

Guide d'Installation

Harvester HCI, RKE2 Kubernetes & Rancher

Infrastructure RBER - Sites Fidjrossè et Parakou

Version 1.1 - Janvier 2026

Équipe Infrastructure RBER

Table des matières

1. Prérequis
2. Architecture cible
3. Installation Harvester
4. Configuration réseau Harvester
5. Configuration stockage
6. Création des VMs pour RKE2
7. Installation RKE2
8. Installation et configuration Rancher
9. Intégration d'un cluster RKE2 dans Rancher
10. Modification du nom d'un cluster dans Rancher
11. Configuration post-installation
12. Validation
13. Troubleshooting

1. Prérequis

1.1 Matériel requis

Composant	Minimum	Recommandé RBER	Notes
Serveurs	3 nœuds	6 nœuds (3/site)	HPE DL380 Gen10+
CPU	8 cœurs/nœud	22+ cœurs	VT-x/AMD-V requis
RAM	32 GB/nœud	64+ GB	ECC recommandée
Stockage système	250 GB SSD	500 GB NVMe	RAID1 recommandé
Stockage données	500 GB SSD	2+ TB NVMe	JBOD pour Longhorn
Réseau	2x 10 GbE	2x 25 GbE	Bonding optionnel

1.2 Configuration réseau

Planification des réseaux pour le site Fidjrossè :

- Management Network : 10.29.112.0/24
- VM Network : 10.29.113.0/24
- Storage Network : 10.29.114.0/24 (VLAN isolé)
- VIP HAProxy : 10.29.112.100
- VIP Rancher : 10.29.112.101

1.3 Prérequis logiciels

- Image ISO Harvester v1.3.x ou supérieur
- Helm v3.x installé sur le poste d'administration
- cert-manager pour les certificats TLS
- Accès console IPMI/iLO sur chaque serveur
- Serveur DNS configuré

2. Architecture cible

2.1 Topologie Harvester

Le cluster Harvester se compose de 3 nœuds par site en mode haute disponibilité :

Nœud	Rôle	IP Management	Ressources
harv-fidj-01	Master	10.29.112.11	22 cores, 64 GB
harv-fidj-02	Master	10.29.112.12	22 cores, 64 GB
harv-fidj-03	Master	10.29.112.13	22 cores, 64 GB

VIP du cluster Harvester : 10.29.112.10

2.2 Topologie RKE2

VM	Rôle	vCPU	RAM	IP
rke2-etcd-01	etcd	4	8 GB	10.29.113.11
rke2-etcd-02	etcd	4	8 GB	10.29.113.12
rke2-etcd-03	etcd	4	8 GB	10.29.113.13
rke2-cp-01	control-plane	4	16 GB	10.29.113.21
rke2-cp-02	control-plane	4	16 GB	10.29.113.22
rke2-cp-03	control-plane	4	16 GB	10.29.113.23
rke2-wk-01	worker	8	32 GB	10.29.113.31
rke2-wk-02	worker	8	32 GB	10.29.113.32
rke2-wk-03	worker	8	32 GB	10.29.113.33

3. Installation Harvester

3.1 Préparation des serveurs

Configurer le BIOS de chaque serveur :

1. Activer VT-x/AMD-V (virtualisation matérielle)
2. Activer VT-d/AMD-Vi (IOMMU)
3. Désactiver Secure Boot temporairement
4. Configurer l'ordre de boot : USB/ISO > Disque
5. Configurer RAID1 pour le disque système, JBOD pour les données

3.2 Installation du premier nœud (Create)

1. Booter sur l'ISO Harvester
2. Sélectionner "Create a new Harvester cluster"
3. Configurer le disque d'installation : /dev/sda (système), /dev/sdb (données)
4. Hostname : harv-fidj-01

Configuration réseau (IP statique) :

```
Interface: ens192
IP Address: 10.29.112.11
Subnet Mask: 255.255.255.0
Gateway: 10.29.112.1
DNS: 10.29.112.1
```

Configuration du VIP cluster :

```
Management VIP: 10.29.112.10
VIP Mode: Static
```

Générer un token de cluster sécurisé :

```
openssl rand -hex 32
```

3.3 Installation des nœuds suivants (Join)

1. Booter sur l'ISO Harvester
2. Sélectionner "Join an existing Harvester cluster"
3. Configurer le hostname : harv-fidj-02 (ou harv-fidj-03)
4. Configurer l'IP statique : 10.29.112.12 (ou .13)
5. Entrer le Management VIP : 10.29.112.10
6. Entrer le token du cluster

