

# Documentation Complète : Déploiement, Supervision et Administration de Cluster Via Rancher avec Traefik, MetalLB et mkcert

## Table des Matières

1. Architecture et Principe
2. Prérequis et Installation
3. Configuration mkcert
4. Déploiement Pas à Pas
5. Fichiers de Configuration
6. Résolution des Problèmes
7. Maintenance et Surveillance
8. Resultats

## Architecture et Principe

### Schéma d'Architecture

text

```
[Utilisateur]
  ↓ HTTPS (rancher.k8s.ci) → DNS → 10.64.13.210
[Traefik LoadBalancer] (10.64.13.210)
  ↓ Présente certificat mkcert valide
[Ingress Kubernetes]
  ↓ Route vers service Rancher
[Rancher Service] (LoadBalancer: 10.64.13.211)
  ↓
[Pods Rancher] (avec certificat mkcert monté)
```

### 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

bash

```
# 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

└── certificates/
    ├── 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

bash

```
# 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

bash

```
# 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

bash

```
# 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

### Étape 1: Préparation des Namespaces

**namespaces/namespaces.yaml**

yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: metallb-system
  labels:
    name: metallb-system
---
apiVersion: v1
kind: Namespace
metadata:
  name: traefik-system
  labels:
    name: traefik-system
---
apiVersion: v1
kind: Namespace
metadata:
  name: cattle-system
  labels:
    name: cattle-system
```

### Étape 2: Configuration MetalLB

**metallb/metallb-config.yaml**

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: {}
```

## Étape 3: Configuration Traefik

### traefik/traefik-values.yaml

```
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 # MetallLB attribuera une IP
```

## Étape 4: Configuration Rancher

### rancher/rancher-values.yaml

```
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
```

## Étape 5: Configuration Ingress

### rancher/rancher-ingress.yaml

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
```

## Étape 6: Lancement du Déploiement

bash

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

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



## Fichiers de Configuration Complets

### Script de Déploiement Principal

#### scripts/deploy.sh

bash

```
#!/bin/bash
set -e
```

```

echo "==== Déploiement de Rancher avec Traefik et MetallLB ==="

# Variables
METALLB_VERSION="v0.13.12"
TRAEFIK_VERSION="23.0.0"
RANCHER_VERSION="2.12.2"

echo "📦 Installation des namespaces..."
kubectl apply -f namespaces/namespaces.yaml

echo "🔧 Installation de MetallLB..."
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/${METALLB_VERSION}/config/manifests/metallb-native.yaml

echo "⏳ Attente du déploiement de MetallLB..."
kubectl -n metallb-system wait --for=condition=ready pod -l app=metallb --timeout=120s

echo "📡 Configuration de MetallLB..."
kubectl apply -f metallb/metallb-config.yaml

# Installation Traefik
if ! helm repo list | grep -q "traefik"; then
    echo "🚀 Installation de Traefik avec Helm..."
    helm repo add traefik https://traefik.github.io/charts
    helm repo update
else
    echo "⚠️ Le dépôt Traefik est déjà ajouté, on saute cette étape."
fi

helm upgrade --install traefik traefik/traefik \
    --namespace traefik-system \
    --version ${TRAEFIK_VERSION} \
    --values traefik/traefik-values.yaml

echo "⏳ Attente du déploiement de Traefik..."
kubectl -n traefik-system wait --for=condition=ready pod -l app.kubernetes.io/name=traefik --timeout=120s

echo "📦 Installation de Rancher avec Helm..."
helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
helm repo update

helm upgrade --install rancher rancher-latest/rancher \
    --namespace cattle-system \
    --version ${RANCHER_VERSION} \
    --values rancher/rancher-values.yaml

echo "📋 Application de l'ingress Rancher..."
kubectl apply -f rancher/rancher-ingress.yaml

echo "⏳ Attente du déploiement de Rancher..."
kubectl -n cattle-system wait --for=condition=ready pod -l app=rancher --timeout=600s

echo "✅ Déploiement terminé!"
echo ""
echo "📊 Tableau de bord Traefik: https://10.64.13.210:9000/dashboard/"
echo "🌐 Rancher: https://rancher.k8s.ci"
echo ""
echo "🔍 Vérification des services:"
kubectl get svc -n traefik-system
kubectl get svc -n cattle-system

```

## Script de Vérification

### scripts/check-status.sh

bash

