Edge Multi-Cluster Orchestrator (EMCO) Installation

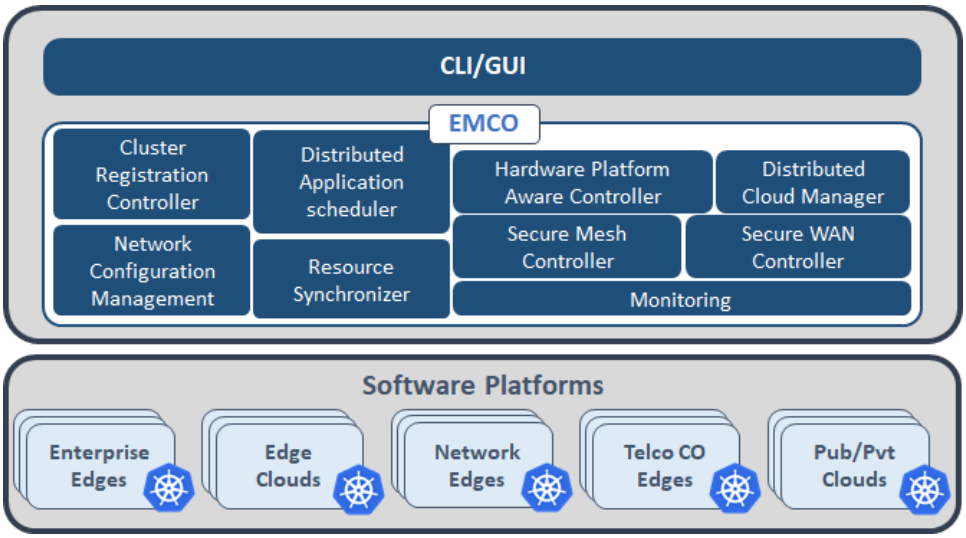
Sl. No.	Document Name	Version	Authors
1.	EMCO Installation	0.1	Athira Vinod

Edge Multi-Cluster Orchestrator (EMCO) is an open-source project aimed at simplifying the management and orchestration of edge computing infrastructure. It provides a set of tools and functionalities to help organizations deploy, monitor, and manage applications and services across multiple edge clusters. Here are some of its key functionalities:

1. **Multi-Cluster Management:** EMCO enables the management of multiple edge clusters, allowing you to deploy and manage applications consistently across a distributed edge infrastructure.
2. **Application Orchestration:** It provides tools for defining and orchestrating applications to run on edge clusters, ensuring that services are distributed efficiently and operate reliably in a distributed environment.
3. **Resource Allocation:** EMCO helps allocate and manage resources such as CPU, memory, and storage across edge clusters, optimizing the utilization of available resources for applications and services.
4. **Service Discovery and Load Balancing:** It offers service discovery mechanisms to help applications locate and interact with services running on various edge clusters. Load balancing capabilities ensure that traffic is distributed efficiently.
5. **Health Monitoring:** EMCO provides monitoring and health-checking features, allowing you to track the status of applications and services running on edge clusters and respond to issues proactively.

6. **Security:** It incorporates security features to ensure the integrity and confidentiality of data and applications at the edge. This includes authentication, authorization, and encryption mechanisms.
7. **Scaling:** EMCO allows for auto-scaling of applications based on demand, ensuring that resources are dynamically allocated as needed to handle varying workloads.
8. **Configuration Management:** It offers configuration management tools to define and manage the configurations of applications and services, ensuring consistency and compliance across edge clusters.
9. **Extensibility:** EMCO is designed to be extensible, allowing you to integrate it with other tools and services to meet specific edge computing requirements.

Architecture



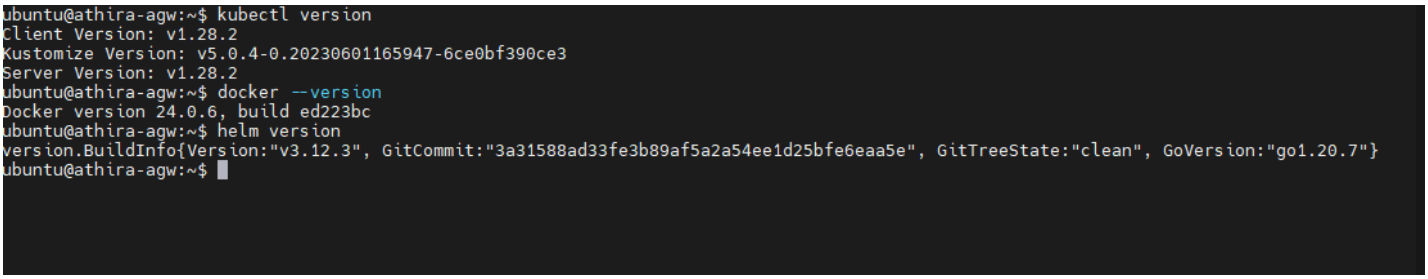
Ref: https://wiki.onap.org/pages/viewpage.action?pageId=84668166

