Kubernetes Installation Guide and dashboard configuration

Title	Kubernetes Installation & dashboard configuration
Version	1.0
Author	Mohammad Delshad
Change record	-

Contents

Disable swap on all the nodes:

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

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-zqwB0xjG8ciLbK63SNNainAp3HiKWO4NH3DPreAFvmYke71LFk5OIwtgt9xe3Agg / ext4 defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/59aa8b42-81ba-4b96-950e-91682e5bf6e9 /boot ext4 defaults 0 1
#swap.img       none    swap    sw      0       0
~
```

Installing a container runtime:

- `sudo apt update`

- `sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release -y`

- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg`

Add docker to your install repository then update your repository:

- `echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`

- `apt-get update`

Install Docker:

- `sudo apt update`
- `sudo apt install docker-ce docker-ce-cli containerd.io -y`

Enable CGroup driver then start Docker:

-
```
cat << EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
```

- `sudo systemctl enable docker`
- `sudo systemctl daemon-reload`
- `sudo systemctl restart docker`

```
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

configuration of some network modules & install containerd:

- `sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/g' /etc/sysctl.conf`
- `sysctl --system; sysctl -a | grep net.ipv4.ip_forward`
- `apt update; apt install apt-transport-https ca-certificates curl conntrack -y`
- ```
  cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
  overlay
  br_netfilter
  EOF
  ```
- `sudo modprobe overlay`
- `sudo modprobe br_netfilter`
- ```
  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`
- `sudo apt-get install -y containerd.io`
- `sudo mkdir -p /etc/containerd`
- `sudo containerd config default | sudo tee /etc/containerd/config.toml`
- `sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/containerd/config.toml`
- `systemctl restart containerd`

Add Kubernetes to your repository list then update your repository:

- `echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list`

- `apt-get update`

Install kubadm & kunctl & kubelet and enable it:

- `sudo apt-get install -y kubelet kubeadm kubectl`

- `sudo apt-mark hold kubelet kubeadm kubectl`

- `sudo systemctl enable –now kubelet`

Kubeadm Initiation & Flannel installation (only on Master):

- `sudo kubeadm init --pod-network-cidr=10.244.0.0/16`

- `mkdir -p $HOME/.kube`

- `sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`

- `sudo chown $(id -u):$(id -g) $HOME/.kube/config`

- `kubectl create ns kube-flannel`

- `kubectl label --overwrite ns kube-flannel pod-security.kubernetes.io/enforce=privileged`

- `helm repo add flannel` https://flannel-io.github.io/flannel/

- `helm install flannel --set podCidr="10.244.0.0/16" --namespace kube-flannel flannel/flannel`

To check pods:

- `kubectl get pods --all-namespaces`
- `kubectl get pods -A`

```
root@buntus02:~# kubectl get pods -A
NAMESPACE      NAME                                READY   STATUS    RESTARTS      AGE
kube-flannel   kube-flannel-ds-tpznq               1/1     Running   0             16m
kube-system    coredns-76f75df574-5sxm5            1/1     Running   0             41m
kube-system    coredns-76f75df574-xzmdh            1/1     Running   0             41m
kube-system    etcd-buntus02                       1/1     Running   1 (23m ago)   42m
kube-system    kube-apiserver-buntus02             1/1     Running   1 (23m ago)   42m
kube-system    kube-controller-manager-buntus02    1/1     Running   1 (23m ago)   42m
kube-system    kube-proxy-4wbj8                    1/1     Running   1 (23m ago)   41m
kube-system    kube-scheduler-buntus02             1/1     Running   1 (23m ago)   42m
```

Generate join token on master node and run the result on worker nodes to join mater and workers:

- `kubeadm token create --print-join-command`

