**Objectif : Créer un dépôt GitHub centralisé**

* **Versionner tous les fichiers critiques (.ini, .json, .yml, etc.) liés à Jitsi-Meet.**
* **Automatiser leur transfert vers l'instance Infomaniak, malgré une instanciation manuelle locale.**
* **Garantir une synchronisation distante fiable et traçable (CI/CD ou push contrôlé).**

**Étapes nécessaires**

**1. Créer un dépôt GitHub structuré**

* **Arborescence suggérée :**
* **jitsi-meet-infra/**
* **├── etc/**
* **│ ├── jitsi/**
* **│ ├── prosody/**
* **│ └── nginx/**
* **├── opt/**
* **│ └── jitsi/**
* **├── scripts/**
* **│ ├── push\_to\_infomaniak.sh**
* **│ └── update\_configs.sh**
* **├── .github/**
* **│ └── workflows/**
* **│ └── deploy.yml (si CI/CD)**
* **└── README.md**

**2. Automatiser la synchronisation vers Infomaniak**

**Option A — SSH + rsync ou scp depuis un script Bash**

* **Écriture d’un push\_to\_infomaniak.sh : Une image contenant texte, Police, capture d’écran

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

**Option B — CI/CD avec GitHub Actions**

* **Déploiement conditionné par main ou tag : .github/workflows/deploy.yml**

**Une image contenant texte, capture d’écran

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

**⚠️ Requiert un accès SSH avec clé privée configurée via GitHub Secrets.**

**3. Faut-il utiliser n8n ?**

| **Besoin** | **n8n est-il nécessaire ?** |
| --- | --- |
| **Déployer des fichiers .ini, .yml, .json vers un serveur distant via SSH/SCP/rsync** | **❌ Non** |
| **Automatiser une suite de tâches conditionnelles, multi-services (ex : validation + appel API + trigger de backup)** | **✅ Oui** |
| **Intégrer des APIs externes (GitHub, Slack, mail, webhook) pour informer ou agir sur événements** | **✅ Oui** |
| **Avoir une interface graphique de pipeline sans script** | **✅ Oui** |

**Donc :**

**❌ Non, n8n n'est pas indispensable pour ton usage actuel (pousser des fichiers de conf).  
✅ Il devient utile si tu veux déclencher des actions complexes ou multi-étapes (ex : valider, notifier, exécuter, journaliser dans Notion).**

**Recommandation**

**Pour ton cas (Jitsi / Infomaniak) :**

* **Priorise rsync/scp** avec **clé SSH,** ou **GitHub Actions** si tu veux **pousser via git.**
* **Si besoin** futur de workflow visuel multi-actions **: ajoute n8n (ex : déclenchement si IP change, notification Telegram, backup GCP, etc.).**

**1. Arborescence GitHub**

**Nom du dépôt proposé** **: jitsi-meet-infra**

**jitsi-meet-infra/**

**├── etc/**

**│ ├── jitsi/**

**│ ├── prosody/**

**│ └── nginx/**

**├── opt/**

**│ └── jitsi/**

**├── scripts/**

**│ ├── push\_to\_infomaniak.sh**

**│ └── update\_configs.sh**

**├── .github/**

**│ └── workflows/**

**│ └── deploy.yml**

**└── README.md**

**Copier tes fichiers réels dans les sous-dossiers etc/ et opt/, puis les versionner avec Git.**

**2. Script push\_to\_infomaniak.sh**

**Nom du fichier** : **push\_to\_infomaniak.sh  
Emplacement** : scripts/push\_to\_infomaniak.sh  
**Rôle** : Transférer manuellement tous les **fichiers critiques /etc/ et /opt/** vers le **serveur Infomaniak via rsync + SSH**

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Ne pas oublier pas : chmod +x scripts/push\_to\_infomaniak.sh**

**3. Fichier GitHub Actions : deploy.yml**

**Nom du fichier** : deploy.yml  
**Emplacement** : **.github/workflows/deploy.yml  
Rôle** : Déclencher un déploiement automatique sur push (branche main)  
**Prérequis** :

