

PCEI R6 Installation Guide

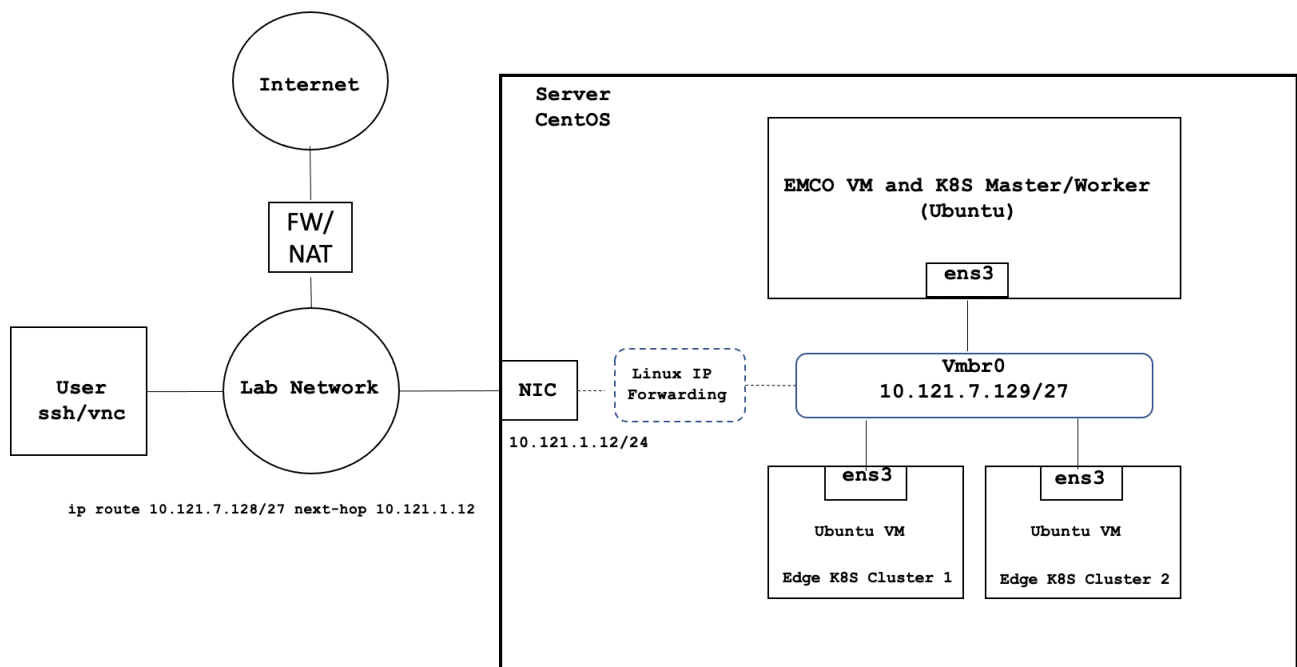
- [Introduction](#)
- [Deployment Architecture](#)
- [Pre-Installation Requirements](#)
 - [Recommended Hardware Requirements](#)
 - [Network Requirements](#)
 - [Software Prerequisites](#)
- [Installation High-Level Overview](#)
- [Deployment Guide](#)
 - [Install CentOS on Host Server](#)
 - [Prepare Host Server for EMCO Deployment](#)
 - [Step 1. Enable sudo without entering sudo password](#)
 - [Step 2. Add sudo user to wheel group](#)
 - [Step 3. Enable nested virtualization](#)
 - [Step 4. Install VNC Server \(Optional\)](#)
 - [Step 5. Modify libvirt bridge IP and route mode](#)
 - [Deploy EMCO](#)
 - [Step 1. Generate SSH Keys](#)
 - [Step 2. Download software and install EMCO \(AMCOP 2.2\)](#)
 - [Step 3. Monitor the installation](#)
 - [Step 4. Install Controller Blueprint Archives \(CBA\)](#)
 - [Update CDS py-executor](#)
 - [Kubernetes Cluster Registration CBA](#)
 - [Terraform Executor CBA](#)
 - [Helm Chart Processor CBA](#)
 - [Composite App Deployment Processor CBA](#)
 - [Ansible Playbook Executor CBA](#)
 - [Deploy Edge Clusters](#)
 - [Step 1. Edit VM creation script.](#)
 - [Step 2. Deploy two Edge Cluster VMs.](#)
 - [Step 3. Setup worker clusters inside VMs](#)
 - [Modify sshd_config on VMs](#)
- [Deployment Verification](#)
 - [EMCO Deployment Verification](#)
 - [Edge Cluster Deployment Verification](#)
- [Uninstall Guide](#)
- [License](#)
- [References](#)

