Déploiement Jitsi Meet (paquets Debian) derrière Caddy

Guide d'installation d'une instance Jitsi Meet auto-hébergée pour réunions clients, sans Docker, avec Caddy en reverse proxy frontal et TURN configuré proprement.

- **Cible** : réunions clients en ligne (2 à ~20 participants)
- **OS** : Debian 12
- **Reverse proxy** : Caddy (gère le TLS)
- **FQDN d'exemple** : `meet.makeitsimple.be`

1. Prérequis

Élément	Détail
FQDN	`meet.makeitsimple.be` pointant vers l'IP publique
Port 443/tcp	Web + WebSocket (via Caddy)
Port 10000/udp	Média JVB — **direct vers la VM**, ne passe PAS par Caddy
Port 5349/tcp+udp	TURN over TLS (fallback NAT restrictif)
Ressources	4 vCPU / 4–8 Go RAM recommandés

<note warning> Le média audio/vidéo (10000/udp) et le TURN ne transitent **jamais** par Caddy. Caddy ne proxifie que le HTTP/WebSocket. C'est la source de confusion n°1. </note>

2. Préparation système

```
hostnamectl set-hostname meet.makeitsimple.be
# Vérifier /etc/hosts :
#   127.0.0.1 meet.makeitsimple.be meet
apt update && apt upgrade -y
apt install -y gnupg2 curl
```

3. Dépôts Jitsi

```
curl -sL https://download.jitsi.org/jitsi-key.gpg.key \
  | gpg --dearmor -o /usr/share/keyrings/jitsi-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/jitsi-keyring.gpg]
https://download.jitsi.org stable/" \
  > /etc/apt/sources.list.d/jitsi-stable.list

apt update
```

4. Installation

```
apt install -y jitsi-meet
```

Pendant l'installation :

1. **Hostname** : saisir `meet.makeitsimple.be`
2. **Certificat** : choisir **« I want to use my own certificate »** — c'est Caddy qui gère le TLS, il ne faut PAS laisser Jitsi tenter Let's Encrypt. On peut pointer vers un certificat existant ou bidon, il sera contourné.

Cette commande installe : `jitsi-meet` (web), `jicofo` (focus/orchestration), `jitsi-videobridge2` (JVB/média), `prosody` (XMPP), un **nginx local** et un **coturn local**.

5. Basculer le web Jitsi en HTTP local

Par défaut nginx écoute en 443 et se battra avec Caddy. On le fait écouter en local uniquement.

Éditer `/etc/nginx/sites-available/meet.makeitsimple.be.conf` :

```
server {
    listen 127.0.0.1:8000;
    server_name meet.makeitsimple.be;

    # ... conserver tout le reste de la conf générée par Jitsi ...
    # (locations /, /http-bind, /xmpp-websocket, etc.)
}
```

- Faire écouter nginx sur `127.0.0.1:8000` uniquement.
- Supprimer / commenter les blocs `listen 443 ssl` et `listen [::]:443 ssl`.

```
nginx -t && systemctl restart nginx
```

6. Caddy en frontal

Dans le `Caddyfile` :

```
meet.makeitsimple.be {
    reverse_proxy 127.0.0.1:8000
}
```

```
caddy validate --config /etc/caddy/Caddyfile
systemctl reload caddy
```

Caddy récupère le certificat Let's Encrypt automatiquement et proxifie HTTP + WebSocket (BOSH/XMPP-WS) vers Jitsi.

7. Configuration du média (JVB)

Point de rupture le plus fréquent. Le JVB doit annoncer les bonnes adresses IP pour la négociation ICE.

Éditer `/etc/jitsi/videobridge/sip-communicator.properties` :

```
org.ice4j.ice.harvest.NAT_HARVESTER_LOCAL_ADDRESS=<IP_PRIVEE_VM>
org.ice4j.ice.harvest.NAT_HARVESTER_PUBLIC_ADDRESS=<IP_PUBLIQUE>
```

- **VM derrière NAT** : renseigner l'IP privée locale ET l'IP publique.
- **VM en IP publique directe** : les deux valeurs peuvent être identiques (l'IP publique).

Ouvrir **10000/udp** de l'extérieur vers la VM (pare-feu + NAT éventuel).

```
systemctl restart jitsi-videobridge2
```

8. Configuration TURN (coturn) — TURN/UDP sans TLS

Le TURN sert de **relais média** quand un participant est derrière un NAT/firewall restrictif qui bloque l'UDP direct (10000/udp). Typiquement : **clients en réseau d'entreprise**. Sans TURN fonctionnel, ces participants se connectent mais n'ont **ni son ni vidéo**.

