VK WorkSpace

Корпоративный мессенджер VK Teams

Руководство по администрированию (версия 24.3)

Оглавление

Назначение документа

В данной инструкции описана авторизация в VK Teams, основные настройки клиентских приложений, а также операции для управления пользователями при отсутствии контроллера домена.

Документ предназначен для использования администраторами организации.

> **Примечание**
>
> Ранее VK Teams назывался Myteam, что находит отражение в технических моментах (например, команды в консоли).

Дополнительная документация

Инструкция по интеграции с контроллером домена по протоколу LDAP — в документе описано управление параметрами синхронизации LDAP.

Инструкция по настройке интеграции с антивирусом — в документе описана архитектура и способы антивирусной проверки, а также ее конфигурирование через протокол ICAP и через интеграцию с KATA.

Инструкция по настройке интеграции с DLP-системой — в документе описан механизм отправки запросов в DLP-систему, а также процесс активации отправки данных в DLP-систему.

Инструкция по подключению внешнего S3-хранилища — в документе описан процесс подключения внешнего S3-хранилища.

Инструкция по настройке интеграции с SIEM-системой — в документе описаны логируемые события и формат log-файлов, а также настройка отправки log-файлов в SIEM-систему.

Инструкция по установке на одну виртуальную машину /Инструкция по установке кластера — в документах описана настройка доступа в окружение администратора в процессе установки VK Teams.

Инструкция по настройке Single Sign-On аутентификации — в документе описана настройка Single Sign-On аутентификации по протоколам OIDC, SAML и Kerberos.

Архитектура и описание системы VK Teams — в документе описаны механизмы аутентификации в VK Teams. Не является частью публичной документации, обратитесь к представителю VK Tech, чтобы ознакомиться с документом.

Компоненты системы

Сервис авторизации

Синхронизация пользователей из LDAP-клиента, отправка и проверка одноразовых паролей (OTP). Проверка авторизации, анти-брутфорс меры (защита от перебора авторизационных данных).

Ядро системы

Передача и хранение сообщений, контакт-листов, чатов и т. д. Взаимодействие пользователя и ядра системы происходит через веб-API.

Push-сервисы

Отправка push-сообщений на платформы iOS (Apple) и Android (Google).

Веб

HTTPS-интерфейс для пользователя. HTTP API для передачи и приема сообщений, поиска, передачи профилей, контакт-листов, передачи и обработки файлов, передачи стикеров.

Голосовые шлюзы

Возможность голосовых, видеозвонков и конференций в случае, когда участники не могут соединиться напрямую.

Хранилище S3

Хранение файлов пользователя, пользовательских стикеров. Возможно использование внешнего хранилища S3.

Сервисы мониторинга

Проверка работоспособности системы. Встроенные сервисы требуют интеграции в ядро системы мониторинга обслуживающей организации и не имеют веб-интерфейса или каналов оповещения.

Базы данных

Хранение данных для сервисов. На данный момент используются БД с открытым кодом MySQL, Tarantool, а также встраиваемая БД KUST (внутренняя разработка компании с закрытым исходным кодом).

Настройки пользователя затрагивают, в основном, следующие компоненты: веб, голосовые сервисы, авторизация. Компоненты, активно взаимодействующие с пользователями: авторизация, веб, push-сервисы, голосовые сервисы. Далее рассматриваются компоненты, наиболее подверженные внешним воздействиям (например, вмешательство в настройки, DDoS атаки и т. п.).

Авторизация в системе

Варианты доступа к системе (используются совместно):

- SSO-аутентификация.

  Описание процесса настройки и диаграммы запросов представлены документе Инструкция по настройке Single Sign-on аутентификации.

- Аутентификация с помощью одноразовых кодов (OTP).

  При помощи LDAP-коннектора осуществляется подключение к службе каталогов. Пользователи загружаются в сервис Keycloak и далее в БД Tarantool. Пароли для доступа в систему в сервисе не хранятся, при каждой авторизации пользователю на его почтовый ящик отправляется одноразовый пароль. Логином выступает электронная почта пользователя.

  Описание механизма аутентификации представлено в документе «Архитектура и описание системы VK Teams» (не является частью публичной документации, обратитесь к представителю VK Tech, чтобы ознакомиться с документом).

При эксплуатации системы сложности чаще всего возникают именно с доступами, так как Active Directory является сложной системой, требующей аккуратного отношения.

Компоненты системы авторизации

- Сервис Keycloak (https://www.keycloak.org/). Это приложение осуществляет синхронизацию пользователей через LDAP. Все данные хранятся в БД MySQL.
- Сервис Nomail — основной сервис авторизации внутри системы. Может работать как самостоятельно, так и совместно с Keycloak. Все данные сервиса Nomail хранятся в БД tarantool.
- MTA Postfix — принимает OTP-сообщения от сервиса Nomail и перенаправляет эти сообщения на SMTP-релей, указанный пользователем.

Сервис Keycloak (синхронизация пользователей)

Сервис синхронизирует пользователей через LDAP. Расположение журнала: **/var/log/vector/k8s/keycloak/keycloak/* .** При любых ошибках синхронизации имеет смысл в первую очередь смотреть в журнал этого сервиса.

Доступ к веб-интерфейсу

У сервиса есть веб-интерфейс для управления, однако пользоваться этим интерфейсом следует осторожно, так как изменения, сделанные через этот интерфейс, могут быть изменены при обновлении

системы. Любые изменения обязательно нужно внести в конфигурационный файл **/usr/local/etc/premsetup/defaults.yaml**.

По умолчанию интерфейс недоступен из внешней сети. Варианты доступа к веб-интерфейсу:

Вариант 1: Доступ через https://mridme.<YOUR_DOMAIN>/auth

Открыть доступ для домена mridme. <DOMAIN> и перейти в браузере https://mridme.<DOMAIN>

> **Примечание**
>
> По умолчанию имя mridme не заведено в DNS, и в настройках nginx выставлено deny all. Не рекомендуется использовать этот способ доступа без крайней необходимости.

Вариант 2: SSH-туннель (предпочтительный)

Выполнить команду:

```
ssh -L 8080:keycloak-http.keycloak.svc.cluster.local:80 centos@<server>
```

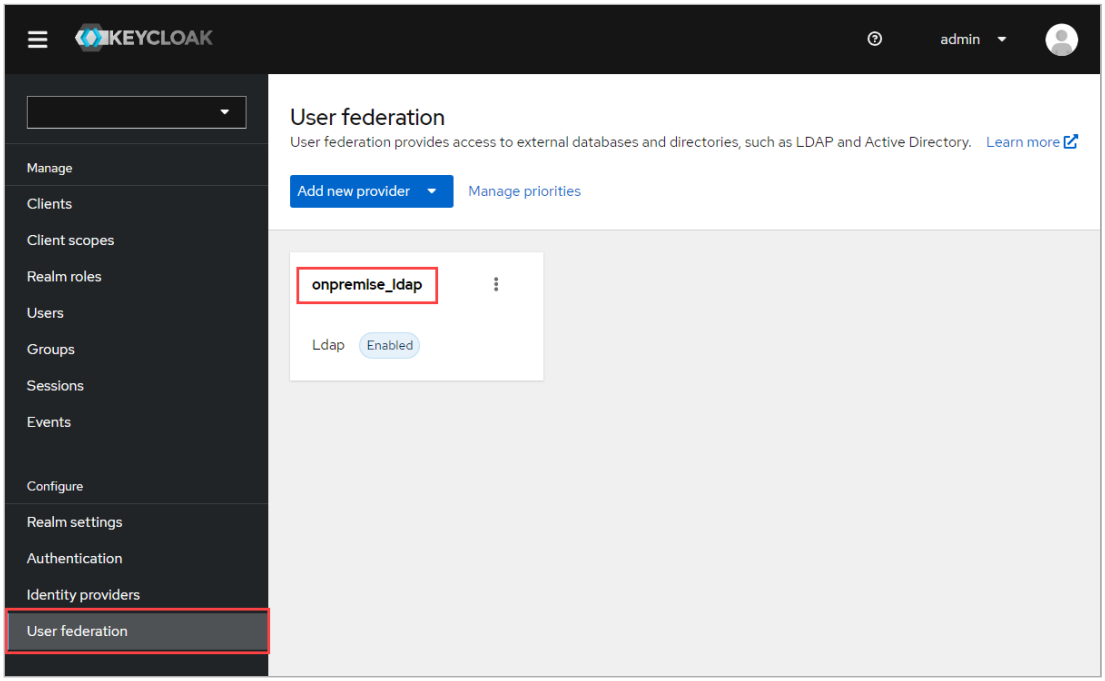
и перейти в браузере http://127.0.0.1:8080/auth

