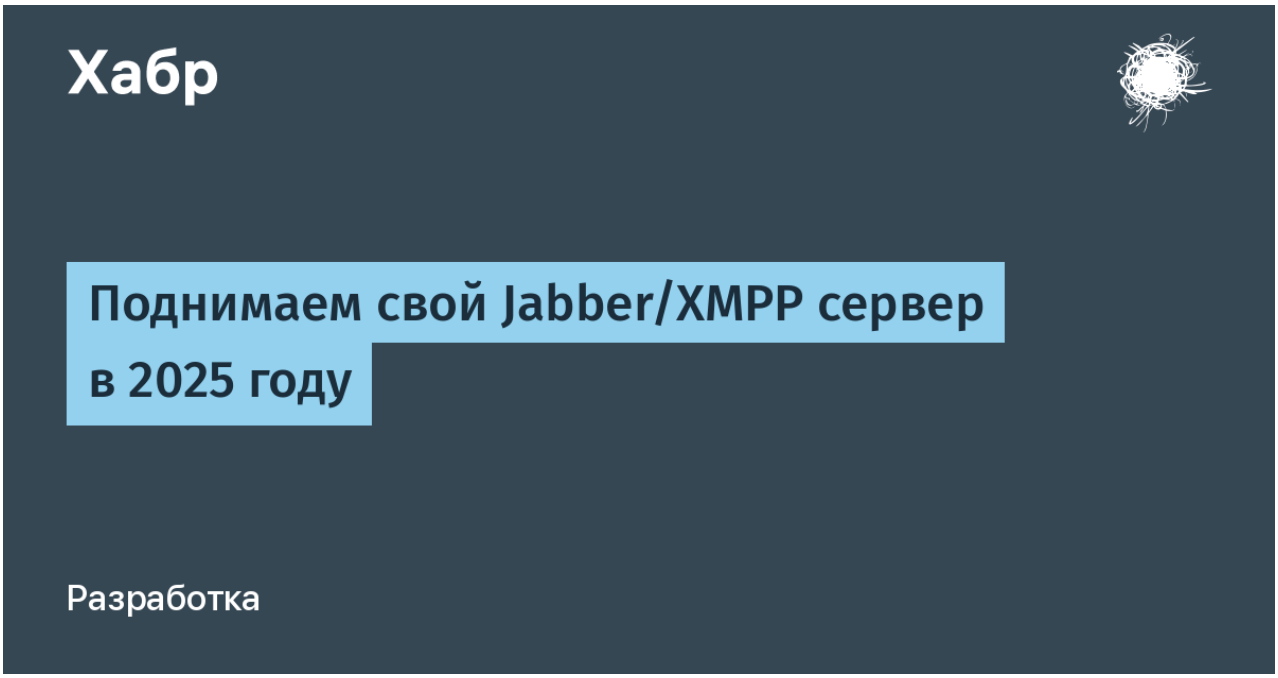
Поднимаем свой Jabber/XMPP сервер в 2025 году

habr.com/ru/articles/931292

Uporoty July 26, 2025



Средний

8 мин

11K
Туториал

WhatsApp в России депутаты обещают заблокировать уже в очень скором времени. Доверие к Telegram у многих довольно сильно было подорвано после публикации расследования одного издания об особенностях их протокола вкупе с подробностями о том, кто владеет их сетевой инфраструктурой и с кем он связан (администрация Хабра попросила меня не прикладывать ссылку, т.к. издание признано "нежелательной организацией" в РФ), да и недавнее появление Telegram в реестре РКН тоже оптимизма в их отношении не добавляет. Max - оставим для сумасшедших и безвыходных, учитывая, кто и зачем его создал. Signal - отличный мессенджер и всем хорош, но в России его тоже периодически пытаются заблокировать.

На фоне всего этого многие начали задумываться о поднятии своего сервера для обмена сообщениями. С самодельными серверами и "не-попсовыми" протоколами всегда встает проблема "а с кем там общаться?", потому что перетащить прям вот вообще всех собеседников и контрагентов на что-то им непривычное практически нереально. Но если речь идет только, например, об инструменте общения внутри семьи, небольшой группы единомышленников или внутри одной компании, то это может быть вполне неплохим вариантом.