1. **Cluster Registration Controller** registers clusters by cluster owners
2. **Network Configuration Management** handles creation/management of virtual and provider networks
3. **Distributed Application Scheduler** provides simplified, and extensible placement

4. **Hardware Platform Aware Controller** enables scheduling with auto-discovery of platform features/ capabilities
5. **Distributed Cloud Manager** presents a single logical cloud from multiple edges
6. **Secure Mesh Controller** auto-configures both service mesh (ISTIO) and security policy (NAT, firewall)
7. **Secure WAN Controller** automates secure overlays across edge groups
8. **Resource Syncronizer** manages instantiation of resources to clusters
9. **Monitoring** covers distributed application

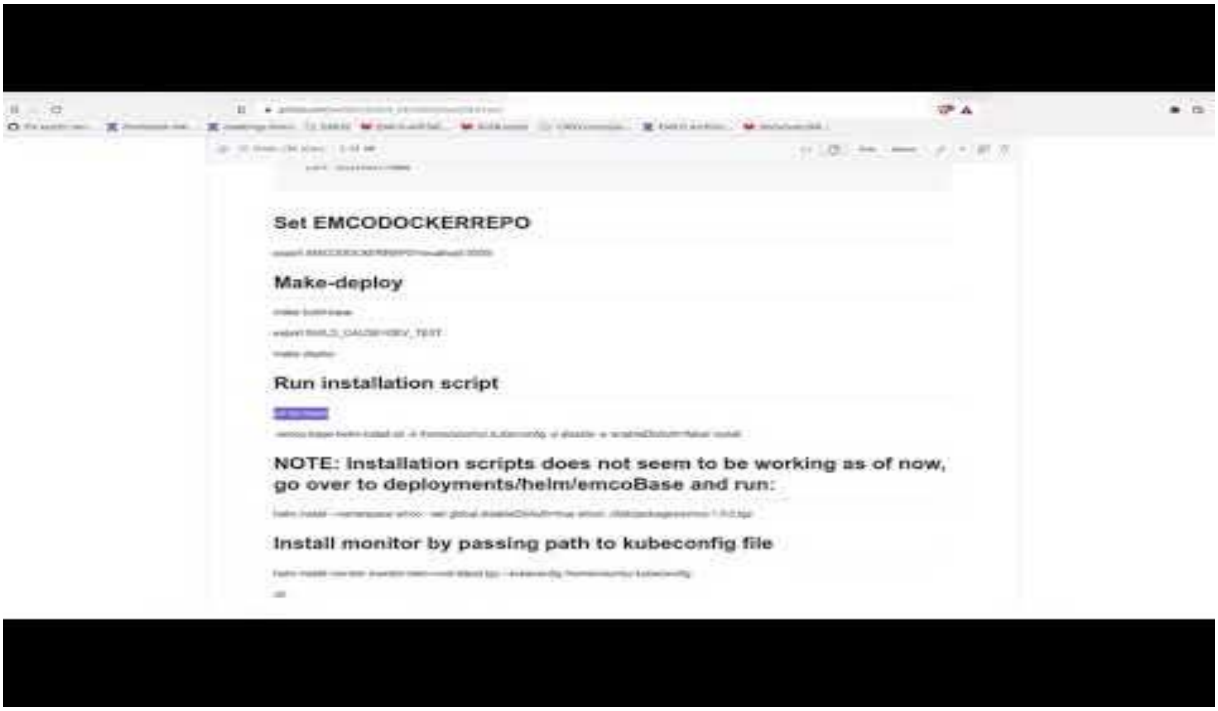
Prerequisites

1. Ubuntu 20.04.5
2. Kubeadm Cluster
3. docker (v18.09.6 or later)
4. helm (v3.3.4 or later)
5. kubectl (v1.19.0 or later)



How to setup Prerequisites(2 – 5)

