

eXpress

Система
коммуникаций

Настройка для Microsoft Outlook

Руководство по установке

Версия manifest (OWA/macOS/Windows) 1.1.2.



© Компания «Анлимитед продакшен», 2026. Все права защищены.

Все авторские права на эксплуатационную документацию защищены.

Этот документ входит в комплект поставки изделия. На него распространяются все условия лицензионного соглашения. Без специального письменного разрешения компании «Анлимитед продакшен» этот документ или его часть в печатном или электронном виде не могут быть подвергнуты копированию или передаче третьим лицам с коммерческой целью.

Информация, содержащаяся в этом документе, может быть изменена разработчиком без специального уведомления, что не является нарушением обязательств по отношению к пользователю со стороны компании «Анлимитед продакшен».

Указанные в документе адреса серверов, значения конфигурационных файлов, учетные пользовательские данные указаны для примера и носят исключительно ознакомительный характер. Пользовательские данные, в том числе биометрические, вымышленные и не содержат персональных данных.

Предоставляемые компоненты СК «Express» в составе поставки предназначены исключительно для демонстрации функциональности и не предназначены для эксплуатации в продуктивной среде. Для корректного функционирования СК «Express» требуется разработка архитектурной схемы инсталляции с учетом специфики инфраструктуры до продуктивной эксплуатации.

Почтовый адрес:	127030, г. Москва, ул. Новослободская, д. 24, стр. 1
Телефон:	+7 (499) 288-01-22
Email:	sales@express.ms
Web:	https://express.ms/

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	3
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	4
ГЛАВА 1	
ОБЩИЕ ТРЕБОВАНИЯ.....	5
Требования к уровню подготовки администратора	5
Требования к программным и техническим средствам	5
ГЛАВА 2	
АРХИТЕКТУРА	6
Single CTS или разделенный CTS	7
Несколько CTS	8
Несколько CTS и ETS	9
ГЛАВА 3	
УСТАНОВКА НАДСТРОЙКИ ДЛЯ OUTLOOK	10
Шаг 1. Обновление серверной части (опционально)	10
Шаг 2. Подготовка БД	10
Шаг 3. Развертывание в Docker express-core-macos-service	10
Шаг 4. Развертывание надстройки для Outlook через Манифест	11
ПРИЛОЖЕНИЕ 1	
ДОБАВЛЕНИЕ СЕРТИФИКАТА В КОНТЕЙНЕР EXPRESS-MEETING-CORE-MACOS-SERVICE	13
ПРИЛОЖЕНИЕ 2	
СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ.....	15

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Термин	Определение
AD	Active Directory — служба каталогов корпорации Microsoft для операционных систем семейства Windows Server
API	Application Programming interface — интерфейс для взаимодействия программ и приложений
CTS	Corporate Transport Server — корпоративный сервер
Docker	платформа контейнеризации с открытым исходным кодом, с помощью которой можно автоматизировать создание приложений, их доставку и управление
Docker-compose	Надстройка(утилита) над Docker, приложение на Python, которое позволяет запускать множество контейнеров одновременно и маршрутизировать потоки данных между ними.
ETS	Enterprise Transport Server — сервер предприятия
express-core-service	Серверное приложение для надстройки для Outlook, обеспечивающее основную функциональность (создание и изменение конференций)
express-template-service	Серверное приложение для надстройки для Outlook, обеспечивающее дополнительную функциональность (создание и изменение шаблонов письма, текст, логотип, подпись и т. п.)
KeyCloak	Приложение для реализации единой точки идентификации, аутентификации и авторизации. Это система идентификации и управления доступом с открытым исходным кодом (Open Source)
MS Exchange	Почтовый сервер Microsoft
MS Outlook	Почтовый клиент Microsoft, часть пакета приложений MS Office
Single CTS	Единый корпоративный сервер
БД	База данных
Контейнер	Стандартизированный, изолированный и портативный пакет программного обеспечения, который включает в себя все необходимое для запуска приложения, включая код, среду выполнения, системные инструменты, библиотеки и настройки.
Манифест приложения	XML-документ, который либо встроен в исполняемый файл, либо содержится в отдельном XML-файле, сопровождающем его. Он содержит имя, версию, информацию о доверии, привилегии, необходимые для выполнения, и зависимости от других компонентов, ссылки на серверную часть ПО.
Лог	Запись в журнале событий сервера и/или клиента
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
Приглашенный участник	Пользователь(и), которого(ых) пригласил создатель конференции Express с помощью надстройки Outlook
Разделенный CTS	Разделенный корпоративный сервер: Front CTS и Back CTS
СК «Express», Express, система	Система коммуникаций «Express»
Создатель конференции	Пользователь, который создает и может редактировать конференцию Express с помощью надстройки Outlook
Чат-бот	Чат-бот Conference Notifier Bot

Глава 1

ОБЩИЕ ТРЕБОВАНИЯ

ТРЕБОВАНИЯ К УРОВНЮ ПОДГОТОВКИ АДМИНИСТРАТОРА

Выполнение описанных ниже операций требует от администратора следующей квалификации:

- администрирование Keycloak;
- администрирование Linux;
- администрирование Docker;
- администрирование MS Exchange;
- понимание XML.