4. Configuration réseau Harvester

4.1 Accéder à l'interface Harvester

```
URL: https://10.29.112.10
User: admin
Password: [mot de passe configuré]
```

4.2 Créer un réseau VM (VLAN)

Aller dans Networks > VM Networks > Create :

```
Name: vm-network
Namespace: default
VLAN ID: 113
Cluster Network: mgmt
```

5. Configuration stockage

5.1 Ajouter des disques au cluster Longhorn

Aller dans Hosts > [nœud] > Storage, puis pour chaque disque :

- Sélectionner le disque non utilisé
- Cliquer sur "Add as Longhorn Disk"
- Configurer : Name, Path, Storage Reserved, Tags (ssd, nvme)

6. Création des VMs pour RKE2

6.1 Télécharger l'image Ubuntu

Aller dans Images > Create :

```
Name: ubuntu-22.04-server
URL: https://cloud-images.ubuntu.com/releases/22.04/release/ubuntu-22.04-server-cloudimg-amd64.img
```

6.2 Configuration cloud-init

Template cloud-init pour chaque VM :

```
#cloud-config
hostname: rke2-etcd-01
manage_etc_hosts: true
users:
  - name: rber
    sudo: ALL=(ALL) NOPASSWD:ALL
    shell: /bin/bash
    ssh_authorized_keys:
      - ssh-ed25519 AAAA... admin@rber.bj
package_update: true
packages:
  - qemu-guest-agent
  - curl
  - open-iscsi
  - nfs-common
runcmd:
  - systemctl enable --now qemu-guest-agent
  - systemctl enable --now iscsid
```


7. Installation RKE2

7.1 Préparation des nœuds

Exécuter sur tous les nœuds :

```
# Désactiver swap
swapoff -a
sed -i '/swap/d' /etc/fstab

# Charger les modules kernel
modprobe overlay
modprobe br_netfilter

# Paramètres sysctl
cat <<EOF | tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
sysctl --system
```

7.2 Premier nœud etcd

Sur rke2-etcd-01 :

```
mkdir -p /etc/rancher/rke2

cat <<EOF > /etc/rancher/rke2/config.yaml
token: "rber-cluster-token-secret-2024"
tls-san:
  - "10.29.112.100"
  - "rke2-api.rber.bj"
disable:
  - rke2-ingress-nginx
node-taint:
  - "node-role.kubernetes.io/etcd=true:NoExecute"
node-label:
  - "node-role.kubernetes.io/etcd=true"
EOF

curl -sfl https://get.rke2.io | INSTALL_RKE2_TYPE="server" sh -
systemctl enable rke2-server.service
systemctl start rke2-server.service
```

7.3 Nœuds etcd additionnels

Sur rke2-etcd-02 et rke2-etcd-03 :

```
cat <<EOF > /etc/rancher/rke2/config.yaml
server: https://10.29.113.11:9345
token: "rber-cluster-token-secret-2024"
tls-san:
  - "10.29.112.100"
disable:
  - rke2-ingress-nginx
node-taint:
  - "node-role.kubernetes.io/etcd=true:NoExecute"
node-label:
  - "node-role.kubernetes.io/etcd=true"
EOF

curl -sfl https://get.rke2.io | INSTALL_RKE2_TYPE="server" sh -
systemctl enable rke2-server.service
systemctl start rke2-server.service
```

7.4 Nœuds Control Plane

Sur rke2-cp-01, rke2-cp-02, rke2-cp-03 :

```
cat <<EOF > /etc/rancher/rke2/config.yaml
server: https://10.29.113.11:9345
token: "rber-cluster-token-secret-2024"
tls-san:
  - "10.29.112.100"
disable:
  - rke2-ingress-nginx
node-taint:
  - "node-role.kubernetes.io/control-plane=true:NoSchedule"
node-label:
  - "node-role.kubernetes.io/control-plane=true"
EOF

curl -sL https://get.rke2.io | INSTALL_RKE2_TYPE="server" sh -
systemctl enable rke2-server.service
systemctl start rke2-server.service
```

7.5 Workers

Sur rke2-wk-01, rke2-wk-02, rke2-wk-03 :

```
cat <<EOF > /etc/rancher/rke2/config.yaml
server: https://10.29.113.11:9345
token: "rber-cluster-token-secret-2024"
node-label:
  - "node-role.kubernetes.io/worker=true"
EOF

curl -sL https://get.rke2.io | INSTALL_RKE2_TYPE="agent" sh -
systemctl enable rke2-agent.service
systemctl start rke2-agent.service
```

8. Installation et configuration Rancher

8.1 Prérequis Rancher

Rancher nécessite :

