«Demand Response Mobile»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ

Москва
2021

СОДЕРЖАНИЕ

1. ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

API	Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
CPU	Центральное процессорное устройство.
DNS	Система доменных имён.
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.
HDD	Постоянное запоминающее устройство.
HTTP	HyperText Transfer Protocol – протокол прикладного уровня передачи данных.
HTTPS	HyperText Transfer Protocol Secure – расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
JavaScript	Прототипно-ориентированный сценарный язык программирования.
JSON	Текстовый формат обмена данными, основанный на JavaScript.
LDAP	Протокол взаимодействия со службой каталогов.
LVM	Менеджер логических томов – система управления дисковым пространством, абстрагирующаяся от физических устройств.
Nexus	Менеджер репозиториев, предназначенный для проксирования репозиториев и хранения ПО.
RAM	Оперативное запоминающее устройство.
SSH	Сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой.
SSL	Криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети.
WAF	Межсетевой экран для веб-приложений, который выявляет разнообразные информационные атаки
ПО	Программное обеспечение
СУБД	Система управления базами данных
УЗ	Учтённая запись
ФПА	Фонд программ и алгоритмов

2. НАЗНАЧЕНИЕ РУКОВОДСТВА

Инструкция описывает действия администратора по установке и настройке программы для ЭВМ «Demand Response Mobile» (далее по тексту – DRM, Система) на серверы с уже установленной системой «Demand Response».

DRM является подсистемой программного обеспечения «Demand Response».

Перечисленные в инструкции команды выполняются с использованием SSH-клиента, например – PuTTY.

3. ТРЕБОВАНИЯ К ПРОГРАММНЫМ/АППАРАТНЫМ РЕСУРСАМ

3.1. Требования к аппаратному обеспечению

Рекомендованные характеристики серверов системы «Demand Resonse» указаны в таблице 1.

Таблица 1 – Рекомендуемая конфигурация серверов «Demand Resonse»

Тип сервера	Кол-во	Характеристики сервера		
		CPU, core	RAM, Gb	HDD, Gb
front-srv0№	2	4	4	30
inner-front-srv	1	4	4	30
backend-srv0№	2	6	12	50
db-srv0№	2	4	8	600

3.2. Требования к программному обеспечению

Серверы приложений :

- Операционная система – Astra Linux Common Edition;
- Kaspersky Endpoint Security;
- ПО Docker Engine версии 19.03+.

Серверы СУБД:

- Операционная система – Astra Linux Common Edition;
- СУБД – PostgreSQL версии 9.6;
- Kaspersky Endpoint Security;
- Диски в конфигурации LVM;
- ПО Keepalived.

Web-серверы:

- Операционная система – Astra Linux Common Edition;
- Kaspersky Endpoint Security;
- ПО Nginx версии 1.16.1+;

3.3. Предварительная настройка окружения

Для запуска Системы необходимо:

1. Зарегестировать DNS имя для frontend сервиса Системы (**mobile-web**).
2. Предоставить доступ к серверам front-srv0№ через WAF. Сервисы, предоставляемые через WAF: **mobile-web**.
3. Выпустить SSL сертификаты в PEM формате для сайта Системы.
4. Выпустить SSL сертификат для сервера приложений Системы в PEM[1] формате.
5. Запросить УЗ для доступа к ФПА с исходным кодом Системы, а также для получения конфигурационных файлов и артефактов сборки.
6. Запросить УЗ для доступа к артефактам Системы расположенным на Nexus сервере ФПА.

3.4. Сетевой доступ

Для обновления ПО и компонентов Системы серверам Системы необходим доступ в сеть Интернет. Список ресурсов, к которым необходим доступ, указан в таблице 2.

Таблица 2 – Список внешних ресурсов

Сервер	**Сайт**
Сервер приложений Системы	download.astralinux.ru download.docker.com

Необходимо обеспечить доступ к указанным сайтам напрямую, или через proxy-сервер.

Для проверки доступности ресурсов можно выполнить следующие команды:

```
curl -Is https://download.astralinux.ru | head -n 1
curl -Is https://download.docker.com | head -n 1
```

Ожидаемый ответ:

```
HTTP/1.0 200 Connection established
```

Кроме этого, в инструкции по установке и настройке «Demand Response» указано требуемое сетевое взаимодейтсвие серверов.

[1] Необходима пара ключей (открытый и закрытый ключ), расширения по умолчанию данной пары -.ctr и .key

4. УСТАНОВКА И НАСТРОЙКА КОМПОНЕНТОВ СИСТЕМЫ

4.1. Настройка сервера приложений Системы

4.1.1. Настройка Docker-engine

На серверах приложений должно быть установлено и настроено программное обеспечение Docker-engine версии 19.03+ в соответсвии с требованиями раздела 4.3.1. инструкции по установке и настройке програмного беспечения «Demand Response».

