Despliegue de Rook-Ceph en clúster K3s (SUSE) con discos locales (Bluestore)

Esta guía describe cómo desplegar un clúster Rook-Ceph sobre K3s en servidores SUSE con **discos locales** estándar (sin iSCSI). Se adapta a la arquitectura de 4 servidores: `srvfkvm1`, `srvfkvm2`, `srvfkvm3`, `srvfkvm4`, cada uno con 6 discos de ~900GB y una red de almacenamiento dedicada en la VLAN 30 (`192.168.3.0/24`).

---

1. Requisitos previos

- 4 nodos K3s funcionando: `SRVFKVM01`, `SRVFKVM02`, `SRVFKVM03`, `SRVFKVM04`
- Cada nodo con 6 discos locales dedicados para Ceph (`/dev/sdj` a `/dev/sdo` en cada servidor)
- K3s y `kubectl` funcionando y configurado
- Acceso completo a Internet desde todos los nodos
- Cada nodo debe tener una IP fija en `192.168.3.0/24` (VLAN 30) usada exclusivamente para almacenamiento

---

2. Preparar los nodos SUSE (sólo discos locales)

No es necesario configurar iSCSI ni multipath. **Asegúrate de que los discos están vacíos y sin particionar,** o bien elimina las particiones creadas (Ceph las sobreescribirá).

Verifica los discos en cada nodo:

```
lsblk | grep sd[j-o]
```

---

3. Configurar el firewall para red de almacenamiento (VLAN 30)

Asumiendo que tu interfaz de red para la VLAN 30 se llama `vlan30`, añádela a la zona `trusted` para permitir el tráfico de Ceph:

```
sudo firewall-cmd --zone=trusted --add-interface=vlan30 --permanent
sudo firewall-cmd --reload
```

Puedes verificar:

```
sudo firewall-cmd --list-all --zone=trusted
```

Deberías ver `interfaces: vlan30` listada.

---

4. Crear namespace y CRDs de Rook-Ceph

```
kubectl create namespace rook-ceph

# Clona el repositorio oficial de Rook
git clone https://github.com/rook/rook.git
cd rook/deploy/examples

# Aplica CRDs y recursos comunes
kubectl apply -f crds.yaml
kubectl apply -f common.yaml
```

---

5. Desplegar el operador Rook-Ceph

```
kubectl apply -f operator.yaml
```

---

6. Crear el clúster Ceph con discos locales

Crea un archivo `ceph-cluster.yaml` con el siguiente contenido (ajusta nombres/discos según corresponda):

```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
  name: rook-ceph
  namespace: rook-ceph
spec:
  cephVersion:
    image: quay.io/ceph/ceph:v18
  dataDirHostPath: /var/lib/rook
  network:
    connections:
      clusterNetwork: "192.168.3.0/24"
      publicNetwork: "192.168.3.0/24"
  mon:
    count: 3
    allowMultiplePerNode: false
  dashboard:
    enabled: true
  storage:
    useAllNodes: false
```

```
    useAllDevices: false
    nodes:
      - name: SRVFKVM01
        devices:
          - name: /dev/sdj
          - name: /dev/sdk
          - name: /dev/sdl
          - name: /dev/sdm
          - name: /dev/sdn
          - name: /dev/sdo
      - name: SRVFKVM02
        devices:
          - name: /dev/sdj
          - name: /dev/sdk
          - name: /dev/sdl
          - name: /dev/sdm
          - name: /dev/sdn
          - name: /dev/sdo
      - name: SRVFKVM03
        devices:
          - name: /dev/sdj
          - name: /dev/sdk
          - name: /dev/sdl
          - name: /dev/sdm
          - name: /dev/sdn
          - name: /dev/sdo
      - name: SRVFKVM04
        devices:
          - name: /dev/sdj
          - name: /dev/sdk
          - name: /dev/sdl
          - name: /dev/sdm
          - name: /dev/sdn
          - name: /dev/sdo
```

> **Asegúrate de que los nombres de los nodos (`name:`) coinciden con el valor mostrado por `kubectl get nodes`.

Aplica el manifiesto:

```
kubectl apply -f ceph-cluster.yaml
```

---

7. Verifica el despliegue de Ceph

```
kubectl -n rook-ceph get pods
```

- Espera a que los pods estén en estado `Running`.

Para comprobar el estado de Ceph:

```
# Primero espera a que el pod rook-ceph-tools esté disponible
kubectl -n rook-ceph exec -it deploy/rook-ceph-tools -- ceph status
```

---

8. Crear CephBlockPool y StorageClass (replica:4)

ceph-blockpool.yaml:

```
apiVersion: ceph.rook.io/v1
kind: CephBlockPool
metadata:
  name: replicado-4x
  namespace: rook-ceph
spec:
  failureDomain: host
  replicated:
    size: 4
```

ceph-storageclass.yaml:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ceph-rbd-replica4
provisioner: rook-ceph.rbd.csi.ceph.com
parameters:
  clusterID: rook-ceph
  pool: replicado-4x
  imageFormat: "2"
  imageFeatures: layering
  csi.storage.k8s.io/fstype: ext4
reclaimPolicy: Delete
allowVolumeExpansion: true
mountOptions:
  - discard
```

Aplica ambos:

```
kubectl apply -f ceph-blockpool.yaml
kubectl apply -f ceph-storageclass.yaml
```

9. Exponer el dashboard de Ceph

Crea el siguiente servicio `dashboard-service.yaml` para exponer el Dashboard vía `NodePort`:

```
apiVersion: v1
kind: Service
metadata:
  name: rook-ceph-mgr-dashboard
  namespace: rook-ceph
spec:
  type: NodePort
  ports:
    - port: 8443
      targetPort: 8443
      protocol: TCP
      name: https-dashboard
  selector:
    app: rook-ceph-mgr
```

Aplica el manifiesto:

```
kubectl apply -f dashboard-service.yaml
```

> Cuando esté disponible una capa de ingress con TLS (por ejemplo cert-manager + ingress-nginx), se recomienda eliminar este `NodePort` y crear un `Ingress` con dominio y certificado TLS.

Obtén el puerto del dashboard:

```
kubectl -n rook-ceph get svc | grep dashboard
```

Obtén la contraseña:

```
kubectl -n rook-ceph get secret rook-ceph-dashboard-password -o jsonpath="
{.data.password}" | base64 -d
```

Accede en tu navegador a:

```
https://<IP_nodo>:<NodePort>
```

Usuario: `admin`
Contraseña: (la anterior)