- Un cluster Kubernetes fonctionnel (RKE2 ou dédié)
- cert-manager pour la gestion des certificats
- Un Ingress Controller (nginx ou traefik)
- Un nom DNS résolvant vers le Load Balancer

Configuration DNS requise :

```
rancher.rber.bj -> 10.29.112.101 (VIP Rancher)
```

8.2 Installer cert-manager

```
# Ajouter le repo Jetstack
helm repo add jetstack https://charts.jetstack.io
helm repo update

# Installer les CRDs cert-manager
kubectl apply -f https://github.com/cert-manager/cert-
manager/releases/download/v1.14.4/cert-manager.crds.yaml

# Créer le namespace
kubectl create namespace cert-manager

# Installer cert-manager
helm install cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --version v1.14.4 \
  --wait

# Vérifier l'installation
kubectl get pods -n cert-manager
```

8.3 Installer Rancher

```
# Ajouter le repo Rancher
helm repo add rancher-stable https://releases.rancher.com/server-charts/stable
helm repo update

# Créer le namespace cattle-system
kubectl create namespace cattle-system

# Installer Rancher avec certificats Let's Encrypt
helm install rancher rancher-stable/rancher \
  --namespace cattle-system \
  --set hostname=rancher.rber.bj \
  --set bootstrapPassword=admin \
  --set ingress.tls.source=letsEncrypt \
  --set letsEncrypt.email=admin@rber.bj \
  --set letsEncrypt.ingress.class=nginx \
  --set replicas=3 \
  --wait
```

Alternative avec certificats auto-signés (environnement de test) :

```
helm install rancher rancher-stable/rancher \
  --namespace cattle-system \
  --set hostname=rancher.rber.bj \
  --set bootstrapPassword=admin \
  --set ingress.tls.source=rancher \
  --set replicas=3 \
  --wait
```

8.4 Vérifier l'installation

```
# Vérifier les pods Rancher
kubectl get pods -n cattle-system

# Vérifier le déploiement
kubectl get deployment rancher -n cattle-system

# Vérifier l'ingress
kubectl get ingress -n cattle-system

# Récupérer le mot de passe bootstrap (si oublié)
kubectl get secret --namespace cattle-system bootstrap-secret \
  -o go-template='{{.data.bootstrapPassword|base64decode}}{{"\n"}}'

# Attendre que Rancher soit prêt
kubectl rollout status deployment rancher -n cattle-system
```

8.5 Première connexion à Rancher

1. Accéder à <https://rancher.rber.bj>
2. Entrer le mot de passe bootstrap : admin
3. Définir un nouveau mot de passe administrateur sécurisé
4. Configurer l'URL du serveur Rancher : <https://rancher.rber.bj>
5. Accepter les conditions d'utilisation

8.6 Configuration post-installation Rancher

Paramètres recommandés dans Settings :

- server-url : <https://rancher.rber.bj>
- ui-default-landing : clusters (page d'accueil sur les clusters)
- ui-pl : rancher (branding Rancher)
- auth-user-session-ttl-minutes : 960 (16 heures)

9. Intégration d'un cluster RKE2 dans Rancher

9.1 Méthodes d'intégration

Deux méthodes pour ajouter un cluster RKE2 existant :

Méthode	Cas d'usage	Avantages
Import manuel	Cluster existant	Simple, rapide
Agent Rancher	Gestion complète	Monitoring, logs intégrés

9.2 Import d'un cluster existant (méthode recommandée)

Étape 1 : Créer l'import dans Rancher

1. Se connecter à Rancher : <https://rancher.rber.bj>
2. Aller dans Cluster Management > Clusters
3. Cliquer sur "Import Existing"
4. Sélectionner "Generic" pour un cluster RKE2 standard
5. Entrer le nom du cluster : rke2-fidjrosse
6. Optionnel : ajouter une description et des labels
7. Cliquer sur "Create"

Étape 2 : Appliquer la commande d'import

Rancher génère une commande kubectl à exécuter sur le cluster cible :

```
# Copier la commande générée par Rancher
# Elle ressemble à :
curl --insecure -sL https://rancher.rber.bj/v3/import/xxxxx.yaml | kubectl
apply -f -

# Ou avec le manifest direct :
kubectl apply -f https://rancher.rber.bj/v3/import/xxxxx.yaml
```

Si le certificat est auto-signé, utiliser la commande avec `--insecure` :

```
curl --insecure -sL https://rancher.rber.bj/v3/import/xxxxx.yaml | kubectl
apply -f -
```

