

eXpress

Система
коммуникаций

Руководство администратора

Обновление

Сборка 3.42
23.06.2025



© Компания «Анлимитед продакшен», 2025. 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

ВВЕДЕНИЕ	4
ОБНОВЛЕНИЕ ОС	5
РУЧНОЕ ОБНОВЛЕНИЕ СЕРВЕРА	6
Single CTS	6
Front CTS и Back CTS	7
Обновление Express с версии 3.25	8
Подготовка перед обновлением серверов:.....	8
Обновление Media-сервера.....	9
Обновление Media на серверах Front или Single.....	10
Обновление Single, Front и Back серверов.....	12
Отказоустойчивая конфигурация	12
ОБНОВЛЕНИЕ POSTGRESQL	16
Процедура обновления	16
Резервная копия	17
Откат версии	17
Обновление без доступа к apt.postgresql.org	17
АВАРИЙНЫЕ СИТУАЦИИ ПРИ ОБНОВЛЕНИИ ИЗ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ REGISTRY	19
ПРОЦЕДУРА ОБНОВЛЕНИЯ СЕРТИФИКАТА	20
ОБНОВЛЕНИЕ НАСТРОЕК СЕРВЕРА ЗВОНКОВ VOEX	21
МИГРАЦИЯ MESSAGING ВЕРСИИ 3.10 И НИЖЕ НА 3.10 И ВЫШЕ	23
Определение размера таблицы	23
Миграция больших баз данных	24
Подготовка к миграции.....	25
Миграция.....	27
Обновление статистики	30
ИСТОРИЯ ИЗМЕНЕНИЙ	31

ВВЕДЕНИЕ

Руководство предназначено для администраторов изделия «Система коммуникаций «Express» (далее — СК «Express», Express, система). В нем содержатся сведения, необходимые для обновления и настройки системы.

Служба технической поддержки. Связаться со службой технической поддержки можно по электронной почте support@express.ms. Страница службы технической поддержки на сайте компании «Анлимитед продакшен» <https://express.ms/faq/>.

Сайт в интернете. Информацию о продукте компании «Анлимитед продакшен» представлена на сайте <https://express.ms/>.

ОБНОВЛЕНИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ

Для обновления операционной системы (далее – ОС) серверных компонентов Express:

1. Остановите контейнеры ПО Express. Для этого выполните команду:

```
cd /opt/express && DPL_PULL_POLICY=never dpl --dc stop
```

Затем выполните команду:

```
cd /opt/express-voice && DPL_PULL_POLICY=never dpl --dc stop
```

2. Убедитесь, что все контейнеры остановлены (находятся в статусе «exited»). Для этого на каждом сервере выполните команду:

```
docker ps -a
```

3. Выключите сервис Docker при помощи команды:

```
systemctl stop docker.service docker.socket
```

4. Выполните резервное копирование каталога /var/lib/docker или сделайте резервную копию образа виртуальной машины.

5. Обновите компоненты ОС.

Примечание. Процедура обновления зависит от вида ОС. При обновлении следуйте инструкциям производителя по обновлению ОС.

Пример команды для Debian/Ubuntu/Astra Linux:

```
apt-get update && apt-get upgrade
```

Пример команды для Red Hat/Centos/AlmaLinux/Rocky Linux:

```
yum update
```

6. После обновления ОС перезагрузите систему и убедитесь, что ОС загрузилась.

7. Проверьте, запустились ли все контейнеры, с помощью команды:

```
docker ps -a
```

Если контейнеры не запустились, для просмотра журнала событий выполните команду:

```
dpl --dc logs --tail=200 <не_запускаемый_контейнер>
```

Если процедура обновления сервера выполнена правильно, через пять минут будет доступен веб-интерфейс администратора https://ccs_host/admin.

Проверьте, отображаются ли пользователи в разделе «Пользователи» веб-интерфейса администратора.

8. Проверьте, функционируют ли в СК «Express»:

- сообщения в чатах;
- персональные звонки;
- групповые звонки.

РУЧНОЕ ОБНОВЛЕНИЕ СЕРВЕРА

SINGLE CTS

Внимание! Начиная с версии 3.26 изменяется архитектура приложения Express:

- для обновления приложения Express до версии 3.26 и выше переместите медиакомпоненты Express на отдельный Media-сервер в соответствии с инструкцией «[Руководство администратора. Серверы Media и Transcoding](#)»;
- если медиакомпоненты Express остаются развернутыми на сервере CTS Front (либо Single), обновите сервера согласно разделу «[Обновление Media на серверах Front или Single](#)».

Обновление сервера Media:

1. Перейдите в директорию Express `cd /opt/express-voice/`.
2. Примените новую конфигурацию:

```
dpl -p
```

3. Запустите обновление:

```
dpl -d
```

4. Проверьте логи на наличие ошибок командой:

```
dpl --dc logs --tail=200 -f
```

Для обновления сервера CTS:

1. Перейдите в директорию Express `cd /opt/express/`.
2. Остановите сервисы из директории Express:

```
DPL_PULL_POLICY=never dpl --dc stop
```

3. Выполните резервное копирование `/var/lib/docker/volumes` (или где они лежат в этой системе).

Если предыдущая версия nginx меньше, чем 1.20.1, и используются letsencrypt сертификаты, очистите хранилище letsencrypt (один раз):

```
rm -rf cts/letsencrypt
dpl cadvinstall && dpl nxinstall
```

4. Обновите node exporter и container advisor:

```
dpl cadvinstall && dpl nxinstall
```

5. Запустите обновление:

```
dpl -d
```

После запуска обновления требуется время на проведение внутренних процедур (примерно 10–15 минут).

6. Проверьте логи на наличие ошибок командой:

```
dpl --dc logs --tail=200 -f
```

Для отката обновления поправьте файл `settings`, указав параметр, например:

```
images:
  trusts: trusts:1.28.0
```

FRONT CTS И BACK CTS

Внимание! Начиная с версии 3.26 изменяется архитектура приложения Express:

- для обновления приложения Express до версии 3.26 и выше переместите медиакомпоненты Express на отдельный Media-сервер в соответствии с инструкцией «[Руководство администратора. Серверы Media и Transcoding](#)»;
- если медиакомпоненты Express остаются развернутыми на сервере CTS Front (либо Single), обновите сервера согласно разделу «[Обновление Media на серверах Front или Single](#)».

Перед началом процедуры обновления проверьте изменения по таблице сетевого взаимодействия. Первым обновляется сервер Front CTS, затем сервер Back CTS.

