Proxmox whit Terraform

End-to-End

Kubernetes Enterprise sur Infrastructure Personnelle

Objectif :

- Environnement Kubernetes automation complète avec GitOps.

Architecture

- Infrastructure : Proxmox VE → Talos Linux → Kubernetes

Automation :

- Terraform (IaC) + ArgoCD

Stack Principal

- OS : Talos (immutable, API-only)
- Réseau : Cilium CNI (eBPF performance)

GitOps :

- ArgoCD + ApplicationSets (scaling pattern)

Supervision

- Hubble Network

Infrastructure Foundation

Infrastructure Foundation
- Architecture Réseau Proxmox Sécurisée
- Terraform Cluster Talos Kubernetes - Déploiement IaC
- Chart Values - Pattern Helm Centralisé
- Configuration Réseau avec Cilium

Sécurité & Configuration
- Gestion Sécurité TLS Complète - Cert-Manager & Trust-Manager
- Reloader - Auto-Restart pour ConfigMaps et Secrets

Automation GitOps
- GitOps Automation & Bootstrap
- Renovate - Automation des Mises à Jour de Dépendances

Git Repositories:
- k8s_talos.git (Infrastructur
- argocd.git (Applications)

Renovate:
- Automation des mises à jour avec détection et PR automatiques

Workflow Infrastructure:
- Terraform → Proxmox VMs
- Talos Linux → Kubernetes Cluster

Composants Core:
- Cilium: Réseau eBPF
- Cert-Manager: TLS automatisé
- Reloader: Hot-reload des configurations

Déploiement GitOps:
- ArgoCD synchronise Git → Cluster
- Applications multi-environnements

