



# **S5.B.01 Phase 4**

## **Configurations Réseaux - Processus**

Maxence Lagourgue

8 janvier 2026

# Table des matières

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>VMs</b>	<b>3</b>
II.A	Configurations Interfaces Réseaux	3
II.A.1	Création d'un utilisateur qui aura la propriété des fichiers	3
II.A.2	Définition d'une clé SSH	3
II.A.3	Liaison de la clé SSH au compte sisyphé	3

# I Introduction

Ce document a pour vocation d'exposer tous les artéfacts créés / utilisés pour optimiser les processus de configurations dans le cadre de cette SAE.

## II VMs

Dans cette section, nous évoquerons les outils utilisés pour faciliter la définition des configurations des interfaces réseaux notamment, ainsi que le déploiement du Site Web et des Clusters Kubernetes (RKE2 / k3s).

### II.A Configurations Interfaces Réseaux

Pour cette partie, seront détaillés les différentes étapes afin de récupérer et d'envoyer les fichiers \*.network pour les interfaces réseaux au travers de ssh via sa sous-commande scp.

#### II.A.1 Création d'un utilisateur qui aura la propriété des fichiers

Pour plus de sécurité, nous n'utiliserons pas l'utilisateur root avec une clé ssh mais un utilisateur spécial qui aura le droit de lecture seulement sur /etc/systemd/network, et sur tous les dossiers nécessitant un accès spécial dont nous voulons récupérer et définir la configuration.

Cet utilisateur se nommera : sisyphé.

On réalise cette manipulation avec :

```
useradd -m sisyphé
passwd sisyphé
chown sisyphé:sisyphé -R /etc/systemd/network/
```

Si comme moi, vous avez oublié l'option **-m** pour créer le répertoire utilisateur dans /home, vous pouvez exécuter la commande suivante.

```
install -d -m 0700 -o sisyphé -g sisyphé /home/sisyphé
```

#### II.A.2 Définition d'une clé SSH

Maintenant, donc devons donc générer une clé SSH pour pouvoir par la suite activer la connexion sans mot de passe.

```
ssh-keygen -f sisyphé_key
```

#### II.A.3 Liaison de la clé SSH au compte sisyphé

```
ssh-copy-id -i sisyphé_key.pub sisyphé@10.0.1.3
```

## II.A.4 Fichier exécutable pour automatiser le processus

Maintenant, nous allons créer deux fichiers, l'un pour récupérer les fichiers distants, l'un pour les envoyer.

Listing II.1 – get-config

```
#!/bin/bash

scp -v -i ./sisyphe_key sisyphe@10.0.1.2:/etc/systemd/network/* Reseau/
scp -i ./sisyphe_key sisyphe@10.0.1.3:/etc/systemd/network/* Applicatif/
scp -i ./sisyphe_key sisyphe@10.0.1.4:/etc/systemd/network/* k8sa2/
scp -i ./sisyphe_key sisyphe@10.0.1.5:/etc/systemd/network/* k8sb2/
scp -i ./sisyphe_key sisyphe@10.0.1.6:/etc/systemd/network/* k8sc2/
scp -i ./sisyphe_key sisyphe@10.0.1.10:/etc/systemd/network/* Backup/
```

Listing II.2 – push-config

```
#!/bin/bash

scp -i sisyphe_key Applicatif/* sisyphe@10.0.1.2:/etc/systemd/network/
scp -i sisyphe_key Reseau/* sisyphe@10.0.1.3:/etc/systemd/network/
scp -i sisyphe_key k8sa2/* sisyphe@10.0.1.4:/etc/systemd/network/
scp -i sisyphe_key k8sb2/* sisyphe@10.0.1.5:/etc/systemd/network/
scp -i sisyphe_key k8sc2/* sisyphe@10.0.1.6:/etc/systemd/network/
scp -i sisyphe_key Backup/* sisyphe@10.0.1.10:/etc/systemd/network/
```

Il faut définir ces fichiers exécutables avec la commande suivante :

```
chmod +x ./get-config ./push-config
```

## II.B Configuration du fichier /etc/hosts

Pour cette configuration on va utiliser le même principe que précédemment sauf que le fichier est le même pour toutes les VMs donc copiera le même fichier à chaque fois.

Il sera d'ailleurs seulement utile de le push.