Ubuntu

- Accès SSH, Webmin, serveur Web et SFTP sous Ubuntu Server 16.04
- Installation de Netdata
- Visioconférence avec Jitsimeet
    - Jitsi

Accès SSH, Webmin, serveur Web et SFTP sous Ubuntu Server 16.04

Installation du serveur SSH



Pour accéder au serveur à distance, il est nécessaire d'installer openssh-server.

On met tout d'abord à jour le serveur Ubuntu et on installe ensuite openssh-server.

```
sudo apt-get update && apt-get upgrade -y
sudo apt-get install openssh-server
```

On active le service au démarrage.

```
sudo systemctl start ssh
sudo systemctl enable ssh
```

Installation de Netdata

Netdata fournit un script bash pour l'installation de Netdata.

```
apt-get install curl
bash <(curl -Ss https://my-netdata.io/kickstart-static64.sh
```

Netdata est ensuite accessible via : http://ip_du_serveur:19999

On a ici une vue d'ensemble du serveur : charge du processeur, réseau, écriture et lecture du disque dur, utilisation de la RAM (mémoire vive), ...



Installation de LAMP (Linux Apache MySQL PHP)



On va ici installer Apache2 en tant que serveur Web.

```
sudo apt-get install apache2
sudo apt-get install mysql-server
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql php-curl
sudo apt-get install phpmyadmin apache2-utils
```

Lors de l'installation du serveur mysql, on définit un mot de passe root.

On choisit Apache2 avec la barre espace parce qu'on a utilisé Apache2 et non lighttpd.

Installation de Webmin



Webmin n'a pas besoin d'apache pour fonctionner. Webmin est fourni avec un simple serveur web nommé miniserv.py. Selon la documentation de Webmin, l'installer sous Apache impacterait les performances. Cela n'est pas recommandé.

Pour installer Webmin sur un serveur Ubuntu 16.04, on commence par installer quelques dépendances :

```
sudo apt-get install -y perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime
libio-pty-perl apt-show-versions python libwww-perl liblwp-protocol-https-perl
```

On ajoute la clef pour vérifier l'intégrité des paquets du dépôts de Webmin:

```
sudo wget -O- http://www.webmin.com/jcameron-key.asc | sudo apt-key add -
```

On ajoute les dépôts à la fin du fichier sources.list :

```
sudo nano /etc/apt/sources.list
```

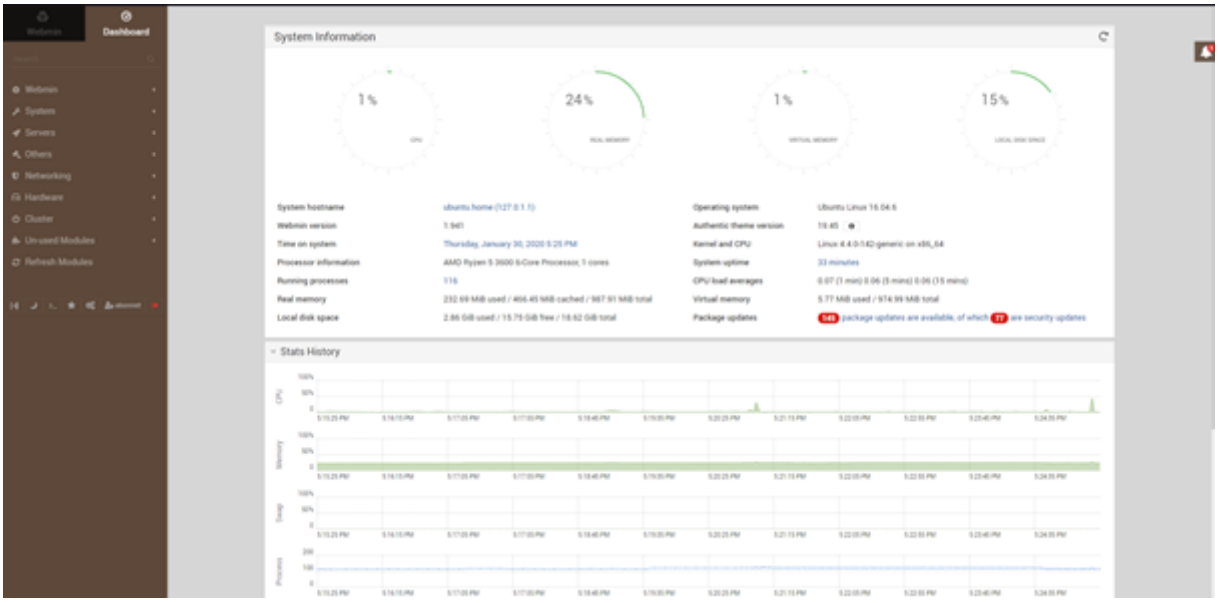
```
#Webmin
deb http://download.webmin.com/download/repository sarge contrib
deb http://webmin.mirror.somersettechsolutions.co.uk/repository sarge contrib
```

Il ne reste plus qu'à mettre à jour la liste des paquets et à l'installer :

```
sudo apt-get update
sudo apt-get install webmin
```

Webmin est ensuite disponible à l'adresse suivante :

https://192.168.1.31:10000/



Installation du serveur FTP sécurisé



Installation du paquet :

```
sudo apt-get install mysecureshell

#Ajout d'un nouvel utilisateur:
```

```
sudo adduser --home /home/bonnet --shell /usr/bin/mysecureshell bonnet
```

On rentre un mot de passe pour l’utilisateur.

On se connecte via un client avec le port 22 (qui doit être ouvert en TCP au niveau du pare-feu). Par exemple : Filezilla Client, qui est gratuit et open-source.