Cette section décrit un TURN **sans TLS**, en TURN/UDP sur le port **3478** uniquement. Avantage : **aucun certificat à gérer** côté coturn (pas de synchro depuis Caddy, pas de renouvellement à automatiser). Le TURN/UDP 3478 couvre la grande majorité des NAT restrictifs.

<note> **Limite** : le TURN over TLS (5349) n'est utile que pour traverser les firewalls corporate les plus verrouillés, qui n'autorisent que le trafic HTTPS sortant. Ceux-là ne seront pas couverts par cette config. Si un client réel se retrouve bloqué malgré 3478/udp, il faudra ajouter un TURN/TLS 5349 (nécessite alors un certificat). </note>

8.1 Config coturn

Éditer `/etc/turnserver.conf` (généré par Jitsi). Vérifier / ajuster :

```
listening-port=3478
listening-ip=<IP_PRIVEE_VM>
relay-ip=<IP_PRIVEE_VM>
external-ip=<IP_PUBLIQUE>/<IP_PRIVEE_VM>

# Auth partagée avec Jitsi
use-auth-secret
static-auth-secret=<SECRET_GENERE_PAR_JITSI>
realm=meet.makeitsimple.be

# Pas de TLS : on ne définit ni tls-listening-port, ni cert, ni pkey
```

```
no-tls
no-dtls

# Durcissement
no-tcp-relay
no-multicast-peers
no-cli
```

<note> Le `static-auth-secret` est généré à l'install. Le récupérer dans `/etc/turnserver.conf` (existant) ou dans `/etc/prosody/conf.d/meet.makeitsimple.be.cfg.lua` (bloc `turncredentials`). Commenter ou supprimer les lignes `tls-listening-port`, `cert` et pkey générées par défaut si elles sont présentes. </note>

8.2 Pointer Jitsi vers le TURN/UDP

Vérifier dans `/etc/prosody/conf.d/meet.makeitsimple.be.cfg.lua` que le bloc `turncredentials` n'annonce **que** STUN et TURN/UDP sur 3478 (retirer toute entrée `turns` / 5349) :

```
turncredentials = {
  { type = "stun", host = "meet.makeitsimple.be", port = "3478" },
  { type = "turn", host = "meet.makeitsimple.be", port = "3478", transport =
"udp" }
}
```

Ouvrir **3478/udp** de l'extérieur vers la VM.

```
systemctl restart coturn prosody
```

9. Redémarrage complet

```
systemctl restart prosody jicofo jitsi-videobridge2 coturn nginx
systemctl reload caddy
```

10. Récapitulatif pare-feu

Port	Proto	Sens	Rôle
443	tcp	Externe → VM	Web + WebSocket (Caddy)
80	tcp	Externe → VM	ACME / redirection (Caddy)
10000	udp	Externe → VM	Média JVB (direct)
3478	udp	Externe → VM	STUN / TURN (relais fallback)

11. Checklist de test de bout en bout

1. [ ] **DNS** : `dig +short meet.makeitsimple.be` renvoie bien l'IP publique.

2. [ ] **HTTPS** : la page d'accueil Jitsi charge en `https://` sans alerte de certificat (Caddy).
3. [ ] **Création de salle** : créer une réunion, la page se charge, caméra/micro détectés.
4. [ ] **Test à 2 participants** sur réseaux différents (ex. un en 4G, un en fibre) → audio + vidéo OK dans les deux sens.
5. [ ] **Test NAT restrictif** : un participant sur réseau d'entreprise / VPN corporate → valide que le TURN prend le relais (audio/vidéo présents).
6. [ ] **Partage d'écran** fonctionnel.
7. [ ] **Invité sans compte** : ouvrir le lien en navigation privée → accès direct sans login.
8. [ ] **Vérif média** : sur https://webrtc.github.io/samples/ ou via les stats Jitsi (bouton i), confirmer que la connexion passe en `udp` direct (host/srflx) et bascule en `relay` seulement si nécessaire.

<note tip> Pour diagnostiquer un problème “je vois mais pas de son” : ouvrir les stats de connexion dans Jitsi (icône `i` en haut à gauche d'un participant). Si le type de connexion reste bloqué ou vide, c'est côté JVB (10000/udp) ou TURN. </note>

12. Vérification des logs

```
journalctl -u jitsi-videobridge2 -f    # média
journalctl -u jicofo -f                # orchestration des conférences
journalctl -u prosody -f               # XMPP / auth
journalctl -u coturn -f                # relais TURN
journalctl -u caddy -f                 # reverse proxy / TLS
```

13. Restreindre la création de réunions (secure domain)

Objectif : seuls les **hôtes authentifiés** peuvent créer une réunion. Les invités rejoignent librement une salle **déjà ouverte** par un hôte, sans compte.