4.1.2. Настройка конфигурационных файлов

Для загрузки конфигурационных файлов сервисов приложений необходимо подключиться к каждому серверу приложений по SSH и выполнять следующую последовательность действий:

1. Загрузить репозиторий с шаблоном конфигурации запуска, используя команду:

```
git clone https://[...].cdu.so/dr/config.git ~/config/
```

На запрос авторизации необходимо ввести данные УЗ, имеющей доступ к репозиторию проекта в ФПА.

2. Перейти в директорию с шаблоном запуска `cd ~/config/` и используя в качестве шаблона файл: `~/config/.env-example` создать новый файл: `~/config/.env`, используя команду:

```
cp ~/config/.env-example ~/config/.env
```

3. Заменить значение параметра CONFIG_SERVICE_IP на адрес основного сервера приложений. Формат переменной – http://10.0.0.1.
4. Изменить значение переменной **«all_version»**. Для продуктивного стенда значение переменной – «prod», для полигонного «poligon»

4.1.3. Настройка и запуск сервиса config-service

Для настройки и запуска сервиса **config-service** необходимо:

1. Перейти в директорию с шаблоном запуска `cd ~/config/` и используя в качестве шаблона файл: `~/config/config-service/.env-example` создать новый файл: `~/config/config-service/.env`.
2. Заполнить параметры в файле `~/config/config-service/.env` в соовтетсвии с таблицей 3. В таблице 3 приводится полный список переменных необходимых для заполнения при развертывании програмного обеспеечния «Demand Response», примеры заполнения, а также описание каждой переменной доступны в инструкции по установке и настройке програмного беспечения «Demand Response». Для запуска и корректной работы DRM необходимо заполнить

параметры, обозначенные в таблице №3 (для которых приведено описание и пример заполнения).

Таблица 3 – Список параметров, используемых в env-example

Переменная	Пример	Описание
DB_IP		
$CAPTCHA_DB		
$CAPTCHA_DB_USER		
$CAPTCHA_DB_PASS		
$CORE_DB		
$CORE_FS_DB		
$CORE_DB_USER		
$CORE_DB_PASS		
$MOBILE_SITE	https://dr-mobile.so-ups.ru	Адрес сайта Системы для мобильных устройств (Сервис mobile-web)
$EXTERNAL_SITE		
$EUREKA_SERVICE_URI		
$CORS_URLS	"https://xxxx.ru; https://dr-mobile.so-ups.ru; https://xxxxx.cdu.so"	Перечисление всех сайтов Системы
$COOKIE_DOMAINS		
$JWT_TOKEN_SECRET		
$LDAP_URL_PORT		
$LDAP_BASE		
$LDAP_MANAGER_LOGIN		
$LDAP_MANAGER_PASS		
$MAIL_USER_NAME		
$MAIL_USER_PASS		
$SMTP_HOST		
$SOI_URL		
$SOI_USER		
$SOI_PASS		
$ESG_SERVICE_ROOT_PATH		
$ESG_SERVICE_CRL_PATH		
$PROXY_USE		
$PROXY_HOST		
$PROXY_PORT		
$PROXY_USER		
$PROXY_PASS		

3. Запустить сервис. Для запуска сервиса необходимо использовать SH скрипт, выполнив команду:

```
./config-service.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации)

Ожидаемый результат выполнения команды – ***запущен docker-контейнер config-service.***

4.1.4. Запуск сервиса core-service

Для запуска сервиса необходимо выполнить действия, указанные в разделе 4.1.2. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

1. Перейти в директорию с шаблоном запуска `cd ~/config/` и запустить скрипт запуска сервиса:

```
./core-service.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации)

Ожидаемый результат выполнения команды – ***запущен docker-контейнер core-service.***

4.1.5. Запуск сервиса captcha-service

Для запуска сервиса необходимо выполнить действия, указанные в разделе 4.1.2. Подключиться по SSH к каждому серверу приложений и выполнить следующую последовательность действий:

1. Перейти в директорию с шаблоном запуска `cd ~/config/` и запустить скрипт запуска сервиса:

```
./captcha-service.sh
```

(SH скрипт расположен в директории с шаблоном конфигурации)

Ожидаемый результат выполнения команды – запущен **docker-контейнер captcha-service**.

4.2. Настройка сервиса mobile-web

Для настройки сервиса необходимо авторизоваться на сайте ФПА и загрузить артефакт сервиса **mobile-web**.

После чего необходимо загрузить на каждый web-сервер SSL сертификат и артефакт в домашнюю папку (~/), расположенную на сервере Системы (рекомендуется использовать ПО WinSCP[2]).