Configuration du pare-feu d'Ubuntu : UFW

```
sudo ufw enable
sudo ufw allow « Apache Full »
sudo ufw allow 22/tcp
sudo ufw allow 10000/tcp
sudo ufw allow 19999/tcp
ufw status
```

Installation de Netdata

Qu’est-ce que Netdata ?

Netdata est un outil open source permettant de visualiser et de surveiller des métriques en temps réel, optimisé pour accumuler tous les types de données, telles que l'utilisation du processeur, l'activité du disque, les requêtes SQL, les visites sur un site Web, etc.

Il est disponible sur Linux, FreeBSD et macOS.

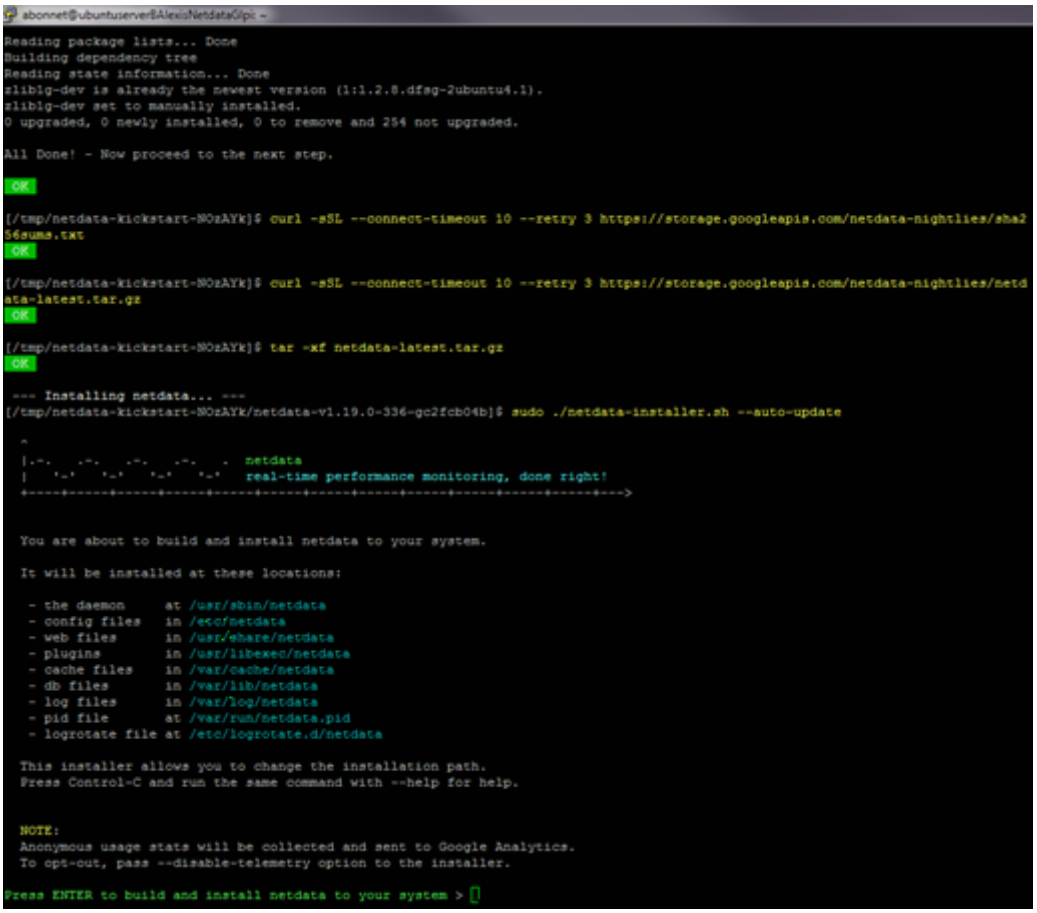
Netdata permet la supervision à distance du serveur.

Installation de Netdata

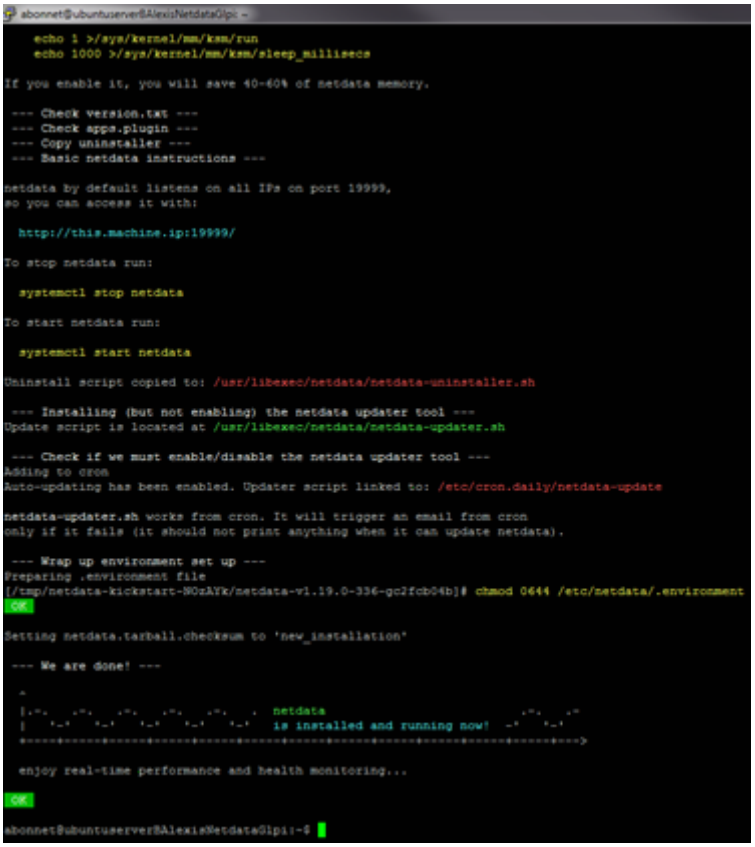
Afin d’installer Netdata, nous allons utiliser la ligne de commande mise à notre disposition par Netdata :

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

On aura régulièrement des messages nous demandant de confirmer l’installation de paquets. On écrit et valide « Y ».