```
#!/bin/bash
echo "==== Statut du déploiement ==="

echo ""
echo "🔍 Pods:"
kubectl get pods -n metallb-system
echo ""
kubectl get pods -n traefik-system
echo ""
kubectl get pods -n cattle-system

echo ""
echo "🌐 Services:"
kubectl get svc -n traefik-system
kubectl get svc -n cattle-system

echo ""
echo "🟠 Ingress:"
kubectl get ingress -A

echo ""
echo "📡 IPs MetallLB:"
kubectl get ipaddresspool -n metallb-system
```

## Script de Nettoyage

### scripts/cleanup.sh

bash

```
#!/bin/bash
echo "==== Nettoyage du déploiement ==="

helm uninstall rancher -n cattle-system || true
helm uninstall traefik -n traefik-system || true

kubectl delete -f rancher/rancher-ingress.yaml || true
kubectl delete -f metallb/metallb-config.yaml || true

kubectl delete -f
https://raw.githubusercontent.com/metallb/metallb/v0.13.12/config/manifests/
metallb-native.yaml || true

kubectl delete namespace cattle-system traefik-system metallb-system || true
echo "✅ Nettoyage terminé"
```

Modifie ton DNS/hosts

bash

```
# 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

### Problème: Certificat non reconnu

bash

```
# 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
```

### Problème: Services non accessibles

bash

```
# 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
```

### Problème: Ingress non configuré

bash

```
# 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

bash

```
# 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

```
bash  
  
# 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

```
bash  
  
# 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'avertissemens 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

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

## Resultats

```
Nov 26 14:37 ⓘ
d-a-s@k8s-master: ~/rancher-deployment
d-a-s@k8s-master: ~  d-a-s@k8s-master: ~  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...
d-a-s@k8s-master: ~/rancher-deployment$ ./scripts/deploy.sh
--- Déploiement de Rancher avec Traefik et MetalLB ===
📦 Installation des namespaces...
namespace/metallb-system unchanged
namespace/traefik-system unchanged
namespace/cattle-system unchanged
🔧 Installation de MetalLB...
namespace/metallb-system configured
customresourcedefinition.apiextensions.k8s.io/addresspools.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/bfdprofiles.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/bgpadvertisements.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/bgppeers.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/communities.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/ipaddresspools.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/l2advertisements.metallb.io configured
serviceaccount/controller unchanged
serviceaccount/speaker unchanged
role.rbac.authorization.k8s.io/controller unchanged
role.rbac.authorization.k8s.io/pod-lister unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker unchanged
rolebinding.rbac.authorization.k8s.io/controller unchanged
rolebinding.rbac.authorization.k8s.io/pod-lister unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller unchanged
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker unchanged
configmap/metallb-excludel2 unchanged
secret/webhook-server-cert unchanged
service/webhook-service unchanged
deployment.apps/controller unchanged
daemonset.apps/speaker unchanged
validatingwebhookconfiguration.admissionregistration.k8s.io/metallb-webhook-configuration configured
⚠️ Attente du déploiement de MetalLB...
pod/controller-c76b688-9q5dp condition met
pod/speaker-bt5fw condition met
🔧 Configuration de MetalLB...
```

