

# ?? Architecture Infrastructure as Code

Le document détaille la structure et le fonctionnement du dépôt de configuration Terraform utilisé pour piloter le cluster Proxmox. L'objectif est d'assurer une gestion 100% automatisée, versionnée et sécurisée.

## 1. Structure du Dépôt (Pattern Live/Modules)

Nous utilisons la séparation entre les **Modules** (plans de fabrication) et le **Live** (instances réelles).

```
terraform/
├─ modules/           # --- L'USINE (Templates réutilisables) ---
│   ├─ proxmox_lxc/   # Code générique pour créer un conteneur
│   ├─ proxmox_vm/    # Code générique pour créer une machine virtuelle
│   └─ synology_vm/   # Spécificités pour les VMs sur stockage Synology
└─ live/              # --- LA PRODUCTION (Ce qui tourne vraiment) ---
    ├─ 01-core/       # Fondations : Pools, Réseaux, SDN
    ├─ 02-vms/        # Instances de serveurs (Bastion, Plex, etc.)
    └─ 03-k8s/        # Cluster Kubernetes spécifique
```

## 2. Pourquoi ce découpage ?

- **Modules** : Évite de répéter le code (DRY - Don't Repeat Yourself). Si on améliore la sécurité des LXC, on le fait dans le module et tous les serveurs en profitent.
- **Live (01, 02, 03)** : Réduit le "Blast Radius" (rayon d'explosion). Si une erreur est faite dans le dossier K8s, les fondations (Core) ne sont pas impactées car elles ont leur propre fichier d'état (State).

## 3. Sécurité : Authentification par Token

Ne jamais écrire de mots de passe dans les fichiers `.tf`. On utilise les variables d'environnement.

```
# Export du token API Proxmox (Indispensable avant chaque run)
# Format : 'USER@REALM!TOKENID=UUID'
export PROXMOX_VE_API_TOKEN='terraform@pam!awx-provisioning=c20efe74-ec94-4623-a53d-613ad538fcc9'
```

## 4. Commandes Vitales

Commande	Action
<code>terraform init</code>	Télécharge les connecteurs (Providers) Proxmox. À faire une seule fois par dossier.
<code>terraform plan</code>	<b>L'aperçu</b> : Compare le code avec la réalité. Ne modifie rien.
<code>terraform apply</code>	<b>L'exécution</b> : Applique les changements sur Proxmox. Demande "yes".

## 5. Comment modifier l'infrastructure ?

### A. Ajouter ou renommer un Pool (Dossier Proxmox)

Modifier le fichier `live/01-core/main.tf` :

```
resource "proxmox_virtual_environment_pool" "mon_nouveau_pool" {
  pool_id = "NOM_DU_POOL"
  comment = "Description du contenu"
}
```

## B. Mettre à jour des ressources

Il suffit de changer la valeur (ex: RAM ou CPU) dans le fichier `.tf` et de relancer un `terraform apply`. Terraform ne supprimera pas la machine, il modifiera ses paramètres à chaud si possible.

# 6. Résultats attendus (Proxmox)

Après l'exécution du **01-core**, les "Pools" apparaissent dans Proxmox. Pour les voir, passez la vue Proxmox de "Server View" à "**Folder View**".

- ☐ **INFRASTRUCTURE** : Services vitaux (Réseau, DNS, Proxy)
- ☐ **KUBERNETES** : Cluster K8s
- ☐ **SERVICES** : Plex, Bookstack, etc.
- ☐ **TESTS** : Laboratoire temporaire

--- ### Pourquoi est-ce que c'est "Magique" ? Si demain ton Proxmox brûle ou si tu achètes un deuxième serveur : 1. Tu installes Proxmox. 2. Tu crées le token API. 3. Tu lances Terraform. 4. **\*\*En 10 secondes, toute ton arborescence, tes réseaux et tes pools sont recréés à l'identique.\*\***  
### Et maintenant, la suite ? Maintenant que nous avons les "pièces" (les Pools), nous allons pouvoir passer au dossier `**`02-vms`**`. C'est ici que nous allons créer nos premiers serveurs en utilisant les **\*\*Modules\*\***.

---

Revision #1

Created 2026-04-30 12:58:07 UTC by Nico là

Updated 2026-04-30 13:00:22 UTC by Nico là