Installieren von Jitsi Meet unter Debian 10

Schritt 1 — Konfigurieren der Firewall

Wenn Sie dem Leitfaden Ersteinrichtung eines Servers mit Debian 10 gefolgt sind, haben Sie die UFW-Firewall aktiviert und den SSH-Port geöffnet. Der Jitsi-Server benötigt einige geöffnete Ports, damit er mit den Anrufclients kommunizieren kann. Außerdem ist der TLS-Installationsprozess auf einen geöffneten Port angewiesen, damit er die Zertifikatsanfrage authentifizieren kann.

Die Ports, die Sie öffnen werden, sind folgende:

- 443/tcp, der für die Webseite zur Erstellung von Konferenzräumen verwendet wird.
- 10000/udp, der zur Übertragung und zum Empfang des verschlüsselten Anrufverkehrs dient.

Der Server ist nun bereit für die Installation von Jitsi, die Sie im nächsten Schritt vornehmen werden.

Schritt 2 — Installieren von Jitsi Meet

In diesem Schritt fügen Sie Ihrem Server das Jitsi-Stable-Repository hinzu und installieren dann das Jitsi Meet-Paket aus diesem Repository. Dadurch wird sichergestellt, dass Sie stets das neueste stabile Jitsi Meet-Paket ausführen.

Installieren Sie zuerst das gnupg-Paket, das es dem System ermöglicht, kryptographische GPG Schlüssel zu verwalten:

```
$ apt-get install gnupg
```

Laden Sie zuerst den Jitsi-GPG-Schlüssel mit dem Download-Dienstprogramm wget herunter:

```
$ wget https://download.jitsi.org/jitsi-key.gpg.key
```

Der Paketmanager apt verwendet diesen GPG-Schlüssel zur Validierung der Pakete, die Sie aus dem Jitsi-Repository herunterladen werden.

Fügen Sie jetzt den GPG-Schlüssel hinzu, den Sie in den Keyring von apt über das Dienstprogramm apt-key heruntergeladen haben:

```
$ apt-key add jitsi-key.gpg.key
```

Sie können die GPG-Schlüsseldatei nun löschen, da sie nicht mehr benötigt wird:

```
$ rm jitsi-key.gpg.key
```

Nun fügen Sie Ihrem Server das Jitsi-Repository hinzu, indem Sie eine neue Quelldatei erstellen, die das Jitsi-Repository enthält. Öffnen und erstellen Sie die neue Datei mit Ihrem Editor:

```
$ vim /etc/apt/sources.list.d/jitsi-stable.list
```

Fügen Sie der Datei für das Jitsi-Repository folgende Zeile hinzu:

```
deb https://download.jitsi.org stable/
```

Speichern und schließen Sie Ihren Editor. Führen Sie abschließend eine Systemaktualisierung durch, um die Paketliste aus dem Jitsi-Repository zu erhalten, und installieren Sie dann das Paket jitsi-meet:

```
$ apt-get update
$ apt-get install jitsi-meet
```

Während der Installation von ***jitsi-meet*** werden Sie aufgefordert, den Domänennamen einzugeben (z. B. ***jitsi.your-domain***), den Sie für Ihre Jitsi Meet-Instanz verwenden möchten.

Dann sehen Sie ein neues Dialogfeld, in dem Sie gefragt werden, ob Jitsi ein selbstsigniertes TLS-Zertifikat erstellen und verwenden oder ein bestehendes Zertifikat, das bereits vorhanden ist, nutzen soll:

Wenn Sie über kein TLS-Zertifikat für Ihre Jitsi-Domäne verfügen, wählen Sie die erste Option Generate a new self-signed certificate (Generieren eines neuen selbstsignierten Zertifikats).

Ihre Jitsi Meet-Instanz wurde nun mit einem selbstsignierten TLS-Zertifikat installiert. Das führt dazu, dass Browserwarnungen angezeigt werden. Darum erhalten Sie im nächsten Schritt ein signiertes TLS-Zertifikat.