ТРЕБОВАНИЯ К ПРОГРАММНЫМ И ТЕХНИЧЕСКИМ СРЕДСТВАМ

Перед началом работ убедитесь, что программные и технические средства соответствуют следующим требованиям:

- выполнены все требования, описанные в [инструкции](#) по развертыванию msi-версии [плагина](#) под Keycloak аутентификацию;
- развернут ExpressMeeting Core Service с Keycloak аутентификацией msi-версии не ниже 1.2.0.0;
- версии MS Exchange + OWA не ниже 2016 CU19 (другие способы распространения манифеста не тестировались);
- версия клиентской части Outlook под MS Windows не ниже MS Office 2016 (если планируется использовать данный плагин под MS Windows);
- настроена MS Exchange OWA (если планируется использовать данный плагин под MS Exchange OWA);
- версия клиентской части Outlook под MacOS не ниже 16.66.1 (если планируется использовать данный плагин под MacOS);
- открыта данная [ссылка](#) с клиентских ПК (для корректной работы плагина в MS Exchange OWA);
- в профиле пользователя MS Outlook заполнено поле «Полное имя» ([Рисунок 1](#)).

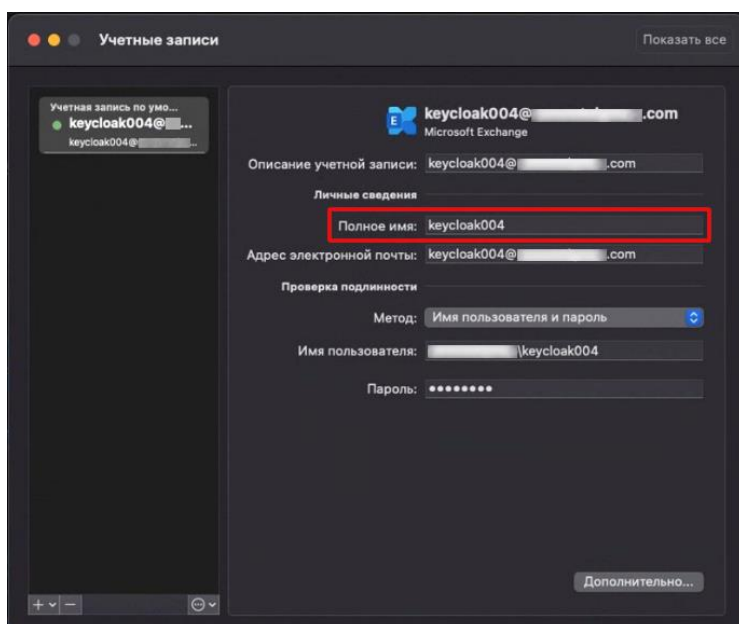


Рисунок 1. Настройки профиля пользователя

Глава 2

АРХИТЕКТУРА

В зависимости от используемого архитектурного решения СК «Express» применяются различные схемы развертывания серверной части надстройки для Outlook. Наиболее распространенные типовые архитектурные решения описаны в [инструкции](#) по развертыванию msi-версии плагина под Keycloak аутентификацию. В случае использования Манифест-версии в архитектурную схему Keycloak плагина добавляется еще один докер-контейнер, express-macos-core-service и контейнер, обеспечивающий SSL соединение, например Nginx.

Возможные схемы для развертывания:

- [Single CTS или разделенный CTS](#);
- [Несколько CTS](#);
- [Несколько CTS и ETS](#).

Конкретные схемы развертывания определяются требованиями заказчика после консультации с компанией-разработчиком.

При использовании единого шаблона письма все express-core-service могут использовать одну БД. При использовании разных шаблонов письма для разных express-core-service разверните для каждого из них свою БД и свой express-template-service.