Далее необходимо подключиться к каждому web-серверу по SSH и выполнить следующую последовательность действий:

[2] Документация на ПО доступна по ссылке https://winscp.net/eng/docs/start

2. Установить Nginx (если отсутсвует) при помощи команды:

```
sudo apt install -y nginx
```

3. Удалить автоматически созданный файл конфигурации nginx:

```
sudo rm /etc/nginx/sites-available/default
```

4. Запустить файловый менеджер командой sudo mc и перенести SSL сертификат в директорию `/etc/nginx/conf.d/`.
5. Создать директорию сервиса **mobile-web**:

```
sudo mkdir /var/www/html/mobile-web
```

6. Предоставить права user на директорию с web-приложением, используя команду:

```
sudo chown -R user:user /var/www/html/mobile-web
```

7. Разархивировать артефакт сервиса командой:

```
unzip ./artifacts.zip
```

(необходимо заменить «./artifacts.zip» на путь к артефакту сервиса)

8. Очистить директорию web сайта командой:

```
rm -r /var/www/html/mobile-web/*
```

9. Переместить файлы сервиса в директорию web сайта командой:

```
cp -r ./build/* /var/www/html/mobile-web/
```

(необходимо заменить «./» на путь к разархивированному артефакту)

10. Удалить временные файлы сервиса:

```
rm –rf ./build/
```

(необходимо заменить «**.**/build» на путь к разархивированному артефакту)

Таблица 4 – Список переменных в конфигурационных файлах Nginx

$SITE-NAME	DNS имя сервиса Системы
$KEY_PATH	Путь до файла, содержавшего закрытый ключ
$CRT_PATH	Путь до файла, содержавшего открытый ключ
$BACKEND-IP1	IP адрес основного backend сервера.
$BACKEND-IP2	IP адрес резервного backend сервера.

11. Далее необходимо заполнить файл конфигурации сервиса командой:

```
sudo nano /etc/nginx/conf.d/mobile-web.conf,
```

согласно нижеприведенному шаблону:

```
server {
```

```
  server_name  $SITE-NAME  ;
  listen 443 ssl;
  ssl_certificate /etc/nginx/conf.d/$CRT_PATH;
  ssl_certificate_key /etc/nginx/conf.d/$KEY_PATH;
  error_log /var/log/nginx/error.log warn;
  access_log /var/log/nginx/access.log combined;

  root  /var/www/html/mobile-web/;
  underscores_in_headers on;

  client_max_body_size 100m;

  location / {
    try_files $uri $uri/ /index.html =404;
  }
  location /index.html {
    add_header Cache-Control no-cache;
  }
  location /config.json {
    add_header Cache-Control no-cache;
  }
  location /captcha-service/api/ {
     proxy_pass http://dr-captcha-service;
  }
    location /captcha-service/api/ {
         proxy_pass http://dr-captcha-service;
    }
    location /core-service/api/v1/external/ {
        proxy_pass http://dr-web-api;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
    }
  error_page   500 502 503 504  /50x.html;
  location = /50x.html {
    root   /usr/share/nginx/html;
  }
}
server {
  if ($host = $SITE-NAME  ) {
    return 301 https://$host$request_uri;
  }
  server_name  $SITE-NAME;
  listen 80;
  return 404;
}
```

Пример:

```
server {
  server_name  dr-mobile.so-ups.ru;
```

```
  listen 443 ssl;
  ssl_certificate /etc/nginx/conf.d/dr-mobile.crt;
  ssl_certificate_key /etc/nginx/conf.d/dr-mobile.key;
  error_log /var/log/nginx/error.log warn;
  access_log /var/log/nginx/access.log combined;

  root  /var/www/html/frontend-web/;
  underscores_in_headers on;

  client_max_body_size 100m;

  location / {
    try_files $uri $uri/ /index.html =404;
  }
  location /index.html {
    add_header Cache-Control no-cache;
  }
  location /config.json {
    add_header Cache-Control no-cache;
  }
    location /captcha-service/api/ {
         proxy_pass http://dr-captcha-service;
    }
    location /core-service/api/v1/external/ {
        proxy_pass http://dr-web-api;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
    }  error_page   500 502 503 504  /50x.html;
  location = /50x.html {
    root   /usr/share/nginx/html;
  }
}
server {
  if ($host = dr-mobile.so-ups.ru) {
    return 301 https://$host$request_uri;
  }
  server_name  dr-mobile.so-ups.ru;
  listen 80;
  return 404;
}
```

12. Для применения настроек необходимо перезагрузить Nginx командой:

```
sudo systemctl restart nginx
sudo systemctl enable nginx
```

Установка и настройка сервиса **mobile-web** закончена. Для проверки работоспособности Nginx необходимо выполнить команду:

```
systemctl status nginx |grep active
```

Ожидаемый ответ:

```
Active: active (running)
```

Для проверки работоспособности сервиса **mobile-web** необходимо перейти по web-ссылке, соответствующей имени сайта сервиса **mobile-web**. Ожидаемый результат – отображение стартовой страницы сервиса **mobile-web**.

В случае, если стартовая страница приложения не загружается, рекомендуется обратиться к лог файлам Nginx и устранить зафиксированную в них проблему. Для просмотра лог файлов Nginx (если не менялись пути в конфигурационном файле выше) необходимо воспользоваться следующими командами:

```
#Error лог
sudo cat /var/log/nginx/error.log
#Access лог
sudo cat /var/log/nginx/access.log
```