Quick Jitsi-Meet installation on a Debian-based GNU/Linux system

Self-Hosting Guide - Debian/Ubuntu server

Follow these steps for a quick Jitsi-Meet installation on a Debian-based GNU/Linux system. The following distributions are supported out-of-the-box:

Debian 11 (Bullseye) or newer

Ubuntu 22.04 (Jammy Jellyfish) or newer

note

Many of the installation steps require root or sudo access. So it's recommended to have sudo/root access to your system.

Required packages and repository updates

You will need the following packages:

gnupg2

nginx-full

sudo => Only needed if you use sudo

curl => Or wget to Add the Jitsi package repository

Note

OpenJDK 17 must be used.

Make sure your system is up-to-date and required packages are installed:

Run as root or with sudo:

Retrieve the latest package versions across all repositories

sudo apt update

Ensure support for apt repositories served via HTTPS

sudo apt install apt-transport-https

On Ubuntu systems, Jitsi requires dependencies from Ubuntu's universe package repository. To ensure this is enabled, run this command:

sudo apt-add-repository universe

Retrieve the latest package versions across all repositories:

sudo apt update

Install Jitsi Meet

Domain of your server and set up DNS

Decide what domain your server will use. For example, meet.example.org.

Set a DNS A record for that domain, using:

your server's public IP address, if it has its own public IP; or

the public IP address of your router, if your server has a private (RFC1918) IP address (e.g. 192.168.1.2) and connects through your router via Network Address Translation (NAT).

If your computer/server or router has a dynamic IP address (the IP address changes constantly), you can use a dynamic dns-service instead. Example DuckDNS.

DNS Record Example:

Record Type \tHostname \tPublic IP \tTTL (Seconds)

A \tmeet.example.org \tYour Meeting Server Public IP (x.x.x.x) \t1800

Set up the Fully Qualified Domain Name (FQDN) (optional)

If the machine used to host the Jitsi Meet instance has a FQDN (for example meet.example.org) already set up in DNS, you can set it with the following command:

sudo hostnamectl set-hostname meet.example.org

Then add the same FQDN in the /etc/hosts file:

127.0.0.1 localhost x.x.x.x meet.example.org

note

x.x.x.x is your server's public IP address.

Finally on the same machine test that you can ping the FQDN with:

ping "$(hostname)"

If all worked as expected, you should see: meet.example.org

Add the Prosody package repository

This will add the Prosody repository so that an up to date Prosody is installed, which is necessary for features including the lobby feature.

sudo curl -sL https://prosody.im/files/prosody-debian-packages.key -o /usr/share/keyrings/prosody-debian-packages.key

echo "deb [signed-by=/usr/share/keyrings/prosody-debian-packages.key] http://packages.prosody.im/debian $(lsbrelease -sc) main" ¦ sudo tee /etc/apt/sources.list.d/prosody-debian-packages.list

sudo apt install lua5.2

Add the Jitsi package repository

This will add the jitsi repository to your package sources to make the Jitsi Meet packages available.

curl -sL https://download.jitsi.org/jitsi-key.gpg.key ¦ sudo sh -c 'gpg --dearmor > /usr/share/keyrings/jitsi-keyring.gpg'

echo "deb [signed-by=/usr/share/keyrings/jitsi-keyring.gpg] https://download.jitsi.org stable/" ¦ sudo tee

/etc/apt/sources.list.d/jitsi-stable.list

Update all package sources:

sudo apt update

Setup and configure your firewall

The following ports need to be open in your firewall, to allow traffic to the Jitsi Meet server:

80 TCP => For SSL certificate verification / renewal with Let's Encrypt. Required

443 TCP => For general access to Jitsi Meet. Required

10000 UDP => For General Network Audio/Video Meetings. Required

22 TCP => For Accessing your Server using SSH (change the port accordingly if it's not 22). Required

3478 UDP => For querying the stun server (coturn, optional, needs config.js change to enable it).

5349 TCP => For fallback network video/audio communications over TCP (when UDP is blocked for example), served by coturn. Required

If you are using ufw, you can use the following commands:

sudo ufw allow 80/tcp

sudo ufw allow 443/tcp

sudo ufw allow 10000/udp

sudo ufw allow 22/tcp

sudo ufw allow 3478/udp

sudo ufw allow 5349/tcp

sudo ufw enable

Check the firewall status with:

sudo ufw status verbose

Using SSH

For more details on using and hardening SSH access, see the corresponding Debian or Ubuntu documentation.

Forward ports via your router

If you are running Jitsi Meet on a server behind NAT, forward the ports on your router to your server's IP address.

Note: if participants cannot see or hear each other, double check your firewall / NAT rules.

TLS Certificate

In order to have encrypted communications, you need a TLS certificate.

During installation of Jitsi Meet you can choose between different options:

The recommended option is to choose Let's Encrypt Certificate option

But if you want to use a different certificate you should get that certificate first and then install jitsi-meet and choose I want to use my own certificate.

You could also use the self-signed certificate(Generate a new self-signed certificate) but this is not recommended for the following reasons:

Using a self-signed certificate will result in warnings being shown in your users browsers, because they cannot verify your 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 your server if you choose a self-signed certificate.

Install Jitsi Meet

Note: The installer will check if Nginx or Apache are present (in that order) and configure a virtual host within the web server it finds to serve Jitsi Meet.

If you are already running Nginx on port 443 on the same machine, turnserver configuration will be skipped as it will conflict with your current port 443.

jitsi-meet installation

sudo apt install jitsi-meet

SSL/TLS certificate generation: You will be asked about SSL/TLS certificate generation. See above for details.

Hostname: You will also be asked to enter the hostname of the Jitsi Meet instance. If you have a domain, use the specific domain name, for example: meet.example.org. Alternatively you can enter the IP address of the machine (if it is static or doesn't change).

This hostname will be used for virtualhost configuration inside Jitsi Meet and also, you and your correspondents will be using it to access the web conferences.

Access Control

Jitsi Meet server: Note: By default, anyone who has access to your Jitsi Meet server will be able to start a conference: if your server is open to the world, anyone can have a chat with anyone else. If you want to limit the ability to start a conference to registered users, follow the instructions to set up a secure domain.

Conferences/Rooms: The access control for conferences/rooms is managed in the rooms, you can set a password on the webpage of the specific room after creation. See the User Guide for details: https://jitsi.github.io/handbook/docs/user-guide/user-guide-start-a-jitsi-meeting

Advanced configuration

If the installation is on a machine behind NAT jitsi-videobridge should configure itself automatically on boot. If three way calls do not work, further configuration of jitsi-videobridge is needed in order for it to be accessible from outside.

Provided that all required ports are routed (forwarded) to the machine that it runs on. By default these ports are TCP/443 and UDP/10000.

Add a static mapping to the ice4j.harvest.mapping section in /etc/jitsi/videobridge/jvb.conf:

```
ice4j {
  harvest {
    mapping {
      static-mappings = [
        {
          local-address = "
```

σ υ ν η μ μ έ ν α::

Μ ο ν α δ ι κ ό α ν α γ ν ω ρ ι σ τ ι κ ό λ ύ σ η ς: #1017
Σ υ γ γ ρ α φ έ α ς:: Kosmas
Τ ε λ ε υ τ α ί α ε ν η μ έ ρ ω σ η: 2026-05-11 14:32