Логин: admin

Пароль: пароль необходимо получить в службе технической поддержки.

Наиболее важные параметры, доступные в веб-интерфейсе:

Настройка LDAP-подключения



Если LDAP-подключение уже создано, но с ним есть проблемы, то основные настройки, с которыми нужно работать:

- baseDN (usersDN) и фильтр (влияют на то, какие объекты будут получены через LDAP);
- частота синхронизации (влияет на то, как часто будут производиться изменения).

> **Внимание**
>
> При большом количестве пользователей, попадающих под полную фильтрацию, полная синхронизация может быть очень тяжелой для Active Directory. Выполнять такую операцию часто нельзя. Например, Active Directory на 10 тыс. пользователей отрабатывается около трех минут, что является существенной нагрузкой на Active Directory.

Соответствия названий в веб-интерфейсе:

- **Users DN** — base DN в варианте Active Directory.
- **Custom User LDAP Filter** — LDAP-фильтр.
- **Search Scope / one level** — поиск только по дочерним объектам указанного объекта. Поиск по вложенным объектам не производится, также в результаты поиска не попадает сам базовый объект.
- **Search Scope / subtree** — поиск по всем дочерним объектам, включая вложенные. Базовый объект в поиск не попадает.
- **Full Sync Period** — частота выполнения полной синхронизации. Чем меньше пользователей, тем чаще можно выполнять полную синхронизацию.
- **Changed Users Sync Period** — частота выполнения частичного обновления пользователей (оптимальное значение по умолчанию — 60 секунд).

Список пользователей

Список пользователей позволяет проверить наличие пользователя и наполнение полей. Модификация пользователей из Active Directory не предусмотрена.

Состояние профилей (Active Directory + Keycloak конфигурация)

Поддерживается три состояния профиля (на основании состояния профиля (enabled/disabled) и принадлежности к группе). Поддержка состояния suspended включается в конфигурационном файле **nomail.keycloak.suspended.state.enabled**, название группы задается переменной **nomail.keycloak.allowed_group.name**:

- Активный — отсутствие каких либо флагов (в контексте Active Directory аккаунт должен быть enabled=True и принадлежать группе);
- Приостановлен — flag=SUSPENDED (либо enabled и не в группе, либо disabled и в группе);
- Удаленный — flag=DELETED (enabled=false + не в группе).

Если пользователь получает статус SUSPENDED (приостановлен), то осуществляется выход пользователя из системы. На устройства отправляется команда для выхода пользователя из системы и удаления всех локальных данных. Пользователь не может пользоваться сервисом, но в случае необходимости его можно быстро восстановить, и вся история у него сохранится.

При получении статуса DELETED (удален) на устройства пользователя отправляется команда для выхода из системы и удаления всех локальных данных. Вся история переписки на его устройствах удаляется. Также происходит удаление пользователя из всех групповых чатов, где он участвовал, без уведомлений участников групп. В случае необходимости пользователя можно восстановить из статуса DELETED — история личной переписки восстановится, но пользователю будет необходимо заново вступать в групповые чаты, в которых он ранее участвовал.

Журнал Keycloak

Расположение журнала: **/var/log/vector/k8s/keycloak/keycloak/*.**

В первую очередь следует отслеживать записи уровня ERROR. Пример такой ошибки: дубликат пользователя — когда один и тот же пользователь заведен в нескольких ветках LDAP-каталога. Как правило, после каждой записи ERROR идет трассировка ошибки для более детального анализа.

Пример лога

```
2020-04-14 12:06:12,984 ERROR [org.keycloak.services.error.KeycloakErrorHandler] (default
task-38) Uncaught server error: org.keycloak.models.ModelDuplicateException: Can't import user
'user2' from LDAP because email 'user@vkteams.example.com' already exists in Keycloak.
Existing user with this email is 'user1'
        at
org.keycloak.storage.ldap.mappers.UserAttributeLDAPStorageMapper.checkDuplicateEmail(UserAttribut
176)
        at
org.keycloak.storage.ldap.mappers.UserAttributeLDAPStorageMapper.onImportUserFromLDAP(UserAttribu
107)
        at
org.keycloak.storage.ldap.LDAPStorageProvider.importUserFromLDAP(LDAPStorageProvider.java:517)
        at
org.keycloak.storage.ldap.LDAPStorageProvider.searchForUser(LDAPStorageProvider.java:372)
        at
org.keycloak.storage.UserStorageManager.lambda$searchForUser$2(UserStorageManager.java:556)
        at org.keycloak.storage.UserStorageManager.query(UserStorageManager.java:505)
        at org.keycloak.storage.UserStorageManager.searchForUser(UserStorageManager.java:554)
        at
org.keycloak.models.cache.infinispan.UserCacheSession.searchForUser(UserCacheSession.java:583)
        at
org.keycloak.services.resources.admin.UsersResource.searchForUser(UsersResource.java:247)
        at org.keycloak.services.resources.admin.UsersResource.getUsers(UsersResource.java:
218)
```

Из примера видно, что сервис не может добавить пользователя user2 , так как почта user@vkteams.example.com принадлежит пользователю user1. Как правило, такие проблемы происходят при неправильном формировании фильтров и/или baseDN. Например, под фильтрование могут попадать календари или адресные книги пользователей, либо в Active Directory действительно заведены разные пользователи с одинаковым адресом почты.

Журнал сервиса Nomail (отправка OTP, авторизация)

Расположение журнала сервиса: **/oap/icq/logs/nomail-1.log.**

В журнале отображаются записи по синхронизации данных через LDAP — дополнительно к журналу сервиса Keycloak. Ошибки записываются с уровнем логирования ERROR, для просмотра таких записей выполните:

```
grep '] E ' /oap/icq/logs/nomail-1.log
```

Чаще всего ошибки возникают на этапе синхронизации пользователя или на этапе отправки OTP-сообщения, например из-за срабатывания антиспам-систем.

Если появился новый пользователь или произошли изменения в профиле, сервис Nomail оповещает об изменениях все остальные сервисы, которым необходимы эти данные. Например, сервисы поиска или хранения профилей пользователя.

Проблема синхронизации пользователей

В случае проблемы с синхронизацией пользователей ошибка, скорее всего, относится к сервису Keycloak. Поэтому:

1. Сначала проверьте журнал сервиса Keycloak.
2. Если пользователь заведен корректно, тогда проверьте журнал сервиса Nomail.

Проблемы с пользователем

Если есть проблемы с пользователем, изучите вхождения по конкретному пользователю в журнале Nomail.

