Fornax – Sample Application deployment Report

Date: 9th 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 https://github.com/CentaurusInfra/arktos/ and start an Arktos cluster by running the script arktos-up.sh.

6. Install golang 1.13 version- all nodes

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#
```

Deployment :

1. **Go to the ai_app directory in node-a:**

   cd /root/fornax/tests/edgecluster/data/ai_app

   Output:

```
root@node-a:~/fornax/tests/edgecluster/data/ai_app# ls
01_create_ns_face.yaml     08_cp_known.yaml           15_nsqlookup_deployment.yaml        22_face_recog_deployment.yaml
02_label_nodes.yaml        09_cp_unknown.yaml         16_receiver_service.yaml            23_frontend_service.yaml
03_create_secret.yaml      10_unknown_pv.yaml         17_nsqd_service.yaml                24_frontend_deployment.yaml
04_mysql_service.yaml      11_known_pv.yaml           18_receiver_deployment.yaml         30_port_forward_mysql.yaml
05_mysql_pvc.yaml          12_unknown_pvc.yaml        19_nsqd_deployment.yaml             31_port_forward_receiver.yaml
06_create_configmap.yaml   13_known_pvc.yaml         20_image_processor_deployment.yaml  32_port_forward_frontend.yaml
07_mysql_deployment.yaml   14_nsqlookup_service.yaml  21_face_recog_service.yaml          mysql-pv.yml
```

2. **Apply the deployments**

   cd **..**

   kubectl apply -Rf ai_app

   Output

```
root@node-a:~/fornax/tests/edgecluster/data# kubectl apply -Rf ai_app/
mission.edgeclusters.kubeedge.io/command-create-ns-face created
mission.edgeclusters.kubeedge.io/command-label-node created
mission.edgeclusters.kubeedge.io/resource-secret created
mission.edgeclusters.kubeedge.io/resource-mysql-service created
mission.edgeclusters.kubeedge.io/resource-mysql-pvc created
mission.edgeclusters.kubeedge.io/command-create-configmap created
mission.edgeclusters.kubeedge.io/resource-mysql-deployment created
mission.edgeclusters.kubeedge.io/command-cp-known created
mission.edgeclusters.kubeedge.io/command-cp-unknown created
mission.edgeclusters.kubeedge.io/resource-unknown-pv created
mission.edgeclusters.kubeedge.io/resource-known-pv created
mission.edgeclusters.kubeedge.io/resource-unknown-pvc created
mission.edgeclusters.kubeedge.io/resource-known-pvc created
mission.edgeclusters.kubeedge.io/resource-nsqlookup-service created
mission.edgeclusters.kubeedge.io/resource-nsqlookup-deployment created
mission.edgeclusters.kubeedge.io/resource-receiver-svc created
mission.edgeclusters.kubeedge.io/resource-nsqd-svc created
mission.edgeclusters.kubeedge.io/resource-receiver-deployment created
mission.edgeclusters.kubeedge.io/resource-nsqd-deployment created
mission.edgeclusters.kubeedge.io/resource-image-processor-deployment created
mission.edgeclusters.kubeedge.io/resource-face-recog-svc created
mission.edgeclusters.kubeedge.io/resource-face-recog-deployment created
mission.edgeclusters.kubeedge.io/resource-frontend-svc created
mission.edgeclusters.kubeedge.io/resource-frontend-deployment created
mission.edgeclusters.kubeedge.io/command-mysql-port-forward created
mission.edgeclusters.kubeedge.io/command-receiver-port-forward created
mission.edgeclusters.kubeedge.io/command-frontend-port-forward created
mission.edgeclusters.kubeedge.io/resource-mysql-pv created
```

3. **Check the pods in node-b**

   kubectl get pods -A

   Output :

```
root@node-b:~/fornax# kubectl get pods -A
NAMESPACE     NAME                                        READY   STATUS              RESTARTS   AGE
face          face-recog-698dc6b88f-87pnl                 0/1     CrashLoopBackOff    8          24m
face          frontend-56b6fd5f8c-c2n2t                   1/1     Running             0          24m
face          image-processor-deployment-7d6d54d996-mkxtl 1/1     Running             0          24m
face          mysql-67ff5f6bf4-24lgz                      0/1     ContainerCreating   0          24m
face          nsqd-54667b87f4-gz4h2                       1/1     Running             0          24m
face          nsqlookup-56768d5bd8-8rvmp                  1/1     Running             0          24m
face          receiver-deployment-74b5c7d449-rfs6f        1/1     Running             0          24m
kube-system   coredns-558bd4d5db-pxlm6                    1/1     Running             0          119m
kube-system   coredns-558bd4d5db-zgght                    1/1     Running             0          119m
kube-system   etcd-node-b                                 1/1     Running             0          119m
kube-system   kube-apiserver-node-b                       1/1     Running             0          119m
kube-system   kube-controller-manager-node-b              1/1     Running             0          119m
kube-system   kube-flannel-ds-rxgkl                       1/1     Running             0          118m
kube-system   kube-proxy-qv42t                            1/1     Running             0          119m
kube-system   kube-scheduler-node-b                       1/1     Running             0          119m
root@node-b:~/fornax#
```

4. **Check the pods in node-c**

kubectl get pods -A

Output :