Обновление сервера Media:

1. Перейдите в директорию Express `cd /opt/express-voice/`.
2. Остановите сервисы из директории Express:

```
DPL_PULL_POLICY=never dpl --dc stop
```
3. Запустите обновление:

```
dpl -d
```
4. Проверьте логи на наличие ошибок командой:

```
dpl --dc logs --tail=200 -f
```

Для обновления сервера Front CTS:

1. Перейдите в директорию Express `cd /opt/express/`.
2. Остановите работу приложения командой:

```
DPL_PULL_POLICY=never dpl --dc stop
```
3. Выполните резервное копирование файлов `/var/lib/docker/volumes` (после нескольких дней эксплуатации резервные копии файлов можно удалить).
Если предыдущая версия nginx меньше, чем 1.20.1, и используются letsencrypt сертификаты, очистите хранилище letsencrypt (один раз):

```
rm -rf cts/letsencrypt
```
4. Запустите обновление:

```
dpl -d
```

После запуска обновления требуется время на проведение внутренних процедур (примерно 10–15 минут).
5. Проверьте логи на наличие ошибок командой:

```
dpl --dc logs --tail=200 -f
```

Для обновления сервера Back CTS:

1. Перейдите в директорию Express `cd /opt/express/`.
2. Остановите работу приложения командой:

```
DPL_PULL_POLICY=never dpl --dc stop
```
3. Выполните резервное копирование файлов `/var/lib/docker/volumes` (после нескольких дней эксплуатации скопированные файлы можно удалить).
4. Примените новую конфигурацию nginx:

```
dpl -p
```

5. Запустите служебные сервисы:

```
dp1 -d nginx postgres kafka etcd redis
```

6. Запустите обновление:

```
dp1 -d
```

После запуска обновления требуется время на проведение внутренних процедур (примерно 10–15 минут).

7. Проверьте логи на наличие ошибок командой:

```
dp1 --dc logs --tail=200 -f
```

ОБНОВЛЕНИЕ EXPRESS С ВЕРСИИ 3.25

Внимание! Начиная с версии 3.26 изменяется архитектура приложения Express:

- для обновления приложения Express до версии 3.26 и выше переместите медиакомпоненты Express на отдельный Media-сервер в соответствии с инструкцией [«Руководство администратора. Серверы Media и Transcoding»](#);
- если медиакомпоненты Express остаются развернутыми на сервере CTS Front (либо Single), обновите сервера согласно разделу [«Обновление Media на серверах Front или Single»](#).

ПОДГОТОВКА ПЕРЕД ОБНОВЛЕНИЕМ СЕРВЕРОВ:

Важно! Перед началом процедуры обновления компонентов до версии 3.39 необходимо проверить состояние обработки записей (в веб-интерфейсе администратора раздел «Звонки и конференции» → «Записи звонков»). При наличии сбойных (находящихся в статусе «in_porgress») записей, важно до начала процесса обновления перезапустить их обработку и дождаться ее завершения. В противном случае обработать зависшую запись невозможно, и она будет удалена.

Внимание! Перед выполнением обновления убедитесь, что на всех серверах Express, FQDN-media разрешается в IP-адрес из локальной сети Media-сервера.

1. Перейдите в веб-интерфейс администратора.
2. Перейдите в раздел «VoEX».
3. В поле «Janus WS URL» введите URL Media-сервера в формате `ws://fqdn-media:8188`.
4. В поле «Внешний хост Janus» введите IP-адрес, прописанный в параметре «janus_nat_1_1_mapping».

Примечание. В случае использования звонков между разными серверами в локальной сети в поле «Внешний хост Janus» необходимо указать FQDN, прописанный в настройках Split DNS (имя должно разрешаться во внешний IP-адрес в сети Интернет и во внутренний IP-адрес в локальной сети).

5. Нажмите кнопку .
6. Убедитесь, что новый сервер отобразился в списке на экране, а в шапке окна появилось сообщение «Janus URL добавлен».
7. Проверьте подключение сервера CTS к Media-серверу по параметру «Нагрузка» (рис. 1):

Janus load							
Janus WS URL	Нагрузка	publishers_n	audio_n	video_n	screen_n	recording_n	rooms_n
ws://janus.com:8188	590	121	78	7	20	0	34

рис. 1

- Если ранее был добавлен Janus сервер по адресу, во вкладке «Janus инстансы» удалите Janus-сервер нажав значок  (рис. 2).

Janus инстансы					
Janus WS URL	Внешний порт Janus	Кнопка	Janus WS URL	Внешний порт Janus	Кнопка
ws://janus.com:8188			ws://janus.com:8188		
Включено	Кнопка выключения	Нагрузка	Janus WS URL	Внешний порт Janus	Кнопка
<input checked="" type="checkbox"/>		14	ws://janus.com:8188		
<input checked="" type="checkbox"/>		23	ws://janus.com:8188		

рис. 2

- Перейдите в директорию Express `cd /opt/express`.
- Обновите сервисы Express:

```
DPL_PULL_POLICY=never dpl --dc stop && dpl -d
```

ОБНОВЛЕНИЕ MEDIA-СЕРВЕРА

Если запись не используется:

- Перейдите в директорию Express `cd /opt/express-voice`.
- Отредактируйте конфигурационный файл `settings.yaml`, дописав в конце файла:

```
transcoding_storage_enabled: false
```

- Обновите сервисы Express:

```
DPL_PULL_POLICY=never dpl --dc stop && dpl -d
```

Если запись используется:

- Перейдите в директорию Express `cd /opt/express-voice`.
- Добавьте сертификаты от FQDN-media (автоматический выпуск сертификатов от Let's Encrypt не поддерживается):

```
certs/express.crt
certs/express.key
```

- Отредактируйте конфигурационный файл `settings.yaml`, параметры для редактирования в примере выделены красным цветом. Перечень параметров перечислены в табл. 1:

```
transcoding_hosts:
  cts:
    ccs_host: fqdn-cts
    api_internal_token: *****
    storages_tokens_mapping:
      fqdn-media:
        token: *****
        ssl_envs_prefix: "TSS"
    api_internal_token: *****
    ccs_host: fqdn-media
    transcoding_storage_enabled: true

# не обязательные параметры
```

```
# если не публичные сертификаты, то отключаем проверку
сертификата:
tc-cts_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none
```

табл. 1

Название настройки	Значение
transcoding_hosts	Список объектов hosts (CTS), состоит из параметров: <ul style="list-style-type: none"> • ccs_host — FQDN CTS-сервера; • api_internal_token — токен для запросов к API (скопируйте из файлов /opt/express/settings.yaml, расположенных на соответствующих серверах ccs_hosts)
storages_tokens_mapping	Список объектов hosts (FQDN Media-сервера) состоит из параметров: <ul style="list-style-type: none"> • token — api_internal_token Media-сервера; • ssl_envs_prefix — префикс сертификата
api_internal_token	Токен для запросов к API, можно сгенерировать произвольные 16 символов (у CTS есть аналогичный параметр, значение которого может не совпадать с текущим)
transcoding_storage_enabled	Включение сервиса временного хранения записей (по умолчанию выключен)
ccs_host	FQDN-имя Media-сервера
tc-ct_env_override	Дополнительные параметры для transcoding
TSS_SSL_ENABLE	Включение/отключение дополнительных настроек transcoding
TSS_SSL_VERIFY	Проверка сертификата для transcoding

4. Запустите обновление командой:

```
dpl --dc down && docker volume prune -a && dpl -d
```

Важно! Команду обновления необходимо выполнять только на Media-сервере.

