

# NVS PS HÜ 2: Jitsi Meet

Abgabe von David Pape (01634454)

## Umgebung

Ich habe einen ehemals ausrangierten 2012 Mac Mini als Heimserver umfunktioniert. Dieser hostet u.A. bereits einen Musik-Player und einen Fileserver, entsprechend war es naheliegend, ihn dann auch statt einer VM als Host für Jitsi Meet zu nutzen. Der Mac Mini hängt headless an meinem lokalen Netzwerk und wird über das Screen Sharing-Feature von macOS und/oder SSH bedient.

Der Vollständigkeit halber sei hier noch wiedergegeben, wie eine Debian-VM z.B. mit **QEMU** aufzusetzen wäre:

```
qemu-img create -f qcow2 debian.qcow2 32G
curl -O https://cdimage.debian.org/debian-cd/current/amd64/[...]
qemu-system-x86_64 \
  -m 4G \
  -vga virtio \
  -display default,show-cursor=on \
  -usb \
  -device usb-tablet \
  -machine type=q35,accel=hvf \
  -smp 2 \
  -cdrom debian-10.8.0-amd64-netinst.iso \
  -drive file=debian.qcow2,if=virtio \
  -cpu host \
```

## Installation

Um die ~~Faulheit~~ Zeiteffizienz zu maximieren, habe ich mich für die **Docker-Version von Jitsi** entschieden. Unter der Annahme, dass Docker bereits installiert ist, sollten theoretisch nur folgende Befehle vonnöten sein:

```
git clone https://github.com/jitsi/docker-jitsi-meet
cd docker-jitsi-meet
cp env.example .env
./gen-passwords.sh
mkdir -p ~/.jitsi-meet-cfg/{web/letsencrypt,transcripts, \
  prosody/config,prosody/prosody-plugins-custom,jicofo,jvb, \
  jigasi,jibri}
```

An dieser Stelle muss noch die Datei `.env` konfiguriert werden. Ich habe folgende Einträge angepasst:

```
# Directory where all configuration will be stored
CONFIG=~/.jitsi-meet-cfg
```

```
# Exposed HTTP port
HTTP_PORT=8420
```

```
# Exposed HTTPS port
HTTPS_PORT=8421
```

```
# System time zone
TZ="Europe/Amsterdam"
```

```
# Public URL for the web service (required)
PUBLIC_URL=https://192.168.8.117:8421
```

```
# IP address of the Docker host
DOCKER_HOST_ADDRESS=192.168.8.117
```

Schlussendlich startet man Jitsi mit dem Kommando `docker-compose up -d`.

```

# ariez @ mini in ~/docker-jitsi-meet on git:master o [18:33:49]
$ docker-compose up -d
Creating network "docker-jitsi-meet_meet.jitsi" with the default driver
Pulling web (jitsi/web:latest)...
latest: Pulling from jitsi/web
a076a628af6f: Pull complete
d8bbdc3034e9: Pull complete
c9c66a2abff9: Pull complete
6efafe2980ef: Pull complete
38b1dca54dff: Pull complete
e3607b8e8e57: Pull complete
0356d3e1d1b9: Pull complete
Digest: sha256:018f7407c2514b5eeb27f4bc4d887ae4cd38d8446a0958c5ca9cee3fa811f575
Status: Downloaded newer image for jitsi/web:latest
Pulling prosody (jitsi/prosody:latest)...
latest: Pulling from jitsi/prosody
a076a628af6f: Already exists
d8bbdc3034e9: Already exists
c9c66a2abff9: Already exists
6efafe2980ef: Already exists
76f4aac0fe14: Pull complete
e9cc1d87805f: Pull complete
0575f9c1efab: Pull complete
956b3e545ec5: Pull complete
86661edb39e5: Pull complete
fe714c744b7e: Pull complete
Digest: sha256:fe12d7c610f890d1050c9c41e7a5316a26182d59336b69162aae8ac8a61c1770
Status: Downloaded newer image for jitsi/prosody:latest
Pulling jicofo (jitsi/jicofo:latest)...
latest: Pulling from jitsi/jicofo
a076a628af6f: Already exists
d8bbdc3034e9: Already exists
c9c66a2abff9: Already exists
6efafe2980ef: Already exists
bbf9a6408ba0: Pull complete
1689b97804a4: Pull complete
0e642aa926bd: Pull complete
Digest: sha256:cf0d686a0f0f88361ec94576480f2d35b973d24383527715f6da2c7812790885
Status: Downloaded newer image for jitsi/jicofo:latest
Pulling jvb (jitsi/jvb:latest)...
latest: Pulling from jitsi/jvb
a076a628af6f: Already exists
d8bbdc3034e9: Already exists
c9c66a2abff9: Already exists
6efafe2980ef: Already exists
bbf9a6408ba0: Already exists
fd0890df9577: Pull complete
577324bdc9d3: Pull complete
Digest: sha256:7be4149dc43463bf7be85259ebfa3f5504eaf2ff508a566a2552936c29df5d00
Status: Downloaded newer image for jitsi/jvb:latest
Creating docker-jitsi-meet_prosody_1 ... done
Creating docker-jitsi-meet_web_1 ... done
Creating docker-jitsi-meet_jicofo_1 ... done
Creating docker-jitsi-meet_jvb_1 ... done

