**Cloud Core**

**Step 1: keadm init --advertise-address="192.168.1.43"**

**Step 2:** Get Token

**keadm gettoken**

**Edge Core**

**Step 3:** Join with token

**keadm join --cloudcore-ipport=192.168.1.43:10000 --token=**27ab684d63d522e1fe8242333738cd069e41803d121f26c786edfad949893ef8.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2Mjg1NzkwODl9.QDZ6ZODnDFaaSp1h8PViQG-1N-b9dPY4zJeBF5Ju6VU

After edge is joined

**Cloud Core**

**Step 4:** Find ca.crt and ca.key files

**ls /etc/kubernetes/pki/**

**Step 5:** Same terminal is required

**export CLOUDCOREIPS="192.168.1.43"**

**Step 6:** Checks the cloudcore ip

**echo $CLOUDCOREIPS**

**Step 7:** Copy the certgen file

**cp /usr/local/go/src/github.com/kubeedge/kubeedge/build/tools/certgen.sh /etc/kubeedge/**

**Step 8:** Go to the directory

**cd /etc/kubeedge/**

**Step 9:** Generate certificate

**/etc/kubeedge/certgen.sh stream**

**Step 10:** Set ip tables on host

**kubectl get cm tunnelport -nkubeedge -oyaml**

**```bash**

**iptables -t nat -A OUTPUT -p tcp --dport 10350 -j DNAT --to 192.168.1.43:10003**

**```**

**Optional:** To clean iptables

**``` shell**

**iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables -X**

**```**

**Cloud Core**

**Step 11:** Open yaml in cloudcore

**sudo nano /etc/kubeedge/config/cloudcore.yaml**

**Step 12:** Modify cloudcore.yaml

cloudStream:

enable: true

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Edge Core**

**Step 13:** Open yaml in edgecore

**sudo nano /etc/kubeedge/config/edgecore.yaml**

**Step 14:** Modify cloudcore.yaml

edgeStream:

enable: true

server: 192.168.1.43:10004 (check ip)

**Step 15:** Check registerNode is true or not

edged:

registerNode: true

**Cloud Core**

**Step 16:** Restart CloudCore

**pkill cloudcore**

**nohup cloudcore > cloudcore.log 2>&1 &**

**Edge Core**

**Step 17:** Restart EdgeCore

**systemctl restart edgecore.service**

**Step 18:** Check if edgecore service is running or not

**systemctl**

**Cloud Core**

**Step 19:** Get Nodes

**kubectl get nodes**

**Step 20:** Get Pods

**kubectl get pods -A -o wide**

**Cloud Core**

**Step 21:** If fails to restart edgecore service than modify the proxy

**kubectl edit daemonsets.apps -n kube-system kube-proxy**

(add the following in spec inside spec)

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: node-role.kubernetes.io/edge

operator: DoesNotExist

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To reset the cloudcore and edgecore use **keadm reset** cmd.