**Jour-1 Tercium : Infomaniak.cloud // Paramétrage // Instanciation**

**TERCIUM : Une entreprise constituée de micro-entreprises dans divers domaines du digital. (à détailler). Il y a au total 10 personnes réalisant des projets dans divers secteurs.**

* **C**elle-ci possède ses locaux et ses équipements en propre comme ses **PC** et ses **NAS.** Chacun étant relier aux équipes dédiées.
* Celle-ci ne stocke pas ses données dans ses propres serveurs mais les délocalisent dans une ferme de serveur Suisse qui ont une politique éthique (RSE) stricte : **Infomaniak**

Objet du stage : Installation de Jitsi un software de visioconférence. Ces besoins en capacité seront calculés par rapport aux besoins spécifiques.

* Website Jitsi : <https://jitsi.github.io/handbook/docs/architecture>

**Pourquoi faire des visioconférences ?**

**Il n’est pas toujours possible de se rendre à une réunion pour diverses raisons comme la distance ou un empêchement quelconque.**

**Pour faire des conversations en**[**webcam**](https://cours-informatique-gratuit.fr/dictionnaire/webcam/)**, rencontrer plusieurs personnes, de partager des**[**documents**](https://cours-informatique-gratuit.fr/dictionnaire/fichier/)**, des**[**liens URL**](https://cours-informatique-gratuit.fr/dictionnaire/url/)**ou votre**[**écran**](https://cours-informatique-gratuit.fr/dictionnaire/ecran/)**pour donner plus d’explications à vos interlocuteurs, il vous faudra utiliser un outil de visioconférence.**

**Zoom : Standard actuel et proposition commerciale**

* [**Zoom Spaces**](https://explore.zoom.us/fr/products/zoom-rooms/)**est un moyen innovant d’organiser des réunions virtuelles au sein d’équipes hybrides. Les services inclus sont la réservation d’un espace de travail (par exemple, un bureau ouvert, un dispositif Zoom), un connecteur de salle de conférence pour rejoindre les réunions avec un équipement SIP/H.323, et des salles Zoom pour les conférences qui impliquent de nombreuses personnes des deux côtés.**
* [**Zoom Events**](https://explore.zoom.us/fr/products/event-solutions/)**est une solution dédiée à l’organisation d’événements virtuels, des annonces internes aux tables rondes. Cette option est idéale pour le réseautage, la présentation de contenu dans des stands d’exposition, la formation d’équipes dans des événements à session unique et des webinaires en ligne.**
* [**Zoom Contact Center**](https://explore.zoom.us/fr/products/contactcenter/)**vous permet de créer un environnement omnicanal pour fournir une assistance à vos clients. Les outils disponibles comprennent un centre de contact optimisé pour la vidéo dans le cloud et l’IA conversationnelle.**

**La concurrence OpenSource :**

**Jitsi Meet est une solution de visioconférence multiplateforme (Windows, macOS, Linux, Android, iOS) qui brille par sa modularité et son efficacité. Conçu pour les secteurs professionnels, éducatifs et médicaux, Jitsi offre une alternative open source robuste aux services payants comme Zoom ou Microsoft Teams.**

**Hébergé sur un cloud multirégional, Jitsi garantit une expérience audio et vidéo fluide, même pour des utilisateurs géographiquement dispersés. Soutenue par la communauté tech et des experts en sécurité, l’application continue de se distinguer en étant entièrement gratuite et hautement personnalisable.**

**Une sécurité, qui n’est pas un détail, est l’ajout d’un mot de passe à l’appel. Cela empêche à la fois une intrusion inopportune à une réunion ainsi que l’impossibilité à celle-ci de bloquer les appels.**

* **Usage de l’icône i afin d faire apparaître les informations de la salle virtuelle de la réunion.**
* **Envoi via un courriel ou d’un SMS de l’URL avec le mot de passe.**
* **Comme zoom il suffit de placer le lien dans un navigateur ; il est recommandé d’utiliser Chrome ou Firefox.**

**La capacité a évolué est permet maintenant de gérer une réunion de 500 participants.**

**Au-delà Jitsi a un service payant : JaaS permettant une réunion de 10 000 participants.**

**Particularités :**

* **Comme ses concurrents payants anciennement Skype puis Zoom,Teams, Whereby, eyeson,,hangouts Meet ; Jitsi peut partager l’écran, enregistrer et passer en mode vignette.**
* **L’application mosaïque en est le parfait exemple.**
* **Le contrôle de sons environnement, comprendre le respect de la vie privée par l’escamotage de son espace privatif par l’usage d’un arrière-plan flou ou fictif est une fonction (usage de trois petits points).**
* **L’enregistrement de la session est possible pour une diffusion indépendante du flux avec YouTube.**
* **Celui-ci peut être aussi réaliser, l’enregistrement via Dropbox avec un compte basique gratuit, pour ce faire il faut comme précédemment utiliser les trois petits points.**

Besoins matériels :

* <https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-requirements/>

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Usage & Compatibilité :

* <https://linuxfr.org/users/lebouquetin/journaux/organiser-des-visioconferences-de-haute-qualite-avec-le-logiciel-libre-jitsi-meet>

Réseau :

* OpenRC peut être utiliser pour gérer le réseau :

<https://www.linuxtricks.fr/wiki/openrc-gestion-du-reseau>

**Aborder Infomaniak : Cloud Provider** Suisse

<https://www.infomaniak.com/fr/support/faq/2601/guide-de-demarrage-public-cloud>

Infomaniak [Public Cloud](https://infomaniak.com/gtl/hosting.public-cloud), une solution [Infrastructure\_as\_a\_service (IaaS)](https://fr.wikipedia.org/wiki/Infrastructure_as_a_service" \t "_blank) basée sur [OpenStack](https://docs.openstack.org/yoga/" \t "_blank) qui met à disposition les ressources dont vous avez besoin pour le développement de vos projets.

**Démarche pour une instanciation :**

**Inscription & accès :** [**https://api.pub2.infomaniak.cloud/horizon/auth/login/**](https://api.pub2.infomaniak.cloud/horizon/auth/login/)

**Pages explicatives :** <https://docs.infomaniak.cloud/compute/>

Create a keypair :

Une image contenant texte, Police, algèbre, reçu

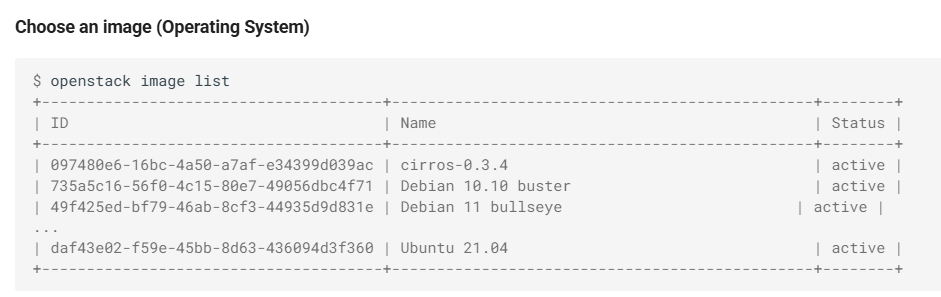
Le contenu généré par l’IA peut être incorrect.

Create an instance :

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

**Choose an image (operating System):**



**Exemple de création :**

**Une image contenant texte, capture d’écran, logiciel, nombre

Le contenu généré par l’IA peut être incorrect.**

**Contrôle de création d’Instance:**

****

**Firewall :**

**Une image contenant texte, capture d’écran, Police, algèbre

Le contenu généré par l’IA peut être incorrect.**

**Clef SSH : modèle**

****

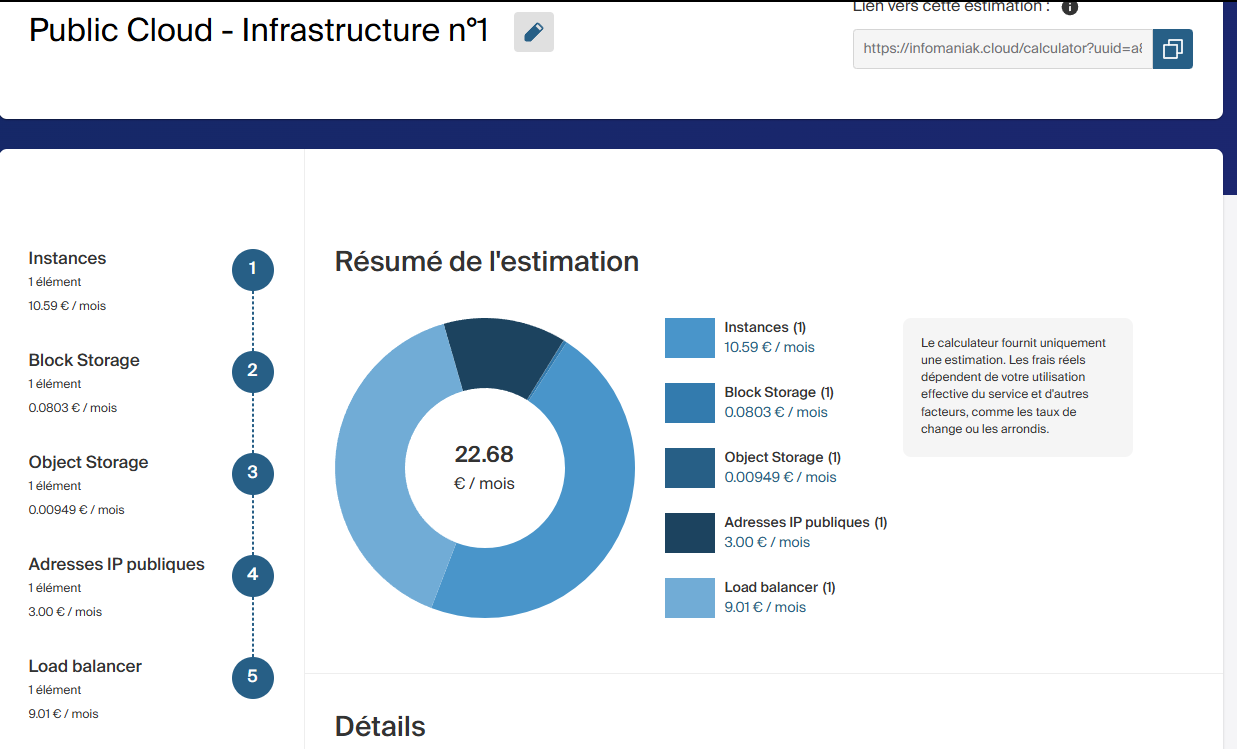
**Availability Zones :** openstack availability zone list

Une image contenant texte, Police, capture d’écran, reçu

Le contenu généré par l’IA peut être incorrect.

**Coût d’une instanciation pour la compagnie : Infomaniak permet la simulation.**

**Ci-dessous les choix matériels.**

****

**Détails :**

**Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.**

**Rappel :**

**Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

**Choix des availability Zone : az-1 / az-2 / az-3**

**Ces trois zones sont des zones matérielles dans la ferme à serveur.**

**La région sera dc4-a : La région dc4-a utilisée par Infomaniak Public Cloud fait référence à un datacenter situé en Suisse, plus précisément dans la zone Genève / Satigny (canton de Genève).**

**Putty Key Generator : génére deux clefs RSA, une publique et une privée.**

**Je peus faire des imports dans le launch Instance fenêtre de Infomaniak.**

**Putty : me permettre en connexion, avec l’interface Putty.**

**Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

Dans d’autres versions de **Putty** il est possible d’exporter la clef publique directement via la barre de menu, Conversions en format **OpenSS**.

**Ici je copie colle** la clef publique directement de **Putty** pour la transmettre à l’interface de création d’instance de d’infomaniak.

Dans l’impossibilité le pis allé a été de générer un fichier txt / bloc-note. Puis de l’appeler dans l’interface Infomaniak. Ce qui se déroule en deux temps.

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

**Jour-2 Tercium : Instanciation Réussie :**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Il est plus aisé cependant de faire générer ses clefs via le terminal / console de VSCode : Premier problème l’authentification :**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Usage de nano pour modifier les permissions :**

**cd :** **sudo nano /etc/ssh/sshd\_config Action : PasswordAuthentication no**

**Renforcer la sécurité (recommandé) :**

**bash**

**CopyEdit**

**sudo nano /etc/ssh/sshd\_config**

* **Modifie ou décommente :**

**conf**

**CopyEdit**

**PasswordAuthentication no**

**PermitRootLogin prohibit-password**

* **Puis redémarre le service :**

**bash**

**CopyEdit**

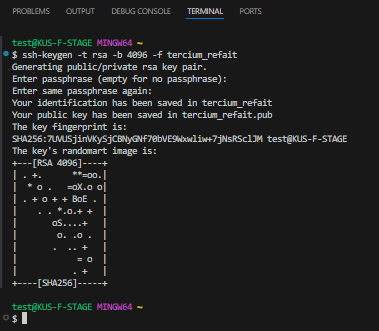
**sudo systemctl restart ssh**

**Via VScode je vais pouvoir faire des commandes administrateur.**

* **Générer et valider des droits d’usages avec chmod.**
* **Créer des dossiers : mkdir, des fichiers touch pour y insérer les clefs des instances.**

**Via le terminal Je recrée des clefs : ssh-keygen -t rsa -b 4096 -f tercium\_refait\_key** qui sera le nom du fichier la contenant. Extension.pub pour la clef publique et la privée sans extension. Powershell n’étant pas le seul terminal que j’utilise, je travaille aussi avec GitBash pour avoir une ligne de commande puissante.