Пример

```
>grep user@vkteams.example.com /oap/icq/logs/nomail-1.log
[2769513 14.04.2020 14:33:10] W NOMAIL: kkapi::load_user(user@vkteams.example.com)
[2769513 14.04.2020 14:33:11] W KEYCLOAK: user@vkteams.example.com's keycloak profile not
changed
```

Из примера видно, что в 14:33:10 Nomail проверил синхронизацию пользователя и обнаружил, что профиль пользователя не изменился и никаких действий выполнять не нужно.

Журнал базы данных

Расположение журнала: **/data/tarantool/logs/nomail-1.log.**

После старта БД в этот журнал записываются ошибки отправки OTP-сообщений. Поэтому любые записи, не связанные со стартом БД, говорят о потенциальных проблемах. OTP-сообщение отправляется в локальный MTA, поэтому дальнейший путь этого сообщения можно найти по логам MTA Postfix:

```
journalctl -t postfix/smtpd -t postfix/smtp
```

Пример отправки сообщения

```
2020-04-14T13:38:33.684799+03:00 onpremise postfix/smtpd[947406]: A72C4B059E:
client=localhost[127.0.0.1]
2020-04-14T13:38:33.724571+03:00 onpremise postfix/cleanup[947410]: A72C4B059E: message-
id=<20200414103833.A72C4B059E@onpremise.localdomain>
2020-04-14T13:38:33.726111+03:00 onpremise postfix/qmgr[554740]: A72C4B059E:
from=<otp@vkteams.example.com>, size=525, nrcpt=1 (queue active)
2020-04-14T13:38:33.730092+03:00 onpremise postfix/smtp[947412]: A72C4B059E:
to=<tester@vkteams.example.com>, relay=10.10.10.10[10.10.10.10]:25, delay=0.05,
delays=0.04/0/0/0, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as 870A62070BEC)
2020-04-14T13:38:33.730302+03:00 onpremise postfix/qmgr[554740]: A72C4B059E: removed
```

Из примера видно, что локальный MTA отправил почтовое сообщение на релей 10.10.10.10, и релей сообщение принял. Дальнейший путь этого сообщения не зависит от инсталляции VK Teams.

Так как OTP-сообщения отправляются ядром БД, в случае проблем с OTP следует смотреть оба журнала. Но прежде чем изучать проблему внутри этих сервисов, проверьте, что пользователь действительно существует в системе. Резюме процедуры:

1. Проверьте, что пользователь существует в системе.
2. Проверьте журнал БД:

```
/data/tarantool/logs/nomail-1.log
```

3. Проверьте журнал MTA Postfix:

```
journalctl -t postfix/smtpd -t postfix/smtp
```

Пример работы с журналами

Авторизация

Авторизация выполняется с помощью запроса `POST /wim/auth/clientLogin` на `u.<user_domain>` и проходит через лог-файл **nginx-im /oap/icq/domains/local_proxy.icq.com/logs/u-access.log.** Так как POST-запросы неудобно отслеживать через логи Nginx, то при условии, что в них нет явной ошибки (кодов 5xx, 4xx), в журналах сервиса стоит отслеживать `sapi_aim_web_service`, в который перенаправляются запросы этого типа.

Пример

```
grep -h 'POST /wim/auth/clientLogin' /oap/icq/logs/sapi_aim_web_services-*.err.log
14/150409 COMPAT0: sajp_wrapper.c:1539 SAJP_SendResponse >>10.10.10.10 POST /wim/auth/
clientLogin "clientName=Android%20myteam&clientVersion=1.0&devId=on2fah4R-
android&idType=ICQ&pwd=XXXXXXXXXXXXXX&s=tester%40example.com&tokenType=otp_via_email" 200
"200[0] OK" 0.091s [onpremise:2:38826580:1586865849.365] "myteam Android #no_user_id#
on2fah4R-android 1.0(9999999) Android_10_29 SM-G973F"
14/150851 COMPAT0: sajp_wrapper.c:1539 SAJP_SendResponse >>10.10.10.10 POST /wim/auth/
clientLogin "clientName=Android%20myteam&clientVersion=1.0&devId=on2fah4R-
android&idType=ICQ&pwd=XXXXXXXXXXXXXX&s=tester%40example.com&tokenType=otp_via_email" 200
"200[0] OK" 0.094s [onpremise:4:38826580:1586866131.237] "myteam Android #no_user_id#
on2fah4R-android 1.0(9999999) Android_10_29 SM-G973F
```

В примере имеются два запроса `clientLogin` для одного пользователя. В случае использования OTP это правильно, так как первый запрос вызывает отправку OTP-кода, а вторым запросом этот код отправляется на сервер.

Создание сессии

После авторизации пользователь может создавать новые сессии, которых может быть несколько при использовании различных устройств. Создание сессии выполняется запросом `POST /wim/aim/startSession.`

Пример

```
>grep -h 'startSession' /oap/icq/logs/sapi_aim_web_services-*.err.log
14/151808 COMPAT0: sajp_wrapper.c:1539 SAJP_SendResponse >>10.10.10.10 POST /wim/aim/
startSession
"a=%252Fw8BAAAAADJhgEAAAAAKz6rxWnlkhWnNX70ISPgqMAAAAXZC5hdmV0aXNvdkBjb3JwLm1haWwucnUAAAAFZW1hXX
cd527a2f2cdeff1a&events=myInfo%2Cpresence%2Cbuddylist%2Ctyping%2CwebrtcMsg%2CMChat%2Creplace%2Cpe
b80a-
fb80d2813a56&imf=plain&includePresenceFields=quiet%2Cssl%2CabFriendly%2Ccapabilities%2Cnick%2Crol
android&language=ru-
RU&minimizeResponse=0&mobile=1&pollTimeout=30000&rawMsg=0&sessionTimeout=31536000&ts=1586866688&v
200 "200[1] Ok" 0.004s [onpremise:3:39040498:1586866688.837] "myteam Android
tester@example.com on2fah4R-android 1.0(9999999) Android_10_29 SM-G973F"
```

Взаимодействие с пользователем

HTTPS API, сервис nginx-im

Протокол HTTPS (443/TCP) — основа коммуникации «клиент — сервер», поэтому при любых массовых проблемах с приложением необходимо проверять доступность по HTTPS. Все запросы от клиента (за исключением звонков) принимает сервис Nginx-im (сборка Nginx с дополнительными модулями, необходимыми для функционирования проекта).

Управление:

```
systemctl status nginx-im
```

Расположение конфигурационных файлов:

- /usr/local/nginx-im/conf/nginx.conf — основной файл;
- /usr/local/nginx-im/conf/conf.d/ — настройки отдельных виртуальных хостов.

Расположение журналов /oap/icq/domains/local_proxy.icq.com/logs/:

- *error.log — журналы ошибок отдельных виртуальных хостов;
- *access.log — журналы доступа отдельных виртуальных хостов.

В лог-файлах проверяйте наличие запросов и статусы HTTP-ответов. Основной точкой входа для запросов пользователя является `u.<user_domain>`, поэтому наиболее важными журналами являются:

- /oap/icq/domains/local_proxy.icq.com/logs/u-access.log
- /oap/icq/domains/local_proxy.icq.com/logs/u-error.log

Ротация лог-файлов происходит каждый час. Предыдущие журналы доступны в каталоге **/oap/icq/domains/local_proxy.icq.com/logs/old_logs/** и сжаты с использованием gzip (по умолчанию), lz4 или zstd.

