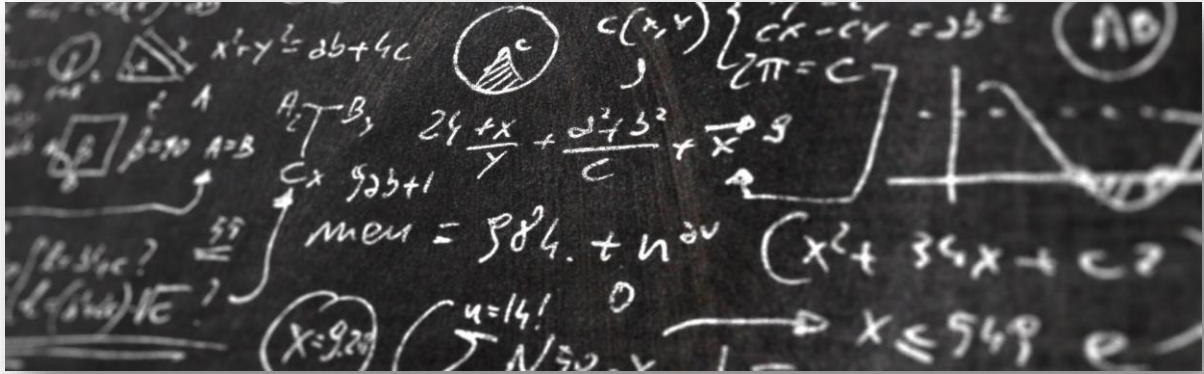


2024
2025

Chiffrement - Déchiffrement

RSX112 – SECURITE DES RESEAUX
STEPHANE LARCHER



Atelier PKI

Sécurité en Mode Survie

Document de Déploiement - Groupe ____	3
 Informations de Connexion	3
Serveur Proxmox OVH	3
Identifiants de votre groupe	3
 Phase 0 : Connexion et Préparation.....	3
Étape 0.1 : Accès à l'Interface Proxmox	3
Étape 0.2 : Vue d'Ensemble de Votre Environnement.....	4
 Phase 1 : Création des Conteneurs LXC	5
Étape 1.1 : Téléchargement du Template Alpine Linux.....	5
Étape 1.2 : Création du Premier Conteneur (PKI-CA)	5
Étape 1.3 : Création du Deuxième Conteneur (Web-Server).....	6
Étape 1.4 : Création du Troisième Conteneur (Client-Admin).....	6
 Phase 2 : Configuration Initiale des Conteneurs	8
Étape 2.1 : Démarrage des Conteneurs	8
Étape 2.2 : Première Connexion via Console.....	8
Étape 2.3 : Configuration de Base.....	8
Étape 2.4 : Test de Connectivité Inter-Conteneurs.....	9
 Phase 3 : Configuration Réseau Avancée	10

Étape 3.1 : Configuration du Pare-feu (Si activé)	10
Étape 3.2 : Configuration des Hostnames.....	10
Étape 3.3 : Optimisation Mémoire.....	10
 Phase 4 : Préparation de l'Environnement PKI	11
Étape 4.1 : Structure des Répertoires	11
Étape 4.2 : Scripts de Connexion Rapide.....	12
 Phase 5 : Monitoring depuis Proxmox.....	13
Visualisation des Ressources.....	13
Création d'un Script de Monitoring Global.....	13
 Phase 6 : Sauvegarde et Snapshots	14
Création de Snapshots	14
Script de Backup Local	14
 Troubleshooting Spécifique Proxmox	15
Problème : "Permission denied" dans Proxmox.....	15
Problème : Conteneur ne démarre pas	15
Problème : Réseau non fonctionnel.....	15
Problème : Performances dégradées	16
 Checklist de Validation d'Infrastructure.....	17
Pour chaque conteneur :	17
Tests de connectivité :	17
Dans Proxmox :	17
 Prochaines Étapes.....	18
 Support	18
Problèmes Proxmox.....	18
Commandes Utiles Proxmox.....	18

Guide d'Installation Infrastructure PKI sur Proxmox OVH

Document de Déploiement - Groupe ____

Informations de Connexion

Serveur Proxmox OVH

- **URL :** <https://ns3061298.ip-162-19-107.eu:8006/>

Identifiants de votre groupe

À compléter avec les informations fournies :

- Utilisateur Proxmox : groupe____
- Mot de passe : _____
- Numéro de groupe : ____
- VLAN attribué : 11____
- Plage d'adresses : 10.1____.0.0/24
- IDs de conteneurs : 1____1, 1____2, 1____3

