**Rapport d’étapes : 30-06-2025 // 01-07-2025 // 02-07-2025 // 03 -07-2025 // 04 -07-2025.**

**Semaine-1**

Parfait. Voici une synthèse structurée pour t’aider à cadrer les suites de ton journal, en cohérence avec ton projet en 4 points et ton audit matériel à venir :

**✅ À ce stade (Jour 3 sur 50)**

**✔️ Réalisé :**

* Instanciation manuelle d’une VM OpenStack sur Infomaniak
* Connexion sécurisée SSH (keypair + configuration sshd\_config)
* Installation serveur Apache (LAMP) et test HTTP local
* Début d’intégration de certificats SSL (objectif : HTTPS sécurisé)
* Préparation à l’audit matériel (micro-entreprises, NAS, usage cloud)

**Pistes critiques immédiates – sécurité / SSL**

**1. Durcissement initial à renforcer :**

| **Zone** | **Action recommandée** |
| --- | --- |
| **SSH** | **AllowUsers, désactivation root, logs auth.log** |
| **Firewall** | **Vérification des security-groups Infomaniak** |
| **Ports ouverts** | **sudo ss -tuln, restreindre à 80/443 uniquement** |
| **Fail2Ban** | **À prévoir rapidement si usage exposé** |
| **Certificat** | **Génération propre avec CERFORED (PEM/CRT) — validité, autorité, renouvellement automatique** |

**2. Gestion des certificats SSL :**

* Prépare une procédure complète :
  + Génération CSR avec openssl
  + Soumission à CERFORED
  + Installation (/etc/ssl/certs, /etc/ssl/private)
  + Configuration Apache (SSLEngine, SSLCertificateFile, etc.)
  + Test via navigateur + curl -v https://...
  + Planification du renouvellement manuel ou automatique

**Pour ton audit matériel Lundi / Mardi Prochain:**

**🗂️ Checklist minimum à préparer :**

| **Équipement / VM** | **OS / version** | **Rôle fonctionnel** | **Utilisateur** | **Sécurité (authentification, pare-feu)** | **Sauvegarde** |
| --- | --- | --- | --- | --- | --- |
| PC1 | Win10 Pro | NAS local | Dev | Bitlocker, compte local | Rsync/NAS |
| PC2 | Ubuntu 22.04 | Frontend Web | DevOps | UFW, fail2ban, clé SSH | GitHub + Cloud |
| etc. | ... | ... | ... | ... | ... |

➡️ Modèle à décliner sur 3 à 5 machines max (pertinent pour micro-entreprises)

**🔁 Complémentarité avec ton projet 4 points :**

| **Élément** | **À démarrer après audit** |
| --- | --- |
| Déploiement dynamique | Jitsi ou backend Flask (via Docker ?) |
| Monitoring | Prometheus + Grafana, Wazuh (option) |
| Backup & PCA/PRA | Snapshot OpenStack + sauvegardes NAS |
| Certificats SSL | Centralisation + procédure CERFORED |

Souhaites-tu que je t’aide à :

* Structurer ton modèle d’audit matériel à remplir sur site ?
* Générer un script Apache + SSL compatible CERFORED ?
* Intégrer une check-list PRA/PCA avec snapshot OpenStack et export .pem ?

**Certification SSL / TLS et l’IP + Domaine: Mercredi :**

**Oui, avec Infomaniak Public Cloud (OpenStack), tu as tout à fait raison :  
tu peux instancier une VM (même vide), ne pas la détruire, lui attribuer un nom de domaine via DNS, et maîtriser les coûts tant que tu ne consommes pas de ressources excédentaires.**

**✅ RÉPONSE STRUCTURÉE POUR INSTANCIATION INFOMANIAK**

**🔹 1. Instance vide = coût minimal ?**

**Oui, à condition de :**

* Ne pas l’allumer constamment,
* Ne pas lui attacher un volume massif (> 20 Go),
* Ne pas la lier à une IP flottante inutilisée,
* \*\*Choisir un type d’instance basique (ex : B1.1 = 1 vCPU / 1 Go RAM).

**💡 Une VM arrêtée ne consomme que l’espace disque + IP si statique.  
Si tu libères l’IP flottante → coût = stockage uniquement.**

**🔹 2. Peux-tu associer un nom de domaine à une IP d’Infomaniak ?**

**✅ Oui. Procédure :**

1. **Depuis ton manager Infomaniak DNS (ou tout registrar tiers) :**
   * **Accède à la zone DNS du domaine (tercium.xyz, test-jitsi.io, etc.)**
   * **Ajoute un enregistrement A :**
2. **Type: A**
3. **Nom: jitsi.tercium.xyz**
4. **Valeur: 185.XXX.XXX.XXX (IP flottante Infomaniak de l’instance)**
5. **TTL: 3600**
6. **Attends la propagation DNS.**
7. **Teste avec : dig +short jitsi.tercium.xyz & curl -Iv http://jitsi.tercium.xyz**