Schritt 3 — Sperren der Erstellung von Konferenzräumen

Im nächsten Schritt konfigurieren Sie Ihren Jitsi Meet-Server so, dass nur registrierte Benutzer Konferenzräume erstellen können. Die Dateien, die Sie bearbeiten werden, wurden vom Installer generiert und mit Ihrem Domänennamen konfiguriert.

Die Variable ***your_domain*** wird in den folgenden Beispielen anstelle eines Domänennamens verwendet.

Öffnen Sie zuerst ***/etc/prosody/conf.avail/your_domain.cfg.lua*** mit einem Texteditor:

```
$ /etc/prosody/conf.avail/your_domain.cfg.lua
```

Bearbeiten Sie diese Zeile:

```
...
        authentication = "anonymous"
...
```

Tun Sie Folgendes:

```
...
        authentication = "internal_plain"
...
```

Diese Konfiguration weist Jitsi Meet dazu an, eine Authentifizierung mit Benutzername und Passwort durchzusetzen, bevor neue Besucher einen Konferenzraum erstellen können.

Fügen Sie dann in der gleichen Datei folgenden Abschnitt am Ende der Datei hinzu:

```
...
VirtualHost "guest.your_domain"
    authentication = "anonymous"
    c2s_require_encryption = false
```

Diese Konfiguration ermöglicht es anonymen Benutzern, Konferenzräumen beizutreten, die von einem authentifizierten Benutzer erstellt wurden. Der Gast muss jedoch über eine eindeutige Adresse und ein optionales Passwort für das Betreten des Raums verfügen.

Hier haben Sie ***guest***. am Anfang Ihres Domänennamens hinzugefügt. Für ***jitsi.your-domain*** würden Sie zum Beispiel ***guest.jitsi.your-domain eingeben***. Den ***guest.***-Hostnamen nutzt Jitsi Meet ausschließlich intern. Sie werden ihn niemals in einen Browser eingeben oder einen DNS-Eintrag dafür erstellen müssen.

Öffnen Sie mit einem Texteditor eine weitere Konfigurationsdatei unter ***/etc/jitsi/meet/your_domain-config.js***:

```
$ vim /etc/jitsi/meet/your_domain-config.js
```

Bearbeiten Sie diese Zeile:

```
...
        // anonymousdomain: 'guest.example.com',
...
```

Tun Sie Folgendes:

```
...
        anonymousdomain: 'guest.your_domain',
...
```

Auch hier weist der Hostname ***guest.your_domain***, den Sie zuvor in dieser Konfiguration verwendet haben, Jitsi Meet an, welcher interne Hostname für die nicht authentifizierten Gäste verwendet werden soll. Öffnen Sie als Nächstes ***/etc/jitsi/jicofo/sip-communicator.properties***:

```
$ vim /etc/jitsi/jicofo/sip-communicator.properties
```

Und fügen Sie die folgende Zeile hinzu, um die Konfigurationsänderungen abzuschließen:

```
org.jitsi.jicofo.auth.URL=XMPP:your_domain
```

Diese Konfiguration verweist einen der Jitsi Meet-Prozesse auf den lokalen Server, der die jetzt obligatorische Benutzerauthentifizierung vornimmt.

Ihre Jitsi Meet-Instanz ist nun so konfiguriert, dass nur registrierte Benutzer Konferenzräume erstellen können. Nach der Erstellung eines Konferenzraums können ihm sowohl registrierte als auch nicht registrierte Benutzer beitreten. Sie benötigen lediglich die eindeutige Adresse des Konferenzraums und ein optionales Passwort, das ggf. vom Ersteller des Raums festgelegt wurde.

Nachdem Jitsi Meet nun so konfiguriert ist, dass nur authentifizierte Benutzer Räume erstellen können, müssen Sie diese Benutzer und ihre Passwörter registrieren. Dazu verwenden Sie das Dienstprogramm ***prosodyctl***.

Führen Sie den folgenden Befehl aus, um Ihrem Server einen Benutzer hinzuzufügen:

```
$ prosodyctl register user your_domain password
```

Der Benutzer, den Sie hier hinzufügen, ist ***kein*** Systembenutzer. Sie können lediglich einen Konferenzraum erstellen und sich nicht über SSH bei Ihrem Server anmelden.

Starten Sie abschließend die Jitsi Meet- und Nginx-Prozesse neu, um die neue Konfiguration zu laden:

```
$ systemctl restart prosody.service
$ systemctl restart jicofo.service
$ systemctl restart jitsi-videobridge2.service
$ systemctl restart nginx
```

Die Jitsi Meet-Instanz wird nun mit einem Dialogfeld einen Benutzernamen und ein Passwort anfordern, sobald ein Konferenzraum erstellt wird.

Ihr Jitsi Meet-Server ist nun eingerichtet und sicher konfiguriert.

https://www.kuketz-blog.de/jitsi-meet-optimierung-der-performance/

Schritt 4 — Optimierung der Performance

Mit ein paar kleinen Eingriffen in die Jitsi-Meet-Konfiguration könnt ihr die Leistung optimieren. Insbesondere Video-Konferenzen mit mehreren Teilnehmern profitieren von den Einstellungen.

Öffnet dazu die ***config.js-Datei*** von Jitsi Meet:

```
$ vim /etc/jitsi/meet/<domain>-config.js
```

[1] Festlegen der Standardsprache auf Deutsch

```
defaultLanguage: 'de',
```

[2] Reduzierung der Auflösung von 720 auf 480

```
resolution: 480,

constraints: {
   video: {
      aspectRatio: 16 / 9,
      height: {
         ideal: 480,
         max: 480,
         min: 240
      }
   }
},
```

Die Limitierung der Videoauflösung spart sowohl auf Client- als auch auf der Server-Seite Ressourcen.

[3] Limitierung der übertragenen Video-Feeds

```
channelLastN: 4,
```

Nur die Videodaten bzw. Streams der letzten vier aktiven Sprecher wird übermittelt. Alle anderen Teilnehmer werden sozusagen »eingefroren«, bis sie wieder sprechen. Weitere Infos zur Einstellung.

[4] Enable Layer Suspension

```
enableLayerSuspension: true,
```

Der Client (ab Chrome 69) sendet nur jene Streams, die zu einem bestimmten Zeitpunkt angesehen werden, wodurch der CPU- und Bandbreitenverbrauch sowohl auf der Client- als auch auf der Server-Seite reduziert und gleichzeitig die Videoqualität verbessert wird. Weitere Infos zur Einstellung.

[5] Videokonferenz nur mit Audio starten

```
startAudioOnly: true,
```

Video kann dann bei Bedarf aktiviert werden. Gerade wenn viele Teilnehmer einen Konferenzraum gleichzeitig betreten, sorgt das für eine deutliche Entlastung.

[6] Deaktivierung der blauen Audio-Dots beim Speaker

```
disableAudioLevels: true,
```

Reduziert die CPU-Auslastung bei den Clients.

[7] Externe Verbindungen zu gravatar.com und Co. unterbinden

Über ein Einstellungsmenü erlaubt Jitsi Meet den Teilnehmern ihr Profil bzw. ihren Namen anzupassen, der anschließend in der Konferenz angezeigt wird. Beim Klick auf die Einstellungen (Rädchen-Symbol) gelangt man in ein kleines Menü, wo man seine Geräte, Profil und Mehr (Sprache) konfigurieren kann. Unter ***Profil*** lässt sich eine E-Mail-Adresse für Gravatar eingeben. Eine Eingabe dort löst wiederum eine Verbindung zu gravatar.com aus:

```
https://www.gravatar.com/avatar/ee1bf01414ae7f3b3fe3d33a2aaf9480?d=404&size=200
```

Eine Verbindung zu einem Drittdienst wie gravatar.com würde ich persönlich gerne vermeiden. Das lässt sich über die Konfiguration wie folgt erledigen:

```
$ vim /etc/jitsi/meet/<domain>-config.js
```

```
disableThirdPartyRequests: true,
```

Danach sind die Third-Party-Requests deaktiviert. Davon nicht betroffen sind die Einstellungen der STUN- und TURN-Server.

[8] Logging reduzieren

Standardmäßig wird Jitsi Meet mit dem Logging-Level **INFO** ausgeliefert. In diesem Modus erfasst die Videobridge die IP-Adressen der Teilnehmer. Da ich diese Informationen nicht erfassen und speichern möchte, habe ich das Logging-Level auf **WARNING** reduziert:

```
$ vim /etc/jitsi/videobridge/logging.properties
```

```
.level=WARNING
```

[9] Keine »Werbung« für Apps mit Trackern

Der Versuch, die Tracker aus der iOS- und Android-Version von Jitsi Meet zu entfernen, war bisher leider nicht von Erfolg gekrönt. Das dazu passende Issue mit der Nummer 5799 wurde leider geschlossen, ohne eine weitere Diskussion zuzulassen. Aktuell ist ein datenschutzfreundlicher Betrieb mit der iOS- und Android-Version daher leider nicht möglich. Beide Versionen beinhalten drei Tracker:

- Google CrashLytics
- Google Firebase Analytics

- Amplitude

Es ist daher davon abzuraten, die Jitsi-Meet-App aus den offiziellen Stores von Apple und Google zu verwenden. Für Android gibt es allerdings Abhilfe. Die F-Droid-Version ist komplett ohne Tracker. Daher gilt die Empfehlung:

- Nutzt Jitsi Meet entweder ausschließlich am Rechner (via Browser)
- oder nutzt die F-Droid-Version auf Android

Aber auch Server-Betreiber haben Einfluss auf das Tracking bzw. können den Link anpassen, den Teilnehmer beim Aufruf über einen Browser auf ihrem Mobilgerät eingeblendet bekommen:

Dieser Link führt standardmäßig, je nach Geräte-Plattform (Android / iOS), zum Google Play oder Apple Store. Über einen Eingriff in der Konfiguration können Betreiber zumindest den Link für Android-Nutzer so anpassen, dass dieser auf die trackerfreie Version im F-Droid-Store verlinkt:

```
$ vim /usr/share/jitsi-meet/interface_config.js
```

```
SHOW_CHROME_EXTENSION_BANNER: false,

/**
* Specify custom URL for downloading android mobile app.
*/
MOBILE_DOWNLOAD_LINK_ANDROID:
'https://f-droid.org/en/packages/org.jitsi.meet/'
```

[10] STUN- bzw. TURN-Server

Das STUN-Protokoll erkennt Clients, die sich bspw. hinter einem Router oder einer Firewall befinden und eine NAT-Adresse haben. Mit Hilfe des STUN-Servers können NAT-Clients ihre öffentliche IP-Adresse erfahren und sind anschließend in der Lage eine direkte Kommunikationsverbindung zwischen (zwei) Teilnehmern herzustellen. Um die Übermittlung der IP-Adresse an externe Anbieter zu vermeiden, könnt ihr einen eigenen STUN- / TURN-Server betreiben. Alternativ könnt ihr natürlich auch einfach bestehende STUN-Server wählen, die öffentlich zur Verfügung gestellt werden.

Wer wissen möchte, welche »Probleme« ein STUN- / TURN-Server lösen kann, der kann das hier im Detail nachlesen: Funktionsweise – STUN/TURN Server.

