

Hosting and installing a server JitsiMeet

Difficulté



Moyen

This server provides you your own system for video conferencing. It should be available at URL "<https://jitsi.mydomain.tld>" and use a valid SSL certificate.

Installing the container

Installation of a container "Debian 10" on the Proxmox. From a terminal of the Proxmox:

```
# pveam available      ### to find the right template
# pveam download local debian-10.0-standard_10.0-1_amd64.tar.gz      ### in
my case
```

The template is stored under "local".

The rest of the installation is done from the GUI. I set 2 Cores and 4GB RAM.

The next step (not mandatory) is to activate SSH on the container (more comfortable than using the Console of the Proxmox). ⇒ you know what to do....

Installation of JitsiMeet

Following these tutos:

<https://github.com/jitsi/jitsi-meet/blob/master/doc/quick-install.md>

<https://www.scaleway.com/en/docs/setting-up-jitsi-meet-videoconferencing-on-debian-buster/> and

<https://community.nethserver.org/t/deploying-jitsi-meet-on-nethserver-independent-video-conferencing/15051/9>

- The hostname "**jitsi**" is given by the container name set in Proxmox.

```
# apt update && apt upgrade
# sed -i 's/^127.0.1.1.*$/127.0.1.1 jitsi.mydomain.tld jitsi/g'
/etc/hosts
# sed -i 's/^127.0.0.1.*$/127.0.0.1 localhost jitsi.mydomain.tld
jitsi/g' /etc/hosts
```

- no installation of nginx nor apache

```
# wget -qO - https://download.jitsi.org/jitsi-key.gpg.key | apt-key add
-
# sh -c "echo 'deb https://download.jitsi.org stable/' >
/etc/apt/sources.list.d/jitsi-stable.list"
# apt update
```

- `# apt-get -y --no-install-recommends install jitsi-meet`

Settings of the internal network

The jitsi machine must be reachable at ports 80 TCP, 443 TCP and 10000 UDP.

In my case, the container “jitsi” is placed into the DMZ, behind a firewall.

My main web server is a Nethserver placed into the DMZ too. The firewall forwards all http/https request to this Nethserver (NAT).

Settings for firewall/DNS

- port opening and forwarding for 10000 UDP to the jitsi
- into the DNS-resolver:
 - jitsi.mydomain.tld ⇒ as an alias of the Nethserver
 - container-jitsi.mydomain.tld ⇒ the IP of the container (I use it for SSH connections with the container)

Settings of the ReverseProxy

The webserver Nethserver acts as [ReverseProxy](#) to redirect http/https requests “jitsi.domain.tld” to the container.

This will allows to use the valid SSL certificate of the Nethserver for “jitsi.domain.tld” without copying the certificate on the jitsi machine.

- Get the Letsencrypt certificate covering the subdomain “jitsi.domain.tld” by the nethserver and use it as default certificate.
- Settings of the ReverseProxy:
 - URL: [https://IP](#) of the jitsi container
 - Certificate SSL/TLS: default
 - Accept non valid SSL certificate from the target: ticked
 - Forward the name of the host to the target: ticked

Enabling the authentication

Following this tuto: <https://crosstalksolutions.com/how-to-enable-jitsi-server-authentication/>

Settings for using with a dynamic IP

Target: the current external IP must be present into the conf file in order that Jitsi runs correctly. Following steps are therefore necessary by using a dynamic IP:

- a script in order to compare the current external IP with the IP present into the conf file and to replace it if it has changed since last IP-check
- run the script at bootup
- run the script regularly

Finding the current external IP and enter it into the conf file

(this script comes from an internet forum!)

```
# nano /etc/init.d/script_IP.sh
```

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          Nom du script
# Required-Start:    $local_fs $network
# Required-Stop:     $local_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Description courte
# Description:       Description longue
### END INIT INFO

DNSNAME="jitsi.domain.tld"          ##### adjust according
your settings

# get the actual IP from the Internet
IPint=$(host -tA $DNSNAME 8.8.8.8 | grep address | cut -d " " -f4 )

# get the configured IP of Jitsi
IPjitsi=$(grep 'NAT_HARVESTER_PUBLIC_ADDRESS' /etc/jitsi/videobridge/sip-communicator.properties | grep -oE "[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+")

if [ "$IPjitsi" == "$IPint" ]
then
    echo "IP has not been changed!"
    exit 0
fi

#clear config
sed -i '/NAT_HARVESTER_PUBLIC_ADDRESS/d' /etc/jitsi/videobridge/sip-
```

```
communicator.properties
```

```
#get IP and renew line  
echo org.ice4j.ice.harvest.NAT_HARVESTER_PUBLIC_ADDRESS=$IPint >>  
/etc/jitsi/videobridge/sip-communicator.properties
```

```
#restart services  
systemctl restart jicofo  
systemctl restart prosody  
systemctl restart jitsi-videobridge2
```

```
# chmod +x /etc/init.d/script_IP.sh
```

run the script at bootup

Source:

<https://www.jbnet.fr/systeme/linux/debian-executer-un-script-au-demarrage-de-la-machine.html>

```
# cd /etc/init.d  
# update-rc.d script_IP.sh defaults
```

run the script every hour

```
# nano /etc/cron.d/IP_jitsi
```

```
0 */1 * * * root /etc/init.d/script_IP.sh
```



When the script runs, it happens that Jitsi get interrupted (even if the IP is still current)
⇒ I run it only every hour to avoid frequent breakdowns of the communication. As the IP changes only 1x per day I think this is sufficient

From:
<https://wiki.guedel.eu/> - Wiki-Guedel

Permanent link:
https://wiki.guedel.eu/doku.php?id=welcome:self_hosting:installing_a_server_jitsimeet&rev=1593977172

Last update: 2020/07/05 19:26