Étape 3 : Vérifier l'intégration

```
# Sur le cluster importé, vérifier les agents Rancher
kubectl get pods -n cattle-system

# Vérifier les déploiements
kubectl get deployment -n cattle-system

# Résultat attendu :
# cattle-cluster-agent      1/1      Running
# cattle-node-agent         (DaemonSet sur chaque nœud)
```

Dans Rancher, le cluster passe par les états :

- Pending → Waiting → Active
- L'import complet prend 2-5 minutes

9.3 Configuration des labels et annotations

Ajouter des métadonnées au cluster après import :

```
# Via kubectl sur le cluster Rancher (local)
kubectl label cluster.management.cattle.io/rke2-fidjrosse \
  environment=production \
  location=fidjrosse \
  -n fleet-default
```

```
# Ou via l'interface Rancher :  
# Cluster Management > [cluster] > Config > Labels & Annotations
```

9.4 Activer le monitoring sur le cluster importé

- Aller dans Cluster Management > [cluster] > Cluster Tools
- Installer "Monitoring" (Prometheus + Grafana)
- Configurer la rétention : 15 jours (recommandé)
- Configurer les ressources selon la taille du cluster

10. Modification du nom d'un cluster dans Rancher

IMPORTANT : L'interface Rancher ne permet pas de modifier le nom d'un cluster après sa création. Tu dois utiliser kubectl, l'API, ou réimporter le cluster.

10.1 Comprendre les identifiants de cluster

Dans Rancher, un cluster a plusieurs identifiants :

Identifiant	Description	Modifiable
ID interne	Ex: c-m-xxxxx (génééré)	Non
Display Name	Nom affiché dans l'UI	Oui (kubectl/API)
metadata.name	Nom Kubernetes du cluster	Non (recréation requise)

10.2 Méthode 1 : Modifier le Display Name via kubectl (recommandé)

Le nom du cluster n'est PAS modifiable via l'interface Rancher. Tu dois utiliser kubectl ou l'API.

Cette méthode modifie le nom affiché dans l'interface sans supprimer le cluster.

Via kubectl

```
# Lister les clusters pour trouver l'ID
kubectl get clusters.management.cattle.io -A

# Modifier le display name
kubectl patch clusters.management.cattle.io <cluster-id> \
  --type='merge' \
  -p '{"spec":{"displayName":"rke2-fidjrosse-prod"}}'

# Exemple concret :
kubectl patch clusters.management.cattle.io c-m-abc123 \
  --type='merge' \
  -p '{"spec":{"displayName":"RKE2 Fidjrossè Production"}}'

# Vérifier la modification
kubectl get clusters.management.cattle.io c-m-abc123 \
  -o jsonpath='{.spec.displayName}'
```

10.3 Méthode 2 : Modifier via l'API Rancher

```
# Récupérer un token API
# Rancher UI > User Avatar > Account & API Keys > Create API Key

# Lister les clusters
curl -s -k \
  -H "Authorization: Bearer token-xxxxx:yyyyyyyy" \
  https://rancher.rber.bj/v3/clusters | jq '.data[].name'

# Modifier le nom d'affichage
curl -s -k -X PUT \
  -H "Authorization: Bearer token-xxxxx:yyyyyyyy" \
  -H "Content-Type: application/json" \
  -d '{"name":"RKE2 Fidjrossè Production"}' \
  https://rancher.rber.bj/v3/clusters/<cluster-id>
```

10.4 Méthode 3 : Renommer complètement (réimport)

Si tu as besoin de modifier le metadata.name (nom Kubernetes), il faut supprimer et réimporter le cluster.

ATTENTION : Cette méthode supprime l'historique des événements dans Rancher.

Étape 1 : Exporter la configuration actuelle

```
# Sauvegarder les infos du cluster avant suppression
kubectl get clusters.management.cattle.io <cluster-id> -o yaml > cluster-backup.yaml

# Noter les labels et annotations importants
kubectl get clusters.management.cattle.io <cluster-id> \
  -o jsonpath='{.metadata.labels}' > cluster-labels.json
```

Étape 2 : Supprimer l'agent du cluster cible

```
# Sur le cluster RKE2 (pas le cluster Rancher)
kubectl delete namespace cattle-system
kubectl delete namespace cattle-fleet-system
kubectl delete namespace cattle-fleet-local-system

# Supprimer les CRDs cattle si présents
kubectl delete crd -l cattle.io/creator=norman
```

Étape 3 : Supprimer le cluster de Rancher

- Dans Rancher : Cluster Management > [cluster] > : > Delete
- Ou via kubectl : `kubectl delete clusters.management.cattle.io <cluster-id>`

