

Setting up a Jitsi Meet server
on a virtual machine or on a Raspberry Pi 4

Bernd Braemer

November 14, 2020

Contents

Introduction	3
Why run and self-host a Jitsi Meet server?	3
Requirements to host the software	4
Creating the VM	5
Changing server settings to make the VM accessible from the internet	8
Installation of Jitsi Meet	9
Using a “Let’s Encrypt” certificate	11
Additional steps needed for Raspberry Pi only	11
Access Control	13
Creating moderators	15
How to update the installation?	16
Clients	16

Introduction

Jitsi is a collection of free and open-source multi-platform voice, videoconferencing and instant messaging applications for the web platform, Windows, Linux, macOS, iOS and Android.

Why I wanted to use a videoconferencing software?

- Due to COVID-19 my wife and I are not able to travel to see friends and parents in Germany or in the US.
- The group of people I virtually meet with online via our flight simulator needed a solution to talk to each other during group meetings/flights.

Why run and self-host a Jitsi Meet server?

Tests have shown that the voice quality over the official Jitsi servers becomes too poor when there are more than two participants in a conference. Technically Jitsi stops using a bi-directional connection with more than two participants. Data privacy is another aspect why it might be interesting to use a self-hosted Jitsi instance.

Requirements to host the software

- Red Hat Virtual Machine Manager also known as virt-manager.
- As an alternative to a VM a Raspberry Pi 4 can be used.
- An ISO image of Ubuntu Server 20.04. Other Linux distributions are available but not covered in this presentation.



Figure 1: About info of the Virtual Machine Manager

Installation:

```
sudo apt-get install qemu qemu-kvm libvirt-bin bridge-utils  
↪ virt-manager
```