> **Примечание**
>
> Файлы с расширением .zst необходимо открывать при помощи команды `zstdcat` и искать в них при помощи `zstdgrep`. Например: zstdgrep user@vkteams.example.com/oap/icq/domains/local_proxy.icq.com/logs/old_logs/u-access.log-20200410-1900.rot.zst

Проверка конфигурационных файлов на ошибки:

```
/usr/local/nginx-im/sbin/nginx -tc /usr/local/nginx-im/conf/nginx.conf
```

SSL-сертификаты:

- /usr/local/etc/im_ssl/default_ssl.cert
- /usr/local/etc/im_ssl/default_ssl.key

Голосовые и видеозвонки

Для работы голосовых и видеозвонков необходим доступ к внешнему IP-адресу VK Teams через порт 3478 (TCP/UDP) и UDP-порты выше 1024. Если доступа нет, звонки не будут работать.

В случае проблем со звонками в первую очередь проверьте, что предоставлены все необходимые доступы.

При необходимости возможно изменение количества UDP-портов, используемых для звонков. Для этого необходимо отредактировать конфигурационный файл **/usr/local/janus/etc/janus/janus.jcfg**, указав диапазон в параметре **rtp_port_range** секции **media**:

```
media: {
    rtp_port_range = "1024-65535"
}
```

При расчёте количества портов исходите из максимального количества пользователей, находящихся одновременно в звонках. Для каждого пользователя необходимо два порта для обеспечения возможности передачи медиаданных.

Если количество одновременно находящихся в звонке пользователей превышает расчетное количество портов, это может повлечь за собой недоступность функционала звонков. Поэтому сужение диапазона портов не должно производиться без крайней необходимости.

Push-сообщения

При получении сообщения на устройство пользователя отправляется push-уведомление. В зависимости от платформы уведомление отправляется либо на серверы Apple, либо на серверы Google. Журнал push-сервиса: **/var/log/mru-push-me/gosender.log.** Ошибки логируются с пометкой ERROR (или пометкой более высокого уровня), стандартные сообщения — с пометкой INFO.

Пример 1

```
2020-04-14 16:56:08:INFO:5852:+ios:icq aimsid=XXX.XXXXXXXX.topsecret:user@vkteams.example.com;
bundle_id=ru.mail.myteam-onpremise; hist_msg_id=0; token=XXXXXXXXXXXXXXXXXX
```

Отправлено push-сообщение для пользователя user, платформа iOS.

Пример 2

```
2020-03-22 10:45:27:FATAL:1098524:error create queueCluster nodes: [pusher.intim]; queues:
[127.0.0.1:3321]
```

Недоступна БД Tarantool с очередью на отправку.

Пример работы с журналами

В качестве примера рассматривается отправка сообщения. Для отправки сообщения выполняется запрос `POST /wim/im/sendIM.`

Пример

```
14/151406 COMPAT0: sajp_wrapper.c:1539 SAJP_SendResponse >>10.10.10.10 POST /wim/im/sendIM
"aimsid=001.3656109597.XXXXXXXXXX%3Atester%40example.com&f=json&message=XXXXXXX&notifyDelivery=tr
f56d4de3dfd9-15868662101408592&t=exampler%40example.com&updateMsgId=6815539145092366384" 200
"200[0] Ok" 0.009s [onpremise:50:38975693:1586866446.257] "myteam Desktop tester@example.com
on2fah4R-win 10.0.0(2132) Windows_10 PC"
```

В примере приведена отправка сообщения от пользователя user1 к пользователю user2. `sendIM` означает, что пользователь A отправил сообщение пользователю B, безотносительно факта получения сообщения. Для полноценной проверки потребуются дополнительные журналы, например журнал сервиса bos_srv, который отправляет сообщения от пользователя к пользователю.

Сервис bos_srv имеет несколько экземпляров приложения (инстансов), у каждого из них свой набор журналов, поэтому ниже используются подстановочные знаки (wildcards):

- **/oap/icq/logs/bos_srv-*.err.log** — лог ошибок. Кроме ошибок в этот лог попадает большое количество дополнительной информации, поэтому его неудобно использовать для поиска отправлений и получений.
- **/oap/icq/logs/bos_srv-*.access.log** — лог обращений клиента (аналог лога доступа в Nginx).
- **/oap/icq/logs/bos_srv-*.fss.log** — файл отражает весь путь сообщений. Он позволяет понять, было ли отправлено и получено сообщение.

Проверяем факт отправки сообщения:

```
grep '|IM|' /oap/icq/logs/bos_srv-*.fss.log
2020-04-14 00:05:38|10.10.10.10|user1@vkteams.example.com|user2@vkteams.example.com|->|myteam
Desktop user1@vkteams.example.com on2fah4R-mac 5.0.0(2134) MacOSX_10.15 PC|IM|-|
aimsid=001.0805324701.XXXXXXXXXX:user1@vkteams.example.com,devId=on2fah4R-
mac,mmb=50;0;2134;1999,device=XXXXXXXXXX,msgId=86a7345a-7dca-11ea-952d-52540098da02,pin=0,telecha
```

Для одного сообщения таких записей будет две, так как запись происходит как на сервисе bos_srv отправителя, так и на сервисе bos_srv получателя. Направление отправки зависит от указателя после второго адреса почты. В данном примере `->` означает отправку от user1 к user2. Отправка от user2 к user1 будет обозначена как `<-`.

Проверяем, что приложение клиента забрало сообщение (тип записи `|HIST|`):

```
2020-04-14 00:00:36|46.73.143.144|user1@vkteams.example.com|user2@vkteams.example.com|<-|
myteam Desktop user1@vkteams.example.comon2fah4R-mac 5.0.0(2796) MacOSX_10.15 PC|HIST|-|
aimsid=001.1947658530.XXXXXXXXXX:aaa@vkteams.example.com,devId=on2fah4R-
mac,mmb=50;0;2796;1999,device=XXXXXXXXX,histId=6815180970589684273,phone=00000000000,stranger=0
```

Из примера видно, что приложение клиента user1 забрало сообщение, которое было отправлено от user2@vkteams.example.com (указатель направления в данной строчке `<-`).

Управление пользователями без контроллера домена

Через CLI-утилиту users.py

Чтобы добавить, удалить или обновить пользователей, минуя контроллер домена, можно использовать специальную CLI-утилиту users.py:

Команды

```
usage: users.py [-h] [-c CONFIG_FILE] --cmd COMMAND [-m MAX_USERS]
                [--users [N [N ...]]]
 optional arguments:
  -h, --help           show this help message and exit
  -c CONFIG_FILE       Path to users list
  --cmd COMMAND        Command: add, update, list, delete, search
  -m MAX_USERS         Max users count for list and search
  --users [N [N ...]]  user's emails for search
```

- Показать список всех пользователей (в том числе и пользователей из контроллера домена):

  ```
  /usr/local/bin/users.py --cmd list
  ```

  Количество пользователей, отображаемых командой list, ограничивается параметром `-m`. По умолчанию выводится 100 пользователей. Вывод большого количества пользователей может занять длительное время, поэтому рекомендуется использовать команду `search`, которая позволяет выдавать список пользователей поиском по подстроке.

- Показать список всех пользователей (в том числе и пользователей из контроллера домена), у которых в адресе почты, фамилии или имени есть требуемая подстрока (в данном примере две подстроки user01 и user02). Количество пользователей ограничивается параметром `-m`.

  ```
  /usr/local/bin/users.py --cmd search --users user01 user02
  ```

- Добавить пользователей:

  ```
  /usr/local/bin/users.py --cmd add -c ./users.yaml
  ```

  Прежде чем отдавать команду на добавление пользователей, необходимо в любой удобной папке создать файл **users.yaml** и заполнить его данными пользователей (описание формата YAML-файла представлено ниже).