Instead of **./start.sh** in the repo mentioned in this video Deploying Edge-Multicluster orchestrator (EMCO) on KOO(Kubernetes On OpenStack) Cluster, broke down the steps to make troubleshooting easier.



installing kubeadm,kubectl,kubelet

1. *sudo apt-get update*
2. *sudo apt-get install -y apt-transport-https ca-certificates curl*
3. *sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://dl.k8s.io/apt/doc/apt-key.gpg*
4. *echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list*
5. *sudo apt-get update*
6. *sudo apt-get install -y kubelet kubeadm kubectl*
7. *sudo apt-mark hold kubelet kubeadm kubectl*

Docker

1. *curl -fsSL https://get.docker.com -o get-docker.sh*
2. *sudo sh get-docker.sh*

Resolving container runtime errors:

sudo systemctl restart containerd

Create cluster:

1. *sudo kubeadm init --pod-network-cidr=10.244.0.0/16*
2. *mkdir -p $HOME/.kube*
3. *sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config*
4. *sudo chown $(id -u):$(id -g) $HOME/.kube/config*
5. *kubectl apply -f https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml*
6. *kubectl taint nodes --all node-role.kubernetes.io/control-plane:NoSchedule-*

Installing Helm

1. *curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null*
2. *sudo apt-get install apt-transport-https –yes*
3. *echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list*
4. *sudo apt-get update*
5. *sudo apt-get install helm*
6. *kubectl get pods -A*

```
root@athira-docker:/home/ubuntu/Kubeadm-Cluster# kubectl get pods -A
NAMESPACE      NAME                                    READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-cw6xw                   1/1     Running   0          40h
kube-system    coredns-5dd5756b68-cpdng                1/1     Running   0          40h
kube-system    coredns-5dd5756b68-fnbm2                1/1     Running   0          40h
kube-system    etcd-athira-docker                      1/1     Running   0          40h
kube-system    kube-apiserver-athira-docker            1/1     Running   0          40h
kube-system    kube-controller-manager-athira-docker   1/1     Running   0          40h
kube-system    kube-proxy-w82zq                        1/1     Running   0          40h
kube-system    kube-scheduler-athira-docker            1/1     Running   0          40h
```