В наше время как self-hosted альтернативу популярным мессенджерам часто упоминают Matrix, например, с клиентом Element. На Хабре есть подробные инструкции, например вот эта: https://habr.com/ru/articles/837904/. Я попробовал, и мне не понравилось. Клиенты тормозные, через нестабильный интернет-канал все работает просто отвратительно, а сам сервер просто неповоротливый и укладывает не слишком богатый на процессор и память VPS даже всего с парой клиентов.

И тут мне вспомнилось про XMPP, он же Jabber. Он родом еще из тех времен, когда люди пользовались процессорами на 200-300 мегагерц и подключались к интернету через dial-up модемы - то есть он изначально очень нетребовательный к ресурсам. Между тем, развитие его не замерло на месте, а на сегодняшний день он умеет почти все то что требуется от современного мессенджера: хранение истории, передача файлов, аудио-видео звонки, end-to-end шифрование, и другое.

Я буду настраивать XMPP-сервер Prosody. Система у меня на сервере Ubuntu 22.04

Подключаем официальные deb-репозитории от проекта Prosody. Сам prosody есть в обычных репах Debian и Ubuntu, но там старые версии, а я советую использовать версию сервера не ниже 0.12, а в идеале версию 13.x, потому что в более старых версиях не хватает некоторых полезных фич, и некоторые модули не включены в поставку и их нужно устанавливать отдельно. Поэтому подключаем репы от разработчиков, чтобы иметь самые свежие версии:

```
wget https://prosody.im/downloads/repos/$(lsb_release -sc)/prosody.sources -
O/etc/apt/sources.list.d/prosody.sources

apt update
apt install prosody
```

У меня установилась версия 13.0.2. Она требует довольно свежего интерпретатора Lua (Prosody написан на Lua), и если при запуске сервиса оно ругается на интерпретатор, нужно доустановить и выбрать правильный:

```
update-alternatives --config lua-interpreter
# в появившемся списке выбираем версию 5.4
```

Далее нам нужен домен для работы сервера, XMPP работает через TLS со всеми вытекающими. В принципе, можно использовать и бесплатный домен с какого-нибудь dynu.com, freedns.afraid.org, или даже sslip.io, но тут все упирается в доверие к сервису - кто владеет доменом, тот имеет доступ ко всему (если только вы не сделаете certificate pinning). Параноики могут и без доменов использовать самосгенеренные сертификаты, вручную добавляя их в список доверенных - в prosody генерация самоподписанных сертификатов делается очень легко командой `prosodyctl cert generate habr.com` (либо используйте прям IP-адрес сервера вместо домена).

Окей, допустим мы используем домен habr.com. Нам нужно будет направить на IP-адрес вашего сервера сам habr.com, и еще сделать поддомен типа upload.habr.com (чуть позже расскажу зачем), который тоже будет указывать на IP сервера.

Далее делаем получение сертификатов через LetsEncrypt для вашего домена (если оно у вас уже есть, можете пропустить):

```
apt install certbot
certbot certonly --standalone -d habr.com -d upload.habr.com
```

Оно получит от LetsEncrypt сертификаты и настроит их автоматическое обновление.

Теперь настало время отредактировать конфиг сервера, он располагается обычно в `/etc/prosody/prosody.cfg.lua`.

В самом начале там идет список модулей, активированных на сервере. Я приведу то что в итоге получилось у меня, а комментарием -- ! отмечу те, что были отключены в конфиге по умолчанию и которые я включил специально:

```
modules_enabled = {
                "disco";
                "roster";
                "saslauth";
                "tls";                  "blocklist";
                "bookmarks";
                "carbons";
                "dialback";
                "limits";
                "pep";
                "private";
                "smacks";
                "vcard4";
                "vcard_legacy";                 "account_activity";
                "cloud_notify";
                "csi_simple";
                "invites";
                "invites_adhoc";
                "invites_register";
                "ping";
                "register";
                "time";
                "uptime";
                "version";                  "cloud_notify_extensions"; -- !
добавил
                "turn_external"; -- ! добавил
                "mam"; -- ! добавил
                "muc_mam"; ! -- тоже
                "csi_simple"; -- ! добавил
                "pubsub"; -- ! может быть полезным
                -- "http_file_share"; -- ! вот этот используется, но
закомментирован, чуть позже расскажу почему
                "admin_adhoc";
                "admin_shell";
}
```