SINGLE CTS ИЛИ РАЗДЕЛЕННЫЙ CTS

Если СК «Express» содержит один CTS-сервер, то разверните серверную часть надстройки в одном экземпляре. Ниже представлена типовая схема развертывания данного типа ([Рисунок 2](#)):

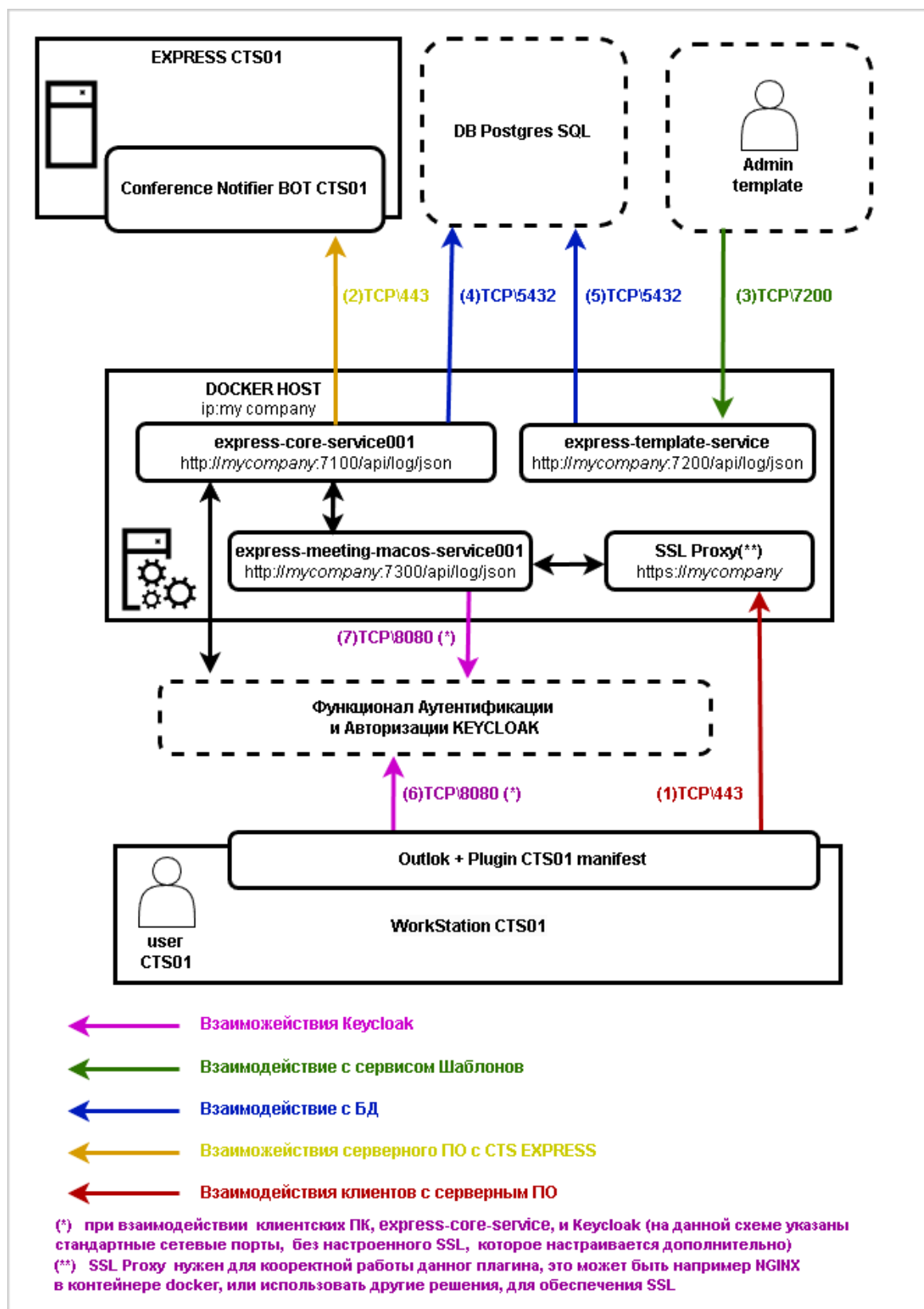


Рисунок 2. Типовая схема: 1 CTS, 1 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в Таблица 1 Приложения 2.

НЕСКОЛЬКО CTS

Если СК «Express» содержит несколько CTS-серверов, то разверните серверную часть надстройки для каждого CTS-сервера отдельно (допускается развертывание на одном сервере Docker, но с разными портами и именами контейнеров, или на нескольких серверах Docker). Ниже представлена типовая схема развертывания данного типа (Рисунок 3):