- Обновить пользователей:

  ```
  /usr/local/bin/users.py --cmd update -c ./users.yaml
  ```

- Удалить пользователей:

```
/usr/local/bin/users.py --cmd delete -c ./users.yaml
```

или

```
/usr/local/bin/users.py --cmd delete --users user01@vkteams.example.com
```

- Вывод списка пользователей поиском по подстроке (количество пользователей ограничивается параметром `-m`):

```
// найти всех пользователей, у которых в полях login, lastName,
// firstName присутствует слово Petrov
users.py --cmd search --users Petrov.
```

Формат файла users.yaml

При добавлении пользователей по умолчанию ищется файл **./users.yaml** (можно изменить параметром `-c`).

1. Основной формат файла **users.yaml** (объект users имеет тип Array):

```
users:
- 'user1@mail.ru'
- 'Surname user2@bk.ru'
- 'Name Surname user3@list.ru'
- 'Surname Name MiddleName user4@inbox.ru'
```

Варианты описания пользователя:

- email
- lastName email
- firstName lastName email
- lastName firstName middleName email

В базе пользователей нет понятия отчества, поэтому оно не обрабатывается. В варианте email (только адрес почты), lastName (фамилия) и firstName (имя) будут заполнены именем до знака @ в адресе почты.

Пример

```
users:
- 'user1@icq.com'
- 'Ivanov user2@icq.com'
- 'Alexander Petrov user3@icq.com'
- 'Alexander Ivanov Ivanovich user4@icq.com'
```

2. Расширенный формат файла **users.yaml** (объект users имеет тип Hash):

```
users:
    user01@vkteams.example.com:
        email: user01@vkteams.example.com #почта
        firstName: FirstName01 #имя
        lastName: LastName01 #фамилия
        enabled: false #заблокирован ли пользователь
        attributes:
            mobile: +000000000 #телефон
            department: Test Department #отдел
            title: Job title N1 #должность
            middleName: MiddleName01 #отчество
            memberOf: ["VIP1"] #член группы VIP1
    user02@vkteams.example.com:
        email: user02@vkteams.example.com
        firstName: FirstName02
        lastName: LastName02
        enabled: false
        attributes:
            mobile: +000000001
            department: Test Department
            title: Job title N2
            middleName: MiddleName02
            memberOf: ["VIP1"]
```

При использовании расширенного формата .yaml-файла, username должен совпадать с email. В примере выше это user01@vkteams.example.com.

Примечание

Возможно использование утилиты `users.py` без использования YAML-файла. Вместо этого можно использовать параметр `--users`, например:

`users.py --cmd delete --users user01@vkteams.example.com user02@vkteams.example.com`

`users.py --cmd search --users user01@vkteams.example.com user02@vkteams.example.com`

Обратите внимание, что при использовании команд `add` и `update` пользователи будут созданы без фамилии и имени (lastName, firstName).

Через kccli-утилиту

Ниже представлены команды для управления пользователями при помощи kccli-утилиты:

- Поиск информации по пользователю с использованием основных атрибутов пользователя:

```
kccli user search <email> или <имя|фамилия>
```

- Поиск информации по пользователю только по почте:

```
kccli user get <email>
```

- Добавить пользователя:

```
kccli user add  <email>
```

При добавлении пользователя из консоли пользователи будут созданы без фамилии, имени и отчества (lastName, firstName и middleName) — они будут назначены автоматически. Добавить пользователя с заданными фамилией, именем и отчеством возможно при помощи YAML-файла (см. описание ниже).

- Удалить пользователя:

```
kccli user delete <email>
```

- Добавить пользователя при помощи YAML-файла:

Предварительно создайте в любой удобной директории файл **users.yaml** и заполните его данными пользователей.

Расширенный формат файла **users.yaml** (объект users имеет тип Hash):

```
users:
    user01@vkteams.example.com:
        email: user01@vkteams.example.com #почта
        firstName: FirstName01 #имя
        lastName: LastName01 #фамилия
        enabled: false #заблокирован ли пользователь
        attributes:
            mobile: +000000000 #телефон
            department: Test Department #отдел
            title: Job title N1 #должность
            middleName: MiddleName01 #отчество
            memberOf: ["VIP1"] #член группы VIP1
    user02@vkteams.example.com:
        email: user02@vkteams.example.com
        firstName: FirstName02
        lastName: LastName02
        enabled: false
        attributes:
            mobile: +000000001
            department: Test Department
            title: Job title N2
            middleName: MiddleName02
            memberOf: ["VIP1"]
```

После создания **users.yaml** выполните в консоли команду:

```
kccli user add -f <имя файла>
```

- Удалить пользователя при помощи YAML-файла:

`kccli user delete -f <имя YAML-файла>`

Через веб-интерфейс сервиса Keycloak

Инструкции см. в разделе Авторизация в системе.

Необходимо учесть, что веб-интерфейс не поддерживает batch-режим, поэтому каждым пользователем придется управлять отдельно, что неудобно при обработке большого количества пользователей.

Доступ в окружение администратора

Доступ в окружение администратора настраивается при установке системы и описан в документации по установке:

- На одну виртуальную машину.
- Кластера.

Доступ в окружение администратора проверяется пересечением атрибута **memberOf** в файле **users.yaml** с перечнем групп, указанных в секции **otp_permission** конфигурационного файла инсталляции **/usr/local/etc/premsetup/defaults.yaml**.

Добавить этот атрибут к существующему пользователю можно при помощи команды `--cmd update`, с использованием расширенного формата **users.yaml**.

```
users:
    user01@vkteams.example.com:
        attributes:
            memberOf: [ 'myteam-admin' ]
```

Настройки клиентских приложений

Разрешить изменение имени контакта

Чтобы настроить для пользователей возможность изменять имена контактов, в конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** необходимо:

1. Добавить параметр **allow-contacts-rename** и указать для него флаг true или false:

   ```
   {
       },
       "allow-contacts-rename": true,
       ....
       },
   }
   ```

   где:

   - false — отключает возможность изменения имени контакта.
   - true — разрешает изменение имени контакта. Измененное имя будет видно только тому пользователю, который внес это изменение.

2. Пересоздать под админ-консоли:

   ```
   kubectl delete pod -n vkteams myteam-admin-\*
   ```

   где: ***** — уникальное имя пода. Имя пода необходимо получить с помощью вывода команды:

   ```
   kubectl get pods -A | grep myteam-admin
   ```

Настроить порядок отображения фамилии, имени и отчества в клиентском приложении

По умолчанию отображение фамилии, имени и отчества в инсталляциях следующее: имя, отчество, фамилия.

Чтобы изменить порядок отображения фамилии, имени и отчества в клиентском приложении:

Шаг 1. В конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** добавить в основную секцию следующие настройки:

```
"leading-last-name": true,
"allow-contacts-rename": false
```

где:

- leading-last-name: false — отображает фамилию контакта в конце;
- leading-last-name: true — отображает фамилию контакта в начале.

Шаг 2. Пересоздать под админ-консоли:

```
kubectl delete pod -n vkteams myteam-admin-\*
```

где: ***** — уникальное имя пода. Имя пода необходимо получить с помощью вывода команды:

```
kubectl get pods -A | grep myteam-admin
```

Шаг 3. В конфигурационных файлах сервисов Prof-st, Front и Beagle добавить настройку:

```
swap_person_first_last_name true
```

Расположение конфигурационных файлов:

/usr/local/etc/front-1.conf

/usr/local/etc/front-2.conf

/usr/local/etc/front-3.conf

/usr/local/etc/front-4.conf

/usr/local/etc/beagle-1.conf

/usr/local/etc/prof-st-1.conf

Для инсталляций, где уже были созданы пользователи и эти пользователи имеют непустые контакт листы выполните шаги, описанные ниже.

