

*System Admin*

*Training Assignments*

|  |  |
| --- | --- |
| **Program Code** |  |
| **Issue/Revision** | **x/y** |
| **Effective date** | **04/Aug /2023** |

**Assignment: Kubernetes Cluster Installation**

**Part 1: Pre requirement**

* 01 Server Ubuntu 22.04 has server type: t3.small (2Gb RAM, 2vCPU) and 20Gb storage, has public IP, named this server “k8s\_master”.
* 01 Server Ubuntu 22.04 has server type: t2.micro (1Gb RAM, 1vCPU) and 20Gb storage, has public IP, named this server “k8s\_worker”.
* Security Group that allow port 22, 80, 443, 8080, 30000-32767
* Run below command in “Advanced details > User data” when you launch new EC2:

*#!/bin/bash*

*apt-get update*

*apt-get upgrade -y*

*mkdir -p -m 755 /etc/apt/keyrings*

*# Change the host name depend on the server name was set above*

*hostnamectl set-hostname k8s\_master*

**Part 2: Install containerd service (On both Master and Worker servers)**

1. Disable swap

*$ sudo swapoff -a*

*$ sudo sed -i '/ swap / s/^\(.\*\)$/#\1/g' /etc/fstab*Install docker.

1. Load the required kernel modules.

*$ sudo tee /etc/modules-load.d/containerd.conf <<EOF*

*overlay*

*br\_netfilter*

*EOF*

*$ sudo modprobe overlay*

*$ sudo modprobe br\_netfilter*

1. Configure the critical kernel parameters for Kubernetes

*$ sudo tee /etc/sysctl.d/kubernetes.conf <<EOF*

*net.bridge.bridge-nf-call-ip6tables = 1*

*net.bridge.bridge-nf-call-iptables = 1*

*net.ipv4.ip\_forward = 1*

*EOF*

1. Reload to apply the changes

*$ sudo sysctl --system*

1. Install Containerd runtime dependencies

*$ sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates*

1. Add Docker repository, and install containerd:

*$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg*

*$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"*

*$ sudo apt update*

*$ sudo apt install -y containerd.io*

1. Configure containerd to start using systemd as cgroup

*$ containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1*

*$ sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml*

1. Restart and enable containerd service

*$ sudo systemctl restart containerd*

*$ sudo systemctl enable containerd*

**Part 3: Install Kubernetes (On both Master and Worker servers)**

1. Add repository for K8s. Repository link for K8s may change depend on the provider, pls check in K8s home page for correct link:   
   https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

*$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg*

*$ 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*

*$ ll /etc/apt/sources.list.d/*

*$ cat /etc/apt/sources.list.d/*

*$ sudo apt-get update*

1. Install kubelet, kubeadm and kubectl

*$ sudo apt-get install -y kubelet kubeadm kubectl*

*$ sudo apt-mark hold kubelet kubeadm kubectl*

**Part 4: K8s\_Master**

1. Initialize Kubernetes Cluster with Kubeadm.

*$ sudo kubeadm init # If you success, there will be annouced as below*

*….*

*Alternatively, if you are the root user, you can run:*

*export KUBECONFIG=/etc/kubernetes/admin.conf*

*You should now deploy a pod network to the cluster.*

*Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:*

*https://kubernetes.io/docs/concepts/cluster-administration/addons/*

*Then you can join any number of worker nodes by running the following on each as root:*

*kubeadm join 10.0.0.24:6443 --token isuvgz.iqg25miqzamsbgyb \*

*--discovery-token-ca-cert-hash sha256:04fb6e85d5825f0fa47a4613c0f37287654de22945c27da8e6cb0c98f57e7f32*

*$ mkdir -p $HOME/.kube*

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

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

1. Install K8s Network Plugin (calico)

*$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml*

1. Check cluster and node status

*$ kubectl get pods -n kube-system*

*$ kubectl get nodes*

**Part 5: K8s\_Worker**

1. Add Worker Nodes to the Cluster

*$ sudo kubeadm join 10.0.0.24:6443 --token isuvgz.iqg25miqzamsbgyb \*

*--discovery-token-ca-cert-hash sha256:04fb6e85d5825f0fa47a4613c0f37287654de22945c27da8e6cb0c98f57e7f32*

*# If you success, there will be annouced as below*

….

*This node has joined the cluster:*

*\* Certificate signing request was sent to apiserver and a response was received.*

*\* The Kubelet was informed of the new secure connection details.*

*Run 'kubectl get nodes' on the control-plane to see this node join the cluster.*

**Part 6: Checking K8s Cluster**

1. Check Nodes status on **K8s\_Master**

*$ kubectl get nodes*

*NAME STATUS ROLES AGE VERSION*

*k8smaster Ready control-plane 28m v1.29.3*

*k8sworker Ready <none> 22s v1.29.3*

1. Run a deploy test application on cluster (Run on Master node)

*$ kubectl run nginx --image=nginx*