Инструкция для моулей ADO (Abanking Digital Office)

Для того, чтобы развернуть сервисы Личного Кабинета Регистрации бизнеса + РКО ADO модули:

- «Abanking Digital Office - Регистрация бизнеса»
- «Abanking Digital Office - Личный кабинет клиента»
- «Abanking Digital Office - Личный кабинет администратора»
- «Abanking Digital Office - Личный кабинет оператора»
- «Abanking Digital Office - Личный кабинет агента. Веб приложение»
- «Abanking Digital Office - Личный кабинет агента. Мобильное приложение»

необходимо перейти по ссылке на **ftp-сервер по адресу** 93.171.206.140 (конкретная директория и реквизиты доступа направляются лично лицензиату)

Состав поставки:

1. config.tar.gz - содержит директории с файлами конфигурации (docker-compose.yml, .env & etc).
2. images.tar.gz - содержит все необходимые образы, которые надо загрузить в докер-среду на сервере.
3. license-keys/ - директория с ключами лицензирования
4. Инструкция по развертыванию

Состав сервисов:

1. Фронтенд и бекенд сервисы (директория release):

   1.1. lkk-front - фронтенд ЛК Клиента - к модулю «Abanking Digital Office - Личный кабинет клиента». Адрес демонстрационного стенда: http://lkk-front-rko.test.abanking-dev.ru/

   1.2. lka-front - фронтенд ЛК Агента - к модулю «Abanking Digital Office - Личный кабинет агента. Веб приложение». Адрес демонстрационного стенда: http://lka-front-rko.test.abanking-dev.ru/account/login

   1.3. lko-front - фронтенд ЛК Оператора – к модулю «Abanking Digital Office - Личный кабинет оператора». Адрес демонстрационного стенда: http://lko-front-rko.test.abanking-dev.ru

   1.4. cms-front - фронтенд CMS – к модулю «Abanking Digital Office - Личный кабинет администратора». Адрес демонстрационного стенда: http://cms-front-rko.test.abanking-dev.ru/account/login

   1.5. license-front - фронтенд сервера лицензирования – к модулю «Сервер лицензирования». Адрес демонстрационного стенда: http://license-front-rko.test.abanking-dev.ru/

   1.6. cs-api - бекенд сервиса подтверждения

   1.7. idsrv-api - бекенд сервера авторизации

   1.8. kaluga-api - бекенд сервиса интеграции с Калуга-Астрал

   1.9. license-api - бекенд сервера лицензирования

   1.10. lk-api - бекенд сервиса основной логики личного кабинета

   1.11. ns-api - бекенд сервиса уведомлений

   1.12. crm-connector-api - бекенд сервиса интеграции с CRM

2. Сервисы инфраструктуры (директория services)

2.1. rabbitmq - менеджер очередей

2.2. postgres - сервер БД

2.3. kannel - коннектор к SMS-шлюзу по протоколу

2.4 smsbox - smsbox

2.5. nginx - маршрутизатор запросов на сервисы

Список названий баз данных созданных в postgresql:

1. crm-connector-api
2. cs-api
3. idsrv-api
4. kaluga-api
5. license-api
6. lk-api
7. ns-api

Требования к инфраструктуре:

Так как сервисы запускаются как докер контейнеры, то могут запускаться на любой ОС, но в целях лучшей производительности рекомендуется запускать их на Linux системах. Обязательным требованием для запуска на хостовой машине является наличие следующего предустановленного ПО:

1. Docker (версия не ниже 19.03.0)
2. docker-compose (версия не ниже 1.20.0)

Также рабочая конфигурация предполагает наличие веб-сервера реверс-прокси, например nginx, на сервисы docker. Вебсервер может располагаться как на одной машине с сервисами, так и отдельно, основным требованием является наличие доступа веб-сервера к порту хостовой машине, на которой запущены сервис nginx маршрутизатора запроса на сервисов, по умолчанию 10000.