Creating the VM

Choose create a new virtual machine

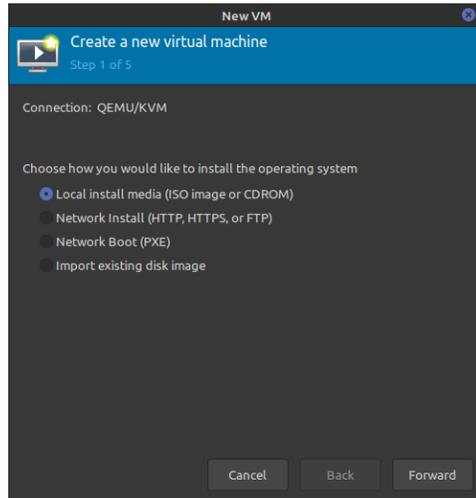


Figure 2: Create a new VM dialogue

Choose an ISO image or a CDROM

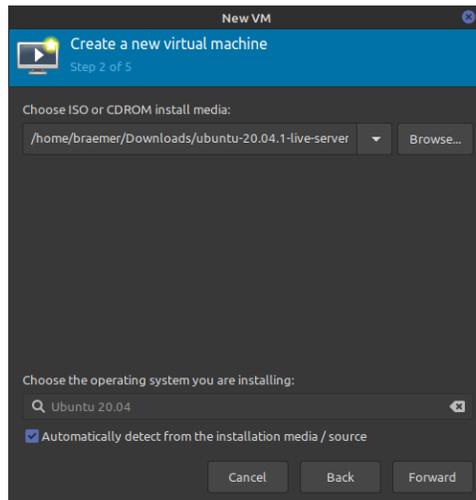


Figure 3: Choose a boot medium

Choose the memory and CPU settings

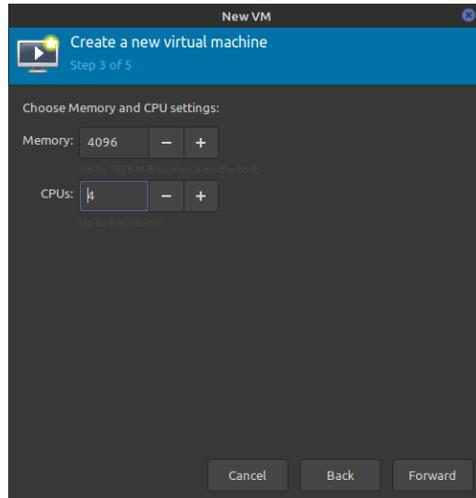


Figure 4: Choose memory and CPU

Select storage options

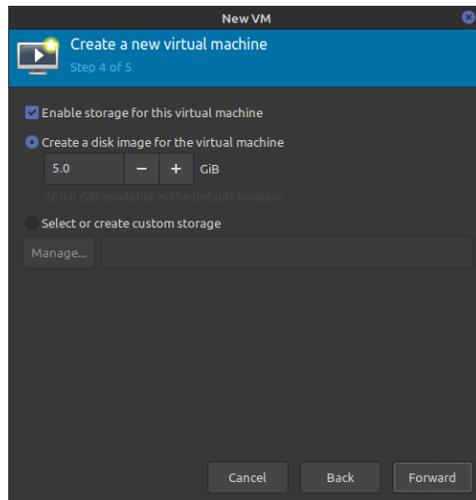


Figure 5: Select the storage for the virtual machine

Begin the installation

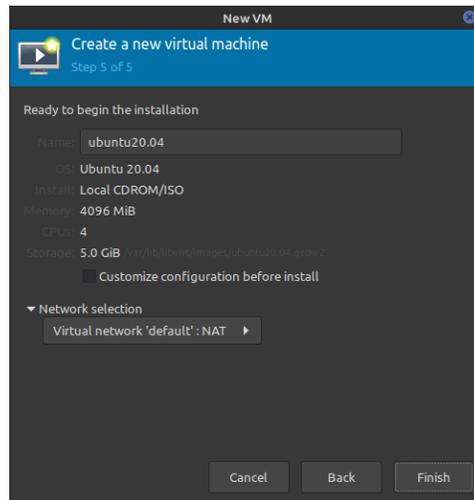


Figure 6: Begin the installation

The standard Ubuntu Server installation is executed. This is not part of this presentation.

Changing server settings to make the VM accessible from the internet

Virtual Network Interface

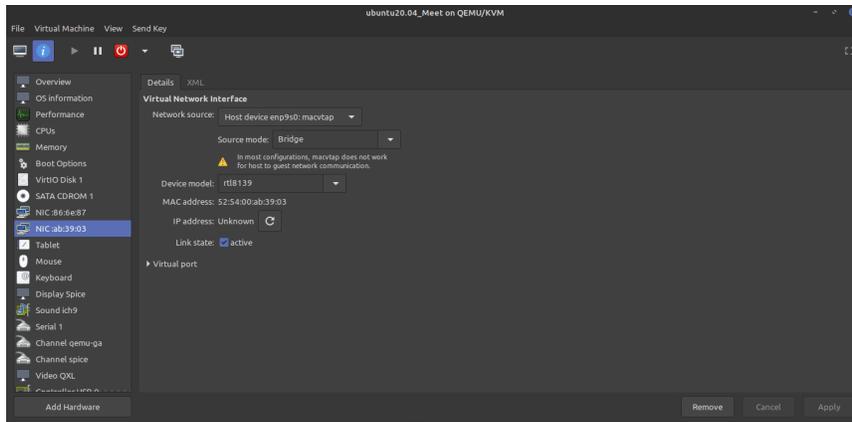


Figure 7: Network settings

Netplan within the VM. For the IP address DHCP is used and a static IP address is provided by the router.

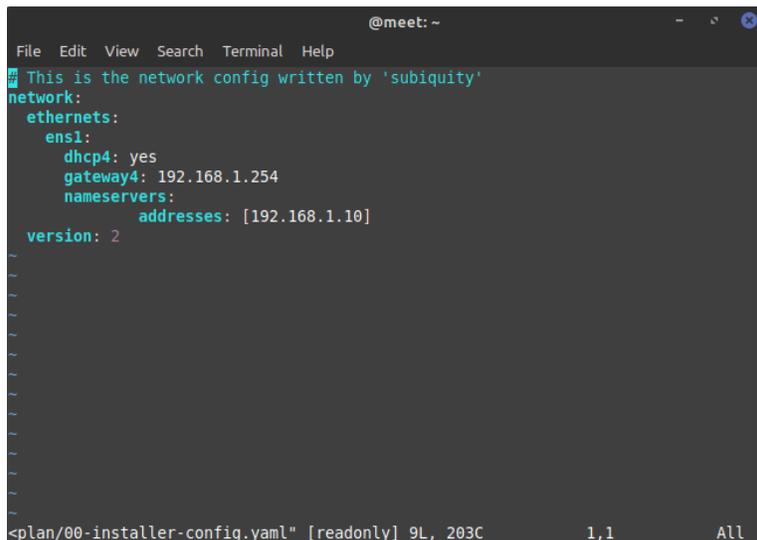


Figure 8: Netplan IP settings

Installation of Jitsi Meet

The following commands need to be executed in a Bash shell.

```
sudo echo 'deb https://download.jitsi.org stable/' | sudo tee
↳ /etc/apt/sources.list.d/jitsi-stable.list
sudo wget -qO - https://download.jitsi.org/jitsi-key.gpg.key
↳ | sudo apt-key add -
sudo apt install apt-transport-https
sudo apt update
sudo apt install jitsi-meet
```

The FDQN (not the hostname) must be entered during the installation.