Phase 0 : Connexion et Préparation

Étape 0.1 : Accès à l'Interface Proxmox

1. **Ouvrir le navigateur** et aller à : <https://ns3061298.ip-162-19-107.eu:8006/>
2. **Accepter l'avertissement de sécurité :**
 - a. Firefox : "Avancé" → "Accepter le risque et poursuivre"
 - b. Chrome : "Avancé" → "Continuer vers le site"
3. **Se connecter** avec vos identifiants :

Username: groupe[X]

Password: [fourni par l'enseignant]

Realm: Proxmox VE authentication server

Language: French (ou English)

✓ **Checkpoint** : Vous devez voir l'interface Proxmox avec votre nom d'utilisateur en haut à droite

Étape 0.2 : Vue d'Ensemble de Votre Environnement

Dans le panneau de gauche, vous devriez voir :

Datacenter

└── ns3061298 (node)

└── Votre pool de ressources (si configuré)

Vérification des permissions :

- Cliquez sur "Datacenter" → "Permissions"
- Vérifiez que votre groupe a les droits sur les ressources assignées

Phase 1 : Création des Conteneurs LXC

Étape 1.1 : Téléchargement du Template Alpine Linux

Si le template n'est pas déjà disponible :

1. Cliquez sur le nœud ns3061298 dans l'arborescence
2. Aller dans local (ns3061298) → CT Templates
3. Cliquer sur Templates en haut
4. Chercher "Alpine" dans la liste
5. Sélectionner alpine-3.19-default_*_amd64.tar.xz
6. Cliquer Download

✓ **Checkpoint** : Le template apparaît dans la liste des templates

Étape 1.2 : Création du Premier Conteneur (PKI-CA)

1. **Clic droit** sur le nœud ns3061298 → Create CT
2. **Onglet General** :

Node: ns3061298

CT ID: 1[XX]1 (remplacer XX par votre numéro de groupe)

Hostname: G[X]-PKI-CA

Unprivileged container: ✓ (coché)

Password: pkilab2025

Confirm password: pkilab2025

3. **Onglet Template** :

Storage: local

Template: alpine-3.19-default_*_amd64.tar.xz

4. **Onglet Root Disk** :

Storage: local-lvm

Disk size (GiB): 4

5. **Onglet CPU** :

Cores: 1

6. Onglet Memory :

Memory (MiB): 256

Swap (MiB): 0

7. Onglet Network :

Name: eth0

Bridge: vmbr0

VLAN Tag: 11[X] (votre VLAN)

IPv4: Static

IPv4/CIDR: 10.1[X].0.2/24

Gateway (IPv4): 10.1[X].0.1

8. Onglet DNS :

DNS domain: groupe[X].local

DNS servers: 8.8.8.8

9. Confirmer → Finish

✓ **Checkpoint** : Le conteneur apparaît dans la liste avec l'ID 1[XX]1

Étape 1.3 : Création du Deuxième Conteneur (Web-Server)

Répéter le processus avec ces modifications :

CT ID: 1[XX]2

Hostname: G[X]-Web-Server

Memory: 384 MiB

IPv4/CIDR: 10.1[X].0.3/24

Disk size: 5 GiB

Étape 1.4 : Création du Troisième Conteneur (Client-Admin)

Répéter le processus avec ces modifications :

CT ID: 1[XX]3

Hostname: G[X]-Client-Admin

Memory: 384 MiB

IPv4/CIDR: 10.1[X].0.4/24

Disk size: 6 GiB

Phase 2 : Configuration Initiale des Conteneurs

Étape 2.1 : Démarrage des Conteneurs

Pour chaque conteneur :

1. Sélectionner le conteneur dans la liste
2. Cliquer sur Start dans la barre d'outils
3. Attendre que le statut passe à "running" (icône verte)

Ordre de démarrage recommandé :

1. D'abord : Client-Admin (1[XX]3)
2. Ensuite : PKI-CA (1[XX]1)
3. Enfin : Web-Server (1[XX]2)

Étape 2.2 : Première Connexion via Console

1. Sélectionner le conteneur G[X]-Client-Admin
2. Cliquer sur Console dans le menu
3. Se connecter : login: rootpassword: pkilab2024

Étape 2.3 : Configuration de Base

Sur CHAQUE conteneur, exécuter via la console :

Mise à jour du système

apk update

apk upgrade

Installation des paquets essentiels

apk add --no-cache \

openssh \

openssl \

sudo \

bash \

vim nano \

curl wget \

net-tools \

nginx \

git

```
# Création de l'utilisateur pkilab
adduser -D -s /bin/bash pkilab
echo "pkilab:pkilab2024" | chpasswd

# Configuration sudo
echo "pkilab ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers

# Configuration SSH
sed -i 's/^#PermitRootLogin.*/PermitRootLogin no/' /etc/ssh/sshd_config
sed -i 's/^#PasswordAuthentication.*/PasswordAuthentication yes/'
/etc/ssh/sshd_config

# Démarrage SSH
rc-service sshd start
rc-update add sshd

# Vérification réseau
ip addr show
ping -c 2 8.8.8.8
```

