

Instalasi K3S Cluster di Ubuntu 22.04

DISCLAIMER:

Disini saya telah melakukan instalasi dan config k3s cluster sebelumnya pada server kantor saya dengan OS ubuntu

Pertama kita siapkan terlebih dahulu server kita

10.101.110.10 => k3s master

10.101.110.11 => worker 1

10.101.110.12 => worker 2

Instalasi K3S Master

Kita akan melakukan instalasi k3s master terlebih dahulu pada vm kita Login pada vps yang ingin kita jadikan master

```
" ssh root@10.101.110.10 "
```

Kemudian kita pastikan update terlebih dahulu repository ubuntu kita dengan menjalankan perintah

```
" sudo apt update "
```

Kemudian kita download k3s pada server kita dan jalankan instalasinya

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

Setelah selesai kita akan melihat status k3s kita apakah running atau tidak dengan menjalankan

```
" systemctl status k3s "
```

```
root@k3s-master:~# systemctl status k3s
k3s.service - Lightweight Kubernetes
Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor preset: enabled)
Active: active (running) since Tue 2023-11-07 21:38:00 WIS; 6 days ago
Docs: https://k3s.io
Process: 633 ExecStartPre=/bin/sh -xc ' /usr/bin/systemctl is-enabled --quiet ne-cloud-setup.service (code=exited, status=0/SUCCESS)
Process: 609 ExecStartPre=/bin/modprobe br netfilter (code=exited, status=0/SUCCESS)
Process: 609 ExecStartPre=/bin/modprobe overlay (code=exited, status=0/SUCCESS)
Main PID: 673 (k3s-server)
Tasks: 68
Memory: 1.1G
CPU: 12h 56min 33.162s
CGroup: /system.slice/k3s.service
└─ 673 /usr/local/bin/k3s server
├─ 674 Containerd
├─ 1308 /var/lib/rancher/k3s/data/3d6c958b39d2e2b435291abdc1333aa051cf46325aee898819f3b9940b21/bin/containerd-shim-runc-v2 -namespace k3s.io -id 9e10f6ed275869123cd5d172170fde3d0a9b456c97b9b08644f08097215c -address
└─ 1309 /var/lib/rancher/k3s/data/3d6c958b39d2e2b435291abdc1333aa051cf46325aee898819f3b9940b21/bin/containerd-shim-runc-v2 -namespace k3s.io -id c06c7fab802b150998534f24f4483543a1f28b597adb52b27494c51f31783 -address
ov 14 19:53:07 kube-master k3s[673]: 11114 19:53:07.674183 673 handler.go:232 Adding GroupVersion cert-manager.io v1 to ResourceManager
ov 14 19:53:07 kube-master k3s[673]: 11114 19:53:07.687009 673 handler.go:232 Adding GroupVersion kube.metrics.k8s.io v1beta1 to ResourceManager
ov 14 19:53:07 kube-master k3s[673]: 11114 19:53:07.691218 673 handler.go:232 Adding GroupVersion metallb.io v1beta1 to ResourceManager
ov 14 19:53:19 kube-master k3s[673]: 11114 19:53:19.555131 673 handler.go:232 Adding GroupVersion metrics.k8s.io v1beta1 to ResourceManager
ov 14 19:53:21 kube-master k3s[673]: 11114 19:53:21.706068 673 horizontal.go:274 failed to compute desired number of replicas based on listed metrics for Deployment/apigateway/klikapin: invalid metrics (1 invalid out of 1), fir
ov 14 19:53:21 kube-master k3s[673]: 11114 19:53:21.701129 673 event.go:307 "Event occurred" object="apigateway/klikapin" fieldPath="" kind="HorizontalPodAutoscaler" apiVersion="autoscaling/v2" type="Warning" reason="FailedDetai
ov 14 19:53:21 kube-master k3s[673]: 11114 19:53:21.701333 673 event.go:307 "Event occurred" object="apigateway/klikapin" fieldPath="" kind="HorizontalPodAutoscaler" apiVersion="autoscaling/v2" type="Warning" reason="FailedDetai
ov 14 19:53:36 kube-master k3s[673]: 11114 19:53:36.718526 673 horizontal.go:274 failed to compute desired number of replicas based on listed metrics for Deployment/apigateway/klikapin: invalid metrics (1 invalid out of 1), fir
ov 14 19:53:36 kube-master k3s[673]: 11114 19:53:36.718822 673 event.go:307 "Event occurred" object="apigateway/klikapin" fieldPath="" kind="HorizontalPodAutoscaler" apiVersion="autoscaling/v2" type="Warning" reason="FailedDetai
ov 14 19:53:36 kube-master k3s[673]: 11114 19:53:36.718841 673 event.go:307 "Event occurred" object="apigateway/klikapin" fieldPath="" kind="HorizontalPodAutoscaler" apiVersion="autoscaling/v2" type="Warning" reason="FailedDetai
Des 1-27/12 (Info)
```

