Documentation Complète

Déploiement, Supervision et Administration de Cluster Via Rancher avec Traefik, MetalLB et mkcert

⚙️ Prérequis et Installation

Une solution complète et professionnelle pour déployer Rancher avec des certificats TLS valides dans un environnement, intégrant MetalLB pour le load balancing et Traefik comme reverse proxy.

Architecture et Principe

Schéma d'Architecture

Principe de Fonctionnement

MetalLB

Fournit les IPs LoadBalancer dans le cluster local

Traefik

Reverse proxy et ingress controller avec termination SSL

mkcert

Génère des certificats TLS valides localement

Rancher

Interface de gestion Kubernetes avec certificats valides

Prérequis et Installation

Prérequis Système

```
# Vérifier l'environnement
kubectl version --short
helm version
ip addr show # Vérifier l'interface WiFi (10.64.13.203/24)
```

Structure du Projet

```
rancher-deployment/
├── metalib/
│   └── metalib-config.yaml
├── namespace/
│   └── namespace.yaml
├── render/
│   ├── rancher-ingress.yaml
│   └── rancher-values.yaml
├── scripts/
│   ├── checkstatus.sh
│   └── debug.sh
├── traefik/
│   └── traefik-values.yaml
├──
│   ├── rancher-l8s.c1-i.pem
│   ├── rancher-l8s.c1-i-key.pem
│   └── rancher-l8s.c1.pem
├── deploy.sh
└── check-start.sh
```

Configuration mkcert

Installation de mkcert

```
# Téléchargement et installation
wget -O mkcert https://github.com/FiloSottile/mkcert/releases/download/v1.4.4/mkcert-v1.4.4-linux-amd64
chmod +x mkcert
sudo mv mkcert /usr/local/bin/

# Installation de l'Autorité de Certification locale
mkcert -install
```

Génération des Certificats

```
# Générer le certificat pour le domaine et l'IP
mkcert rancher.k8s.ci 10.64.13.211 #votre ip correspondante

# Vérifier les fichiers générés
ls -la *.pem
# rancher.k8s.ci+1.pem # Certificat
# rancher.k8s.ci+1-key.pem # Clé privée
```

Création des Secrets Kubernetes

```
# Secret pour Traefik (format TLS)
kubectl create secret tls tls-rancher-ingress \
 -n cattle-system \
 --cert=rancher.k8s.ci+1.pem \
 --key=rancher.k8s.ci+1-key.pem
```

Déploiement Pas à Pas

01

Préparation des Namespaces

Création des namespaces metallb-system, traefik-system et cattle-system avec les labels appropriés

02

Configuration MetalLB

Configuration de l'IPAddressPool (10.64.13.210-10.64.13.220) et du L2Advertisement pour le load balancing

03

Configuration Traefik

Déploiement de Traefik avec redirection HTTP vers HTTPS, activation TLS et service LoadBalancer

04

Configuration Rancher

Installation de Rancher avec montage des certificats mkcert et désactivation de l'ingress natif

05

Configuration Ingress

Création de l'Ingress Kubernetes pour router le trafic HTTPS vers Rancher via Traefik

06

Lancement du Déploiement

Exécution du script deploy.sh pour automatiser l'ensemble du processus

Lancement du Déploiement

```
# Rendre les scripts exécutables
chmod +x scripts/*.sh

# Lancer le déploiement complet
./scripts/deploy.sh
```

Fichiers de Configuration

Configuration MetalLB

metallb/metallb-config.yaml

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: my-ip-pool
  namespace: metallb-system
spec:
  addresses:
  - 10.64.13.210-10.64.13.220
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2-ad
  namespace: metallb-system
spec: {}
```

Configuration Traefik

traefik/traefik-values.yaml

```
deployment:
 replicas: 1
providers:
 kubernetesIngress:
 publishedService:
 enabled: true
ports:
 web:
 redirectTo: websecure # Redirection HTTP → HTTPS automatique
 websecure:
 tls:
 enabled: true # Activation TLS
ingressRoute:
 dashboard:
 enabled: true # Activation dashboard Traefik
service:
 type: LoadBalancer # MetalLB attribuera une IP
```

Configuration Rancher et Ingress

rancher/rancher-values.yaml