## Execution du Script de Deploiement

```
Nov 26 14:38 ⓘ
d-a-s@k8s-master: ~/rancher-deployment
d-a-s@k8s-master: ~  d-a-s@k8s-master: ~  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...  d-a-s@k8s-master: ~/rancher...
d-a-s@k8s-master: ~/rancher-deployment$ ./scripts/deploy.sh
validatingwebhookconfiguration.admissionregistration.k8s.io/metallb-webhook-configuration configured
⚠️ Attente du déploiement de MetalLB...
pod/controller-c76b688-9q5dp condition met
pod/speaker-bt5fw condition met
🔧 Configuration de MetalLB...
ipaddresspool.metallb.io/my-ip-pool unchanged
l2advertisment.metallb.io/l2-ad unchanged
⚠️ Le dépôt Traefik est déjà ajouté, on saute cette étape.
Release "traefik" has been upgraded. Happy Helm'ing!
NAME: traefik
LAST DEPLOYED: Wed Nov 26 00:22:34 2025
NAMESPACE: traefik-system
STATUS: deployed
REVISION: 33
TEST SUITE: None
NOTES:
Traefik Proxy v2.10.0 has been deployed successfully
on traefik-system namespace !
⚠️ Attente du déploiement de Traefik...
pod/traefik-85645b5bd5-wwq2b condition met
⚙️ Installation de Rancher avec Helm...
"rancher-latest" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
... Successfully got an update from the "jetstack" chart repository
... Successfully got an update from the "puppygraph" chart repository
... Successfully got an update from the "rancher-stable" chart repository
... Successfully got an update from the "nessie-helm" chart repository
... Successfully got an update from the "rancher-latest" chart repository
... Successfully got an update from the "traefik" chart repository
Update Complete. *Happy Helm'ing!*
Release "rancher" has been upgraded. Happy Helm'ing!
NAME: rancher
LAST DEPLOYED: Wed Nov 26 00:22:41 2025
NAMESPACE: cattle-system
STATUS: deployed
```

```
Nov 26 14:40 ⓘ
d-a-s@k8s-master: ~/rancher-deployment
d-a-s@k8s-master: ~  x  d-a-s@k8s-master: ~  x  d-a-s@k8s-master: ~/rancher...  x  d-a-s@k8s-master: ~/rancher...  x  d-a-s@k8s-master: ~/rancher...  x  d-a-s@k8s-master: ~/rancher...
If you provided your own bootstrap password during installation, browse to https://rancher.k8s.ci to get started.
If this is the first time you installed Rancher, get started by running this command and clicking the URL it generates:
```
echo https://rancher.k8s.ci/dashboard/?setup=$(kubectl get secret --namespace cattle-system bootstrap-secret -o go-template='{{.data.bootstrapPassword|base64decode}}')
```

To get just the bootstrap password on its own, run:
```
kubectl get secret --namespace cattle-system bootstrap-secret -o go-template='{{.data.bootstrapPassword|base64decode}}{{ " \n" }}'
```

Happy Containering!
  └ Application de l'ingress Rancher...
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/rancher created
  └ Attente du déploiement de Rancher...
pod/rancher-9948d9484-257pv condition met
pod/rancher-f4f9bb4bf-zjx9d condition met
  ✓ Déploiement terminé!

  └ Tableau de bord Traefik: https://10.64.13.210:9000/dashboard/
  └ Rancher: https://rancher.10.64.13.211.nip.io

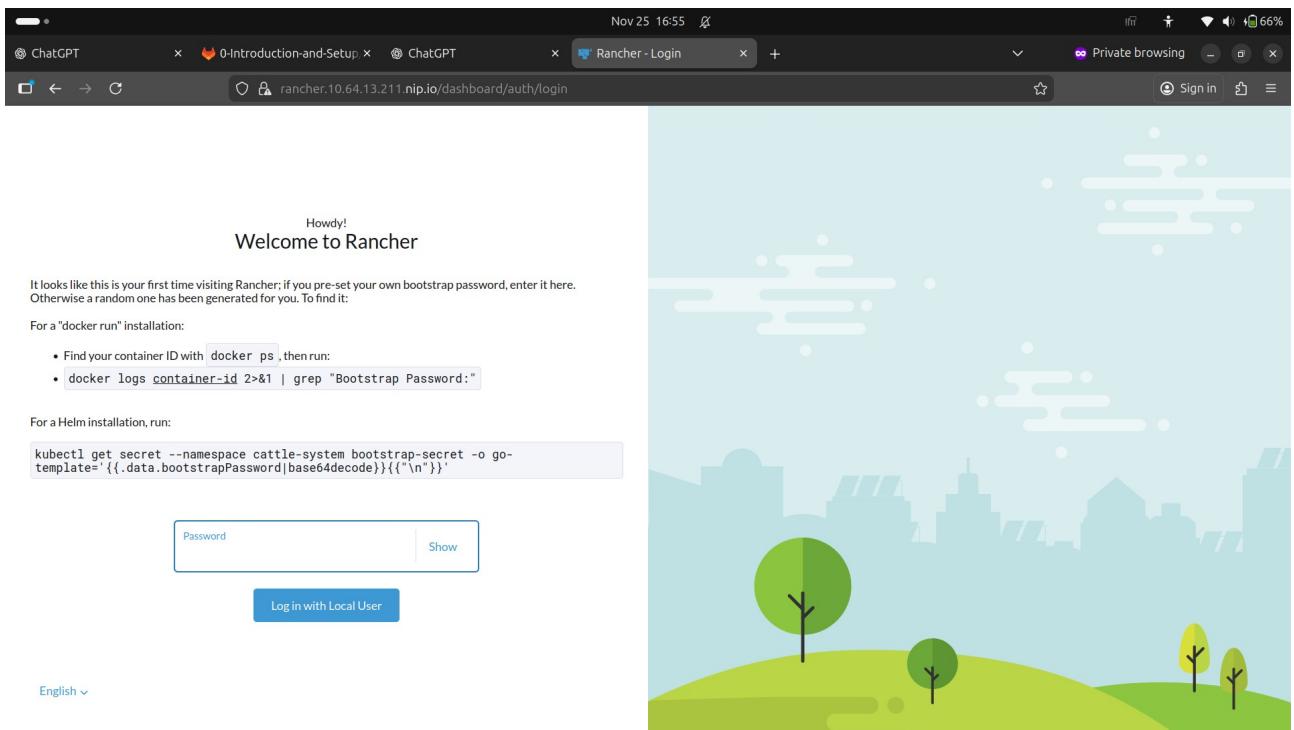
  └ Vérification des services:
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
traefik   LoadBalancer  10.107.83.14  10.64.13.210  80:30589/TCP,443:32748/TCP  5h35m
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
imperative-api-extension  ClusterIP  10.98.102.160  <none>        6666/TCP        8h
rancher   LoadBalancer  10.97.1.202   10.64.13.211  80:32474/TCP,443:30504/TCP  5h35m
rancher-webhook  ClusterIP  10.106.241.123  <none>        443/TCP         8h
d-a-s@k8s-master:~/rancher-deployment$
```