```
NAMESPACE     NAME                                        READY   STATUS              RESTARTS   AGE
face          face-recog-698dc6b88f-p2n76                 0/1     CrashLoopBackOff    8          25m
face          frontend-56b6fd5f8c-92mgb                   1/1     Running             0          26m
face          image-processor-deployment-7d6d54d996-wqmfl 1/1     Running             0          25m
face          mysql-67ff5f6bf4-vmdgg                      0/1     ContainerCreating   0          26m
face          nsqd-54667b87f4-6f768                       1/1     Running             0          26m
face          nsqlookup-56768d5bd8-ktwbd                  1/1     Running             0          25m
face          receiver-deployment-74b5c7d449-54ttd        1/1     Running             0          25m
kube-system   coredns-558bd4d5db-gjfcq                    1/1     Running             0          121m
kube-system   coredns-558bd4d5db-ktd47                    1/1     Running             0          121m
kube-system   etcd-node-c                                 1/1     Running             0          121m
kube-system   kube-apiserver-node-c                       1/1     Running             0          121m
kube-system   kube-controller-manager-node-c              1/1     Running             0          121m
kube-system   kube-flannel-ds-srw44                       1/1     Running             0          120m
kube-system   kube-proxy-bskm8                            1/1     Running             0          121m
kube-system   kube-scheduler-node-c                       1/1     Running             0          121m
root@node-c:~/fornax#
```

5. **Check the pods in node-d**

kubectl get pods -A

Output :

```
root@node-d:~/go/src/k8s.io/arktos/fornax# kubectl get pods -A
NAMESPACE     NAME                                HASHKEY              READY   STATUS    RESTARTS   AGE
default       mizar-daemon-tv4xh                  6827121902641575131  1/1     Running   0          62m
default       mizar-operator-6b78d7ffc4-2692j     2302138113806156596  1/1     Running   0          63m
kube-system   kube-dns-554c5866fc-vsglf           4296459504327514785  0/3     Pending   0          63m
root@node-d:~/go/src/k8s.io/arktos/fornax#
```

ERROR:

*Deployment in **node-d** failed due to **generation of 2 Private IP's***

Output :

```
root@node-d:~/go/src/k8s.io/arktos/fornax# cat  /tmp/arktos-network-controller.log
F1209 06:20:29.339031   30842 network-controller.go:62] --kube-apiserver-ip must be the valid ip address of kube-apiserver.
root@node-d:~/go/src/k8s.io/arktos/fornax# hostname -i
192.168.1.213 172.17.0.1 fe80::250:56ff:fe86:1f62 fe80::c437:60ff:fe26:b65 fe80::7052:2aff:fecd:98b1
```

SOLUTION:

1. **Edit the hosts file with a specific** *ip and name*

vi /etc/hosts

Output:

```
127.0.0.1 localhost
192.168.1.213 node-d

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
```

2. **Now go to previous node-d terminal and re-run the arktos script :**

```
CNIPLUGIN=mizar ./hack/arktos-up.sh -O
```

3. **Now apply the deployment in node-a again:**

```
cd  /root/fornax/tests/edgecluster/data/ai_app
cd ..
kubectl apply -Rf ai_app
```

After few minutes verify the deployment in all nodes

Output Of Node-d :

```
kubectl get pods -A
```

```
root@node-d:~/go/src/k8s.io/arktos/fornax# kubectl get pods -A
NAMESPACE     NAME                                        HASHKEY              READY   STATUS              RESTARTS   AGE
default       mizar-daemon-h2w69                          6592199866192543920  1/1     Running             0          37m
default       mizar-operator-6b78d7ffc4-rfz8j             8671591457198608881  1/1     Running             0          37m
face          face-recog-cc5788dff-snzfq                  640869359021234371   0/1     ContainerCreating   0          10m
face          frontend-64f9fd599c-6pldr                   7194155719604393031  0/1     ContainerCreating   0          10m
face          image-processor-deployment-54488487c7-m2vl9 4942787007233899615  0/1     ContainerCreating   0          11m
face          mysql-59b99c5f5c-ncq2g                      3242139854942213360  0/1     ContainerCreating   0          8m58s
face          nsqd-594c8db6dd-bwsc4                       8468725754715426589  0/1     ContainerCreating   0          12m
face          nsqlookup-b986db78f-2j64w                   4020931345773023195  0/1     ContainerCreating   0          12m
face          receiver-deployment-559c44888f-dlqp8        1334860494849302785  0/1     ContainerCreating   0          11m
kube-system   coredns-default-fc74854f6-pzz4s             7524768366422865253  0/1     ContainerCreating   0          37m
kube-system   kube-dns-554c5866fc-t9tgg                   6702434315285952260  0/3     ContainerCreating   0          37m
kube-system   virtlet-f9m97                               6831290709867304711  3/3     Running             0          28m
```