> **Примечание**
>
> Рекомендуется проводить в часы наименьшей активности пользователей, чтобы снизить нагрузку на сервера.

Шаг 1. Средствами виртуальной машины выполнить бэкап/точку восстановления на случай сбоя.

Шаг 2. В каждом инстансе сервиса Cox:

1. Включить уровень логирования в сервисе Cox (main) >= 3 (INFO).

   Уровень логирования при старте сервиса задаётся аргументом сервиса `-l`:

   ```
   systemctl status cox-1
   ```

   Изменить уровень логирования в работающем сервисе можно через ввод в управляющий порт `set log_level 3`.

   Номер командного порта можно узнать из настроек сервиса в значении переменной `compot_bind`. Пример для конфигурационного файла **/usr/local/etc/cox-1.conf**:

   ```
   sudo grep compot_bind /usr/local/etc/cox-1.conf
   ```

   Подключиться к командному порту можно утилитой nc (netcat):

   ```
   nc 0.0.0.0 4221
   set log_level 3
   ```

2. В настройках сервиса Cox (main) присвоить значение переменной **cox.remove_buddy_aliases.dryrun false**.

   В работающем сервисе это значение меняется через введение в управляющий порт команды `set cox.remove_buddy_aliases.dryrun false`:

   ```
   nc 0.0.0.0 4221
   set cox.remove_buddy_aliases.dryrun false
   ```

3. В управляющий порт сервиса Cox (main) ввести команду `remove_buddy_aliases`:

   ```
   nc 0.0.0.0 4221
   remove_buddy_aliases
   ```

4. Просмотреть логи сервиса Cox:

   ```
   tail -f /oap/icq/logs/cox-1.log
   ```

   В логах сервиса ожидаются записи вида `Remove buddy aliases for sn`. Окончание операции логируется записью вида `BuddyAliasRemover finished`.

> **Важно**
>
> Не рекомендуется выполнять команды на нескольких инстансах одновременно. Это может привести к избыточной нагрузке.

Шаг 3. В каждом инстансе сервиса Feedog:

1. Определить управляющий порт инстанса командой `ps aux | grep feedog_srv`. Порт задаётся аргументом `-p`:

```
ps aux | grep feedog_srv
```

```
[centos@ ~]$ ps aux | grep feedog_srv
quantum    29772  0.3  0.7 1606040 393752 ?      S<l   2023 425:05 ./feedog_srv im -p 4331 -n feedog_
quantum    29860  0.3  0.7 1581456 345884 ?      S<l   2023 400:11 ./feedog_srv im -p 4332 -n feedog_
quantum    29943  0.3  0.8 1622428 399772 ?      S<l   2023 407:12 ./feedog_srv im -p 4333 -n feedog_
quantum    30021  0.3  0.6 1581464 345424 ?      S<l   2023 403:47 ./feedog_srv im -p 4334 -n feedog_
quantum    30087  0.3  0.7 1625492 385160 ?      S<l   2023 445:11 ./feedog_srv im -p 4335 -n feedog_
quantum    30141  0.3  0.7 1583516 347436 ?      S<l   2023 408:19 ./feedog_srv im -p 4336 -n feedog_
quantum    30237  0.3  0.7 1599896 355912 ?      S<l   2023 430:32 ./feedog_srv im -p 4337 -n feedog_
quantum    30266  0.3  0.8 1625496 418484 ?      S<l   2023 424:26 ./feedog_srv im -p 4338 -n feedog_
centos   1293857  0.0  0.0 112832  2332 pts/0    S+   10:44   0:00 grep --color=auto feedog_srv
```

2. Подключиться к управляющему порту можно с помощью утилиты `cpsh`.
3. Выполнить `kill_bat_all -delay <delay_milliseconds>`. delay указывать исходя из нагрузки. Значение по умолчанию (если не указать ключ `-delay`) — 256:

```
cpsh 4331
im:feedog_srv-a01.1% kill_bat_all
```

4. Просмотреть логи сервиса Feedog (просмотр логов — **/oap/logs/feedog_srv-a01.<INSTANCE_NUMBER>.err**):

```
tail -f /oap/logs/feedog_srv-a01.1.err
```

   Операция сопровождается логированием сообщений вида `BUCKY_DOMAIN: Dropping next bucket`. Прекращение логирования подобных сообщений соответствуют окончанию операции.

> **Важно**
>
> Не рекомендуется выполнять команды на нескольких инстансах одновременно. Это может привести к избыточной нагрузке.

Шаг 4. В каждом инстансе сервиса Boss:

1. Определить управляющий порт инстанса командой `ps aux | grep bos_srv`. Порт задаётся аргументом `-p`:

```
ps aux | grep bos_srv
```

```
[centos@ ~]$ ps aux | grep bos_srv
centos   1097888  0.0  0.0 112828  2328 pts/0    S+   09:54   0:00 grep --color=auto bos_srv
root     2472058  0.0  0.0 241456  7088 pts/2    S+    2023   0:00 /usr/bin/sudo -E env LD_LIBRARY_PA
root     2472060  0.0  0.0   4376  1380 pts/2    S+    2023   0:00 scl enable devtoolset-10  'tail' '
root     2472082  0.0  0.0 108112   720 pts/2    S+    2023   0:00 tail -f /oap/icq/logs/bos_srv-a01.
quantum  2907004  0.7  0.8 2476364 425924 ?      S<l   2023 702:23 ./bos_srv im -p 4330 -n bos_srv-a0
quantum  2907006  0.7  0.8 2444616 444080 ?      S<l   2023 721:05 ./bos_srv im -p 4323 -n bos_srv-a0
quantum  2907036  0.7  0.9 2495816 453940 ?      S<l   2023 747:30 ./bos_srv im -p 4329 -n bos_srv-a0
quantum  2907050  0.6  0.8 2521412 425448 ?      S<l   2023 680:12 ./bos_srv im -p 4328 -n bos_srv-a0
quantum  2907075  0.6  0.8 2470212 424704 ?      S<l   2023 694:07 ./bos_srv im -p 4327 -n bos_srv-a0
quantum  2907083  0.7  0.8 2601288 424712 ?      S<l   2023 695:29 ./bos_srv im -p 4325 -n bos_srv-a0
quantum  2907093  0.6  0.8 2520388 424652 ?      S<l   2023 678:00 ./bos_srv im -p 4326 -n bos_srv-a0
quantum  2907103  0.6  0.8 2520388 424588 ?      S<l   2023 690:51 ./bos_srv im -p 4324 -n bos_srv-a0
```

2. Подключиться к управляющему порту можно с помощью утилиты `cpsh`.
3. Выполнить `kill_bat_all -delay <delay_milliseconds>`. delay указывать исходя из нагрузки. Значение по умолчанию (если не указать ключ `-delay`) — 256:

```
cpsh 4323
im:bos_srv-a01.1% kill_bat_all -delay 1000
```

4. Просмотреть логи сервиса Boss (просмотр логов — /oap/icq/logs/bos_srv-a01.<INSTANCE_NUMBER>.err.log):

```
tail -f /oap/icq/logs/bos_srv-a01.1.err.log
```

Операция сопровождается логированием сообщений вида `BUCKY_DOMAIN: Dropping next bucket`. Прекращение логирования подобных сообщений соответствуют окончанию операции.

> **Важно**
>
> Не рекомендуется выполнять команды на нескольких инстансах одновременно. Это может привести к избыточной нагрузке.

Шаг 5. Вернуть в исходное состояние уровень логирования в сервисе Cox.

Настроить витрину чатов

В витрине отображаются чаты в зависимости от региона пользователя, который вычисляется по геолокации IP-адресов.