5. Проверьте доступность Media используя команду:

```
curl https://fqdn-media/testtest
```

6. Выполните команду для получения логов контейнера nginx на сервере Media:

```
docker logs voex-nginx-1 | grep testtest
```

В логе должны отображаться запрос (запросы) curl, например:

```
fqdn-media X.X.X.X - - [02/Oct/2024:08:50:34 +0000] "GET /testtest
HTTP/1.1" 204 0 "-" "curl/8.5.0" "X.X.X.X"
fqdn-media X.X.X.X - - [02/Oct/2024:08:50:42 +0000] "GET
/testtest HTTP/1.1" 204 0 "-" "curl/8.5.0" "X.X.X.X"
```

Если в логах запросов curl нет, то необходимо настроить split DNS на локальный IP-адрес Media-сервера.

ОБНОВЛЕНИЕ MEDIA НА СЕРВЕРАХ FRONT ИЛИ SINGLE

1. Перейдите в директорию Express `cd /opt/express-voice`.
2. Добавьте сертификаты от FQDN-media (автоматический выпуск сертификатов от Let's Encrypt не поддерживается):

```
transcoding_storage_enabled: true
traefik_bind_port_https: 8443
nginx_listen_http: false
ccs_host: fqdn_media
phoenix_secret_key_base: phoenix_secret_key
```

```

api_internal_token: api_internal_token_cts
coturn_options:
  cpuset: 10-11
  mem_limit: 2048m
janus_options:
  cpuset: 12-13
tc-cts_options:
  cpuset: 14-15

```

табл. 2

Название настройки	Значение
api_internal_token	Токен для запросов к API, можно сгенерировать произвольные 16 символов (у cts есть аналогичный параметр, значение которого может не совпадать с текущим)
transcoding_storage_enabled	Включение сервиса временного хранения записей (по умолчанию выключен)
ccs_host	FQDN имя Media-сервера
phoenix_secret_key	api token cts сервера указанный на back сервере в файле /opt/express/settings.yaml в параметре phoenix_secret_key_base
cpuset	Ограничение использование ядер CPU контейнером по номерам ядер, нумерация ядер начинается с 0
mem_limit	Ограничения использования памяти контейнером
nginx_listen_http	Прослушивание 80-го порта (по умолчанию true) отключаем, так как он занят
traefik_bind_port_https	Порт для прослушивания https, стандартный занят

3. Измените блок, отвечающий за настройку Transcoding, согласно примеру ниже:

```

transcoding_hosts:
  cts:
    ccs_host: fqdn-cts
    api_internal_token: *****
    storages_tokens_mapping:
      fqdn-media:
        token: *****
        ssl_envs_prefix: "TSS"
        override_endpoint: https://fqdn_media:8443
    # этот блок не обязательный (если janus_url по ip)
    ip-media:
      token: *****
      ssl_envs_prefix: "TSS"
      override_endpoint: https://fqdn_media:8443

tc-cts_env_override:
  TSS_SSL_ENABLED: true
  TSS_SSL_VERIFY: verify_none

```

табл. 3

Название настройки	Значение
transcoding_hosts	Список объектов hosts (CTS), состоит из параметров: <ul style="list-style-type: none"> • ccs_host — FQDN CTS-сервера; • api_internal_token — токен для запросов к API (скопируйте из файлов /opt/express/settings.yaml расположенных на соответствующих серверах ccs_hosts)
storages_tokens_mapping	Список объектов hosts (FQDN Media-сервера или IP) состоит из параметров: <ul style="list-style-type: none"> • token — api_internal_token Media-сервера, который делали в 3-м пункте; • ssl_envs_prefix — префикс сертификата; • override_endpoint — URL transcoding_storage
api_internal_token	Токен для запросов к API, можно сгенерировать произвольные 16 символов (у cts есть аналогичный параметр, значение которого может не совпадать с текущим)
transcoding_storage_enabled	Включение сервиса временного хранения записей (по умолчанию выключен)
ccs_host	FQDN имя Media-сервера
tc-ct_env_override	Дополнительные параметры для transcoding
TSS_SSL_ENABLE	Включение/отключение дополнительных настроек transcoding
TSS_SSL_VERIFY	Проверка сертификата для transcoding

ОБНОВЛЕНИЕ SINGLE, FRONT И BACK СЕРВЕРОВ

1. Перейдите в директорию Express `cd /opt/express/`.
2. Остановите сервисы из директории Express:

```
DPL_PULL_POLICY=never dpl --dc stop
```
3. Выполните резервное копирование /var/lib/docker/volumes (или укажите расположение директории для текущей системы).
4. Обновите node exporter и container advisor:

```
dpl cadvinstall && dpl nxinstall
```
5. Запустите обновление выполнив команду:
 - для Front и Single-сервера:

```
dpl -d
```
 - для Back-сервера:

```
dpl -d traefik nginx postgres redis kafka etcd && dpl -d
```

ОТКАЗОУСТОЙЧИВАЯ КОНФИГУРАЦИЯ

В случае невозможности использования скриптов автоматического обновления выполните обновление в ручном режиме.

Для обновления отказоустойчивой конфигурации:

1. Скопируйте новые версии образов Docker и скрипт загрузки образов в каталог /tmp/images и загрузите образы с помощью скрипта load.sh:

```
cp *.tar /tmp/images/
cp /opt/deploy/script/load.sh /tmp/images/
cd /tmp/images/
./load.sh
```
2. Подключитесь к консоли сервера Back и Front кластера с индексом 01 и 02.

3. Выполните команду для остановки антивируса:

```
systemctl stop kes1 klnagent64
```

Примечание. В случае зависания сервера при остановке антивируса перезагрузите оба узла кластера через систему виртуализации.

