaml.


При настройке production.yaml **обязательно** нужно проверить и при необходимости изменить (привести в соответствие с имеющимися данными) следующие параметры:

1. секция ingress:
 - a. host - хост-имя, по которому будет доступен portal-ui.
2. секция dirConfig (см. [ниже](#)):
 - a. enabled - флаг включения работы с папкой project-static-data. Должен быть enabled: true
 - b. mountPath - путь к папке project-static-data.

3. секция service:

а. подсекция environments:

- i. API_HOST_ACM - адрес сервера DRE Account Manager (ACM) - для получения информации пользователя и списка прав

 Описание остальных переменных из этой подсекции приведено в таблице ниже. Их значения можно оставить без изменений.

Остальные параметры (если ниже не указано иное) можно оставить без изменений.

Таблица с описанием используемых в PlatformUI переменных (production.yaml, секция service, подсекция environments)

Название	Описание	Значение по умолчанию	Обязательность
PAGE_TITLE	Название ключа в json конфигах, отвечающего за название страницы	'pageTitle'	-
IS_TABS	Включить режим отображения табов (групп продуктов) и поиска	'false'	-
IS_SUBSECTION	Включить режим отображения разделов продуктов	'false'	-
IS_FAVORITES	Включить режим работы функционала добавления и просмотра избранного	'false'	-
API_HOST_ACM	Адрес сервера DRE Account Manager (ACM) - для получения информации пользователя и списка прав	' '	+
THEME	Цветовая тема веб-интерфейса	'GSTheme'	-
PRODUCT_CODE	Код продукта для авторизации через единое окно авторизации	' '	-
PRODUCT_VERSION	Отображается версия продукта в формате: v. X.Y.Z, где X - мажорная версия, Y - минорная версия, Z - номер патча	' '	-
SSO_ACTIVE	Включить режим работы: Авторизация через единое окно авторизации	'false'	-
SSO_HOST	Адрес веб-интерфейса единого окна авторизации	' '	-
TOKEN_NOTIFY_TIME	За сколько минут нужно уведомить пользователя об истечении жизни токена (показ нотификации)	'5'	-

4.6. Настройка для работы папки со статичными файлами

Для работы с созданной папкой `project-static-data` нужно внести соответствующие изменения в следующих файлах:

1. В файле `production.yaml`:


```
dirConfig:
  enabled: true
  mountPath: "/code/dist/project-static-data/"
```

2. В файле `.gitlab-ci.yml`:

```
install_project_data:
  stage: before
  script:
    - |
      if kubectl get configmap project-config -n $NAMESPACE > /dev/null 2>&1; then
        kubectl delete configmap project-config -n $NAMESPACE
      fi
    - kubectl create configmap project-config --from-file=project-static-data/ -n $NAMESPACE
```

4.7. Настройка доступа пользователя к сервисам

Доступ к сервисам настраивается в Account Manager (ACM) путем создания специальной группы прав (как и для других пользователей других систем).

 PlatformUI цепляется к уже созданным правам на определенные интерфейсы. Иными словами, если в Account Manager для продукта (например, Система условного доступа DREGUARD) добавлен сервис (например, "dre_platformui"), то PlatformUI использует этот сервис (dre_platformui) для понимания наличия прав на продукт (Система условного доступа DREGUARD).

4.8. Развертывание системы

Развертывание PlatformUI осуществляется в Gitlab CI/CD:

1. В боковом меню выбрать **Build** (на панели слева) -> **Pipelines**.
2. Запустить pipeline, путем нажатия на шаг `init` (кнопка ">>"). Во всплывающем окне нажать кнопку `Play`.
3. Дождаться окончания операции.

© ООО "Цифра", 2024

Документация "DRE Advanced Media Platform UI. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя