

jitsi-Meet - Self-Hosting Guide - openSUSE

Self-Hosting Guide - openSUSE

This document describes the steps for a quick Jitsi-Meet installation, paired with a single Videobridge and a single Jicofo on openSUSE Leap 15.2.

Note: Many of the installation steps require root access.

Installation

1. Add the OBS repository:

Note: When Jitsi-Meet is merged into openSUSE Factory, this will be obsolete.

```
zypper addrepo https://download.opensuse.org/repositories/home:/SchoolGuy:/jitsi/openSUSE_Leap_15.2/home:SchoolGuy:jitsi.repo
```

2. Refresh the repositories:

```
zypper refresh
```

3. Install Jitsi-Meet and its dependencies:

```
zypper install nginx prosody lua51-zlib jitsi-meet jitsi-videobridge jitsi-jicofo
```

optional Add-Ons

- Install the Jibri Add-On: `zypper install jitsi-jibri`
- Install the Jigasi Add-On: `zypper install jitsi-jigasi`

Configuration

The following sections describe how to configure the different packages.

Replace <FQDN> with your domain name and YOURSECRET3 with a strong password.

openSUSE

Prosody

- Open and adjust the Prosody configuration file under `/etc/prosody/prosody.cfg.lua`:

```
----- Server-wide settings -----
admins = { "focus@auth.<FQDN>" }
cross_domain_bosh = true;
component_ports = { 5347 }
modules_enabled = {
    -- HTTP modules
    "bosh"; -- Enable BOSH clients, aka "Jabber over HTTP"
    -- jitsi
    "smacks";
    "mam";
    "lastactivity";
    "offline";
    "pubsub";
    "adhoc";
    "websocket";
    "http_altconnect";
    "compression";
}
```

- Create a new configuration file named `<FQDN>.cfg.lua` in `/etc/prosody/conf.avail/` with the following content:

```
plugin_paths = { "/usr/share/jitsi-meet/prosody-plugins/" }

-- As per https://prosody.im/doc/setting_up_bosh#proxying_requests
consider_bosh_secure = true

-- domain mapper options, must at least have domain base set to use the mapper
muc_mapper_domain_base = "<FQDN>";

turncredentials_secret = "YOURSECRET3";

turncredentials = {
    { type = "stun", host = "<FQDN>", port = "3478" },
    { type = "turn", host = "<FQDN>", port = "3478", transport = "udp" },
    -- { type = "turns", host = "<FQDN>", port = "443", transport = "tcp" }
};

VirtualHost "<FQDN>"
    authentication = "anonymous"
    ssl = {
        key = "/var/lib/prosody/<FQDN>.key";
        certificate = "/var/lib/prosody/<FQDN>.crt";
    }
    speakerstats_component = "speakerstats.<FQDN>"
    conference_duration_component = "conferenceduration.<FQDN>"
    modules_enabled = {
        "bosh";
        "pubsub";
        "speakerstats";
        "turncredentials";
        "conference_duration";
    }
    c2s_require_encryption = false
```

openSUSE

```
Component "conference.<FQDN>" "muc"  
  modules_enabled = {  
    "muc_meeting_id";  
    "muc_domain_mapper";  
  }  
  admins = { "focus@auth.<FQDN>" }  
  muc_room_locking = false  
  muc_room_default_public_jids = true
```

```
-- internal muc component  
Component "internal.auth.<FQDN>" "muc"  
  modules_enabled = {  
    "ping";  
  }  
  admins = { "focus@auth.<FQDN>" }  
  muc_room_locking = false  
  muc_room_default_public_jids = true  
  muc_room_cache_size = 1000
```

```
Component "jitsi-videobridge.<FQDN>"  
  component_secret = "YOURSECRET3"
```

```
VirtualHost "auth.<FQDN>"  
  ssl = {  
    key = "/var/lib/prosody/auth.<FQDN>.key";  
    certificate = "/var/lib/prosody/auth.<FQDN>.crt";  
  }  
  authentication = "internal_plain"
```

```
Component "focus.<FQDN>"  
  component_secret = "YOURSECRET3"
```

```
Component "speakerstats.<FQDN>" "speakerstats_component"  
  muc_component = "conference.<FQDN>"
```

```
Component "conferenceduration.<FQDN>" "conference_duration_component"  
  muc_component = "conference.<FQDN>"
```

