Guida Rapida: Comandi per Setup Cluster Kubernetes

Preparazione Sistema (Tutti i Nodi)

Aggiornamento Sistema

bash

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

Disabilitazione Swap

bash

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

Configurazione Moduli Kernel

bash

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

sudo modprobe overlay
sudo modprobe br_netfilter
```

Configurazione Parametri Rete

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
```

Installazione Container Runtime

Installazione containerd

bash

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

Installazione Kubernetes

Preparazione Repository

bash

```
sudo rm -f /etc/apt/sources.list.d/kubernetes.list
sudo rm -f /etc/apt/keyrings/kubernetes-apt-keyring.gpg
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
```

Configurazione Repository Kubernetes

bash

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmo
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/
```

Installazione Componenti Kubernetes

bash

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

Configurazione Master Node

Identificazione IP

bash

```
ip addr show
```

Inizializzazione Cluster

bash

```
# Sostituire <MASTER-IP> con l'IP reale del master
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=<MAST
```

Configurazione kubectl

bash

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

Verifica Inizializzazione

bash

```
# IMPORTANTE: Verificare che questi comandi funzionino prima di procedere
kubectl get nodes
kubectl get pods -n kube-system
kubectl cluster-info
```

Installazione Network Plugin

bash

```
# Solo dopo aver verificato che kubectl funzioni correttamente
kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-f
```

Configurazione Worker Nodes

Join al Cluster

bash

```
# Utilizzare il comando fornito da kubeadm init
sudo kubeadm join <MASTER-IP>:6443 --token <TOKEN> --discovery-token-ca-cert-hash sha2
```

Rigenerazione Comando Join (se necessario)

bash

```
# Eseguire sul master node
kubeadm token create --print-join-command
```

Verifica e Test

Controllo Stato Cluster

bash

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

Test Deployment

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

Comandi di Gestione Quotidiana

Informazioni Cluster

bash

```
kubectl cluster-info
kubectl get nodes -o wide
kubectl get pods --all-namespaces
kubectl get services --all-namespaces
```

Monitoraggio Risorse

bash

```
kubectl top nodes
kubectl top pods --all-namespaces
kubectl describe node <node-name>
```

Gestione Pod

bash

```
kubectl get pods
kubectl describe pod <pod-name>
kubectl logs <pod-name>
kubectl exec -it <pod-name> -- /bin/bash
kubectl delete pod <pod-name>
```

Gestione Deployment

bash

```
kubectl get deployments
kubectl describe deployment <deployment-name>
kubectl scale deployment <deployment-name> --replicas=<number>
kubectl rollout status deployment/<deployment-name>
kubectl rollout history deployment/<deployment-name>
```

Gestione Servizi

bash

```
kubectl get services
kubectl describe service <service-name>
kubectl expose deployment <deployment-name> --port=<port> --type=NodePort
kubectl delete service <service-name>
```

Comandi di Troubleshooting

Debug Problemi API Server

bash

```bash
# Verifica stato servizi
sudo systemctl status kubelet
sudo systemctl status containerd

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

# Test connettività API server
curl -k https://<MASTER-IP>:6443

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

# Log dettagliati
sudo journalctl -u kubelet -f
sudo journalctl -u containerd -f
```

Risoluzione Errore "Connection Refused"

bash

```bash
# Riavvio servizi
sudo systemctl restart kubelet
sudo systemctl restart containerd

# Verifica firewall
sudo ufw status
sudo ufw allow 6443

# Reset completo se necessario
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
```

Debug Problemi Rete

bash

```
kubectl get pods -n kube-flannel
kubectl describe pod -n kube-flannel <flannel-pod-name>
kubectl logs -n kube-flannel <flannel-pod-name>
```

Verifica Connettività di Rete

bash