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect. **

**Verrouillage d’une IP :**

**Host terciump**

**HostName 37.156.45.22**

**User ubuntu**

**IdentityFile ~/Documents/Tercium\ Stage/tercium-instance\_key/tercium-instance\_key**

**Connexion standard : ssh -i ~/.ssh/tercium\_key** [**ubuntu@84.234.28.241**](mailto:ubuntu@84.234.28.241)

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**cd : direction instance**

**ssh -i "~/Documents/Tercium Stage/test3\_instance\_key/test3publickey\_refait\_key"** [**ubuntu@84.234.28.241**](mailto:ubuntu@84.234.28.241)

**Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.**

**Sur VsCode on sort de l’instance du terminal :ctrl + D**

**Infomaniak.cloud : interface de contrôle des instances**

**Une image contenant texte, capture d’écran, ligne, Police

Le contenu généré par l’IA peut être incorrect.**

**Maintenant étapes de contrôle de connexion via les Adresses IP pour un poste extérieur. Installation d’un serveur Apache pour Linux : LAMP**

**cd : sudo apt update && sudo apt install apache2 php libapche2-mod-php -y**

**On Active et démarre Apache :**

**cd : sudo systemctl enable apache2**

**sudo systemctl start apache2**

**Côté serveur : Apache est-il bien démarré ? sudo systemctl status apache2**

**Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

* **Écoute-t-il sur les bons ports ? sudo netstat -tlnp | grep :80**

**La version moderne de netstat => sudo ss -tlnp | grep :80**

* **Côté réseau/firewall :**

**Le firewall du serveur autorise-t-il le trafic sur les ports 80/443 sur infomaniak ?**

**Infomaniak a des règles de sécurité/firewall à configurer dans Security Group :**

**Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.**

**Test simple : Créez un fichier index.html basique dans /var/www/html/ et tentez d'y accéder via** [**http://VOTRE\_IP/**](http://VOTRE_IP/)

**Via le terminal écrire :**

**cd : echo "<h1>OK LAMP / INFOMANIAK</h1>" | sudo tee /var/www/html/index.html**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Explication du code, qui fait deux choses :**

1. **echo produit la chaîne : "<h1>OK LAMP / INFOMANIAK</h1>"**
2. **tee :**
   * **écrit cette chaîne dans le fichier /var/www/html/index.html**
   * **et réimprime sur l’écran (stdout) ce qu’elle écrit.**

**Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Jour-3 Tercium :**

**A / Fingerprint : Explication structurée issue de ChatGPT**

**Dans le contexte d’OpenStack (ici via l’interface *Infomaniak Public Cloud*), le fingerprint SSH associé à chaque paire de clés a les usages suivants :**

**Usage du Fingerprint SSH**

**1. Vérification d’authenticité**

* Le *fingerprint* est un empreinte condensée (hash) de la clé publique associée à une paire SSH.
* Il permet de vérifier rapidement que la clé utilisée pour se connecter à une machine correspond bien à celle attendue, sans afficher toute la clé (souvent longue).

**2. Identification rapide des clés**

* Permet de différencier deux clés ayant le même nom ou des noms similaires (ex : Public key, Public-key).
* Ex : **tercium-instance\_key** et **terciup-instance\_key** ont la même empreinte → c’est exactement la même clé.

**3. Sécurité (anti-usurpation)**

* Lorsqu’un utilisateur ou script tente une connexion SSH, le système peut comparer le fingerprint local (de .ssh/known\_hosts) à celui enregistré dans OpenStack pour valider l’identité**.**

**4. Audit / Journalisation**

* En environnement multi-utilisateur ou en audit de sécurité, l’empreinte permet de tracer quelle clé a été utilisée, sans afficher ou stocker la clé complète.

**Format de l’empreinte :**

* **Généralement SHA-1 (comme ici), parfois SHA-256 selon la configuration.**
* **Exemple : ad:31:ae:a1:76:9d:29:de:65:71:fb:07:c2:d9:6f:98**

**Cas spécifique :**

* tercium-instance\_key et terciup-instance\_key ont exactement le même fingerprint : Cela indique que la même clé publique a été réutilisée pour plusieurs instances.

**✅ Bonnes pratiques**

| **Recommandation** | **Description** |
| --- | --- |
| **Générer une clé unique par instance** | **Pour éviter des accès non maîtrisés entre VM** |
| **Stocker et vérifier les fingerprints dans un**  **fichier de suivi** | **Par exemple inventory.md ou dans un dépôt Git** |
| **Comparer fingerprint local / cloud** | **Pour éviter les erreurs de clé en cas de multiples jeux** |

**B / Test de connexion simple : Ping avec le terminal sans connexion direct :**

**Une image contenant texte, capture d’écran, Police, conception

Le contenu généré par l’IA peut être incorrect.**

**C / 2nd Test simple : Créez un fichier index.html basique dans /var/www/html/ et tentez d'y accéder via** [**http://VOTRE\_IP/**](http://VOTRE_IP/)

**Pour se connecte via son terminal il faut avoir activer son agent SSH avec sa clef privée :**

* **Se positionner dans le dossier maître :**

**ils se situe dans C:\Users\test\Documents\Tercium\_Stage\Tercium-instance\_key**

* **Puis Naviguer dans le dossier contenant les deux fichiers Public\_key & Private\_Key.pub.**

**cd : ssh -i Tercium-instance\_key/tercium-instance\_key** [**ubuntu@84.234.28.98**](mailto:ubuntu@84.234.28.98)

**si l’on oublie l’adresse IP : Résultat**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Résumé des processus à respecter et comprendre :**

1. **Chemin correct** de la clé privée :
2. ssh -i Tercium-instance\_key/tercium-instance\_key ubuntu@84.234.28.98
3. **Clé privée avec permissions sécurisées** (si besoin) :
4. chmod 600 Tercium-instance\_key/tercium-instance\_key
5. **Pas besoin de ssh-add** dans ce cas : ssh -i suffit pour l’authentification directe.
6. **Pas de session root à maintenir ouverte pour SSL** : les certificats sont **posés une fois** (ex. via certbot) et renouvelés **automatiquement** en tâche de fond.

Trois éléments critiques à documenter dans le **manuel d’usage** :

**✅ Résumé à intégrer dans ton rapport (section SSH / accès distant)**

1. **Structure des fichiers :**
   * Dossier : Tercium-instance\_key/
     + Fichier : tercium-instance\_key → **clé privée**
     + Fichier : tercium-instance\_key.pub → **clé publique**
2. **Connexion réussie via Git Bash :** ssh -i Tercium-instance\_key/tercium-instance\_key [ubuntu@84.234.28.98](mailto:ubuntu@84.234.28.98) **/ à chaque instance détruite / recréer, l’IP adresse change**.
3. **Écueils à éviter :**
   * Ne pas confondre **répertoire** et **fichier de clé**
   * La commande ssh-add est inutile si on utilise directement ssh -i
   * Vérifier les permissions (chmod 600) si accès refusé
   * Le format Windows (\) ne fonctionne pas dans Git Bash, utiliser / ou ~/Documents/...

**Proposition bonus**

**Créer un alias dans Git Bash pour ne pas avoir à taper toute la ligne :**

**echo "alias ssh-tercium='ssh -i ~/Documents/Tercium\_Stage/Tercium-instance\_key/tercium-instance\_key ubuntu@84.234.28.98'" >> ~/.bashrc**

**source ~/.bashrc**

**Ensuite, il te suffira de taper : ssh-tercium**

**Vos pages HTML sont-elles dans /var/www/html/ ?**

**Ayant détruit l’instance il n’est plus possible d’obtenir quelque chose qui a été effacée.**

**Pour repartir proprement, séquence minimale sur Ubuntu (cloud ou local) pour reconstruire :**

**sudo apt update**

**sudo apt install apache2 -y**

**echo "<h1>Site opérationnel</h1>" | sudo tee /var/www/html/index.html**

**sudo systemctl restart apache2**

**D / Test local :**

* **En local :** [**http://localhost**](http://localhost)

**Tester le serveur Apache localement (via WSL2 à installer au préalable via powershell)**

**cd : wsl --list –verbose // Vérification si WSL est installé dans VSCode**

**Sinon : wsl --install -d Ubuntu**

**Aussi Installer l’extension “Remote - WSL” dans VS Code Extensions (Ctrl+Shift+X) → cherche : nginx CopyEdit Remote - WSL**

**Lancer un projet ou un terminal dans WSL via VS Code**

* **Ouvrir la palette de commandes Ctrl+Shift+P**
* **Tape : WSL: New WSL Window**
* **Un nouveau VS Code va s’ouvrir avec le terminal WSL actif**
* **Navigue dans le projet (cd /mnt/c/Users/...) ou clone un dépôt**

**Intérêts :**

* **Ecrire depuis Windows mais exécutes sur Linux réel.**
* **Faire sudo, apt, systemctl, etc.**
* **Ouvrir des projets en un clic avec code . dans WSL**

**Test en local :** [**http://localhost**](http://localhost) **ou plus précisément** [**http://localhost:80**](http://localhost:80)

**Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant texte, Logiciel multimédia, logiciel, Logiciel de graphisme

Le contenu généré par l’IA peut être incorrect.**

**Si en cloud : http://<ip\_publique\_VM> déjà réaliser.**

**E / Certificat SSL via Cerbot !**

**Synthèse du document LE-SA-v1.5 (Let’s Encrypt Subscriber Agreement) :**

**Daté du 24 février 2025, utile pour toute implémentation réelle ; en entreprise, audit ou projet certbot / HTTPS.**

**Objectif du document :**

**Contrat juridique entre toi (ou ton organisation) et ISRG (Internet Security Research Group), organisme émetteur des certificats Let’s Encrypt via le protocole ACME.**

**1. Définitions clés à retenir**

* **ACME :** Protocole automatisé de gestion des certificats**.**
* **Certificat :** Lien validé entre un nom de domaine et une clé publique.
* **Key Pair :** Paire de clés asymétriques **(privée / publique).**
* **Private Key compromise :** Clé privée compromise ou à risque => certificat à révoquer.

**2. Conditions d’entrée en vigueur**

* **Accord effectif** dès que tu demandes un certificat **Let’s Encrypt (même via ACME).**
* Il reste valide tant que tu possèdes un certificat actif, même s’il est renouvelé automatiquement.

**3. Engagements du souscripteur :**

**Tu garantis :**

* Que tu es légitime sur le domaine visé.
* Que tu n’as pas obtenu ce domaine illégalement.
* Que tu protèges ta clé privée.
* Que les infos du certificat sont exactes, à jour et sincères.
* Que tu révoques immédiatement le certificat si :
  + la clé est compromise,
  + le domaine t’échappe,
  + ou les données sont obsolètes.

**4. Gestion technique**

* Le certificat est généré à partir des infos envoyées par ton client ACME **(e.g., certbot).**
* Tu dois vérifier les infos avant usage.
* Tu as le droit d’installer le certificat uniquement sur les serveurs mentionnés dans le champ **subjectAltName.**

**5. Usage interdit**

* **Écoute active (attaque MITM)**
* **Interception ou redirection de trafic non autorisé**
* **Toute architecture facilitant la violation de la confidentialité HTTPS**

**6. Révocation**

* **Tu dois révoquer un certificat via l’API ACME si :**
  + clé compromise,
  + changement de domaine,
  + données fausses.
* **ISRG peut révoquer sans ton accord pour :**
  + **usage frauduleux,**
  + **décision judiciaire,**
  + **certificat incorrect ou détourné.**

**7. Clause de non-responsabilité (ISRG)**

* Let’s Encrypt est un service gratuit sans garantie contractuelle.
* ISRG décline toute responsabilité en cas :
  + de perte,
  + de poursuite,
  + de dommage technique ou juridique.

**8. Droit applicable**

* **Loi de Californie.**
* **Tribunal compétent : San Jose, CA.**
* **Aucune tierce partie n’a de droit par ce contrat.**
* **Limite d’action légale : 1 an.**

**En résumé pour usage réel :**

| **Action** | **Obligation** |
| --- | --- |
| **Génération des clés** | **En local, jamais par ISRG** |
| **Protection de la clé privée** | **Obligatoire (clé = confidentielle)** |
| **Vérification du certificat** | **Avant toute utilisation** |
| **Révocation** | **Immédiate en cas de doute ou perte de contrôle** |
| **Usage** | **Limité au domaine validé, usage HTTPS standard uniquement** |
| **Comportement interdit** | **MITM, reroutage de trafic, spoofing** |

**Avec WSL en tant qu’administrateur :**

**cd : sudo apt update && sudo apt install cerbot python3-certbot-apache y**

**Jour-4 Tercium**

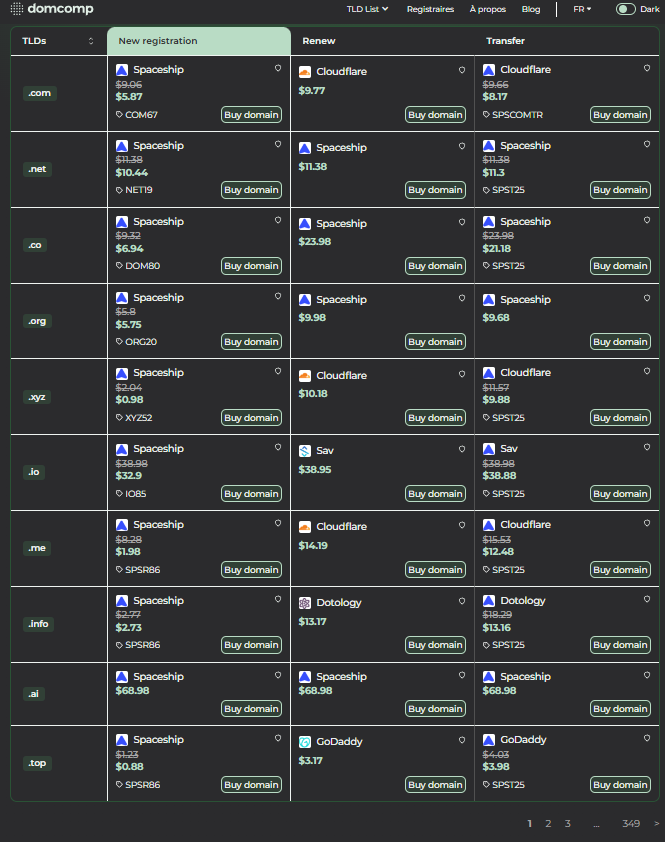
**A / Informations quant au coût évaluation exemple choisi : Jitsi.tercium.xyz**

**Coûts complets et implications techniques pour le nom de domaine jitsi.tercium.xyz, y compris la partie certification SSL/TLS. Voici la synthèse structurée, point par point :**

**1. Nom de domaine ; Jitsi.tercium.xyz**

**▸ Enregistrement :**

* Registrar : **Spaceship**
* Coût initial : **$0.98 (≈ 0.91 €)**
* Renouvellement : **$10.18 annuelle**
* Transfert : **$9.88**  
  Total pour 1 an (base) : ≈ 1 € pour l’enregistrement + SSL gratuit via Certbot (Let’s Encrypt) = ≈ 1 € initialement, renouvellement ≈ 10 €

****

**2. Certificat SSL/TLS via Let’s Encrypt**

**▸ Certbot avec Apache :**

* **Commande (déjà mentionnée dans ton journal) :**

**sudo apt update**

**sudo apt install certbot python3-certbot-apache -y**

1. Configuration automatique : certbot configure Apache et obtient un certificat gratuitement via ACME.
2. Renouvellement automatique : inclus (/etc/cron.d ou via systemd timer), sans coût**.**

**▸ Conditions :**

* Tu dois posséder le domaine (tercium.xyz)
* Le port 80 (HTTP) doit être ouvert pour validation ACME
* Le certificat est valide 90 jours, renouvelable