Installation

Ref: https://github.com/wafi981/EMCO_DEV/blob/main/Start.md

mkdir work

cd work

git clone https://gitlab.com/project-emco/core/emco-base.git

cd emco-base

Docker Registry

docker run -d -p 5000:5000 --name registry registry:2.7

curl localhost:5000

Set EMCODOCKERREPO

export EMCODOCKERREPO=localhost:5000/

Make-deploy

make build-base

export BUILD_CAUSE=DEV_TEST

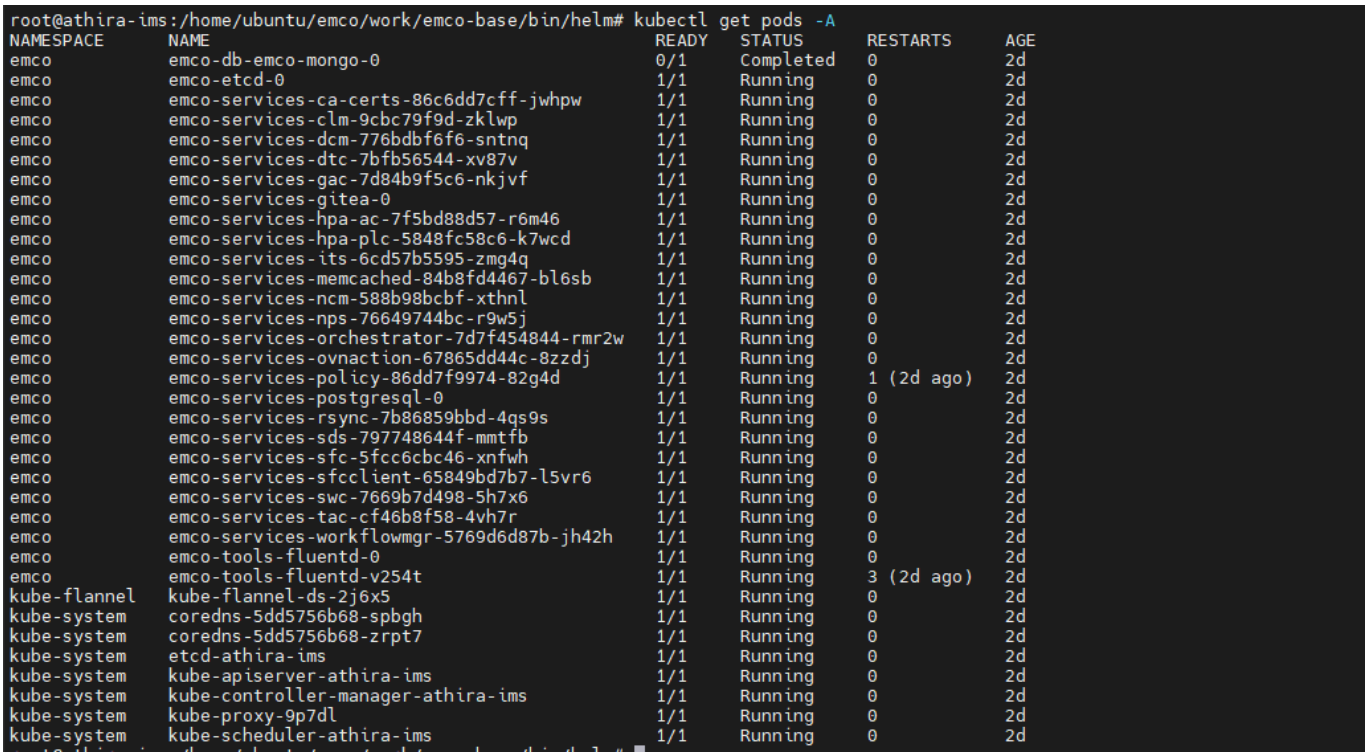
make deploy

Run Installation Script

cd bin/helm

./emco-base-helm-install.sh -k /home/ubuntu/.kube/config -p disable -s 'enableDbAuth=false' install

In another terminal check all the Kubernetes services state.



If all the emco services are running, installation is successful.

Troubleshooting Guide

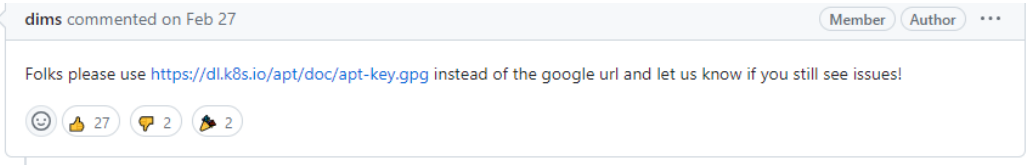
ISSUE 1:

W: GPG error: https://packages.cloud.google.com/apt kubernetes-xenial InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY B53DC80D13EDEF05

E: The repository 'https://apt.kubernetes.io kubernetes-xenial InRelease' is not signed.

installing kubeadm,kubectl,kubelet (Step 3): The command is updated with new repo link. Please follow this STEP 3.

Ref: :



ISSUE 2:

If you are trying again or already you have Kubernetes/ Docker in your server which is showing some clash, remove the *.list files from the path *`/etc/apt/sources.list.d`*

ISSUE 3:

ubuntu@athira-ims:~/emco/work$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16

[init] Using Kubernetes version: v1.28.2

[preflight] Running pre-flight checks

error execution phase preflight: [preflight] Some fatal errors occurred:

[ERROR Port-6443]: Port 6443 is in use

[ERROR Port-10259]: Port 10259 is in use

[ERROR Port-10257]: Port 10257 is in use

[ERROR FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube-controller-manager.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-scheduler.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etcd.yaml already exists

[ERROR Port-10250]: Port 10250 is in use

[ERROR Port-2379]: Port 2379 is in use

[ERROR Port-2380]: Port 2380 is in use

[ERROR DirAvailable--var-lib-etcd]: /var/lib/etcd is not empty

[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`

To see the stack trace of this error execute with --v=5 or higher

In this case, try to get PID of the port and kill the process using the following

1. *sudo ss -tulnp | grep <port number in use>*
2. *kill <pid from above command>*

[ERROR FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube-controller-manager.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-scheduler.yaml already exists

[ERROR FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etcd.yaml already exists

For this error, we need to remove these files.

Eg :

1. *rm -rf /etc/kubernetes/manifests/kube-apiserver.yaml*
2. *rm -rf /etc/kubernetes/manifests/kube-controller-manager.yaml*
3. *rm -rf /etc/kubernetes/manifests/kube-scheduler.yaml*
4. *rm -rf /etc/kubernetes/manifests/etcd.yaml*

Again, try **Step 1 of Create cluster**, if you can see the following output at end, the command is successful.

