

Année 2024-2025

Annexe Technique



Cloud



Benoît ROBART

Table des matières

Sommaire

1. Instanciation dans le Cloud : Infomaniak	page 3
• 1.1 Présentation du fournisseur	page 3
• 1.2 Objectif visio-conférence avec Jitsi Meet	page 3
• 1.3 Spécificités techniques et sécurisation des réunions	page 4
• 1.4 Matériel, compatibilité et réseau requis	page 4 - 5
• 1.5 Étapes : création des clefs, instances, zones	page 6 - 8
2. Connexion SSH et gestion des accès	page 8
• 2.1. Génération et enregistrement des clefs	page 8 - 9
• 2.2. Configuration SSH (VSCode, GitBash, Putty)	page 10 - 11
• 2.3. Connexions testées, erreurs fréquentes et solutions	page 11 - 12
• 2.4. Usage & Compréhension : Fingerprint	page 12 - 13
3. Tests fonctionnels de l'instance	page 13
• 3.1. Ping & accès HTTP	page 13 - 14
• 3.2. Permissions, index.html, test Apache	page 14
4. Certification SSL avec Certbot (HTTPS)	page 14
• 4.1. Installation locale	page 14 – 15
• 4.2. Synthèse du contrat Let's Encrypt (LE-SA v1.5)	page 15 - 16
• 4.3. Obligations techniques et juridiques	page 16
5. Proposition Bonus	page 17
• 5.1. Alias Git Bash personnalisé	page 17
• 5.2. Déploiement Apache local via WSL2	page 17
• 5.3. Cas test localhost et localhost:80	page 18

1. Instanciation dans le Cloud : Infomaniak

1.1 Aborder Infomaniak : Cloud Provider Suisse

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

Infomaniak [Public Cloud](#), une solution [Infrastructure as a service \(IaaS\)](#) basée sur [OpenStack](#) 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/>

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

1.2 Objectif visio-conférence : Jitsi-Meet

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.

1.3 Spécificités techniques et sécurisation des réunions

Besoins matériels :

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

These requirements concern the videobridge. If there are only external videobridges (as can be the case on high end Jitsi Meet servers), network performance matters much less.

- **RAM:** it's usually suggested to get 8 GB. For small meetings you can get away with 4 GB, for test servers or very small meetings you can try to use 2 GB. For big meetings it's suggested to go the scalable way over getting huge amounts of memory.
- **CPU:** very low processor performance can seriously harm a real time system, especially when using a shared server (where your CPU performance can be stolen by other customers of your hoster, check on 'dedicated CPU' if you are getting a VPS, rather than a physical server). However, a consideration is that a Jitsi Meet component, Prosody, can only use ONE (1) core. So getting a lot of cores, let's say more than 32, is not always useful. For a basic server, 4 dedicated cores can be enough.
- **Disk:** unless you are doing heavy logging or have very specific needs, you can get away with 20 Gbytes of standard hard disk. SSD are more a nice to have than a necessity.

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>

1.4 Matériel, compatibilité et réseau requis

Create a keypair :

CLI Horizon

```
openstack keypair create my_keypair > ~/.ssh/my_keypair
chmod 600 ~/.ssh/my_keypair
```

The private key is saved to `~/.ssh/my_keypair` and can then be used for instance creation (using the `--key-pair` argument on the command line) and/or for SSH.

You may now use the keypair to connect to a instance with this command :

```
ssh -i ~/.ssh/ ~/.ssh/ssh_my_rsa_key adminuser@my-instance-name
```

Create an instance :

Choose a flavor

```
$ openstack flavor list
```

ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
21aad244-a330-4e79-ba80-4c057cf742f9	a1-ram2-disk20-perf1	2048	20	0	1	True
7918af3e-aa2a-4aa4-976d-9056490a4654	a4-ram8-disk20-perf1	8192	20	0	4	True
a1d6e394-e4db-486b-8091-5d95cfbf3952	a12-ram24-disk20-perf1	24576	20	0	12	True
a35c6646-0f3c-464b-b50d-2a76cad0bd7b	a16-ram32-disk20-perf1	32768	20	0	16	True
...						
b6b7baeb-2328-48c9-8543-88cccec8ec4b	a2-ram4-disk20-perf1	4096	20	0	2	True
cd0483a8-ca2a-466b-89b2-f8d0d005408a	a8-ram16-disk20-perf1	16384	20	0	8	True

