

DRE Defender

Руководство по установке

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

Содержание

1. Аннотация	3
2. Термины и сокращения	4
3. Минимальные системные требования	5
4. Внедрение компонентов в свои проекты	6
4.1.1. Версии компонентов	6
4.1.2. Настройки компонентов	6
4.1.3. Как внедрить компонент к себе в репозиторий	6
4.1.4. Изменение дефолтных параметров	6
4.2. Добавление наполнения маршрутов для связки apf_server - route_server	7

1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой сервиса. Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области инсталляции программного обеспечения.

i Данный документ опубликован исключительно для изучения системных требований при установке 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\Сокращение	Определение\Расшифровка
БД	База данных.
DRE Defender	Сервис маршрутизации запросов, продукт ООО "Цифра".
AGS	API Gateway Server, рабочее название продукта DRE Defender.

3. Минимальные системные требования

В текущей реализации сервис работает как компонент, встраиваемый в другие продукты ООО "Цифра".
Вследствие этого, минимальные системные требования применяются к продукту, в который встраивается DRE Defender и в данном руководстве не рассматриваются.

4. Внедрение компонентов в свои проекты

4.1.1. Версии компонентов

Актуальные версии компонентов можно найти в файле `versions.yaml` репозитория ООО "Цифра" (доступ ограничен, предоставляется по запросу).

4.1.2. Настройки компонентов

Настройки компонентов хранятся в папке `values` репозитория ООО "Цифра" (доступ ограничен, предоставляется по запросу).

4.1.3. Как внедрить компонент к себе в репозиторий

1. Добавить submodule AGS

```
git submodule add git@gitlab.gs-labs.tv:releases/ags.git
```

2. В файле `helmfile.yaml` продукта добавить необходимой сервис. Ниже представлен пример для `apf_server`

```
- name: apf-server
  condition: apf_server.enabled
  labels:
    stage: second
  chart: chartmuseum/apf_server
  version: {{ .Values.versions.apf_server }}
  values:
    - ags/values/apf_server/values.yaml.gotmpl
```

4.1.4. Изменение дефолтных параметров

1. Для дополнения или изменения параметров, заданных по умолчанию, переопределите конфигурацию у себя на стенде. Конфигурация хранится в репозиториях сервисов. Пример переопределения конфигурации для `apf_server` можно увидеть ниже:

```
apf_server:
  enabled: true
  replicas: 1
  hpa:
    enabled: true
    minReplicas: 1
    maxReplicas: 2
    targetCPUUtilizationPercentage: 25
  ingress:
    hostname: apf.testdrp.tz.cas
    enabled: true
    useHttps: false
  resources:
    default_limits:
      cpu: 102m
      memory: 128Mi
    default_requests:
      cpu: 96m
      memory: 64Mi
  env:
    GOGC: '400'
    GOMAXPROCS: '1'
  configEnabled: true
  config:
    logger:
      level: debug
  route_server:
    host: "route-server"
    timeout: 5
    protocol: http
    health_check: false
    sync_period: 300
```

2. Если требуется зафиксировать версию для проекта, введите актуальную по аналогии с versions.yaml

3. Задеплойте необходимые изменения

4.2. Добавление наполнения маршрутов для связки apf_server - route_server

1. Добавляем базовый адрес "some.address.ru" системы

```
curl --location 'route_server/api/v1/domain_addresses' \
--header 'Content-Type: application/json' \
--data '{
  "domain_address_code": "address_code",
  "domain_address_name": "some.address.ru",
  "domain_address_description": "address description"
}'
```

2. Далее, необходимо создать маршрут для "some.address.ru"

```
curl --location 'route_server/api/v1/routes' \  
--header 'Content-Type: application/json' \  
--data '{  
  "incoming_url": "/api/v1/incoming/url",  
  "incoming_domain_addresses": [],  
  "outgoing_url": "/api/v1/outgoing/url",  
  "outgoing_domain_address": "address_code",  
  "outgoing_method": "get",  
  "incoming_method": "get",  
  "protocol": "http",  
  "outgoing_format": "base",  
  "description": "URL for get individual playlist",  
  "timeout": 15  
}'
```