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.16.5.92:6443 --token 4bisk8.knkjve8ysj8jdh6s \
        --discovery-token-ca-cert-hash sha256:daba3d08785cf8b64fd56aabb589dd77e2e59f2d54e78c5828dfea63ee85f681
```

ISSUE 4:

If you are facing issues like /.kube/config file not found, please try Kubernetes installation from scratch.

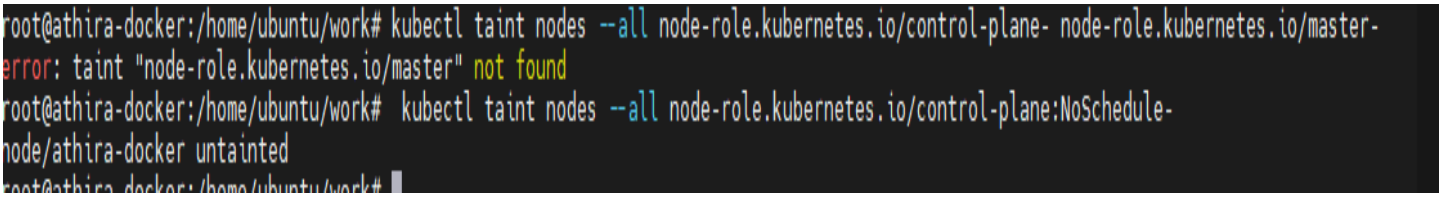
```
error: stat /home/ubuntu/.kube/config: no such file or directory
Installing EMCO DB. Please wait...
Error: INSTALLATION FAILED: Kubernetes cluster unreachable: stat /home/ubuntu/.kube/config: no such file o
Uninstalling emco
Error: Kubernetes cluster unreachable: stat /home/ubuntu/.kube/config: no such file or directory
Error: Kubernetes cluster unreachable: stat /home/ubuntu/.kube/config: no such file or directory
Error: Kubernetes cluster unreachable: stat /home/ubuntu/.kube/config: no such file or directory
Deleting namespace emco
```

ISSUE 5:

root@athira-ims:/home/ubuntu/emco/work# kubectl taint nodes --all node-role.kubernetes.io/control-plane- node-role.kubernetes.io/master-

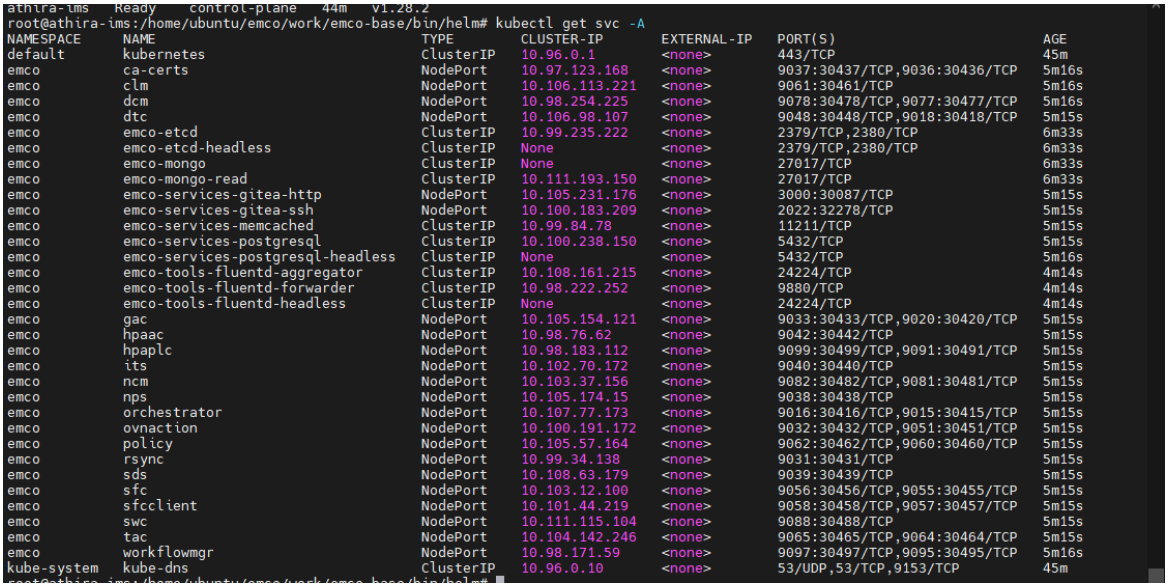
error: taint "node-role.kubernetes.io/master" not found

This issue is resolved in **Step 6 of Create cluster**



If you want to verify services later in the setup, use command:

kubectl get svc -A