Choose an image (operating System):

Choose an image (Operating System)

```
$ openstack image list
```

ID	Name	Status
097480e6-16bc-4a50-a7af-e34399d039ac	cirros-0.3.4	active
735a5c16-56f0-4c15-80e7-49056dbc4f71	Debian 10.10 buster	active
49f425ed-bf79-46ab-8cf3-44935d9d831e	Debian 11 bullseye	active
...		
daf43e02-f59e-45bb-8d63-436094d3f360	Ubuntu 21.04	active

Exemple de création :

Create your instance

```
$ openstack server create --image "Debian 11 bullseye" --flavor a2-ram4-disk20-perf1 --key-name mykeypair --network ext
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-STS:power_state	NOSTATE
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	None
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	
adminPass	mii5bBNRGRF6
config_drive	
created	2021-02-24T15:51:17Z
flavor	a2-ram4-disk20-perf1 (b6b7baeb-2328-48c9-8543-88cccec8ec4b)
hostId	
id	5bf0ebf6-825d-4879-b4b8-90245ec4dc19
image	Debian 11 bullseye
key_name	mykeypair
name	infomaniak-vm-1
progress	0
project_id	ac4fafd60021431585bbb23470119557
properties	
security_groups	name='default'
status	BUILD
updated	2021-02-24T15:51:17Z
user_id	b1580497f51e4d10b9110c60c154562c
volumes_attached	

1.5 Étapes : création des clefs, instances, zones

Contrôle de création d'Instance:

Check your instance is active

```
$ openstack server show infomaniak-vm-1
+-----+-----+
| Field | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | b10 |
| OS-EXT-STS:power_state | Running |
| OS-EXT-STS:task_state | None |
| OS-EXT-STS:vm_state | active |
| OS-SRV-USG:launched_at | 2021-02-24T15:51:27.000000 |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | ext-net1=2001:1600:115:1::3d8, 195.15.242.18 |
| config_drive | |
| created | 2021-02-24T15:51:17Z |
| flavor | a2-ram4-disk20-perf1 (b6b7baeb-2328-48c9-8543-88cccec8ec4b) |
| hostId | 1baedae8de146b81f259cfec3cf33fcae980bb274f8fef46a5f49ba9 |
| id | 5bf0ebf6-825d-4879-b4b8-90245ec4dc19 |
| image | Debian 11 bullseye |
| key_name | mykeypair |
| name | infomaniak-vm-1 |
| progress | 0 |
| project_id | ac4fafd60021431585bbb23470119557 |
| properties | |
| security_groups | name='default' |
| status | ACTIVE |
| updated | 2021-02-24T15:51:27Z |
| user_id | b1580497f51e4d10b9110c60c154562c |
| volumes_attached | |
+-----+-----+
```

Firewall :

Firewall

By default, no incoming traffic is allowed to your instance but the outgoing traffic is allowed.

To allow the SSH connection you have to add a rule to the `default` security group this way :

```
openstack security group rule create --ingress --protocol tcp --dst-port 22 --ethertype IPv4 default
```

Clef SSH : modèle

```
ssh -i path/to/your/private_key_file debian@195.15.242.18
```

Availability Zones : openstack availability zone list

