Fornax: Deployment (ON Premise)

Date: 9 Dec. 2021

Machine Preparation

1. Prepare 4 machines, 16 Gb RAM,8 Vcpu, 80G storage, ubuntu 18.04, for the clusters of A, B, C and D.

2. Open the port of 10000 & 10002 in the security group of machine A, B and C.

3. Open the port of 6443 in the security group of machine A, B, C and D.

4. In machine A, B, C, create a Kubernetes cluster following doc https://kubernetes.io/docs/setup/productionenvironment/tools/kubeadm/create-cluster-kubeadm/.

5. In machine D, clone the repo arktos/arktos-with-mizar-cni.md at guide-cni-updates · Click2Cloud-Centaurus/arktos (github.com) and start an Arktos with mizar CNI cluster by running the script arktos-up.sh.

6. Install golang 1.13.9 version- all nodes

MachineA: (root operator machine)- IP: 192.168.1.210

MachineB: IP: 192.168.1.211

MachineC: IP: 192.168.1.212

MachineD: IP : 192.168.1.213

Setup Fornax: Machine A

1. Make sure Kubernetes cluster is ready:

   Kubectl get nodes
   Output:

```
root@node-a:~# kubectl get nodes
NAME     STATUS   ROLES                  AGE   VERSION
node-a   Ready    control-plane,master   46m   v1.21.1
root@node-a:~#
```

2. Clone a repo of https://github.com/CentaurusInfra/fornax, sync to the branch/commit to test.

   git clone https://github.com/CentaurusInfra/fornax.git
   cd fornax

   Output:

```
root@node-a:~# git clone https://github.com/CentaurusInfra/fornax.git
Cloning into 'fornax'...
git clone https://github.com/CentaurusInfra/fornax.gitremote: Enumerating objects: 50967, done.
remote: Counting objects: 100% (781/781), done.
remote: Compressing objects: 100% (385/385), done.
remote: Total 50967 (delta 429), reused 664 (delta 362), pack-reused 50186
Receiving objects: 100% (50967/50967), 114.10 MiB | 5.69 MiB/s, done.
Resolving deltas: 100% (28074/28074), done.
Checking out files: 100% (8784/8784), done.
root@node-a:~# cd fornax/
root@node-a:~/fornax#
```

3. Build the binaries of edgecore and cloudcore using the commands:

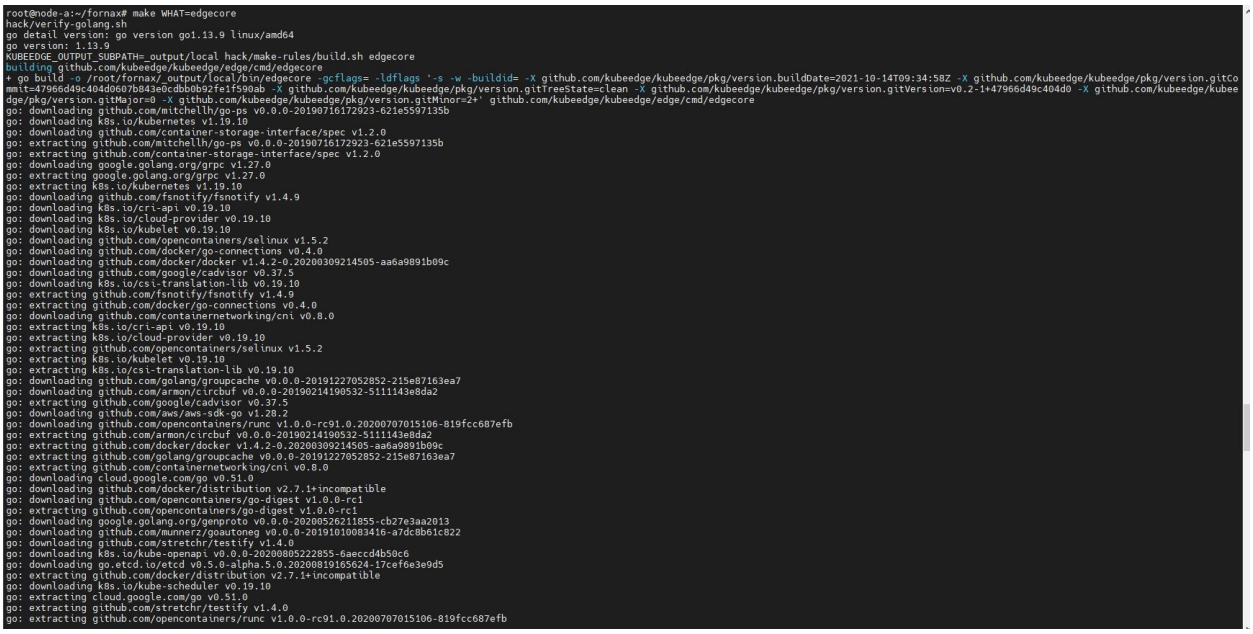
   `make WHAT=cloudcore`

Output:

```
root@node-a:~/fornax# make WHAT=cloudcore
hack/verify-golang.sh
go detail version: go version go1.13.9 linux/amd64
go version: 1.13.9
KUBEEDGE_OUTPUT_SUBPATH=_output/local hack/make-rules/build.sh cloudcore
building github.com/kubeedge/kubeedge/cloud/cmd/cloudcore
+ go build -o /root/fornax/_output/local/bin/cloudcore -gcflags= -ldflags '-s -w -buildid= -X github.com/kubeedge/kubeedge/pkg/version.buildDate=2021-10-14T09:26:58Z -X github.com/kubeedge/kubeedge/pkg/version.gitCommit=47966d49c404d0607b843e0cdbb0b92fe1f590ab -X github.com/kubeedge/kubeedge/pkg/version.gitTreeState=clean -X github.com/kubeedge/kubeedge/pkg/version.gitVersion=v0.2-1+47966d49c404d0 -X github.com/kubeedge/kubeedge/pkg/version.gitMajor=0 -X github.com/kubeedge/kubeedge/pkg/version.gitMinor=2+' github.com/kubeedge/kubeedge/cloud/cmd/cloudcore
go: downloading k8s.io/component-base v0.19.10
go: downloading github.com/spf13/cobra v1.0.0
go: downloading k8s.io/apimachinery v0.19.10
go: downloading github.com/gorilla/mux v1.7.3
go: downloading sigs.k8s.io/yaml v1.2.0
go: downloading k8s.io/klog v1.0.0
go: downloading k8s.io/client-go v0.19.10
go: extracting github.com/gorilla/mux v1.7.3
go: extracting sigs.k8s.io/yaml v1.2.0
go: extracting k8s.io/klog v1.0.0
go: extracting github.com/spf13/cobra v1.0.0
go: downloading k8s.io/klog/v2 v2.2.0
go: extracting k8s.io/component-base v0.19.10
go: extracting k8s.io/apimachinery v0.19.10
go: extracting k8s.io/client-go v0.19.10
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/satori/go.uuid v1.2.0
go: downloading github.com/gorilla/websocket v1.4.0
go: extracting k8s.io/klog/v2 v2.2.0
go: downloading github.com/go-logr/logr v0.2.0
go: downloading sigs.k8s.io/structured-merge-diff v0.0.0-20190525122527-15d366b2352e
go: downloading github.com/gogo/protobuf v1.3.2
go: downloading k8s.io/api v0.19.10
go: extracting gopkg.in/yaml.v2 v2.2.8
go: extracting github.com/gorilla/websocket v1.4.0
go: extracting github.com/spf13/pflag v1.0.5
go: extracting github.com/satori/go.uuid v1.2.0
go: downloading github.com/google/go-cmp v0.4.0
go: extracting sigs.k8s.io/structured-merge-diff v0.0.0-20190525122527-15d366b2352e
go: extracting github.com/go-logr/logr v0.2.0
go: downloading github.com/dgrijalva/jwt-go v3.2.0+incompatible
go: downloading sigs.k8s.io/structured-merge-diff/v4 v4.0.3
go: extracting github.com/gogo/protobuf v1.3.2
go: extracting k8s.io/api v0.19.10
go: extracting github.com/google/go-cmp v0.4.0
go: downloading github.com/emicklei/go-restful v2.9.6+incompatible
go: extracting github.com/dgrijalva/jwt-go v3.2.0+incompatible
go: extracting sigs.k8s.io/structured-merge-diff/v4 v4.0.3
go: extracting github.com/emicklei/go-restful v2.9.6+incompatible
go: downloading github.com/google/gofuzz v1.1.0
go: downloading go.uber.org/zap v1.10.0
go: downloading github.com/google/uuid v1.1.1
go: downloading k8s.io/utils v0.0.0-20200729134348-d5654de09c73
go: downloading github.com/davecgh/go-spew v1.1.1
go: downloading github.com/json-iterator/go v1.1.10
```