4. Выполните команду:

```
pcs status
```

5. Убедитесь, что ресурсы кластера запущены согласно списку ниже:

- dlm-clone [dlm] (back кластер) — запущен на обоих узлах кластера;
- clvmd-clone [clvmd] (back кластер) — запущен на обоих узлах кластера;
- clusterfs-clone [clusterfs] (back кластер) — запущен на обоих узлах кластера;
- cluster_ip — запущен на одном узле кластера;
- dockerd — запущен на одном узле кластера;
- node_exporter (back кластер) — запущен на одном узле кластера;
- cadvisor (back кластер) — запущен на одном узле кластера;
- vmfence (back кластер) — запущен на одном узле кластера.

Если статус ресурсов кластера не соответствует перечисленным выше, выполните команду, заменив `resource_name` на имя проблемного ресурса:

```
pcs resource cleanup resource_name
```

6. На узлах кластера Back с индексом 01 и 02 выполните команду ниже:

```
ls -la /opt/ex_data/files
```

Примечание. В случае зависание вывода списка директорий необходимо перезагрузить оба узла кластера через систему виртуализации.

7. Подключитесь к консоли сервера Back кластера с индексом 01 или 02 и выполните команду:

```
pcs status | grep dockerd
```

Примечание. Команда выполняется для определения текущего первичного узла, на котором запущены ресурсы кластера.

8. Подключитесь к консоли текущего первичного узла кластера Back и последовательно выполните команды:

```
cd /opt/express  
dpl -g
```

9. Подключитесь к консоли сервера Front кластера с индексом 01 или 02 и выполните команду:

```
pcs status | grep dockerd
```

Примечание. Команда выполняется для определения текущего первичного узла, на котором запущены ресурсы кластера.

10. Подключитесь к консоли текущего первичного узла кластера Front и последовательно выполните команды:

```
cd /opt/express  
dpl -g  
cd /opt/express-voice  
dpl -g
```

11. Подключитесь к консоли вторичного узла кластера Back и последовательно выполните команды, заменив `full_fqdn_slave_server` на полное доменное имя вторичного узла кластера:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

12. Дождитесь переключения ресурсов кластера `dockerd` и `cluster_ip` на вторичный узел кластера Back. Для мониторинга состояния ресурсов периодически выполняйте команду:

```
pcs status
```

13. После переключения ресурсов на вторичный узел кластера Back последовательно выполните команды:

```
cd /opt/express
dpl -g
```

14. Подключитесь к консоли вторичного узла кластера Front и последовательно выполните команды, заменив `full_fqdn_slave_server` на полное доменное имя вторичного узла кластера:

```
pcs resource move cluster_ip full_fqdn_slave_server
pcs resource move dockerd full_fqdn_slave_server
```

15. Дождитесь переключения ресурсов кластера `dockerd` и `cluster_ip` на вторичный узел кластера Front. Для мониторинга состояния ресурсов периодически выполняйте команду:

```
pcs status
```

16. После переключения ресурсов на вторичный узел кластера Front последовательно выполните команды:

```
cd /opt/express
dpl -g
cd /opt/express-voice
dpl -g
```

17. Подключитесь к консоли текущего первичного узла кластера Back и последовательно выполните команды:

```
cd /opt/express
dpl --dc stop
dpl nxinstall && dpl cadvinstall
dpl -d
```

18. После завершения обновления сервера откройте вывод логов работы контейнеров:

```
dpl --dc logs --tail=100 -f
```

19. Дождитесь остановки вывода логов контейнеров кроме контейнера `nginx`.

20. Подключитесь к консоли сервера Front кластера с индексом 01 или 02 и выполните команду:

```
pcs status | grep dockerd
```

Примечание. Команда выполняется для определения текущего первичного узла, на котором запущены ресурсы кластера.

21. Подключитесь к консоли текущего первичного узла кластера Front и последовательно выполните команды:

```
cd /opt/express
dpl --dc stop
dpl -d
cd /opt/express-voice
dpl --dc stop
dpl -d
```

22. После обновления первичных узлов кластеров Front и Back проверьте функционирование системы, выполните проверку логов на наличие ошибок командой:

```
dpl --dc logs --tail=200 -f
```

23. Подключитесь к консоли вторичного узла кластера Back и последовательно выполните команды, заменив full_fqdn_slave_server на полное доменное имя вторичного узла кластера:

```
pcs resource move cluster_ip full_fqdn_slave_server  
pcs resource move dockerd full_fqdn_slave_server
```

24. Дождитесь переключения ресурсов кластера dockerd и cluster_ip на вторичный узел кластера Back. Для мониторинга состояния ресурсов периодически выполняйте команду:

```
pcs status
```

25. После переключения ресурсов на вторичный узел кластера Back последовательно выполните команды:

```
cd /opt/express  
dpl --dc stop  
dpl nxinstall && dpl cadvinstall  
dpl -d
```

26. После завершения обновления сервера, откройте вывод лога работы контейнеров и дождитесь остановки вывода логов контейнеров кроме контейнера nginx:

```
dpl --dc logs --tail=100 -f
```

27. Подключитесь к консоли вторичного узла кластера Front и последовательно выполните команды, заменив full_fqdn_slave_server на полное доменное имя вторичного узла кластера:

```
pcs resource move cluster_ip full_fqdn_slave_server  
pcs resource move dockerd full_fqdn_slave_server
```

28. Дождитесь переключения ресурсов кластера dockerd и cluster_ip на вторичный узел кластера Front. Для мониторинга состояния ресурсов периодически выполняйте команду:

```
pcs status
```

29. После переключения ресурсов на вторичный узел кластера Front последовательно выполните команды:

```
cd /opt/express  
dpl --dc stop  
dpl -d  
cd /opt/express-voice  
dpl --dc stop  
dpl -d
```

30. Подключитесь к консоли сервера Back и Front кластера с индексом 01 и 02, выполните команду для запуска антивируса:

```
systemctl start kes1 klnagent64
```

ОБНОВЛЕНИЕ POSTGRESQL

Важно! Существует риск потери данных в момент обновления БД. Процедуру обновления рекомендуется выполнять во время минимальной пользовательской активности или остановки всех сервисов. Перед процедурой строго необходимо выполнить полное резервное копирование базы данных.

ПРОЦЕДУРА ОБНОВЛЕНИЯ

Перед обновлением встроенной БД:

1. Убедитесь в наличии свободного места на диске.