Introduction

This document describes steps required to deploy a sample environment for the Public Cloud Edge Interface (PCEI) Blueprint.

Deployment Architecture

The deployment architecture is shown below. All Addressing and naming shown are for example purposes only.



Deployment environment components:

1. One Host Server
2. One Edge Multi-Cluster Orchestrator VM
3. Two Edge K8S Cluster VMs
4. Internal network connectivity to the Host Server and to the VMs

Pre-Installation Requirements

Recommended Hardware Requirements

Requirement	Value
CPU	18 Core / 36 vCPU
RAM	128 GB
DISK	500 GB
NETWORK	1 Gbps NIC Port

Network Requirements

- Internal Lab Network (RFC 1918 space)
- Internet access (NAT'd)
- IP routing in the Lab Network to reach VM interfaces

Software Prerequisites

Requirement	Value
Server OS	CentOS 7.x or above
VM OS	Ubuntu 18.04
Upstream SW	Refer to Deployment Guide Section

Installation High-Level Overview

The installation/deployment process consists of the following steps:

1. Install CentOS on Host Server.
2. Prepare Host Server for EMCO Deployment.
3. Deploy EMCO.
4. Deploy Edge Clusters.

Deployment Guide

Install CentOS on Host Server

1. Connect to Host Server iLO interface.
2. Start Virtual Console.
3. Mount Virtual Media with CentOS 7 ISO.
4. Install CentOS
 - a. Assign correct IP address, Subnet, Gateway and DNS to the NIC.
 - b. Include OpenSSH Server.
 - c. Install KVM/virtualization.
 - d. Add a user with admin privileges: **onaplab** user is used in this guide.

Prepare Host Server for EMCO Deployment

Step 1. Enable sudo without entering sudo password

```
sudo -i
visudo
# Uncomment the line below
%wheel  ALL=(ALL)        NOPASSWD: ALL
```

Step 2. Add sudo user to wheel group

```
usermod -aG wheel onaplab
```

Step 3. Enable nested virtualization

```
# Login as super user
sudo -i

# Check for the following parameter
cat /sys/module/kvm_intel/parameters/nested
N
# If it is Y, there is nothing else to be done. It is already enabled!

# If it is N, do the following:
# Edit /etc/default/grub file, and include kvm-intel.nested=1
GRUB_CMDLINE_LINUX parameter
GRUB_CMDLINE_LINUX="crashkernel=auto
console=ttyS0,38400n8,kvm-intel.nested=1"

# Rebuild your GRUB configuration
sudo grub2-mkconfig -o /boot/grub2/grub.cfg

# Enable nested KVM capabilities in /etc/modprobe.d/kvm.conf
# By uncommenting the below line

options kvm_intel nested=1

#### Reboot the server
reboot

# Now, we should have the nested KVM capabilities enabled
cat /sys/module/kvm_intel/parameters/nested
Y
```

Step 4. Install VNC Server (Optional)

Follow instructions at:

<https://www.tecmint.com/install-and-configure-vnc-server-in-centos-7/>

Step 5. Modify libvirt bridge IP and route mode

This will allow connecting to VMs and pods directly from the Lab Network. Please replace the sample IPs with your IP addresses. Please replace the interface name (eno24 used in the example) with you server's interface name.

```
cat <<\EOF >> netdefault.xml
<network>
  <name>default</name>
  <bridge name="vmb0" />
  <forward mode='route' dev='eno24' />
  <ip address="10.121.7.129" netmask="255.255.255.224">
    <dhcp>
      <range start="10.121.7.144" end="10.121.7.158" />
    </dhcp>
  </ip>
</network>
EOF

sudo virsh net-list
sudo virsh net-destroy default
sudo virsh net-undefine default
sudo virsh net-define netdefault.xml
sudo virsh net-start default
sudo virsh net-autostart default
```