**🔹 3. Comment garder l'IP sans surcoût ?**

* **Utilise une IP flottante uniquement quand nécessaire.**
* **Si la machine est arrêtée, pense à :**
  + Soit libérer **l’IP flottante,**
  + Soit la conserver et l’exploiter dans ton **DNS.**

**💡 Conseil : associer ton domaine au nom DNS dès le début et tester Certbot uniquement quand la VM est en ligne et que l’IP est attribuée**

**📌 En résumé opérationnel :**

| **Action** | **Possibilité** | **Remarques** |
| --- | --- | --- |
|  |  |  |
| **Instancier une VM vide** | **✅ Oui** | **Choisir un gabarit minimal (B1.1)** |
| **Conserver IP + associer à domaine** | **✅ Oui** | **Enregistrement A vers IP flottante** |
| **Coût d’une VM arrêtée** | **⚠️ Disque + IP flottante** | **Supprimer IP flottante si inutilisée** |
| **Test avec Certbot + Apache** | **✅ Dès que IP + DNS OK** | **Ports 80/443 doivent être ouverts** |

Voici une **checklist complète Infomaniak / OpenStack / Certbot** pour **statuer et finaliser proprement la certification SSL/TLS** demain. Elle couvre :

* l’instanciation minimale (coût maîtrisé),
* l’attribution d’une IP flottante,
* la liaison DNS (nom de domaine),
* la configuration Apache + Certbot.

**✅ CHECKLIST – INFOMANIAK OPENSTACK + CERTIFICATION SSL/TLS**

**🧱 1. INSTANCIATION MINIMALE**

| **Action** | **Description** | **Fait ✔️ / ❌** |
| --- | --- | --- |
| Créer une VM (via Horizon ou CLI) | Choisir un flavor léger : B1.1 (1 vCPU, 1 Go RAM) |  |
| Sélectionner une image Ubuntu stable | Ex : Ubuntu 22.04 |  |
| Attacher un disque de taille réduite | 10–20 Go suffisent (évite surcharge) |  |
| Générer une paire de clés SSH | Sauvegarder \*.pem en local (sécurisé) |  |
| Associer un réseau privé et public (NAT ou floating IP possible) |  |  |

**🌐 2. GESTION DES IP (RÉSEAU)**

| **Action** | **Description** | **Fait ✔️ / ❌** |
| --- | --- | --- |
| Créer ou réserver une IP flottante | Dans l’onglet « Réseau > IP flottantes » |  |
| Associer cette IP à l’interface réseau de l’instance | VM > Réseau > Associer IP flottante |  |
| Vérifier accessibilité de l’IP publique | ping et curl depuis machine locale |  |

**📛 3. LIAISON NOM DE DOMAINE (DNS)**

| **Action** | **Description** | **Fait ✔️ / ❌** |
| --- | --- | --- |
| Se connecter à la gestion DNS du domaine | Infomaniak, OVH, Gandi, etc. |  |
| Créer un enregistrement A | jitsi.tercium.xyz → IP publique |  |
| Ajouter un enregistrement TXT (facultatif, DNS-01 challenge) | Si besoin méthode DNS |  |
| Tester la résolution DNS | dig +short, ping, curl -Iv |  |

**🔐 4. INSTALLATION ET CERTIFICATION HTTPS**

| **Action** | **Description** | **Fait ✔️ / ❌** |
| --- | --- | --- |
| Installer Apache2 | sudo apt install apache2 |  |
| Vérifier qu’un site test est servi | echo "test" > /var/www/html/index.html |  |
| Ouvrir les ports 80 et 443 dans le groupe de sécurité | Règles de firewall OpenStack |  |
| Installer Certbot + plugin Apache | sudo apt install certbot python3-certbot-apache -y |  |
| Exécuter la commande de certification | sudo certbot --apache -d jitsi.tercium.xyz |  |
| Vérifier le certificat | https://jitsi.tercium.xyz + openssl s\_client |  |
| Ajouter tâche cron ou timer certbot renew | Assure renouvellement automatique |  |

**📁 5. MAINTIEN COÛT MINIMAL (optionnel mais conseillé)**

| **Action** | **Description** | **Fait ✔️ / ❌** |
| --- | --- | --- |
| Arrêter la VM quand non utilisée | shutdown ou via Horizon |  |
| Libérer l’IP flottante si inutilisée | Attention, l’IP sera perdue |  |
| Vérifier stockage de disques persistants | Supprimer volumes inutiles |  |
| Étiqueter les ressources projet-stage | Facilite le suivi budgétaire |  |

**🧩 CONSEILS FINAUX**

* **Ne jamais lancer Certbot tant que le domaine ne pointe pas vers la bonne IP**.
* **Si besoin local**, commencer avec un **certificat auto-signé** avant Let’s Encrypt.
* **Documenter chaque étape** → captures, logs, tests (curl -Iv, openssl s\_client, etc.).
* **Durée de validité Let’s Encrypt** : 90 jours → penser au renouvellement auto.