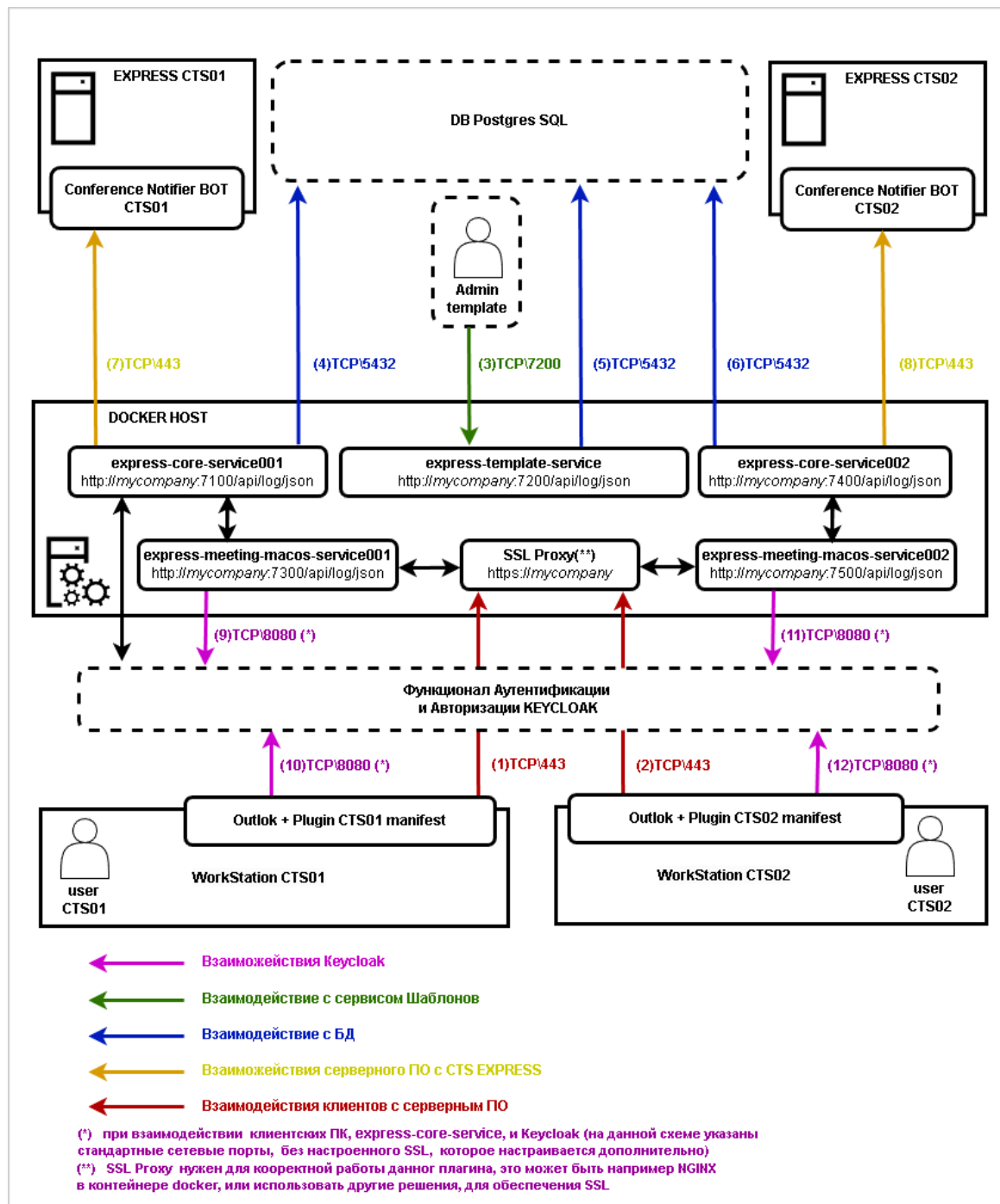


Рисунок 3. Типовая схема: 2 CTS, 2 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в Таблице 2 Приложения 2.

НЕСКОЛЬКО CTS И ETS

Если СК «Express» содержит несколько CTS-серверов, объединенных ETS-сервером, разверните серверную часть надстройки только для ETS-сервера. Ниже представлена типовая схема развертывания данного типа ([Рисунок 4](#)):

