Set up the new vm with ubuntu on it

Sudo apt update

**Install docker using docker documentation**

1. sudo install -m 0755 -d /etc/apt/keyrings
2. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
3. sudo chmod a+r /etc/apt/keyrings/docker.gpg

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

1. sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
2. sudo apt-get update
3. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

**golang**

1. wget <https://go.dev/dl/go1.21.4.linux-amd64.tar.gz>
2. sudo rm -rf /usr/local/go && sudo tar -C /usr/local -xzf go1.21.4.linux-amd64.tar.gz
3. export PATH=$PATH:/usr/local/go/bin
4. open the .bashrc file and add the “export PATH=$PATH:/usr/local/go/bin”line at the end

download cri-dockerd(<https://github.com/mialeevs/kubernetes_installation_docker>)

**To set up the post docker installation**

git clone <https://github.com/Mirantis/cri-dockerd.git>

1. cd cri-dockerd
2. mkdir bin
3. mkdir -p /usr/local/bin
4. install -o root -g root -m 0755 bin/cri-dockerd /usr/local/bin/cri-dockerd
5. cp -a packaging/systemd/\* /etc/systemd/system
6. sudo sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service
7. sudo systemctl daemon-reload
8. sudo systemctl enable cri-docker.service
9. sudo systemctl enable --now cri-docker.socket

**Seting up the Kubernetes:**

1. sudo curl -fsSLo /etc/apt/keyrings/kubernetes-archive-keyring.gpg <https://dl.k8s.io/apt/doc/apt-key.gpg>
2. sudo echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
3. sudo apt-get update
4. # Use the same versions to avoid issues with the installation.
5. sudo apt-get install -y docker-ce kubelet=1.27.3-00 kubeadm=1.27.3-00 kubectl=1.27.3-00
6. sudo apt-mark hold docker-ce kubelet kubeadm kubectl
7. cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

overlay

br\_netfilter

EOF

1. sudo modprobe overlay
2. sudo modprobe br\_netfilter
3. 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

1. sudo sysctl –system
2. sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock --pod-network-cidr=192.168.0.0/16
3. mkdir -p $HOME/.kube
4. sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
5. sudo chown $(id -u):$(id -g) $HOME/.kube/config
6. # Use this if you have initialised the cluster with Calico network add on.
7. kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml
8. curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml -O

Change the ip to 10.244.0.0/16 if the node network is 192.168.x.x

1. kubectl create -f custom-resources.yaml

**After seting up the Kubernetes seting up the loki in it:**

1. sudo snap install helm –classic
2. helm repo add grafana <https://grafana.github.io/helm-charts>
3. helm repo update
4. helm search repo loki
5. helm show values grafana/loki-stack
6. helm show values grafana/loki-stack > values.yaml

open the values.yaml file and change the value of the Grafana “False” to “true”

and the change the version of it also.

1. helm install --values values.yaml loki grafana/loki-stack

To check the pods are created and running or not

1. kubectl get po
2. loki-kudernetes$ kubectl port-forward pod/loki-grafana-6f87885599-srfc92d 3000:3000

**If you want to connect your cluster in open lens:**

Go to the .kube / directory

Cd .kube/

Cat config

Copy the code and paste in the open lens on the add cluster>kubeconfig>