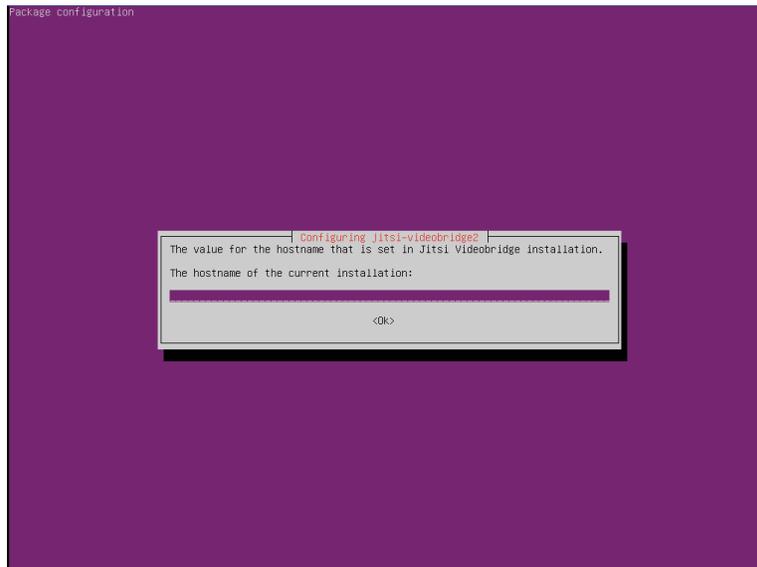


Figure 9: Enter hostname

Options for certificates. For this presentation the first option to generate a self-signed certificate is used.

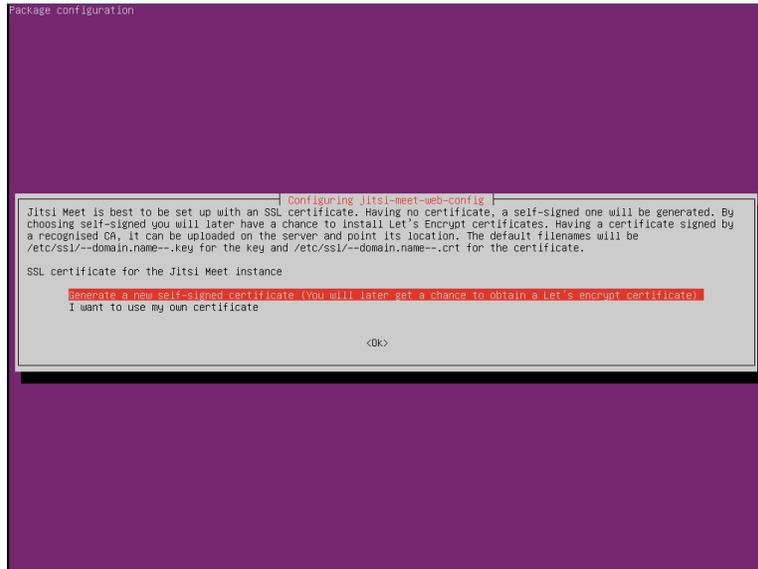


Figure 10: SSH certificate configuration

Tips:

Using a self-signed certificate will result in warnings being shown in browsers, because they cannot verify the server's identity.

Jitsi Meet mobile apps require a valid certificate signed by a trusted Certificate Authority and will not be able to connect to a server if a self-signed certificate was chosen.

The Jitsi Meet installation itself is completed here.

Using a “Let’s Encrypt” certificate

A Bash script is provided that can be executed to receive and use a “Let’s Encrypt” certificate.

```
sudo  
↪ /usr/share/jitsi-meet/scripts/install-letsencrypt-cert.sh
```

Additional steps needed for Raspberry Pi only

Unfortunately, the Jitsi Server on a Raspberry Pi does not work with multiple clients and ends the connection with an error message.

To run Jitsi Meet successfully on a Raspberry Pi, we need to run some extra commands.