Une fois l'installation terminée, nous aurons cela à l'écran. On presse entrée.



Netdata est désormais installé ! On peut accéder à Netdata via son ip depuis un autre poste du réseau. http://192.168.0.200/

Visioconférence avec Jitsimeet

Visioconférence avec Jitsimeet

Jitsi

Introduction

Jitsi est application libre multiplateforme de messagerie instantanée, voix sur IP et visioconférence. Les conférences sont sécurisées et chiffrées. Le logiciel est intégré à la liste des logiciels libres préconisés par l'État français dans le cadre de la modernisation globale de ses systèmes d'informations. Jitsi est disponible sur navigateur web, Android et iOS. On peut l'installer dans le cloud ou sur une machine comme Linux.

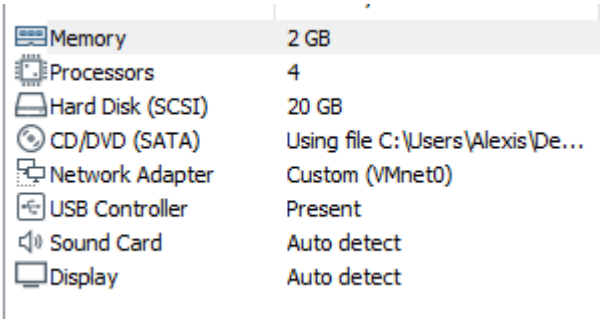


Jitsi est très utilisé en ce moment en tant qu'alternative à Microsoft Teams ou Skype Business. Le fait que Jitsi soit gratuit et open-source est très avantageux. On peut même ajouter des plugins à Jitsi et il est customisable.

Développement

Personnellement, j'ai un nom de domaine enregistré chez OVH (khroners.fr). Je vais donc utiliser ce domaine avec un sous-domaine jitsimeet.

Préparation de la machine virtuelle



La carte réseau est en bridge.

On met ensuite le serveur à jour.

```
apt update && upgrade -y
```

Installation du serveur web Nginx et du serveur Jitsi

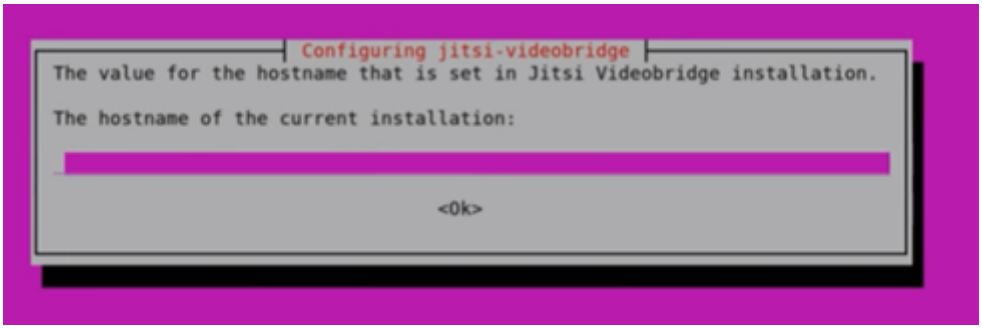
On installe ensuite nginx, un serveur web.

```
apt-get -y install nginx
```

On ajoute la clé publique de jitsi, le dépôt et on installe.

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

Arrivé à cette fenêtre, on rentre le domaine complet, dans mon cas jitsimeet.khroners.fr.



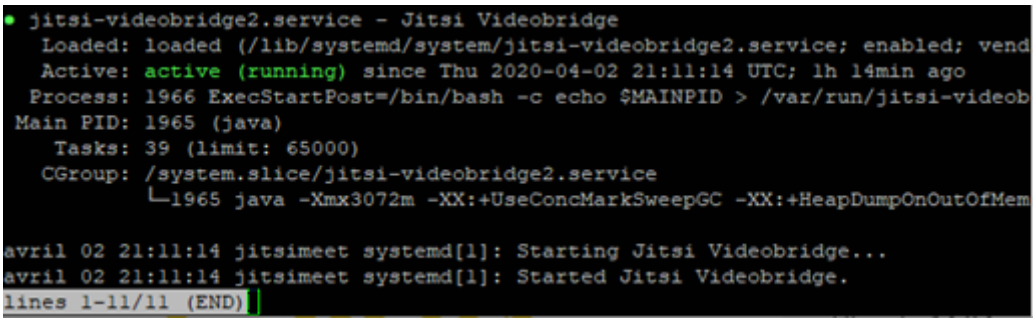


On choisit ici la seconde option « I want to use my own certificate ».

On presse « Entrée » pour les deux autres propositions.