## Fin de Deploiement de Scripts

```

Nov 26 14:43  ↵
d-a-s@k8s-master: ~ x d-a-s@k8s-master: ~ x d-a-s@k8s-master: ~/rancher... x d-a-s@k8s-master: ~/rancher... x d-a-s@k8s-master: ~/rancher... x d-a-s@k8s-master: ~/rancher... x
d-a-s@k8s-master: ~ kube-system kube-controller-manager-k8s-master 1/1 Running 5 (18m ago) 10h
d-a-s@k8s-master: ~ kube-system kube-proxy-5pbkt 1/1 Running 0 10h
d-a-s@k8s-master: ~ kube-system kube-scheduler-k8s-master 1/1 Running 8 (18m ago) 10h
d-a-s@k8s-master: ~ metallb-system controller-c76b688-9q5dp 1/1 Running 0 7h36m
d-a-s@k8s-master: ~ metallb-system speaker-bt5fw 1/1 Running 17 (18m ago) 7h36m
d-a-s@k8s-master: ~ traefik-system traefik-b86df5745-5kt98 1/1 Running 0 14s
d-a-s@k8s-master:~/rancher-deployment$ kubectl get pods -n traefik-system -w
NAME READY STATUS RESTARTS AGE
traefik-b86df5745-5kt98 1/1 Running 0 28s
^C
d-a-s@k8s-master:~/rancher-deployment$ kubectl get pods -A
NAMESPACE NAME READY STATUS RESTARTS AGE
cattle-fleet-local-system fleet-agent-666ddccdf8-528v8 1/1 Running 5 (13h ago) 18h
cattle-fleet-system fleet-controller-5cc66bb5f6-n7dlw 3/3 Running 39 (15m ago) 21h
cattle-fleet-system gitjob-5cff5b4bff-vdqpm 1/1 Running 25 (15m ago) 21h
cattle-fleet-system helmops-574f89c75d-htbdf 1/1 Running 26 (15m ago) 21h
cattle-provisioning-capi-system capi-controller-manager-77f6bb9c69-dvp4d 1/1 Running 28 (15m ago) 21h
cattle-system rancher-5f4c49f685-cmjkw 1/1 Running 0 7m20s
cattle-system rancher-webhook-7ff56cb48f-8k5rm 1/1 Running 0 21h
cert-manager cert-manager-597f9759f-d6r9p 1/1 Running 2 (12h ago) 14h
cert-manager cert-manager-cainjector-6dc4d8556-qhl5t 1/1 Running 9 (15m ago) 14h
cert-manager cert-manager-webhook-6787b9c856-7wz8p 1/1 Running 0 14h
fleet-default rke2-machineconfig-cleanup-cronjob-29401925-8vw2c 0/1 Completed 0 13h
kube-system calico-kube-controllers-75cd4cc5b9-24mz7 1/1 Running 21 (12h ago) 23h
kube-system calico-node-74h45 1/1 Running 0 23h
kube-system coredns-674b8bbfcf-k8tkx 1/1 Running 0 24h
kube-system coredns-674b8bbfcf-wmljc 1/1 Running 0 24h
kube-system etcd-k8s-master 1/1 Running 2 24h
kube-system kube-apiserver-k8s-master 1/1 Running 8 (15m ago) 24h
kube-system kube-controller-manager-k8s-master 1/1 Running 7 (13h ago) 24h
kube-system kube-proxy-5pbkt 1/1 Running 0 24h
kube-system kube-scheduler-k8s-master 1/1 Running 10 (13h ago) 24h
metallb-system controller-c76b688-9q5dp 1/1 Running 0 21h
metallb-system speaker-bt5fw 1/1 Running 27 (15m ago) 21h
traefik-system traefik-b86df5745-5kt98 1/1 Running 0 13h
d-a-s@k8s-master:~/rancher-deployment$
```