Karen kita mau menjadikan ini master maka kita butuh workernya, untuk melakukan koneksi k3s node worker ke master maka kita butuh TOKEN dari master.

TOKEN ini bisa kita lihat pada file

```
/var/lib/rancher/k3s/server/node-token
```

```
"
```

```
K108eb3391a37d0df451a1875d32a0b25af235b31872770cb48bb75df6f
fe20f61e::server:a2xpa2xhYnNAa3ViZXJuZXRlcmw=" "
```

```
K108eb3391a37d0df451a1875d32a0b25af235b31872770cb48bb75df6ffe20f61e::server:a2xpa2xhYnNAa3ViZXJuZXRlcmo=
/var/lib/rancher/k3s/server/node-token (END)
```

Setelah selesai instalasi kita perlu copy file kubeconfig dengan menjalankan perintah

```
sudo cp /etc/rancher/k3s/k3s.yaml $HOME/.kube/config
```

Kita sudah berhasil menjalankan k3s cluster pada server kita, kita bisa melakukan pengecekan node kita

```
root@kube-master:~# kubectl get node
NAME                STATUS    ROLES                  AGE      VERSION
kube-master         Ready    control-plane,master   35d      v1.27.6+k3s1
kube-worker-4       Ready    <none>                 35d      v1.27.6+k3s1
kube-worker-1       Ready    <none>                 4d20h    v1.27.6+k3s1
kube-worker-8       Ready    <none>                 35d      v1.27.6+k3s1
kube-worker-9       Ready    <none>                 24d      v1.27.6+k3s1
kube-worker-6       Ready    <none>                 35d      v1.27.6+k3s1
kube-worker-3       Ready    <none>                 35d      v1.27.6+k3s1
kube-worker-2       Ready    <none>                 4d20h    v1.27.6+k3s1
kube-worker-7       Ready    <none>                 35d      v1.27.6+k3s1
kube-worker-5       Ready    <none>                 35d      v1.27.6+k3s1
root@kube-master:~#
```

Jika sudah maka kita perlu melakukan instalasi node worker kita

Instalasi K3S Node Worker

Yang pertama kita lakukan adalah sama dengan akses pada server kita dan lakukan update repository.

Karena disini kita akan menjadikan worker maka pada saat instalasi k3s kita perlu menambahkan token dan ip node master kita tadi.

```
curl -sL https://get.k3s.io |
K3S_URL=https://10.101.110.11:6443
K3S_TOKEN=K108eb3391a37d0df451a1875d32a0b25af235b31872770cb
48bb75df6ffe20f61e::server:a2xpa2xhYnNAa3ViZXJuZXRlcmo= sh
-
```

Begitu juga pada server node ke-3 kita lakukan hal yang sama

Seperti screen Shoot diatas bahwa semua node worker dalam status ready, nah kita bisa melakukan deploy apps service kita pada masing masing node kita tersebut.