**Tugas Day 15**

Instalasi dan Konfigurasi Kubernetes (K8)

Master dan Worker pada Sistem Operasi CentOS

**Intro**

Kubernetes (K8) merupakan platform open-source yang berfungsi sebagai aplikasi yang dapat memenuhi kebutuhan akan *automation deployment*, *scaling*, dan manajemen berbasis *container*.

**Skenario**

Pada skenario kali ini, kita diminta untuk membuat dua buah *virtual machine* yang menjalankan aplikasi kubernetes. Kemudian, lakukan konfigurasi sedemikian sehingga *VMs* tersebut dapat terhubung dan berperan sebagai **kubemaster** dan **kubeworker**

**Spesifikasi**

Berikut detail spesifikasi dari VMs yang dijalankan

|  |
| --- |
| # Kubemaster  OS : CentOS 7 Server  IP : 10.8.0.2/24  Hostname : master.kube  # Kubeworker  OS : CentOS 7 Server  IP : 192.168.226.143/24  Hostname : node1.kube |

**Pengaturan Host [Kubemaster & Kubeworker]**

Pada server kubemaster, lakukan konfigurasi sebagai berikut

|  |
| --- |
| $ sudo hostnamectl set-hostname master.kube  $ sudo vim /etc/hosts |
| 127.0.0.1 localhost master.kube  ::1 localhost master.kube  10.8.0.2 master.kube  192.168.226.143 node1.kube |

Pada server kubeworker, lakukan konfigurasi sebagai berikut

|  |
| --- |
| $ sudo hostnamectl set-hostname node1.kube  $ sudo vim /etc/hosts |
| 127.0.0.1 localhost node1.kube  ::1 localhost node1.kube  10.8.0.2 master.kube  192.168.226.143 node1.kube |

**Persiapan Environment [Kubemaster & Kubeworker]**

Terlebih dahulu kita lakukan persiapan pada kedua VMs dengan menginstal beberapa *dependencies* yang diperlukan untuk menjalankan kubernetes

1. **Instalasi Docker**

Pertama-tama, lakukan instalasi dan jalankan layanan docker yang berfungsi sebagai *container manager* bagi kubernetes.

|  |
| --- |
| $ curl -fsSL https://get.docker.com -o get-docker.sh  $ sudo sh get-docker.sh  $ sudo systemctl enable docker  $ sudo systemctl start docker |

**b. Instalasi Kubernetes**

Kemudian, tambahkan repository kubernetes ke dalam yum repository

|  |
| --- |
| $ sudo vim /etc/yum.repos.d/kubernetes.repo |
| vim /etc/yum.repos.d/kubernetes.repo  [kubernetes]  name=Kubernetes  baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64  enabled=1  gpgcheck=1  repo\_gpgcheck=1  gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg  https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg |

Barulah, lakukan instalasi komponen serta jalankan layanan kubernetes sebagai berikut

|  |
| --- |
| $ sudo yum install -y kubelet kubeadm kubectl  $ sudo systemctl enable kubelet  $ sudo systemctl start kubelet |

**c. Konfigurasi Firewall**

Untuk dapat berkomunikasi dengan kluster-kluster lainnya, kita bisa menonaktifkan firewall yang sedang berjalan

|  |
| --- |
| $ sudo systemctl stop firewalld |

**d. Konfigurasi Swap**

Untuk mengoptimalkan kinerja dari layanan *kubelet*, nonaktifkan memori swap pada sistem operasi. Perlu diketahui bahwa pada versi kubernetes 1.18 ke atas, hal ini bersifat *mandatory*.

|  |
| --- |
| $ sudo swapoff -a |

**e. Konfigurasi SElinux**

Untuk mengizinkan container agar dapat mengakses *file system* pada host, nonaktifkan selinux dengan mengubahnya dari mode *enforcing* ke *disabled*

|  |
| --- |
| $ sudo setenforce 0 |

**e. Konfigurasi Iptables**

Untuk memastikan bahwa trafik dapat diproses selama *packet filtering* dan *port* *forwarding* berlangsung, lakukan konfigurasi sebagai berikut

|  |
| --- |
| $ sudo vim /etc/sysctl.conf |
| net.bridge.bridge-nf-call-ip6tables = 1  net.bridge.bridge-nf-call-iptables = 1 |

Terakhir *reload* config dengan perintah berikut ini

|  |
| --- |
| $ modprobe br\_netfilter  $ sudo sysctl -p |

**Inisiasi Kubernetes Cluster [Kubemaster]**

Untuk menginisiasi kubernetes cluster, kita dapat menjalankan perintah **kubeadm** dengan memasukkan *server ip* pada *apiserver-advertise-address* dan *cluster ip* pada *pod-network-cidr*

|  |
| --- |
| $ sudo kubeadm init --apiserver-advertise-address **10.8.0.2** --pod-network-cidr=**192.168.0.0/16** |
|  |

