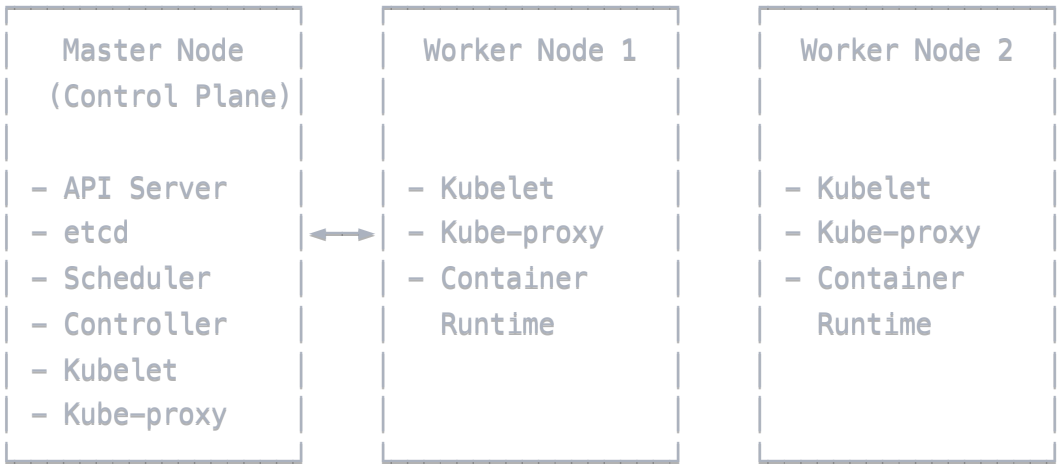
Setup Cluster Kubernetes su Raspberry Pi 5 con Ubuntu Server 24.04.2

Panoramica del Progetto

Questo documento descrive il processo completo per la creazione di un cluster Kubernetes su tre Raspberry Pi 5 utilizzando Ubuntu Server 24.04.2. Il cluster è composto da:

- 1 Master Node (Control Plane)
- 2 Worker Nodes

Architettura del Sistema



Hardware e Software Utilizzati

Hardware

- 3x Raspberry Pi 5 (8GB RAM consigliato)
- 3x MicroSD Card (Classe 10, 64GB+ consigliato)
- Switch di rete / Router
- Cavi Ethernet

Software

- Ubuntu Server 24.04.2 LTS (ARM64)
- Kubernetes v1.28
- containerd (Container Runtime)
- Flannel (Network Plugin)

Fase 1: Preparazione del Sistema

1.1 Installazione Ubuntu Server

- Installato Ubuntu Server 24.04.2 su tutti e tre i Raspberry Pi
- Configurata connessione SSH per accesso remoto
- Aggiornato il sistema all'ultima versione

1.2 Configurazione di Base

Aggiornamento del sistema (tutti i nodi):

```bash
sudo apt update && sudo apt upgrade -y
```

Disabilitazione dello swap:

```bash
sudo swapoff -a
```

PROBLEMA RISCONTRATO: Il comando originale per commentare lo swap in `/etc/fstab` dava errore di sintassi.

Comando che causava errore:

```bash
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
# Errore: sed: -e expression #1, char 25: unknown option to 's'
```

SOLUZIONE APPLICATA:

```bash
sudo sed -i '/swap/s/^/#/' /etc/fstab
```

Verifica disabilitazione swap:

```bash
free -h  # Deve mostrare 0B per Swap
```

1.3 Configurazione Moduli Kernel

Caricamento moduli necessari:

bash

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter
```

Configurazione parametri sysctl:

bash

```
cat <<EOF | sudo 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

sudo sysctl --system
```

Fase 2: Installazione Container Runtime

2.1 Installazione containerd

Installazione del pacchetto:

bash

```
sudo apt install -y containerd
```

Configurazione containerd:

bash

```
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/containerd/config.tom
sudo systemctl restart containerd
sudo systemctl enable containerd
```

Fase 3: Installazione Kubernetes

3.1 Primo Tentativo di Installazione

PROBLEMA RISCONTRATO: I pacchetti kubelet, kubeadm e kubectl non venivano trovati.

Errore ricevuto:

```
E: unable to locate package kubelet
E: unable to locate package kubeadm
E: unable to locate package kubectl
```

CAUSA: Repository Kubernetes non configurato correttamente per Ubuntu 24.04.2

3.2 Risoluzione Problema Repository

Rimozione configurazioni precedenti:

bash