Installation of openjdk-8 and building the requirements.

```
sudo apt install openjdk-8-jdk build-essential libtool maven
```

If another Java version is installed, follow the next step first.

```
sudo update-alternatives --config java  
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64/
```

Stop the Jitsi services.

```
sudo systemctl stop prosody jitsi-videobridge2 jicofo
```

Compilation of jniwrapper-native-1.0-SNAPSHOT.

```
git clone https://github.com/sctplab/usrctp.git  
git clone https://github.com/jitsi/jitsi-sctp  
mv ./usrctp ./jitsi-sctp/usrctp/  
cd ./jitsi-sctp  
mvn package -DbuildSctp -DbuildNativeWrapper  
↪ -DdeployNewJnilib -DskipTests
```

Copy libjnisctp.so.

```
cp ./jniwrapper/native/target/libjnisctp-linux-aarch64.so \  
./jniwrapper/native/src/main/resources/lib/linux/  
↪ libjnisctp.so
```

Re-package and Copy jniwrapper-native-1.0-SNAPSHOT.jar into Jitsi Video-Bridge2.

```
mvn package  
sudo cp ./jniwrapper/native/target/  
↪ jniwrapper-native-1.0-SNAPSHOT.jar \  
/usr/share/jitsi-videobridge/lib/  
↪ jniwrapper-native-1.0-SNAPSHOT.jar
```

Restart the services.

```
sudo systemctl start prosody jitsi-videobridge2 jicofo
```

Check if it worked and the video bridge is up.

```
systemctl status jitsi-videobridge2
```

Access Control

One disadvantage to this standard installation is that anybody who knows the URL can create a new meeting room.

The Prosody XMPP server is the Jitsi Meet component responsible for user management or authentication.

```
vim /etc/prosody/conf.d/YourDomain.cfg.lua
```

The following part needs to be changed from

```
VirtualHost "YourDomain"  
    -- enabled = false -- Remove this line to enable this  
    ↪ host  
    authentication = "anonymous"
```

to

```
VirtualHost "YourDomain"  
    -- enabled = false -- Remove this line to enable this  
    ↪ host  
    -- authentication = "anonymous"  
    authentication = "internal_hashed"
```

A new virtual host definition at the end of the same file needs to be added. The domain of the guest VirtualHost is internal only. It does not require any DNS record or SSL certificate.

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

The primary VirtualHost of the Jitsi instance will require any kind of authentication to create a conference meeting room whereas the VirtualHost for guests still grants access to anonymous users from now on.

The next step is to add the guest domain to Jitsi Meet frontend.

```
vim /etc/jitsi/meet/YourDomain-config.js
```

The directive `anonymousdomain` has to be added to the `hosts` object.

```
hosts: {
    // XMPP domain.
    domain: 'YourDomain',

    // When using authentication, domain for guest users.
    // anonymousdomain: 'guest.example.com',
    anonymousdomain: 'guest.YourDomain',
}
```

The Jitsi Conference Focus (`jicofo`) component needs to be configured to allow requests from an authenticated domain only. For that the protected URL needs to be added to the properties files.

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

Add the following key-value pair at the end of the file.

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

Restart all involved services.

```
service prosody restart
service jicofo restart
service jitsi-videobridge2 restart
service nginx restart
```

Should you come across some unexpected issues always have a look at the log files first. Here is a brief overview of where to check.

```
# Prosody
tail -f /var/log/prosody/prosody.log

# Jicofo
tail -f /var/log/jitsi/jicofo.log

# Jitsi video bridge
tail -f /var/log/jitsi/jvb.log

# nginx
tail -f /var/log/nginx/error.log
```

Creating moderators

The last step is to create moderators. Only moderators can create new meeting rooms.

Parameters to create a moderator can be obtained by running the following command:

```
prosodyctl register --help
Usage: /usr/bin/prosodyctl register USER HOST [PASSWORD]
Register a user on the server, with the given password
```

Example:

```
prosodyctl register UserName \$(hostname -f)
↪ VerySecretPassword
```

Note for Europeans

Unfortunately, this is not GDPR-compliant, because “enabling users to set their password without the admin knowing it” is a basic and unavoidable security measure.

How to update the installation?

The server can be updated via the apt command. On a Raspberry Pi the additional steps above need to be repeated.

In case the server is accessible from the internet, a cronjob can be configured to update the certificate when this is necessary.

```
# Renew certificates  
0 1 * * * /usr/bin/certbot renew
```

Clients

Jitsi Meet clients are available for iOS, Android, Windows macOS X, Ubuntu, Debian, RPM based distros and Arch Linux. A modern web browser is also working very well.