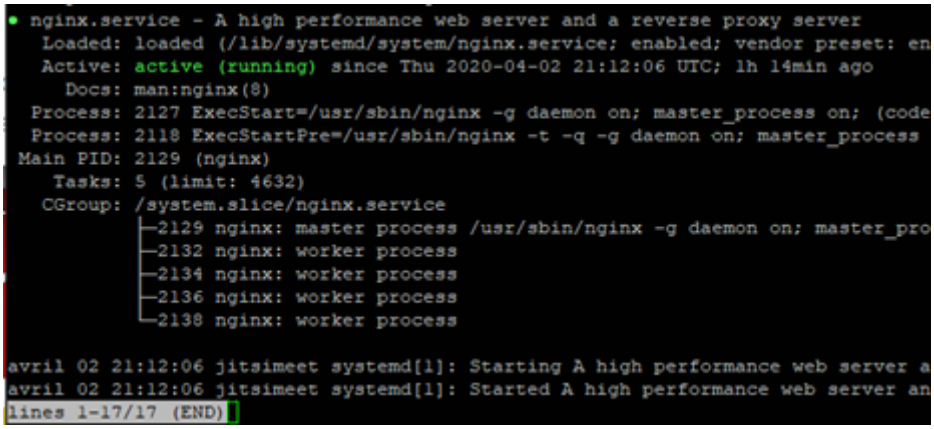
On peut voir le statut de jitsi-videobridge en rentrant la commande :

```
service jitsi-videobridge2 status
```



Idem pour nginx :

```
service nginx status
```



Puisque le serveur Jitsi est derrière un NAT, on rajoute ces deux options dans le fichier /etc/jitsi/videobridge/sip-communicator.properties.

```
nano /etc/jitsi/videobridge/sip-communicator.properties
```

Puis on insère ces deux lignes :

> org.ice4j.ice.harvest.DISABLE_AWS_HARVESTER=true
>
> org.ice4j.ice.harvest.STUN_MAPPING_HARVESTER_ADDRESSES=meet-jit-si-turnrelay.jitsi.net:443

Ces deux lignes vont permettre à Jitsi de récupérer l’adresse IP publique.

Création du certificat

On installe ensuite Certbot afin d’avoir un certificat valide.

On ajoute aux dépôts, on télécharge puis on installe :

```
add-apt-repository ppa:certbot/certbot
apt-get update
apt-get install letsencrypt -y
wget https://dl.eff.org/certbot-auto -P /usr/local/bin
chmod a+x /usr/local/bin/certbot-auto
export DOMAIN="jitsi.khroners.fr"
export EMAIL_ALERT=admin@khroners.fr
```

La commande chmod permet l'attribution de droits.

Pour les deux commandes export, on rentre le domaine complet et une adresse e-mail.

Pour mon domaine, je dois rentrer un enregistrement DNS, qui va pointer le FQDN « jitsee.khroners.fr » vers « khroners.fr. ».

]	jitsimeet.khroners.fr.	0	CNAME	khroners.fr.

On doit également ouvrir les ports et les rediriger vers la machine virtuelle Ubuntu avec le NAT. Dans mon cas sur une Livebox Orange :

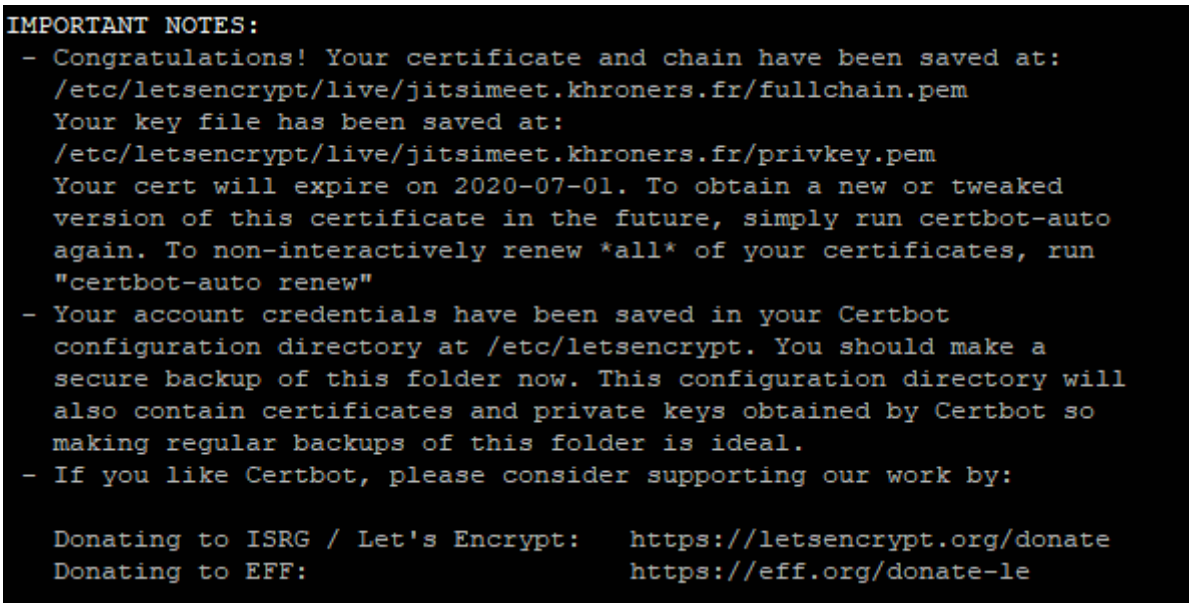
Jitsi	80	80	TCP	jitsimeet	☑
Jitsi2	443	443	TCP	jitsimeet	☑

Ici les ports 80 et 443 sont ouverts en direction de la machine virtuelle. Ces deux ports correspondent au protocole HTTP et HTTPS.

On ouvre également les ports 10000-20000 en UDP pour l'audio.

On peut ensuite rentrer cette commande qui va permettre la demande et la création du certificat TLS Let's Encrypt.

```
/usr/local/bin/certbot-auto certonly --standalone -d $DOMAIN  --preferred-challenges http --
agree-tos -n -m  $EMAIL_ALERT --keep-until-expiring
```



Le certificat est créé.