```
athira-ims   Ready    control-plane   44m   v1.28.2
root@athira-ims:/home/ubuntu/emco/work/emco-base/bin/helm# kubectl get svc -A
NAMESPACE     NAME                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                          AGE
default       kubernetes                          ClusterIP   10.96.0.1        <none>        443/TCP                          45m
emco          ca-certs                            NodePort    10.97.123.168    <none>        9037:30437/TCP,9036:30436/TCP    5m16s
emco          clm                                 NodePort    10.106.113.221   <none>        9061:30461/TCP                   5m16s
emco          dcm                                 NodePort    10.98.254.225    <none>        9078:30478/TCP,9077:30477/TCP    5m16s
emco          dtc                                 NodePort    10.106.98.107    <none>        9048:30448/TCP,9018:30418/TCP    5m15s
emco          emco-etcd                           ClusterIP   10.99.235.222    <none>        2379/TCP,2380/TCP                6m33s
emco          emco-etcd-headless                  ClusterIP   None             <none>        2379/TCP,2380/TCP                6m33s
emco          emco-mongo                          ClusterIP   None             <none>        27017/TCP                        6m33s
emco          emco-mongo-read                     ClusterIP   10.111.193.150   <none>        27017/TCP                        6m33s
emco          emco-services-gitea-http            NodePort    10.105.231.176   <none>        3000:30087/TCP                   5m15s
emco          emco-services-gitea-ssh             NodePort    10.100.183.209   <none>        2022:32278/TCP                   5m15s
emco          emco-services-memcached             ClusterIP   10.99.84.78      <none>        11211/TCP                        5m15s
emco          emco-services-postgresql            ClusterIP   10.100.238.150   <none>        5432/TCP                         5m15s
emco          emco-services-postgresql-headless   ClusterIP   None             <none>        5432/TCP                         5m16s
emco          emco-tools-fluentd-aggregator       ClusterIP   10.108.161.215   <none>        24224/TCP                        4m14s
emco          emco-tools-fluentd-forwarder        ClusterIP   10.98.222.252    <none>        9880/TCP                         4m14s
emco          emco-tools-fluentd-headless         ClusterIP   None             <none>        24224/TCP                        4m14s
emco          gac                                 NodePort    10.105.154.121   <none>        9033:30433/TCP,9020:30420/TCP    5m15s
emco          hpaac                               NodePort    10.98.76.62      <none>        9042:30442/TCP                   5m15s
emco          hpaplc                              NodePort    10.98.183.112    <none>        9099:30499/TCP,9091:30491/TCP    5m15s
emco          its                                 NodePort    10.102.70.172    <none>        9040:30440/TCP                   5m15s
emco          ncm                                 NodePort    10.103.37.156    <none>        9082:30482/TCP,9081:30481/TCP    5m15s
emco          nps                                 NodePort    10.105.174.15    <none>        9038:30438/TCP                   5m15s
emco          orchestrator                        NodePort    10.107.77.173    <none>        9016:30416/TCP,9015:30415/TCP    5m15s
emco          ovnaction                           NodePort    10.100.191.172   <none>        9032:30432/TCP,9051:30451/TCP    5m15s
emco          policy                              NodePort    10.105.57.164    <none>        9062:30462/TCP,9060:30460/TCP    5m15s
emco          rsync                               NodePort    10.99.34.138     <none>        9031:30431/TCP                   5m15s
emco          sds                                 NodePort    10.108.63.179    <none>        9039:30439/TCP                   5m15s
emco          sfc                                 NodePort    10.103.12.100    <none>        9056:30456/TCP,9055:30455/TCP    5m15s
emco          sfcclient                           NodePort    10.101.44.219    <none>        9058:30458/TCP,9057:30457/TCP    5m15s
emco          swc                                 NodePort    10.111.115.104   <none>        9088:30488/TCP                   5m15s
emco          tac                                 NodePort    10.104.142.246   <none>        9065:30465/TCP,9064:30464/TCP    5m15s
emco          workflowmgr                         NodePort    10.98.171.59     <none>        9097:30497/TCP,9095:30495/TCP    5m16s
kube-system   kube-dns                            ClusterIP   10.96.0.10       <none>        53/UDP,53/TCP,9153/TCP           45m
root@athira-ims:/home/ubuntu/emco/work/emco-base/bin/helm#
```