✓ **Checkpoint** : SSH doit être actif et ping doit fonctionner

Étape 2.4 : Test de Connectivité Inter-Conteneurs

Depuis le conteneur Client-Admin :

```
# Test de connexion aux autres conteneurs
ping -c 2 10.1[X].0.2 # PKI-CA
ping -c 2 10.1[X].0.3 # Web-Server

# Test SSH
ssh pkilab@10.1\[X\].0.2
# (accepter la fingerprint, tester le mot de passe, puis exit)
```

Phase 3 : Configuration Réseau Avancée

Étape 3.1 : Configuration du Pare-feu (Si activé)

Sur chaque conteneur :

```
# Vérifier si iptables est installé
which iptables
```

```
# Si installé, autoriser les connexions nécessaires
iptables -A INPUT -p tcp --dport 22 -j ACCEPT # SSH
iptables -A INPUT -p tcp --dport 443 -j ACCEPT # HTTPS
iptables -A INPUT -p tcp --dport 80 -j ACCEPT # HTTP
iptables -A INPUT -p icmp -j ACCEPT           # Ping
iptables -A INPUT -i lo -j ACCEPT             # Loopback
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Étape 3.2 : Configuration des Hostnames

Sur chaque conteneur, éditer /etc/hosts :

```
# Éditer le fichier hosts
cat >> /etc/hosts << EOF

# PKI Lab Groupe [X]
10.1[X].0.2  pki-ca.groupe[X].local pki-ca
10.1[X].0.3  web.groupe[X].local web-server
10.1[X].0.4  admin.groupe[X].local client-admin
EOF
```

Étape 3.3 : Optimisation Mémoire

```
# Désactiver les services inutiles
rc-update del acpid
rc-update del cron

# Configurer les limites système
cat >> /etc/security/limits.conf << EOF
* soft nofile 1024
* hard nofile 2048
```

EOF

```
# Réduire la swappiness (si swap configuré)
echo "vm.swappiness=10" >> /etc/sysctl.conf
sysctl -p
```

Phase 4 : Préparation de l'Environnement PKI

Étape 4.1 : Structure des Répertoires

Sur le conteneur Client-Admin (1[XX]3) :

```
# Se connecter en tant que pkilab
su - pkilab
```

```
# Créer la structure complète
mkdir -p ~/pki-lab/{scripts,certs,docs,backup,config}
cd ~/pki-lab
```

```
# Créer un README pour le groupe
cat > README.md << EOF
# Infrastructure PKI - Groupe [X]
```

```
## Informations Réseau
- VLAN: 11[X]
- Subnet: 10.1[X].0.0/24
- PKI-CA: 10.1[X].0.2
- Web-Server: 10.1[X].0.3
- Client-Admin: 10.1[X].0.4
```

```
## Membres du Groupe
- PKI Admin: _____
- Web Admin: _____
- Security Officer: _____
```

```
## Journal de Bord
$(date) - Infrastructure déployée sur Proxmox OVH
EOF
```

```
# Initialiser Git
git init
git add README.md
git commit -m "Initial setup Groupe [X]"
```

Étape 4.2 : Scripts de Connexion Rapide

```
# Créer des alias pour faciliter les connexions
cat > ~/.ssh/config << EOF
Host pki-ca
  HostName 10.1[X].0.2
  User pkilab
  Port 22

Host web-server
  HostName 10.1[X].0.3
  User pkilab
  Port 22

Host client-admin
  HostName 10.1[X].0.4
  User pkilab
  Port 22
EOF

chmod 600 ~/.ssh/config

# Test des alias
ssh pki-ca "hostname"
ssh web-server "hostname"
```

Phase 5 : Monitoring depuis Proxmox

Visualisation des Ressources

1. Dans Proxmox, sélectionner un conteneur
2. Aller dans l'onglet Summary pour voir :
 - a. CPU usage
 - b. Memory usage
 - c. Network traffic
 - d. Disk I/O

Création d'un Script de Monitoring Global

Sur votre poste local, créer un script pour surveiller tous vos conteneurs :

```
#!/bin/bash
# monitor-proxmox-groupe.sh

PROXMOX_HOST="ns3061298.ip-162-19-107.eu"
PROXMOX_USER="groupe[X]@pve"
GROUP_NUM="[X]"

echo "=== Monitoring Groupe $GROUP_NUM ==="
echo "Time: $(date)"
echo ""