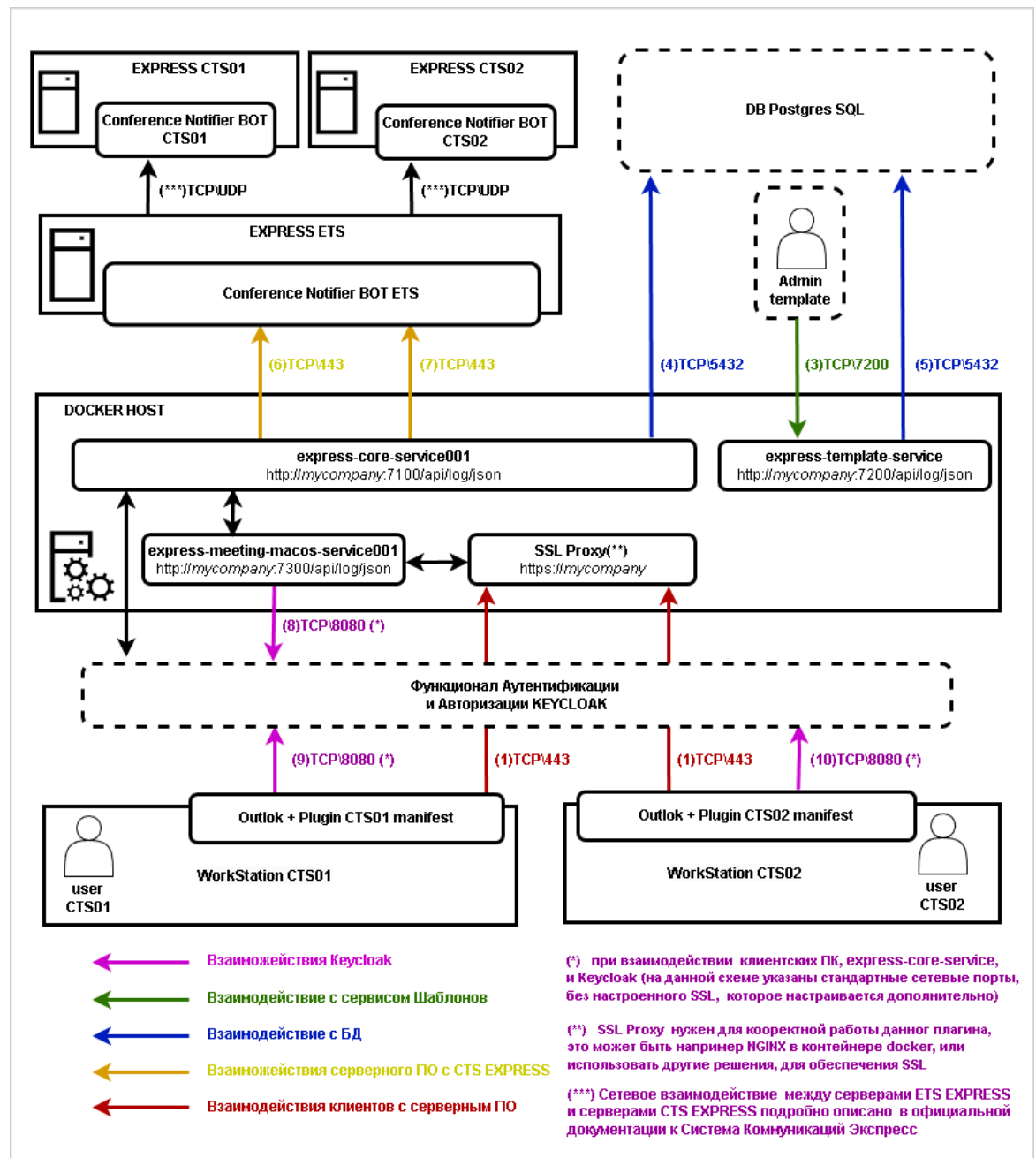


Рисунок 4. Типовая схема: 1 ETS, 2 CTS, 1 express-core-service, 1 express-template-service

Номера сетевых взаимодействий соответствуют номеру строки в [Таблице 3 Приложения 2](#).

Глава 3

УСТАНОВКА НАДСТРОЙКИ ДЛЯ OUTLOOK

Для установки надстройки для Outlook Manifest выполните шаги, описанные ниже. Дистрибутив — серверная версия и файл манифеста доступен для скачивания по [ссылке](#) (выберите текущую версию продукта).

ШАГ 1. ОБНОВЛЕНИЕ СЕРВЕРНОЙ ЧАСТИ (ОПЦИОНАЛЬНО)

Шаг выполняется, если сервис уже был развернут в рамках предыдущей версии надстройки.

Для обновления серверной части:

1. Загрузите новую версию image-образов сервисов express-meeting-macos-core-service. В папках, где лежат файлы форматов ENV и YML, запустите команду:

```
docker-compose --env-file ./env -f ./docker-compose.yml down
```
2. Из новой версии дистрибутива возьмите новые файлы форматов .env и .yml и укажите в них ваши параметры и новые image-образы.
3. Замените старые файлы форматов ENV и YML на новые в соответствующей папке на сервере Docker.
4. Выполните [Шаг 2](#) текущей главы для обновления структуры БД.
5. Запустите из папки docker контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

ШАГ 2. ПОДГОТОВКА БД

БД уже должна быть подготовлена, в рамках развертывания ExpressMeeting Core Service с Keycloak аутентификацией ([инструкция](#) по развёртыванию msi-версии плагина под Keycloak аутентификации).

ШАГ 3. РАЗВЕРТЫВАНИЕ В DOCKER EXPRESS-CORE-MACOS-SERVICE

Для развертывания сервиса:

1. Загрузите из папки images-образов сервисов в docker:

```
docker load --input express-meeting-macos-service.tar
```
2. Подготовьте настройки запуска сервисов в файле docker\env.

