Installation

K8s ??? CLI

Kubectl

K8s ?????

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

# Use snap
sudo snap install kubectl
```

?????

- `~/.kube/config`
- `export KUBECONFIG=/path/to/config`

Helm

K8s ??????

```
# Use snap
sudo snap install helm

# Use curl
curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-4 | bash
```

K3s

- K3s
- Doc: K3s - Lightweight Kubernetes | K3s

- How to Install Kubernetes Using K3s On Ubuntu 24.04

```
curl -sfL https://get.k3s.io | sh -
```

After running this installation:

- The K3s service will be configured to automatically restart after node reboots or if the process crashes or is killed
- Additional utilities will be installed, including kubectl, crictl, ctr, k3s-killall.sh, and k3s-uninstall.sh
- A kubeconfig file will be written to /etc/rancher/k3s/k3s.yaml and the kubectl installed by K3s will automatically use it

```
sudo cp /etc/rancher/k3s/k3s.yaml ~/.kube/config
sudo chown $(id -u):$(id -g) ~/.kube/config
kubectl config view

# Alternatively
sudo cp /etc/rancher/k3s/k3s.yaml ~
sudo chown <your-id> k3s.yaml
export KUBECONFIG=~/k3s.yaml
kubectl config view
```

Install additional agent node

- The value to use for K3S_TOKEN is stored at `/var/lib/rancher/k3s/server/node-token` on your server node.

```
curl -sfL https://get.k3s.io | K3S_URL=https://myserver:6443 K3S_TOKEN=mynodetoken sh -
```

Microk8s

- https://github.com/canonical/microk8s
- Install Microk8s: Ultimate Beginners Configuration Guide - Virtualization Howto

Installation

```
sudo snap install microk8s --classic
```

Verify

```
root@k8s-vm:~# microk8s version
MicroK8s v1.32.9 revision 8511

root@k8s-vm:~# snap list
Name      Version   Rev    Tracking       Publisher   Notes
core20    20251031  2686   latest/stable  canonical✓  base
microk8s  v1.32.9   8511   1.32/stable    canonical✓  classic
snapd     2.73      25935  latest/stable  canonical✓  snapd

root@k8s-vm:~# microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dns                  # (core) CoreDNS
    ha-cluster           # (core) Configure high availability on the current node
    helm                 # (core) Helm - the package manager for Kubernetes
    helm3                # (core) Helm 3 - the package manager for Kubernetes
  disabled:
    cert-manager         # (core) Cloud native certificate management
    cis-hardening        # (core) Apply CIS K8s hardening
    community            # (core) The community addons repository
    dashboard            # (core) The Kubernetes dashboard
    gpu                  # (core) Alias to nvidia add-on
    host-access          # (core) Allow Pods connecting to Host services smoothly
    hostpath-storage     # (core) Storage class; allocates storage from host directory
    ingress              # (core) Ingress controller for external access
    kube-ovn             # (core) An advanced network fabric for Kubernetes
    mayastor             # (core) OpenEBS MayaStor
    metallb              # (core) Loadbalancer for your Kubernetes cluster
    metrics-server       # (core) K8s Metrics Server for API access to service metrics
    minio                # (core) MinIO object storage
    nvidia               # (core) NVIDIA hardware (GPU and network) support
    observability        # (core) A lightweight observability stack for logs, traces and
metrics
    prometheus           # (core) Prometheus operator for monitoring and logging
    rbac                 # (core) Role-Based Access Control for authorisation
```

```
    registry             # (core) Private image registry exposed on localhost:32000
    rook-ceph            # (core) Distributed Ceph storage using Rook
    storage              # (core) Alias to hostpath-storage add-on, deprecated
```

Setting up an alias for microk8s kubectl

```
alias kubectl="microk8s kubectl"
```

To use MicroK8s with your existing kubectl:

```
sudo microk8s kubectl config view --raw > $HOME/.kube/config
```

Microk8s Commands

```
# Start a Microk8s cluster
sudo microk8s start

# Stop a Microk8s cluster
sudo microk8s stop

# Reset your Microk8s cluster
sudo microk8s reset

# Enabling the Microk8s dashboard
sudo microk8s enable dashboard

# Adding a node to your microk8s Kubernetes cluster
microk8s add-node

# Joining a node
microk8s join <your master node and token string>
```

Upgrading your Microk8s cluster

```
# First, drain your Kubernetes node:
microk8s kubectl drain <your node> --ignore-daemonsets --delete-emptydir-data --force

# Upgrade your Microk8s node
sudo snap refresh microk8s --channel=1.32/stable
```

```
# Check the version of your Microk8s nodes:
microk8s.kubectl get no

# Finally, uncordon your node:
microk8s kubectl uncordon <node>
```

Enable addons

- metrics-server : ??????? Lens ????? Overview ???
- hostpath-storage : ??????? PersistentVolume

GPU Support

- Enable GPU Support in Kubernetes: Complete Guide
- ?? LLM ? AI OPS ???? - ???? GPU ? Kubernetes ??-????????-???
- Deploying Llama 3.2 3B in a Kubernetes (K8s) cluster - Lambda Docs
- https://github.com/NVIDIA/k8s-device-plugin
- https://github.com/NVIDIA/gpu-operator

NVIDIA GPU

Installation

```
# nvidia-container-toolkit
sudo apt-get install -y nvidia-container-toolkit

# Deploy NVIDIA Device Plugin
snap install --classic helm
helm repo add nvidia-device-plugin https://nvidia.github.io/k8s-device-plugin
helm repo update
helm upgrade -i nvidia-device-plugin nvidia-device-plugin/nvidia-device-plugin --version
0.15.0 --set runtimeClassName=nvidia --namespace kube-system
helm upgrade -i nvidia-device-discovery nvidia-device-plugin/gpu-feature-discovery --version
0.15.0 --namespace gpu-feature-discovery --create-namespace --set runtimeClassName=nvidia
```

K8s Cloud Providers

- GCP: GKE
- AWS: EKS
- Azure: AKS

- kOps - k8s ??????????? GKE/EKS/AKS?
  - GitHub: https://github.com/kubernetes/kops
  - [Day 15] ?? kops - ? AWS ??? Kubernetes Cluster (?) - iT ???::???????????? IT ????

K8s

- Kubernetes Home Lab Setup Step-by-Step - Virtualization Howto
- 30??????? kubernetes :: ? 11 ? iThome ???

Revision #34
Created 2025-12-04 15:25:07 CST by A-Lang (Admin)
Updated 2026-02-03 17:29:50 CST by A-Lang (Admin)