   `make WHAT=edgecore`

Output:



4. config cloudcore

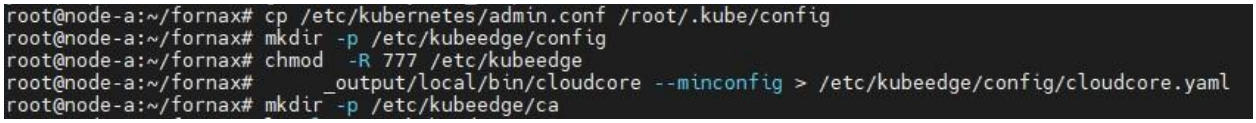
cp /etc/kubernetes/admin.conf /root/.kube/config

mkdir -p /etc/kubeedge/config

chmod -R 777 /etc/kubeedge

_output/local/bin/cloudcore --minconfig > /etc/kubeedge/config/cloudcore.yaml

Output:



5. Generate security data

mkdir -p /etc/kubeedge/ca

build/tools/certgen.sh genCA IP_A IP_B IP_C IP_D

build/tools/certgen.sh genCertAndKey server IP_A IP_B IP_C IP_D

Output:

```
root@node-a:~/fornax# mkdir -p /etc/kubeedge/ca
root@node-a:~/fornax# ls -l  /etc/kubeedge/ca
total 0
root@node-a:~/fornax# ls -l  /etc/kubeedge/
total 8
drwxr-xr-x 2 root root 4096 Oct 14 09:44 ca
drwxrwxrwx 2 root root 4096 Oct 14 09:44 config
root@node-a:~/fornax# build/tools/certgen.sh genCA 192.168.4.51 192.168.4.52 192.168.4.53 192.168.4.54
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................++++
.....................................++++
e is 65537 (0x010001)
Can't load /root/.rnd into RNG
139765808099776:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/root/.rnd
root@node-a:~/fornax# build/tools/certgen.sh genCertAndKey server 192.168.4.51 192.168.4.52 192.168.4.53 192.168.4.54
Generating RSA private key, 2048 bit long modulus (2 primes)
...........+++++
.......+++++
e is 65537 (0x010001)
Can't load /root/.rnd into RNG
139775158559168:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/root/.rnd
Signature ok
subject=C = CN, ST = Zhejiang, L = Hangzhou, O = KubeEdge, CN = kubeedge.io
Getting CA Private Key
root@node-a:~/fornax# ls -l  /etc/kubeedge/ca
total 12
-rw-r--r-- 1 root root 2057 Oct 14 09:47 rootCA.crt
-rw------- 1 root root 3311 Oct 14 09:47 rootCA.key
-rw-r--r-- 1 root root   41 Oct 14 09:47 rootCA.srl
root@node-a:~/fornax# ls -l  /etc/kubeedge/certs/
total 12
-rw-r--r-- 1 root root 1602 Oct 14 09:47 server.crt
-rw-r--r-- 1 root root  989 Oct 14 09:47 server.csr
-rw------- 1 root root 1675 Oct 14 09:47 server.key
root@node-a:~/fornax#
```

6. **Copy the files of folder /etc/kubeedge/ca and /etc/kubeedge/certs in machine A to the folder of /etc/kubeedge/ca and /etc/kubeedge/certs in machine B, C and D.**

Copy ca and certs to machineB:

scp -r /etc/kubeedge/ca/ root@node-b-ip:/etc/kubeedge
scp -r /etc/kubeedge/certs/ root@node-b-ip:/etc/kubeedge

Output:

```
root@node-a:~/fornax# scp -r /etc/kubeedge/ca/ root@192.168.4.52:/etc/kubeedge
The authenticity of host '192.168.4.52 (192.168.4.52)' can't be established.
ECDSA key fingerprint is SHA256:CB1dE2wmZ3LcNKU6RDD0D8xosH3X8k+M4YQMZtC34RI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.4.52' (ECDSA) to the list of known hosts.
root@192.168.4.52's password:
rootCA.crt                                    100% 2057     3.2MB/s   00:00
rootCA.key                                    100% 3311     4.7MB/s   00:00
rootCA.srl                                    100%   41    75.8KB/s   00:00
root@node-a:~/fornax# scp -r /etc/kubeedge/certs/ root@192.168.4.52:/etc/kubeedge
root@192.168.4.52's password:
server.crt                                    100% 1602     3.4MB/s   00:00
server.csr                                    100%  989     2.7MB/s   00:00
server.key                                    100% 1675     5.0MB/s   00:00
root@node-a:~/fornax#
```

Copy ca and certs to machineC:

scp -r /etc/kubeedge/ca/ root@node-c-ip:/etc/kubeedge
scp -r /etc/kubeedge/certs/ root@node-c-ip:/etc/kubeedge

Output:

```
root@node-a:~/fornax# scp -r /etc/kubeedge/ca/ root@192.168.4.53:/etc/kubeedge
The authenticity of host '192.168.4.53 (192.168.4.53)' can't be established.
ECDSA key fingerprint is SHA256:CB1dE2wmZ3LcNKU6RDD0D8xosH3X8k+M4YQMZtC34RI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.4.53' (ECDSA) to the list of known hosts.
root@192.168.4.53's password:
rootCA.crt                                    100% 2057     3.0MB/s   00:00
rootCA.key                                    100% 3311     5.9MB/s   00:00
rootCA.srl                                    100%   41    77.6KB/s   00:00
root@node-a:~/fornax# scp -r /etc/kubeedge/certs/ root@192.168.4.53:/etc/kubeedge
root@192.168.4.53's password:
server.crt                                    100% 1602     3.0MB/s   00:00
server.csr                                    100%  989     2.5MB/s   00:00
server.key                                    100% 1675     3.1MB/s   00:00
root@node-a:~/fornax#
```

Copy ca and certs to machineD:

scp -r /etc/kubeedge/ca/ root@node-d-ip:/etc/kubeedge
scp -r /etc/kubeedge/certs/ root@node-d-ip:/etc/kubeedge

Output:

7. Copy Kubeconfig file of cluster A(machine A) to Cluster B(machine B)

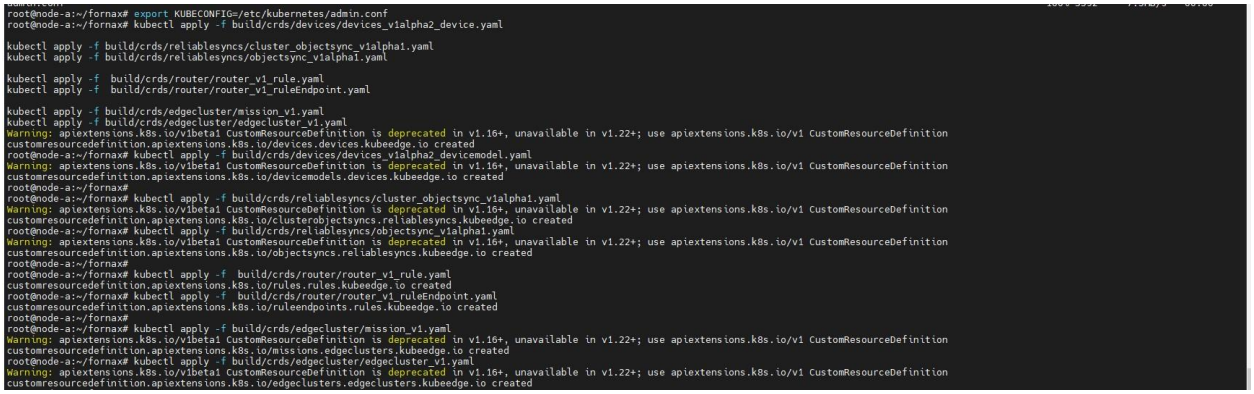
   scp /etc/kubernetes/admin.conf root@node-b-ip:/root/fornax

   Output:

8. Install CRDs

```
export KUBECONFIG=/etc/kubernetes/admin.conf
kubectl apply -f build/crds/devices/devices_v1alpha2_device.yaml
kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml

kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml

kubectl apply -f  build/crds/router/router_v1_rule.yaml
kubectl apply -f  build/crds/router/router_v1_ruleEndpoint.yaml

kubectl apply -f build/crds/edgecluster/mission_v1.yaml
kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml
```

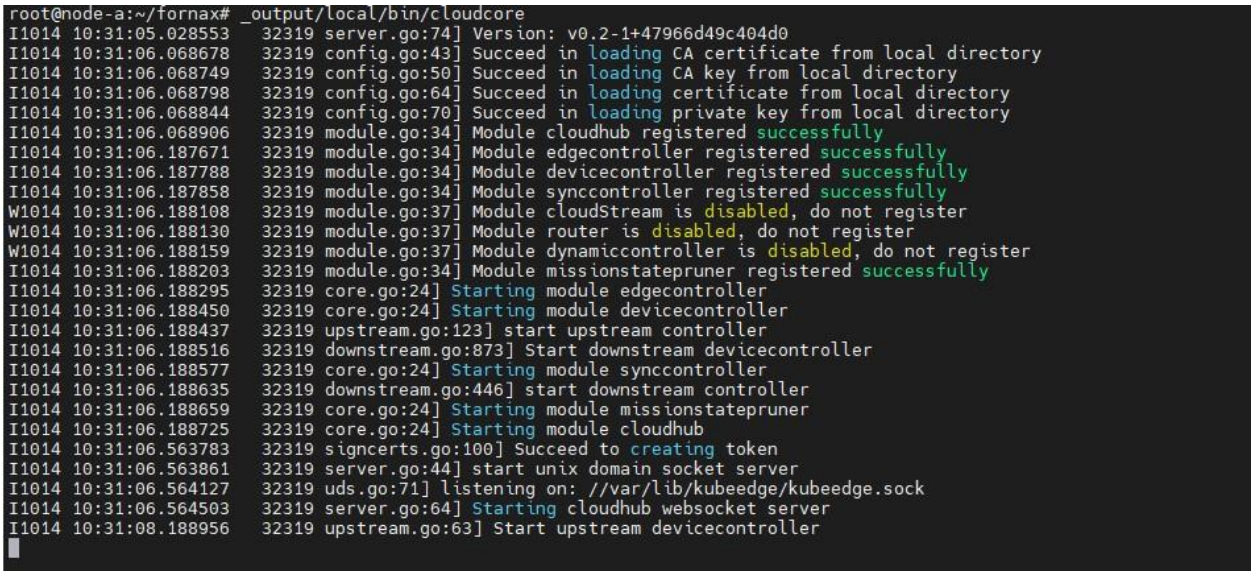
Output:



9. start kubeedge cloudcore in the root operator machine, using command:

   _output/local/bin/cloudcore

Output:



Leave the terminal as it is after cloudcore started in Machine A.

Setup Fornax: Machine B

1. Make sure Kubernetes cluster is ready:

   Kubectl get nodes

Output:

```
root@node-b:~# kubectl get nodes
NAME     STATUS   ROLES                  AGE   VERSION
node-b   Ready    control-plane,master   56m   v1.21.1
```

2. Clone a repo of https://github.com/CentaurusInfra/fornax, sync to the branch/commit to test.

   git clone https://github.com/CentaurusInfra/fornax.git
   cd fornax

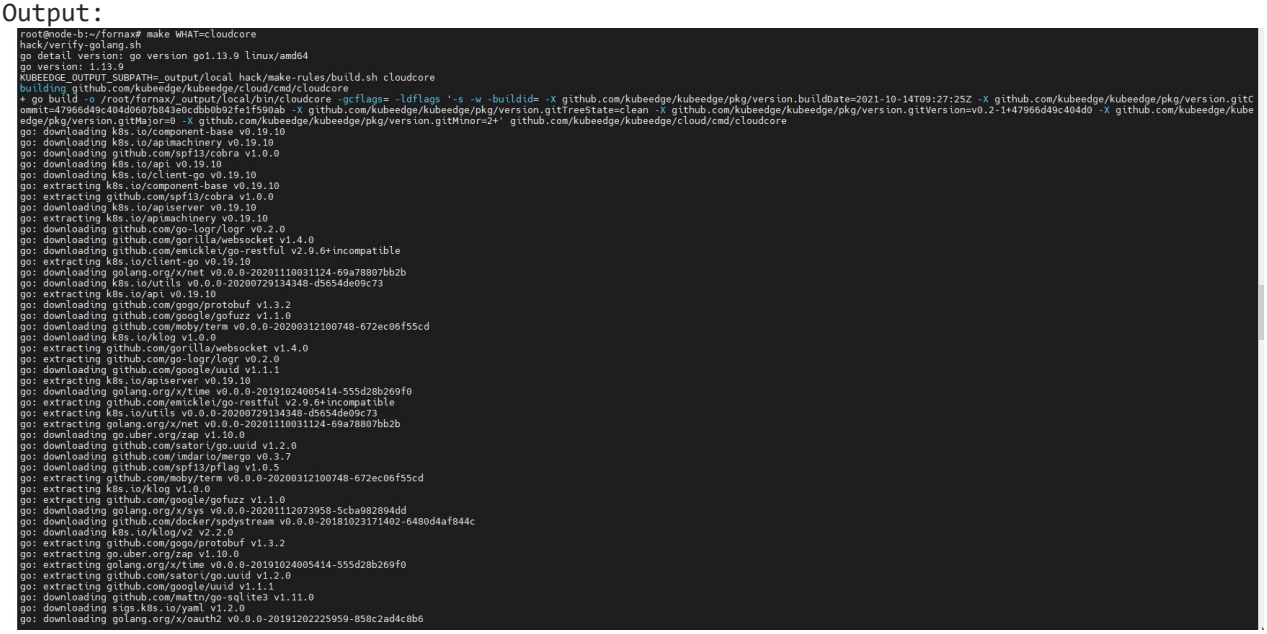
   Output:

```
root@node-b:~# git clone https://github.com/CentaurusInfra/fornax.git
Cloning into 'fornax'...
remote: Enumerating objects: 50967, done.
remote: Counting objects: 100% (781/781), done.
remote: Compressing objects: 100% (385/385), done.
remote: Total 50967 (delta 429), reused 664 (delta 362), pack-reused 50186
Receiving objects: 100% (50967/50967), 114.10 MiB | 7.22 MiB/s, done.
Resolving deltas: 100% (28074/28074), done.
Checking out files: 100% (8784/8784), done.
root@node-b:~# cd fornax/
root@node-b:~/fornax# make WHAT=cloudcore
```