Add necessary routes to your Lab Network routers. The example below assumes the the Host Server IP address is 10.121.1.12:

```
ip route 10.121.7.128 255.255.255.224 10.121.1.12
```

Deploy EMCO

Step 1. Generate SSH Keys

```
# Run commands below on the Host Server
ssh-keygen
cd ~/.ssh
chmod 600 id_rsa
chmod 600 id_rsa.pub
chmod 700 config
chmod 600 known_hosts

cat id_rsa.pub >> authorized_keys
chmod 600 authorized_keys

echo "# Increase the server timeout value" >> ~/.ssh/config
echo "ServerAliveInterval 120" >> ~/.ssh/config
```

Step 2. Download software and install EMCO (AMCOP 2.2)

Note that the install process will:

- Deploy a VM amcop-vm-01
- Create a K8S cluster inside the VM
- Deploy EMCO components on the K8S cluster
- Deploy ONAP components on the K8S cluster

```
#### On the Host Server

sudo yum install -y git deltarpm
mkdir -p amcop_deploy
cd amcop_deploy

## Download the installation package zip file

wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate 'https://docs.google.com/uc?export=download&id=1PMyc8yULDeTIY0xNvY0Rdf_7CSWvli3l' -O- | sed -rn 's/.*confirm=([0-9A-Za-z_]+).*/\1\n/p')&id=1PMyc8yULDeTIY0xNvY0Rdf_7CSWvli3l" -O amcop_install_v2.2.zip && rm -rf /tmp/cookies.txt

unzip amcop_install_v2.2.zip

sudo chown -R onaplab:onaplab ~/amcop_deploy/

cd ~/amcop_deploy/aarna-stream/util-scripts
./prep_baremetal_centos.sh

##### Install EMCO/AMCOP
# Edit inventory.ini file. Use IP address of Host Server and the username.

cd ~/amcop_deploy/aarna-stream/amcop_deploy/ansible/deployment
vi inventory.ini
[deployment_host]
10.121.1.12 ansible_user=onaplab

nohup ansible-playbook ./main.yml -i inventory.ini -e deployment_env=on-prem -e jump_host_user=onaplab --private-key=/home/onaplab/.ssh/id_rsa -e vm_user=onaplab &
```

Step 3. Monitor the installation

```
# On the Host Server

cd /home/onaplab/aarna-stream/anod_lite/logs
[onaplab@os12 logs]$ ls -l
total 1980
-rw-r--r--. 1 root    root      510417 Nov 24 07:06 cluster_setup.log
-rw-r--r--. 1 root    root        2019 Nov 24 06:54 create_vm.log
-rw-r--r--. 1 root    root    1366779 Nov 24 07:15 deploy_emco_components.log
-rw-r--r--. 1 root    root    138233 Nov 24 07:35 deploy_onap.log
-rw-rw-r--. 1 onaplab onaplab      83 Nov 24 06:53 README.md

tail -f create_vm.log

tail -f cluster_setup.log

tail -f deploy_emco_components.log

tail -f deploy_onap.log
```

If Install fails and you need to restart, please do the cleanup steps below on the Host Server.

```
sudo virsh destroy amcop-vm-01
sudo virsh undefine amcop-vm-01
sudo virsh pool-destroy amcop-vm-01
sudo virsh pool-undefine amcop-vm-01
sudo rm /var/lib/libvirt/images/amcop-vm-01/amcop-vm-01-cidata.iso
sudo rm /var/lib/libvirt/images/amcop-vm-01/amcop-vm-01.qcow2
```

Step 4. Install Controller Blueprint Archives (CBA)

Update CDS py-executor

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc>

Kubernetes Cluster Registration CBA

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/cds-blueprints/emco-api-executor>

Terraform Executor CBA

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/cds-blueprints/terraform-plan-executor>

Helm Chart Processor CBA

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/cds-blueprints/helm-chart-processor>

Composite App Deployment Processor CBA

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/cds-blueprints/composite-app-deployment-processor>