```
hostname: rancher.k8s.ci
replicas: 1
# DÉSACTIVATION de l'ingress Rancher - Traefik gère
ingress:
  enabled: false
service:
  type: LoadBalancer
# Montage du certificat mkcert dans Rancher
volumes:
  - name: ssl-cert
    secret:
      secretName: tls-rancher-ingress # Secret créé avec mkcert
volumeMounts:
  - name: ssl-cert
    mountPath: /etc/rancher/ssl
    readOnly: true
# Variables d'environnement pour utiliser le certificat
extraEnv:
  - name: SSL_CERT_DIR
    value: /etc/rancher/ssl
  - name: CATTLE_PROMETHEUS_METRICS
    value: "true"
resources:
  limits:
    cpu: 1000m
    memory: 1536Mi
  requests:
    cpu: 500m
    memory: 1024Mi
```

rancher/rancher-ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: rancher
  namespace: cattle-system
  annotations:
    kubernetes.io/ingress.class: traefik
    traefik.ingress.kubernetes.io/router.entrypoints: websecure
    traefik.ingress.kubernetes.io/router.tls: "true"
spec:
  rules:
  - host: rancher.k8s.ci
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: rancher
            port:
              number: 80
  tls:
  - hosts:
    - rancher.k8s.ci
    secretName: tls-rancher-ingress # Utilise le secret mkcert
```

Modification DNS/hosts

```
# Sur ta machine, modifie /etc/hosts pour pointer vers Traefik
sudo nano /etc/hosts

# Ajoute cette ligne :
10.64.13.210 rancher.k8s.ci

# Maintenant teste la validité du certificat
curl -v https://rancher.k8s.ci
```

🔧

Résolution des Problèmes

1

Certificat non reconnu

```
# Vérifier le certificat présenté
openssl s_client -connect rancher.k8s.ci:443 -servername rancher.k8s.ci < /dev/null | openssl x509 -noout -issuer -subject

# Vérifier le secret
kubectl get secret -n cattle-system tls-rancher-ingress -o yaml

# Regénérer les certificats si nécessaire
mkcert -uninstall
mkcert -install
mkcert rancher.k8s.ci 10.64.13.211
```

2

Services non accessibles

```
# Vérifier les IPs attribuées
kubectl get svc -A

# Vérifier les logs
kubectl logs -n traefik-system deployment/traefik
kubectl logs -n cattle-system -l app=rancher

# Vérifier la résolution DNS
nslookup rancher.k8s.ci
```

3

Ingress non configuré

```
# Vérifier l'ingress
kubectl describe ingress -n cattle-system rancher

# Vérifier les endpoints
kubectl get endpoints -n cattle-system rancher
```

Maintenance et Surveillance

Surveillance des Ressources

```
# Vérifier l'utilisation des ressources
kubectl top pods -A
kubectl top nodes

# Vérifier les événements
kubectl get events -A --sort-by='.lastTimestamp'
```

Sauvegarde des Certificats

```
# Sauvegarder les certificats mkcert
cp rancher.k8s.ci+1.pem ~/backup/
cp rancher.k8s.ci+1-key.pem ~/backup/

# Sauvegarder les secrets
kubectl get secret -n cattle-system tls-rancher-ingress -o yaml > ~/backup/secret-backup.yaml
```

Mise à Jour

```
# Mettre à jour Rancher
helm upgrade rancher rancher-latest/rancher -n cattle-system -f rancher/rancher-values.yaml

# Mettre à jour Traefik
helm upgrade traefik traefik/traefik -n traefik-system -f traefik/traefik-values.yaml
```

Points Clés de la Solution

Avantages de l'Architecture

1. **Certificats Valides** : Plus d'avertissements de sécurité
2. **SSL Termination** : Traefik gère le TLS de manière centralisée
3. **Load Balancing** : MetalLB fournit des IPs stables
4. **Haute Disponibilité** : Architecture scalable

Flux de Trafic

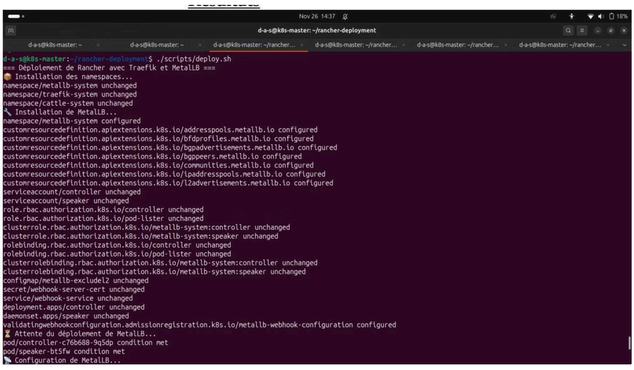
1. **Client** → HTTPS avec certificat mkcert valide
2. **Traefik** → Termination SSL et routage
3. **Rancher** → Service avec certificat monté
4. **Réponse** → Retour via Traefik avec chiffrement