3. Build the binaries of edgecore and cloudcore using the commands:

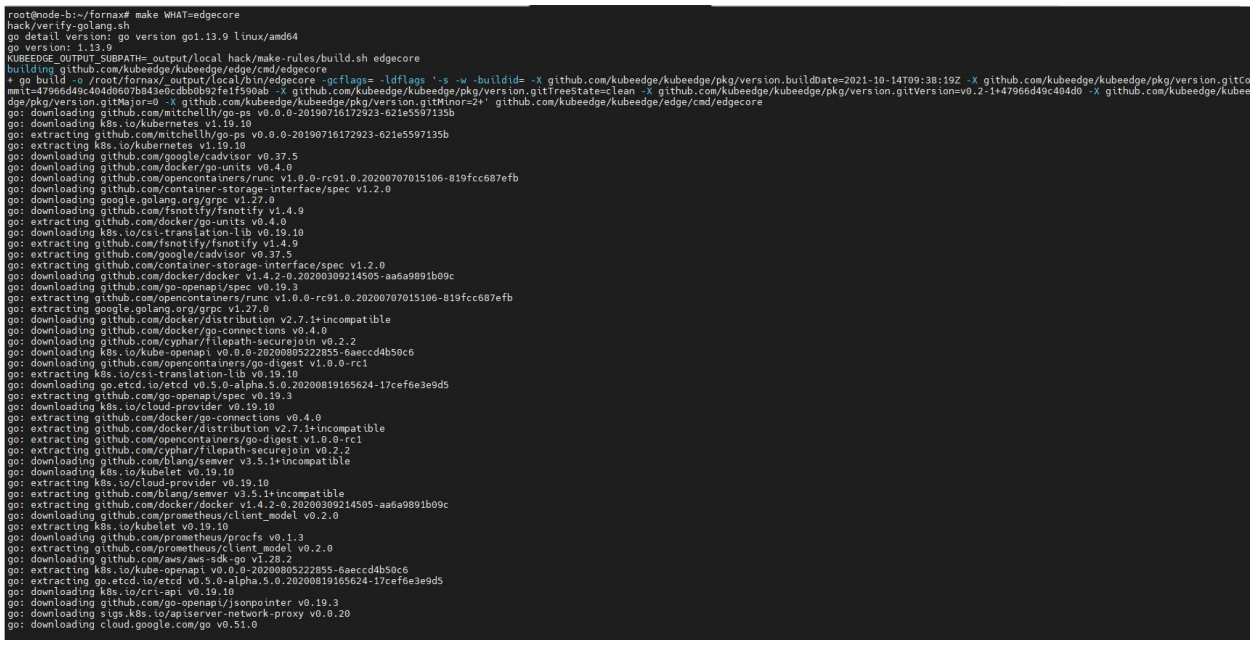
```
make WHAT=cloudcore
```

Output:




```
make WHAT=edgecore
```

Output:



4. config cloudcore

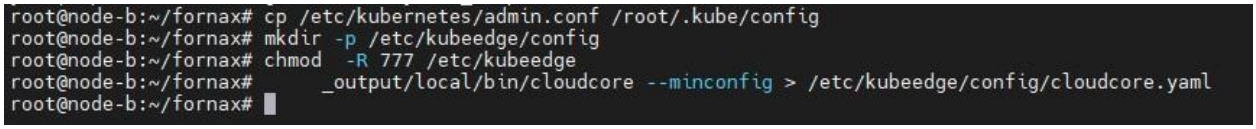
cp /etc/kubernetes/admin.conf /root/.kube/config

mkdir -p /etc/kubeedge/config

chmod -R 777 /etc/kubeedge

_output/local/bin/cloudcore --minconfig > /etc/kubeedge/config/cloudcore.yaml

Output:



5. Copy Kubeconfig file of clusterB(machineB) to ClusterC(machineC)

scp /etc/kubernetes/admin.conf root@node-c-ip:/root/fornax

Output:

6. config edgecore:

```
cp /etc/kubernetes/admin.conf /root/edgecluster.kubeconfig
_output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml
chmod 777 tests/edgecluster/hack/update_edgecore_config.sh
tests/edgecluster/hack/update_edgecore_config.sh admin.conf
```

Output:

```
root@node-b:~/fornax# cp /etc/kubernetes/admin.conf /root/edgecluster.kubeconfig
root@node-b:~/fornax#  _output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml
root@node-b:~/fornax# chmod 777 tests/edgecluster/hack/update_edgecore_config.sh
root@node-b:~/fornax# tests/edgecluster/hack/update_edgecore_config.sh admin.conf
root@node-b:~/fornax#
```

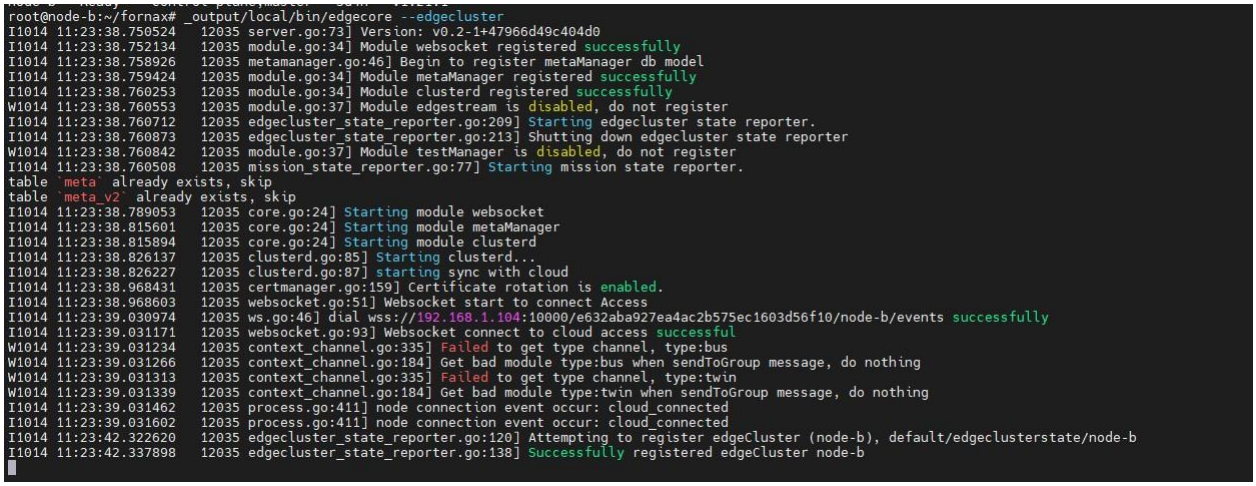
7. Install CRDs

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

kubectl apply -f build/crds/devices/devices_v1alpha2_device.yaml
kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml

kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml

kubectl apply -f  build/crds/router/router_v1_rule.yaml
kubectl apply -f  build/crds/router/router_v1_ruleEndpoint.yaml

kubectl apply -f build/crds/edgecluster/mission_v1.yaml
kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml
```

```
root@node-b:~/fornax# export KUBECONFIG=/etc/kubernetes/admin.conf
root@node-b:~/fornax# kubectl apply -f build/crds/devices/devices_v1alpha2_device.yaml
        kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml

        kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
        kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml

        kubectl apply -f  build/crds/router/router_v1_rule.yaml
        kubectl apply -f  build/crds/router/router_v1_ruleEndpoint.yaml

        kubectl apply -f build/crds/edgecluster/mission_v1.yaml
        kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml

Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/devices.devices.kubeedge.io created
root@node-b:~/fornax# kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/devicemodels.devices.kubeedge.io created
root@node-b:~/fornax#
root@node-b:~/fornax# kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/clusterobjectsyncs.reliablesyncs.kubeedge.io created
root@node-b:~/fornax# kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/objectsyncs.reliablesyncs.kubeedge.io created
root@node-b:~/fornax#
root@node-b:~/fornax# kubectl apply -f  build/crds/router/router_v1_rule.yaml
customresourcedefinition.apiextensions.k8s.io/rules.rules.kubeedge.io created
root@node-b:~/fornax# kubectl apply -f  build/crds/router/router_v1_ruleEndpoint.yaml
customresourcedefinition.apiextensions.k8s.io/ruleendpoints.rules.kubeedge.io created
root@node-b:~/fornax#
root@node-b:~/fornax# kubectl apply -f build/crds/edgecluster/mission_v1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/missions.edgeclusters.kubeedge.io created
root@node-b:~/fornax# kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/edgeclusters.edgeclusters.kubeedge.io created
root@node-b:~/fornax#
```

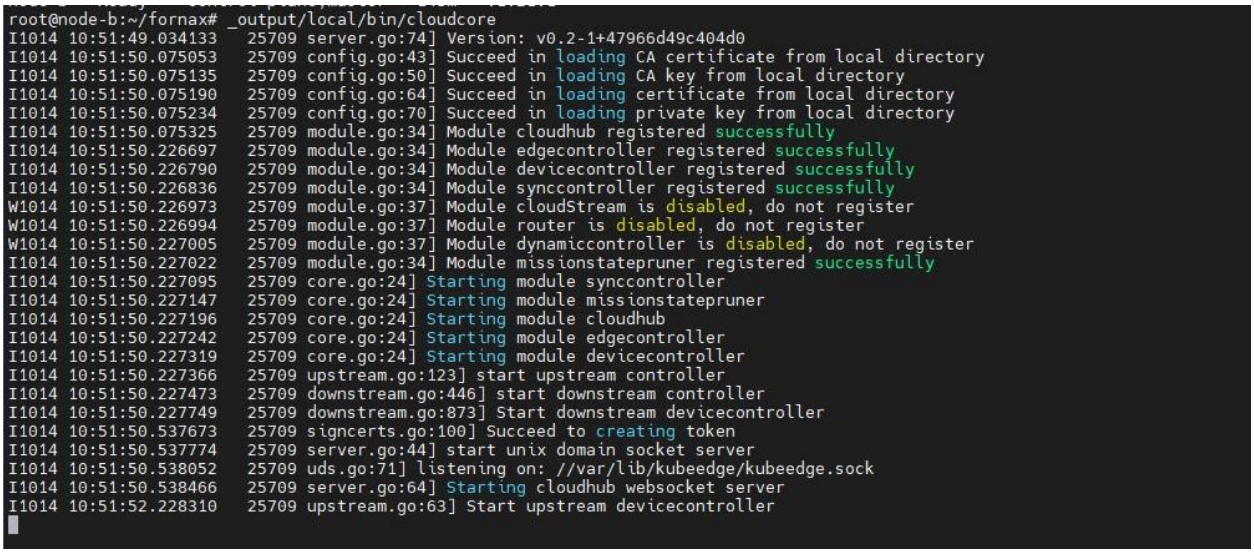
8. Start the edgecore in edge-cluster mode, with command:
   _output/local/bin/edgecore –edgecluster

   Output:



9. start cloudcore:
   _output/local/bin/cloudcor e

   Output:



Setup Fornax: Machine C

1. Make sure Kubernetes cluster is ready:

   Kubectl get nodes

   Output:

```
root@node-c:~# kubectl get nodes

NAME     STATUS   ROLES                  AGE   VERSION
node-c   Ready    control-plane,master   48m   v1.21.1
```

2. Clone a repo of https://github.com/CentaurusInfra/fornax, sync to the branch/commit to test.

   git clone https://github.com/CentaurusInfra/fornax.git
   cd fornax

   Output:

```
root@node-c:~# git clone https://github.com/CentaurusInfra/fornax.git
Cloning into 'fornax'...
remote: Enumerating objects: 50967, done.
remote: Counting objects: 100% (781/781), done.
remote: Compressing objects: 100% (385/385), done.
remote: Total 50967 (delta 429), reused 664 (delta 362), pack-reused 50186
Receiving objects: 100% (50967/50967), 114.10 MiB | 7.10 MiB/s, done.
Resolving deltas: 100% (28074/28074), done.
Checking out files: 100% (8784/8784), done.
root@node-c:~# cd fornax/
root@node-c:~/fornax# make WHAT=cloudcore
```

3. Build the binaries of edgecore and cloudcore using the commands:

   `make WHAT=cloudcore`

   Output:

```
root@node-c:~/fornax# make WHAT=cloudcore
hack/verify-golang.sh
go detail version: go version go1.13.9 linux/amd64
go version: 1.13.9
KUBEEDGE_OUTPUT_SUBPATH=_output/local hack/make-rules/build.sh cloudcore
building github.com/kubeedge/kubeedge/cloud/cmd/cloudcore
+ go build -o /root/fornax/_output/local/bin/cloudcore -gcflags= -ldflags '-s -w -buildid= -X github.com/kubeedge/kubeedge/pkg/version.buildDate=2021-10-14T09:27:33Z -X github.com/kubeedge/kubeedge/pkg/version.gitCommit=47966d49c404d0607b843e0cdbb0b92fe1f590ab -X github.com/kubeedge/kubeedge/pkg/version.gitTreeState=clean -X github.com/kubeedge/kubeedge/pkg/version.gitVersion=v0.2-1+47966d49c404d0 -X github.com/kubeedge/kubeedge/pkg/version.gitMajor=0 -X github.com/kubeedge/kubeedge/pkg/version.gitMinor=2+' github.com/kubeedge/kubeedge/cloud/cmd/cloudcore
go: downloading k8s.io/component-base v0.19.10
go: downloading k8s.io/apimachinery v0.19.10
go: downloading k8s.io/klog v1.0.0
go: downloading k8s.io/client-go v0.19.10
go: extracting k8s.io/klog v1.0.0
go: downloading k8s.io/klog/v2 v2.2.0
go: extracting k8s.io/apimachinery v0.19.10
go: extracting k8s.io/component-base v0.19.10
go: extracting k8s.io/client-go v0.19.10
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/moby/term v0.0.0-20200312100748-672ec06f55cd
go: downloading github.com/go-logr/logr v0.2.0
go: downloading k8s.io/apiserver v0.19.10
go: downloading github.com/google/uuid v1.1.1
go: extracting k8s.io/klog/v2 v2.2.0
go: downloading k8s.io/api v0.19.10
go: downloading github.com/dgrijalva/jwt-go v3.2.0+incompatible
go: downloading sigs.k8s.io/yaml v1.2.0
go: downloading github.com/gorilla/websocket v1.4.0
go: downloading k8s.io/utils v0.0.0-20200729134348-d5654de09c73
go: extracting github.com/spf13/pflag v1.0.5
go: extracting github.com/google/uuid v1.1.1
go: extracting sigs.k8s.io/yaml v1.2.0
go: extracting github.com/dgrijalva/jwt-go v3.2.0+incompatible
go: extracting github.com/go-logr/logr v0.2.0
go: extracting github.com/moby/term v0.0.0-20200312100748-672ec06f55cd
go: extracting github.com/gorilla/websocket v1.4.0
go: downloading github.com/google/gofuzz v1.1.0
go: downloading github.com/imdario/mergo v0.3.7
go: downloading github.com/emicklei/go-restful v2.9.6+incompatible
go: downloading go.uber.org/zap v1.10.0
go: extracting k8s.io/utils v0.0.0-20200729134348-d5654de09c73
go: extracting k8s.io/apiserver v0.19.10
go: extracting k8s.io/api v0.19.10
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading github.com/google/go-cmp v0.4.0
go: downloading github.com/spf13/cobra v1.0.0
go: downloading github.com/docker/spdystream v0.0.0-20181023171402-6480d4af844c
go: downloading github.com/astaxie/beego v1.12.0
go: downloading github.com/davecgh/go-spew v1.1.1
go: extracting go.uber.org/zap v1.10.0
go: extracting github.com/imdario/mergo v0.3.7
go: extracting github.com/google/gofuzz v1.1.0
go: downloading github.com/gogo/protobuf v1.3.2
go: downloading gopkg.in/inf.v0 v0.9.1
go: extracting github.com/emicklei/go-restful v2.9.6+incompatible
```

`make WHAT=edgecore`

Output:

```
root@node-c:~/fornax# make WHAT=edgecore
hack/verify-golang.sh
go detail version: go version go1.13.9 linux/amd64
go version: 1.13.9
KUBEEDGE_OUTPUT_SUBPATH=_output/local hack/make-rules/build.sh edgecore
building github.com/kubeedge/kubeedge/edge/cmd/edgecore
+ go build -o /root/fornax/_output/local/bin/edgecore -gcflags= -ldflags '-s -w -buildid= -X github.com/kubeedge/kubeedge/pkg/version.buildDate=2021-10-14T09:39:22Z -X github.com/kubeedge/kubeedge/pkg/version.gitCommit=47966d49c404d0607b843e0cdbb0b92fe1f590ab -X github.com/kubeedge/kubeedge/pkg/version.gitTreeState=clean -X github.com/kubeedge/kubeedge/pkg/version.gitVersion=v0.2-1+47966d49c404d0 -X github.com/kubeedge/kubeedge/pkg/version.gitMajor=0 -X github.com/kubeedge/kubeedge/pkg/version.gitMinor=2+' github.com/kubeedge/kubeedge/edge/cmd/edgecore
go: downloading github.com/eclipse/paho.mqtt.golang v1.2.0
go: downloading k8s.io/kubernetes v1.19.10
go: downloading sigs.k8s.io/apiserver-network-proxy v0.0.20
go: extracting sigs.k8s.io/apiserver-network-proxy v0.0.20
go: extracting github.com/eclipse/paho.mqtt.golang v1.2.0
go: downloading sigs.k8s.io/apiserver-network-proxy/konnectivity-client v0.0.20
go: extracting sigs.k8s.io/apiserver-network-proxy/konnectivity-client v0.0.20
go: extracting k8s.io/kubernetes v1.19.10
go: downloading k8s.io/kubelet v0.19.10
go: downloading k8s.io/kube-openapi v0.0.0-20200805222855-6aeccd4b50c6
go: downloading google.golang.org/grpc v1.27.0
go: downloading github.com/google/cadvisor v0.37.5
go: downloading k8s.io/cloud-provider v0.19.10
go: downloading github.com/go-openapi/spec v0.19.3
go: extracting k8s.io/cloud-provider v0.19.10
go: extracting k8s.io/kubelet v0.19.10
go: extracting github.com/google/cadvisor v0.37.5
go: downloading github.com/coreos/go-systemd v0.0.0-20190620071333-e64a0ec8b42a
go: downloading k8s.io/kube-scheduler v0.19.10
go: downloading github.com/opencontainers/runc v1.0.0-rc91.0.20200707015106-819fcc687efb
go: downloading k8s.io/cri-api v0.19.10
go: extracting github.com/go-openapi/spec v0.19.3
go: extracting google.golang.org/grpc v1.27.0
go: extracting k8s.io/kube-openapi v0.0.0-20200805222855-6aeccd4b50c6
go: downloading github.com/docker/go-units v0.4.0
go: downloading github.com/evanphx/json-patch v4.9.0+incompatible
go: downloading k8s.io/apiextensions-apiserver v0.19.10
go: downloading k8s.io/csi-translation-lib v0.19.10
go: extracting github.com/coreos/go-systemd v0.0.0-20190620071333-e64a0ec8b42a
go: downloading go.etcd.io/etcd v0.5.0-alpha.5.0.20200819165624-17cef6e3e9d5
go: extracting github.com/evanphx/json-patch v4.9.0+incompatible
go: extracting k8s.io/cri-api v0.19.10
go: extracting k8s.io/kube-scheduler v0.19.10
go: extracting github.com/docker/go-units v0.4.0
go: extracting github.com/opencontainers/runc v1.0.0-rc91.0.20200707015106-819fcc687efb
go: extracting k8s.io/csi-translation-lib v0.19.10
go: extracting k8s.io/apiextensions-apiserver v0.19.10
go: downloading github.com/blang/semver v3.5.1+incompatible
go: downloading google.golang.org/genproto v0.0.0-20200526211855-cb27e3aa2013
go: downloading github.com/docker/docker v1.4.2-0.20200309214505-aa6a9891b09c
go: downloading github.com/prometheus/client_model v0.2.0
go: downloading github.com/container-storage-interface/spec v1.2.0
go: downloading github.com/prometheus/client_golang v1.7.1
go: downloading github.com/go-openapi/jsonpointer v0.19.3
go: downloading github.com/cyphar/filepath-securejoin v0.2.2
go: downloading github.com/godbus/dbus/v5 v5.0.3
go: extracting github.com/blang/semver v3.5.1+incompatible
go: downloading github.com/256dpi/gomqtt v0.10.4
```

4. config cloudcore

   `mkdir -p /etc/kubeedge/config`

   `chmod  -R 777 /etc/kubeedge`

   `_output/local/bin/cloudcore --minconfig > /etc/kubeedge/config/cloudcore.yaml`

Output:

```
root@node-c:~/fornax# cp /etc/kubernetes/admin.conf /root/.kube/config
root@node-c:~/fornax#
root@node-c:~/fornax# mkdir -p /etc/kubeedge/config
root@node-c:~/fornax# chmod  -R 777 /etc/kubeedge
root@node-c:~/fornax#  _output/local/bin/cloudcore --minconfig > /etc/kubeedge/config/cloudcore.yaml
root@node-c:~/fornax#
```

5. Copy Kubeconfig file of clusterC(machineC) to ClusterD(machineD)

scp /etc/kubernetes/admin.conf root@node-d-ip:/root/go/src/k8s.io/arktos/fornax

Output:

```
root@node-c:~/fornax# scp /etc/kubernetes/admin.conf  root@192.168.4.54:/root/go/src/k8s.io/arktos/fornax
The authenticity of host '192.168.4.54 (192.168.4.54)' can't be established.
ECDSA key fingerprint is SHA256:CB1dE2wmZ3LcNKU6RDD0D8xosH3X8k+M4YQMZtC34RI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.4.54' (ECDSA) to the list of known hosts.
root@192.168.4.54's password:
admin.conf                                                                    100% 5596     6.4MB/s   00:00
root@node-c:~/fornax#
```

6. config edgecore:

cp /etc/kubernetes/admin.conf /root/edgecluster.kubeconfig
_output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml
chmod 777 tests/edgecluster/hack/update_edgecore_config.sh
tests/edgecluster/hack/update_edgecore_config.sh admin.conf

Output:

```
root@node-c:~/fornax# cp /etc/kubernetes/admin.conf /root/edgecluster.kubeconfig
root@node-c:~/fornax# _output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml
root@node-c:~/fornax#
root@node-c:~/fornax#       chmod  777 tests/edgecluster/hack/update_edgecore_config.sh
root@node-c:~/fornax# tests/edgecluster/hack/update_edgecore_config.sh admin.conf
root@node-c:~/fornax#
```

7. Install CRDs

export KUBECONFIG=/etc/kubernetes/admin.conf

kubectl apply -f build/crds/devices/devices_v1alpha2_device.yaml
kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml

kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml

kubectl apply -f build/crds/router/router_v1_rule.yaml
kubectl apply -f build/crds/router/router_v1_ruleEndpoint.yaml

kubectl apply -f build/crds/edgecluster/mission_v1.yaml

kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml

Output:

