

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 terlabih 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 “
```

[illegible]

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.

Deploy vote-app

Setelah kita menyiapkan cluster kita, sekarang kita saatnya melakukan deployment pada cluster kita.

NB:

Untuk Environment saat ini saya jadikan config-map

Deploy configmap

Kita buat file config-map pada folder kita dengan format yml file. setelah selesai kita deploy ke kubernetes kita dengan menjalankan perintah

```
Kubectl apply -f config-map.yml -n tlab
```

```
beny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$ kubectl get cm -n tlab
NAME          DATA  AGE
kube-root-ca.crt  1      5m50s
postgres-config  2      5m37s
vote-app-config  4      5m29s
beny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$
```

Deploy APP

Untuk deploy app kita ke kubernetes sama kita bikin file deployment yml dan deploy

```
beny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$ kubectl get pod -n tlab
NAME                                READY   STATUS    RESTARTS   AGE
edis-67bbd49d85-6fkxt               1/1     Running   0           41m
edis01-f87d7b5df-nwnxq              1/1     Running   0           58m
results-app-6d57dc876-xl6ff         0/1     CrashLoopBackOff   8 (119s ago)    58m
tore-5c78d8c557-nx6ks               1/1     Running   0           6s
lab-result-64d6746dfb-d7rlm         0/1     CrashLoopBackOff   5 (106s ago)    5m10s
lab-vote-6845b84c55-rcbmh           1/1     Running   0           5m3s
lab-web-6b655d6445-l2jr2            1/1     Running   0           4m57s
ote-worker-55666f59bf-s2b22         1/1     Running   0           23m
eb-vote-app-c8ccffbc-rccd6          1/1     Running   0           30m
beny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$
```

NB:

Disini untuk vote-result ada menemukan error pada pod kubernetesnya, dikarenakan setelah saya cek log ada error di code node_modulesnya

```
eny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$ kubectl logs tlab-result-64d6746dfb-d7rlm -n tlab
node_modules/socket.io/node_modules/socket.io-client/node_modules/engine.io-client/node_modules/xmlhttprequest-ssl/lib/XMLHttpRequest.js:6
    for (let i = 0, len = listeners[event].length; i < len; i++) {
    ^^^
SyntaxError: Unexpected strict mode reserved word
    at Module.compile (module.js:439:25)
    at Object.Module._extensions..js (module.js:474:10)
    at Module.load (module.js:356:32)
    at Function.Module._load (module.js:312:12)
    at Module.require (module.js:364:17)
    at require (module.js:380:17)
    at Object.<anonymous> (/node_modules/socket.io/node_modules/socket.io-client/node_modules/engine.io-client/lib/transports/index.js:5:22)
    at Module.compile (module.js:456:26)
    at Object.Module._extensions..js (module.js:474:10)
    at Module.load (module.js:356:32)
eny@sakubeny-pc:~/Test/swarm-microservice-demo-v1/kubernetes$ kubectl describe pod results-app-6d57dc876-xl6ff -n tlab
Name:
Namespace:
Priority:
Service Account:
Node:
Start Time:
Labels:
Annotations:
Status:
IP:
Phase:
```