Модуль `mod_cloud_notify_extensions` не входит в стандартную поставку, поэтому его надо будет доустановить отдельно:

```
apt install liblua5.4-dev
prosodyctl install --server=https://modules.prosody.im/rocks/
mod_cloud_notify_extensions
```

XMPP работает по принципу федерации - то есть пользователи вашего сервера могут коммуницировать с пользователями, подключенными к другим серверам, и наоборот. Если вам нужно сделать изолированный сервер, чтобы никто не заходил на него "со стороны", отключите модуль s2s:

```
modules_disabled = {
        "s2s"; -- Handle server-to-server connections
}
```

Также Prosody позволяет выбрать, как будут храниться данные пользователей (аккаунты, сообщения, и т.д.). По умолчанию используется хранилище на простых файлах. Можно выбрать чтобы использовалась база SQLite, это будет более эффективно. Для серверов с большим количеством пользователей и сообщений можно использовать MySQL и PostgreSQL.

Дальше скроллим ниже до раздела с `VirtualHost`. По умолчанию там прописан localhost, заменяем его на наш основной домен:

```
VirtualHost "habr.com"
```

Сохраняем конфиг, запускаем

```
prosodyctl --root cert import /etc/letsencrypt/live
```

Prosody автоматически найдет сертификаты для доменов и скопирует их к себе для использования. Чтобы они обновлялись автоматически, можно добавить хук, создав файл типа
`/etc/letsencrypt/renewal-hooks/deploy/prosody.sh`
с содержимым

```
#!/bin/sh
/usr/bin/prosodyctl --root cert import /etc/letsencrypt/live
```

и сделать его исполняемым: `chmod +x /etc/letsencrypt/renewal-hooks/deploy/prosody.sh`

В принципе, все почти готово. Мы настроили сервер со всеми необходимыми модулями, сертификатами, и прочим. И даже E2E-шифрование (OMEMO) будет работать из коробки, если оно поддерживается клиентами.
Но есть еще пара нюансов.

Первое - это передача файлов. Сам по себе XMPP предусматривает передачу файлов только peer to peer без участия сервера. Это требует чтобы оба клиента в момент передачи находились онлайн, да и в наши времена повсеместного NAT'а практически не работает.

Есть решение - использовать http_file_share модуль. Тогда при отправке файла через чат, клиент зальет его на сервер по HTTP, а собеседники получат ссылку на загруженный файл. URL представялет из себя огромный UUID, так что перебором собрать файлы на сервере не получится.
Добавим в конфиг следующий текст прямо под описанием вашего VirtualHost:

```
VirtualHost "habr.com"
disco_items = {
    { "upload.habr.com", "file sharing service" },
}
Component "upload.habr.com" "http_file_share"
http_file_share_expires_after = 31 * 24 * 60 * 60 -- сколько должны храниться
файлы на сервере, в моем примере 31 день
http_file_share_global_quota = 1024*1024*1024*10 -- ограничение объема хранилища
на сервере, в моем случае 10 гигабайт
```

Поскольку вы тут явно объявили "component", не надо перечислять "http_file_share" в списке модулей в начале конфига (у меня он там закомментирован), иначе сервер ругается.

Если по каким-то причинам вам не нравится идея с HTTP-ссылками, есть еще модуль `proxy_65` (https://prosody.im/doc/modules/mod_proxy65), который позволяет передавать файлы *через* сервер (используя его как прокси чтобы обойти проблему прямых подключений через NAT), но не сохраняя их на сервере.

Второе - это аудио/видео звонки. В принципе, того, что мы имеем в конфиге, уже достаточно для них. Для установления соединений между клиентами для аудио-видеозвноков в условиях невозможности прямых подключений из-за NAT'а клиенты будут использовать публичные STUN/TURN сервера.
Однако если вы параноик, или хочется ни от кого не зависеть, можно поднять свой TURN-сервер:

```
apt install coturn
```

Далее редактируем `/etc/turnserver.conf`:

```
realm=habr.com
use-auth-secret
static-auth-secret=verysecretsecret
```

Перезапускаем coturn: `systemctl restart coturn` и добавлем в конфиг Prosody *выше* описания вашего VirtualHost следущие строки:

```
turn_external_host = "habr.com"
turn_external_port = 3478
turn_external_secret = "verysecretsecret"
```

Естественно, coturn вы можете разместить на отдельном сервере, например сделав для него домен `turn.habr.com`.