**3. Instance cloud utilisée**

**D’après capture :**

* **Type : a4-ram4-disk80-perf1**
  + **4vCPU**
  + **8 GB RAM**
  + **20 GB disque SSD**
* **OS : Ubuntu 24.04 Kube v1.30 (image publique, QCOW2)  
  ➡️ Suffisant pour héberger Jitsi, mais attention au coût horaire chez Infomaniak**

**4. Configuration Apache pour Jitsi**

**Si tu dédies un sous-domaine jitsi.tercium.xyz :**

* **Ajouter un VirtualHost Apache :**

<VirtualHost \*:80>

ServerName jitsi.tercium.xyz

DocumentRoot /var/www/jitsi

</VirtualHost>

* Et SSL après certbot --apache :

<VirtualHost \*:443>

ServerName jitsi.tercium.xyz

SSLEngine on

SSLCertificateFile /etc/letsencrypt/live/jitsi.tercium.xyz/fullchain.pem

SSLCertificateKeyFile /etc/letsencrypt/live/jitsi.tercium.xyz/privkey.pem

</VirtualHost>

**5. Checklist récapitulative**

| **Étape** | **Action technique** | **Coût** |
| --- | --- | --- |
| **Nom de domaine tercium.xyz** | **Achat sur Spaceship** | **~1 €** |
| **Création sous-domaine jitsi.\*** | **Ajout dans DNS (type A vers IP publique)** | **0 €** |
| **Serveur Apache / installation Jitsi** | **Apache + Jitsi install sur Ubuntu 24.04** | **0 €** |
| **Ouverture port 80 / 443** | **Configuration Security Groups Infomaniak** | **0 €** |
| **Certificat TLS** | **Certbot + Let’s Encrypt (ACME)** | **0 €** |
| **Renouvellement automatique SSL** | **Certbot cron ou systemd timer** | **0 €** |
| **Renouvellement domaine .xyz** | **Annual fee Spaceship** | **~10 €** |

**Conclusion :**

* **Domaine tercium.xyz** : 0.98 $ la première année
* **SSL Let’s Encrypt :** 0 €
* **Infomaniak** (coût instance selon durée d’usage)
* Possibilité de pointer **jitsi.tercium.xyz** via un enregistrement **DNS** vers l’**IP** publique
* **Jitsi Meet** peut être installé et certifié gratuitement si ton reverse proxy Apache est bien configuré

**Calcul Gobal :**

**Coût d’une instance : Infomaniak // https://www.infomaniak.com/fr/domaines https://hellosafe.fr/hebergeurs/infomaniak**

1. **Domaine réel  : visio.workeezconnect.fr**
2. **Choix de l’OS :**

****

1. **Achat d’un volume / basique puis extensible.**

****

1. **IP fixe ou flottante : une seule adresse menant vers le serveur Jitsi**
2. **Instance basique : moindre coût**

**Pour calculer le coût réel d'une instance cloud déployée comme Jitsi, surtout chez Infomaniak Public Cloud (OpenStack), tu dois prendre en compte plusieurs paramètres facturables à l’usage. Voici la méthode de calcul complète et générique, suivie de son application à ton cas (instance Jitsi jusqu’au 19 août).**

**1. Paramètres de tarification à surveiller (chez Infomaniak)**

Infomaniak Public Cloud facture à la minute (via OpenStack), sur les bases suivantes :

| **Composant** | **Base de coût** | **Unité** |
| --- | --- | --- |
| **vCPU** | **€ par vCPU par heure** | **€/vCPU/h** |
| **RAM** | **€ par GB RAM par heure** | **€/GB/h** |
| **Stockage disque** | **€ par GB par mois** | **€/GB/mois** |
| **Trafic sortant** | **€ par Go sortant par mois** | **€/Go/mois** |
| **Adresse IP fixe** | **parfois facturée séparément** | **€/IP/mois** |

**2. Application à ton instance actuelle : Instance utilisée** : **a4-ram8-disk20-perf1**

| **Ressource** | | **Valeur** | |
| --- | --- | --- | --- |
| **vCPU** | | **4** | |
| **RAM** | | **8 GB** | |
| **Disque** | | **20 GB SSD** | |
| **Durée d’usage** | | **du 03/07 au 19/08 = 47 jours ≈ 1.53 mois** | |

**Estimation du coût pour cette instance (Infomaniak Public Cloud) :**

| **Ressource** | **Tarif unitaire estimé** | **Calcul sur 47 jours (1.53 mois)** | **Coût** |
| --- | --- | --- | --- |
| **vCPU (4)** | **0.01 €/vCPU/h = 0.04 €/h** | **0.04 × 24 h × 47 j = 45.12 €** | **45 €** |
| **RAM (8 GB)** | **0.01 €/GB/h = 0.08 €/h** | **0.08 × 24 h × 47 j = 90.24 €** | **90 €** |
| **Disque (20 GB)** | **0.05 €/GB/mois** | **20 × 0.05 × 1.53 mois = 1.53 €** | **1.5 €** |
| **IP publique fixe** | **~1.50 €/mois** | **1.50 × 1.53 = 2.30 €** | **2.3 €** |
| **Trafic sortant** | **Estimation : 30 Go @ 0.03 €/Go** | **30 × 0.03 = 0.90 €** | **0.9 €** |

**3. Estimation du coût total pour 47 jours (1.53 mois)**

| **Ressource** | | **Calcul** | | **Total approximatif** | |
| --- | --- | --- | --- | --- | --- |
| vCPU + RAM | | (0.02 + 0.04) €/h × 24 h × 47 jours = ~68 € | | 68 € | |
| Disque 80 GB | | 4 €/mois × 1.53 mois = 6.12 € | | 6.12 € | |
| IP publique fixe | | 1.5 €/mois × 1.53 mois = 2.30 € | | 2.30 € (option) | |
| Trafic sortant | | Ex : 30 Go × 0.03 €/Go = 0.90 € | | 0.90 € | |

➡️ Total estimé : 77 à 80 € pour 47 jours.

**4. Paramètres à monitorer activement**

| Élément | Commande / outil (Linux Ubuntu) |
| --- | --- |
| Uptime | uptime ou who -b |
| Usage CPU/RAM | htop, top, free -h |
| Utilisation disque | df -h |
| Bande passante | vnstat, iftop, ou Grafana/Prometheus si installé |
| Durée instance | Interface Infomaniak (historique / instance details) |

**5. Choix à prévoir avant le 19 août**

| **Option** | **Avantage** | **Risque ou coût** |
| --- | --- | --- |
| **Arrêt instance** | **Ne facture plus CPU/RAM, garde stockage/IP** | **Disque/IP facturés** |
| **Suppression instance** | **Zéro coût** | **Perte de config complète** |
| **Snapshot + arrêt** | **Archiver Jitsi (image QCOW2), puis stop** | **Très faible coût** |
| **Continuer** | **Service en ligne** | **50–80 €/mois** |

**Recommandation**

Créer un tableur de calcul automatique avec :

* Coût horaire vCPU
* Coût horaire RAM
* Coût mensuel disque
* IP (fixe ou non)
* Trafic estimé

**B / Le 03 / 07 / 2025 Lier Adresse IP et Nom de Domaine :**

[**https://www.infomaniak.com/fr/support/faq/2025/lier-un-nom-de-domaine-a-un-hebergement-web-infomaniak**](https://www.infomaniak.com/fr/support/faq/2025/lier-un-nom-de-domaine-a-un-hebergement-web-infomaniak)

**Nom de domaine fixé :** [**https://visio.workeezconnect.fr/**](https://visio.workeezconnect.fr/)

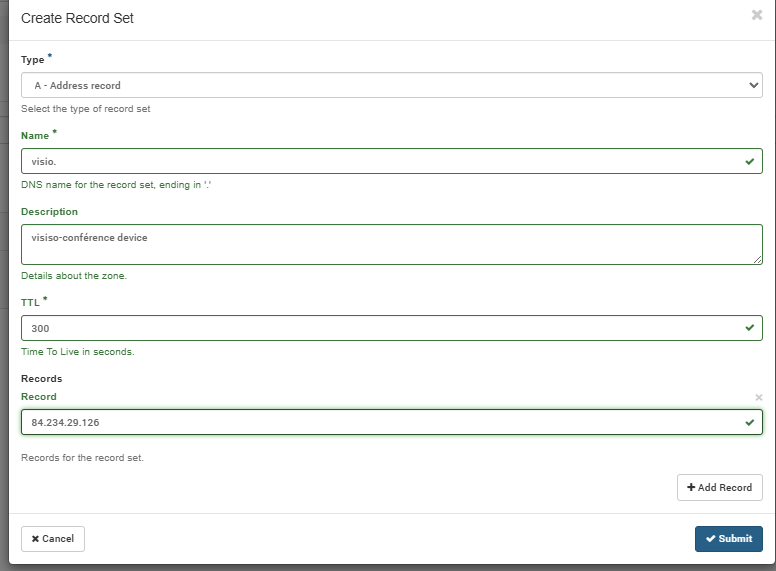
**Adresse IP fixée : 84.234.29.126**

**Explication théorique claire :**

| **Élément** | **Rôle** |
| --- | --- |
| **DNS (A record)** | **Associe visio.workeezconnect.fr → 84.234.29.126** |
| **Certificat SSL (Let's Encrypt)** | **Prouve que le serveur web contrôle réellement ce domaine** |
| **ACME challenge** | **Mécanisme utilisé pour prouver que tu possèdes le domaine** |

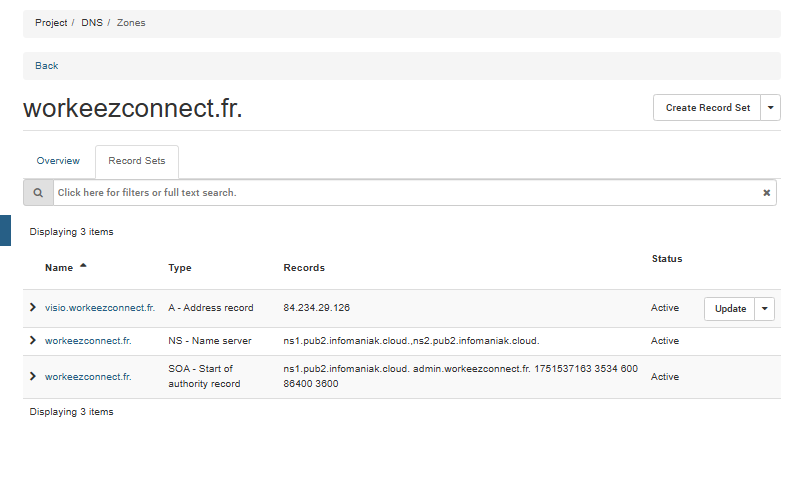
**Avant même l’installation du certificat il faut lier domaine et l’IP  : Interface Infomaniak**

**Interface Infomaniak : DNS Zones Create Zone**

****

**Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.**

****

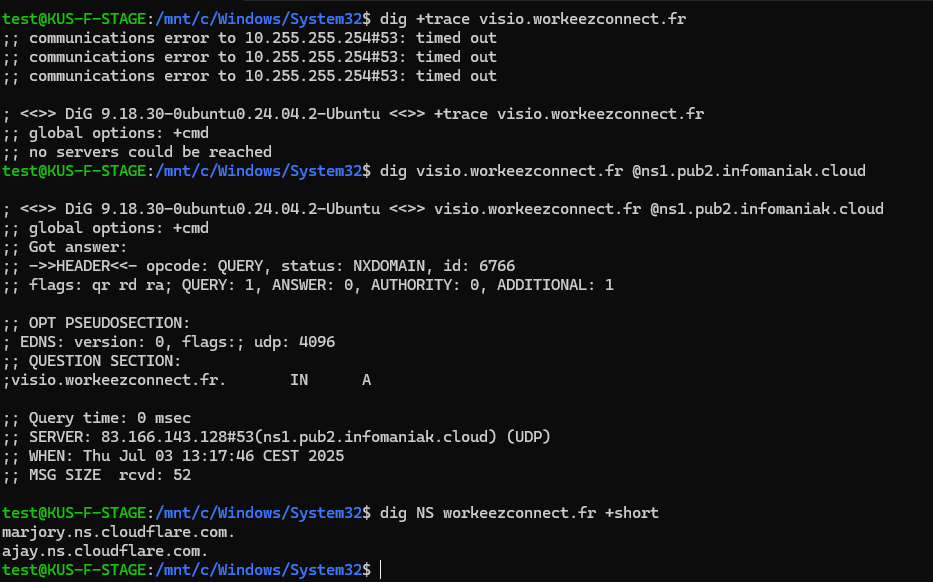
**Details des créations :**

**Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.**

**Tests avec : dig workeezconnect.fr NS +short**

* **dig :** commande DNS utilisée pour interroger les serveurs DNS.
* **workeezconnect.fr :** le nom de domaine que tu veux interroger.
* **NS :** signifie *Name Server*, c’est-à-dire que tu demandes les serveurs de noms responsables de ce domaine.
* **+short :** format de sortie simplifié (juste la réponse, pas les détails).



**Problèmes : les tests de connexions et d’identifications ont échoué, les serveurs sont des cloudflares non des infomaniak. Il faudra changer via l’accès client permettant la création de nom de domaines pour switcher les serveurs.**

**Actions requises :**

| **Action** | **Où ?** | **Objectif** |
| --- | --- | --- |
| **Modifier les serveurs de noms NS** | **Chez le registrar du domaine (OVH, Infomaniak, Gandi, etc.)** | **Remplacer \*.ns.cloudflare.com → ns1.pub2.infomaniak.cloud** |
| **Attendre propagation DNS** | **Internet (1h à 24h)** | **Reflète les bons serveurs** |
| **Tester avec dig ou nslookup** | **Ton terminal** | **Confirmer propagation** |

**Vérification après propagation : avec ces commandes et les réponses attendues**

1. **dig NS workeezconnect.fr +short**

**# Doit répondre avec :**

**ns1.pub2.infomaniak.cloud.**

**ns2.pub2.infomaniak.cloud.**

1. **dig visio.workeezconnect.fr +short**

**# Doit renvoyer :**

**84.234.29.126**

**Délai de propagation mondial réaliste.**

| **Niveau de cache / infrastructure** | **Délai estimé max** |
| --- | --- |
| **Serveurs racines & TLD (.fr)** | **15 à 30 min** |
| **Résolveurs publics (Google 8.8.8.8, etc.)** | **15 à 60 min** |
| **CDN (Cloudflare, Akamai...)** | **1 à 2 h** |
| **FAI (France, Europe, Afrique...)** | **2 à 24 h** |
| **Pays lointains (zones isolées, Asie, etc.)** | **24 à 48 h max** |

**Test : Réussi**

****

**2nd Test : Réussi**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Cependant : le test avec le navigateur accède au site avec une non secure alert.**

**Relance donc de la création du certificat :**

**cd : sudo certbot –apache -d visio.workeezconnect.fr**

**Création réussi :**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.**

**C / Relions maintenant le serveur Jitsi à notre instance domaine + Adresse IP**

**Se connecter à son dépôt via le dossier ou se situent les clefs privée et publique :**

**ssh -i Tercium-instance\_key/tercium-instance\_key** [**ubuntu@84.234.29.126**](mailto:ubuntu@84.234.29.126)

**Ajouter le dépôt de [Jitsi Meet](https://jitsi.org/jitsi-meet/) :**

**Préambule :** Le dépôt fourni par Jitsi utilise le chiffrement, donc vérifier que vous avez bien le paquet apt-transport-https.