- Create a symlink for the configuration:
In --symbolic /etc/prosody/conf.avail/<FQDN>.cfg.lua /etc/prosody/conf.d/<FQDN>.cfg.lua
- Create the certificates via prosodyctl cert generate <DOMAIN>.
The value <DOMAIN> represents the following URLs.
 - auth.<FQDN>
 - conference.<FQDN>
 - conferenceduration.<FQDN>
 - internal.auth.<FQDN>
 - FQDN
 - focus.<FQDN>
 - jitsi-videobridge.<FQDN>
 - callcontrol.<FQDN> **Note:** This is only needed if you deploy Jigasi
 - recorder.<FQDN> **Note:** This is only needed if you deploy Jibri
- /var/lib/prosody/: Symlink all generated *.crt and *.key files to /etc/prosody/certs/.

openSUSE

Note: Please do not link other certificates.

- Add the certificates to the system keystore:
 - In `--symbolic --force /var/lib/prosody/auth.<FQDN>.cert /usr/local/share/ca-certificates/auth.<FQDN>.cert`
 - `update-ca-certificates --fresh`
- Create conference focus user: `prosodyctl register focus auth.<FQDN> YOURSECRET3`

Nginx

Edit the file `jitsi-meet.conf` in `/etc/nginx/vhosts.d/` (which was installed along with `jitsi-meet`) and do the following:

- Check the `server_name` value.
- Check the TLS certificates (Let's Encrypt for production use, Prosody for testing, for example).

Note: If you are using an existing server, please make sure to adjust the websocket and bosh part, too.

Jitsi-Meet

- Go to `/srv/jitsi-meet` and edit `config.js`:

```
var config = {
  hosts: {
    domain: '<FQDN>',
    muc: 'conference.<FQDN>',
    bridge: 'jitsi-videobridge.<FQDN>',
    focus: 'focus.<FQDN>'
  },
  useNicks: false,
  bosh: '///<FQDN>/http-bind',
};
```

Note: Please be aware that this is the minimal configuration.

Jitsi-Videobridge

Note: We use a combination of the [new Videobridge configuration](#) and the legacy one with the `sip-communicator.properties` file. We have to do this because of the `STATISTICS_TRANSPORT` property.

If we remove `org.jitsi.videobridge.STATISTICS_TRANSPORT=muc,colibri` from `sip-communicator.properties`, the videobridge will not work!

- Go to the directory `/etc/jitsi/videobridge`
- Edit the file `jitsi-videobridge.conf`
 - Set `JVB_HOSTNAME` to your `<FQDN>`.
 - Set `JVB_SECRET` to your own secret.
- Edit the file `application.conf` and adjust the values under `apis` and `websockets`, especially set

Seite 4 / 7

© 2026 Eric Schirra <webmaster@schirra.net> | 2026-06-20 11:27

URL: https://faq.schirra.net/phpMyFAQ/content/3/113/de/jitsi_meet-_self_hosting-guide-_opensuse.html

openSUSE

a unique ID as muc_nickname with uuidgen for example.

```
apis {
  xmpp-client {
    configs {
      xmpp-server-1 {
        hostname="localhost"
        domain = "auth.${FQDN}"
        username = "focus"
        password = "YOURSECRET3"
        muc_jids = "JvbBrewery@internal.auth.${FQDN}"
        # The muc_nickname must be unique across all jitsi-videobridge instances
        muc_nickname = "unique-id"
        disable_certificate_verification = true
      }
    }
  }
}
websockets {
  enabled=true
  server-id="default-id"
  domain="${FQDN}"
}
```

Jitsi-Jicofo

- Go to the directory /etc/jitsi/jicofo
- Edit the file jitsi-jicofo.conf
 - Set the property JICOFO_HOSTNAME to <FQDN>.
 - Set the property JICOFO_SECRET to the password the Prosody user got in above setup.
 - Set the property JICOFO_AUTH_DOMAIN to auth.<FQDN>.
 - Set the property JICOFO_AUTH_USER to the Prosody user from above setup.
- Edit the file sip-cmmunicator.properties
 - Set the property org.jitsi.jicofo.BRIDGE_MUC to JvbBrewery@internal.auth.<FQDN>.
 - Set the property org.jitsi.jicofo.jibri.BREWERY to JibriBrewery@internal.auth.<FQDN>.
 - Depending on your cert setup set org.jitsi.jicofo.ALWAYS_TRUST_MODE_ENABLED to true or false.

