Pre-requites: Ubuntu 18.04., docker v19.03.13, kubernetes v1.21.0, go language 1.15.5 linux/amd64

Go Language version 1.15.5 must be installed and environment variables must also be set.

KubeEdge is now continuing to be upgraded. To avoid version differences, please add the syntax

"--kubeedge-version=1.9.1" when configuring kubeedge topology with keadm.

For example) sudo keadm join --kubeedge-version=1.9.1 --cloudcore-ipport=~~~~~~:10000 -- token

Red: Enter from the cloud core.

Blue: Enter from the edge core.

Green: Enter from the both core.

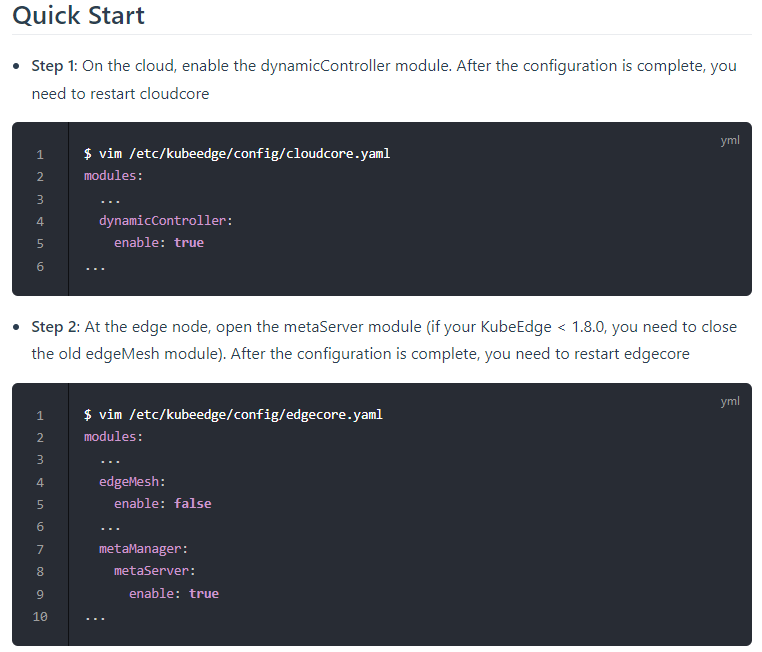
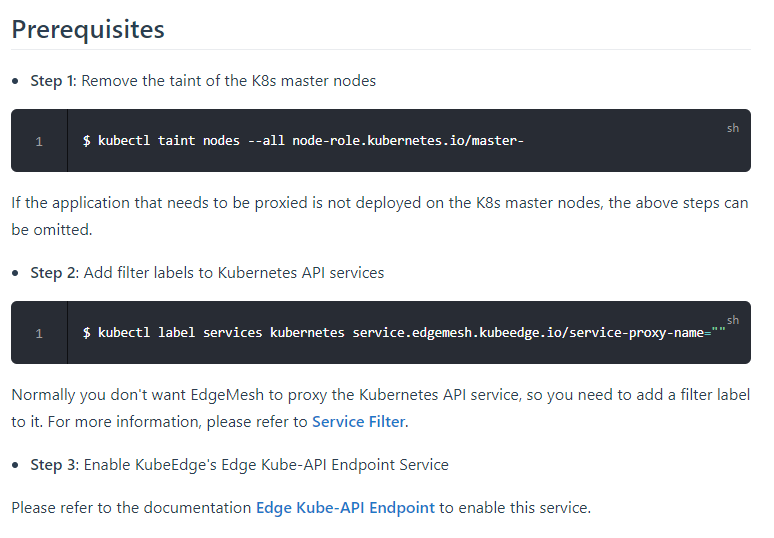
1. Follows Quan’s TUTORIAL ON SETUP KUBEEDGE ENVIRONMENT ON CLOUD AND EDGE

---------------------------------------------------------------------------------------------------------

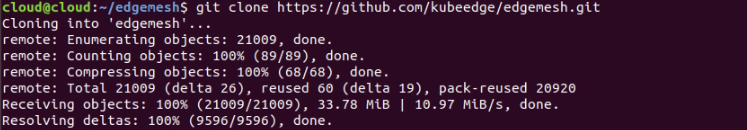
1. Refer to <https://edgemesh.netlify.app/>, and follows Prerequisites and install edgemesh, follows manual install, not helm install: In this case, I used vim editor, Search the Vim editor and learn how to use it. For example, how to modify, find letters, save and exit.