Файл содержит следующие настройки:

- наименование образа, из которого стартует контейнер с core-service (не нужно менять):

```
EXPRESS_MEETING_SERVICE_IMAGE_NAME=express-meeting-macos-service
```
- порт, по которому будет доступен core-service:

```
SERVICE_PORT=3001
```
- имя файла с настройками сервиса (не нужно менять):

```
EXPRESS_MEETING_ENV_FILE=service.env
```
- папка контейнера с логами сервиса (не нужно менять):

```
EXPRESS_MEETING_DOCKER_LOGS_PATH=/public/express-meeting-logs
```
- папка docker-хоста, в которую будут добавляться логи из контейнеров сервисов (при необходимости изменить на нужную):

```
EXPRESS_MEETING_LOGS_PATH=/public/express-outlook/logs
```

3. Подготовьте настройки запуска сервиса в файле docker\service.env.

Файл содержит следующие настройки:

- адрес Keycloak (указать реальный):
`KEYCLOAK_URL=https://keycloak.mycompany`
- realm, который используется для авторизации в Keycloak (указать реальный):
`KEYCLOAK_REALM=realm`
- значение Client ID, который используется для авторизации в Keycloak (указать реальный):
`CLIENT_ID=client`
- значение Client Secret, который используется для авторизации в Keycloak (указать реальный):
`CLIENT_SECRET=secret`
- среда выполнения сервиса (не нужно менять):
`NODE_ENV=production`
- Порт, на котором будет запущен сервис (должен совпадать с портом, который указан в п.2 в параметре SERVICE_PORT):
`PORT=3001`
- Адрес сервиса, по которому он доступен с клиентской машины (с машины, на которой установлен outlook):
`SERVER_SOURCE=https://service.mycompany`
- Адрес core-сервиса (см. требование о необходимости развернуть Express Meeting Core Service из п.1 Общие требования), по которому он доступен с docker-хоста:
`EXPRESS_SERVICE_URL=http://core-service.mycompany`
- Папка контейнера с логами сервиса (не нужно менять, должна совпадать с папкой, указанной в п.2 в параметре EXPRESS_MEETING_DOCKER_LOGS_PATH):
`LOG_FOLDER=/public/express-meeting-logs`

4. Запустите из папки docker-контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

Так как плагин может обращаться к сервису только по https-протоколу, а сам сервис в docker работает по http-протоколу, необходимо настроить перенаправление https трафика на http-вход сервиса (например, используя nginx).

После запуска контейнеров, если в файле docker\core.env в ключах ExpressOptions__Uri, AuthorizationOptions__Configuration__Url были указаны HTTPS-адреса, в случае необходимости (например, не доступен удостоверяющий центр для используемых компанией сертификатов) добавьте в контейнер express-meeting-core-service-main полную цепочку сертификатов для соответствующих HTTPS-соединений. Пример добавления сертификатов в контейнер описан в [Приложении 1](#).

ШАГ 4. РАЗВЕРТЫВАНИЕ НАДСТРОЙКИ ДЛЯ OUTLOOK ЧЕРЕЗ МАНИФЕСТ

Для развертывания надстройки для Outlook через Манифест:

1. Откройте в редакторе файл OutlookPlugin\manifest.xml.
2. Замените в файле OutlookPlugin\manifest.xml:
 - 'mycompany' на dns-имя хоста, по которому доступен сервис, развернутый в Шаге 1 (например, если сервис доступен по адресу `https://service.company:4587`, то 'mycompany' необходимо заменить на

'service.company'). Замене подлежат все встречающиеся в файле 'mycompany';

- ':443' на порт, по которому доступен сервис, развернутый в Шаге 1 (например, если сервис доступен по адресу <https://service.company:4587>, то ':443' необходимо заменить на ':4587'). Замене подлежат все встречающиеся в файле ':443'.

3. Загрузить в Outlook надстройку manifest.xml. Подробнее см. в [статье](#).

Приложение 1

ДОБАВЛЕНИЕ СЕРТИФИКАТА В КОНТЕЙНЕР EXPRESS-MEETING-CORE-MACOS-SERVICE

Данный раздел описывает типовое решение по установке сертификатов в контейнер Docker. Процедура может меняться в зависимости от инфраструктуры и архитектурного решения.

Важно! Данная процедура опциональна. Для корректной работы SSL в контейнер следует поместить все сертификаты в цепочки сертификатов (корневой сертификат, промежуточные сертификаты, конечный сертификат).