Для обновления требуется дисковое пространство, равное по размеру существующей базе. Текущий размер базы можно узнать с помощью команды:

```
docker system df -v | grep postgres_data
```

2. Уточните текущую версию PostgreSQL.

Если после обновления возникнут ошибки, может потребоваться откат версии PostgreSQL. Поэтому желательно знать, на какой версии БД работала до обновления. Важно выполнить команду вывода текущей версии БД до обновления образа PostgreSQL:

```
DPL_PULL_POLICY=never dpl --dc exec postgres cat /var/lib/postgresql/data/PG_VERSION
```

3. Убедитесь, что имеется доступ к apt.postgresql.org. Доступ требуется для загрузки исполняемых файлов старой версии. При отсутствии доступа см. раздел «[Обновление без доступа к apt.postgresql.org](#)».

Чтобы откатить версию базы, нужно перенести файлы из директории с резервной копией в директорию уровнем выше.

4. В конец файла `settings` добавьте настройку, запускающую обновление контейнера БД при старте:

```
postgres_upgrade: true
```

Примечание. Если данной настройки не будет, новые версии образа PostgreSQL при запуске будут выдавать ошибку «`postgres_upgrade disabled, exiting`».

5. Выполните обновление образа PostgreSQL и его запуск с помощью команды:

```
dpl -d postgres
```

6. Для отслеживания процесса обновления в логах используйте команду:

```
dpl --dc logs -f --tail=1 postgres
```

Когда обновление будет закончено в логах появится сообщение:

```
DB upgrade is done, please disable postgres_upgrade in the settings
```

7. Отключите настройку `postgres_upgrade`, иначе контейнер с PostgreSQL не запустится.

8. Уберите настройку `postgres_upgrade` и загрузите стандартный образ с помощью команды:

```
DPL_PULL_POLICY=never dpl -d postgres
```

Для автоматического обновления воспользуйтесь настройкой `postgres_auto_upgrade`, с ней база будет автоматически обновляться при выпуске образа с новой версией.

Если версию БД по каким-то причинам нужно оставить без изменений, остаются доступны образы postgres без скрипта обновления. Их можно включить через параметр images, например:

```
images:
  postgres: postgres:9.5.24
```

РЕЗЕРВНАЯ КОПИЯ

После обновления базы старая версия сохраняется для возможности отката версии. Копия расположена в том же volume, который использует контейнер postgres, в директории с именем upgrade_backup_<timestamp>.

Если по итогам обновления все сервисы работают нормально, и откат не требуется, удалите резервную копию командой:

```
dpl --dc exec postgres find /var/lib/postgresql/data -type d -name
upgrade_backup_* -exec rm -r {} \;
```

ОТКАТ ВЕРСИИ

Для того чтобы откатить версию базы, нужно перенести файлы из директории с резервной копией в директорию уровнем выше.

Для работы с файловой системой volume запустите временный контейнер:

```
DPL_PULL_POLICY=never dpl --dc run --rm --entrypoint=/bin/bash
postgres
```

Или выполните операцию непосредственно с хоста. Docker volumes обычно хранятся в /var/lib/docker/volumes.

После переноса файлов выставите образ postgres предыдущей версии в файле в settings, например:

```
images:
  postgres: postgres:9.5.24
```

ОБНОВЛЕНИЕ БЕЗ ДОСТУПА К ART.POSTGRESQL.ORG

Для обновления без доступа к art.postgresql.org собран специальный образ 14.4-from-9.5.

Примечание. Поддерживается только обновление с версии 9.5.

Для обновления БД до версии 14.4:

1. В файл settings добавьте секцию images с соответствующим тэгом:

```
images:
  postgres: postgres:14.4-from-9.5
```

2. В конец файла settings добавьте настройку, запускающую обновление контейнера БД при старте:

```
postgres_upgrade: true
```

Примечание. Если данной настройки не будет, новые версии образа PostgreSQL при запуске будут выдавать ошибку «postgres_upgrade disabled, exiting».

3. Выполните обновление образа PostgreSQL и его запуск с помощью команды:

```
dpl -d postgres
```

4. Чтобы отследить процесс обновления в логах, используйте команду:

```
dpl --dc logs -f --tail=1 postgres
```

Когда обновление будет закончено, в логах появится сообщение:

```
DB upgrade is done, please disable postgres_upgrade in the settings
```

5. Уберите добавленную секцию `images` и настройку `postgres_upgrade` и загрузите стандартный образ с помощью команды:

```
DPL_PULL_POLICY=never dpl -d postgres
```

АВАРИЙНЫЕ СИТУАЦИИ ПРИ ОБНОВЛЕНИИ ИЗ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ REGISTRY

Аварийные ситуации, перечисленные ниже, могут произойти в том случае, если имеется локально развернутый сервер Registry.

Ситуация 1. Отсутствия доступа к сети интернет с узла с репозиторием:

1. С узла, имеющего доступ в интернет, скачайте актуальные контейнеры с помощью скрипта [по ссылке](#) (вложение download.sh).
2. Запустите второй скрипт [по ссылке](#) (вложение upload.sh) и дождитесь окончания загрузки.
3. Сделайте тестовый запрос из консоли при помощи обращения к URL и получите версии, находящиеся в репозитории.

(Пример:

```
curl -u userregistry  
http://cts.server.single.local/v2/ad_integration/tags/list  
{ "name": "ad_integration", "tags": ["1.42.0", "1.38.1"] }
```

Команда

Результат
команды

Ситуация 2. Если в п.3 предыдущей операции результат команды – no basic auth credentials:

1. Удалите файл .docker/config.json.
2. Пройдите повторную авторизацию в Docker registry.

ПРОЦЕДУРА ОБНОВЛЕНИЯ СЕРТИФИКАТА

Для работы изделия требуется оформить сертификат на внешнее имя сервиса Express (FQDN или wildcard), выпущенный публичным доверенным центром сертификации и удовлетворяющий следующим требованиям:

- версия стандарта сертификата 3 и не ниже TLS 1.2;
- длина ключа не меньше 2048 бит;
- алгоритм подписи SHA 256;
- версия синтаксиса X.509 3;
- незашифрованный закрытый ключ.

Файл должен содержать в себе сертификат сервера, сертификаты промежуточного центра сертификации и корневого центра сертификации. Формат сертификатов должен соответствовать кодировке Base64. Файл закрытого ключа должен содержать нешифрованный закрытый ключ кодировки Base64.

Примерная структура файла сертификата изображена на рисунке ниже (рис. 3):

```
-----BEGIN CERTIFICATE-----
Base64 server certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64 intermediate ca
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64 root ca
-----END CERTIFICATE-----
```

рис. 3

Поддерживается использование бесплатного сертификата от компании Let`s Encrypt.