```
sudo rm -f /etc/apt/sources.list.d/kubernetes.list
sudo rm -f /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

Installazione dipendenze:

bash

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
```

Configurazione corretta del repository:

bash

```
# Creazione directory per le chiavi
sudo mkdir -p /etc/apt/keyrings

# Download e installazione chiave GPG
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmo

# Aggiunta repository
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/
```

Aggiornamento indice pacchetti e installazione:

bash

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
sudo systemctl enable kubelet
```

Verifica installazione:

```bash
kubeadm version
kubelet --version
kubectl version --client
```

Fase 4: Configurazione Master Node

4.1 Identificazione IP Master

Comando per trovare l'IP:

```bash
ip addr show
```

4.2 Inizializzazione Cluster

PROBLEMA RISCONTRATO: Errore di sintassi bash con comando multi-riga.

Comando che causava errore:

```bash
sudo kubeadm init \
  --pod-network-cidr=10.244.0.0/16 \
  --apiserver-advertise-address=<MASTER-IP> \
  --control-plane-endpoint=<MASTER-IP>
# Errore: bash: syntax error near unexpected token 'newline'
```

SOLUZIONE APPLICATA: Comando su singola riga:

```bash
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=<MASTI
```

Esempio con IP reale:

```bash
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=192.1
```

4.3 Configurazione kubectl

Configurazione per utente corrente:

bash

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

4.4 Installazione Network Plugin

PROBLEMA RISCONTRATO: Errore di connessione durante installazione Flannel.

Errore ricevuto:

```
The connection to the server 192.168.0.202:6443 was refused
```

CAUSA: Il servizio API di Kubernetes non è correttamente avviato o raggiungibile.

DIAGNOSI APPLICATA:

bash

```
# Verifica stato kubelet
sudo systemctl status kubelet

# Verifica porta API server
sudo netstat -tlnp | grep 6443
sudo ss -tlnp | grep 6443

# Test connettività
curl -k https://192.168.0.202:6443

# Verifica configurazione kubectl
kubectl config view
cat ~/.kube/config
```

SOLUZIONI APPLICATE:

Opzione 1 - Riavvio servizi:

bash

```
sudo systemctl restart kubelet
sudo systemctl restart containerd
```

Opzione 2 - Verifica firewall:

bash

```
sudo ufw status
sudo ufw allow 6443  # Se firewall attivo
```

Opzione 3 - Reset e reinizializzazione (se necessario):

bash

```
sudo kubeadm reset -f
sudo rm -rf ~/.kube/
sudo rm -rf /etc/kubernetes/
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=192.1
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Verifica funzionamento prima di installare Flannel:

bash

```
kubectl get nodes
kubectl get pods -n kube-system
```

Installazione Flannel (dopo verifica):

bash

```
kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-f
```

Fase 5: Configurazione Worker Nodes

5.1 Join Command

Il comando `kubeadm init` fornisce automaticamente il comando join:

bash

```
kubeadm join <MASTER-IP>:6443 --token <TOKEN> --discovery-token-ca-cert-hash sha256:<H
```

Esecuzione sui worker nodes:

bash

```
sudo kubeadm join <MASTER-IP>:6443 --token <TOKEN> --discovery-token-ca-cert-hash sha2
```

Rigenerazione comando join (se necessario):

bash

```
# Dal master node
kubeadm token create --print-join-command
```

Fase 6: Verifica e Test del Cluster

6.1 Verifica Stato Cluster

Controllo nodi:

bash

```
kubectl get nodes
```

Output atteso:

```
NAME      STATUS   ROLES           AGE   VERSION
master    Ready    control-plane   10m   v1.28.x
worker1   Ready    <none>          5m    v1.28.x
worker2   Ready    <none>          5m    v1.28.x
```

Controllo pod di sistema:

bash

```
kubectl get pods -n kube-system
```

Informazioni cluster:

bash

```
kubectl cluster-info
```

6.2 Test Deployment

Creazione deployment di test:

bash

```
kubectl create deployment nginx-test --image=nginx:alpine
kubectl scale deployment nginx-test --replicas=3
kubectl expose deployment nginx-test --port=80 --type=NodePort
```

Verifica deployment:

```bash
kubectl get pods -o wide
kubectl get services
```

Test connettività:

```bash
curl http://<NODE-IP>:<NODE-PORT>
```

Configurazioni Aggiuntive e Ottimizzazioni

Ottimizzazioni per Raspberry Pi

Limitazione risorse kubelet:

```bash
sudo nano /etc/default/kubelet
# Aggiungere:
KUBELET_EXTRA_ARGS="--max-pods=50 --kube-reserved=cpu=100m,memory=256Mi --system-reser

sudo systemctl restart kubelet
```

Dashboard Kubernetes (Opzionale)

Installazione:

```bash
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/dep
```

Creazione utente admin:

```bash
kubectl create serviceaccount admin-user -n kubernetes-dashboard
kubectl create clusterrolebinding admin-user --clusterrole=cluster-admin --serviceacco
```

Generazione token accesso:

```bash
kubectl -n kubernetes-dashboard create token admin-user
```

Risoluzione Problemi Comuni

Problemi di Connessione API Server

Errore "connection refused" su porta 6443:

Diagnosi:

```bash
sudo systemctl status kubelet
sudo netstat -tlnp | grep 6443
curl -k https://<MASTER-IP>:6443
kubectl config view
```

Possibili cause e soluzioni:

1. **Kubelet non avviato correttamente:**

```bash
sudo systemctl restart kubelet
sudo systemctl restart containerd
sudo journalctl -u kubelet -f
```

2. **Firewall che blocca la porta:**

```bash
sudo ufw status
sudo ufw allow 6443
```

3. **Inizializzazione fallita - Reset necessario:**

```bash
sudo kubeadm reset -f
sudo rm -rf ~/.kube/ /etc/kubernetes/
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=<MASTI
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

4. **Problemi di configurazione rete:**

bash

```
ip addr show
ip route show
```

Verifica prima di procedere:

bash

```
kubectl get nodes  # Deve funzionare senza errori
kubectl get pods -n kube-system  # Deve mostrare pod di sistema
```

Problemi di Networking

Controllo stato Flannel:

bash

```
kubectl get pods -n kube-flannel
```

Restart Flannel se necessario:

bash

```
kubectl delete pods -n kube-flannel -l app=flannel
```

Problemi containerd

Controllo stato servizio:

bash

```
sudo systemctl status containerd
```

Restart containerd:

bash

```
sudo systemctl restart containerd
```

Reset Completo (Emergenza)

Reset nodo:

bash

```
sudo kubeadm reset
sudo rm -rf ~/.kube/
sudo rm -rf /etc/kubernetes/
```

Monitoraggio e Manutenzione

Comandi Utili per il Monitoraggio

Risorse cluster:

bash

```
kubectl top nodes
kubectl top pods --all-namespaces
```

Logs sistema:

bash

```
kubectl logs -n kube-system <pod-name>
journalctl -u kubelet
```

Descrizione risorse:

bash

```
kubectl describe node <node-name>
kubectl describe pod <pod-name>
```

Backup Configurazione

Backup etcd (Master):

bash

```
sudo cp -r /var/lib/etcd /backup/etcd-$(date +%Y%m%d)
```

Backup configurazioni:

bash

```
sudo cp -r /etc/kubernetes /backup/kubernetes-$(date +%Y%m%d)
```

Conclusioni

Il cluster Kubernetes è stato configurato con successo sui tre Raspberry Pi 5. I principali problemi riscontrati sono stati:

1. **Configurazione repository Kubernetes** - Risolto con la corretta procedura per Ubuntu 24.04.2
2. **Sintassi comando kubeadm init** - Risolto utilizzando comando su singola riga
3. **Configurazione swap** - Risolto con comando sed corretto
4. **Connessione API Server rifiutata** - Risolto con verifica servizi e possibile reset/reinizializzazione del cluster

Il cluster è ora operativo e pronto per il deployment di applicazioni containerizzate.

Metriche del Setup Finale

- **Tempo totale installazione**: ~45 minuti
- **Memoria RAM utilizzata**: ~1.2GB per master, ~800MB per worker
- **Spazio disco utilizzato**: ~3GB per l'installazione base
- **Numero pod massimi**: 110 per nodo (configurabile)

Raccomandazioni per Uso in Produzione

- Implementare monitoraggio con Prometheus/Grafana
- Configurare backup automatici di etcd
- Implementare storage persistente (NFS/Longhorn)
- Configurare ingress controller (NGINX/Traefik)
- Implementare certificate management (cert-manager)