# Pour chaque conteneur
for i in 1 2 3; do
    CT_ID="1${GROUP_NUM}${i}"
    echo "Container $CT_ID:"

    # Status via API Proxmox (nécessite token API)
    # ou via SSH direct
    ssh pkilab@10.1${GROUP_NUM}.0.$((i+1)) "free -h | grep Mem" 2>/dev/null || echo
    " Unreachable"
    echo ""
done
```

Phase 6 : Sauvegarde et Snapshots

Création de Snapshots

Dans l'interface Proxmox :

1. Sélectionner un conteneur
2. Aller dans Snapshots
3. Cliquer Take Snapshot
4. Nom : initial-setup-\$(date +%Y%m%d)
5. Description : "Configuration initiale Groupe [X]"

Important : Créer un snapshot pour chaque conteneur AVANT de commencer la configuration PKI

Script de Backup Local

Sur le conteneur Client-Admin :

```
#!/bin/bash
# backup-config.sh

BACKUP_DIR=~/.pki-lab/backup/$(date +%Y%m%d-%H%M%S)
mkdir -p $BACKUP_DIR

# Sauvegarde des configs
for host in pki-ca web-server; do
    echo "Backing up $host..."
    ssh $host "sudo tar czf - /etc/nginx /etc/hosts /home/pkilab" >
$BACKUP_DIR/${host}-config.tar.gz
done

echo "Backup completed in $BACKUP_DIR"
ls -la $BACKUP_DIR
```

Troubleshooting Spécifique Proxmox

Problème : "Permission denied" dans Proxmox

Vérifier vos permissions

```
pveum user permissions groupe[X]@pve
```

Si besoin, demander à l'administrateur :

```
pveum acl modify /vms/1[XX][1-3] -user groupe[X]@pve -role PVEVMUser
```

Problème : Conteneur ne démarre pas

1. Vérifier les logs dans Proxmox :
 - a. Sélectionner le conteneur
 - b. Aller dans Task History
 - c. Double-cliquer sur la tâche failed
2. Vérifier l'espace disque :

Sur le nœud Proxmox (si accès SSH)

```
df -h /var/lib/vz
```

Problème : Réseau non fonctionnel

1. Vérifier la configuration VLAN :

Dans le conteneur

```
ip addr show eth0
```

```
ip route show
```

Vérifier la config

```
cat /etc/network/interfaces
```

2. Vérifier le bridge dans Proxmox :
 - a. Node → Network → vmbr0
 - b. Vérifier que "VLAN aware" est activé

Problème : Performances dégradées

Vérifier la charge du node

Dans Proxmox : Node → Summary → CPU/Memory graphs

Optimiser un conteneur

pct set 1[XX]1 --memory 512 # Augmenter temporairement la RAM



Checklist de Validation d'Infrastructure

Avant de passer à la configuration PKI, vérifier :

Pour chaque conteneur :

- ☐ Conteneur créé avec le bon ID (1[XX][1-3])
- ☐ Conteneur démarre sans erreur
- ☐ Réseau configuré (bonne IP, bon VLAN)
- ☐ SSH accessible avec user pkilab
- ☐ Paquets essentiels installés (openssl, nginx)
- ☐ Hostname correctement configuré
- ☐ Fichier /etc/hosts à jour

Tests de connectivité :

- ☐ Ping entre tous les conteneurs OK
- ☐ SSH entre conteneurs fonctionne
- ☐ Accès Internet fonctionnel (ping 8.8.8.8)
- ☐ Résolution DNS fonctionne

Dans Proxmox :

- ☐ Snapshots créés pour chaque conteneur
- ☐ Monitoring visible dans Summary
- ☐ Pas d'erreurs dans Task History

Prochaines Étapes

Une fois l'infrastructure validée, vous pouvez passer à :

1. **Construction de la PKI** (voir guide principal)
2. **Configuration du serveur web HTTPS**
3. **Mise en place des scripts de gestion**
4. **Tests de sécurité et incidents**

Support

Problèmes Proxmox

- Vérifier d'abord la documentation : <https://pve.proxmox.com/wiki/>
- Logs des conteneurs : accessible via l'interface web
- Console d'urgence : toujours disponible même si SSH ne fonctionne pas

Commandes Utiles Proxmox

Via SSH sur le node (si autorisé)

```
pct list          # Liste tous les conteneurs
pct status 1[XX]1 # Status d'un conteneur
pct enter 1[XX]1  # Entrer dans un conteneur
pct stop 1[XX]1   # Arrêter un conteneur
pct start 1[XX]1  # Démarrer un conteneur
pct destroy 1[XX]1 # Supprimer (attention!)
```

Infrastructure prête ? Passez au guide de configuration PKI ! 

Document v2.0 - Infrastructure Proxmox OVH - RSX112 CNAM