* Ajoute dans les **GitHub Secrets** :
  + **SSH\_PRIVATE\_KEY**
  + **REMOTE\_USER**
  + **REMOTE\_HOST**

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

**Deux compléments, pour finaliser un dépôt GitHub propre et opérationnel :**

**4. Makefile**

**Nom du fichier : Makefile  
Emplacement : racine du projet (jitsi-meet-infra/)  
Rôle : Simplifier l'exécution de commandes avec des cibles explicites (make deploy, make push, make check…)**

**Une image contenant texte, capture d’écran

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

**✔️ make deploy exécute ton script Bash.  
✔️ make push lance rsync directement.  
✔️ make check teste si ton SSH est bien configuré.**

**5. README.md — Procédure Markdown**

**Nom du fichier : README.md  
Emplacement : racine du dépôt  
Rôle : Documenter la structure et les instructions d’utilisation (locale ou CI/CD)**

**jitsi-meet-infra/  
├── etc/ # Fichiers système de /etc/jitsi, /etc/prosody, etc.  
├── opt/ # Configuration et modules spécifiques de /opt/jitsi  
├── scripts/ # Scripts de push SSH et de mise à jour  
├── .github/workflows # Déploiement automatique via GitHub Actions  
├── Makefile # Commandes rapides (push, deploy, check)  
└── README.md**

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

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

**Configuration SSH (manuel ou GitHub Actions)**

* **Génère une paire de clés SSH : cd : ssh-keygen -t rsa -b 4096 -C "deploy@jitsi"**

**Pour sécuriser les échanges entre ton terminal, GitHub, et ton serveur Infomaniak :**

**Résumé des rôles des clés SSH**

| **Destination** | **Clé recommandée** | **Usage** |
| --- | --- | --- |
| **GitHub (Git push/pull)** | ✅ **Clé SSH spécifique GitHub** | Authentification au dépôt distant |
| **Serveur Infomaniak** | ✅ **Clé SSH machine/infra** | Déploiement automatisé via rsync, scp, etc. |
| **GitHub Actions → SSH** | ✅ **Clé dédiée CI/CD (privée dans Secrets)** | Déploiement distant automatisé |

**Cas actuels et conseils**

**1. ✅ Tu as déjà une clé SSH locale pour GitHub**

Si elle fonctionne (git push sans mot de passe), tu peux **la garder pour ton usage personnel**.

Vérifie avec :

**cd : ssh -T git@github.com**

**2. ✅ Clé SSH pour le serveur Infomaniak**

Clé locale que tu ajoutes via :

**cd : ssh-copy-id -i ~/.ssh/id\_rsa.pub ubuntu@IP\_INFOMANIAK**

**3. ✅ Clé SSH pour GitHub Actions (CI/CD)**

Tu dois **générer une nouvelle paire de clés dédiée au bot GitHub Actions** (et **ne pas** utiliser celle de ton terminal !)

**Étapes :**

# Sur ton terminal local (pas sur GitHub Actions)

**cd :** ssh-keygen -t rsa -b 4096 -f ~/.ssh/github\_action\_rsa -C "gh-action@jitsi-infra"

Cela crée :

* github\_action\_rsa (clé privée)
* github\_action\_rsa.pub (clé publique)

**Ensuite :**

* **Sur Infomaniak** :

Ajoute la clé **publique** (.pub) dans ~/.ssh/authorized\_keys de l'utilisateur ubuntu

* **Sur GitHub (repo > Settings > Secrets)** :
  + Ajoute la **clé privée** comme secret SSH\_PRIVATE\_KEY

**Résumé final des clés utilisées**

| **Contexte** | **Fichier (exemple)** | **Localisation** | **Usage** |
| --- | --- | --- | --- |
| **GitHub (terminal perso)** | **~/.ssh/id\_rsa\_github** | Local uniquement | **Git push/pull** |
| **Serveur Infomaniak (local)** | **~/.ssh/id\_rsa\_infomaniak** | Local + authorized\_keys | **scp, rsync, ssh** |
| **GitHub Actions → SSH** | **github\_action\_rsa (privée)** | **Privée dans GitHub**, publique sur le serveur | **CI/CD secure** |