Для конфигурации витрины чатов необходимо получить доступ по SSH к машине, на которой запущен сервис Chatexpo:

1. Проверить, что подключились к машине, на которой запущен сервис Chatexpo:

```
pgrep -a chatexpo
6442 /usr/local/bin/chatexpo -c /usr/local/etc/chatexpo-1.conf -l 2 -o /oap/icq/logs/
chatexpo-1.log
```

2. Получить доступ к командному порту сервиса Chatexpo:

```
sudo rg compot /usr/local/etc/chatexpo-1.conf
49:# compot
50:compot_bind 127.0.0.1:4523
```

3. Подключиться к командному порту:

```
rlwrap nc 127.0.0.1 4523
```

Далее можно приступать к конфигурации витрины чатов.

Добавить чат в витрину

1. Для добавления чата в витрину выполнить:

```
app.tnt.region_chats.add RU 1@chat.agent 1
status=0, reason=ok
```

2. Проверить в клиентском приложении, что чат добавлен.

Удалить чат из витрины

1. Для удаления чата из витрины выполнить:

```
app.tnt.region_chats.remove RU 1@chat.agent
status=0, reason=ok
```

2. Проверить в клиентском приложении, что чат удален.

Сменить порядок чатов в витрине

Наиболее простой путь изменение порядка чатов в витрине — удаление чатов из витрины и добавление заново в необходимом порядке.

Предположим, в витрине отображаются 3 чата, например:

```
app.tnt.region_chats.add RU 1@chat.agent 1
status=0, reason=ok
app.tnt.region_chats.add RU 14@chat.agent 2
status=0, reason=ok
app.tnt.region_chats.add RU 26@chat.agent 3
status=0, reason=ok
```

и необходимо поставить третий чат на первое место, первый на второе и второй на третье.

Для этого необходимо:

1. Удалить первый и третий чаты:

   ```
   app.tnt.region_chats.remove RU 1@chat.agent
   status=0, reason=ok
   app.tnt.region_chats.remove RU 26@chat.agent
   status=0, reason=ok
   ```

2. Добавить удаленные чаты заново в нужном порядке:

   ```
   app.tnt.region_chats.add RU 26@chat.agent 1
   status=0, reason=ok
   app.tnt.region_chats.add RU 1@chat.agent 2
   status=0, reason=ok
   ```

3. Проверить, что чаты добавились:

   ```
   app.tnt.region_chats.list RU
   status=0, reason=ok
   {"id":"26@chat.agent","pos":1}
   {"id":"1@chat.agent","pos":2}
   {"id":"14@chat.agent","pos":3}
   ```

   Второй чат сам сдвинется на третье место.

4. Проверить порядок чатов в клиентском приложении.

Удалить всю витрину

1. Проверить, какие чаты добавлены в витрину:

   ```
   app.tnt.region_chats.list RU
   status=0, reason=ok
   {"id":"26@chat.agent","pos":1}
   {"id":"1@chat.agent","pos":2}
   ```

```
{"id":"14@chat.agent","pos":3}
{"id":"2@chat.agent","pos":5}
{"id":"3@chat.agent","pos":6}
{"id":"1586@chat.agent","pos":7}
{"id":"1585@chat.agent","pos":8}
{"id":"1587@chat.agent","pos":9}
{"id":"1787@chat.agent","pos":10}
```

2. Удалить чаты по одному:

```
app.tnt.region_chats.remove RU 26@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 1@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 14@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 2@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 3@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 1586@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 1587@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 1787@chat.agent
status=0, reason=ok
app.tnt.region_chats.remove RU 1585@chat.agent
status=0, reason=ok
app.tnt.region_chats.list RU
status=0, reason=ok
```

3. Проверить в клиентском приложении, что витрина удалена.

Включить папки в клиентском приложении

Максимальное количество папок по умолчанию — 10. Папки синхронизируются между всеми активными сессиями и платформами.

Чтобы включить отображение папок в клиентском приложении, необходимо:

1. Указать в конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** значение true для секции **folders-enabled**:

   ```
   {
     "api-version": 112,
     "archive-enabled": true,
     "folders-enabled": true,
     //
   }
   ```

2. Пересоздать под админ-консоли:

   ```
   kubectl delete pod -n vkteams myteam-admin-\*
   ```

   где: ***** — уникальное имя пода. Имя пода необходимо получить с помощью вывода команды:

   ```
   kubectl get pods -A | grep myteam-admin
   ```

Включить мини-апп «Опросы»

Если мини-апп «Опросы» еще не установлен, его необходимо установить:

1. Если в конфигурации инсталляции один экземпляра сервиса Nomail, пропустите данный шаг и перейдите к шагу 2.

   Если в конфигурации инсталляции существует более одного экземпляра сервиса Nomail, определите сервер на котором располагается необходимый экземпляр. Выполните скрипт instance_detect.sh на любом сервере в инсталляции:

   ```
   /usr/local/miniapp_deploy/instance_detect.sh
   ```

   В выводе команды будет необходимый сервер, в примере ниже это nomail.onpremise.nomail-1:

   ```
   d57ae511 in [0 ffffffff] 01 nomail.onpremise.nomail-1 alive
   ```

2. Подключитесь к серверу из шага 1 по SHH.
3. Авторизуйтесь под root-пользователем.
4. Выполните скрипт установки мини-аппа опросов survey_deploy.sh:

   ```
   /usr/local/miniapp_deploy/survey_deploy.sh
   ```

   Во время установки мини-аппа в консоли будет отображаться информация о пройденных этапах. В случае возникновения ошибки установки отобразится описание, какой этап не удалось выполнить. В случае успешной установки мини-аппа отобразится строка «All operations have been completed successfully. Check working miniapp!».

После того как мини-апп опросов установлен, управлять им можно в панели администратора VK WorkSpace (доступна при наличии интеграции с Почтой VK WorkSpace), см. документацию.

Запретить использование устаревших версий клиентских приложений

Начиная с версии VK Teams 24.2 вы можете запретить пользователям использовать клиентские приложения, версия которых ниже минимально поддерживаемой версии (устанавливается администратором организации). По умолчанию запрет не включен.

Если запрет установлен и версия клиентского приложения ниже минимально поддерживаемой, пользователю для продолжения работы необходимо обновить приложение.

Чтобы установить запрет:

1. В конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** в секции **updates** установите минимально поддерживаемую версию для каждой платформы:

   ```
   "min_supported_version": {
     "version": "10.6.0",
     "build": "528",
     "full_version": "10.6.0.528"
   ```

   где:

   - version — номер версии.
   - build — номер сборки. Если номер сборки неизвестен, допустимое значение — 0.
   - full_version — номер версии и номер сборки, разделенные точкой (full_version: version.build).

   Если минимально поддерживаемая версия указана некорректно, запрет на использование устаревших версий клиентских приложений не будет выполняться.

2. В поле **force_update_enabled** установите запрет на использование устаревших версий клиентских приложений:

   ```
   "force_update_enabled": true
   ```

   где:

   - true — включает запрет на использование устаревших версий клиентских приложений.
   - false — выключает.

3. Укажите в поле **updates.customLandingUrl** адрес dl-лендинга, с которого можно скачать обновление для клиентского приложения:

   ```
   "customLandingUrl": "https://dl.<your_domain>.ru",
   ```

   Если поле **customLandingUrl** не заполнено, пользователю будет предложено скачать обновление с apps.<platform>.url.