Add-On: Jitsi-Jibri

- Add to the file /etc/prosody/conf.avail/<FQDN>.cfg.lua the following snippet at the end of the file.

```
VirtualHost "recorder.<FQDN>"
  modules_enabled = {
    "ping";
  }
  authentication = "internal_plain"
```

openSUSE

- Run `prosodyctl register jibri auth.<FQDN> YOURSECRET3` and replace `YOURSECRET3` with an appropriate one.
- `prosodyctl register recorder recorder.<FQDN> YOURSECRET3` and replace `YOURSECRET3` with an appropriate one.
- Go to the directory `/etc/jitsi/jibri` and edit the following properties you see listed below. The rest can be left as is.

```
jibri{
  api{
    environments = [
      {
        xmpp-domain = "<FQDN>"
        control-muc {
          domain = "internal.<FQDN>"
        }
        control-login {
          domain = "recorder.<FQDN>"
          username = "recorder"
          password = "YOURSECRET3"
        }
        call-login {
          domain = "recorder.<FQDN>"
          username = "recorder"
          password = "YOURSECRET3"
        }
      }
    ]
  }
}
```

- Edit the file `/etc/jitsi/jicofo/sip-communicator.properties` and add the following properties:

```
org.jitsi.jicofo.jibri.BREWERY=JibriBrewery@internal.auth.<FQDN>
org.jitsi.jicofo.jibri.PENDING_TIMEOUT=90
```

- Edit the file `/srv/jitsi-meet/config.js` and set the following properties:

```
fileRecordingsEnabled: true, // If you want to enable file recording
liveStreamingEnabled: true, // If you want to enable live streaming
hiddenDomain: 'recorder.<FQDN>',
```

- Edit `/srv/jitsi-meet/interface_config.js` and make sure the `TOOLBAR_BUTTONS` array contains the recording and the livestreaming value if you want those features.

```
TOOLBAR_BUTTONS: [
  'microphone', 'camera', 'closedcaptions', 'desktop',
  'embedmeeting', 'fullscreen',
  'fodeviceselection', 'hangup', 'profile', 'chat', 'recording',
  'livestreaming', 'etherpad', 'sharedvideo', 'settings', 'raisehand',
  'videoquality', 'filmstrip', 'invite', 'feedback', 'stats', 'shortcuts',
  'tileview', 'videobackgroundblur', 'download', 'help', 'mute-
everyone', 'security'
],
```

Add-On: Jitsi-Jigasi

Note from openSUSE packagers: We've packaged it but we don't have the infrastructure to set up this component. Hence we can't provide a guide for this so far.

Services

Now everything should be working. That means you are ready to start everything up:

1. `systemctl start prosody`
2. `systemctl start jitsi-videobridge`
3. `systemctl start jitsi-jicofo`
4. `systemctl start jitsi-jibri` (if configured and installed beforehand)
5. `systemctl start jitsi-jigasi` (if configured and installed beforehand)
6. `systemctl start nginx`

Final notes

- The Jitsi Software has a lot of dependencies and thus we recommend to run this on a dedicated host for Jitsi.
- Updating Jitsi is crucial to get rid of bugs and updated dependencies with possible security fixes.
- Although tempted through Chrome: Don't install a full X11 stack like KDE or Gnome for this.
- Don't mix the rpms or debs with a source installation of the same component.
- Securely backup your configuration, preferably in a VCS. This saves time and pain when doing rollbacks or dealing with other problems.

Quelle: <https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-opensuse>

Eindeutige ID: #1113

Verfasser: n/a

Letzte Änderung: 2021-06-08 20:19