**Tout les Pods sont en Running**



### La première fois vous allez obtenir le secret et vous logger

Nov 25 16:56

ChatGPT 0-Introduction-and-Setup ChatGPT Rancher Private browsing Sign in

rancher.10.64.13.211.nip.io/dashboard/auth/setup

Welcome to Rancher!

The first order of business is to set a strong password for the default admin user. We suggest using this random one generated just for you, but enter your own if you like.

Use a randomly generated password  
 Set a specific password to use

New Password \*  Show

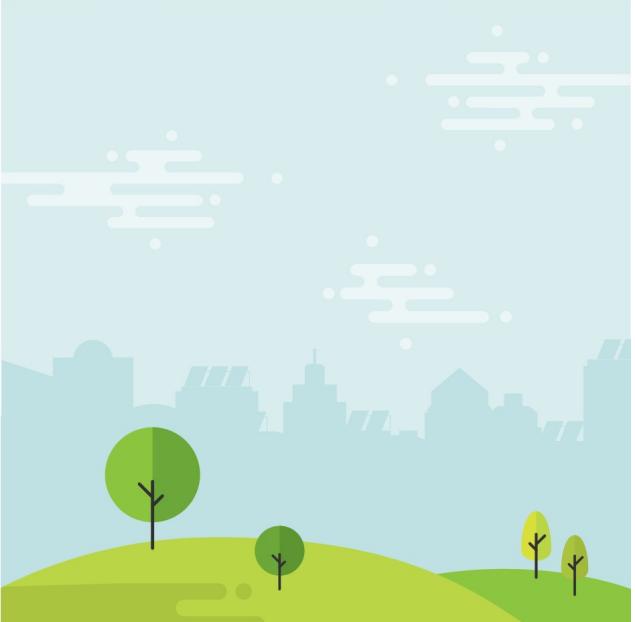
Confirm New Password \*  Show

What URL should be used for this Rancher installation? All the nodes in your clusters will need to be able to reach this.