И еще маленький штрих - настроить многопользовательские комнаты (типа как "группы" в Телеге).

Добавим

```
 Component "conference.habr.com" "muc"
```

под вашим VirtualHost.
По умолчанию комнаты можно создавать всем, но есть параметр

```
restrict_room_creation = false
```

с вариантами "false" (можно всем), "true" (можно только админам) или "local" (можно только пользователям вашего сервера, чужаки пришедшие по server2server в пролете).
Там же есть опция

```
component_admins_as_room_owners = true
```

автоматическая дающая админам сервера права владельца во всех комнатах,
и

```
max_history_messages = 2000
```

определяющая, сколько сообщений хранить в истории.

Вот теперь все готово.
Можно запустить проверку конфига командой `prosodyctl check` - утилита проверит, что все настроено правильно и даст советы, если что-то можно улучшить.

Не забываем после правки конфига перезапустить сервер: `systemctl restart prosody` и можно подключаться.

Как создавать новые аккаунты на сервер? В конфиге можно разрешить регистрацию для всех желающих через клиенты (с этим аккуратнее), а можно добавлять кого надо вручную:

```
prosodyctl adduser user@habr.com
```

Добавленного юзера можно также занести в конфиге в список админов сервера, что даст ему больше полномочий (управление юзерами, если клиент это поддерживает, ad-hoc-команды, просмотр статистики, и т.д.):

```
admins = { "user@habr.com" }
```

А теперь про клиенты.
С мобильными клиентами все довольно неплохо.
Для Android лучшим из них считается **Conversations**. В Google Play он стоит пару

долларов, но можно бесплатно установить его из F-Droid.
Из альтернатив мне еще понравился **Blabber.im**
Видеозвонки работают и там и там.

На iOS традиционно хорошими клиентами считаются **Siskin IM** и его форк **Snikket** (я честно говоря, разницы не заметил). Видеозвонки работают.
Еще иногда упоминают клиент ChatSecure, с ним осторожнее - он кажется в полумертвом состоянии, по крайней мере домен, который там используется для push-уведомлений уже не существует.

Под дестктоп же... Ну, там такое. Есть много старых "классических" клиентов типа Gajim, Pidjin, Psi, но они имеют интерфейс из 90-х и часто не поддерживают современные фичи.
Если у вас нет аллергии на Electron, советую **ConverseJS** - стильный и удобный интерфейс, умеет все что надо (кроме аудиозвонков), может запускаться как Electron-приложение, а можно поставить его на свой сервер и подключаться из браузера.

Из клиентов, в которых заявлена поддержка аудиовидеозвонков - **Movim** (тоже веб), и кто-то хвалил **Dino** (Linux, но есть неофициальная сборка под Windows), и **BeagleIM** под MacOS. **Kaidan** под Linux выглядит очень похоже на Телегу, но видеозвонки, кажется, не умеет.

И еще есть проприетарный **AstraChat**, который, как заявлено, умеет все, но скачать просто так с сайта разработчика его нельзя (требуется регистрация на корп. емайл), на гитхабе кто-то выкладывает бинарники, но это уже на ваш страх и риск.

И в заключение вопрос - а насколько все это дело устойчиво к блокировкам, если РКН решит как-то вычислить и заблокировать именно ваш сервер по какой-то причине?
XMPP по умолчанию работает по стандартному порту 5222. Можно поставить на 443 порт что-то типа `sslh`, за которым еще будет стоять безобидный веб-сервер, и коннектиться клиентами на порт 443 как будто это обычный HTTPS.
Веб-сервер для файлов у Prosody висит тоже на нестандартном порту и отдает страничку prosody при запросе с урлом /. Можно спрятать его за веб-сервером типа Nginx или Caddy, чтобы по / они отдавали свою страницу, и при 404 ошибке - тоже (ну и само собой слушали на 443 порту, в http_file_share модуле есть параметр, который сообщает пользователям куда надо стучаться).

Как еще более эффективное решение, существуют расширения XMPP для работы через чистый HTTP (модуль `bosh`) и вебсокеты (модуль `websocket`), что в теории позволяет полностью замаскироваться под обычный веб-сервер и даже работать через CDN. Но клиентов, умеющих подобное, мне пока не встретилось, если кто-то знает - напишите в комментариях.