```

Figure 1: Starten via docker-compose

## Testlauf

Leider war damit tatsächlich nur *theoretisch* der Setup-Prozess abgeschlossen. In der Praxis lieferte das Docker-Image bei einem Testlauf nur einen Connection Error.

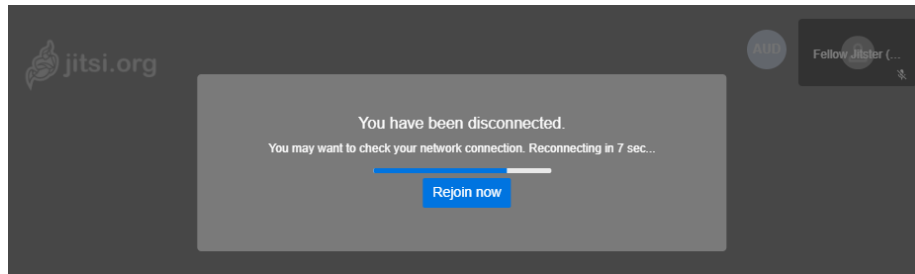


Figure 2: Testlauf

Eine Recherche lieferte den **Tipp**, ein älteres Release des Docker-Images zu verwenden. Nach einem `git checkout stable-5076` mit anschließendem Wiederholen der Konfiguration funktionierte Jitsi dann schlussendlich ohne Probleme.

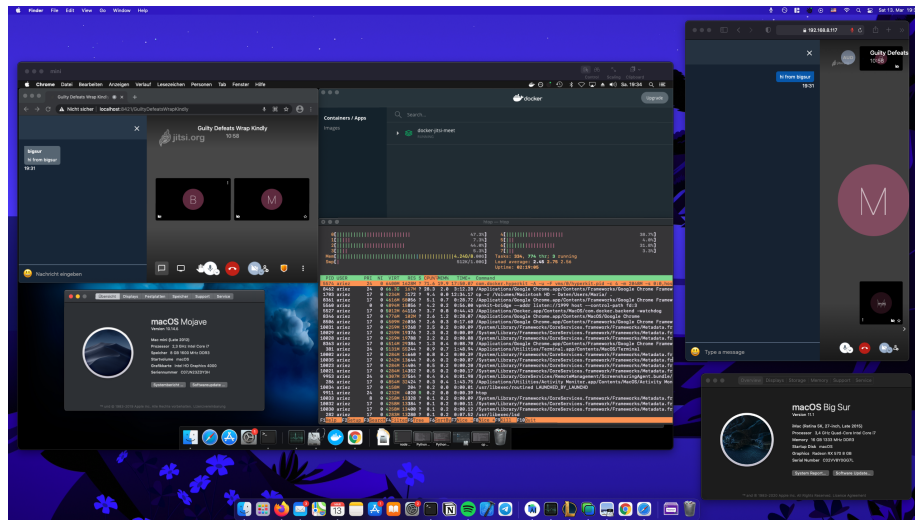


Figure 3: Endergebnis