```
taylor@laptop:~$ openstack availability zone list --compute
+-----+-----+
| Zone Name | Zone Status |
+-----+-----+
| dc3-a-04 | available |
| dc3-a-09 | available |
| dc3-a-10 | available |
+-----+-----+
```

Coût d'une instanciación pour la compagnie : Simulation possible avec les choix matériels.

Public Cloud - Infrastructure n°1

Lien vers cette estimation :

<https://infomaniak.cloud/calculator?uuid=ai>

Instances

1 élément
10.59 € / mois

Block Storage

1 élément
0.0803 € / mois

Object Storage

1 élément
0.00949 € / mois

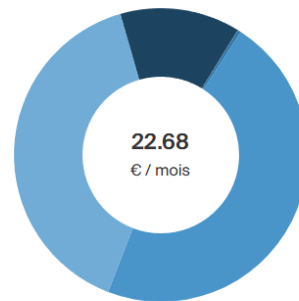
Adresses IP publiques

1 élément
3.00 € / mois

Load balancer

1 élément
9.01 € / mois

Résumé de l'estimation



Instances (1)
10.59 € / mois

Block Storage (1)
0.0803 € / mois

Object Storage (1)
0.00949 € / mois

Adresses IP publiques (1)
3.00 € / mois

Load balancer (1)
9.01 € / mois

Le calculateur fournit uniquement une estimation. Les frais réels dépendent de votre utilisation effective du service et d'autres facteurs, comme les taux de change ou les arrondis.

Détails

Détails :

Instances (1)

	Quantité:	Temps de fonctionnement:	Système d'exploitation:	Total:
 a4_ram8_disk0	1	730h/mois	Linux	10.59 € / mois


Block Storage (1)

	Quantité:	Stockage:	Total:
 Perf1	1	1Go	0.0803 € / mois


Object Storage (1)

	Quantité:	Stockage:	Bande passante:	Total:
 Object storage	1	1Go	1Go	0.00949 € / mois

Adresses IP publiques (1)

	Quantité:	Total:
 IPv4 réservée	1	3.00 € / mois

Load balancer (1)

	Quantité:	Total:
 Load balancer standard	1	9.01 € / mois

Rappel :

Instances 1 élément 10.59 € / mois	1
Block Storage 1 élément 0.0803 € / mois	2
Object Storage 1 élément 0.00949 € / mois	3
Adresses IP publiques 1 élément 3.00 € / mois	4
Load balancer 1 élément 9.01 € / mois	5
Total 22.68 € / mois	

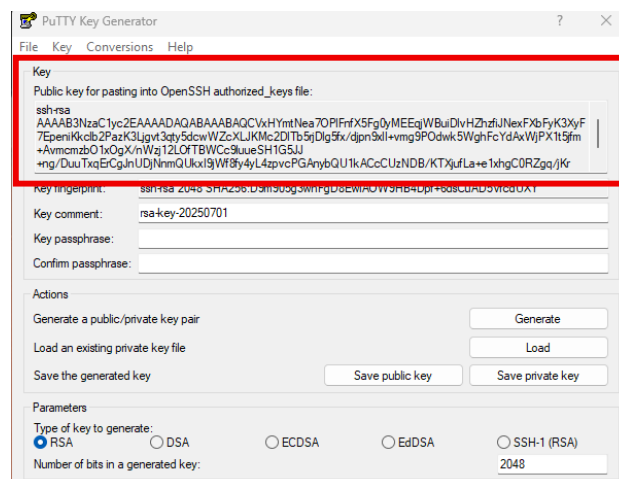
Choix des availability Zone : az-1 / az-2 / az-3

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).

2. Connexion SSH et gestion des accès

2.1. Génération et enregistrement des clefs

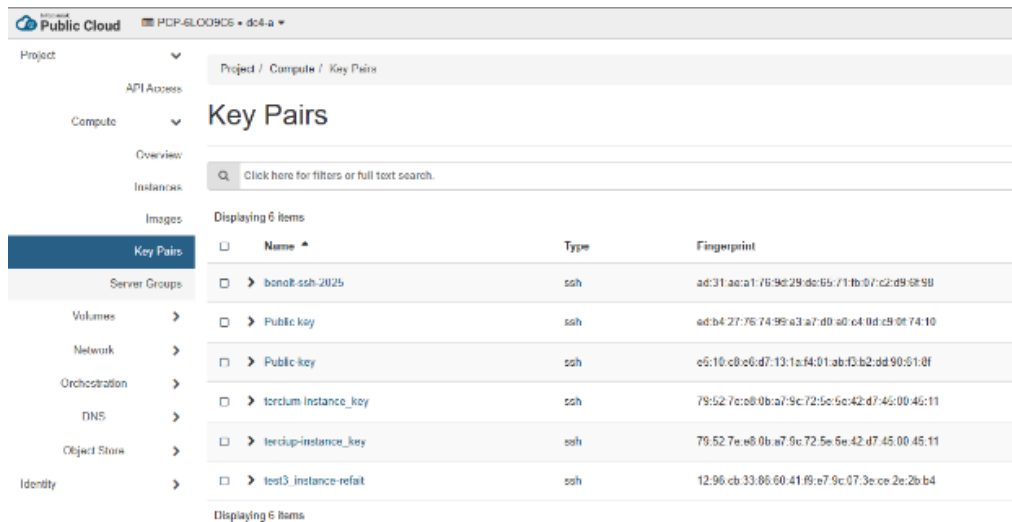


Putty Key Generator : génère deux clefs RSA, une publique et une privée. Je peux faire des imports dans le launch Instance fenêtre de Infomaniak

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

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.



Import Key Pair

Key Pairs are how you login to your instance after it is launched. Choose a key pair name you will recognize and paste your SSH public key into the space provided.

Key Pair Name *

Key Type *

Load Public Key from a file

Choose File No file chosen

Public Key * Content size: 0 bytes of 16.00 KB

Cancel Import Key Pair

2.2. Configuration SSH (VSCode, GitBash)

Instanciation Réussie :



Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
test3Publickey	Kaas Ubuntu 2404 Kube V1.30	84.234.28.241, 2001:1600:16:10:7:c1	a4-ram8-disk20-perf1	Public-key	Active	eu1-az-2	Aucun	En fonctionnement	3 minutes	Créer un instantané
tercium-instance	Kaas Ubuntu 2404 Kube V1.30	84.234.28.158, 2001:1600:16:10:9:86	a4-ram8-disk20-perf1	benoit-ssh-2025	Active	eu1-az-1	Aucun	En fonctionnement	1 heure, 4 minutes	Créer un instantané

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