**cd : sudo apt install apt-transport-https ca-certificates curl gnup -y**

➤ **apt-transport-https**

Permet à apt (le gestionnaire de paquets) de télécharger des paquets depuis des dépôts HTTPS (sécurisés via TLS). Sans cela, apt ne pourra pas accéder à https://download.jitsi.org.

➤ **ca-certificates**

Installe les autorités de certification (CA) racines utilisées pour vérifier la validité des certificats SSL/TLS lors des connexions sécurisées. Indispensable pour vérifier le certificat du dépôt Jitsi, de Let's Encrypt, etc.

➤ **curl**

Outil de ligne de commande pour faire des requêtes HTTP(S), utilisé ici pour récupérer la clé GPG de Jitsi.

➤ **gnupg**

Contient gpg (GNU Privacy Guard), utilisé pour décrypter et gérer les clés de signature GPG, comme celle du dépôt Jitsi.

➤ **-y**

Flag qui force l’acceptation automatique de l’installation sans demander confirmation à chaque paquet. Très utilisé en script pour éviter l’interaction manuelle.

**# Importer la clé de manière sécurisée**

**curl** [**https://download.jitsi.org/jitsi-key.gpg.key**](https://download.jitsi.org/jitsi-key.gpg.key) **| gpg --dearmor | sudo tee /usr/share/keyrings/jitsi-keyring.gpg > /dev/null**

**# Ajouter le dépôt en l’associant à la clé**

**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 > /dev/null**

**# Mettre à jour les paquets**

**sudo apt update**

**# Installation de Jitsi Meet**

**Sudo apt install jitsi-meet -y**

**# Associer l’IP et le nom de domaine : Déjà réaliser !**

**"Si votre nom de domaine pointe déjà via DNS public vers l’adresse IP de l’instance, l’étape /etc/hosts est facultative."**

**# Vérifier l’état des services des serveurs**

**systemctl status jitsi-videobridge2 systemctl status jicofo systemctl status prosody**

**# Analyse DNS complète avec dig.**

**dig +short visio.workeezconnect.fr**

**Jour-5 Tercium**

**# Vérification DNS + HTTP avec curl.**

**curl -i** [**http://visio.workeezconnect.fr**](http://visio.workeezconnect.fr)

**# Vérification croisée : IP publique actuelle du serveur.**

**curl -4 ifconfig.me**

**Fichier à éditer : sudo nano /etc/prosody/conf.d/visio.workeezconnect.fr.cfg.lua**

**Code de base avec GNU nano 7.2 :**

**VirtualHost "visio.workeezconnect.fr"**

**authentication = "internal\_hashed"**

**modules\_enabled = {**

**"bosh";**

**"pubsub";**

**"ping";**

**"speakerstats";**

**"conference\_duration";**

**"muc\_lobby\_rooms"; -- ✅ Ne rien ajouter après ce point-virgule dans la table !**

**}**

**Component "conference.visio.workeezconnect.fr" "muc"**

**Component "auth.visio.workeezconnect.fr"**

**authentication = "internal\_hashed"**

**Component "focus.visio.workeezconnect.fr"**

**component\_secret = "FOCUS\_SECRET" Il faudra ajouter en sus la possibilité d’inviter**

**# Redémarrage Prosody :**

**sudo systemctl restart prosody**

**# Vérifier qu’il est actif :**

**sudo systemctl status prosody**

**# Vérifier si le port** **5280** **est ouvert :** **sudo ss -tuln | grep 5280**

**# Tester l’accès HTTP BOSH : curl http://localhost:5280/http-bind/**

**# Activer si nécessaire le Firewall sur le port ( UFW doit être actif) : sudo ufw allow 5280/tcp**

**# Vérification de la configuration**

**sudo prosodyctl check config**

**1/ Il manque des modules :**

**modulemanager: Unable to load module 'conference\_duration'**

**modulemanager: Unable to load module 'speakerstats'**

**modulemanager: Unable to load module 'muc\_lobby\_rooms'**

**# On les installe avec : sudo apt install prosody-modules && on redémarre avec sudo systemctl restart prosody**

**2/ Bug car il demande un localhost (qui pose problème lors des connexions avec l’instance).** **L’erreur indique que le script post-installation de jitsi-meet-prosody échoue parce qu’il attend un fichier /etc/prosody/conf.avail/localhost.cfg.lua qui n'existe pas.**

**# Création de ce fichier : localhost.cfg.lua**

**sudo nano /etc/prosody/conf.avail/localhost.cfg.lua**

Contenu minimal du fichier (pour débloquer dpkg) :

**VirtualHost "localhost"**

**authentication = "anonymous"**

**# On relance la configuration des paquets cassés**

**sudo dpkg --configure -a**

**# On vérifie l’installation**

**sudo apt install -f**

**sudo systemctl restart prosody jicofo jitsi-videobridge2 nginx**

**# Créer un utilisateur XMPP :**  **sudo prosodyctl register admin visio.workeezconnect.fr motdepasse**

**3/ Cas Particulier source de problèmes :**

**Conflit localhost vs nom de domaine public** C’est **le problème principal qui a perturbé Certbot / Let's Encrypt / Apache** :

**▶️ Cause :**

Jitsi installe par défaut un vhost Apache localhost.conf :

apache

CopyEdit

<VirtualHost \*:80>

ServerName localhost

DocumentRoot "/usr/share/jitsi-meet"

</VirtualHost>

Simultanément, ton domaine réel (visio.workeezconnect.fr) est servi par :

* /etc/apache2/sites-enabled/000-default.conf (HTTP)
* /etc/apache2/sites-enabled/000-default-le-ssl.conf (HTTPS)

Certbot ou acme.sh essaie de **créer un challenge via le nom de domaine réel**, mais Apache répond par défaut avec la conf localhost, car elle est mal priorisée.

**▶️ Symptôme :**

bash

CopyEdit

curl http://localhost/.well-known/acme-challenge/test.txt

donne une **redirection 301 → HTTPS sur localhost**, ce qui empêche la validation http-01 par Let's Encrypt.

**▶️ Solutions recommandées :**

1. **Désactiver localhost.conf** s’il n’est plus nécessaire :

bash

CopyEdit

sudo a2dissite localhost.conf

sudo systemctl reload apache2

1. **S’assurer que visio.workeezconnect.fr est le ServerName prioritaire** dans les fichiers 000-default.conf et 000-default-le-ssl.conf.
2. **Contrôler l’ordre de priorité via** :

bash

CopyEdit

sudo apachectl -S

Et **supprimer tout vhost résiduel localhost** qui interfère.

✅ Résumé logique :

| **Test** | **Commande** | **Interprétation** |
| --- | --- | --- |
| **IP DNS** | **dig +short visio.workeezconnect.fr** | **Résultat doit être l’IP publique du serveur** |
| **IP publique (instance)** | **curl -4 ifconfig.me** | **Doit matcher celle obtenue par dig** |
| **Résolution + Apache OK** | **curl -I http://visio.workeezconnect.fr** | **HTTP 200/301/302 = OK** |
| **Conf Apache (ServerName)** | **sudo apachectl -S** | **Doit afficher visio.workeezconnect.fr en vhost** |

**On va installer Nginx fortement recommandé pour la sécurisation de échanges, cela implique arrêter Apache car il y aurait conflit sur le port 80 !**

**# Installer Nginx**

**sudo apt install -y nginx**

**# Démarrer le service**

**sudo systemctl start nginx**

**# Le rendre actif à chaque démarrage**

**sudo systemctl enable nginx**

**# Arrêt d’Apache2**

**sudo systemctl stop apache2**

**sudo systemctl disable apache2**

**# Supprimer d’Apache2**

**sudo apt purge apache2 apache2-utils apache2-bin apache2.2-common -y**

**Puis, s’assurer que le FQDN Fully Qualified Domain Name : nom de domaine complet incluant tous les niveaux (ex. auth.visio.workeezconnect.fr) est reconnu**

**echo "127.0.1.1 visio.workeezconnect.fr" | sudo tee -a /etc/hosts**

**# redémarrer Nginx**

**sudo systemctl start nginx**

**sudo systemctl enable nginx**

**# Vérifions l’état de Nginx et celui du port 80**

**sudo systemctl status nginx**

**sudo lsof -i :80**

**# Vérification les fichiers de configuration NGINX pour jitsi**

**cat /etc/nginx/sites-available/visio.workeezconnect.fr.conf | grep ssl**

**cat: /etc/nginx/sites-available/visio.workeezconnect.fr.conf: No such file or directory**

**Il faudra donc le crée**

**sudo nano /etc/nginx/sites-available/visio.workeezconnect.fr.conf**

**server {**

**listen 80;**

**server\_name visio.workeezconnect.fr;**

**location / {**

**proxy\_pass http://localhost:8000; # ou 5000 selon ton backend**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection "upgrade";**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**}**

**location /http-bind {**

**proxy\_pass http://localhost:5280/http-bind;**

**proxy\_set\_header Host $host;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header X-Forwarded-For $remote\_addr;**

**}**

**}**

**# Créer le lien symbolique**

**sudo ln -s /etc/nginx/sites-available/visio.workeezconnect.fr.conf /etc/nginx/sites-enabled/**

**# Vérifions la configuration**

**sudo nginx -t**

**# Redémarrer Nginx sudo systemctl restart nginx**

**# Insérons dans ce fichier un bloc sécurisé avec HTTPS + Cerbot et vérifions que le certificat est présent sudo ls /etc/letsencrypt/live/visio.workeezconnect.fr/**

**# Non il faut installer le plugin Certbot pour Nginx**

**sudo apt install python3-certbot-nginx -y**

**Puis Relancer la commande Certbot**

**sudo certbot --nginx -d visio.workeezconnect.fr**

**4/ Ici je stop l’installation et je débug car il y a eu trop de manipulations générant des bugs récurrents.**

**1-Corriger les erreurs d'installation des paquets Jitsi (dpkg)** et assurer une base fonctionnelle propre.

**cd: hostnamectl**

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**2-Corriger /etc/hosts**

**cd:** **sudo nano /etc/hosts**

**Une image contenant texte, logiciel, Logiciel multimédia, Police

Le contenu généré par l’IA peut être incorrect.**

**3-Redémarrer Prosody**

**cd:** **sudo systemctl restart prosody**

**4-Redémarrer Jicofo et JVB**

**cd:** **sudo systemctl restart jicofo**

**cd:** **sudo systemctl restart jitsi-videobridge2**

**5-** **Vérifier l’état**

**cd: sudo systemctl status prosody jicofo jitsi-videobridge2 ok active (running )**

**Jour-6 Tercium 2nde semaine | Déploiement**

**Sur ce PC je ne suis pas en ROOT donc soit on passe en tant qu’administrateur ponctuelle : Sudo, soit passage au mode WSL.**

1. **Recréation d’une instance et modification donc de l’adresse IP.**

**Une image contenant texte, nombre, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Dans DNS insérer la nouvelle IP et surtout submit et contrôler celui-ci.**
2. **Pinger la nouvelle adresse IP.**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Commande : nouvelle instance Infomaniak : DNS**

**Une image contenant texte, nombre, logiciel, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

1. **Connexion réussie via Git Bash :** ssh -i Tercium-instance\_key/tercium-instance\_key [ubuntu@84.234.28.98](mailto:ubuntu@84.234.28.98) **/ à chaque instance détruite / recréer, l’IP adresse change**.

**Tester avec :**

**cd :** **dig visio.workeezconnect.fr +short**

**cd : nslookup visio.workeezconnect.fr**

**Résultat nous pointons toujours vers l’ancienne IP adress de l’instance détruite la semaine dernière.**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Nous avons un conflit pour accéder aux clefs, précisément la privé de mon côté. Pour continuer nous devons alors pointer d’abord dans le bon dossier ; ici changement du nom du dossier en ssh.key:**

**cd : C:\Users\test\Documents\Tercium\_Stage\ssh\_keys**

**En Powershell :**

**En Bash (git\_bash ou en WSL) : ssh -i /c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Ou**

**ssh -i /c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes ubuntu@visio.workeezconnect.fr**

**Lors de la recréation d’une instance, L’on doit à chaque fois relier le nouvel IP Adress au domaine. Autrement il ne peut y avoir connexion SSH et d’un et de deux la capacité d’installer Jitsi dans l’instance préparée.**

**Tester avec :**

**cd :** **dig visio.workeezconnect.fr +short**

**cd : nslookup visio.workeezconnect.fr**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**cd : dig visio.workkeezconnect.fr @1.1.1.1 +trace**

**Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

**On se connecte, chemin vers des fichiers clefs:**

**C:\Users\test\Documents\Tercium\_Stage\ssh\_keys**

**Powershell :**

**cd : ssh -i "C:\Users\test\Documents\Tercium\_Stage\ssh\_keys\tercium-instance\_key" -o IdentitiesOnly=yes** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Ou**

**Git Bash :**

**cd : ssh -i ~/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Vérification complémentaires :**

1**/ Permissions sur la clé privée (depuis WSL/Git Bash) :**

**cd : chmod 600 ~/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

**2/ Connexion avec vérification du fingerprint :  
Tu peux ajouter -v pour obtenir des logs :**

**cd : ssh -i ~/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes -v** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**Nettoyer le cache DNS interne en powershell: cd : ipconfig /flushdns**

**PRÉREQUIS**

1. **Nom de domaine pointant vers l’IP publique : visio.workeezconnect.fr → 37.156.46.238.**
2. **Accès SSH root ou sudo via ta clé (Tercium-instance\_key).**
3. **Résolution DNS locale correcte (dig, ping, etc.).**
4. **Port TCP 443 (HTTPS) et 80 (HTTP) ouverts dans les règles de pare-feu GCP/Infomaniak.**
5. **ufw désactivé ou configuré (on vérifiera ça).**
6. **Pas d’Apache2 actif pour éviter conflit avec NGINX.**

**ÉTAPES POUR INSTALLER JITSI + NGINX AVEC CONFIG LOCALE**

**Installation des paquets requis :**

sudo apt update && sudo apt upgrade -y

sudo apt install curl gnupg2 software-properties-common apt-transport-https -y

**Ajout du dépôt Jitsi officiel :**

**Ajout la clef GPG :**

**curl https://**download.jitsi.org/jitsi-key.gpg.key | sudo gpg --dearmor -o /usr/share/keyrings/jitsi-keyring.gpg

**Ajoute la clé 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

**sudo apt update**

**Installation de Jitsi Meet :**

**cd : sudo apt install -y jitsi-meet**

🧠 Pendant l’installation : Possibilités

* FQDN : visio.workeezconnect.fr
* Choisir **"configurer SSL plus tard (manuel)"**

**7. Génération du certificat SSL (Let’s Encrypt)**

sudo apt install certbot python3-certbot-nginx -y

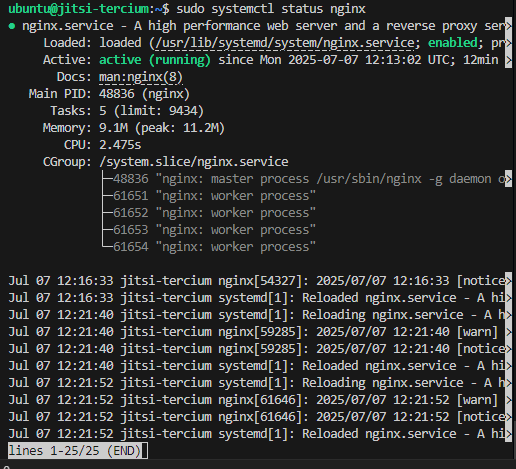
sudo certbot --nginx -d visio.workeezconnect.fr

Valide automatiquement le NGINX + active HTTPS.