Le certificat est situé dans :

/etc/letsencrypt/live/jitsimeet.khroners.fr/fullchain.pem

La clé est située dans :

/etc/letsencrypt/live/jitsimeet.khroners.fr/privkey.pem

Modification de la configuration de nginx pour appliquer le certificat

On doit donc changer le chemin du certificat et de la clé dans le fichier config de nginx du site jitsi.

nano /etc/nginx/sites-available/jitsimeet.khroners.fr.conf

On modifie les lignes ssl_certificate et ssl_certificate_key.

```
server {
    listen 80;
    listen [::]:80;
    server_name jitsimeet.khroners.fr;

    location ^~ /.well-known/acme-challenge/ {
       default_type "text/plain";
       root         /usr/share/jitsi-meet;
    }
    location = /.well-known/acme-challenge/ {
       return 404;
    }
    location / {
       return 301 https://$host$request_uri;
    }
}
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name jitsimeet.khroners.fr;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_ciphers "EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA

    add_header Strict-Transport-Security "max-age=31536000";

    ssl_certificate /etc/letsencrypt/live/jitsimeet.khroners.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/jitsimeet.khroners.fr/privkey.pem;

    root /usr/share/jitsi-meet;

    # ssi on with javascript for multidomain variables in config.js
    ssi on;
    ssi_types application/x-javascript application/javascript;
```

On redémarre le serveur nginx.

```
service nginx restart
```

En cas d’erreur lors du redémarrage de nginx

```
avril 02 20:49:29 jitsimeet systemd[1]: Starting A high performance web server and a reverse proxy server...
avril 02 20:49:29 jitsimeet nginx[1262]: nginx: [emerg] bind() to 0.0.0.0:443 failed (98: Address already in use)
avril 02 20:49:30 jitsimeet nginx[1262]: nginx: [emerg] bind() to 0.0.0.0:443 failed (98: Address already in use)
avril 02 20:49:30 jitsimeet nginx[1262]: nginx: [emerg] bind() to 0.0.0.0:443 failed (98: Address already in use)
avril 02 20:49:31 jitsimeet nginx[1262]: nginx: [emerg] bind() to 0.0.0.0:443 failed (98: Address already in use)
avril 02 20:49:31 jitsimeet nginx[1262]: nginx: [emerg] bind() to 0.0.0.0:443 failed (98: Address already in use)
avril 02 20:49:32 jitsimeet nginx[1262]: nginx: [emerg] still could not bind()
avril 02 20:49:32 jitsimeet systemd[1]: nginx.service: Control process exited, code=exited status=1
avril 02 20:49:32 jitsimeet systemd[1]: nginx.service: Failed with result 'exit-code'.
avril 02 20:49:32 jitsimeet systemd[1]: Failed to start A high performance web server and a reverse proxy server.
```

```
lsof -iTCP -sTCP:LISTEN
```

Avec cette commande, on peut voir les processus utilisants les différents ports. Cependant, ici, aucun n’utilise ce port 443.

Une erreur est connu (https://community.jitsi.org/t/nginx-coturn-port-443/27820 ,

https://community.jitsi.org/t/bind-to-0-0-0-0-443-failed/28615) en ce moment depuis la dernière version stable de Jitsi. Un fichier rentre en conflit avec nginx, ne permettant pas à nginx d'écouter le port 443 (HTTPS). Il faut donc supprimer ce fichier.

On copie le fichier avant la suppression puis on le supprime.

```
cp /etc/nginx/modules-enabled/60-jitsi-meet.conf /home/user
rm /etc/nginx/modules-enabled/60-jitsi-meet.conf
```

On redémarre le serveur web.