Ansible Playbook Executor CBA

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/cds-blueprints/ansible-playbook-executor>

Deploy Edge Clusters

Step 1. Edit VM creation script.

```
# On the Host Server
cd /home/onaplab/amcop_deploy/aarna-stream/util-scripts

# Add "--cpu host" option to the end of the below line
vi create_gem_vm.sh

virt-install --connect qemu:///system --name $vm_name --ram (($mem << 10)) --vcpus=$vCPU --os-type linux --os-variant $os_variant --disk path=/var/lib/libvirt/images/$vm_name/"$vm_name".qcow2,format=qcow2 --disk /var/lib/libvirt/images/$vm_name/$vm_name-cidata.iso,device=cdrom --import --network network=default --noautoconsole --cpu host

# Save the file
```

Step 2. Deploy two Edge Cluster VMs.

These commands will create two Ubuntu 18.04 VMs with 100G Disk, 8 vcpu and 16G RAM and will copy the contents of the ~/.ssh/id_rsa.pub key file from the Host Server to the VMs' ~/.ssh/authorized_keys file.

```
sudo ./create_gem_vm.sh 2 edge_k8s-1 100 8 16 ubuntu18.04 $HOME/.ssh/id_rsa.pub onaplab
sudo ./create_gem_vm.sh 2 edge_k8s-2 100 8 16 ubuntu18.04 $HOME/.ssh/id_rsa.pub onaplab
```

Step 3. Setup worker clusters inside VMs

```
# Find VM's IP addresses. On the Host Server run:
[onaplab@os12 ~]$ sudo virsh list --all
Id      Name                State
-----
 6      amcop-vm-01         running
 9      edge_k8s-1          running
10      edge_k8s-2          running

[onaplab@os12 ~]$ sudo virsh domifaddr edge_k8s-1
Name      MAC address          Protocol  Address
-----
vnet1     52:54:00:19:96:72    ipv4     10.121.7.152/27

[onaplab@os12 ~]$
[onaplab@os12 ~]$ sudo virsh domifaddr edge_k8s-2
Name      MAC address          Protocol  Address
-----
vnet2     52:54:00:c0:47:8b    ipv4     10.121.7.146/27

# ssh to each VM from the Host Server:
ssh onaplab@10.121.7.152
ssh onaplab@10.121.7.146

# Perform the following tasks in each VM:
sudo apt-get update -y
sudo apt-get upgrade -y

sudo apt-get install -y python-pip

git clone https://git.onap.org/multicloud/k8s/
# Run script to setup KUD clusters

nohup k8s/kud/hosting_providers/baremetal/aio.sh %
```

If the edge cluster deployment fails for any reason, please do the clean up steps below before you retry:

```
### Cleanup
sudo virsh destroy edge_k8s-1
sudo virsh undefine edge_k8s-1
sudo virsh pool-destroy edge_k8s-1
sudo virsh pool-undefine edge_k8s-1
sudo rm /var/lib/libvirt/images/edge_k8s-1/edge_k8s-1-cidata.iso
sudo rm /var/lib/libvirt/images/edge_k8s-1/edge_k8s-1.qcow2

sudo virsh destroy edge_k8s-2
sudo virsh undefine edge_k8s-2
sudo virsh pool-destroy edge_k8s-2
sudo virsh pool-undefine edge_k8s-2
sudo rm /var/lib/libvirt/images/edge_k8s-2/edge_k8s-2-cidata.iso
sudo rm /var/lib/libvirt/images/edge_k8s-2/edge_k8s-2.qcow2
```

Modify sshd_config on VMs

To ensure that user *onaplab* can successfully ssh into EMCO and edge cluster VMs, add user *onaplab* to the `sshd_config` file.

```
## ssh to each VM:
ssh onaplab@10.121.7.152

sudo -i
cd /etc/ssh
vi sshd_config
AllowUsers ubuntu onaplab

## Save the changes and exit the file
```

Deployment Verification

EMCO Deployment Verification

Perform the following steps to verify correct EMCO deployment:

```
# Determine IP address of EMCO VM:
[onaplab@os12 ~]$ sudo virsh list --all
```