**Étapes suivantes à vérifier pour finaliser le déploiement :**

1. **Vérifier que Nginx est bien configuré pour Jitsi**

**cd :** **sudo systemctl status nginx**

****

1. **Tester la connexion HTTPS depuis un navigateur : Accède à** [**https://visio.workeezconnect.fr**](https://visio.workeezconnect.fr)

**Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Le contenu généré par l’IA peut être incorrect.**

1. **Vérifier les ports nécessaires ouverts : il faut installer ufw**

**cd : sudo apt update && sudo apt install ufw**

**cd : sudo ufw allow 80,443/tcp**

**sudo ufw allow 10000/udp**

**sudo ufw enable**

**sudo ufw status**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Vérifier les modules Prosody :**

**cd : sudo cat /etc/prosody/conf.avail/visio.workeezconnect.fr.cfg.lua | grep modules\_enabled -A 20**

1. **Test de fonctionnement des modules : relance puis vérification de son état**

**cd : sudo systemctl restart prosody**

**Puis**

**cd : sudo systemctl status prosody**

**Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Ne pouvant contrôler Jitsi du côté d’Infomaniak, car nous ne voyons pas l’installation de celui-ci. Cependant la section DNS via Zones nous donnera ce tableau qui nous permettra de réaliser des vérifications.**

1. **Vérifier la résolution DNS (commande dig) :**

**cd : dig visio.workeezconnect.fr @8.8.8.8 +short**

**Réponse attendue et reçue : 37.156.48.238**

****

1. **Tester la connectivité HTTP(S) (commande curl) :**

**cd : curl -I** <https://visio.workeezconnect.fr>

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **S’il y avait échec avec HTPPS, tester HTTP:**

**cd : curl -I** <http://visio.workeezconnect.fr>

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Tester le port 443 avec telnet ou nc**

**cd : nc -zv visio.workeezconnect.fr 443**

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

1. **Tester la chaîne de certificat SSL (Certbot utilisé)**

**cd : echo | openssl s\_client -connect visio.workeezconnect.fr:443 -servername visio.workeezconnect.fr**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**

* **ISRG Root X1 → racine de Let's Encrypt**
* **Let's Encrypt E6 → intermédiaire**
* **visio.workeezconnect.fr → certificat de ton instance**

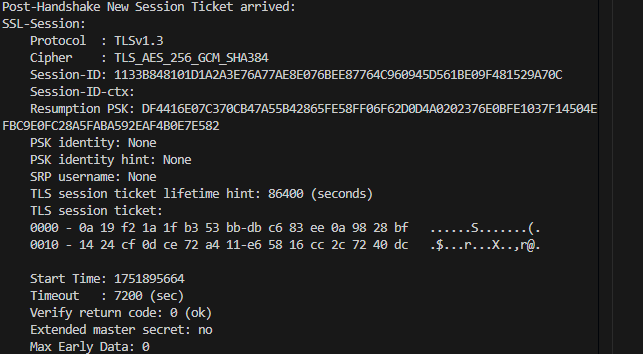
**Vérification SSL :**

**Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

**Verify return code: 0 (ok)**

**Vérification TLS :**

****

**Conclusion :**

| **Élément vérifié** | **État** |
| --- | --- |
| Certificat SSL bien installé | ✅ OK |
| Chaîne complète avec intermédiaire & racine | ✅ OK |
| Nom de domaine valide (CN = visio...) | ✅ OK |
| Protocole TLS récent (1.3) | ✅ OK |

**Jour-7 Tercium 2nde semaine | monitoring**

**Monitoring Infomaniak :**

**Si vous arrêtez temporairement l'instance (sans la supprimer) :**

* Jitsi reste installé sur le disque
* Vous devrez juste vous reconnecter en SSH avec la nouvelle adresse IP
* Aucune réinstallation nécessaire

**Si vous supprimez complètement l'instance :**

* Tout est effacé, y compris Jitsi
* Vous devrez tout réinstaller depuis zéro sur une nouvelle instance

**Si votre instance redémarre avec une nouvelle IP :**

* Jitsi restera installé mais il faudra probablement ajuster la configuration
* Le certificat SSL devra être régénéré si vous utilisez un nom de domaine
* Vous devrez mettre à jour l'adresse IP dans votre configuration SSH

Dans la plupart des cas avec les services cloud, un simple arrêt/redémarrage conserve vos données. Seule la suppression définitive de l'instance efface tout.

Quel est votre cas exactement ? Vous utilisez quel service cloud (AWS, Google Cloud, etc.) ?

Avec Infomaniak, vous pouvez arrêter votre instance sans l'effacer. Voici comment procéder :

**Dans le manager Infomaniak :**

1. Connectez-vous à votre espace client Infomaniak
2. Allez dans la section "Public Cloud" ou "Serveurs"
3. Sélectionnez votre instance
4. Cliquez sur "Arrêter" (bouton stop/pause)

**Ce qui se passe quand vous arrêtez :**

* L'instance s'arrête mais le disque virtuel est conservé
* Vous ne payez plus les ressources CPU/RAM (seulement le stockage)
* Toutes vos données et installations (dont Jitsi) restent intactes

**Pour redémarrer plus tard :**

* Cliquez sur "Démarrer" dans le manager
* L'instance redémarre avec la même configuration
* Vous devrez peut-être noter la nouvelle adresse IP publique
* Connectez-vous en SSH avec cette nouvelle IP

**Important :** Ne cliquez pas sur "Supprimer" ou "Détruire" - cela effacerait définitivement tout.

L'arrêt d'instance chez Infomaniak est réversible et conserve vos données, contrairement à la suppression qui est définitive.

**Planning structuré de l’audit:**

* Audit technique général
* Inventaire des environnements matériels (serveurs, réseau, postes...)
* Demande d'autorisation pour les scans réseau (très important !)
* Validation de la topologie réseau réelle

**Concernant les autorisations :**

* netstat : généralement OK (analyse locale)
* nmap : nécessite autorisation écrite (scan réseau)
* arpscan : idem, peut être perçu comme intrusif

**Conseils pour la demande :**

* Préciser que c'est pour valider la documentation existante
* Mentionner les créneaux horaires prévus
* Expliquer que c'est pour la sécurité/conformité
* Proposer un rapport de synthèse

**Phase discrète :**

* Netstat poste par poste → pas d'impact réseau, juste analyse locale
* Collecte d'infos sans perturber l'activité

**Phase avec autorisation :**

* Nmap et arpscan après accord du boss
* Approche transparente et professionnelle

**Le plus intéressant :** La configuration sécurité avec API et alertes smartphone ! C'est du monitoring intelligent :

* Alertes discrètes en temps réel
* Le boss reste informé sans être submergé
* Réactivité en cas d'incident

**Conseils pour convaincre :**

* Expliquer que nmap/arpscan permettront de valider que cette API de sécurité ne rate rien
* Montrer que c'est pour renforcer son système d'alertes existant
* Proposer un test sur une plage horaire définie

Travail d'auditeur ! Vous combinez discrétion, professionnalisme et valeur ajoutée. L’entreprise appréciera sûrement cette approche qui respecte ses processus tout en améliorant la sécurité.

Le système d'alertes smartphone montre qu'il y a une sensibilisation à la sécurité.

**Déploiement et correction des bugs : pop-ups absent, absence de micro, de caméra etc.**

**Résumé des actions à vérifier côté client**

| **Élément** | **Vérification** |
| --- | --- |
| **Pop-up caméra/micro** | **via cadenas navigateur** |
| **Notifications autorisées** | **dem** |
| **Console JS (erreurs)** | **🔍 check getUserMedia, ICE** |
| **Headers Nginx** | **Permissions-Policy, CORS** |
| **Services Jitsi** | **jicofo, videobridge** |

**09h20 fin des tests :**

* **L’import d’un Micro et d’une caméra externe sont supportés.**
* **Ceux sont pour la caméra une 720p essential B et pour le micro un Razer omnidirectionnel semi-professionnel.**
* **Les fonctions de partage d’écran, d’invitation via un lien simple ou multiple fonctionnent.**
* **L’ensemble des fonctions attendues existantes sur zoom et anciennement skype sont présentes.**

[**https://www.onlyoffice.com/blog/fr/2023/02/zoom-ou-jitsi**](https://www.onlyoffice.com/blog/fr/2023/02/zoom-ou-jitsi)

**En Annexe des saisies d’écran de l’interface avec les fonctionnalités disponibles.**

**Comparaison à titre indicatif de l’usage de Zoom :**

* [**Zoom Spaces**](https://explore.zoom.us/fr/products/zoom-rooms/)**est un moyen innovant d’organiser des réunions virtuelles au sein d’équipes hybrides. Les services inclus sont la réservation d’un espace de travail (par exemple, un bureau ouvert, un dispositif Zoom), un connecteur de salle de conférence pour rejoindre les réunions avec un équipement SIP/H.323, et des salles Zoom pour les conférences qui impliquent de nombreuses personnes des deux côtés.**
* [**Zoom Events**](https://explore.zoom.us/fr/products/event-solutions/)**est une solution dédiée à l’organisation d’événements virtuels, des annonces internes aux tables rondes. Cette option est idéale pour le réseautage, la présentation de contenu dans des stands d’exposition, la formation d’équipes dans des événements à session unique et des webinaires en ligne.**
* [**Zoom Contact Center**](https://explore.zoom.us/fr/products/contactcenter/)**vous permet de créer un environnement omnicanal pour fournir une assistance à vos clients. Les outils disponibles comprennent un centre de contact optimisé pour la vidéo dans le cloud et l’IA conversationnelle.**

**Coût & Résultat d’usage :**

**Comme chaque solution comprend différents ensembles de fonctionnalités, les prix varient en fonction de votre demande.**

**Plans tarifaires :**

**Le coût d’un logiciel de vidéoconférence est un aspect essentiel pour la plupart des équipes, car chaque entreprise souhaite économiser sur les solutions numériques et investir de l’argent dans la croissance.**

**Zoom offre un plan gratuit avec les caractéristiques suivantes :**

* **Jusqu’à 40 minutes par réunion**
* **100 participants par réunion**
* **3 tableaux modifiables avec 25 Mo de stockage en nuage**
* **Chat d’équipe**

**Le plan de base est suffisant pour la communication personnelle, tandis que les entreprises envisageront des plans payants avec des fonctionnalités puissantes. Les prix commencent à 149 $ par utilisateur/mois et peuvent inclure :**

* **Jusqu’à 1000 participants**
* **Réunions illimitées**
* **30 heures par réunion**
* **Stockage illimité des enregistrements sur le cloud**
* **Tableaux blancs illimités**
* **Authentification unique (Single Sign-On – SSO)**
* **Domaines gérés**
* **Marque de l’entreprise**

**Sécurité :**

**La sécurité et la confidentialité sont essentielles au succès de la vidéoconférence. Personne ne souhaite que des personnes non autorisées aient accès à des discussions privées contenant des informations sensibles.**

**Jitsi est axé sur la sécurité et propose des salles de réunion éphémères, ce qui signifie qu’elles n’existent que pendant la réunion. Les autres options de confidentialité comprennent la protection par mot de passe, le lobby, le cryptage de bout en bout.**

**Jitsi Meet pour un déploiement sur site ne dispose pas de moteurs d’analyse préconfigurés. Même dans la version en ligne, il ne conserve aucune information personnelle, comme les noms ou les adresses électroniques.**

**Zoom est considéré comme relativement sûr grâce à son cryptage de bout en bout. Toutefois, des sources fiables signalent que Zoom utilise l’algorithme AES-128 au lieu de AES-256, ce qui rend sa sécurité plus vulnérable dans une certaine mesure. En outre, vous ne serez pas en mesure de vérifier le code source de ce logiciel pour vous assurer qu’il ne contient pas de virus ou de portes dérobées.**

**Dans la pratique, les deux solutions se sont avérées bien protégées. Si vous suivez les directives de base en matière de sécurité de l’information sur votre lieu de travail, les piratages et les attaques extérieures sont peu probables.**

**Plugin Jitsi pour ONLYOFFICE Docs :** [**https://www.onlyoffice.com/blog?p=26614**](https://www.onlyoffice.com/blog?p=26614)

**Monitoring : 3ième étape de la gestion de projet**

**OBJECTIF**

1. **Monitorer l’instance Jitsi (CPU, RAM, disques, etc.).**
2. **Visualiser dans Grafana.**
3. **Superviser la disponibilité du service Jitsi via Prometheus (ou Blackbox Exporter).**
4. **Intégrer Pfsense et ipatable.**
5. **Alerter si nécessaire.**

**Les modes de connexion :**

**1/Connexion avec vérification du fingerprint : Vscode en Git Bash  
Tu peux ajouter -v pour obtenir des logs :**

**cd : ssh -i ~/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes -v** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Naviguer via WSL demande une réécriture. Pour accéder au dossier C:\Users\test\Documents\Tercium\_Stage depuis un terminal WSL, vous devrez naviguer en utilisant le chemin Linux approprié, car WSL monte les disques Windows sous /mnt/c/.**

**cd : cd /mnt/c/Users/test/Documents/Tercium\_Stage**

**Puis**

**cd : ssh -i /mnt/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key -o IdentitiesOnly=yes -v** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Cependant l’accès à une instance infomaniak via WSl est bloquée. La raison les permissions SSH sont trop souples. La commande ci-dessous est refusée.**

**cd : chmod 600 /mnt/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

**Explication : Différences entre Git Bash et WSL**

1. **Git Bash et les permissions de fichiers :**
   * **Git Bash** (sous Windows) fonctionne généralement en émulation Unix, mais il a une gestion un peu plus souple des permissions de fichiers. Il peut donc permettre l'accès à des clés privées même avec des permissions un peu plus larges.
2. **WSL et les permissions sous Windows :**
   * **Sous WSL,** les permissions de fichiers sont plus strictes, car elles sont basées sur le système de fichiers Linux qui s'exécute sur Windows. Lorsque vous accédez à un fichier Windows via /mnt/c/, WSL essaie de respecter les permissions de fichier Linux. Par défaut, les fichiers sous /mnt/c/ peuvent être considérés comme trop accessibles si leurs permissions sont trop larges.