Example) sudo apt install vim

kubectl taint nodes –all node-role.kubernetes.io/master-

Kubectl label services kubernetes service.edgemesh.kubeedge.io/service-proxy-name=””

❖ Stop on step 2

Example) Download EdgeMesh: git clone <https://github.com/kubeedge/edgemesh.git>

cd edgemesh/

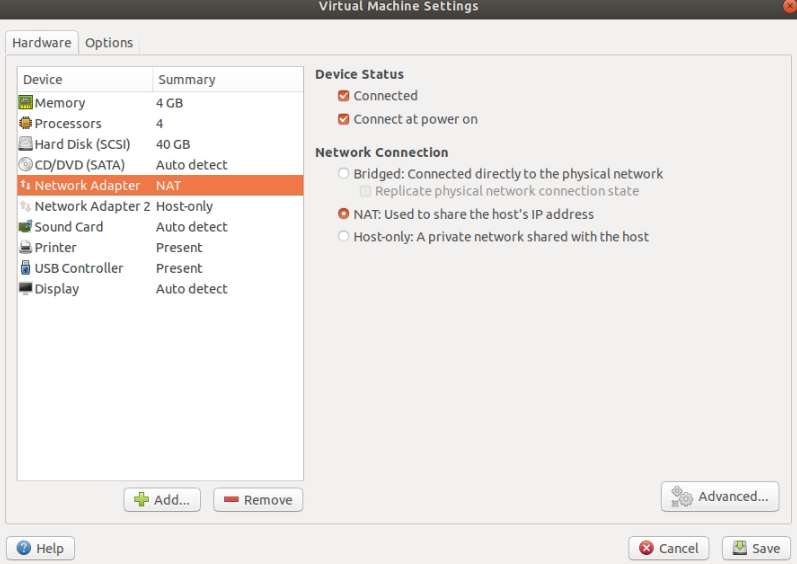
Create CRDs: kubectl apply -f build/crds/istio/

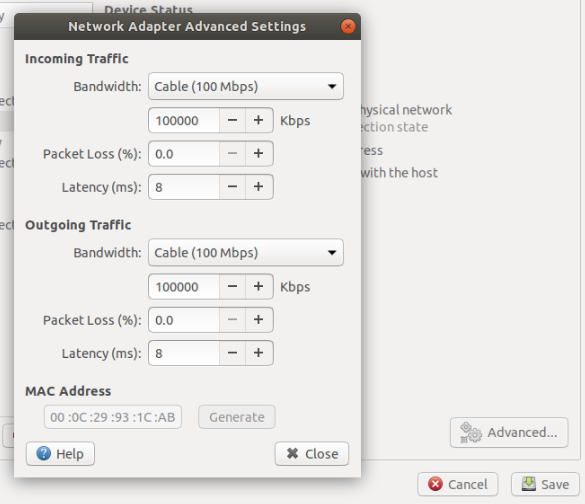
Deploy edgemesh-agent: Kubectl apply -f build/agent/resources/

A screenshot of a computer

Description automatically generated with medium confidence Check it out: Kubectl get all -n kubeedge -o wide

1. In order to configure the environment of edge computing to some extent, an artificial delay of 7 to 8 ms is given to each ‘edge node’ to configure a delay of 15 ms. (virtual Machine Settings Network Adapter)





4-1. ❖ Install Go version 1.15.5

<https://go.dev/dl/>

Find and download go1.15.5.linux-amd64.tar.gz file.

**Remove any previous Go installation** by deleting the /usr/local/go folder (if it exists), then extract the archive you just downloaded into /usr/local, creating a fresh Go tree in /usr/local/go:

cd ~/k8s-scheduler-extender-examples/

rm -rf /usr/local/go && tar -C sudo /usr/local -xzf go1.15.5.linux-amd64.tar.gz

export PATH=$PATH:/usr/local/go/bin

go version



go build .

Text

Description automatically generated

~~❖ Use Phuc’s scheduler-extender and scheduler-extender-config.yaml, scheduler-extender-policy.json file and replace the kube-scheduler (attached file)~~

Before deployment, Enter the number of pods you want to deploy in (~/k8s-scheduler-extender-examples/main.go) file.

❖ Importantly, the number of pods being built must be the same as the total number of deployments in the yaml file being deployed.

- To control the distribution of pods, edit file main.go in line 137:

podsDis[“edge1”] = 6

podsDis[“edge2”] = 6

podsDis[“edge3”] = 6

Delete all application pods cluster. Execute newly created file. Create new pods with yaml file.

4-2. ❖Since scheduler extender is implemented as a webhook call, so first we need to create a server which serves for the extender calls from the kube-scheduler whenever new pod is going to be scheduled.