Kemudian, simpan *token* yang diperoleh. Token ini nantinya akan digunakan oleh **kubeworker** untuk masuk ke dalam kubernetes cluster

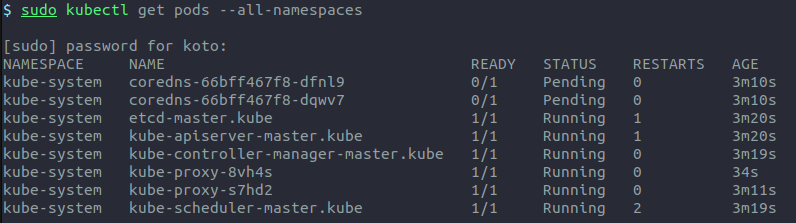
**Konfigurasi Admin Cluster [Kubemaster]**

Untuk mengelola kubernetes cluster dengan *normal user*, lakukan perintah sebagai berikut

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

**Konfigurasi Network Cluster dengan Flannel [Kubemaster]**

Pada tahap ini, kita akan melihat bahwa beberapa pods pada *kube-system* memberikan status *pending*.





Hal ini dapat diselesaikan dengan cara membuat *virtual network* yang memberikan tiap host sebuah subnet agar dapat berkomunikasi dengan *container runtimes*. Terlebih dahulu kita unduh YAML file

|  |
| --- |
| $ wget https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml |

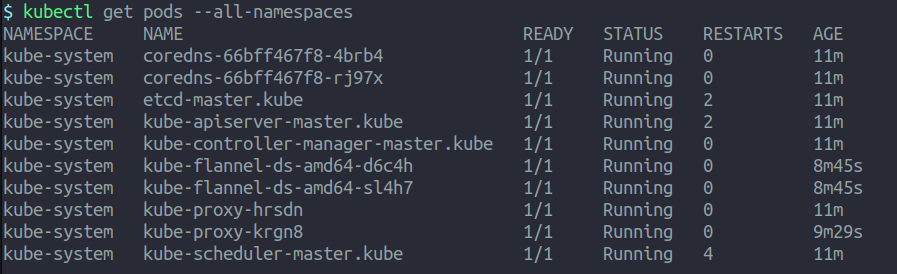
Kemudian kita atur ***flannel network*** agar sesuai dengan ***pod-network-cidr*** yang telah kita definisikan sebelumnya.

|  |
| --- |
| $ vim kube-flannel.yml |
| ---  kind: ConfigMap  apiVersion: v1  metadata:  name: kube-flannel-cfg  namespace: kube-system  labels:  tier: node  app: flannel  data:  cni-conf.json: |  {  "name": "cbr0",  "cniVersion": "0.3.1",  "plugins": [  {  "type": "flannel",  "delegate": {  "hairpinMode": true,  "isDefaultGateway": true  }  },  {  "type": "portmap",  "capabilities": {  "portMappings": true  }  }  ]  }  net-conf.json: |  {  "Network": "**192.168.0.0/16**",  "Backend": {  "Type": "vxlan"  }  }  --- |

Terakhir lakukan deployment terhadap layanan kube-flannel dengan perintah

|  |
| --- |
| $ kubectl apply -f kube-flannel.yml |

Hasilnya, kita dapat melihat bahwa semua pods telah berjalan dan status **kubemaster** berubah menjadi *ready*

****

**Memasukkan node ke dalam Kubernetes Cluster [Kubeworker]**

Untuk memasukkan node sebagai worker dari kubernetes cluster, kita dapat menggunakan token yang diperoleh pada proses **kubeadm** sebelumnya

|  |
| --- |
| $ sudo kubeadm join 10.8.0.2:6443 --token kn6azi.st0xvqd7kaafvxwb --discovery-token-ca-cert-hash sha256:325e4ebceb1094efc9dd23859df5f04553cf8cb182e4dec53ef7384567b029c1 |
|  |

**Mengecek List node pada Kubernetes Cluster [Kubemaster]**

Untuk memastikan apakah **kubeworker** telah bergabung ke dalam kubernetes cluster, kita dapat mengeceknya dengan perintah **kubectl**

|  |
| --- |
| $ kubectl get nodes |
|  |

Hasilnya, terlihat bahwa host **node1** telah masuk sebagai **kubeworker** bagi kubernetes cluster.