**Testons sous le terminal Git Bash :**

**cd : ls -l /c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

* **Permissions actuelles : rw-r--r--**
  + **Propriétaire (test) : lecture et écriture (rw-).**
  + **Groupe et autres utilisateurs : lecture seule (r--).**
* **Propriétaire : test (probablement votre utilisateur actuel).**
* **Taille : 3381 octets.**
* **Date : Le fichier a été modifié pour la dernière fois le 1er juillet.**

**Problème :**

Les permissions actuelles du fichier clé (**rw-r--r--**) ne sont pas suffisantes pour que **SSH** fonctionne correctement. **SSH** exige que la clé privée soit accessible uniquement par le propriétaire du fichier, ce qui nécessite des permissions strictes de type **600**.

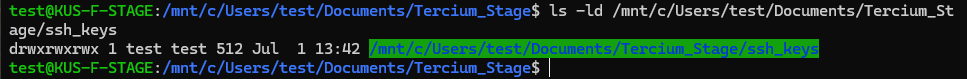
**Modifions les permissions** pour les rendre compatibles avec SSH :

**cd : chmod 600 /c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

**2/ Connexion avec WSL :**

**Testons sous le termina WSL :**

**cd : ls -l /mnt/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

****

**Modifions les permissions** pour les rendre compatibles avec SSH :

**cd : sudo chmod 600 /mnt/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key**

* **Des solutions et explications seront en annexe**

**3/ Connexion avec Powershell :**

**Testons sous le terminal sous Powershell :**

**cd : ssh -i C:\Users\test\Documents\Tercium\_Stage\ssh\_keys\tercium-instance\_key -o IdentitiesOnly=yes -v ubuntu@37.156.46.238**

**Restriction de privilége :**

**Usage possible d’icacls, un utilitaire en ligne de commande de Windows, pour modifier les permissions sur le fichier clé privée. La commande suivante restreint l'accès à la clé privée aux seuls utilisateurs autorisés :**

**cd : icacls "tercium-instance\_key" /inheritance:r /grant:r "test:(R)"**

**Explication :**

* **/inheritance:r :** Supprime l'héritage des permissions (cela empêchera les autres utilisateurs d'hériter des permissions du répertoire parent).
* **/grant:r "test:(R)" :** Donne à l'utilisateur **test (**remplacez par votre nom d'utilisateur si nécessaire**) l'accès en lecture seule (R) sur ce fichier.**

**Vérification :**

**cd : icacls "tercium-instance\_key"**

**L’installation des outils de supervision :**

**L’on part toujours du dossier ou se situe des deux clefs, la publique et la privée.**

**1/ Powershell :**

**cd : test@KUS-F-STAGE MINGW64 ~/Documents/tercium\_Stage**

**Connexion SSH  en Powershell:**

**cd : ssh -i C:\Users\test\Documents\Tercium\_Stage\ssh\_keys\tercium-instance\_key -o IdentitiesOnly=yes -v** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**2/ Connexion SSH en Git Bash :**

**cd : ssh -i ~/.ssh/tercium-instance\_key** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238) en linux ou WSL

**ssh -i "/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Installation de PROMETHEUS:**

**Télécharger :**

**cd : cd ~wget https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz**

**tar -xvf prometheus-2.52.0.linux-amd64.tar.gz**

**sudo mv prometheus-2.52.0.linux-amd64 /opt/prometheus**

**Service :**

**cd : sudo tee /etc/systemd/system/prometheus.service > /dev/null <<EOF**

**[Unit]**

**Description=Prometheus**

**Wants=network-online.target**

**After=network-online.target**

**[Service]**

**User=prometheus**

**Group=prometheus**

**Restart=on-failure**

**ExecStart=/opt/prometheus/prometheus \\**

**--config.file=/etc/prometheus/prometheus.yml \\**

**--storage.tsdb.path=/var/lib/prometheus \\**

**--web.console.templates=/etc/prometheus/consoles \\**

**--web.console.libraries=/etc/prometheus/console\_libraries**

**--web.listen-adress=0.0.0.0 :9091**

**[Install]**

**WantedBy=multi-user.target**

**EOF**

**sudo systemctl daemon-reload**

**sudo systemctl enable --now prometheus**

**Définir les droits :**

**cd :**

**sudo useradd -rs /bin/false prometheus**

**sudo mkdir -p /etc/prometheus /var/lib/prometheus**

**sudo cp /opt/prometheus/prometheus.yml /etc/prometheus/**

**sudo cp -r /opt/prometheus/consoles /opt/prometheus/console\_libraries /etc/prometheus/**

**sudo chown -R prometheus: /etc/prometheus /var/lib/prometheus**

**Créer le dossier de données :**

**cd :**

**sudo mkdir -p /var/lib/prometheus**

**sudo chown -R prometheus: /var/lib/prometheus**

**Recharger systemd :**

**cd : sudo systemctl daemon-reload**

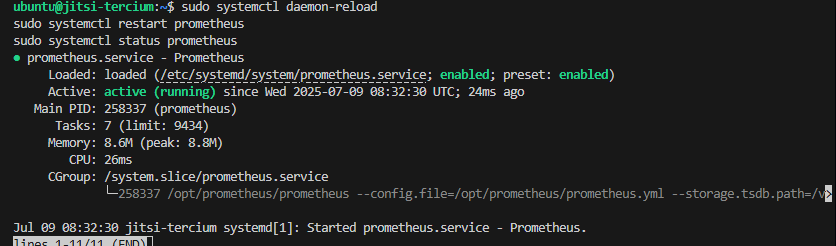
**Lancer Prometheus :**

**cd : sudo systemctl enable --now prometheus**

**Vérifier :**

**cd : sudo systemctl status prometheus**

**curl** [**http://localhost:9090**](http://localhost:9090)

****

**Attention conflit avec Jjtsi\_meet au port 9090 passage de prometheus au port 9091 via le le service group de infomaniak et au changement dans son script voir au-dessus.**

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**Jour-8 Tercium 2nde semaine | monitoring**

**Connexion SSH en Git Bash :**

**cd : ssh -i "/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Installation de NODE EXPORTER :**

**Télécharger :**

**cd : cd ~wget https://github.com/prometheus/node\_exporter/releases/download/v1.8.1/node\_exporter-1.8.1.linux-amd64.tar.gz**

**tar -xvf node\_exporter-1.8.1.linux-amd64.tar.gz**

**sudo mv node\_exporter-1.8.1.linux-amd64/node\_exporter /usr/local/bin/**

**Utilisateur & service :**

**cd : sudo useradd -rs /bin/false node\_exporter**

**sudo tee /etc/systemd/system/node\_exporter.service > /dev/null <<EOF**

**[Unit]**

**Description=Node Exporter**

**After=network.target**

**[Service]**

**User=node\_exporter**

**ExecStart=/usr/local/bin/node\_exporter**

**[Install]**

**WantedBy=default.target**

**EOF**

**sudo systemctl daemon-reload**

**sudo systemctl enable --now node\_exporter**

**Installation de BLACKBOX EXPORTER :**

**Télécharger :**

**cd : cd / opt**

**sudo wget** [**https://github.com/prometheus/blackbox\_exporter/releases/download/v0.25.0/blackbox\_exporter-0.25.0.linux-amd64.tar.gz**](https://github.com/prometheus/blackbox_exporter/releases/download/v0.25.0/blackbox_exporter-0.25.0.linux-amd64.tar.gz)

**sudo tar -xvf blackbox\_exporter-0.25.0.linux-amd64.tar.gz**

**sudo mv blackbox\_exporter-0.25.0.linux-amd64 /opt/blackbox\_exporter**

**Utilisateur & service :**

**cd : sudo nano /etc/systemd/system/blackbox\_exporter.service**

**[Unit]**

**Description=Blackbox Exporter**

**Wants=network-online.target**

**After=network-online.target**

**[Service]**

**User=prometheus**

**ExecStart=/opt/blackbox\_exporter/blackbox\_exporter \**

**--config.file=/opt/blackbox\_exporter/blackbox.yml**

**[Install]**

**WantedBy=multi-user.target**

**Redémarrer :**

**sudo systemctl daemon-reexec**

**sudo systemctl daemon-reload**

**sudo systemctl enable blackbox\_exporter**

**sudo systemctl start blackbox\_exporter**

**Installation de GRAFANA :**

**Installation via dépôt officiel :**

**cd : # Installation officielle (clé + repo)**

**sudo apt-get install -y software-properties-common**

**sudo apt-get install -y gnupg2 curl**

**sudo mkdir -p /etc/apt/keyrings**

**curl -fsSL https://apt.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg**

**echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee /etc/apt/sources.list.d/grafana.list**

**sudo apt update**

**sudo apt install grafana -y**

**# Activation et lancement**

**sudo systemctl enable grafana-server**

**sudo systemctl start grafana-server**

**Vérification avec :**

**cd : sudo systemctl status grafana-server**

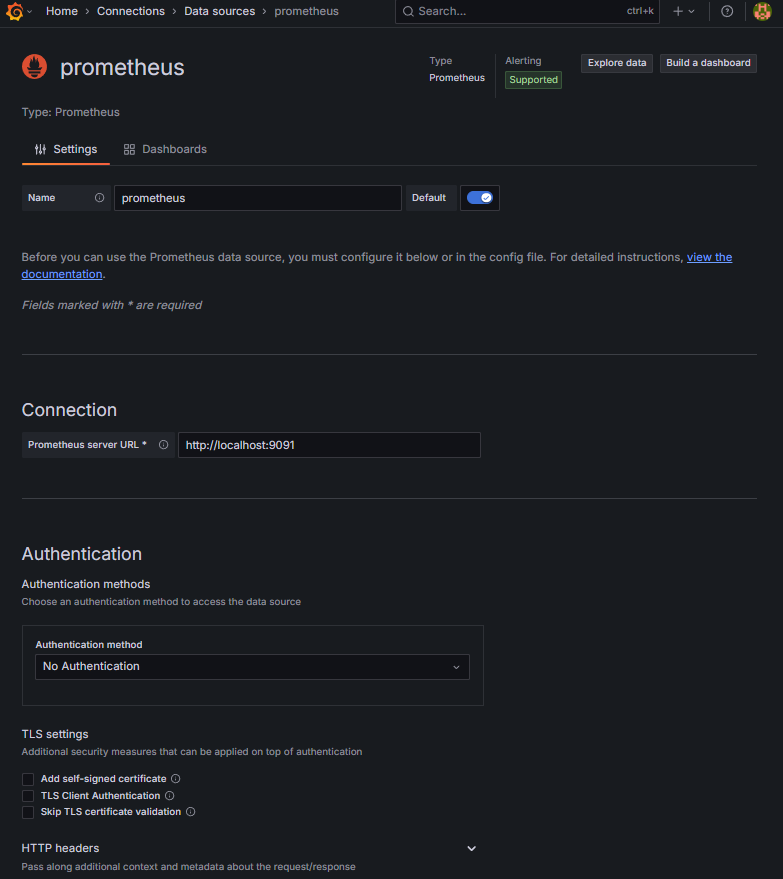
**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.**

**🟢 Interface Web : http://<ip\_publique>:3000**

**Dans notre cas cela sera :** [**http://37.156.46.238:3000**](http://37.156.46.238:3000)

**Identifiant par défaut : admin / admin ici admin /BZTwwtq1964**

****

**Installation de PFSENSE :**

**1/ Commandes pour des dépôt github bash non accessibles :**

**cd : chmod +x pfsense\_exporter-linux-amd64**

**cd : sudo mv pfsense\_exporter-linux-amd64 /usr/local/bin/pfsense\_exporter**

**2/ Télécharger et placer le binaire : sous forme de python script via un repo GitHub** (sviskontree/pfsense\_exporter).