4-3. copy files scheduler-extender-config.yaml and scheduler-extender-policy.json in thpa-config directory to /etc/kubernetes/:

Example) sudo cp /Downloads/thpa-config/scheduler-extender-config.yaml /etc/kubernetes/

sudo cp /Downloads/thpa-config/scheduler-extender-policy.json /etc/kubernetes/

4-4. copy file thpa-config/kube-scheduler.yaml to /etc/kubernetes/manifests/:

Example) sudo cp /Downloads/thpa-config/kube-scheduler.yaml /etc/kubernetes/manifests/

4-5. Execute kube-scheduler pod on master node.:

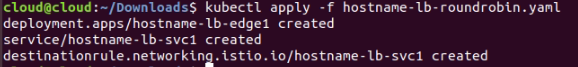
./k8s-scheduler-extender-example

Text

Description automatically generated

5-A. In round-robin case in testbed, Use example yaml file in

<https://github.com/kubeedge/edgemesh/tree/main/examples>

Example) 

5-B. In Local scheduling case in testbed, Use different yaml files to provide different services for each node to be distributed. For example, In my paper, Distribute the yaml file corresponding to each node. (The hostname-lb-roundrobin2.yaml file is distributed to edge2)

Example)

Kubectl apply -f hostname-lb-roundrobin1.yaml

Kubectl apply -f hostname-lb-roundrobin2.yaml

Kubectl apply -f hostname-lb-roundrobin3.yaml

❖❖❖ After deploy, you can check resources, kubectl get (node / pod / service)

A picture containing calendar

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

❖❖❖ If you want to delete it, kubectl delete -f ~~~.yaml

6. Refer to <https://github.com/rakyll/hey> , <https://hub.docker.com/r/williamyeh/hey/> , and use a docker image traffic generator by hey tool: ( c means concurrent requests, n means total requests, z means time to measure and add edge node’s service ip address and port number)

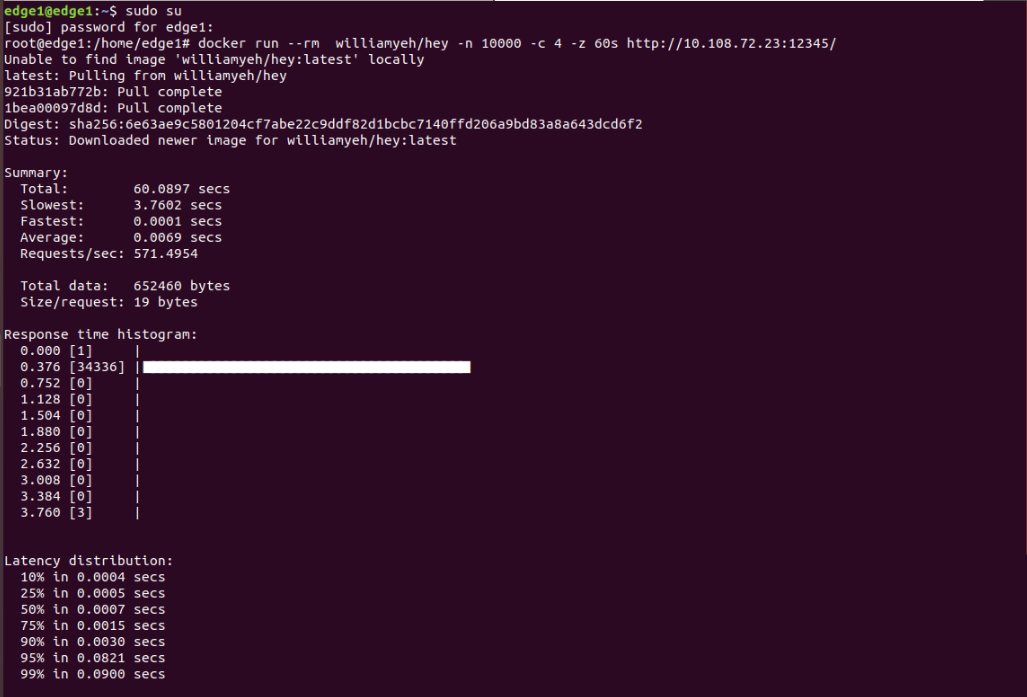
Example) Each Edge node:

sudo su

docker run –rm williamyeh/hey -n 10000 -c 4 -z 60s http://10.108.72.23:12345/

7. Do experiments. Choose 5-A or 5-B. Distribute the hostname-lb-roundrobin 1,2,3 files from (5-B.) to each edge node, and then transform the number of simultaneous users per node with a docker traffic generator from (manual number 6). to obtain throughput and latency.

Example) In my paper, roundrobin 1,2,3 files are 6-6-6,12-5-1,16-1-1 using a total of 18 pods



❖Average is Average response time, Requests/sec is Throughput

\*\*\*\*\*If the topology configuration is unstable due to an error, implement the topology again step by step, resetting it in the order you installed it recently.

❖❖ Let reset and reinstall k8s, kubeedge.

Reset KubeEdge(keadm): ❖ All node

sudo keadm reset –kube-config=$HOME/.kube/config

Reset Kubernetes(kubeadm): ❖ Cloud Core (master node)

sudo kubeadm reset

sudo rm -rf $HOME/.kube/config

sudo rm -rf /etc/cni/net.d