   Пример настройки запрета в **myteam-config.json** для всех платформ:

```
"updates": {
    "customLandingUrl": "https://dl.<YOUR_DOMAIN>.ru",
    "android": {
      "min_supported_version": {
        "version": "10.6.0",
        "build": "528",
        "full_version": "10.6.0.528"
      },
      "force_update_enabled": true
    },
    "ios": {
      "min_supported_version": {
        "version": "10.6",
        "build": "90186",
        "full_version": "10.6.90186"
      },
      "force_update_enabled": true
    },
    "linux_x64": {
      "min_supported_version": {
        "version": "10.0",
        "build": "9706",
        "full_version": "10.0.9706"
      },
      "force_update_enabled": true
    },
    "mac_x64": {
      "min_supported_version": {
        "version": "5.0",
        "build": "9702",
        "full_version": "5.0.9702"
      },
      "force_update_enabled": true
    },
    "win_x32": {
      "min_supported_version": {
        "version": "10.0",
        "build": "9707",
        "full_version": "10.0.9707"
      },
      "force_update_enabled": true
    }
  }
```

4. Пересоздайте под админ-консоли:

```
kubectl delete pod -n vkteams myteam-admin-\*
```

где ***** — уникальное имя пода. Имя пода необходимо получить с помощью вывода команды:

```
kubectl get pods -A | grep myteam-admin
```

Работа с системой

Настроить обратную связь

По умолчанию все обращения пользователей поступают на адрес myteamsupport@USER-DOMAIN, через локальный SMTP-релей. Например, в случае домена example.com обращение поступит на адрес myteamsupport@example.com.

Настройки сервиса обратной связи определяются секцией **email_config** конфигурационного файла **/usr/local/etc/krtek/krtek.config.yaml**.

Базовые настройки сервиса:

В полях **from:** и **rcpt_to:** в адреса, оканчивающиеся символом @, автоматически подставляется домен пользователя.

```
email_config:
  from: "myteamsupport@"
  rcpt_to: ["myteamsupport@"]
  subject: "VK Teams Feedback"
```

Параметр	Тип	Описание	Примеры
from	Строка	Обратный адрес для письма, формируемого системой в адрес технической поддержки	• test@ — обратный адрес будет test@USER-DOMAIN; • test@example.com — обратный адрес будет test@example.com, независимо от домена пользователя; • example@gmail.com
rcpt_to	Массив	Адрес получателей. Получателей может быть несколько	• ['test@'] — получателем письма будет test@USER-DOMAIN • ['test@', 'example@example.com'] — получателями письма будут test@USER-DOMAIN и example@example.com
subject	Строка	Тема отправляемого письма	

Расширенные настройки сервиса:

Используйте расширенные настройки, если хотите отправлять обращения пользователей через отдельный SMTP-сервер с использованием авторизации.

```
email_config:
  host: "localhost"
  port: 25
  username: "example@gmail.com"
  password: ""
  use_tls: false
  from: "myteamsupport@"
  rcpt_to: ["myteamsupport@"]
  subject: "Myteam Feedback"
  templates_dir: "/usr/local/krtek/templates/email"
```

Параметр	**Тип**	**Описание**	**Значение / настройка по умолчанию**	Пример
host	Строка	Адрес SMTP-сервера	localhost	smtp.gmail.com
port	Int	Порт SMTP-сервера	25	587
username	Строка	Имя пользователя для авторизации на SMTP-сервере	без авторизации	example@gmail.com
password	Строка	Пароль для авторизации на SMTP-сервере	без авторизации	
use_tls	Boolean	Форсировать использование TLS для SMTP-сервера	выключено	true
templates_dir	Строка	Место на диске, где хранится шаблон почтовых писем		/usr/local/krtek/templates/email

Примечание

Перед отправкой обращения администратору к пользовательскому обращению будет применен шаблон. За путь к шаблону отвечает параметр **templates_dir** секции **email_config** конфигурационного файла **krtek.yaml.conf**.
Для смены шаблона необходимо отредактировать файл **feedback.html**, расположенный в **/usr/local/krtek/templates/email**.

Оценить состояние сервисов инсталляции

Все сервисы инсталляции взаимодействуют по протоколу IPROS — бинарный протокол обмена сообщениями между сервисами.

Чтобы найти друг друга, сервисы регистрируются в контроллере (сервис Ctrl). Контроллер хранит информацию о всех сервисах, зарегистрированных в нем, и уведомляет об изменениях в инсталляции.

Чтобы запросить у контроллера информацию о состоянии сервисов инсталляции, используется скрипт `ic srvc`.

Скрипт отображает состояние инстансов всех сервисов, которые зарегистрированы в контроллере в данный момент. Отображает IP-адрес и порт, на котором зарегистрирован сервис, а также его состояние. Возможны три состояния сервиса:

- alive — сервис функционирует. Если сервис не в статусе alive, посмотрите состояние инстансов сервиса при помощи команды `system status <наименование сервиса>`.
- busy — сервис какое-то время не связывался с контроллером (например, из-за большой нагрузки на виртуальную машину), или очередь IPROS-сообщений переполнена.
- dead — сервис не функционирует. Для такого состояния отображается последнее время регистрации сервиса в контроллере.

Информация об инструментах сбора логов клиентских приложений и серверных логов, расположение логов, а также примеры логов клиентских приложений описаны в документации по логам.

Отключить кэш файлов, передаваемых через сервис Go-files

Для повышения безопасности системы есть возможность отключить доступ к файлам через кэш сервиса Nginx. Это увеличит нагрузку на сервисы Go-files и Krueger.

Если не отключать кэш, максимальное время хранения файла в кэше 20 минут. После чего файл будет удален из кэша, если к нему не было обращений.

```
sed -i -E ``"/(proxy_cache off;|#.+)/! s/(proxy_cache.*)\ (.+)/#\1 \2/g"` `/usr/local/nginx-im/conf/vhost_servers/gofiles.conf
sed -i -E ``"/(proxy_cache off;|#.+)/! s/(proxy_cache.*)\ (.+)/#\1 \2/g"` `/usr/local/nginx-im/conf/conf.d/files-backend.conf
nginx.sh reload
```

Добавить собственные правила фильтрования сетевого трафика

Если требуется добавить собственное правило в iptables, то сделать это можно стандартными средствами через команду `iptables`.

Например:

```
iptables -A INPUT -p tcp --dport 10050 -j ACCEPT
```

Сохранение правил необходимо выполнять с помощью скрипта:

```
/usr/local/bin/iptables-save-for-puppet.sh
```

> **Важно**
>
> Выполнять сохранение с помощью стандартной команды `iptables-save` нельзя.

Обновить SSL-сертификаты

Действия по обновлению SSL-сертификатов, полученных без использования Let`s Encrypt:

1. Скопируйте обновленный ключ в директорию **/usr/local/etc/im_ssl/**, а сертификат — в директорию **/usr/local/etc/im_ssl/**.
2. Выполните команду:

```
nginx.sh update_certs
```

В дальнейшем для удобства обновления добавьте новые сертификаты в файл **/usr/local/etc/premsetup/defaults.yaml** (см. документ Инструкция по установке VK Teams на одну виртуальную машину).

Переход с Let`s Encrypt на собственные сертификаты

Чтобы перейти на собственные сертификаты:

1. Остановите таймер im-acme-renew.timer, выполнив команды:

```
systemctl stop im-acme-renew.timer
systemctl disable im-acme-renew.timer
```

2. Скопируйте обновленный ключ в директорию **/usr/local/etc/im_ssl/**, а сертификат — в директорию **/usr/local/etc/im_ssl/.**
3. Выполните команду:

```
nginx.sh update_certs
```

В дальнейшем для удобства обновления добавьте новые сертификаты в файл **/usr/local/etc/premsetup/defaults.yaml** (см. документ Инструкция по установке VK Teams на одну виртуальную машину).

Дата обновления документа: 04.07.2024 г.