[10.1] Anpassung der STUN-Server

In der Standardkonfiguration wurde Jitsi Meet noch bis vor einigen Wochen leider mit STUN-Servern von Google ausgeliefert. Als datenschutzsensibler Betreiber sollte man die STUN-Server der Instanz daher anpassen. Öffnet dazu die ***config.js-Datei***, die unter Debian im Pfad

```
# vim /etc/jitsi/meet/<domain>-config.js
```

zu finden ist. Dort sind folgende Einträge anzupassen:

```
stunServers: [
   { urls: 'stun:meet-jit-si-turnrelay.jitsi.net:443' }
],
```

Die Google STUN-Server solltet ihr mit einem anderen öffentlichen STUN-Server ersetzen. Funktionierende STUN-Server sind bspw.:

- stun.1und1.de:3478
- stun.t-online.de:3478
- stun.nextcloud.com:443

Zwei weitere Einstellungen sind sinnvoll, allerdings optional:

Sobald die Videobridge startet ermittelt sie ihre öffentliche IP-Adresse via STUN-Server. Das ist immer dann sinnvoll, wenn die Videobridge in einem NAT-Setup konfiguriert ist. Die zweite Zeile deaktiviert den TCP-Harvester der Videobridge und ausschließlich der TURN-Server wird anschließend für TCP-Verbindungen genutzt.

```
$ vim /etc/jitsi/videobridge/sip-communicator.properties
```

```
org.ice4j.ice.harvest.STUN_MAPPING_HARVESTER_ADDRESSES=meet-jit-si-
turnrelay.jitsi.net:443
org.jitsi.videobridge.DISABLE_TCP_HARVESTER=true
```

Nochmal alle Dienste durchstarten:

```
$ systemctl restart jicofo
$ systemctl restart prosody
$ systemctl restart jitsi-videobridge2
```

[11] Datenschutzerklärung

Jede Jitsi-Meet-Instanz sollte selbstverständlich auch über eine Datenschutzerklärung verfügen. Als Vorlage könnt ihr meine nehmen. Nur eine Bitte: Nicht 1:1 kopieren, sondern abwandeln und vielleicht auch eine eigene Struktur reinbringen bzw. an eure Bedürfnisse anpassen.

Damit ihr Links im HTML-Footer der Jitsi-Meet-Willkommensseite setzen könnt, solltet ihr folgende Anpassung an der CSS-Datei von Jitsi Meet vornehmen. Einfach ganz unten in die Datei einfügen und speichern:

```
$ vim /usr/share/jitsi-meet/css/all.css
```

```
.welcome .welcome-watermark{position:absolute;width:100%;height:auto}
#footer{margin-top:20px;margin-bottom:20px;font-size:14px}
```

Anschließend muss der Text im Footer bzw. die Links noch eingebunden werden:

```
$ vim /usr/share/jitsi-meet/static/welcomePageAdditionalContent.html
```

```
<template id = "welcome-page-additional-content-template">
   <div id="footer">
      <center>Betrieben von Kuketz-Blog | <a
href="https://www.kuketz-blog.de/impressum/">Impressum</a> | <a
href="https://www.kuketz-blog.de/datenschutzhinweis-kuketz-meet-de/">Datensc
hutzhinweis</a> | <a
href="https://www.kuketz-blog.de/jitsi-meet-erste-hilfe-bei-problemen/">Erst
e Hilfe bei Problemen</a></center>
      <center>Diese Jitsi-Instanz ist <a
href="https://www.kuketz-blog.de/jitsi-meet-server-einstellungen-fuer-einen-
datenschutzfreundlichen-betrieb/">datenschutzfreundlich</a> und nutzt
<strong>nicht</strong> die Google STUN-Server.</center>
   </div>
</template>
```

Danach genügt es, den nginx bzw. Webserver einmal neu zu initialisieren:

```
$ systemctl reload nginx
```

[12] - Anpassen des Nginx Webservers

nginx Konfiguration

[13] - weitere Anpassungen des sip-communicator

sip-communicator Konfiguration

14 - Jitsi Meet: Erste Hilfe bei Problemen

Jitsi Meet: Erste Hilfe bei Problemen