**Manajemen dengan Kubernetes Dashboard**

**a. Menjalankan Instance Kubernetes Dashboard**

Dalam mengelola layanan kubernetes, pengguna dapat memanfaatkan layanan **Kubernetes Dashboard** dengan antar muka web. Terlebih dahulu kita unduh konfigurasi YAML seperti berikut ini

|  |
| --- |
| $ wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-beta4/aio/deploy/recommended.yaml |

Kemudian, kita lakukan sedikit perubahan pada YAML file untuk mengekspos **nodePort** pada service **kubernetes-dashboard**.

|  |
| --- |
| $ vim recommended.yaml |
| ---  kind: Service  apiVersion: v1  metadata:  labels:  k8s-app: kubernetes-dashboard  name: kubernetes-dashboard  namespace: kubernetes-dashboard  spec:  **type: NodePort**  ports:  - port: 443  **nodePort: 30080**  targetPort: 8443  selector:  k8s-app: kubernetes-dashboard  --- |

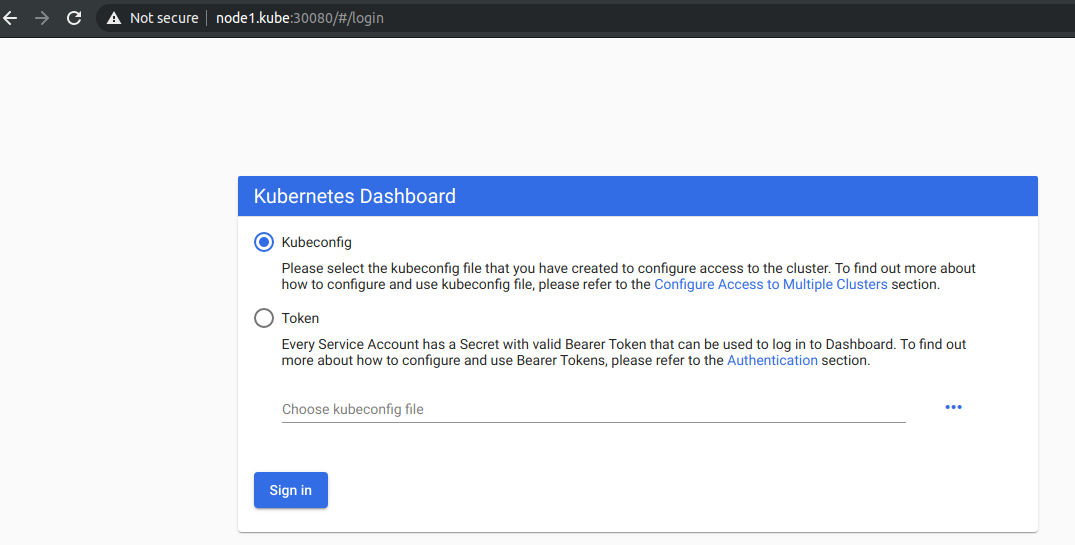
Setelah itu, lakukan deployment dengan perintah sebagai berikut

|  |
| --- |
| $ kubectl apply -f recommended.yaml |
| namespace "kubernetes-dashboard" deleted  serviceaccount "kubernetes-dashboard" deleted  service "kubernetes-dashboard" deleted  secret "kubernetes-dashboard-certs" deleted  secret "kubernetes-dashboard-csrf" deleted  secret "kubernetes-dashboard-key-holder" deleted  configmap "kubernetes-dashboard-settings" deleted  role.rbac.authorization.k8s.io "kubernetes-dashboard" deleted  clusterrole.rbac.authorization.k8s.io "kubernetes-dashboard" deleted  rolebinding.rbac.authorization.k8s.io "kubernetes-dashboard" deleted  clusterrolebinding.rbac.authorization.k8s.io "kubernetes-dashboard" deleted  deployment.apps "kubernetes-dashboard" deleted  service "dashboard-metrics-scraper" deleted  deployment.apps "dashboard-metrics-scraper" deleted |

Apabila proses berjalan dengan benar, kita dapat melihat dua service baru dengan *namespace* **kubernetes-dashboard**

|  |
| --- |
| $ kubectl get service -n kubernetes-dashboard |
|  |

Dari sini, kita dapat mengetahui bahwa layanan dijalankan oleh **kubeworker** pada port 30080. Untuk mengakses layanan, kita dapat menggunakan URL <https://node1.kube:30080>



**b. Konfigurasi Admin User pada Kubernetes Dashboard**

Untuk dapat mengakses dashboard dengan *admin privilege,* lakukan konfigurasi sebagai berikut

|  |
| --- |
| $ vim dashboard-adminuser.yaml |
| apiVersion: v1  kind: ServiceAccount  metadata:  name: admin-user  namespace: kubernetes-dashboard  ---  apiVersion: rbac.authorization.k8s.io/v1  kind: ClusterRoleBinding  metadata:  name: admin-user  roleRef:  apiGroup: rbac.authorization.k8s.io  kind: ClusterRole  name: cluster-admin  subjects:  - kind: ServiceAccount  name: admin-user  namespace: kubernetes-dashboard |

Kemudian apply konfigurasi tersebut dengan perintah berikut

|  |
| --- |
| $ kubectl apply -f dashboard-adminuser.yaml |
| serviceaccount/admin-user created  clusterrolebinding.rbac.authorization.k8s.io/admin-user created |

Terakhir ambil Bearer-Token untuk **admin-user** dan gunakan token untuk melakukan otentikasi

|  |
| --- |
| $ kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep admin-user | awk '{print $1}') | grep -oP '(?<=token:\s{6}).\*' |
|  |