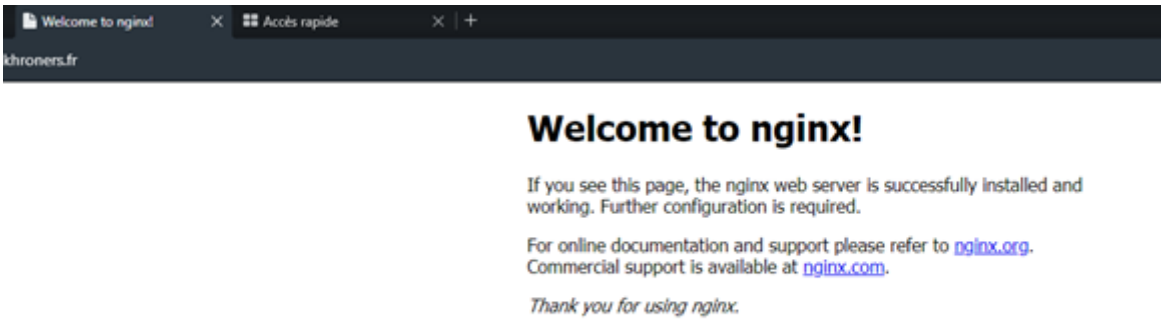
```
service nginx restart
```

Accès à Jitsi, vérification du certificat & différents tests

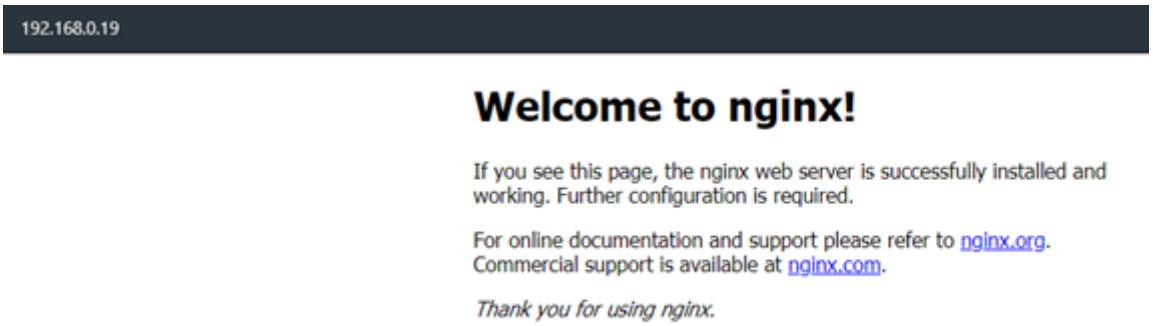
Accès à Jitsi

Jitsi est désormais accessible via l'adresse https://jitsimeet.khroners.fr/

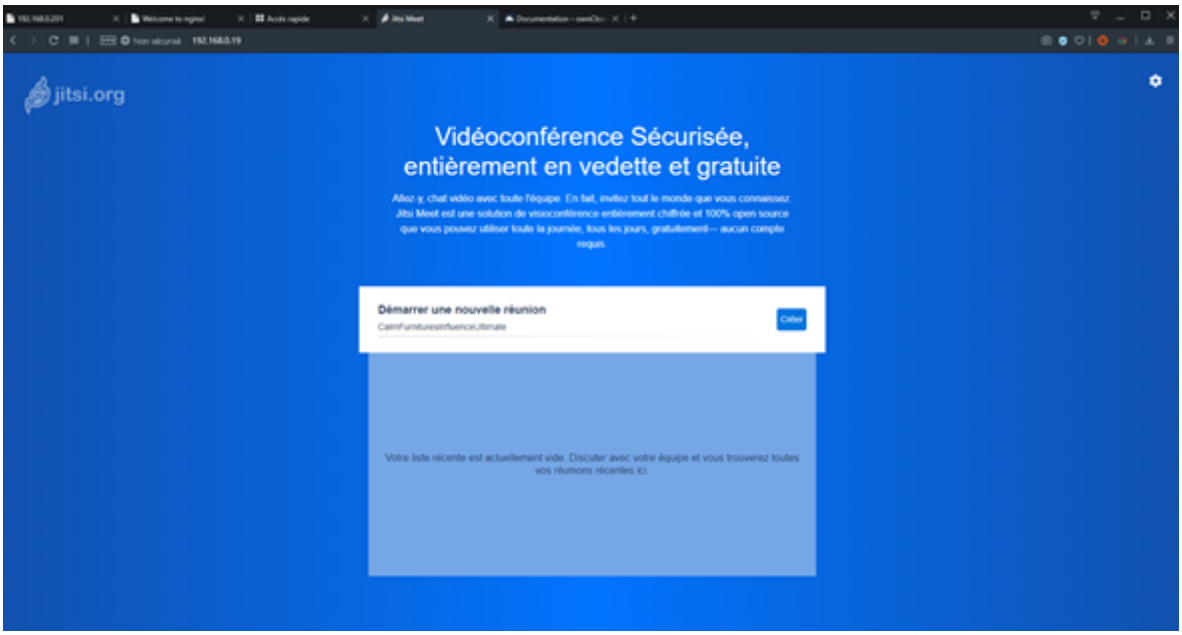
Via le domaine en HTTP :



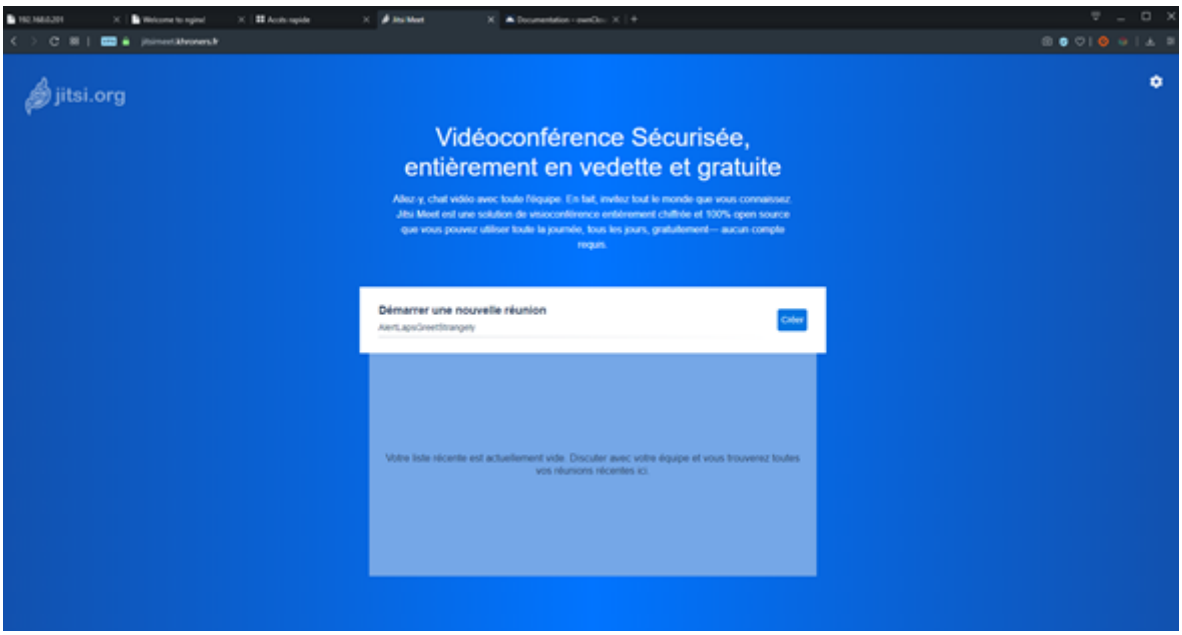
Via l'adresse IP locale en HTTP :



En HTTPS sur l’adresse IP locale du serveur Jitsi :