Flux Git:
- Commit → Review → Merge → Auto-Deploy

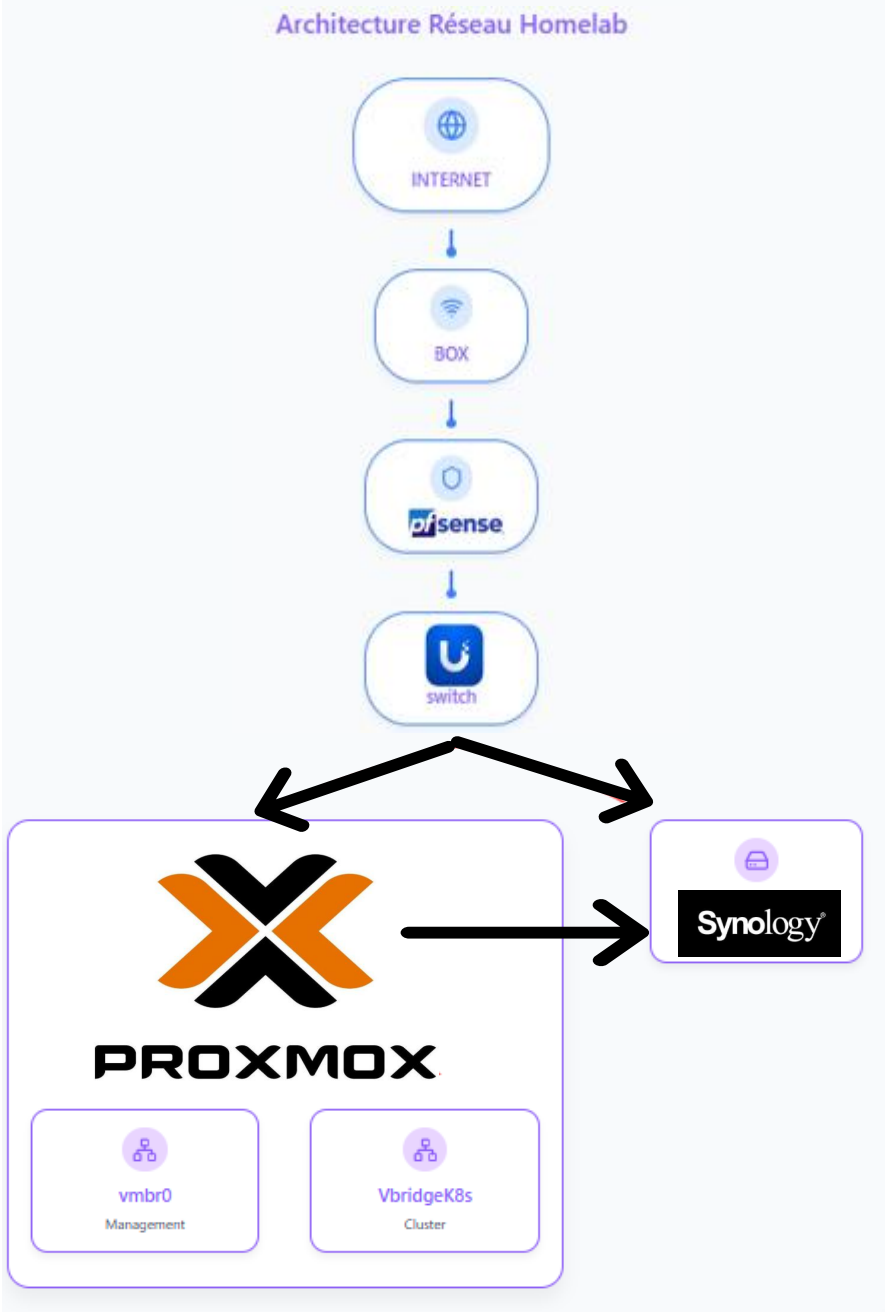
Architecture Réseau Proxmox Sécurisée

Firewall :

- actif avec politique entrante "DROP" (tout trafic bloqué par défaut)

Ports autorisés limités:

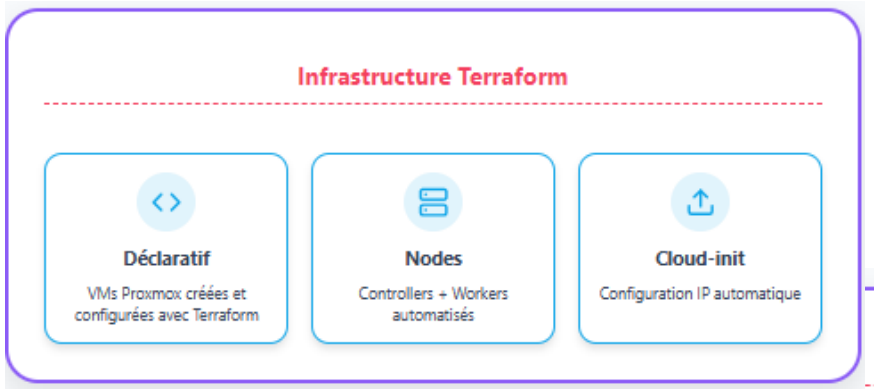
- SSH (xx), Web UI (xx) uniquement depuis réseau xx.xx.xx.0/24
- NAT configuré pour permettre accès Internet depuis cluster (xx.xx.xx.0/24)
- Bridge vmbr1 VLAN-aware: Configuration "bridge-vlan-aware: yes" active
- Communication inter-pods libre à l'intérieur du réseau cluster pour le moment