Для добавления сертификата в контейнер:

1. Соберите все сертификаты в цепочки в виде файлов формата CRT ([Рисунок 5](#)):

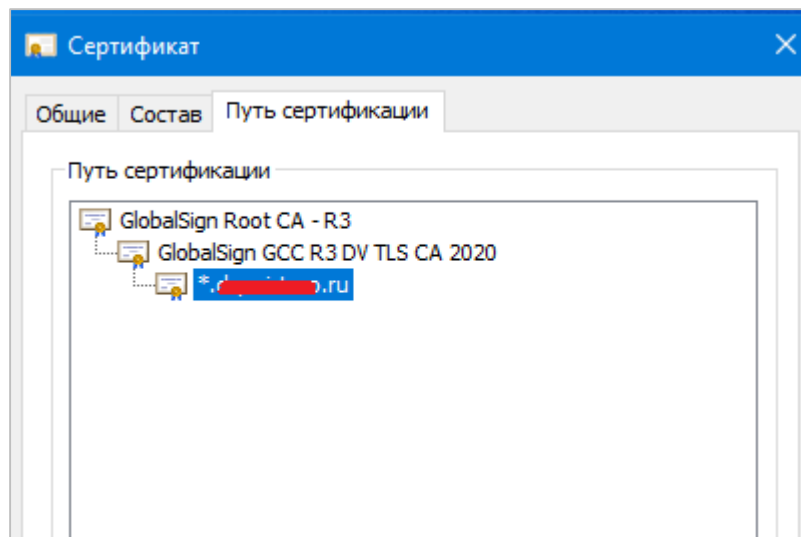


Рисунок 5. Свойства сертификата

2. Создайте на сервере Docker, где развернут контейнер express-meeting-core-service-main, папку. Пример названия папки: /opt/express/bots/cert/.
3. Переместите файлы сертификатов, собранные в п. 1, в папку, созданную в п. 2.
4. Выдайте папке /opt/express/bots/cert/ соответствующие права доступа.
5. В файле docker-compose.yml в параметрах контейнера express-meeting-core-service-main ([Рисунок 6](#)) добавьте в разделе volumes следующий параметр:

```
- /opt/express/bots/cert/:/usr/local/share/ca-certificates/
```

Важно! Первая часть строки /opt/express/bots/cert/ может быть любой и должна соответствовать пути из п.2. Вторая часть данной строки :/usr/local/share/ca-certificates/ постоянная, менять ее запрещено. Из указанной папки express-meeting-core-service-main берет все необходимые сертификаты.

```
GNU nano 6.2 /opt/express/bots/outlook3000/docker-compose.yml *
version: "3.3"

services:
  core-service:
    image: ${EXPRESS_MEETING_CORE_SERVICE_MAIN_IMAGE_NAME:?error}:3.0.0.0
    container_name: ${EXPRESS_MEETING_CORE_SERVICE_MAIN_IMAGE_NAME:?error}
    env_file: ${EXPRESS_MEETING_CORE_ENV_FILE:?error}
    ports:
      - "${EXPRESS_MEETING_CORE_SERVICE_PORT:?error}:7100" # При необходимости отредактируйте порт хоста (первый)
    restart: always
    extra_hosts:
      - "${EXPRESS_MEETING_EXTRA_HOST}"
      - "${EXPRESS_MEETING_EXTRA_HOST2}"
    volumes:
      - type: bind
        source: "${EXPRESS_MEETING_LOGS_PATH:?error}/express-meeting-core-service"
        target: "${EXPRESS_MEETING_DOCKER_LOGS_PATH:?error}"
      - /opt/express/bots/keycloak/cert/crt-cert:/usr/local/share/ca-certificates/
        bind:
          create_host_path: true
#
#
  template-service:
    image: ${EXPRESS_MEETING_TEMPLATE_SERVICE_MAIN_IMAGE_NAME:?error}:3.0.0.0
    container_name: ${EXPRESS_MEETING_TEMPLATE_SERVICE_MAIN_IMAGE_NAME:?error}
    env_file: ${EXPRESS_MEETING_TEMPLATE_ENV_FILE:?error}
    ports:
      - "${EXPRESS_MEETING_TEMPLATE_SERVICE_PORT:?error}:80" # При необходимости отредактируйте порт хоста (первый)
    restart: always
    volumes:
      - type: bind
        source: "${EXPRESS_MEETING_LOGS_PATH:?error}/express-meeting-template-service"
        target: "${EXPRESS_MEETING_DOCKER_LOGS_PATH:?error}"
        bind:
          create_host_path: true
#
#
```

Рисунок 6. Параметры `docker-compose.yml`

- Остановите контейнеры надстройки Outlook, запустив из папки с конфигурационными файлами командой:

```
docker-compose down
```

- Запустите из папки docker-контейнеры:

```
docker-compose --env-file ./env -f ./docker-compose.yml up -d
```

- Зайдите в контейнер `express-meeting-core-service-main` с помощью команды:

```
docker exec -it express-meeting-core-service-main bash
```

- Обновите список сертификатов командой:

```
update-ca-certificates
```

- Выйдите из контейнера `express-meeting-core-service-main` с помощью команды:

```
exit
```

- Попробуйте создать конференцию с помощью клиентского ПО.