```
# Verifica interfacce di rete
ip addr show
ip route show

# Verifica connessioni attive
sudo netstat -tulnp | grep kube
```

Debug Problemi Sistema

bash

```
sudo systemctl status kubelet
sudo systemctl status containerd
journalctl -u kubelet
journalctl -u containerd

# Verifica processi Kubernetes
sudo ps aux | grep kube
```

Procedura di Diagnosi Completa

bash

```bash
# 1. Verifica stato servizi base
sudo systemctl status kubelet containerd

# 2. Verifica che l'API server sia in ascolto
sudo netstat -tlnp | grep 6443

# 3. Test configurazione kubectl
kubectl config view
kubectl version --client

# 4. Se tutto ok, testa connessione
kubectl get nodes

# 5. Solo se tutto funziona, installa network plugin
kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-f
```

Comandi di Manutenzione

Backup

bash

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

# Backup configurazioni
sudo cp -r /etc/kubernetes /backup/kubernetes-$(date +%Y%m%d)
sudo cp ~/.kube/config ~/backup/kubeconfig-$(date +%Y%m%d)
```

Aggiornamento Cluster

bash

```
# Controllo versioni disponibili
kubeadm version
kubectl version

# Aggiornamento kubeadm (master)
sudo apt-mark unhold kubeadm
sudo apt-get update && sudo apt-get install -y kubeadm
sudo apt-mark hold kubeadm

# Piano di aggiornamento
sudo kubeadm upgrade plan
sudo kubeadm upgrade apply v1.28.x
```

Rimozione Nodo

bash

```
# Dal master
kubectl drain <node-name> --ignore-daemonsets --delete-emptydir-data
kubectl delete node <node-name>

# Sul nodo da rimuovere
sudo kubeadm reset
sudo rm -rf ~/.kube/
```

Comandi di Sicurezza

Gestione Token

bash

```
kubeadm token list
kubeadm token create
kubeadm token delete <token>
```

Gestione Certificati

bash

```
sudo kubeadm certs check-expiration
sudo kubeadm certs renew all
```

Comandi Utili per Sviluppo

Port Forwarding

bash

```
kubectl port-forward pod/<pod-name> <local-port>:<pod-port>
kubectl port-forward service/<service-name> <local-port>:<service-port>
```

Copia File

bash

```
kubectl cp <local-file> <pod-name>:<pod-path>
kubectl cp <pod-name>:<pod-path> <local-file>
```

Esecuzione Comandi

bash

```
kubectl exec <pod-name> -- <command>
kubectl exec -it <pod-name> -- /bin/bash
```

Comandi per Namespace

Gestione Namespace

bash

```
kubectl get namespaces
kubectl create namespace <namespace-name>
kubectl delete namespace <namespace-name>
kubectl config set-context --current --namespace=<namespace-name>
```

Reset Completo (Emergenza)

Reset Singolo Nodo

bash

```
sudo kubeadm reset
sudo rm -rf ~/.kube/
sudo rm -rf /etc/kubernetes/
sudo systemctl restart kubelet
sudo systemctl restart containerd
```

Pulizia Completa

bash

```
sudo kubeadm reset --force
sudo rm -rf ~/.kube/ /etc/kubernetes/ /var/lib/etcd/
sudo systemctl restart kubelet containerd
sudo iptables -F && sudo iptables -t nat -F && sudo iptables -t mangle -F && sudo iptal
```

Note Importanti

- Sostituire sempre `<MASTER-IP>` con l'IP reale del master node
- Sostituire `<TOKEN>` e `<HASH>` con i valori reali forniti da kubeadm
- Eseguire i comandi nell'ordine indicato
- Verificare sempre lo stato dopo ogni comando importante
- Fare backup prima di modifiche importanti

Alias Utili

Aggiungi al file `~/.bashrc`:

bash

```
alias k='kubectl'
alias kgp='kubectl get pods'
alias kgs='kubectl get services'
alias kgn='kubectl get nodes'
alias kdp='kubectl describe pod'
alias kds='kubectl describe service'
```