```
root@node-c:~/fornax# export KUBECONFIG=/etc/kubernetes/admin.conf
root@node-c:~/fornax# kubectl apply -f build/crds/devices/devices_v1alpha2_device.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/devices.devices.kubeedge.io created
root@node-c:~/fornax# kubectl apply -f build/crds/devices/devices_v1alpha2_devicemodel.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/devicemodels.devices.kubeedge.io created
root@node-c:~/fornax#
root@node-c:~/fornax# kubectl apply -f build/crds/reliablesyncs/cluster_objectsync_v1alpha1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/clusterobjectsyncs.reliablesyncs.kubeedge.io created
root@node-c:~/fornax# kubectl apply -f build/crds/reliablesyncs/objectsync_v1alpha1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/objectsyncs.reliablesyncs.kubeedge.io created
root@node-c:~/fornax#
root@node-c:~/fornax# kubectl apply -f  build/crds/router/router_v1_rule.yaml
customresourcedefinition.apiextensions.k8s.io/rules.rules.kubeedge.io created
root@node-c:~/fornax# kubectl apply -f  build/crds/router/router_v1_ruleEndpoint.yaml
customresourcedefinition.apiextensions.k8s.io/ruleendpoints.rules.kubeedge.io created
root@node-c:~/fornax#
root@node-c:~/fornax# kubectl apply -f build/crds/edgecluster/mission_v1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/missions.edgeclusters.kubeedge.io created
root@node-c:~/fornax# kubectl apply -f build/crds/edgecluster/edgecluster_v1.yaml
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
customresourcedefinition.apiextensions.k8s.io/edgeclusters.edgeclusters.kubeedge.io created
root@node-c:~/fornax#
```

8. Start the edgecore in edge-cluster mode, with command:

_output/local/bin/edgecore –edgecluster

Output:

```
root@node-c:~/fornax# _output/local/bin/edgecore --edgecluster
I1014 11:26:08.584799   12644 server.go:73] Version: v0.2-1+47966d49c404d0
I1014 11:26:08.586254   12644 module.go:34] Module websocket registered successfully
I1014 11:26:08.604292   12644 metamanager.go:46] Begin to register metaManager db model
I1014 11:26:08.604998   12644 module.go:34] Module metaManager registered successfully
I1014 11:26:08.606103   12644 module.go:34] Module clusterd registered successfully
I1014 11:26:08.606370   12644 mission_state_reporter.go:77] Starting mission state reporter.
I1014 11:26:08.606360   12644 edgecluster_state_reporter.go:209] Starting edgecluster state reporter.
I1014 11:26:08.606731   12644 edgecluster_state_reporter.go:213] Shutting down edgecluster state reporter
W1014 11:26:08.606525   12644 module.go:37] Module edgestream is disabled, do not register
W1014 11:26:08.607177   12644 module.go:37] Module testManager is disabled, do not register
table `meta` already exists, skip
table `meta_v2` already exists, skip
I1014 11:26:08.822910   12644 core.go:24] Starting module websocket
I1014 11:26:08.823040   12644 core.go:24] Starting module metaManager
I1014 11:26:08.840451   12644 core.go:24] Starting module clusterd
I1014 11:26:08.841330   12644 clusterd.go:85] Starting clusterd...
I1014 11:26:08.841394   12644 clusterd.go:87] starting sync with cloud
I1014 11:26:08.902698   12644 certmanager.go:159] Certificate rotation is enabled.
I1014 11:26:08.902847   12644 websocket.go:51] Websocket start to connect Access
I1014 11:26:08.947184   12644 ws.go:46] dial wss://192.168.1.102:10000/e632aba927ea4ac2b575ec1603d56f10/node-c/events successfully
I1014 11:26:08.947387   12644 websocket.go:93] Websocket connect to cloud access successful
W1014 11:26:08.947487   12644 context_channel.go:335] Failed to get type channel, type:twin
W1014 11:26:08.947519   12644 context_channel.go:184] Get bad module type:twin when sendToGroup message, do nothing
W1014 11:26:08.947563   12644 context_channel.go:335] Failed to get type channel, type:bus
W1014 11:26:08.947587   12644 context_channel.go:184] Get bad module type:bus when sendToGroup message, do nothing
I1014 11:26:08.947699   12644 process.go:411] node connection event occur: cloud_connected
I1014 11:26:08.947933   12644 process.go:411] node connection event occur: cloud_connected
I1014 11:26:12.714743   12644 edgecluster_state_reporter.go:120] Attempting to register edgeCluster (node-c), default/edgeclusterstate/node-c
I1014 11:26:12.737210   12644 edgecluster_state_reporter.go:138] Successfully registered edgeCluster node-c
```

9. start cloudcore

_output/local/bin/cloudcore

Output:

```
root@node-c:~/fornax#    _output/local/bin/cloudcore
I1014 11:03:28.078781   32314 server.go:74] Version: v0.2-1+47966d49c404d0
I1014 11:03:29.351015   32314 config.go:43] Succeed in loading CA certificate from local directory
I1014 11:03:29.351093   32314 config.go:50] Succeed in loading CA key from local directory
I1014 11:03:29.351139   32314 config.go:64] Succeed in loading certificate from local directory
I1014 11:03:29.351176   32314 config.go:70] Succeed in loading private key from local directory
I1014 11:03:29.351209   32314 module.go:34] Module cloudhub registered successfully
I1014 11:03:29.375123   32314 module.go:34] Module edgecontroller registered successfully
I1014 11:03:29.375218   32314 module.go:34] Module devicecontroller registered successfully
I1014 11:03:29.375286   32314 module.go:34] Module synccontroller registered successfully
W1014 11:03:29.375407   32314 module.go:37] Module cloudStream is disabled, do not register
W1014 11:03:29.375425   32314 module.go:37] Module router is disabled, do not register
W1014 11:03:29.375441   32314 module.go:37] Module dynamiccontroller is disabled, do not register
I1014 11:03:29.375456   32314 module.go:34] Module missionstatepruner registered successfully
I1014 11:03:29.375518   32314 core.go:24] Starting module cloudhub
I1014 11:03:29.375559   32314 core.go:24] Starting module edgecontroller
I1014 11:03:29.375598   32314 core.go:24] Starting module devicecontroller
I1014 11:03:29.375645   32314 core.go:24] Starting module synccontroller
I1014 11:03:29.375691   32314 upstream.go:123] start upstream controller
I1014 11:03:29.375707   32314 core.go:24] Starting module missionstatepruner
I1014 11:03:29.375722   32314 downstream.go:873] Start downstream devicecontroller
I1014 11:03:29.376190   32314 downstream.go:446] start downstream controller
I1014 11:03:30.050029   32314 signcerts.go:100] Succeed to creating token
I1014 11:03:30.050107   32314 server.go:44] start unix domain socket server
I1014 11:03:30.050341   32314 uds.go:71] listening on: //var/lib/kubeedge/kubeedge.sock
I1014 11:03:30.050693   32314 server.go:64] Starting cloudhub websocket server
I1014 11:03:31.376186   32314 upstream.go:63] Start upstream devicecontroller
```

Leave terminal as it is.

Setup Fornax: Machine D

1. **Make sure Arktos cluster is ready:**

   ./cluster/kubectl.sh get nodes

   Output:

```
root@node-d:~/go/src/k8s.io/arktos# ./cluster/kubectl.sh get nodes
NAME     STATUS   ROLES    AGE   VERSION
node-d   Ready    <none>   29m   v0.9.0
root@node-d:~/go/src/k8s.io/arktos#
```

2. **Clone a repo of https://github.com/CentaurusInfra/fornax, sync to the branch/commit to test.**
3.

   git clone https://github.com/CentaurusInfra/fornax.git
   cd fornax

Output:

```
root@node-d:~/go/src/k8s.io/arktos# git clone https://github.com/CentaurusInfra/fornax.git
Cloning into 'fornax'...
remote: Enumerating objects: 50967, done.
remote: Counting objects: 100% (781/781), done.
remote: Compressing objects: 100% (385/385), done.
remote: Total 50967 (delta 429), reused 664 (delta 362), pack-reused 50186
Receiving objects: 100% (50967/50967), 114.10 MiB | 9.31 MiB/s, done.
Resolving deltas: 100% (28074/28074), done.
Checking out files: 100% (8784/8784), done.
root@node-d:~/go/src/k8s.io/arktos# cd fornax
root@node-d:~/go/src/k8s.io/arktos/fornax#
```

mkdir -p /etc/kubeedge

chmod -R 777 /etc/kubeedge

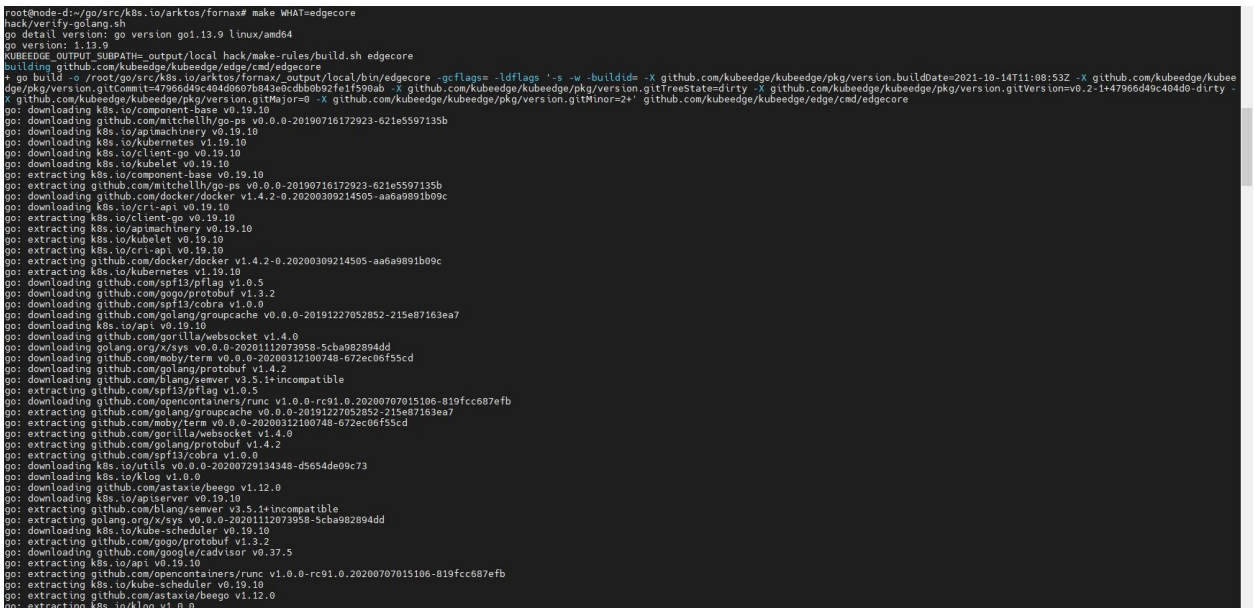
Output:

```
root@node-d:~/go/src/k8s.io/arktos/fornax# mkdir -p /etc/kubeedge
root@node-d:~/go/src/k8s.io/arktos/fornax# chmod  -R 777 /etc/kubeedge
root@node-d:~/go/src/k8s.io/arktos/fornax#
```

3. **Build the binary of edgecore**

make WHAT=edgecore

Output:



4. **config edgecore**

cp /var/run/kubernetes/admin.kubeconfig /root/edgecluster.kubeconfig

mkdir -p /etc/kubeedge/config

chmod -R 777 /etc/kubeedge

_output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml

chmod 777 tests/edgecluster/hack/update_edgecore_config.sh

tests/edgecluster/hack/update_edgecore_config.sh admin.conf

```
root@node-d:~/go/src/k8s.io/arktos/fornax# cp /var/run/kubernetes/admin.kubeconfig /root/edgecluster.kubeconfig
root@node-d:~/go/src/k8s.io/arktos/fornax# mkdir -p /etc/kubeedge/config
root@node-d:~/go/src/k8s.io/arktos/fornax# chmod -R 777 /etc/kubeedge
root@node-d:~/go/src/k8s.io/arktos/fornax# _output/local/bin/edgecore --edgeclusterconfig > /etc/kubeedge/config/edgecore.yaml
root@node-d:~/go/src/k8s.io/arktos/fornax# chmod 777 tests/edgecluster/hack/update_edgecore_config.sh
```

Output

Install ' kubectl ' on machine D

update the /etc/kubeedge/config/edgecore.yaml

clusterd: ...
kubeDistro: arktos
labels:
"company" : "futurewei"

5. **start the edgecore in edge-cluster mode, with command**

chmod 777 /root/go/src/k8s.io/arktos/fornax/_output/local/bin/kubectl/

_output/local/bin/edgecore –edgecluster

Output:

```
root@node-d:~/go/src/k8s.io/arktos/fornax#       _output/local/bin/edgecore --edgecluster
I1014 12:09:40.358894   24085 server.go:73] Version: v0.2-1+47966d49c404d0-dirty
I1014 12:09:40.359219   24085 module.go:34] Module websocket registered successfully
I1014 12:09:40.359282   24085 metamanager.go:46] Begin to register metaManager db model
I1014 12:09:40.359389   24085 module.go:34] Module metaManager registered successfully
I1014 12:09:40.359477   24085 module.go:34] Module clusterd registered successfully
W1014 12:09:40.359513   24085 module.go:37] Module edgestream is disabled, do not register
W1014 12:09:40.359541   24085 module.go:37] Module testManager is disabled, do not register
I1014 12:09:40.359712   24085 edgecluster_state_reporter.go:209] Starting edgecluster state reporter.
I1014 12:09:40.359780   24085 mission_state_reporter.go:77] Starting mission state reporter.
I1014 12:09:40.359738   24085 edgecluster_state_reporter.go:213] Shutting down edgecluster state reporter
create table `meta`
    -- ----------------------------------------------------
    --  Table Structure for `github.com/kubeedge/kubeedge/edge/pkg/metamanager/dao.Meta`
    -- ----------------------------------------------------
    CREATE TABLE IF NOT EXISTS `meta` (
        `key` varchar(256) NOT NULL PRIMARY KEY,
        `type` varchar(32) NOT NULL DEFAULT '' ,
        `value` text
    );

create table `meta_v2`
    -- ----------------------------------------------------
    --  Table Structure for `github.com/kubeedge/kubeedge/edge/pkg/metamanager/dao/v2.MetaV2`
    -- ----------------------------------------------------
    CREATE TABLE IF NOT EXISTS `meta_v2` (
        `key` varchar(256) NOT NULL PRIMARY KEY,
        `groupversionresource` varchar(256) NOT NULL DEFAULT '' ,
        `namespace` varchar(256) NOT NULL DEFAULT '' ,
        `name` varchar(256) NOT NULL DEFAULT '' ,
        `resourceversion` bigint unsigned NOT NULL DEFAULT 0 ,
        `value` text
    );

I1014 12:09:41.431351   24085 core.go:24] Starting module websocket
I1014 12:09:41.431413   24085 core.go:24] Starting module metaManager
```

Leave the terminal as it is..

Verification:

Verify the deployment on clusterA:
Node-b is the edge of node-a
Node-c is the edge of node-b
Node-d is the edge of node-c

```
root@node-a:~/fornax# kubectl get edgecluster
NAME     LASTHEARBEAT   HEALTHSTATUS   SUBEDGECLUSTERS                                    RECEIVED_MISSIONS   MATCHED_MISSIONS
node-b   4s             healthy        {"node-c":"healthy","node-c/node-d":"healthy"}
root@node-a:~/fornax#
```