У хостовой машины сервисов должны быть доступы
для исходящих запросов:

- SMTP-сервера
- CMC-шлюза
- АПИ CRM
- АПИ Калуга Астрал. https://dss.astral.business/

АПИ DaData (сервис получения дополнительной информации из справочников ФИАС и т.п.)

https://suggestions.dadata.ru/ АПИ DBrain (сервис распознавания сканов документов OCR).

https://latest.dbrain.io/ для входящих запросов

- АПИ CRM

1.1. Порядок установки

1.1.1. Распаковка контейнеров

1. Создать в корне сервера директорию /docker-compose/, куда распаковать содержимое архива ado-config.tar.gz
    a. перейти в директорию docker-compose cd /docker-compose/
    b. скопировать архив sudo cp ado-config.tar.gz ./
2. Распаковать содержимое архива ado-images.tar.gz в любую другую директорию.
    a. распаковка архива sudo tar -xvzf ado-config.tar.gz
    b. удаление архива rm ado-config.tar.gz (необязательно)
    c. распаковка докер-образов sudo tar -xvzf ado-images
    d. перейти в директорию, которая содержит скопированные архивы ado-images.tar.gz и ado-config.tar.gz e. удалить архивы sudo rm ado-*
3. Загрузить все образы командой в докер-среду.

```
[sudo] docker load -i lk-api.tar
```

или one-line командой (работает на OS CentOS 7/8):

```
for i in $(ls -la | awk '{print $9}' | sed 1,3d); do docker load -i $i; done
```

4. Обратите внимание образы передаются запакованные в архив .tar.gz , перед установкой надо разархивировать архив, содержащий образ

1.1.2. Запуск

1. Создать виртуальную сеть в докере для работы сервисов (этот пункт опционально - в поставке нет настроек сети в docker-compose.yml).

```
[sudo] docker network create service-net
```

или

```
[sudo] docker network create -d overlay service-net
```

или же любую на свое усмотрение, но стоит помнить о том, что docker-compose release и service надо будет поправить на новую, созданную сеть.

2. Опционально - в файле .env главного docker-compose.yaml можно изменить название vhost'a у rabbitmq. Строчка ApplicationServices:Environment:Name=some-name.
3. Запустить сервисы, необходимые для работы приложений, из директории "/docker-compose/services"

```
[sudo] docker-compose up -d
```

Примечание: если нет необходимости размещение БД postgres в docker образе, то её можно исключить из сервисов удалением соответствующей ей конфигурации из "docker-compose/services/docker-compose.yml" По умолчанию база из контейнера открыта на 5433 порт.

4. На сервере баз данных создать базы данных: cs-api, idsrv-api, license-api, lk-api, ns-api. В случае, если используется база не из docker образа, то изменить параметр "Dal:ConnectionString" в ".env" файлах в соответствующих таблицам категориях "docker-compose/release".
5. Отредактировать параметры запуска контейнеров в .env файлах, находящихся в директории сервиса. Те параметры которые требуется изменить дополнительно - отмечены в конфигурации комментарием с `REDACT_ME`. Подробнее о настройках в блоке "Настройки".
6. Отредактировать в "docker-compose/nginx/conf.d/default.conf" server_name для каждого сервера
7. Добавить проксирование с веб-сервера реверс-прокси на 10000 порт [на 10000 порту находится nginx]

Пример настройки вебсервера реверс-прокси nginx

```
server {
  # публичные домены ЛК Клиента, ЛК Агента,
  # панели управления сервера лицензирования соответственно
  server_name
    lkk-front.ru
    lka-front.ru
    license-front.ru;

  listen 80;

  location / {

     client_max_body_size 30m;

     proxy_set_header  Host $host;
     proxy_set_header  X-Forwarded-Proto $scheme;
     proxy_set_header  X-Real-IP $remote_addr;
     proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;

     # блок настроек необходимый для проксирования websocket запросов
     proxy_set_header Upgrade $http_upgrade;
     proxy_set_header Connection $http_connection;
     proxy_cache_bypass $http_upgrade;
     proxy_http_version 1.1;

     proxy_pass        http://127.0.0.1:10000/;


}
```

8. Запустить образы приложений из директории "/docker-compose/release"

```
[sudo] docker-compose up -d
```

9. В переданных дампах БД уже выполнены все скрипты инициализации, можно развернуть их, или создать БД с нуля. Для этого выполнить методы инициализации. В постмане, либо в curl, выполнить запросы: POST {{endpoint}}/internal /api/seed/namedseed с Basic авторизацией, (по умолчанию реквизиты ApiResource_LK/QXBpUmVzb3VyY2VfTEtfU2VjcmV0) со следующими значениями, передаваемыми в теле запроса для каждого сервиса.