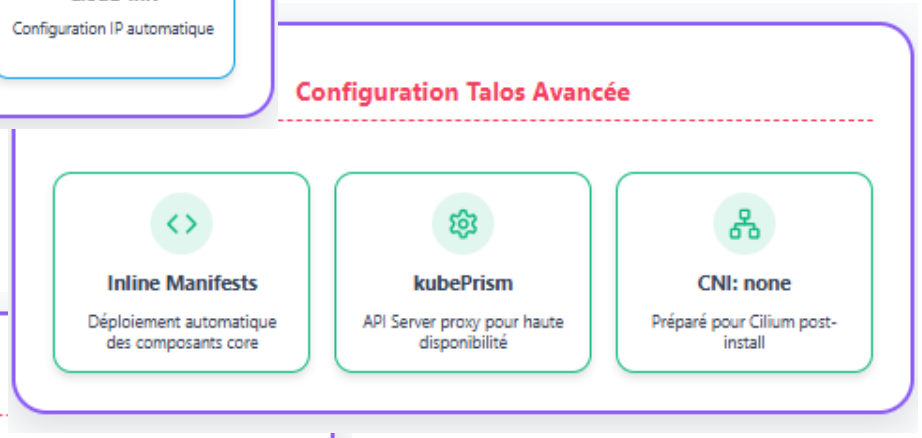


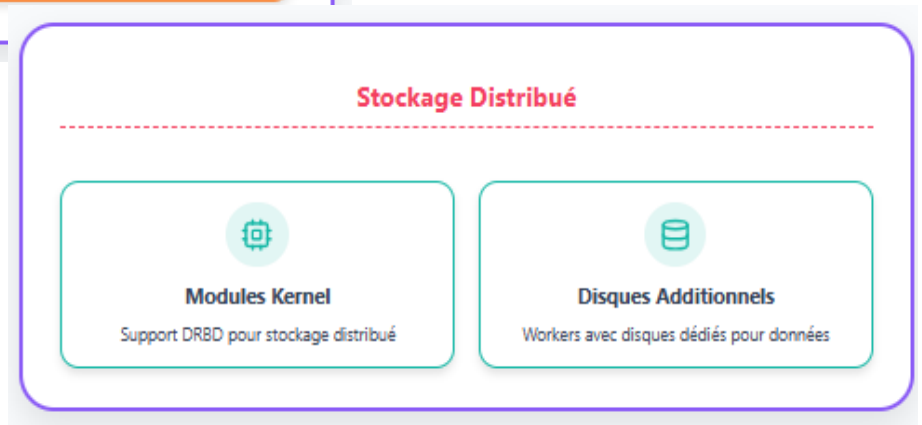
Cluster Talos Kubernetes - Déploiement IaC avec Terraform

Provisioning déclaratif:

- VMs Proxmox créées et configurées avec Terraform
- UEFI + TPM v2.0: Sécurité renforcée et support secure boot
- Cloud-init intégré: Configuration automatique des IPs et réseau
- Talos immutable: Système d'exploitation spécialisé Kubernetes
- Inline manifests: Déploiement automatique des composants core
- kubePrism activé: API Server proxy pour haute disponibilité
- Modules kernel chargés: Support DRBD pour stockage distribué







Pattern HELM Values Centralisés

Provisioning déclaratif:

Structure claire:

- Un fichier values.yaml par composant

Réutilisabilité:

- Configurations modulaires et indépendantes

GitOps compatible:

- Values stockées dans Git pour traçabilité

Bootstrap optimisé:

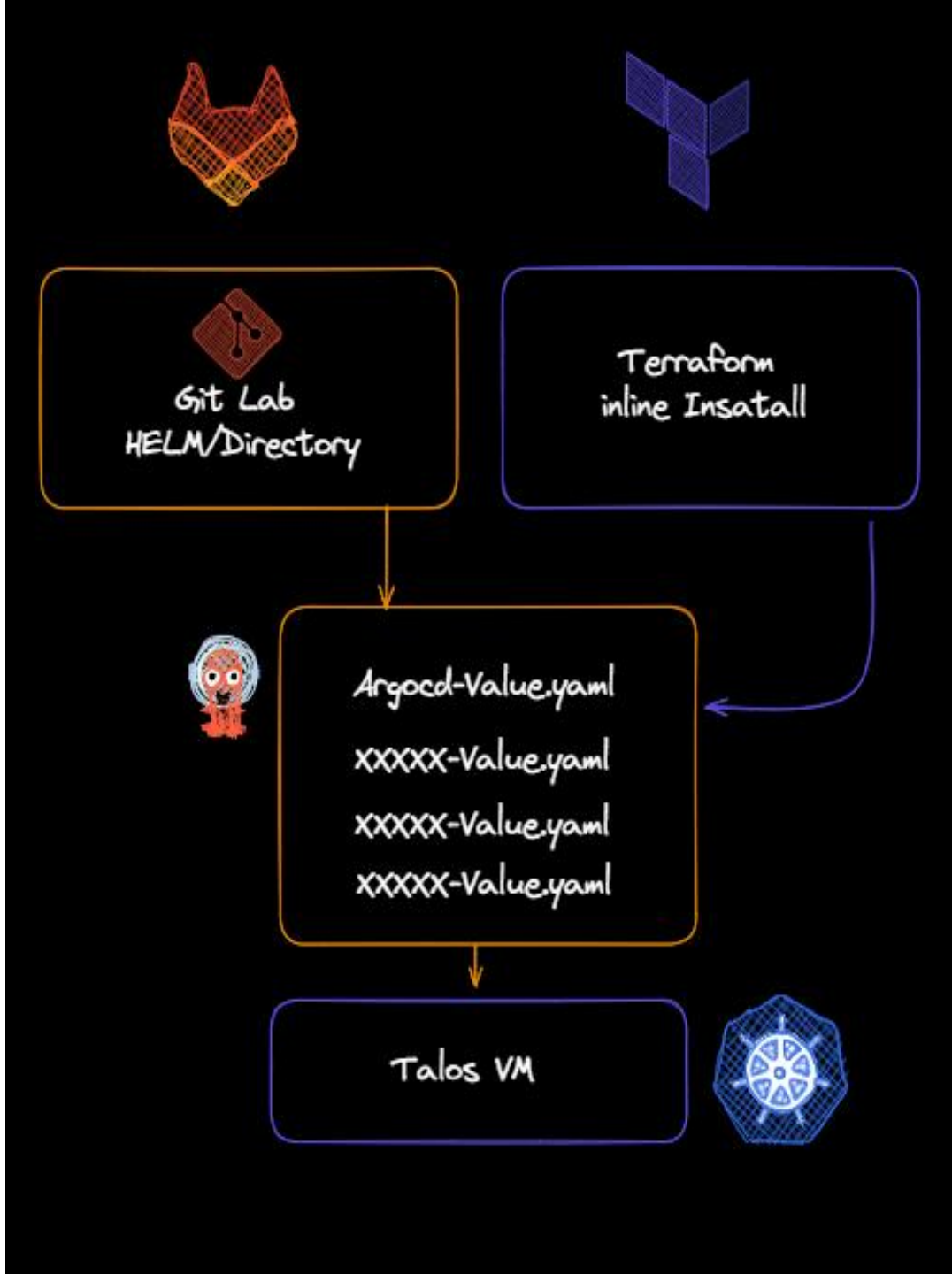
- Installation directe dans la configuration Talos

Inline Manifests:

- Intégrés dans le bootstrapping du cluster

GitLab

argo



Configuration Réseau avec Cilium / Terraform/Hubble

eBPF Dataplane:

- Performance native kernel (pas de virtualisation réseau)
- CNI + Ingress Controller + LoadBalancer dans une seule solution

IP Pool:

- Allocation automatique d'IPs externes (x.x.x.x-100)

L2 Announcements:

- Distribution du trafic sur les nodes workers uniquement

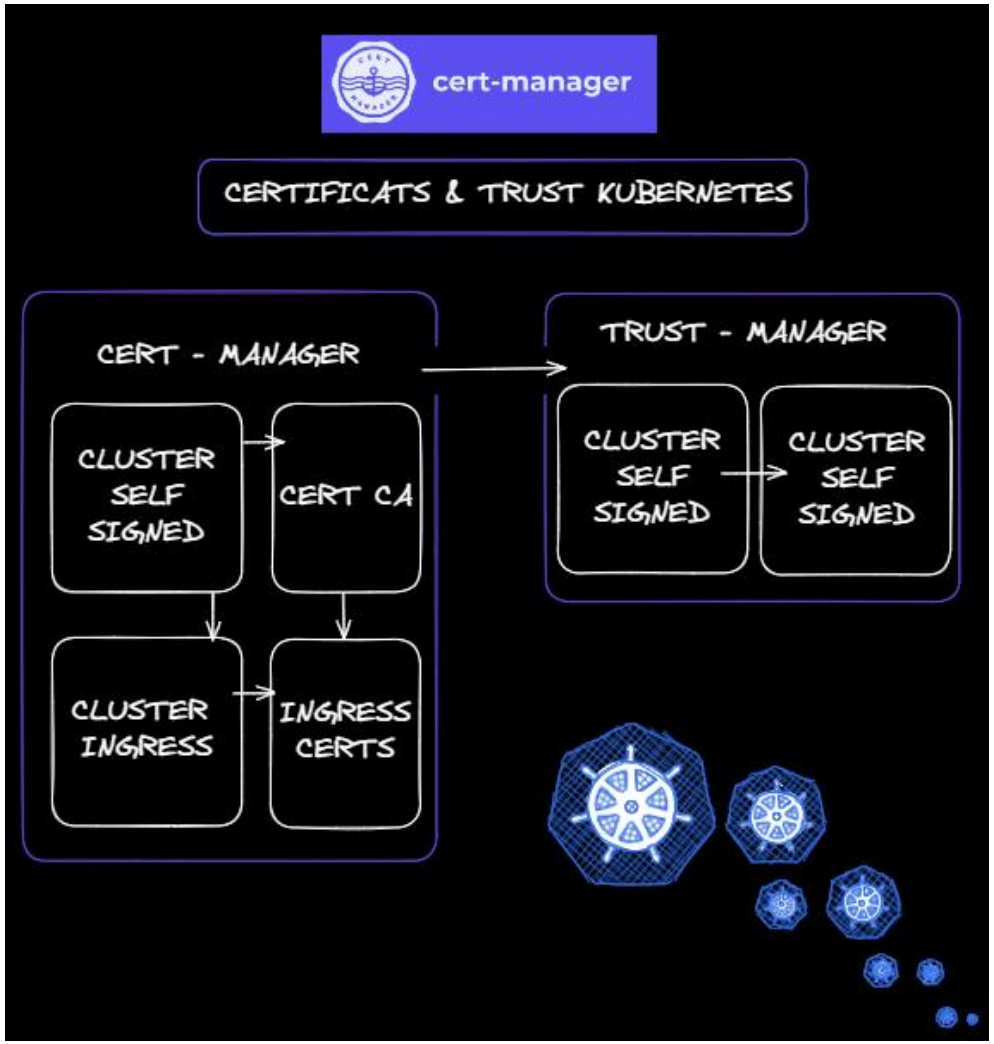
Helm Templating:

- Configuration déclarative complète via Terraform + Helm

Hubble Observability:

- Monitoring réseau eBPF temps réel
- Dashboard Web + métriques Prometheus
- Debugging trafic inter-pods

Gestion Sécurité TLS Complète - Cert-Manager & Trust-Manager/Terraform



PKI complète:

- Chaîne de confiance automatisée (CA racine selfsigned = > CA intermédiaire ingress)

ECDSA P-256:

- Cryptographie moderne plus performante que RSA

Durée optimisée:

- Certificats valides 180 jours (vs 90 jours par défaut)

Distribution Trust:

- Trust-Manager propage les CA dans tout le cluster

Trust Store Linux:

- Configuration automatique sur les nodes et pods

HashiCorp Terraform

Reloader - Auto-Restart pour ConfigMaps et Secrets

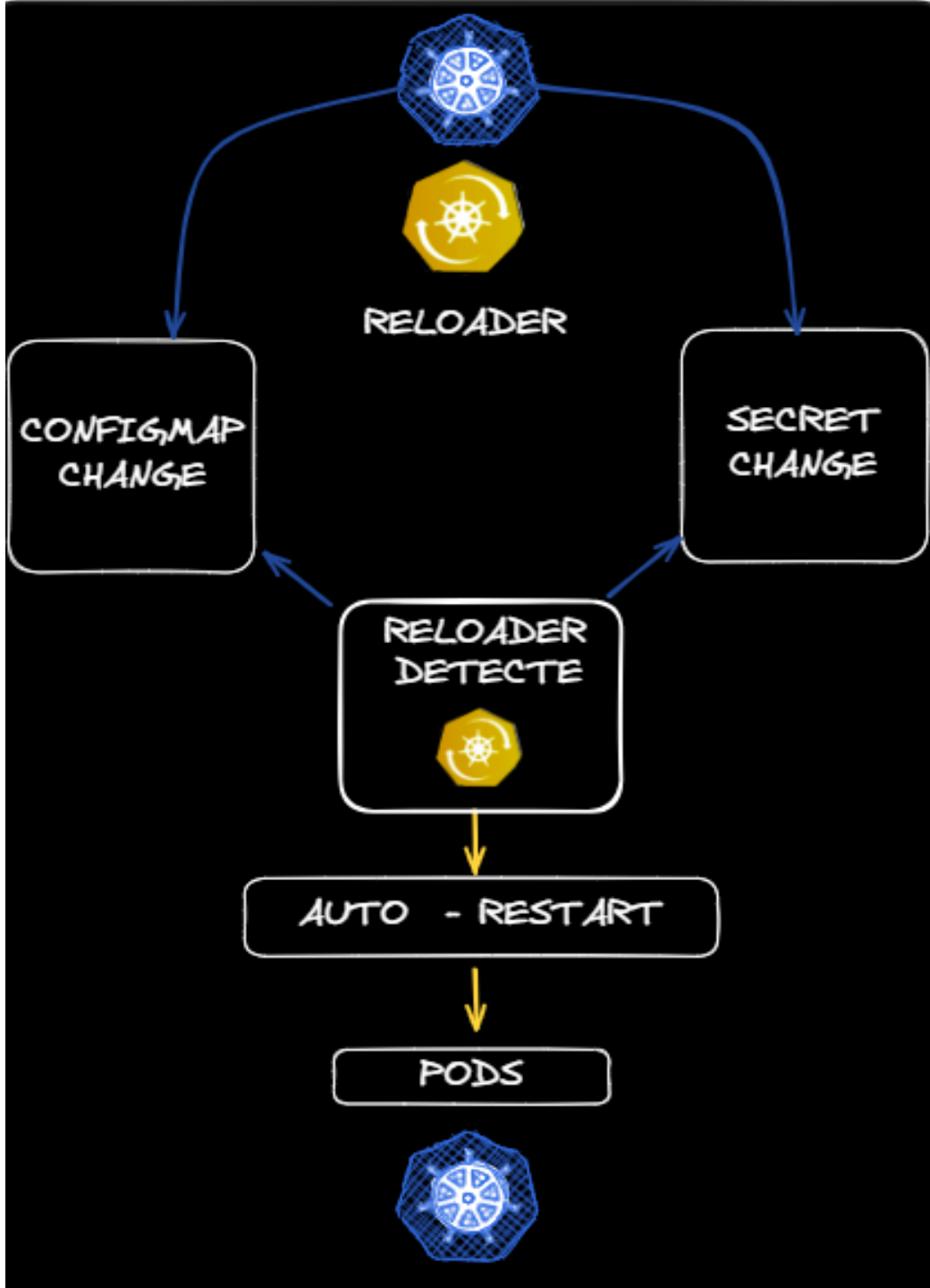
Auto-Reload:
- Détecte les changements de ConfigMaps et Secrets et redémarre les pods

Zero-Config:
- Fonctionne automatiquement sans annotation pour les changements standards

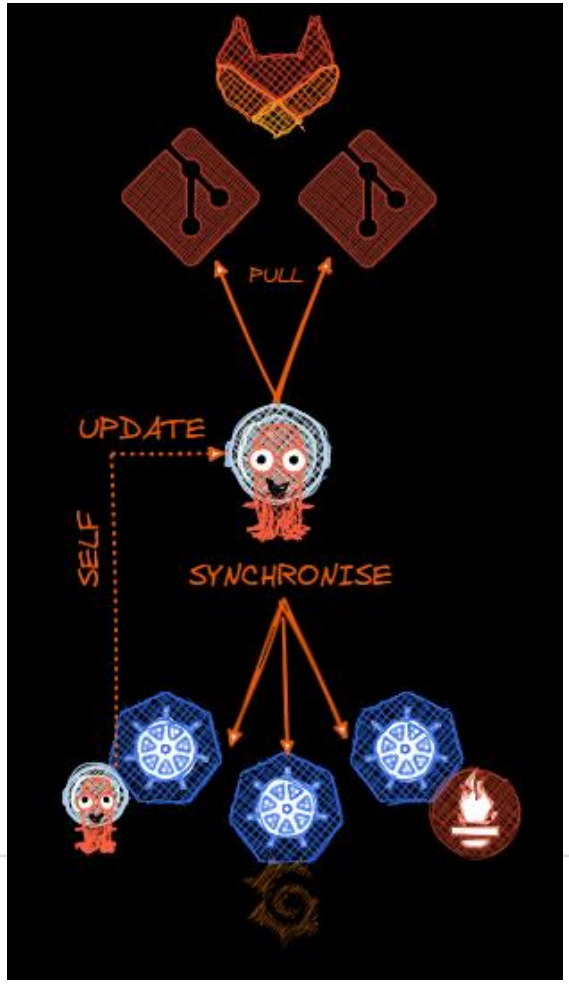
TLS Hot Reload:
- Particulièrement utile pour recharger les certificats TLS mis à jour