Внимание! Имена файлов сертификата и ключа должны быть express.crt, express.key.

Для обновления сертификата на серверах Single, Front, Back (RTS, ETS, CTS):

1. Подготовьте сертификат согласно требованиям выше.
2. Обновите файлы сертификатов, расположенные в папке /opt/express/certs/.
3. Выполните команду:

```
cd /opt/express && dpl -p && dpl --dc restart traefik
```

Для обновления сертификата на Media-сервере:

1. Подготовьте сертификат согласно требованиям выше.
2. Обновите файлы сертификатов, расположенные в папке /opt/express-voice/certs/.
3. Выполните команду в консоли:

```
cd /opt/express-voice && dpl -p && dpl --dc restart coturn traefik
```

ОБНОВЛЕНИЕ НАСТРОЕК СЕРВЕРА ЗВОНКОВ VOEX

Для обновления сервера звонков Single CTS на версию 3.16 и выше:

1. Подключитесь к серверу Single CTS через SSH.
2. Выполните команду:


```
cd /opt/express-voice/
```
3. Отредактируйте конфигурационный файл settings.yaml. В параметре janus_ws_acl допишите IP-адрес сетевой карты:


```
janus_ws_acl: 172.18.0.,X.X.X.X,
```

 где X.X.X.X — IP-адрес сетевой карты для примера.
4. Выполните команду:


```
dp1 -d janus
```
5. Перейдите в веб-интерфейс администратора https://ccs_host/admin.
6. Перейдите в раздел «VoEX» и выполните действия, в зависимости от версии сервера:
 - **Для версии ниже 3.31:**
 7. Введите в поле «Janus WS URL» значение адреса Janus-сервера в соответствии с файлом настроек /opt/express/settings.yaml (рис. 4).



рис. 4

- **Для версии 3.31 и выше:**
 7. Введите в поле «Внешний хост Janus» значение IP-адреса внешнего Media-сервера (если поле оставить пустым, межсерверные звонки работать не будут) (рис. 5):

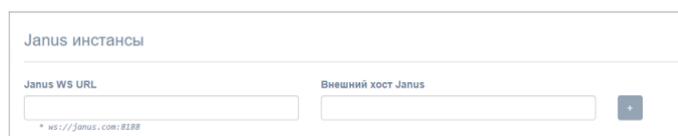


рис. 5

8. Нажмите кнопку .
9. Сервер отобразится в списке, в шапке окна появится сообщение «Janus URL добавлен».

Для обновления сервера звонков Front CTS на версию 3.16 и выше:

1. Перейдите в консоль администратора (веб-интерфейс) https://ccs_host/admin.
2. Перейдите в раздел «VoEX» и выполните действия, в зависимости от версии сервера:
 - **Для версии ниже 3.31:**
 3. Введите в поле «Janus WS URL» значение адреса Janus-сервера (адрес Back CTS) в соответствии с файлом настроек settings.yaml (рис. 6).



рис. 6

- **Для версии 3.31 и выше:**

3. Начиная с версии 3.31 и выше введите в поле «Внешний хост Janus» значение IP-адреса внешнего Media-сервера (если поле оставить пустым, межсерверные звонки работать не будут) (рис. 7).

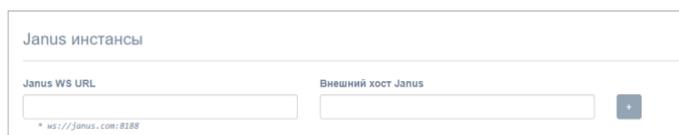


рис. 7

4. Нажмите кнопку .
5. Сервер отобразится в списке, в шапке окна появится сообщение «Janus URL добавлен».

Приложение 1

МИГРАЦИЯ MESSAGING ВЕРСИИ 3.10 И НИЖЕ НА 3.10 И ВЫШЕ

Процедура миграции обязательна для обновления с версии ниже 3.10. После миграции на 3.10+ откатывать messaging на предыдущие версии не рекомендуется! Возвращение на прошлые версии приведет к сложной процедуре ручного удаления новых данных и повторения миграции. В случае возникновения проблем при миграции необходимо обратиться в службу технической поддержки для получения патча.

Примечание. Время простоя в момент миграции может достигать от 5 до 30 минут в зависимости от объема базы.

ОПРЕДЕЛЕНИЕ РАЗМЕРА ТАБЛИЦЫ

Для определения размера таблиц:

1. Войдите в консоль СУБД PostgreSQL и подключитесь к базе данных messaging_prod:

- если база данных реализована в контейнере Docker выполните команды:

```
docker exec -ti cts_postgres_1 psql -U postgres  
\c messaging_prod
```

- если база данных внешняя (без кластера), выполните команды:

```
su postgres  
psql  
\c messaging_prod
```

- если база данных внешняя (кластер на базе Patroni), выполните команды:

```
psql -h внутренний_ип_ноды -p 5432 -U express -d postgres  
\c messaging_prod
```

2. Выполните запрос размера таблиц базы данных:

```
select schemaname as table_schema,  
       relname as table_name,  
       pg_size_pretty(pg_total_relation_size(relid)) as total_size,  
       pg_size_pretty(pg_relation_size(relid)) as data_size,  
       pg_size_pretty(pg_total_relation_size(relid) -  
pg_relation_size(relid))  
       as external_size  
from pg_catalog.pg_statio_user_tables  
where relname = 'user_chat_events' or relname = 'chat_events'  
order by pg_total_relation_size(relid) desc,  
       pg_relation_size(relid) desc;
```

Если total_size user_chat_events и/или chat_events превышает 20 ГБ, то время простоя при миграции в автоматическом режиме может составлять > 1 часа. В этом случае рекомендуется провести миграцию по специальной инструкции (см. «Миграция больших баз данных»). Для удобства рекомендуем заранее подготовить файл со всеми таймстемпами и запросами.

Примечание. Если таблицы в вашей базе данных меньше 20 ГБ, вы можете мигрировать данные в автоматическом режиме (дополнительных действий не требуется).

МИГРАЦИЯ БОЛЬШИХ БАЗ ДАННЫХ

Некоторые запросы, выполняющиеся при подготовке к миграции, могут обрабатываться до десяти часов в зависимости от объемов базы данных. Например, миграция старой истории сообщений. В этом случае рекомендуется использовать специальные утилиты для выполнения долгих задач `screen` или `nohup`.

Пример использования:

1. Добавьте запрос в текстовый файл с расширением `.sql`, например `update.sql`.
2. Добавьте права на чтение файла пользователю, от имени которого запускается `PSQL`.