```
test@KUS-F-STAGE MINGW64 ~
$ # Créez le dossier .ssh
mkdir -p ~/.ssh

# Copiez votre clé
cp "Private_key_01_07_2025_openssh" ~/.ssh/id_rsa

# Essayez chmod (fonctionne souvent mieux dans ~/.ssh)
chmod 600 ~/.ssh/id_rsa

# Vérifiez
ls -la ~/.ssh/id_rsa
cp: cannot stat 'Private_key_01_07_2025_openssh': No such file or directory
chmod: cannot access '/c/Users/test/.ssh/id_rsa': No such file or directory
ls: cannot access '/c/Users/test/.ssh/id_rsa': No such file or directory

test@KUS-F-STAGE MINGW64 ~
$ # Retournez dans votre dossier de travail
cd "/c/Users/test/Documents/Tercium Stage"

# Vérifiez que le fichier est bien là
ls -la Private_key_01_07_2025_openssh

# Maintenant copiez la clé vers ~/.ssh
cp "Private_key_01_07_2025_openssh" ~/.ssh/id_rsa

# Appliquez chmod
chmod 600 ~/.ssh/id_rsa

# Vérifiez
ls -la ~/.ssh/id_rsa
-rw-r--r-- 1 test 197609 1679 Jul 1 09:15 Private_key_01_07_2025_openssh
-rw-r--r-- 1 test 197609 1679 Jul 1 10:46 /c/Users/test/.ssh/id_rsa
```

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.

- 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.

**HostName 37.156.45.22**

IdentityFile ~/Documents/Tercium\ Stage/tercium-instance_key/tercium-instance_key

Connexion standard : `ssh -i ~/.ssh/tercium_key ubuntu@84.234.28.241`



cd : direction instance

```
ssh -i "~/Documents/Tercium Stage/test3_instance_key/test3publickey_refait_key" ubuntu@84.234.28.241
```

```
test@KUS-F-STAGE MINGW64 ~/Documents/Tercium Stage/test3_instance_key
$ ssh -i "~/Documents/Tercium Stage/test3_instance_key/test3publickey_refait_key" ubuntu@84.234.28.227
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-60-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Jul 1 11:27:44 AM UTC 2025

System load:      0.12
Usage of /:       20.1% of 19.52GB
Memory usage:     2%
Swap usage:       0%
Processes:        166
Users logged in:  0
IPV4 address for enp3s0: 84.234.28.227
IPV6 address for enp3s0: 2001:1600:16:10::4bd

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.

ubuntu@test3-instance:~$
```

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

Infomaniak.cloud : interface de contrôle des instances

Instances

Affichage de 2 éléments

ID de l'instance =

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone
<input type="checkbox"/>	test3Publickey	Kaas Ubuntu 2404 Kube V1.30	84.234.28.241, 2001:1600:16:10::7c1	a4-ram8-disk20-perf1	Public-key	Active	az-2
<input type="checkbox"/>	tercium-instance	Kaas Ubuntu 2404 Kube V1.30	84.234.28.158, 2001:1600:16:10::986	a4-ram8-disk20-perf1	benoit-ssh-2025	Active	az-1

Affichage de 2 éléments

2.4 Usage & Compréhension : Fingerprint

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 une 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

3. Tests fonctionnels de l'instance

3.1. Ping & accès HTTP

Test de connexion simple : Ping avec le terminal sans connexion direct :