Comportement :

- Un invité qui tente de créer une salle → écran « en attente de l'hôte ».
- Un hôte se connecte (login/mot de passe) → la salle s'ouvre → les invités entrent.

13.1 Prosody

Éditer `/etc/prosody/conf.d/meet.makeitsimple.be.cfg.lua`.

Le VirtualHost principal passe en authentification interne :

```
VirtualHost "meet.makeitsimple.be"
    authentication = "internal_hashed"
```

Ajouter un VirtualHost anonyme pour les invités (à créer s'il n'existe pas) :

```
VirtualHost "guest.meet.makeitsimple.be"
```

```
    authentication = "anonymous"
    c2s_require_encryption = false
```

<note> `guest.meet.makeitsimple.be` est un domaine **XMPP interne**, pas un vrai DNS : aucun enregistrement DNS ni certificat n'est requis pour lui. </note>

13.2 Jicofo

Éditer `/etc/jitsi/jicofo/jicofo.conf` :

```
jicofo {
  authentication {
    enabled = true
    type = XMPP
    login-url = "meet.makeitsimple.be"
  }
}
```

13.3 Config web

Éditer `/etc/jitsi/meet/meet.makeitsimple.be-config.js` :

```
var config = {
    hosts: {
        domain: 'meet.makeitsimple.be',
        anonymousdomain: 'guest.meet.makeitsimple.be',
        // ...
    },
    // ...
};
```

13.4 Créer les comptes hôtes

```
prosodyctl register vincent meet.makeitsimple.be <motdepasse>
```

- Un hôte = un compte.
- Révoquer un hôte : `prosodyctl deluser vincent@meet.makeitsimple.be`
- Changer un mot de passe : `prosodyctl passwd vincent@meet.makeitsimple.be`

13.5 Redémarrer

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

13.6 Test

1. [ ] En navigation privée (non authentifié), créer une salle → écran « en attente de l'hôte ».
2. [ ] Se connecter avec un compte hôte → la salle s'ouvre.
3. [ ] Un second onglet anonyme rejoint la salle ouverte → accès direct sans login.

14. Cacher le formulaire de la page d'accueil

Objectif : sur la racine `meet.makeitsimple.be`, ne plus afficher le champ « nom de la salle » qui invite un passant à créer/rejoindre une réunion au hasard. Mesure **cosmétique** (elle n'empêche pas d'accéder à une salle dont on connaît l'URL — cf. mot de passe de salle / lobby pour ça), mais elle évite d'exposer le formulaire publiquement.

Éditer `/etc/jitsi/meet/meet.makeitsimple.be-config.js` et ajouter (ou décommenter) :

```
// Masque la zone de saisie / le bouton de création sur la page d'accueil
enableWelcomePage: false,
```

- `enableWelcomePage: false` : la page d'accueil avec le formulaire ne s'affiche plus. Une salle reste accessible directement par son URL (https://meet.makeitsimple.be/NomDeLaSalle).

<note tip> Variante douce si tu veux garder une page d'accueil mais sans le champ de création aléatoire : conserver `enableWelcomePage: true` et désactiver la liste des salles récentes / le générateur de nom via `hideAddRoomButton` et un `MEET_ROOM_NAME` vide selon la version. La bascule à `false` reste la plus simple et la plus robuste. </note>

Rediriger éventuellement la racine vers ta vitrine (au niveau Caddy) si tu veux que `/` ne montre rien du tout :

```
meet.makeitsimple.be {
    @root path /
    redir @root https://www.makeitsimple.be 302

    handle {
        reverse_proxy 127.0.0.1:8000
    }
}
```

Aucun redémarrage de service nécessaire pour `config.js` (rechargé côté client) ; recharger Caddy si tu ajoutes la redirection :

```
systemctl reload caddy
```

15. Personnalisation (logo, taille, favicon, nom)

15.1 Remplacer le logo (watermark en haut à gauche)

Par défaut Jitsi utilise `/usr/share/jitsi-meet/images/watermark.svg` (ou `.png` selon version).

```
# Sauvegarde de l'original
cp /usr/share/jitsi-meet/images/watermark.svg /usr/share/jitsi-meet/images/watermark.svg.bak
# Déployer le logo (garder une copie source HORS du dossier packagé, cf. 15.5)
cp /opt/mis-branding/watermark.svg /usr/share/jitsi-meet/images/watermark.svg
```