```
root@buntus02:/etc/kubernetes#
root@buntus02:/etc/kubernetes#
root@buntus02:/etc/kubernetes# kubeadm token create --print-join-command
kubeadm join 10.0.2.15:6443 --token lpsb3p.kibag6eoydzzi6mv --discovery-token-ca-cert-hash sha256:ef652993
63dc
```

Installing Kubernetes dashboard:

Adding dashboard:

- `kubectl taint node buntus02 node-role.kubernetes.io/control-plane:NoSchedule --overwrite=true`
- `kubectl taint node buntus02 node-role.kubernetes.io/control-plane:NoSchedule-`
- `helm repo add kubernetes-dashboard https://kubernetes.github.io/dashboard/`
- `helm upgrade --install kubernetes-dashboard kubernetes-dashboard/kubernetes-dashboard --create-namespace --namespace kubernetes-dashboard`

- `kubectl -n kubernetes-dashboard port-forward svc/kubernetes-dashboard-kong-proxy 8443:443`

- https://localhost:8443

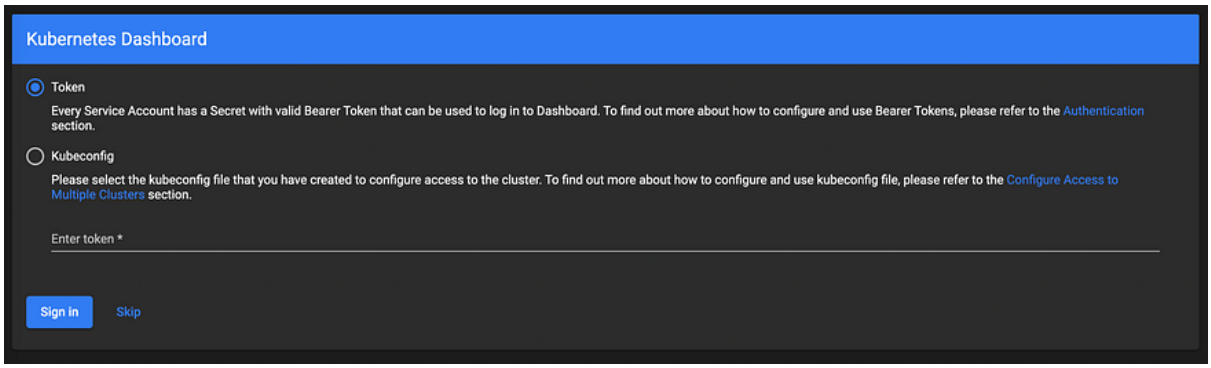
Accessing dashboard:

- `mkdir ~/dashboard && cd ~/dashboard`

- `vi dashboard-admin.yaml`

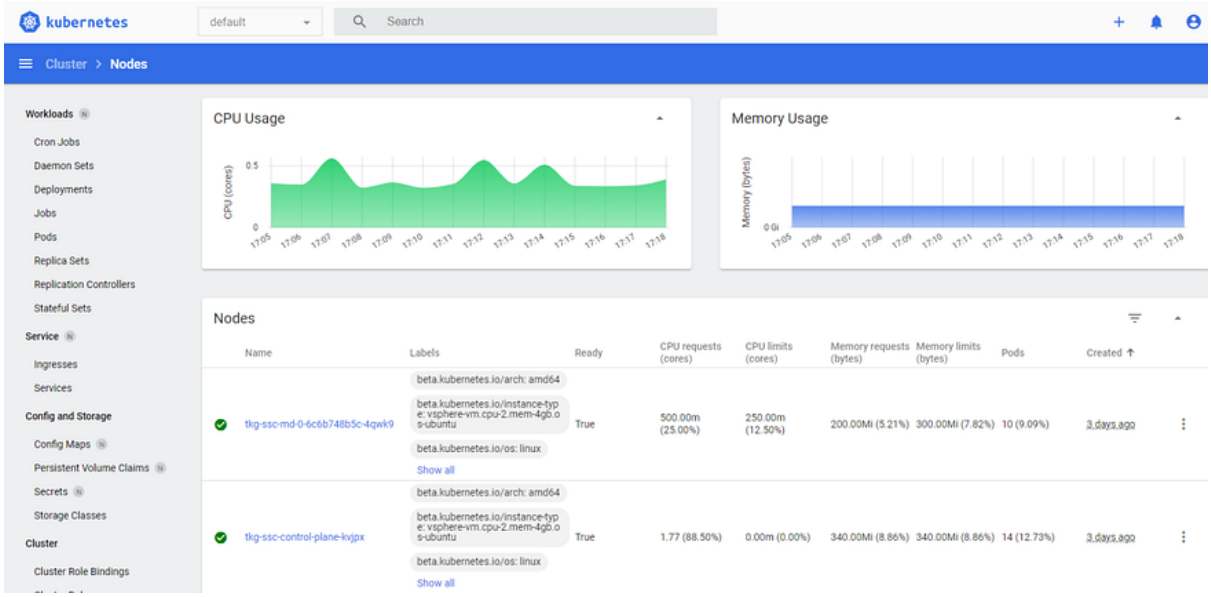
```
#yaml File of admin:
      ---
      apiVersion: v1
      kind: ServiceAccount
      metadata:
        name: admin-user
        namespace: kubernetes-dashboard

      ---
      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

      ---
      apiVersion: v1
      kind: Secret
      metadata:
        name: admin-user-secret
        annotations:
          kubernetes.io/service-account.name: admin-user
      type: kubernetes.io/service-account-token
```

- `kubectl apply -f dashboard-admin.yaml`



Dashboard Token Generation:

- `kubectl -n kubernetes-dashboard get secret admin-user-secret -o jsonpath="{.data.token}" | base64 -d`



Installing Helm:

- `curl https://baltocdn.com/helm/signing.asc | sudo apt-key add –`
- `sudo apt install apt-transport-https -y`
- `echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list`
- `sudo apt update; sudo apt install helm`
- `helm repo add bitnami https://charts.bitnami.com/bitnami`
- `helm repo update`

Thanks for your attention - Best Regards – By Mohammad Delshad