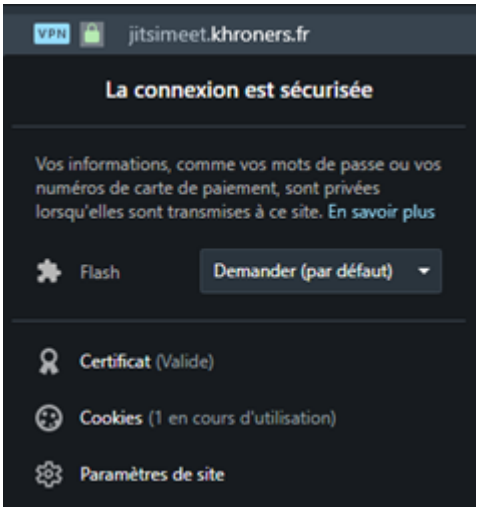


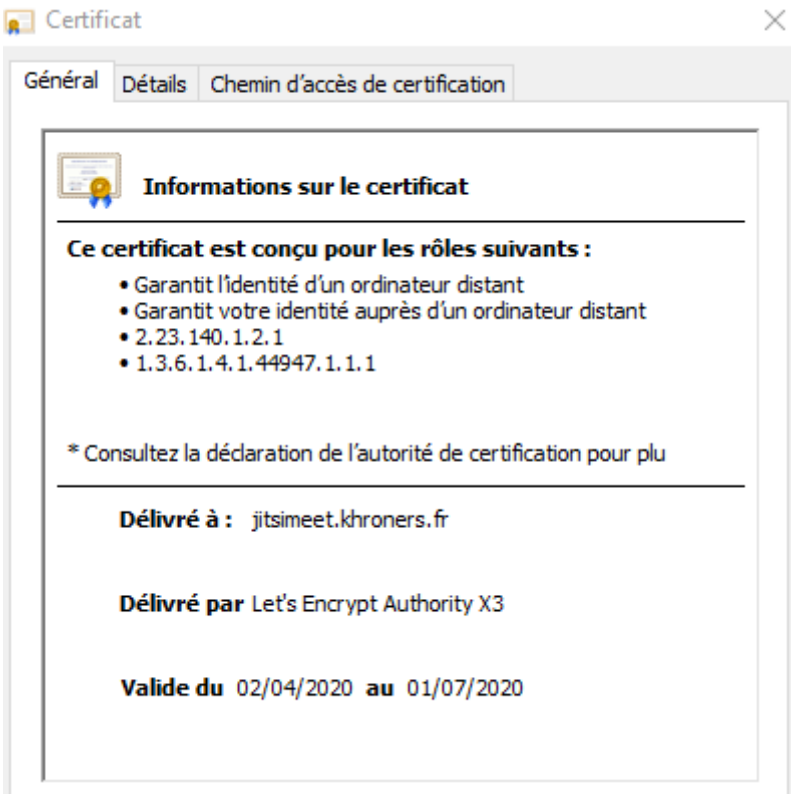
En HTTPS sur le domaine :



Vérification du certificat

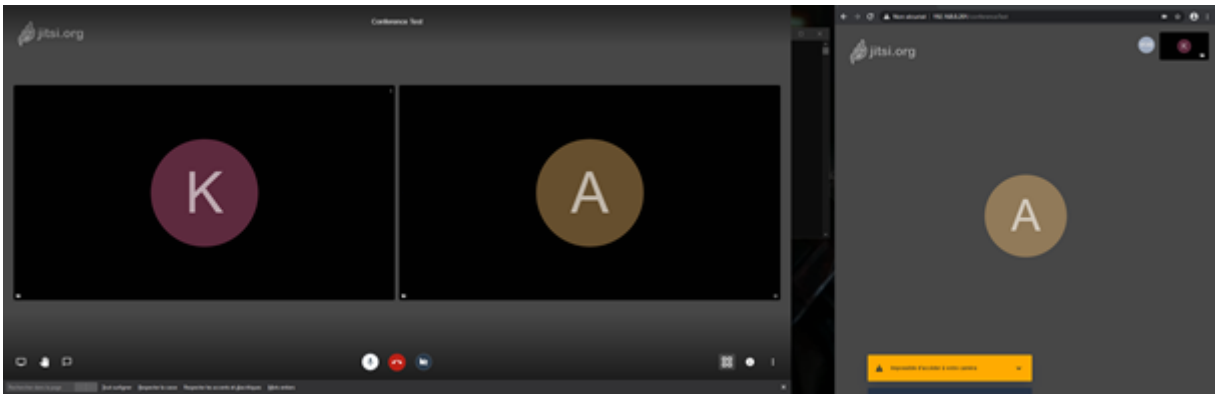
Le certificat valide :





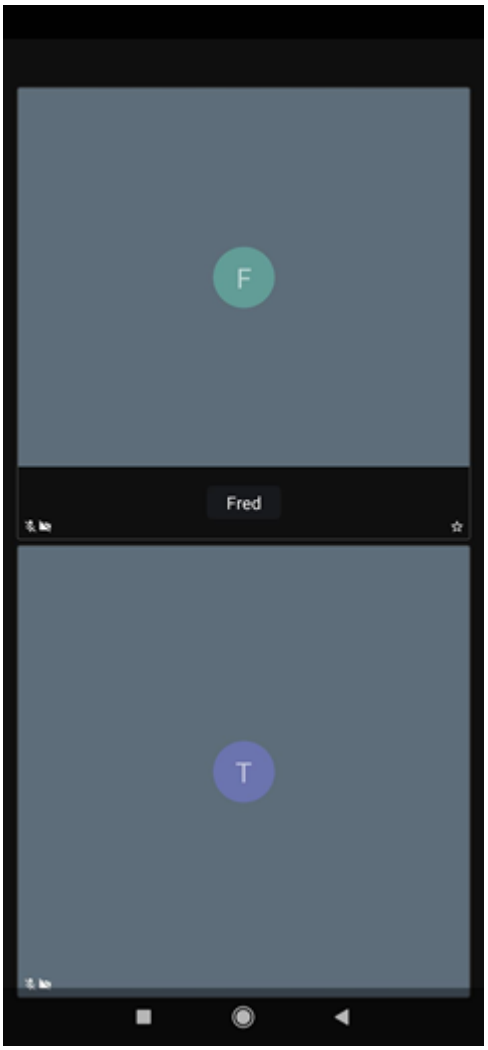
Tests

Voici un test entre deux utilisateurs en local :

