

Система доставки рекламы на клиентские устройства DREASYS

Руководство по установке

Индекс	2060-ADS-IG
Секретность	Публичный - L0
Ревизия	1.0
Статус	Согласован
Подразделение	Департамент по разработке сервисов
Компания	GS Labs

Содержание

1. Аннотация	3
2. Термины и сокращения	4
3. Минимальные системные требования	5
4. Развертывание DREASYS	6
4.1. Yaml-файл	6
4.2. Установка	6

1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой системы доставки рекламы на клиентские устройства DREASYS (далее - DREASYS). Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области инсталляции программного обеспечения.

2. Термины и сокращения

Термин	Определение
AdRiver	Система управления рекламой, является внешним системным продуктом.
СУР	(Система управления рекламой) Комплекс инструментов, позволяющий эффективно управлять рекламой
Устройство	Любое устройство, в котором поддерживается функционал получения рекламного контента (например, приёмник, игровая консоль).

3. Минимальные системные требования

Для установки DREASYS как докер образа на один сервер (без Kubernetes) необходимо наличие одного сервера.

Сервер должен удовлетворять следующим требованиям:

1. Операционная система ubuntu-16.04-server-amd64 (с установленным пакетом sudo).
2. Многоядерный центральный процессор с тактовой частотой каждого ядра 2 ГГц (не менее 4 ядер).
3. Объем оперативной памяти 8 ГБ.
4. 1 жесткий диск или более емкостью не менее чем по 200 ГБ.
5. Интерфейса Ethernet 100 Base-T с поддерживаемой пропускной способностью 100 Мбит/сек соответственно.
6. Свободное место для папки временных файлов /tmp - 10 ГБ.

4. Развертывание DREASYS

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

Настройка осуществляется путем изменения значений в helm-файле с настройками кластера Kubernetes, который необходимо развернуть.

4.1. Yaml-файл

Для проекта DREASYS helm-файл может иметь название "ads.yaml", настройки helm-файла, отвечающие за функционал системы, которые нужно при необходимости указать при развертывании проекта:

- id - ID сайта в СУР, например: "221358"
- pass- адрес СУР, например: "ad.adriver.ru"
- resolver- DNS-сервер, например: "8.8.8.8"
- extid - значение аппаратного идентификатора устройства, если оно явно не указано у клиента, например: "-1"
- log_format - Формат логов. Например: ""\$remote_addr -- \$remote_user \$time_local '\$request' \$status \$body_bytes_sent '\$http_referer' '\$http_user_agent'"

4.2. Установка

Для развертывания приложений в установленном кластере Kubernetes необходимо сделать форк текущего репозитория deploy в свой репозиторий, настроить deployment-keys. Далее приложение развернется в кластере средствами GitLab. Последующее обновление приложения делается модификацией конфигурационных файлов в этом созданном форк-репозитории.

1. Сделайте fork репозитория с конфигурацией DREASYS <https://gitlab.gs-labs.tv/ott-k8s/deploy> (доступ ограничен) в любое приватное пространство. Для этого необходимо нажать кнопку "Fork" и выбрать пространство, в которое будет скопирован репозиторий.
2. Откройте скопированный репозиторий и настройте Runner, который будет развертывать приложение. Перейдите в раздел *Settings->CI/CD->Runners settings*. Если в разделе "Runners activated for this project" есть раннеры с зелёным лейблом, нажмите кнопку "Enable" рядом с ним. Если раннеров нет, добавьте по инструкции, которая находится там же.
3. Настройте kubeconfig для доступа раннера к кластеру. Перейдите в раздел *Settings->CI/CD->Secret variables*. Создайте переменную *kube_config* и в её значении пропишите содержимое файла *.kube/config* с любой мастер-ноды кластера kubernetes, зашифрованное кодировкой base64.

```
base64 -w 0 ~/.kube/config
```

4. Добавьте ключ и сертификат для домена. Перейдите в раздел *Settings->CI/CD->Secret variables*. Создайте переменную *prod_cert* и в её значении пропишите сертификат домена, закодированный в base64. Создайте переменную *prod_key* и в её значении пропишите ключ сертификата домена, закодированный в base64.

```
base64 -w 0 cert.pem  
base64 -w 0 cert.key
```

5. Добавьте `CI_API_TOKEN`. Создать его нужно для своего пользователя, от имени которого будет происходить `deploy`-процесс. Для этого перейдите на страницу https://gitlab.gs-labs.tv/profile/personal_access_tokens (доступ ограничен) и создайте токен с любым именем и правами на `api`. После нажатия на кнопку "Create personal token" на экране появится токен, который в системе не хранится в открытом виде, поэтому его нужно сразу скопировать и создать переменную `CI_API_TOKEN` в `Settings->CI/CD->Variables` с этим значением.
6. Создайте триггер для деплоя. Перейдите в настройки репозитория в раздел `Settings->CI/CDPipeline triggers`. Введите любое описание триггера, нажмите кнопку "Add trigger".
7. После этого необходимо клонировать репозиторий локально, внести необходимые правки в конфигурацию файла `ads.yaml`. В файле `ads.yaml` необходимо внести изменения:
 - a. `KEEPALIVED values IPs` укажите список всех виртуальных IP-адресов.
 - b. `KEEPALIVED values servers` перечислите все физические ноды, где должен быть развёрнут `keepalived`.
 - c. `FLUENT-BIT values elasticsearch` - настройки для передачи логов на `ElasticSearch`.
8. После `git commit` и `git push` запустится процесс деплоя, за результатом которого можно наблюдать на странице `CI/CD->Pipelines`.

После того, как все этапы будут пройдены, сервис считается развёрнутым.

© ООО "Цифра", 2019-2022

Документация "Система доставки рекламы на клиентские устройства DREASYS. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя.