

# Ejabberd - Installation und Einrichtung

Diese Seite beschreibt die Installation und Einrichtung des XMPP/Jabber-Servers [ejabberd](#) auf einem [Debian-Bullseye](#)-System. Wir empfehlen dir zusätzlich einen Blick auf die ausführlichen Blog-Artikel zu werfen:

- [How to move the office to real time IM on ejabberd](#)
- [Check ejabberd XMPP server useful configuration steps](#)
- [How to configure ejabberd to get 100% in XMPP compliance test](#)
- [How to set up ejabberd video & voice calling](#)
- [Anleitung von Kuetz Blog](#)

## Begriffe und Variablen

| Variable | Bedeutung                                        | Beispiel                        |
|----------|--------------------------------------------------|---------------------------------|
| DOMAIN   | Domain des Jabberservers                         | systemausfall.org               |
| JID      | Jabber-Identifizier, vollständige Jabber-Adresse | admin1@jabber.systemausfall.org |
| NAME     | Nutzername                                       | alice                           |
| HOST     | Jabber-Host                                      | ejabberd@jabber                 |
| PASSWORT | Ein Passwort                                     |                                 |
| VHOST    | Virtueller Jabber-Host                           | jabber.systemausfall.org        |

## Installation

In Debian Bullseye uster ist Version 21.01 von ejabberd enthalten. Meist gibt es über die Backports eine aktuellere Version. Zur Installation sind die folgenden Schritte notwendig:

```
apt install ejabberd ejabberd-contrib erlang-p1-mysql -t bullseye-backports
```

Damit wird auch die MariaDB-Datenbankanbindung installiert. Während der Installation wird ein Adminaccount eingerichtet.

## Konfiguration

Die Konfiguration findet über die Datei `/etc/ejabberd/ejabberd.yml` statt. Durch zahlreiche Kommentare ist die Datei gut erklärt - ansonsten hilft die umfangreiche [Doku](#) weiter. Ab und an wird die Syntax vereinfacht - ein Blick in die aktuelle [Beispiel-Datei](#) hilft hier weiter.

Zudem kannst du dir ein Beispiel an der Konfiguration von [Schokokeks](#) und [Trashserver](#) nehmen.

## DNS-Einträge

Über Service-Records (SRV) wird im Domain-Name-System (DNS) hinterlegt, auf welchem Port

bestimmte Dienste für Clients und Server erreichbar sind. Die Konferenzräume des XMPP-Servers stehen meist unter conference.DOMAIN zur Verfügung. Ein entsprechender DNS-Eintrag sollte angelegt werden. Entsprechende SRV- und TXT-Einträge sollten ebenfalls gesetzt werden ([siehe](#)):

| Eintrag | Host                                        | Ziel                                                                                                                    | Anmerkung               |
|---------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------|
| SRV     | _xmpp-client._tcp.jabber.systemausfall.org  | jabber.systemausfall.org:5222                                                                                           | Client-2-Server         |
| SRV     | _xmpp-server._tcp.jabber.systemausfall.org  | jabber.systemausfall.org:5269                                                                                           | Server-2-Server         |
| SRV     | _xmpps-client._tcp.jabber.systemausfall.org | jabber.systemausfall.org:443                                                                                            | Client-2-Server         |
| SRV     | _stun._tcp.jabber.systemausfall.org         | jabber.systemausfall.org:3478                                                                                           | STUN                    |
| SRV     | _stun._udp.jabber.systemausfall.org         | jabber.systemausfall.org:3478                                                                                           | STUN                    |
| SRV     | _stuns._tcp.jabber.systemausfall.org        | jabber.systemaufall.org:5349                                                                                            | STUN, TLS-verschlüsselt |
| SRV     | _turn._udp.jabber.systemausfall.org         | jabber.systemausfall.org:5349                                                                                           | TURN                    |
| SRV     | _turn._tcp.jabber.systemausfall.org         | jabber.systemausfall.org:5349                                                                                           | TURN                    |
| SRV     | _turns._tcp.jabber.systemausfall.org        | jabber.systemausfall.org:5349                                                                                           | TURN, TLS-verschlüsselt |
| TXT     | _xmppconnect.jabber.systemausfall.org       | _xmpp-client-xbosh= <a href="https://jabber.systemausfall.org:5280/bosh">https://jabber.systemausfall.org:5280/bosh</a> | Websocket               |
| CNAME   | conference.jabber.systemausfall.org         | jabber.systemausfall.org                                                                                                |                         |
| CNAME   | echo.jabber.systemausfall.org               | jabber.systemausfall.org                                                                                                |                         |
| CNAME   | irc.jabber.systemausfall.org                | jabber.systemausfall.org                                                                                                |                         |
| CNAME   | proxy.jabber.systemausfall.org              | jabber.systemausfall.org                                                                                                |                         |
| CNAME   | pubsub.jabber.systemausfall.org             | jabber.systemausfall.org                                                                                                |                         |
| CNAME   | upload.jabber.systemausfall.org             | jabber.systemausfall.org                                                                                                |                         |

## Firewall

Folgende Ports müssen in der Firewall geöffnet werden:

| Port        | Protokoll | Verwendung                                                                 |
|-------------|-----------|----------------------------------------------------------------------------|
| 3478        | TCP, UDP  | Für STUN                                                                   |
| 5222        | TCP       | Für Clients, die sich mit dem Server verbinden wollen                      |
| 5269        | TCP       | Für andere Server, die sich mit dem lokalen Server verbinden wollen        |
| 5280        | TCP       | Für den internen Webserver (Adminwebinterface, Registrierung von Accounts) |
| 5349        | TCP, UDP  | Für TURN                                                                   |
| 5443        | TCP       | Für den Dateiapload                                                        |
| 7777        | TCP       | Für Dateitransfers via Proxy                                               |
| 49152-65535 | UDP       | Für TURN                                                                   |

## Sicherheit erhöhen

Über [IM Observatory](#) lässt sich die Einrichtung des jeweiligen Servers nach Sicherheitsaspekten bewerten. Es ist sinnvoll, den eigenen Server dort einen Test zu unterziehen und die bemängelten Punkte zu bearbeiten. Hinweise zur Erhöhung der Sicherheit befinden sich auch im [Blog](#) von process one.

## Zertifikate von Let's Encrypt

Mit [Let's Encrypt](#) gibt es eine einfach zu nutzende Möglichkeit, anerkannte Zertifikate zu erzeugen. Zahlreiche Anleitungen erklären, wie das grundsätzlich geht. Dein Zertifikat sollte nicht nur auf die

XMPP-Domain, sondern gleichzeitig auch auf alle als CNAME angelegten Domainnamen ausgestellt werden. Sind die Zertifikate erzeugt, liegen sie bei Debian unter `/etc/letsencrypt/live`. Da das Verzeichnis `root:root` gehört, hat ejabberd darauf keinen Zugriff. Es bietet sich an, unter `/etc/ejabberd/` eine Zertifikatsdatei zu erstellen (Rechte `ejabberd:root`) und sie mit den Daten aus dem letsencrypt-Verzeichnis zu befüllen:

```
cat /etc/letsencrypt/live/domain.tld/privkey.pem
/etc/letsencrypt/live/domain.tld/fullchain.pem >> /etc/ejabberd/ejabberd.pem
```

Anschließend sollte ein entsprechender cron-job erstellt werden, der die Datei bei jeder Zertifikatserneuerung aktualisiert und die ejabberd-Konfiguration neu lädt.

Seit Version [17.11](#) unterstützt ejabberd das ACME-Protokoll. Folgende Angaben sind für die automatische Erstellung und Nutzung per `ejabberdctl` notwendig:

```
listen:
-
  port: 5281
  module: ejabberd_http

certfiles:
- "/data/ejabberd/ejabberd.pem"

acme:
  contact: "mailto:info@example.org"
  ca_url: "https://acme-v01.api.letsencrypt.org"
```

Anschließend muss der Webserver so konfiguriert werden, dass er Anfragen auf Port 5281 an ejabberd weiter leitet.

## Unsichere SSL-Varianten deaktivieren

Erneut die [Konfigurationsdatei](#) bearbeiten:

```
protocol_options:
- "no_tlsv1"

s2s_protocol_options:
- "no_tlsv1"
```

## Passworte als Hash speichern

Bei Verwendung der internen Mnesia-Datenbank werden die Passworte standardmäßig im Klartext gespeichert. Dies kann mit den folgenden Optionen in der Konfigurationsdatei geändert werden:

```
auth_method: internal
auth_password_format: scram
```

Bereits in Klartext gespeicherte Passworte werden beim nächsten Start von ejabberd umgewandelt.

## Sichere Passworte erzwingen

Damit User nicht zu einfache Passworte bei der Erstellung von neuen Konten verwenden, kann folgende Option im Abschnitt `mod-register` aktiviert werden:

```
password_strength: 32
```

## Spamschutz

XMPP-Spam ist lästig. Sofern für den eigenen Server die offene Anmeldung (In-band Registration) erlaubt ist, sollte zumindest eine Captcha-Abfrage eingerichtet sein, damit Bots nicht zahlreiche Accounts anlegen können. Die meisten Clients betten das Captcha dank [XEP-0158](#) bei der Registrierung direkt ein. Sollte der Client dies nicht unterstützen, wird zumindest eine Link angezeigt, um das Captcha über den Browser aufzurufen.

- Zur Erzeugung sind noch einige Pakete notwendig<sup>1)</sup>:

```
apt install imagemagick ghostscript --no-install-recommends
```

- Die Einrichtung erfolgt über `/etc/ejabberd/ejabberd.yml`
- Anzeigen der Captchas per URL:

```
listen:  
-  
  port: 5280  
  captcha: true
```

- Skript einbinden und Nutzung pro Minute beschränken:

```
captcha_cmd: "/usr/share/ejabberd/captcha.sh"  
captcha_host: "https://jabber.systemausfall.org"  
captcha_limit: 5
```

- Die Angabe des `captcha_host` kann, je nach Setup, auch nach dem Muster `host:port` ohne vorangestelltes `https` erfolgen, bspw. `jabber.systemausfall.org:5280`
- Anzeige der Captchas bei der In-band Registrierung durch den Client:

```
mod_register:  
  captcha_protected: true
```

Weitere Hinweise zur Spamreduzierung befinden sich im [Vortrag](#) von ProcessOne.

## Sprach- und Videochats

Ab Version 20.04 bietet Ejabberd alle Voraussetzungen für Sprach- und Videochats. Hinweise zur Einrichtung findest du auch [hier](#). Um es den XMPP-Clients leichter zu machen, solltest du die [DNS-Einträge](#) entsprechend erstellen und natürlich die Ports in der [Firewall](#) öffnen. In der einfachsten Variante sieht die Konfiguration dafür so aus:

```
-
  port: 3478
  transport: udp
  module: ejabberd_stun
  use_turn: true
  turn_ip: $DEINE_IP

-
  port: 5349
  transport: tcp
  module: ejabberd_stun
  use_turn: true
  turn_ip: $DEINE_IP
  tls: true

mod_stun_disco:
  services:
    -
      host: $DEINE_IP
      port: 3478
      type: stun
      transport: udp
    -
      host: $DEINE_IP
      port: 3478
      type: turn
      transport: tcp
    -
      host: $VHOST
      port: 5349
      type: stuns
      transport: udp
    -
      host: $VHOST
      port: 5349
      type: turns
      transport: tcp
```

Die Funktionalität kannst du anschließend [hier](#) oder mit [diesen](#) Hinweisen testen. Die Verbindungen sind immer per DTLS-SRTP Ende-zu-Ende verschlüsselt und sollten für geringere Latenz über UDP laufen. TLS-verschlüsselte Verbindungen über TCP erzeugen nur einen unnötigen Overhead ohne zusätzliche Sicherheit.

## Optimierung für mobile Clients

Neben klassischen Desktop-Programmen verbreitet XMPP sich auch auf verschiedenen Smartphone-Systemen. So gibt es mit [Conversations](#) bspw. es einen wirklich guten XMPP-Client für Android-Geräte. Damit die Nutzung mit unterschiedlichen Clients angenehmer wird, können einige Optimierungen vorgenommen werden.