Сервис	Сиды
lk-api:	"Workflow" "DictionarySeed" "OpenBusinessWorkflowSeed"
idsrv-api	"Seed"
ns-api	"AbankingSeeds" "BusinessSeeds"
cs-api	"BaseSeeds"

Пример запроса постман:

Пример запроса curl:

```
[                    ]$ curl -iL -X POST '                                    /internal/api/seed/namedseed' \
> -H 'Authorization: Basic                                                    ' \
> -H 'Content-Type: application/json' \
> -d '"OpenBusinessWorkflowSeed"'

HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Fri, 09 Apr 2021 03:26:06 GMT
Content-Type: application/json
Content-Length: 0
Connection: keep-alive
```

Пример запроса curl

```
curl -iL -X POST 'http://lk-api/internal/api/seed/namedseed' \
-H 'Authorization: Basic QXBpUmVzb3VyY2VfTEs6UVhCcFVtVnpiM1Z5WTJWZlRFdGZVMlZqY21WMA==' \
-H 'Content-Type: application/json' \
-d '"OpenBusinessWorkflowSeed"'
```

Также можно передать заполненную данными postgresql в архиве tar. Чтобы развернуть БД с готовыми данными, надо извлечь содержимое из архива postgres.tar.gz в директорию с docker-compose.yaml, отвечающим за сервисы инфраструктуры (docker-compose/services). В архиве находится папка с названием postgres. В ней находятся все данные и настройки БД. После чего выполнить команду `docker-compose up -d` и убедиться, что все работает - `dockercompose ps` - все контейнеры должны быть в состоянии Up.

10. Выполнить запрос инициализации сервера лицензирования (заполнение базы данных начальными данными необходимыми для работы сервиса лицензирования):

    `: GET http://license-api/api/seed/init`

11. Зайти http://license-front.ru/, авторизоваться под администратором. Логин: "admin", пароль: "Pulse1data!"; и загрузить конфигурацию и подпись в интерфейсе загрузки ключа из директории "license-keys". Инструкция по работе с администрированием LS

1.2. Настройки

Существуют следующие настройки сервисов

1. .env-файлы, которые задают переменные окружения запуска контейнеров. Настройки .env-файла, находящегося на одном уровне с docker-compose.yml файлов, являются общими для всех сервисов. В нём также указаны переменные с версиями образов. Настройки, находящиеся в .env-файле в директории сервиса, например `lk-api/.env`, используются только для этого сервиса.

2. abanking-settings.json. json файл настроек фронтенд сервиса, находится в директории ./settings/ внутри директории сервиса фронтенда: например ./lkk-front/settings/abanking-settings.json.
3. partial/ директория внутри директории фронтенд сервиса. Содержит в себе html файлы, внедряемые в разные места index.html фронтенда, например ./lkk-front/partial/head.html. Внедряет html внутри заголовка head индексной страницы.

   Таким образом можно внедрять код метрик, а также SEO-теги в index.html.
4. theme/ - директория с настройками тематизации фронтенда.
5. kannel.conf. Настройки коннектора к SMS-шлюзу по протоколу SMPP, находится в директории /docker-compose/sevices /kannel.

Большинство настроек заданы по умолчанию и менять их перед запуском не требуется. Перед запуском необходимо указать следующие настройки:

- в корневом .env файле: актуализировать, при необходимости, версии докер образов сервисов фронтенда и бекенда; прописать эндпоинт идентити-сервера; указать свой clientCompanyName,

```
# Версии включаемых образов
НАЗВАНИЕ_СЕРВИСА=тэг образа
# Пример
LK_API_TAG=rko-release-17

#Публичный домен identity-server
PublicAppUri=http://idsrv-api.ru
#Адрес Identity Server
IdentityServer:Authority=http://idsrv-api.ru

ClientCompanyName=abanking
```

в файле настроек ./crm-connectora-api/.env . адреса конечных точек, реквизиты Basic авторизации, для интеграции с АПИ CRM

```
#Флаг включения отправки колбеков при смене шаге документа
WebHooks:Crm:StepInfoEnabled=true

#Флаг включения отправки колбеков о зарегистрированном пользователе
WebHooks:Crm:RegisterInfoEnabled=true

#Флаг включения отправки информации о созданном документе
WebHooks:Crm:DocumentInfoEnabled=true

#Адрес хоста для получателя колбеков
WebHooks:Crm:Host=

#Флаг включения отправки колбеков при смене шаге документа
WebHooks:Crm:StepInfoPath=/api/stepinfo

#Флаг включения отправки колбеков о зарегистрированном пользователе
WebHooks:Crm:RegisterInfoPath=/api/registerinfo

#Эндпоинт отправки информации о созданном документе
WebHooks:Crm:DocumentInfoPath=/api/documentinfo

#Эндпоинт для авторизации системы
WebHooks:Crm:AuthorizationPath=/api/auth/sign_in

#Учетная запись системы
WebHooks:Crm:Login=
WebHooks:Crm:Password=

#Флаг включения отправки запросов в CRM
WebHooks:Crm:IsEnabled=true
```