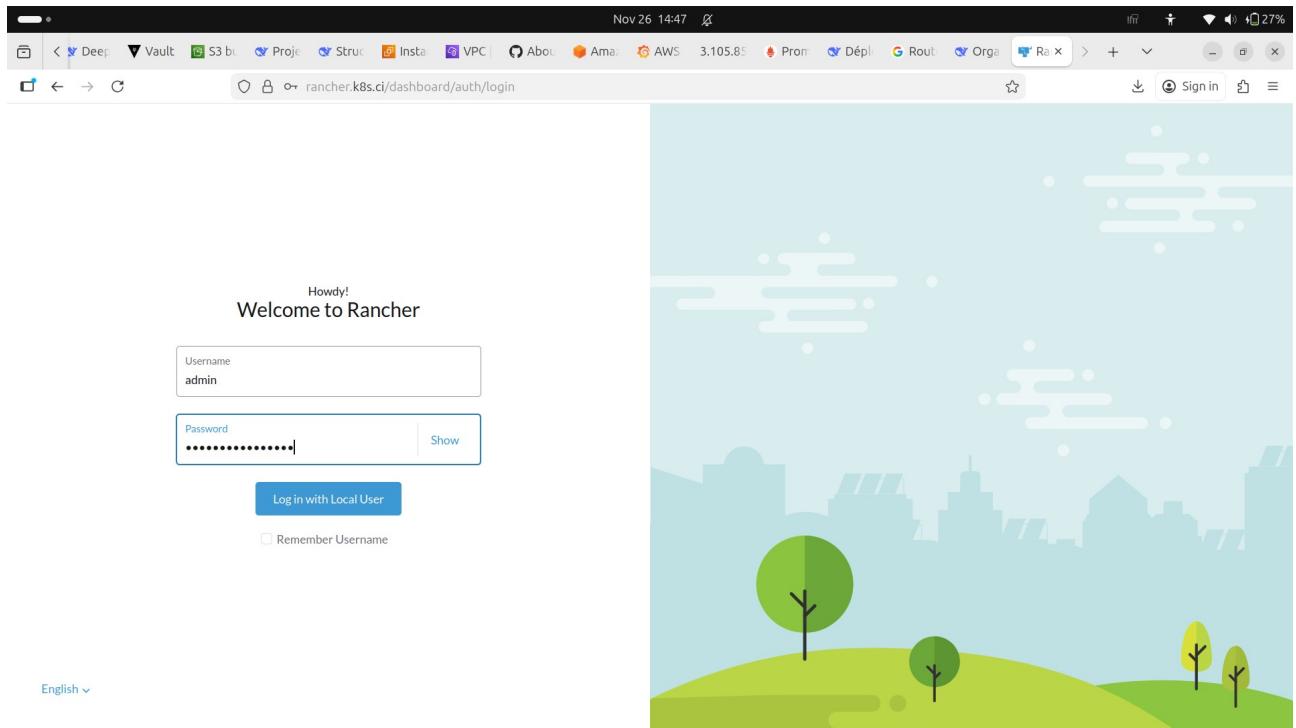
Server URL \*  
https://rancher.10.64.13.211.nip.io

By checking the box, you accept the [End User License Agreement & Terms & Conditions](#)

Continue



**Ensuite Vous decidez soit de generer un mot de passe ou d'en definir un**



On Se Connecte

The screenshot shows the Rancher Cluster Dashboard for a local cluster. The dashboard provides an overview of the cluster's resources, capacity, and event logs.