```
test@KUS-F-STAGE MINGW64 ~
$ ping 84.234.28.98

Envoi d'une requête 'Ping' 84.234.28.98 avec 32 octets de données :
Réponse de 84.234.28.98 : octets=32 temps=18 ms TTL=49
Réponse de 84.234.28.98 : octets=32 temps=16 ms TTL=49
Réponse de 84.234.28.98 : octets=32 temps=15 ms TTL=49
Réponse de 84.234.28.98 : octets=32 temps=17 ms TTL=49

Statistiques Ping pour 84.234.28.98:
  Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
  Minimum = 15ms, Maximum = 18ms, Moyenne = 16ms
```

2nd Test simple : Créez un fichier index.html basique dans /var/www/html/ et tentez d'y accéder via 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`
si l'on oublie l'adresse IP : **Résultat**

```
test@KUS-F-STAGE MINGW64 ~
$ ssh -i /c/Users/test/Documents/Tercium_Stage/Tercium-instance_key
usage: ssh [-46AaCfGgKkMnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
          [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
          [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
          [-J destination] [-L address] [-l login_name] [-m mac_spec]
          [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
          [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
          destination [command [argument ...]]
ssh [-Q query_option]
```

4. Certification SSL avec Certbot (HTTPS) :

4.1. Installation locale : conseils

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 documenter dans le **manuel d'usage** :

✓ Résumé des recommandations (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 / 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/...

4.2. Synthèse du contrat Let's Encrypt (LE-SA v1.5)

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`.

4.3. Obligations techniques et juridiques

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 certbot python3-certbot-apache y`

5. Permissions, index.html, test Apache : **Proposition bonus**

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>

5.1 Alias Bash et automatisation de connexion

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`

5.2. Déploiement Apache local via WSL2

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 :

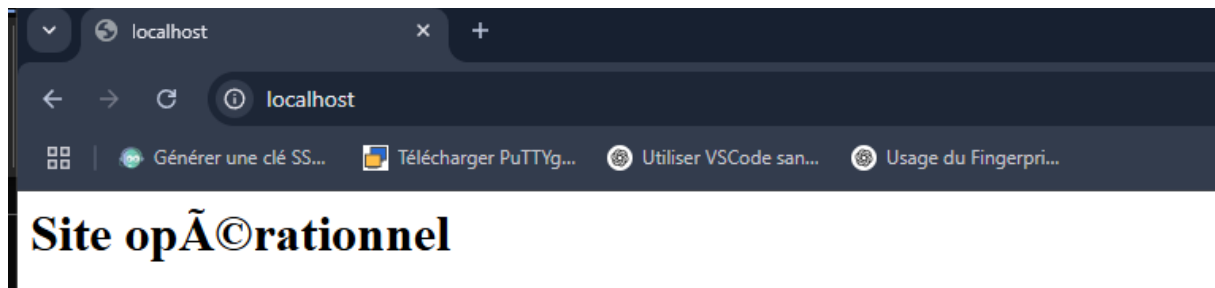
- Ecrire depuis Windows mais exécutes sur Linux réel.
- Faire `sudo`, `apt`, `systemctl`, etc.

- Ouvrir des projets en un clic avec code . dans WSL

5.3. Cas test localhost et localhost:80

Test en local : <http://localhost> ou plus précisément <http://localhost:80>

```
test@KUS-F-STAGE:/mnt/c/Users/test$ sudo apt update
sudo apt install apache2 -y
echo "<h1>Site opérationnel</h1>" | sudo tee /var/www/html/index.html
sudo systemctl restart apache2
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
0% [Waiting for headers]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
120 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.6).
0 upgraded, 0 newly installed, 0 to remove and 120 not upgraded.
<h1>Site opérationnel</h1>
test@KUS-F-STAGE:/mnt/c/Users/test$
```



Si en cloud : http://<ip_publique_vm> déjà réaliser.