**Contient deux fichiers :**

**Script principal :**

**cd : wget** [**https://raw.githubusercontent.com/sviskontree/pfsense\_exporter/master/pfsense\_exporter.py**](https://raw.githubusercontent.com/sviskontree/pfsense_exporter/master/pfsense_exporter.py)

**Script secondaire de lancement :**

**cd : wget**

[**https://raw.githubusercontent.com/sviskontree/pfsense\_exporter/master/pfsense\_exporter.sh**](https://raw.githubusercontent.com/sviskontree/pfsense_exporter/master/pfsense_exporter.sh)

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**Attention les scripts de pfsense sont en FreeBSD**

**cd : sudo mv pfsense\_exporter.py /usr/local/bin/**

**sudo mv pfsense\_exporter.sh /usr/local/etc/rc.d/**

**chmod +x /usr/local/bin/pfsense\_exporter.py /usr/local/etc/rc.d/pfsense\_exporter.sh**

**Ces commandes ne peuvent fonctionner que sous FreeBSD et sous pfsense**

**Il faudra créer un dossier local des services:**

**Donc :**

**cd : sudo mkdir -p /opt/pfsense\_exporter**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Puis déplacer les deux fichiers avec les autorisations vers l’instance et vérifier l’export :**

**cd : sudo mv pfsense\_exporter.py /opt/pfsense\_exporter/**

**sudo mv pfsense\_exporter.sh /opt/pfsense\_exporter/**

**sudo chmod +x /opt/pfsense\_exporter/\***

**cd : ls -l**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Puis vérifier l’emport :**

**cd : ls -l /opt/pfsense\_exporter/**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**/opt/pfsense\_exporter/**

**├── pfsense\_exporter.py**

**└── pfsense\_exporter.sh**

**Créer un service systemd :**

**cd : sudo nano /etc/systemd/system/pfsense\_exporter.service**

**Contenu :**

**cd :**

**[Unit]**

**Description=PfSense Exporter for Prometheus**

**After=network.target**

**[Service]**

**ExecStart=/usr/bin/python3 /opt/pfsense\_exporter/pfsense\_exporter.py**

**Restart=always**

**User=ubuntu**

**WorkingDirectory=/opt/pfsense\_exporter**

**[Install]**

**WantedBy=multi-user.target**

**Activation et démarrage du service :**

**cd : sudo systemctl daemon-reload**

**sudo systemctl enable --now pfsense\_exporter**

**sudo systemctl status pfsense\_exporter**

**ECHEC comme avec le premier dépôt linux.**

**On purge proprement :**

**Supprimer le service systemd :**

**cd : sudo systemctl stop pfsense\_exporter**

**sudo systemctl disable pfsense\_exporter**

**sudo rm /etc/systemd/system/pfsense\_exporter.service**

**sudo systemctl daemon-reload**

**Supprimer les fichiers Python:**

**cd : sudo rm -rf /opt/pfsense\_exporter/**

**Supprimer les logs:**

**cd : sudo journalctl --vacuum-time=1s**

**Configuration SNMP côté pfSense : pourquoi utiliser SNMP ?**

| **Usage** | **Action** |
| --- | --- |
| **Exploiter le script pfsense\_exporter.py avec les métriques pf\_ (comme dans ton JSON Grafana)** | **❌ PAS besoin de Telegraf** |
| **Récupérer via SNMP des métriques système ou bas-niveau (uptime, interfaces, CPU, etc.)** | **✅ Telegraf est utile** |

**Intérêt :**

| **Donnée à récupérer** | **Méthode** | **Outil** |
| --- | --- | --- |
| **États de firewall pfSense** | **API (HTTP)** | **pfsense\_exporter.py** |
| **Statut interfaces SNMP** | **SNMP (port 161)** | **Telegraf** |
| **UpTime général** | **SNMP** | **Telegraf** |
| **CPU, RAM (pfSense)** | **SNMP** | **Telegraf** |

**Prérequis :**

**Installer SNMP :**

**cd : sudo apt install snmp -y**

**1 Sur pfsense : Activer SNMP**

* **Menu : Services > SNMP**
* **Activez le service**
* **Port : 161, community : public (ou sécurisé)**
* **Activez les modules d’interface et de système**

**Mise à jour - Tests SNMP après ouverture du port 161/UDP**

**Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.**

**Ouverture du port 161  
Une règle Ingress a bien été ajoutée dans le groupe de sécurité visio-monitoring-group pour autoriser le trafic entrant sur le port 161/UDP depuis toutes les IPs (CIDR 0.0.0.0/0).**

* **Tests snmpwalk Trois commandes ont été lancées successivement :**
* **snmpwalk -v2c -c prom-readonly 37.156.46.238 1.3.6.1.2.1.1**
* **snmpwalk -v2c -c prom-readonly 192.168.1.1 1.3.6.1.2.1.1**
* **snmpwalk -v2c -c prom-readonly 192.168.0.1 1.3.6.1.2.1.1**

**Toutes ont retourné la même erreur :**

**Timeout: No Response from [IP ciblée]**

* **Hypothèses actuelles sur l'échec du test**
  + **L’interface pfSense à l’adresse supposée 192.168.1.1 ou 192.168.0.1 est injoignable depuis la VM Ubuntu car elle n’est pas sur le même sous-réseau virtuel.**
  + **Le service SNMP n’est pas activé ou autorisé sur pfSense.**
  + **L’adresse publique 37.156.46.238 ne correspond pas directement à pfSense mais à une VM d’application (Jitsi).**
  + **Le filtrage résiduel par un autre pare-feu ou une couche d’abstraction empêche la communication SNMP.**
* **Prochaine étape**
  + **Vérification de l’adresse réelle de pfSense (via console, routage, ou listing Infomaniak s’il y a accès).**
  + **Si accès impossible : documenter ce blocage comme limite d’infrastructure dans un environnement managé.**
  + **Continuer à utiliser uniquement Telegraf, Node Exporter et Blackbox Exporter sur l’instance Ubuntu pour le monitoring.**
* **Conclusion temporaire Le monitoring SNMP depuis la VM vers pfSense est non fonctionnel à ce stade, malgré les ouvertures de port. Une simulation sera envisagée pour compléter le rapport de soutenance.**

**2 Sur le serveur Ubuntu :**

**Contrôle de l’existence de telegraf :**

**cd : telegraf –version**

**réponse attendue : Telegraf 1.28.2 (git: HEAD 3d7aa20c)**

**Installer Telegraf :**

**cd : sudo apt update**

**sudo apt install -y telegraf  
réponse : le dépôt n’est pas connu d’apt == E: Unable to locate package telegraf**

**Étapes pour installer Telegraf sur Ubuntu :**

**# 1. Importer la clé GPG d’InfluxData**

**cd : wget -qO- https://repos.influxdata.com/influxdata-archive\_compat.key | sudo gpg --dearmor -o /etc/apt/keyrings/influxdata.gpg ok**

**# 2. Ajouter le dépôt InfluxData pour Ubuntu (remplacer $(lsb\_release -cs) si besoin)**

**cd : echo "deb [signed-by=/etc/apt/keyrings/influxdata.gpg] https://repos.influxdata.com/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/influxdata.list ok**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**# 3. Mettre à jour les paquets**

**cd : sudo apt update**  **ok**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**# 4. Installer Telegraf**

**cd : sudo apt install telegraf -y ok**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**# 5. Vérifie le statut de Telegraf**

**cd : sudo systemctl status telegraf ok**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**# 6. Activer et démarrer Telegraf**

**cd : sudo systemctl enable --now telegraf**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**échec : Afficher le statut du service avec erreur détaillée :**

**cd : sudo systemctl status telegraf.service**

**✅ Solution directe et stable : régénérer une config minimale fonctionnelle**

**1. Sauvegarder la configuration existante :**

**Configurer Telegraf :**

**cd : [[inputs.snmp]]**

**agents = [ "192.168.1.1:161" ] # IP de pfSense**

**version = 2**

**community = "prom-readonly"**

**name = "pfsense"**

**[[inputs.snmp.field]]**

**name = "uptime"**

**oid = "1.3.6.1.2.1.1.3.0" # sysUpTime**

**[[inputs.snmp.table]]**

**name = "interface"**

**oid = "1.3.6.1.2.1.2.2"**

**Exposer Telegraf à Prometheus :**

**Créer le dossier prometheus :**

**cd : sudo mkdir -p /etc/prometheus**

**Configuration de base de Prometheus :**

**yaml :**

**global:**

**scrape\_interval: 15s**

**evaluation\_interval: 15s**

**scrape\_configs:**

**- job\_name: 'prometheus'**

**static\_configs:**

**- targets: ['localhost:9090']**

**Étapes pour l’ajouter avec nano :**

**cd : sudo nano /etc/prometheus/prometheus.yml**

**Redémarrer Prometheus:**

**cd : sudo systemctl restart prometheus**

**\*\*\*\*\*\*\*\*\***

**Installation de IPATABLES : IMPOSSIBLE Actuellement.**

**Installation de l'exporter : Il y une obsolescence du dépôt ainsi qu’une demande de code absurde sur du github.**

**PRÉREQUIS :**

**cd : sudo apt update**

**sudo apt install -y golang git make**

**VERIFICATION :**

**cd : git --version**

**go version**

**make –version**

**NETTOYAGE :**

**cd :** **cd /opt**

**sudo rm -rf iptables\_exporter**

**CLONAGE & COMPILATION :**

**cd :**

**cd iptables\_exporter**

**make**

**Utilisateur dédié :**

**cd : sudo useradd -rs /bin/false iptables\_exporter**

**sudo chown -R iptables\_exporter:iptables\_exporter /opt/iptables\_exporter**

**Service :**

**cd : sudo nano /etc/systemd/system/iptables\_exporter.service**

**contenu :**

**[Unit]**

**Description=Iptables Exporter**

**After=network.target**

**[Service]**

**User=iptables\_exporter**

**ExecStart=/opt/iptables\_exporter/iptables\_exporter**

**[Install]**

**WantedBy=multi-user.target**

**sudo systemctl daemon-reload**

**sudo systemctl enable --now iptables\_exporter**

**Autoriser le port 9145 choisi dans Service Group :**

**cd : sudo ufw allow 9145/tcp**

**Vérification du lancement du service :**

**cd : sudo systemctl status iptables\_exporter**

**\*\*\*\*\*\*\*\*\***

**Jour-9 Tercium 2nde semaine | monitoring**

**Connexion SSH en Git Bash :**

**cd : ssh -i "/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Connexion SSH en Powershell :**

**cd : ssh -i "C:\Users\test\Documents\Tercium\_Stage\ssh\_keys\tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**\*Contrôle des ports : Permissions**

**cd : sudo ufw allow 9091/tcp Prometheus**

**sudo ufw allow 9100/tcp Node Exporter**

**sudo ufw allow 9115/tcp Blackbox Exporter**

**sudo ufw allow 3000/tcp Grafana**

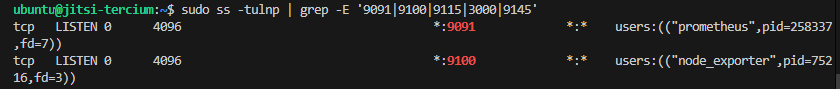
**sudo ufw allow 9145/tcp Iptables\_Exporter ECHEC**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**

**\*Contrôle des ports :Listen**

**cd : sudo ss -tulnp | grep -E ‘9001|9100|9115|3000|9145’**

****

****

**Une image contenant texte, capture d’écran, logiciel, Police

Le contenu généré par l’IA peut être incorrect.**

**Prometheus : Après modifications contenant port et les logs de blackbox\_http et pfsense\_exporter l’on peut relancer et tester l’interface.**

**Redémarrer :**

**cd : sudo systemctl restart prometheus**

**Vérifier l’état :**

**cd : sudo systemctl status prometheus**

**curl** [**http://localhost:9091/targets**](http://localhost:9091/targets)

**Navigateur web :**

**cd :** [**http://37.156.46.238:9091/targets**](http://37.156.46.238:9091/targets)

**Sécurisation HTTPS de Prometheus et Grafana**

**A. Reverse proxy avec NGINX pour Grafana (port 3000)**

**Grafana a besoin d’alimenter ses analyses via des dashboard au standard Json. Ceux-ci doivent être personnalisables et interactifs. Les buts premiers sont la surveillance et l’analyse et surtout une capacité de corriger en temps réel des problèmes et l’expérience aidant être proactif.**

**Usages principaux des dashboards dans Grafana**

**1. Supervision temps réel (real-time monitoring)**

* **Affichage de métriques en continu provenant de sources comme Prometheus, InfluxDB, Loki, Elasticsearch, etc.**
* **Exemple : suivi de l’utilisation CPU/RAM/disque d’un serveur.**

**2. Détection d’anomalies**

* **Mise en évidence de comportements anormaux via des seuils, des pics, ou des écarts.**
* **Exemples :**
  + **Un nombre élevé de requêtes 500 (erreurs HTTP).**
  + **Une latence anormalement élevée sur un service.**

**3. Alerting visuel + automatisé**

* **Couplage avec des alertes configurées (via Alertmanager, mail, Slack, etc.).**
* **Les dashboards permettent de visualiser les alertes déclenchées et leurs causes.**

**4. Diagnostic et analyse post-incident (forensic/debug)**

* **Permet de rejouer un incident dans le temps, via le sélecteur temporel.**
* **Filtrage par label, par hôte, ou par type d’erreur.**

**5. Rapports et communication**

* **Affichage synthétique pour les managers, RSSI, ou clients (en lecture seule).**
* **Possibilité d’export PDF, share URL, ou d’intégration dans un site via iframe.**

**6. Vue consolidée multi-sources**

* **Intégration de plusieurs sources de données dans un même dashboard (ex. : Prometheus + Loki + Blackbox).**
* **Corrélation des événements (ex. : logs + métriques système).**

**7. Pilotage des performances**

* **Suivi des indicateurs clés de performance (KPI) métiers ou techniques.**
* **Exemple : taux de disponibilité (uptime), temps de réponse moyen, nombre d’utilisateurs actifs.**

**8. Suivi de conformité et d’audit**

* **Vérification de l’état de sécurité et de la conformité des systèmes (ISO 27001, RGPD, NIS2…).**
* **Exemple : intégration avec Wazuh, Suricata, ou pfSense Exporter.**

**9. Visualisation de scénarios métier**

* **Dashboards orientés usage spécifique (ex. : flux réseau Jitsi, montée en charge d’une API, monitoring d’un cluster Kubernetes…).**
* **Utilisés pour tester et valider des stratégies de montée en charge ou de résilience.**

**10. Maintenance préventive**

* **Identification des machines ou services qui risquent de tomber en panne (RAM saturée, partition disque à 95 %, etc.).**
* **Utilisation dans les plans de maintenance évolutive (ITIL / ISO 22301).**

**🔧 Exemples de widgets / panels dans un dashboard :**

* **Graphes temporels (CPU, charge, requêtes)**
* **Barres ou jauges (utilisation mémoire)**
* **Statistiques tabulaires (état des services)**
* **Logs filtrables (via Loki)**
* **Heatmaps, alert lists, pie charts, singlestats, etc.**

**Infomaniak ; la gestion des groupes de sécurité pour les ports :**

Voici le **tableau récapitulatif des ports essentiels** à ouvrir par service pour ta plateforme Jitsi + Monitoring (Prometheus, Grafana, Blackbox, pfSense Exporter).