Для утилиты `screen` (рекомендуется):

1. В терминале на сервере с базой данных выполните команду `screen`.
Откроется сессия терминала.
2. Выполните команду (замените значение флага `-u` и `-f` на свой):

```
sudo -u postgres psql -d messaging_prod -f /home/postgres/update.sql
```
3. Выйдите из сессии, нажав сочетания клавиш `<Ctrl>+<A>` и затем `<Ctrl>+<D>`.

Внимание! Не нажимайте `<Ctrl>+<C>` и `<Ctrl>+<D>`, пока находитесь в `screen`-сессии и запрос не завершится.

4. Чтобы вернуться к `screen`-сессии и убедиться, что сессия работает и запрос выполняется, выполните команду:

```
screen -r
```
5. Повторите процедуру из п.4, чтобы проверить состояние задачи.
Чтобы выйти из сессии, повторите процедуру из п.3.

Для утилиты `nohup`:

1. В терминале на сервере с базой данных выполните следующую команду (команда не должна требовать никакого `input`, например пароля от `root` юзера):

```
nohup sudo -u postgres psql -d messaging_prod -f /home/postgres/update.sql &
```
2. Проверьте статус задачи с помощью команды:

```
jobs -l
```
3. Проверьте вывод в файле `nohup.out`.
4. Проверьте статус выполнения запроса (после минуты выполнения):

```
SELECT
  pid,
  user,
  pg_stat_activity.query_start,
  now() - pg_stat_activity.query_start AS query_time,
  query,
  state,
  wait_event_type,
  wait_event
FROM pg_stat_activity
WHERE (now() - pg_stat_activity.query_start) > interval '1
minutes'
```

```
AND datname = 'messaging_prod' AND state = 'active';
```

Результатом станет отображение запроса. Screen-сессия или похуп процесс должны быть активными (рис. 8).

```
pid | user | query_start | query_time | query | state | wait_event_type | wait_event
-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)
```

рис. 8

ПОДГОТОВКА К МИГРАЦИИ

Перед началом операции убедитесь в достаточном количестве свободного места. Рекомендуется, чтобы объем свободного пространства в 2–3 раза превышал объем базы данных.

Примечание. Данные запросы выполняются при запущенных сервисах. Если база данных очень большая, подготовку можно начинать за несколько дней до миграции и выполнять этапы в технические окна.

Все запросы выполняются в PSQL, база данных messaging_prod.

1. Добавьте новые колонки в таблицу chat_events:

```
ALTER TABLE chat_events
ADD COLUMN events_history_scope BOOLEAN,
ADD COLUMN event_info_scope BOOLEAN;
```

2. Запишите текущий timestamp. Этот timestamp необходимо сохранить до момента будущей миграции:

```
messaging_test=# SELECT now();
                now
-----
2023-12-25 11:06:06.037283+03
(1 row)
```

3. Запишите timestamp самого раннего сообщения:

```
messaging_test=# SELECT inserted_at FROM chat_events ORDER BY
inserted_at ASC LIMIT 1;
                inserted_at
-----
2018-02-05 10:21:44.191786
(1 row)
```

4. Выполните миграцию историй сообщений от даты, полученной из таймстемпа в п.2 + 1 минута до даты самого раннего события п.3 с шагом в 1 год. Каждый запрос будет выполняться длительное время. Если по каким-то причинам запрос выполнить не получается, можно уменьшить шаг до полугода/трех месяцев.

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-01-01' AND ce.inserted_at <= '2023-12-25 11:07';
```

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
```

```
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2022-01-01' AND ce.inserted_at <= '2023-01-01';
```

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2021-01-01' AND ce.inserted_at <= '2022-01-01';
```

...

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2018-02-05' AND ce.inserted_at <= '2019-01-01';
```

5. Если total_size ваших таблиц больше 200 ГБ, то быстрее будет выполнить один долгий общий запрос без временных шагов:

```
--- запрос для миграции без временного шага,
--- выполнять только в случае очень объемных баз,
--- обязательно с использованием nohur или screen,
--- время выполнения может достигать > 10 часов
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed';
```

6. Создайте новые индексы для таблицы chat_events:

```
CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_stealth_index ON chat_events
USING btree (group_chat_id, inserted_at) INCLUDE (sync_id,
stealth) WHERE (NOT (stealth IS NULL));
```

```
CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_shared_index ON chat_events
USING btree (group_chat_id, inserted_at) WHERE
((events_history_scope = true) AND (shared = true));
```

```
CREATE INDEX CONCURRENTLY
chat_events_group_chat_id_inserted_at_non_shared_index ON
chat_events USING btree (group_chat_id, inserted_at) WHERE
((events_history_scope = true) AND (shared = false));
```

```
CREATE INDEX CONCURRENTLY
chat_events_sync_id_event_info_scope_index ON chat_events USING
btree (sync_id) WHERE (event_info_scope = true);
```

```
CREATE INDEX CONCURRENTLY chat_events_epoch_migration_index_1 ON
chat_events (event_type) WHERE event_type='added_to_chat' OR
event_type='user_joined_to_chat';
```

```
CREATE INDEX CONCURRENTLY chat_events_epoch_migration_index_2 ON
chat_events (event_type) WHERE event_type='left_from_chat' OR
event_type='deleted_from_chat' OR
event_type='kicked_by_cts_logout';
```

7. Отслеживание прогресса создания индексов

```
--- Отслеживание прогресса создания индексов
SELECT phase, blocks_total, blocks_done, (blocks_total -
blocks_done) as blocks_rest, tuples_total, tuples_done,
(tuples_total - tuples_done) as tuples_rest FROM
pg_stat_progress_create_index;
```

МИГРАЦИЯ

Внимание! Если этап подготовки к миграции был выполнен несколько дней назад, то необходимо мигрировать историю от текущего времени до времени начала этапа подготовки к миграции. Если этап подготовки к миграции и основная миграция выполняются в один день, то следует пропустить п.1 и п.2.

1. Запишите текущий timestamp:

```
messaging_test=# SELECT now();
              now
-----
2023-12-27 12:14:01.026154+03
(1 row)
```

2. Обновите историю от времени, полученного в разделе «Подготовка к миграции» п.2

```
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-25 11:06:00';
```

3. Остановите все сервисы, кроме postgres:

- если база данных реализована в контейнере Docker, выполните команду:

```
docker stop $(docker ps --format "{{.Names}}") |grep -v
postgres)
```

- если база данных внешняя, выполните команду:

```
dpl --dc stop
```

4. Подключитесь к базе данных messaging_prod с помощью psql.