**Cluster Dashboard Statistics:**

- Total Resources: 434
- Nodes: 1
- Deployments: 14

**Capacity Overview:**

- Pods:** Used 22 / 110, 20.00% (green bar)
- CPU:** Reserved 1.6 / 8 cores, 20.00% (green bar)
- Memory:** Reserved 123 / 15 GiB, 8.20% (green bar); Used 0 / 15 GiB, 0.00% (grey bar)

**Events Tab:** Shows a list of events with columns: Reason, Object, Message, Name, First Seen, Last Seen, Count. A link "Full events list" is available.

On peut voir les informations sur notre Cluster

Only User Namespaces							
			Name	Description			
1	FailedMount	Pod traefik-b86df5745-5kt98	Name: "traefik-b86df5745-5kt98", UID: "ac173aa4-5300-423d-8a73-4fd029d0687a"	expired and refresh failed; Post "https://10.64.13.203:6443/api/v1/namespaces/traefik-system/serviceaccounts/traefik/token"; dial tcp 10.64.13.203:6443: connect: network is unreachable	traefik-b86df5745-5kt98.187b6abf4bfce01	13h	13h 6
2	FailedMount	Pod traefik-b86df5745-5kt98	MountVolume.SetUp failed for volume "kube-api-access-fpcqk": failed to fetch token; Post "https://10.64.13.203:6443/api/v1/namespaces/traefik-system/serviceaccounts/traefik/token"; dial tcp 10.64.13.203:6443: connect: connection refused	MountVolume.SetUp failed for volume "kube-api-access-fpcqk": failed to fetch token; Post "https://10.64.13.203:6443/api/v1/namespaces/traefik-system/serviceaccounts/traefik/token"; dial tcp 10.64.13.203:6443: connect: connection refused	traefik-b86df5745-5kt98.187b92b35d866353	59m	59m 1
3	FailedMount	Pod controller-c76b688-9q5dp	MountVolume.SetUp failed for volume "kube-api-access-4zt8x": failed to fetch token; serviceaccounts "controller" is forbidden: User "system:node:k8s-master" cannot create resource "serviceaccounts/token" in API group "" in the namespace "metallb-system": no relationship found between node 'k8s-master' and this object	MountVolume.SetUp failed for volume "kube-api-access-4zt8x": failed to fetch token; serviceaccounts "controller" is forbidden: User "system:node:k8s-master" cannot create resource "serviceaccounts/token" in API group "" in the namespace "metallb-system": no relationship found between node 'k8s-master' and this object	controller-c76b688-9q5dp.187b92b44f7b7608	59m	59m 1
4	RegisteredNode	Node k8s-master	Node k8s-master event: Registered Node k8s-master in Controller	Node k8s-master event: Registered Node k8s-master in Controller	k8s-master.187b92b8afa82296	59m	59m 1
5	BackOff	Pod speaker-bt5fw	Back-off restarting failed container speaker in pod speaker-bt5fw_metalb-system(4ec0f89d-dd02-4266-beab-b387e077607a)	Back-off restarting failed container speaker in pod speaker-bt5fw_metalb-system(4ec0f89d-dd02-4266-beab-b387e077607a)	speaker-bt5fw.187b5db94ea9d7d	17h	55m 165
6	nodeAssigned	Service traefik	announcing from node "k8s-master" with protocol "layer2"	announcing from node "k8s-master" with protocol "layer2"	traefik.187b92f91dcba1510	54m	54m 2
7	FailedBinding	PersistentVolumeClaim traefik	no persistent volumes available for this claim and no storage class is set	no persistent volumes available for this claim and no storage class is set	traefik.187b92b8b5a3ff84	59m	44m 61
8	Unhealthy	Pod speaker-bt5fw	Liveness probe failed: Get "http://10.64.13.203:7472/metrics"; dial tcp 10.64.13.203:7472: connect: network is unreachable	Liveness probe failed: Get "http://10.64.13.203:7472/metrics"; dial tcp 10.64.13.203:7472: connect: network is unreachable	speaker-bt5fw.187b5db445876934	17h	40m 15

**On Peut Superviser notre cluster si l 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 !**