15.2 Lien, affichage et logo par défaut

Éditer `/usr/share/jitsi-meet/interface_config.js` :

```
var interfaceConfig = {
    JITSI_WATERMARK_LINK: 'https://www.makeitsimple.be',
    SHOW_JITSI_WATERMARK: true,
    DEFAULT_LOGO_URL: 'images/watermark.svg',
    DEFAULT_WELCOME_PAGE_LOGO_URL: 'images/watermark.svg',
    // ...
};
```

15.3 Modifier la taille du logo

La taille du watermark est **contrôlée en CSS**, pas par les dimensions du fichier image. Le watermark est affiché par la classe `.watermark .leftwatermark`.

Créer un CSS de surcharge, ex. `/usr/share/jitsi-meet/css/custom.css` :

```
.watermark.leftwatermark {
    width: 160px !important;      /* défaut ~= 140px */
    height: 60px !important;
    background-size: contain !important;
    background-repeat: no-repeat !important;
}
```

Puis référencer ce CSS dans `/usr/share/jitsi-meet/base.html` (juste avant </head>) :

```
<link rel="stylesheet" href="css/custom.css">
```

<note> Ajuster `width/height` selon le ratio de ton logo. `background-size: contain` évite la déformation. Vider le cache navigateur pour voir le changement. </note>

15.4 Favicon, nom de l'application

Dans `interface_config.js` :

```
    APP_NAME: 'Make IT Simple Meet',
    NATIVE_APP_NAME: 'Make IT Simple Meet',
    PROVIDER_NAME: 'Make IT Simple',
```

Favicon : remplacer `/usr/share/jitsi-meet/images/favicon.ico`.

```
cp /opt/mis-branding/favicon.ico /usr/share/jitsi-meet/images/favicon.ico
```

15.5 Préserver la personnalisation aux mises à jour

<note warning> `interface_config.js`, `base.html`, `css/` et `images/` appartiennent au paquet `jitsi-meet-web`. Un `apt upgrade` **peut écraser** ces fichiers. Il faut versionner tes fichiers custom et les redéployer après upgrade. </note>

Garder les sources de branding dans un dossier dédié (idéalement versionné en git) :

```
/opt/mis-branding/
├── watermark.svg
├── favicon.ico
├── custom.css
└── redeploy.sh
```

Script `/opt/mis-branding/redeploy.sh` :

```
#!/bin/bash
set -e
DST=/usr/share/jitsi-meet
cp /opt/mis-branding/watermark.svg "$DST/images/watermark.svg"
cp /opt/mis-branding/favicon.ico   "$DST/images/favicon.ico"
cp /opt/mis-branding/custom.css    "$DST/css/custom.css"

# Ré-injecter le lien CSS dans base.html s'il a été écrasé
grep -q 'css/custom.css' "$DST/base.html" || \
  sed -i 's#</head>#<link rel="stylesheet" href="css/custom.css">\n</head>#'
"$DST/base.html"

echo "Branding redéployé."
```

```
chmod +x /opt/mis-branding/redeploy.sh
# À relancer après chaque mise à jour de jitsi-meet-web
/opt/mis-branding/redeploy.sh
```

<note tip> Aucun redémarrage de service : les fichiers sont statiques (servis par nginx → Caddy) et rechargés côté navigateur. Vider le cache pour tester. </note>

Notes de maintenance

- **TURN** : configuration en TURN/UDP sans TLS (section 8) → **aucun certificat coturn à gérer ni à renouveler**. Si un client tombe sur un firewall corporate bloquant l'UDP, prévoir l'ajout d'un TURN/TLS 5349 (nécessitera alors un certificat dédié, ex. via certbot en `webroot` ou DNS-01, indépendant de Caddy).
- **Personnalisation** : après chaque `apt upgrade` de `jitsi-meet-web`, relancer `/opt/mis-branding/redeploy.sh` (section 15.5) pour restaurer logo/favicon/CSS.
- **Montée en charge** : au-delà de ~20–25 participants simultanés sur une salle, envisager un JVB supplémentaire (scaling horizontal).
- **Mises à jour** : `apt update && apt upgrade` — les paquets Jitsi sont mis à jour régulièrement, tester après upgrade majeur.

From:
https://wiki.makeitsimple.be/ - **makeITsimple wiki**

Permanent link:
https://wiki.makeitsimple.be/doku.php?id=linux:jitsi

Last update: **2026/09/07 07:10**