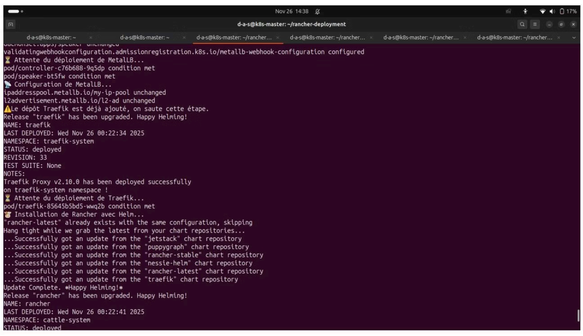
Sécurité

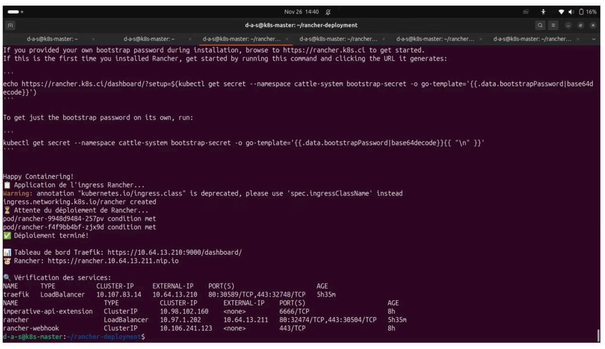
- Certificats TLS valides reconnus par les navigateurs
- Communication chiffrée de bout en bout
- Secrets Kubernetes sécurisés
- Contrôle d'accès via Ingress

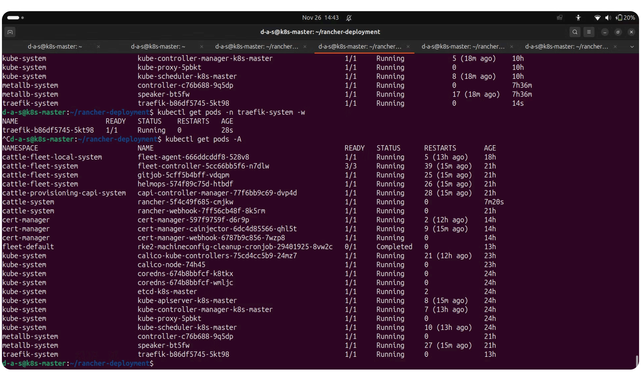
Résultats

Exécution du Script de Déploiement

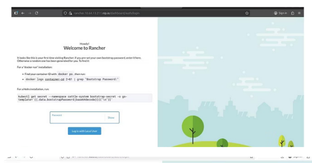


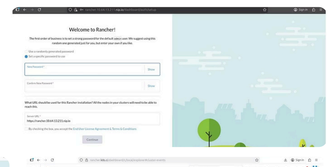


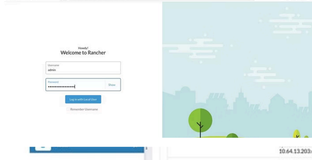


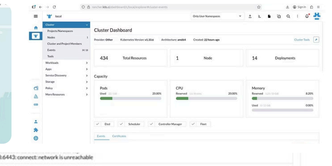


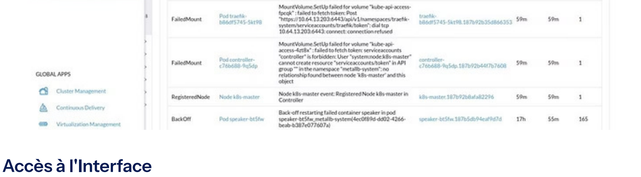
Tous les Pods sont en Running











Accès à l'Interface

- La première fois vous allez obtenir le secret et vous logger
- Ensuite vous décidez soit de générer un mot de passe ou d'en définir un
- On se connecte
- On peut voir les informations sur notre Cluster
- On peut superviser notre cluster s'il y a des erreurs ou tout s'est bien passé, voir l'état du cluster et administrer, gérer le cluster et plein d'autres choses, Amusez-Vous !

> Cette documentation fournit une solution complète et professionnelle pour déployer Rancher avec des certificats TLS valides dans un environnement.