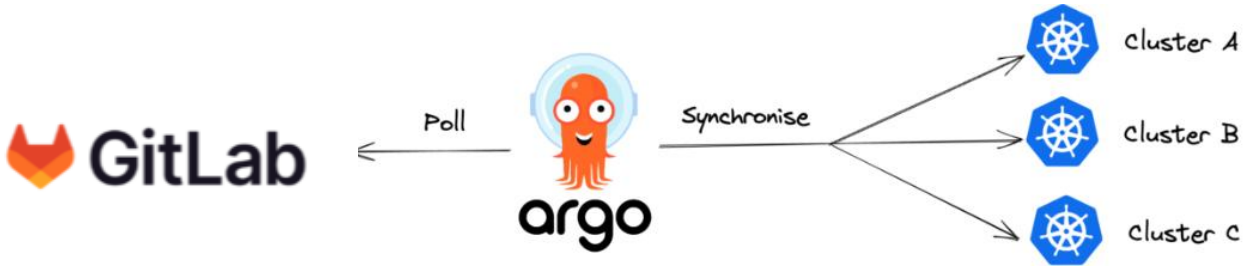
Compatibilité Trust-Manager:
- Assure que les certificats CA sont immédiatement utilisés

Non-intrusif:
- Déploiement simple via Helm sans modification des applications



Self-Bootstrap Pattern





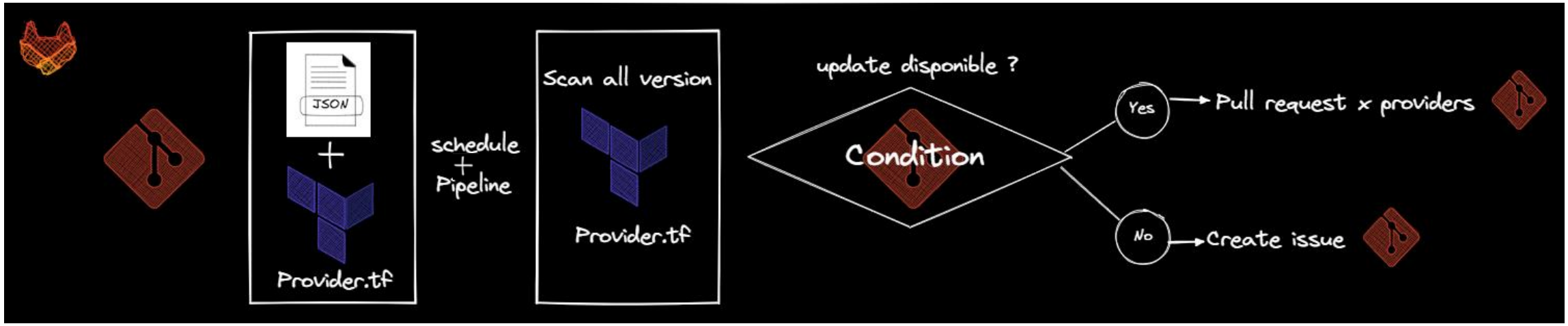
GitOps Automation & Bootstrap

- **Self-Bootstrap Pattern**: ArgoCD déploie sa propre configuration via Application bootstrap
- **TLS Automatique**: Certificat personnalisé généré par cert-manager
- **Terraform + Helm**: Déploiement initial via Terraform puis gestion par
- **Source de Vérité**: Tout changement doit passer par Git (repo bootstr
- **Ingress HTTPS**: Configuration complète avec TLS et domaine dédié

kubernetes

GitLab

Git Flow



Renovate - Mises à Jour automatique des Providers

- **Configuration Déclarative:** renovate.json pour définir comportement et règles
- **Regex Managers:** Détection intelligente des versions via patterns regex
- **Terraform Providers**: Gestion automatique des versions de providers
- **Semantic Commits:** Préfixe "chore(deps):" pour clarté des messages
- **Limites Concurrentes**: Maximum 2 PR simultanées pour éviter surcharge