в файле настроек ./kaluga-api/.env - ключ АПИ Калуга Астрал

```
#Боевой токен Калуга
KalugaAstral:Token=
```

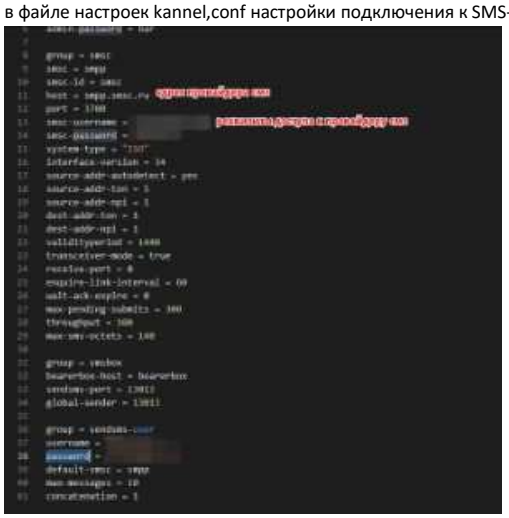
в файле настроек ./lk-api/.env настройки шаблона урла для формирования ссылок на форму заполнения файлов с мобильного, настройки ключа АПИ Dbrain. ключ АПИ DaData, Email для отправки Отправка xml архива

```
#Ссылка для уведомлений
UploadFileLinkTemplate=file-uploading/{0}/{1}

#Шаблон УРЛ для коротких ссылок
ForwardUrlTemplate='api/forward/{0}

#Включение отправки данных об успешной регистрации документа в ФНС, отправляемые на email пользователя
Business:XmlNotification:Enabled=true

#E-mail пользователя куда будут отправлены данные об успешной регистрации документа в ФНС
Business:XmlNotification:Email=

#E-mail для мониторинга заявок в банке (отправляются E-mail при переходе заявки в определенные статусы)
Business:Monitor:Email=

#Токен Dbrain
Dbrain:ApiKey=

# Ключ АПИ DaData для отправки запросов в DaData
DadataSettings:Token=Token
```

в файле настроек ./ns-api/,env указать настройки SMTP-сервера

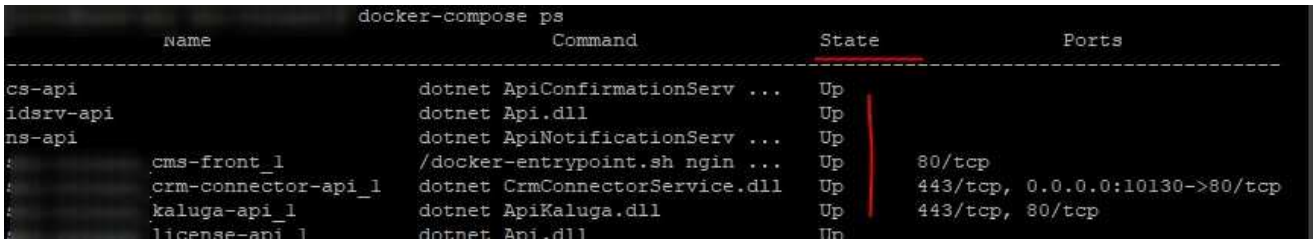
```
#адрес SMTP-сервера
Smtp:Host=smtp.yandex.ru

#пользователь SMTP-сервера
Smtp:User=

#пароль пользователя SMTP-сервера
Smtp:Password=

# Адрес СМС шлюза
Smpp:Host=http://smsbox:13013/cgi-bin/sendsms

# Пользователь СМС шлюза (должен совпадать с параметром username в kannel.conf)
Smpp:UserName=

# Пароль от СМС шлюза (должен совпадать с параметром password в kannel.conf)
Smpp:Password=
```

в файле настроек kannel,conf настройки подключения к SMS-шлюзу по протоколу SMPP



1.3. Проверка успешности запуска

Для начало следует убедиться, что сервисы запустились. Для этого нужно выполнить команду, находясь в директории с docker-compose.yaml файлом: `docker-compose ps`, и посмотреть на столбец "State" - все значения должны быть в значении "Up".



Если какой-то контейнер будет в любом другом статусе, то следует проверить его логи командой `docker logs –f container_name`. Возможные причины, почему контейнер не запускаются, могут быть различны. Например - неправильная конфигурация в .env файле. Подробнее будет в выводе лога контейнера.