5. С помощью команды пропустите миграции, которые выполняли (текст команды вводится как указано, без изменений):

```
INSERT INTO
schema_migrations (version, inserted_at)
VALUES
(20231031201400, now()),
(20231031201640, now()),
(20231101121946, now()),
(20231101134329, now()),
(20231116000454, now()),
(20231129210158, now()),
(20231205114132, now()),
(20231205115340, now()),
(20231205123846, now()),
```

```
(20231211072757, now());
```

6. Запишите текущий timestamp:

```
messaging_test=# SELECT now();
               now
-----
2023-12-27 12:34:39.258821+03
(1 row)
```

7. Мигрируйте историю от момента подготовки миграции. Используйте дату и время с точностью до минуты:

- из п.1, если был выполнен процесс дополнительной миграции истории:

```
---- обновление с временем из пункта 1
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-27 12:14:00';
```

- из п.2 раздела «Подготовка к миграции», если дополнительной миграции истории не было:

```
---- обновление с датой из раздела "Подготовка к миграции"
пункт 2
UPDATE chat_events ce
SET events_history_scope = uce.events_history_scope,
event_info_scope = uce.event_info_scope
FROM user_chat_events uce
WHERE ce.sync_id = uce.event_sync_id
AND ce.event_type != 'routing_changed'
AND ce.inserted_at >= '2023-12-25 11:06:00';
```

8. Запустите сервисы, дождитесь завершения автоматических миграций и запуска сервиса messaging.

9. Мигрируйте следующие данные используя таймстемп, полученный в п.6:

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, ce.inserted_at
FROM chat_events ce
INNER JOIN group_chats gc ON gc.id = ce.group_chat_id
WHERE global = true AND
shared = false AND
chat_type = 'global' AND
ce.inserted_at <= '2023-12-27 12:34:39';

INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, inserted_at
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) > 0
AND
event_type = ANY({'message_new',call_end'}) AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, inserted_at
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) = 1
AND
event_type = 'app_event' AND
payload->>'event_type' = 'bot_notification' AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

10. Выполните дамп системных событий из каналов. Его можно будет удалить спустя время (когда релиз проверится временем):

```
CREATE TABLE channel_events_dump AS
SELECT ce.* FROM chat_events ce INNER JOIN group_chats c ON
c.id = ce.group_chat_id WHERE chat_type = 'channel'
AND event_type =
ANY('{added_to_chat,deleted_from_chat,left_from_chat,admin_added_t
o_chat,kicked_by_cts_logout,user_joined_to_chat}');
```

11. Удалите лишние системные события из каналов:

```
DELETE FROM chat_events ce
USING group_chats c
WHERE c.id = ce.group_chat_id
AND chat_type = 'channel'
AND event_type =
ANY('{added_to_chat,deleted_from_chat,left_from_chat,admin_added_t
o_chat,kicked_by_cts_logout,user_joined_to_chat}');
```

12. Если на сервере использовался notifications_bot (не внутренний бот), потребуется мигрировать его сообщения. Для этого выполните следующий запрос и подставьте в качестве bot_id идентификатор notifications_bot, используя таймстемп, полученный в п.6:

```
INSERT INTO user_chat_events_v2 (user_huid, group_chat_id,
event_sync_id, inserted_at)
SELECT jsonb_array_elements_text((event_params-
>>'recipients')::jsonb)::uuid as user_huid, group_chat_id,
sync_id, now()
FROM chat_events ce
WHERE jsonb_array_length((event_params->>'recipients')::jsonb) > 0
AND
event_type = 'app_event' AND
payload->>'event_type' = 'bot_notification' AND
sender = 'bot_id' AND
ce.inserted_at <= '2023-12-27 12:34:39';
```

13. Обновите статистику (см. раздел «[Обновление статистики](#)»).

14. Удалите миграционные индексы:

```
DROP INDEX chat_events_epoch_migration_index_1;
DROP INDEX chat_events_epoch_migration_index_2;
```

ОБНОВЛЕНИЕ СТАТИСТИКИ

После автоматической или ручной миграции необходимо обновить статистику. В PSQL выполните следующие запросы:

```
VACUUM ANALYZE user_chat_events_v2;  
VACUUM ANALYZE chat_event_meta;  
VACUUM ANALYZE chat_member_epochs;  
VACUUM ANALYZE chat_events;
```

ИСТОРИЯ ИЗМЕНЕНИЙ

Раздел «История изменений» содержит список изменений в документе, связанных с изменениями/доработками СК «Express».

Сборка 2.8.0

№	Раздел	Изменение	Ссылка
1.	Обновление доступа без к apt.postgresql.org	Актуализирован порядок обновления	стр. 17

Сборка 3.0

№	Раздел	Изменение	Ссылка
1.	Обновление ОС	Добавлен раздел	стр. 5
2.	Обновление Deployka	По всему документу исправлена операция DEPLOYKA_SKIP_UPDATE=true на DPL_PULL_POLICY=never	
3.	Процедура обновления сертификата	Изменение директории хранения сертификата /opt/express/certs	стр. 20
4.	Ручное обновление сервера	Замена En-Dash на Hyphen-minus в примерах с командной строкой	

Сборка 3.5

№	Раздел	Изменение	Ссылка
1.	Процедура обновления сертификата	Изменение команды при обновлении сертификата для сервера версии 3.5 и выше	стр. 20

Сборка 3.10

№	Раздел	Изменение	Ссылка
1.	Миграция messaging на 3.10+	Добавлен раздел	стр. 23

Сборка 3.14

№	Раздел	Изменение	Ссылка
1.	Процедура обновления сертификата	Добавлены пункты для обновления сертификата VoEX	стр. 20

Сборка 3.16

№	Раздел	Изменение	Ссылка
1.	Обновление настроек сервера звонков VOEX	Добавлен раздел	стр. 21

Сборка 3.26

№	Раздел	Изменение	Ссылка
1.	Обновление Express до версии 3.26 и выше	Добавлен раздел	стр. 8

Сборка 3.28

№	Раздел	Изменение	Ссылка
1.	Обновление Media-сервера	Добавлен параметр transcoding_storage_enabled	стр. 9

Сборка 3.29

№	Раздел	Изменение	Ссылка
1.	Обновление Media на серверах Front или Single	Добавлен раздел	стр. 10
2.	Обновление Single, Front и Back серверов	Добавлен раздел	стр. 12

Сборка 3.42

№	Раздел	Изменение	Ссылка
1.	Обновление Express с версии 3.25	Обновлен рисунок	стр. 8