3. Дополнительно создаем параметры "Link", "userId", "OperatorId", "token", "domain_code" для использования их в запросе

```
curl --location 'route_server/api/v1/parameters' \  
--header 'Content-Type: application/json' \  
--data '{  
  "parameter_code": "Link",  
  "parameter_name": "Link",  
  "parameter_description": "Link param description",  
  "set_types": [  
    "dynamic"  
  ],  
  "send_types": [  
    "query"  
  ]  
}'  
  
curl --location 'route_server/api/v1/parameters' \  
--header 'Content-Type: application/json' \  
--data '{  
  "parameter_code": "userId",  
  "parameter_name": "userId",  
  "parameter_description": "userId param description",  
  "set_types": [  
    "dynamic"  
  ],  
  "send_types": [  
    "query"  
  ]  
}'  
  
curl --location 'route_server/api/v1/parameters' \  
--header 'Content-Type: application/json' \  
--data '{  
  "parameter_code": "OperatorId",  
  "parameter_name": "OperatorId",  
  "parameter_description": "OperatorId param description",  
  "set_types": [  
    "dynamic"  
  ],  
  "send_types": [  
    "query"  
  ]  
}'  
  
curl --location 'route_server/api/v1/parameters' \  
--header 'Content-Type: application/json' \  
--data '{
```



```
"parameter_code": "token",
"parameter_name": "token",
"parameter_description": "token param description",
"set_types": [
  "dynamic"
],
"send_types": [
  "body"
]
}'

curl --location 'route_server/api/v1/parameters' \
--header 'Content-Type: application/json' \
--data '{
  "parameter_code": "domain_code",
  "parameter_name": "domain_code",
  "parameter_description": "domain_code from DRM token",
  "set_types": [
    "dynamic"
  ],
  "send_types": [
    "body"
  ]
}'
```

4. Добавляем все базовые параметры в базовый маршрут

```
curl --location 'route_server/api/v1/routes/100/parameters/Link' \
--header 'Content-Type: application/json' \
--data '{
  "type": "get",
  "send_type": "query",
  "value_type": "string",
  "set_type": "dynamic"
}
',

curl --location 'route_server/api/v1/routes/100/parameters/OperatorId' \
--header 'Content-Type: application/json' \
--data '{
  "type": "get",
  "send_type": "query",
  "value_type": "string",
  "set_type": "dynamic"
}
',

curl --location 'route_server/api/v1/routes/100/parameters/OperatorId' \
--header 'Content-Type: application/json' \
--data '{
  "type": "send",
  "send_type": "query",
  "value_type": "string",
  "set_type": "dynamic",
  "parameter_code_type": "OperatorId"
}
',
```

5. Создаем дополнительный маршрут

```
curl --location 'route_server/api/v1/additional_requests' \  
--header 'Content-Type: application/json' \  
--data '{  
  "add_request_code": "getUserIdService",  
  "add_request_name": "getUserIdService",  
  "add_request_description": "request description",  
  "add_request_url": "/api/v1/token/check",  
  "method": "POST",  
  "outgoing_domain_address": "outgoing.domain.address",  
  "protocol": "http",  
  "type": "identification"  
}'
```

6. Добавляем дополнительный маршрут в базовый запрос

```
curl --location --request POST 'route_server/api/v1/routes/100/additional_requests/getUserIdService'
```

7. Добавляем параметры в дополнительный запрос

```
curl --location 'http://rs.scrambler-based-stand.drmteam-2.casdev/api/v1/additional_requests/getUserIdService  
/parameters/domain_code' \  
--header 'Content-Type: application/json' \  
--data '{  
  "type": "get",  
  "value_type": "string",  
  "send_type": "body",  
  "set_type": "dynamic",  
  "parameter_code_type": "domain_code"  
}'
```

8. Связываем параметр с исходным запросом

```
curl --location 'route_server/api/v1/routes/100/parameters/userId' \  
--header 'Content-Type: application/json' \  
--data '{  
  "type": "send",  
  "send_type": "query",  
  "value_type": "string",  
  "set_type": "dynamic",  
  "add_request_code": "getuseridservice",  
  "parameter_code_type": "domain_code"  
}'
```

© ООО "Цифра", 2023-2024.

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