- [BOSH](#) für die Nutzung hinter restriktiven Firewalls, die die üblichen XMPP-Ports blocken
- [Roster Versioning](#)

## HTTP File Upload

[HTTP File Upload](#) ermöglicht Dateitransfer per HTTP-Upload. Da ejabberd auf einem speziellen Port auf die Uploads wartet, muss die [Firewall](#) entsprechend konfiguriert sein. Anschließend wird `ejabberd.yml` angepasst:

- Im `listen`-Abschnitt:

```
-  
  port: 5443  
  module: ejabberd_http  
  tls: true  
  certfile: "/etc/ejabberd/ejabberd.pem"  
  request_handlers:  
    "upload": mod_http_upload
```

- In der Modulkonfiguration:

```
mod_http_upload:  
  docroot: "/date/ejabberd/upload"  
  put_url: "https://xmpp.example.org:5443/upload"  
  access: local
```

- Um übertragene Dateien nicht ewig zu speichern, sollte ein automatisches Löschintervall über das Modul `mod_http_upload_quota` festgelegt werden:

```
mod_http_upload_quota:  
  max_days: 60
```

## Tor Hiddenservice einrichten

Der XMPP-Server kann recht einfach als Tor Hiddenservice betrieben werden.

- Tor installieren:

```
apt install tor tor-nyx
```

- /etc/tor/torrc bearbeiten, Port des Dienstes eintragen und Verzeichnis festlegen:

```
HiddenServiceDir /var/lib/tor/hidden_service/  
HiddenServiceVersion 3  
HiddenServicePort 5222 127.0.0.1:5222  
HiddenServicePort 5269 127.0.0.1:5269
```

- Verzeichnis erstellen und Rechte setzen:

```
mkdir /var/lib/tor/hidden_service/  
  
chown -R debian-tor: /var/lib/tor/hidden_service/  
  
chmod 700 /var/lib/tor/hidden_service/
```

- Anschließend Tor neu starten. Mit dem Kommandozeile-Tool nix kannst du die Funktionalität deines Tor-Dienstes überprüfen.

In der Datei /var/lib/tor/hidden\_service/hostname ist der automatisch generierte .onion-Link hinterlegt, der nun als Adresse publiziert werden kann. Du solltest noch folgende [Hinweise](#) beachten.

## Administration

### Webfrontend

Über das Webfrontend kann der Server umfangreich konfiguriert werden. Es ist üblicherweise unter <https://DOMAIN/admin> zu erreichen. Einloggen können sich in /etc/ejabberd/ejabberd.yml eingetragene Admins mit ihrer vollständigen JID und dem entsprechenden Passwort.

### Neue User anlegen

Sofern das entsprechende Modul aktiviert ist, können User mit ihrem Client selbständig neue Accounts registrieren. Für Admins gibt es zwei Möglichkeiten:

- per Webfrontend
- oder per Kommandozeile

```
ejabberdctl register NAME HOST PASSWORT
```

### Userpasswort ändern

Auch hier gilt: sofern das entsprechende Modul aktiviert ist, können User mit ihrem Client das Passwort selbst ändern. Admins können dies über das Webfrontend tun: <https://DOMAIN/admin/server/VHOST/users>.

## Hinweise

- [Konfigurationsschnipsel für STUN/TURN](#)
- [Conversations Kompodium](#) – „Benutzerhandbuch“ für Conversations
- [Conversations Kurzanleitung](#)

<sup>1)</sup>  
Entgegen der Dokumentation ist das convert-Tool aus dem Debian-Paket graphicsmagick-imagemagick-compat nicht vollständig kompatibel zur imagemagick-Version

From: <https://howto.wikis.systemausfall.org/> - **Das HowTo-Wiki**

Permanent link: [https://howto.wikis.systemausfall.org/linux/ejabberd\\_-\\_installtion\\_und\\_einrichtung?rev=1668967476](https://howto.wikis.systemausfall.org/linux/ejabberd_-_installtion_und_einrichtung?rev=1668967476)

Last update: **2022/11/20 19:04**