**✅ Tableau des ports par service**

| **Service** | **Port** | **Protocole** | **Type** | **Description** |
| --- | --- | --- | --- | --- |
| **SSH (admin)** | **22** | **TCP** | **Ingress** | **Connexion au serveur** |
| **HTTP** | **80** | **TCP** | **Ingress** | **Web non chiffré (utile pour redirection)** |
| **HTTPS** | **443** | **TCP** | **Ingress** | **Accès web sécurisé (ex. visio.workeezconnect)** |
| **Grafana** | **3000** | **TCP** | **Ingress** | **Interface Web Grafana** |
| **Jitsi Media** | **10000** | **UDP** | **Ingress** | **Canal de flux audio/vidéo UDP (WebRTC)** |
| **Prometheus UI** | **9090 (ou 9091)** | **TCP** | **Ingress** | **Interface Prometheus** |
| **Node Exporter** | **9100** | **TCP** | **Ingress** | **Export métriques serveur Linux** |
| **Blackbox Exporter** | **9115** | **TCP** | **Ingress** | **Tests ICMP/HTTP externes** |
| **iptables Exporter** | **9145** | **TCP** | **Ingress** | **Export règles netfilter/iptables** |

**📦 Groupes créés :**

| **Nom du groupe** | **Ports inclus** | **Utilisation** |
| --- | --- | --- |
| **default (désactivable)** | **Aucun si retiré** | **Peut être exclu** |
| **monitoring-group** | **9091, 9100, 9115, 9145** | **Prometheus + Exporters** |
| **monitoring-security-group** | **22, 80, 443** | **Accès standard SSH + HTTP/HTTPS** |
| **visio-monitoring-group** | **3000, 10000** | **Grafana + Jitsi UDP (media)** |

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

**Ce Security Group intègre l’ensemble des ports dédiés aux outils de Monitoring.**

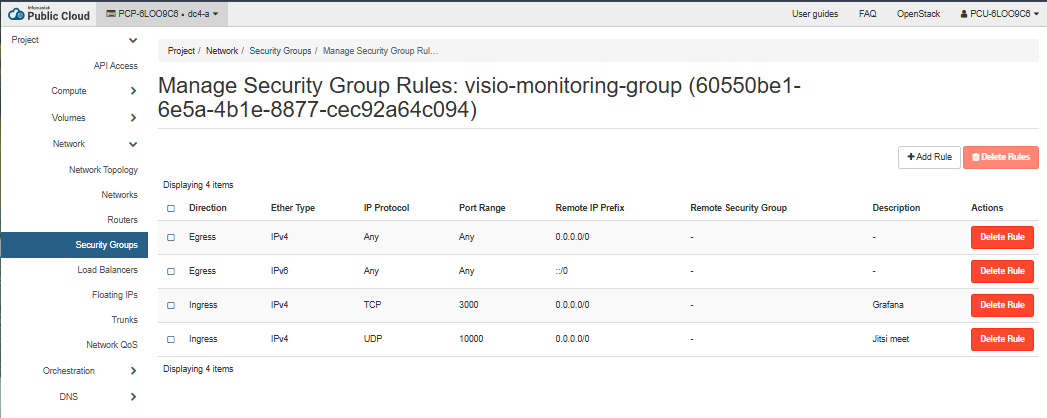
Une image contenant texte, capture d’écran, nombre, logiciel

Le contenu généré par l’IA peut être incorrect.

**Ce Security Group intègre l’ensemble des ports HTTP / HTTPS / SSH.**

**🛑 Attention**

* Le port **10000/UDP** est critique pour que la visioconférence fonctionne.
* **IPv6** peut être ignoré sauf cas spécifique (pas utile pour WebRTC sauf config avancée).
* L’on peut désormais **supprimer "default"** une fois **tous les groupes nécessaires rattachés**.



**Ce dernier Security Group intègre les ports pour Grafana & Jitsi meet.**

**Automatiser avec Terraform pour instancier Infomaniak : Possible**

**Installer Terraform puis :**

**1/ Contenu du projet**

| **Fichier** | **Rôle** |
| --- | --- |
| **clouds.yaml** | **Configuration OpenStack pour Infomaniak** |
| **main.tf** | **Création de l’instance jitsi-tercium avec le bon security\_group** |
| **security.tf** | **Règles de sécurité pour Grafana (TCP 3000) et Jitsi (UDP 10000)** |
| **variables.tf** | **Variable pour le nom de la clé SSH** |
| **.gitignore** | **Ignore les fichiers .tfstate, .terraform/ et credentials sensibles** |

**2/ Les étapes :**

1. **Configurer clouds.yaml localement dans : ~/.config/openstack/.**
2. **Renseigner les vraies valeurs dans clouds.yaml :**
   * **username, password, project\_id, etc.**
3. **Initialiser le projet avec ces commandes :**

**cd : terraform init**

**terraform plan**

**terraform apply**

**Structure recommandée du projet**

**project/**

**├── main.tf**

**├── variables.tf**

**├── terraform.tfvars.example**

**├── terraform.sh**

**3/ Exemple de fichier : Nécessaires à Infomaniak**

**Terraform.tfvars**

**# Clé SSH utilisée pour accéder à l'instance**

**keypair\_name = "tercium-instance\_key"**

**# Nom de l'image (ID ou nom exact d’Infomaniak OpenStack)**

**image\_name = "Debian 12 Bookworm"**

**# Nom du flavor (ex: gp1.large, cp1.medium, etc.)**

**flavor\_name = "gp1.large"**

**# Nom du réseau privé ou réseau par défaut d’Infomaniak**

**network\_name = "Ext-Net1"**

**# Groupe de sécurité à appliquer**

**security\_group\_name = "jitsi-monitoring-secgroup"**

**# Nom d’instance (machine virtuelle)**

**instance\_name = "jitsi-tercium"**

**Modèles de fichier :**

**1/ main.tf**

**provider "openstack" {**

**auth\_url = var.auth\_url**

**tenant\_name = var.project\_name**

**user\_name = var.username**

**password = var.password**

**region = var.region**

**domain\_name = var.domain\_name**

**}**

**# Groupe de sécurité personnalisé**

**resource "openstack\_networking\_secgroup\_v2" "jitsi\_secgroup" {**

**name = "jitsi\_monitoring\_secgroup"**

**description = "Allow SSH, HTTP, HTTPS, Prometheus, Grafana, Jitsi, Node Exporter, Blackbox"**

**}**

**# Règles de sécurité essentielles**

**resource "openstack\_networking\_secgroup\_rule\_v2" "rules" {**

**count = length(var.security\_rules)**

**direction = var.security\_rules[count.index]["direction"]**

**ethertype = var.security\_rules[count.index]["ethertype"]**

**protocol = var.security\_rules[count.index]["protocol"]**

**port\_range\_min = var.security\_rules[count.index]["port\_range\_min"]**

**port\_range\_max = var.security\_rules[count.index]["port\_range\_max"]**

**remote\_ip\_prefix = var.security\_rules[count.index]["remote\_ip\_prefix"]**

**security\_group\_id = openstack\_networking\_secgroup\_v2.jitsi\_secgroup.id**

**}**

**# Clé SSH**

**resource "openstack\_compute\_keypair\_v2" "default" {**

**name = "tercium-key"**

**public\_key = file(var.public\_key\_path)**

**}**

**# Instance Jitsi**

**resource "openstack\_compute\_instance\_v2" "jitsi\_vm" {**

**name = "jitsi-tercium"**

**image\_name = var.image\_name**

**flavor\_name = var.flavor\_name**

**key\_pair = openstack\_compute\_keypair\_v2.default.name**

**security\_groups = [openstack\_networking\_secgroup\_v2.jitsi\_secgroup.name]**

**network {**

**name = var.network\_name**

**}**

**}**

**2/ variables.tf**

**variable "auth\_url" {}**

**variable "username" {}**

**variable "password" {}**

**variable "domain\_name" {}**

**variable "project\_name" {}**

**variable "region" {}**

**variable "public\_key\_path" {}**

**variable "image\_name" {}**

**variable "flavor\_name" {}**

**variable "network\_name" {}**

**variable "security\_rules" {**

**type = list(object({**

**direction = string**

**ethertype = string**

**protocol = string**

**port\_range\_min = number**

**port\_range\_max = number**

**remote\_ip\_prefix = string**

**}))**

**}**

**3/ terraform.tfvars**

**auth\_url = "https://api.infomaniak.com/identity/v3"**

**username = "TON\_IDENTIFIANT"**

**password = "TON\_MOT\_DE\_PASSE"**

**domain\_name = "default"**

**project\_name = "NomDuProjet"**

**region = "dc3-a"**

**public\_key\_path = "~/.ssh/id\_rsa.pub"**

**image\_name = "Ubuntu 22.04"**

**flavor\_name = "a1-ram2-disk20-perf1"**

**network\_name = "Ext-Net"**

**security\_rules = [**

**{**

**direction = "ingress"**

**ethertype = "IPv4"**

**protocol = "tcp"**

**port\_range\_min = 22**

**port\_range\_max = 22**

**remote\_ip\_prefix = "0.0.0.0/0"**

**},**

**{**

**direction = "ingress"**

**ethertype = "IPv4"**

**protocol = "tcp"**

**port\_range\_min = 80**

**port\_range\_max = 80**

**remote\_ip\_prefix = "0.0.0.0/0"**

**},**

**{**

**direction = "ingress"**

**ethertype = "IPv4"**

**protocol = "tcp"**

**port\_range\_min = 443**

**port\_range\_max = 443**

**remote\_ip\_prefix = "0.0.0.0/0"**

**},**

**{**

**direction = "ingress"**

**ethertype = "IPv4"**

**protocol = "tcp"**

**port\_range\_min = 3000**

**port\_range\_max = 3000**

**remote\_ip\_prefix = "0.0.0.0/0"**

**},**

**{**

**direction = "ingress"**

**ethertype = "IPv4"**

**protocol = "tcp"**

**port\_range\_min = 10000**

**port\_range\_max = 10000**

**remote\_ip\_prefix = "0.0.0.0/0"**

**}**

**]**

**Jour-10 Tercium 2nde semaine | monitoring**

**Connexion SSH en Git Bash :**

**cd : ssh -i "/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Connexion SSH en Powershell :**

**cd : ssh -i "C:\Users\test\Documents\Tercium\_Stage\ssh\_keys\tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Relancer les services système via le terminal :**

**cd : sudo systemctl restart prometheus**

**sudo systemctl restart node\_exporter**

**sudo systemctl restart blackbox\_exporter**

**Pour vérifier qu’ils tournent :**

**cd : sudo systemctl status prometheus**

**sudo systemctl status node\_exporter**

**sudo systemctl status blackbox\_exporter**

**Bonnes pratiques :**

* **Prometheus doit être actif pour collecter les métriques.**
* **Node Exporter doit tourner sur chaque machine à surveiller.**
* **Blackbox Exporter doit être lancé une fois (souvent sur la même machine que Prometheus) pour faire les tests de ping, HTTP, TCP, etc.**
* **Grafana se connecte à Prometheus. Il n’a pas besoin d’être redémarré sauf en cas de changement de configuration du data source ou de plantage.**

**Astuce : automatisation au démarrage :**

**cd : sudo systemctl enable prometheus**

**sudo systemctl enable node\_exporter**

**sudo systemctl enable blackbox\_exporter**

**Accès aux interfaces Web**

**Si tous les services sont actifs, tu peux accéder aux interfaces suivantes :**

| **Composant** | **URL locale** | **URL distante (publique)** |  |
| --- | --- | --- | --- |
| **Prometheus** | **http://localhost:9090** | **http://37.156.46.238:9090 *(si exposé)*** |  |
| **Node Exporter** | **http://localhost:9100** | **http://37.156.46.238:9100 *(si exposé)*** |  |
| **Blackbox Exporter** | **http://localhost:9115** | **http://37.156.46.238:9115 *(si exposé)*** |  |
| **Grafana** | **http://localhost:3000** | [**http://37.156.46.238:3000**](http://37.156.46.238:3000) **✅** |  |

**TESTS :**

**Vérifie que ton service est bien bindé sur ce port :  
Exemple avec NGINX : Interdit ici**

**cd : sudo netstat -tulnp | grep 4433**

**ou**

**cd : sudo ss -tuln | grep 4433**

**Test à distance depuis une autre machine :**

* **règle de composition nc -vz <IP> <PORT>**

**cd : nc -vz 37.156.46.238 4433**

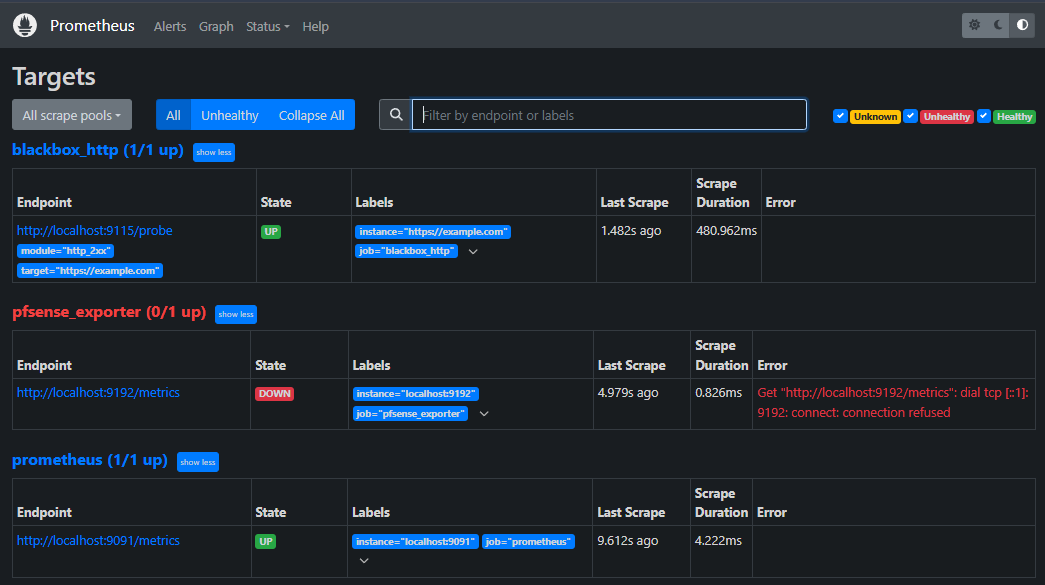
| **Sortie nc -vz** | **Signification technique** |
| --- | --- |
| **Connection to 37.156.46.238 port 4433 succeeded!** | **🔓 Le port est ouvert et une application y écoute** |
| **Connection refused** | **🚫 Aucun service n’écoute sur ce port** |
| **Connection timed out** | **❌ Le port est filtré (firewall ou IP injoignable)** |

**ou via navigateur :**

**https://37.156.46.238:4433**

**Manuel d’usage de visio.workeezconnect.fr**

**Récupération et analyse des données monitoring :**

****

**Problème pfsense\_exporter est inactif : Pourquoi ces commandes ne fonctionnent pas ?**

**cd : sudo systemctl status pfsense\_exporter**

**sudo systemctl restart pfsense\_exporter**

**Retour ce service de monitoring du 8 ième jour : Je compléte et je modifie.**