Étape 4 : Réimporter avec le nouveau nom

- Suivre la procédure de la section 9.2
- Utiliser le nouveau nom souhaité lors de l'import
- Réappliquer les labels et annotations sauvegardés

10.5 Bonnes pratiques de nommage

Convention de nommage recommandée pour RBER :

Pattern	Exemple	Usage
<type>-<site>	rke2-fidjrosse	Simple
<type>-<site>-<env>	rke2-fidjrosse-prod	Multi-environnement
<projet>-<type>-<site>	rber-rke2-fidjrosse	Multi-projet

Règles :

- Utiliser uniquement des minuscules
- Séparer par des tirets (-)
- Éviter les caractères spéciaux et accents
- Maximum 63 caractères (limite Kubernetes)

11. Configuration post-installation

11.1 Configurer kubectl

```
mkdir -p ~/.kube
cp /etc/rancher/rke2/rke2.yaml ~/.kube/config
chmod 600 ~/.kube/config
sed -i 's/127.0.0.1/10.29.112.100/g' ~/.kube/config

echo 'export PATH=$PATH:/var/lib/rancher/rke2/bin' >> ~/.bashrc
source ~/.bashrc

kubectl get nodes
```

11.2 Installer Longhorn

```
helm repo add longhorn https://charts.longhorn.io
helm repo update

kubectl create namespace longhorn-system

helm install longhorn longhorn/longhorn \
  --namespace longhorn-system \
  --set defaultSettings.defaultReplicaCount=3 \
  --set persistence.defaultClass=true \
  --wait
```

11.3 Installer HAProxy Ingress Controller

```
helm repo add haproxytech https://haproxytech.github.io/helm-charts
helm repo update

kubectl create namespace haproxy-controller

helm install haproxy-ingress haproxytech/kubernetes-ingress \
  --namespace haproxy-controller \
  --set controller.replicaCount=2 \
  --set controller.service.type=LoadBalancer \
  --set controller.service.loadBalancerIP=10.29.112.100 \
  --wait
```

11.4 Installer CloudNativePG

```
kubectl apply -f \
  https://raw.githubusercontent.com/cloudnative-pg/cloudnative-pg/release-1.24/releases/cnpg-1.24.0.yaml

kubectl get deployment -n cnpg-system
```

12. Validation

12.1 Checklist Harvester

- Tous les nœuds Harvester en état Ready
- Longhorn (stockage) opérationnel
- VMs créées et démarrées

12.2 Checklist RKE2

```
kubectl get nodes -o wide  
# Résultat : 9 nœuds Ready  
  
kubectl get pods -A | grep -v Running  
# Résultat : vide (tous Running)
```

12.3 Checklist Rancher

- Interface accessible : <https://rancher.rber.bj>
- 3 pods Rancher Running dans cattle-system
- Cluster(s) importé(s) en état Active
- Agents cattle-cluster-agent et cattle-node-agent Running sur chaque cluster

13. Troubleshooting

13.1 Problèmes Rancher courants

Problème	Cause	Solution
Rancher inaccessible	Pods non Ready	kubectl rollout restart deploy/rancher -n cattle-system
Cluster Pending	Agent non déployé	Réappliquer la commande d'import
Erreur certificat	cert-manager non prêt	Vérifier pods cert-manager
Cluster Unavailable	Réseau entre Rancher et cluster	Vérifier connectivité port 443

13.2 Commandes de diagnostic Rancher

```
# Logs Rancher
kubectl logs -n cattle-system -l app=rancher --tail=100 -f

# État des agents sur cluster importé
kubectl get pods -n cattle-system
kubectl logs -n cattle-system -l app=cattle-cluster-agent

# Vérifier la connexion cluster
kubectl get clusters.management.cattle.io

# Forcer la reconnexion d'un agent
kubectl delete pod -n cattle-system -l app=cattle-cluster-agent
```

13.3 Réinstaller un agent cluster

Si l'agent est corrompu ou ne se connecte plus :

```
# Sur le cluster cible
kubectl delete namespace cattle-system

# Dans Rancher, aller sur le cluster > Registration
# Copier la nouvelle commande d'enregistrement
# Exécuter sur le cluster cible
curl --insecure -sfL https://rancher.rber.bj/v3/import/xxxxx.yaml | kubectl
apply -f -
```

13.4 Ports réseau requis

Port	Proto	Source	Dest	Description
443	TCP	Clients	Rancher	UI et API Rancher
443	TCP	Clusters	Rancher	Agent -> Rancher
6443	TCP	Rancher	Clusters	Kubernetes API

— Fin du document —

Contact : devops@rber.bj

Documentation : <https://ranchermanager.docs.rancher.com>