- Проверьте систему на отсутствие ошибок, связанных с SSL, с помощью команды:

```
docker logs --tail=200 -f express-meeting-core-service-main
```

Приложение 2

СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Таблица 1 Single CTS или разделенный CTS

№ п/п	Источник	Получатель	Порт и протокол	Описание взаимодействия
1	Внутренний пользователь (клиентский ПК)	SSL PROXY (docker Host)	443\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с SSL PROXY на сервере docker Host (указан стандартный сетевой порт)
2	express-core-service (docker Host)	CTS Front (CTS Single)	443\TCP	Взаимодействие express-core-service на docker Host с API Conference Notifier Bot на сервере CTS Front (CTS Single)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service001 на docker Host через веб-интерфейс (указан стандартный сетевой порт)
4	express-core-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	Внутренний пользователь (клиентский ПК)	Функционал Keycloak	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
7	express-core-macos-service (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-macos-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечание. В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице).

Таблица 2. Несколько CTS без ETS

№ п/п	Источник	Получатель	Порт и Протокол	Описание взаимодействия
1	Внутренний пользователь на CTS01 (клиентский ПК)	SSL PROXY (docker Host)	443\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с SSL PROXY на docker Host (указан стандартный сетевой порт)
2	Внутренний пользователь на CTS02 (клиентский ПК)	SSL PROXY (docker Host)	443\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с SSL PROXY на docker Host (указан стандартный сетевой порт)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service на docker Host через веб-интерфейс (указан стандартный сетевой порт)
4	express-core-service001 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	express-core-service002 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service002 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
7	express-core-service001 (docker Host)	CTS Front (CTS Single)01	443\TCP	Взаимодействие express-core-service001 на docker Host с API Conference Notifier Bot на сервере CTS Front (CTS Single)01
8	express-core-service002 (docker Host)	CTS Front (CTS Single)02	443\TCP	Взаимодействие express-core-service002 на docker Host с API Conference Notifier Bot на сервере CTS02 Front (CTS02 Single)02
9	express-core-service001 (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
10	Внутренний пользователь (клиентский ПК)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
11	express-core-service0021 (docker Host)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service002 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
12	Внутренний пользователь (клиентский ПК)	Функционал Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечание. В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице).

Таблица 3. Несколько CTS с ETS

№ п/п	Источник	Получатель	Порт и Протокол	Описание взаимодействия
1	Внутренний пользователь на CTS1(клиентский ПК)	SSL PROXY (docker Host)	443\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с SSL PROXY на docker Host (указан стандартный сетевой порт)
2	Внутренний пользователь на CTS2 (клиентский ПК)	SSL PROXY (docker Host)	443\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с SSL PROXY на docker Host (указан стандартный сетевой порт)
3	Администратор template (клиентский ПК)	express-template-service (docker Host)	7200\TCP	Взаимодействие администратора Шаблонов с express-template-service на docker Host через веб-интерфейс (указан стандартный сетевой порт)

№ п/п	Источник	Получатель	Порт и Протокол	Описание взаимодействия
4	express-core-service001 (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-core-service001 на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
5	express-template-service (docker Host)	БД PostgreSQL (сервер БД)	5432\TCP	Взаимодействие express-template-service на docker Host с БД PostgreSQL на сервере БД (указан стандартный сетевой порт)
6	express-core-service (docker Host)	ETS EXPRESS Server	443\TCP	Взаимодействие express-core-service на docker Host с API Conference Notifier Bot на сервере ETS EXPRESS Server
7	express-core-service (Docker Host)	ETS EXPRESS Server	443\TCP	Взаимодействие express-core-service001 на docker Host с API Conference Notifier Bot на сервере ETS EXPRESS Server
8	express-core-service001 (docker Host)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие express-core-service001 на docker Host с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
9	Внутренний пользователь (клиентский ПК)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS01 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)
10	Внутренний пользователь (клиентский ПК)	Функциональность Keycloak (docker Host)	8080\TCP	Взаимодействие надстройки Outlook на ПК пользователя на CTS02 с Keycloak на сервере Keycloak (указан стандартный сетевой порт)

Примечания:

- В таблице указаны стандартные сетевые порты для взаимодействия клиентских ПК, express-core-macos-service и Keycloak (SSL настраивается дополнительно и не учитывается в данной таблице);
- Сетевое взаимодействие между серверами ETS и серверами CTS подробно описано в [базе знаний](#).