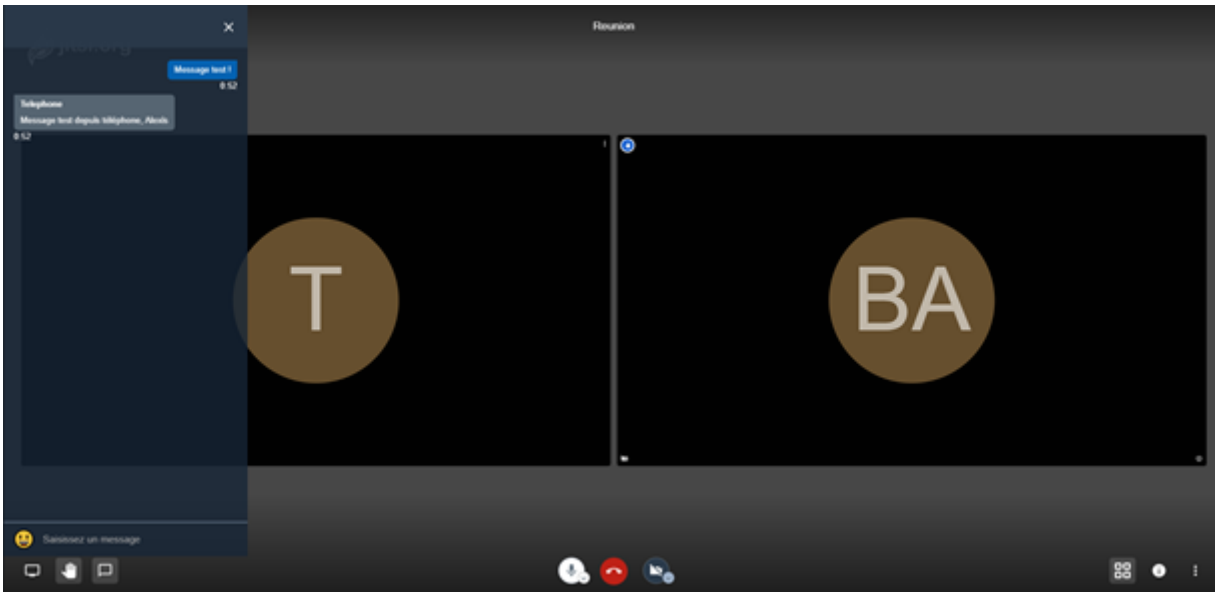


A gauche, l’utilisateur avec l’avatar K et à droite l’utilisateur avec l’avatar A.

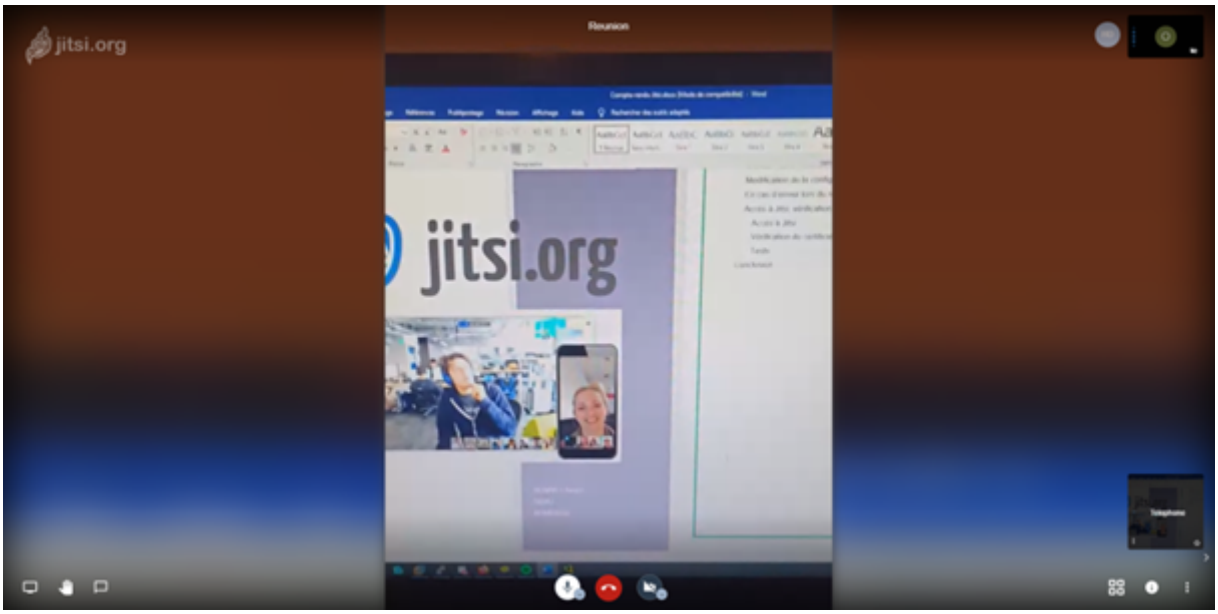
Test entre deux utilisateurs mobiles en 4G :



Test entre un ordinateur et un téléphone :



Deuxième test entre un ordinateur et un téléphone (capture d’écran depuis l’autre utilisateur) :



Conclusion

Jitsi est opérationnel. On peut alors faire des visioconférences à plusieurs, sécurisées via l’HTTPS et chiffrées. Cet outil étant gratuit est parfait pour les petites entreprises et disponible sur de nombreux appareils différents : Smartphones, tablettes, PC, Mac…


Innovative Video
Conferencing