ID	Name	State
6	amcop-vm-01	running
9	edge_k8s-1	running
10	edge_k8s-2	running

```
[onaplab@os12 ~]$ sudo virsh domifaddr amcop-vm-01
```

Name	MAC address	Protocol	Address
vnet0	52:54:00:1a:8e:8b	ipv4	10.121.7.145/27

```
# ssh to EMCO VM. You should be able to ssh without specifying the key:
[onaplab@os12 ~]$ ssh onaplab@10.121.7.145
```

```
# Verify K8S pods:
onaplab@emco:~$ kubectl get pods --all-namespaces
```

Namespace	Name	Ready	Status	Restarts	Age
kube-system	calico-kube-controllers-6f954885fb-bb2mr	1/1	Running	0	28d
kube-system	calico-node-ldcpv	1/1	Running	0	28d
kube-system	coredns-6b968665c4-6558h	1/1	Running	0	28d
kube-system	coredns-6b968665c4-rq6bf	0/1	Pending	0	28d
kube-system	dns-autoscaler-5fc5fdbf6-njch7	1/1	Running	0	28d
kube-system	kube-apiserver-node1	1/1	Running	0	28d
kube-system	kube-controller-manager-node1	1/1	Running	0	28d
kube-system	kube-proxy-gg95d	1/1	Running	0	28d

kube-system	kube-scheduler-node1	1/1	Running	0	28d
kube-system	kubernetes-dashboard-6c7466966c-cjpxm	1/1	Running	0	28d
kube-system	node-local-dns-7pxcs	1/1	Running	0	28d
kube-system	tiller-deploy-8756df4d9-zq52m	1/1	Running	0	28d
onap	dev-cassandra-0	1/1	Running	0	28d
onap	dev-cassandra-1	1/1	Running	0	28d
onap	dev-cassandra-2	1/1	Running	0	28d
onap	dev-cds-blueprints-processor-6d697cc4d6-wzlfj	0/1	Init:1/3	0	28d
onap	dev-cds-db-0	1/1	Running	0	28d
onap	dev-cds-py-executor-7dc5f7f6-tpfmg	1/1	Running	0	28d
onap	dev-cds-sdc-listener-f99d4588d-nt2tk	0/1	Init:0/1	4021	28d
onap	dev-cds-ui-7768bb4b-cfbzd	1/1	Running	0	28d
onap	dev-mariadb-galera-0	1/1	Running	0	28d
onap	dev-mariadb-galera-1	1/1	Running	0	28d
onap	dev-mariadb-galera-2	1/1	Running	0	28d
onap4k8s	clm-668c45d96d-99gpb	1/1	Running	0	28d
onap4k8s	emcoui-57846bd5df-c774f	1/1	Running	0	28d
onap4k8s	etcd-768d5b6cc-ptmmr	1/1	Running	0	28d
onap4k8s	middleend-6d67c9bf54-tvs7s	1/1	Running	0	28d
onap4k8s	mongo-7988cb488b-kf29q	1/1	Running	0	28d
onap4k8s	ncm-9f4b85787-nqnlm	1/1	Running	0	28d
onap4k8s	orchestrator-5fd4845f8f-qsxlf	1/1	Running	0	28d
onap4k8s	ovnaction-f794f65b6-w85ms	1/1	Running	0	28d
onap4k8s	rsync-7d9f5fbd9b-r72sp	1/1	Running	0	28d

Verify K8S services:

onaplab@emco:~\$ kubectl get svc --all-namespaces

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT
(S)		AGE			
default	kubernetes	ClusterIP	10.233.0.1	<none>	443
/TCP		28d			
kube-system	coredns	ClusterIP	10.233.0.3	<none>	53/UDP, 53/TCP, 9153
/TCP		28d			
kube-system	kubernetes-dashboard	ClusterIP	10.233.35.188	<none>	443
/TCP		28d			
kube-system	tiller-deploy	ClusterIP	10.233.33.249	<none>	44134
/TCP		28d			
onap	cassandra	ClusterIP	None	<none>	7000/TCP, 7001/TCP, 7199/TCP, 9042/TCP, 9160/TCP, 61621/TCP
		28d			
onap	cds-blueprints-processor-cluster	ClusterIP	10.233.4.219	<none>	5701
/TCP		28d			
onap	cds-blueprints-processor-grpc	ClusterIP	10.233.34.1	<none>	9111
/TCP		28d			
onap	cds-blueprints-processor-http	ClusterIP	10.233.31.74	<none>	8080
/TCP		28d			
onap	cds-db	ClusterIP	None	<none>	3306
/TCP		28d			
onap	cds-py-executor	ClusterIP	10.233.43.240	<none>	50052/TCP, 50053
/TCP		28d			
onap	cds-sdc-listener	ClusterIP	10.233.2.48	<none>	8080
/TCP		28d			
onap	cds-ui	NodePort	10.233.55.19	<none>	3000:30497
/TCP		28d			
onap	mariadb-galera	ClusterIP	None	<none>	3306
/TCP		28d			
onap4k8s	clm	NodePort	10.233.59.50	<none>	9061:31856
/TCP		28d			
onap4k8s	emcoui	NodePort	10.233.2.5	<none>	9080:30480
/TCP		28d			
onap4k8s	etcd	ClusterIP	10.233.54.80	<none>	2379/TCP, 2380
/TCP		28d			
onap4k8s	middleend	NodePort	10.233.11.225	<none>	9891:31289
/TCP		28d			
onap4k8s	mongo	ClusterIP	10.233.19.133	<none>	27017
/TCP		28d			
onap4k8s	ncm	NodePort	10.233.16.20	<none>	9031:32737
/TCP		28d			
onap4k8s	orchestrator	NodePort	10.233.23.25	<none>	9015:31298
/TCP		28d			
onap4k8s	ovnaction	NodePort	10.233.37.45	<none>	9053:32514/TCP, 9051:31181/TCP
		28d			

onap4k8s	rsync	NodePort	10.233.60.47	<none>	9041:30555
/TCP		28d			

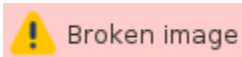
Access EMCOUI GUI:

```
# Determine EMCOUI Service Port:
onaplab@emco:~$ kubectl get svc emcoui -n onap4k8s
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
emcoui    NodePort  10.233.2.5   <none>        9080:30480/TCP   28d

# To connect to EMCOUI GUI use IP address of amcop-vm-01 and port 30480
```

To connect to EMCOUI GUI use IP address of amcop-vm-01 and port 30480:

<http://10.121.1.52:30480>



Edge Cluster Deployment Verification

To verify deployment of Edge Clusters, perform the following steps:

```
# Determine Edge Cluster VM IP addresses:
[onaplab@os12 ~]$ sudo virsh list --all
```

Id	Name	State
6	amcop-vm-01	running
9	edge_k8s-1	running
10	edge_k8s-2	running

```
[onaplab@os12 ~]$ sudo virsh domifaddr edge_k8s-1
```

Name	MAC address	Protocol	Address
vnet1	52:54:00:19:96:72	ipv4	10.121.7.152/27

```
[onaplab@os12 ~]$ sudo virsh domifaddr edge_k8s-2
```

Name	MAC address	Protocol	Address
vnet2	52:54:00:c0:47:8b	ipv4	10.121.7.146/27

```
# ssh to each VM from the Host Server. You should be able to ssh without specifying the key:
ssh onaplab@10.121.7.152
ssh onaplab@10.121.7.146

# Perform the following tasks inside the VMs:
onaplab@localhost:~$ kubectl get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	cmk-rpgd7	2/2	Running	0	28d
kube-system	coredns-dff8fc7d-2xwrk	0/1	Pending	0	28d
kube-system	coredns-dff8fc7d-q2gcr	1/1	Running	0	28d
kube-system	dns-autoscaler-66498f5c5f-2kzmv	1/1	Running	0	28d
kube-system	kube-apiserver-localhost	1/1	Running	0	28d
kube-system	kube-controller-manager-localhost	1/1	Running	0	28d
kube-system	kube-flannel-8rm9p	1/1	Running	0	28d
kube-system	kube-multus-ds-amd64-mt9s5	1/1	Running	0	28d
kube-system	kube-proxy-ggk8m	1/1	Running	0	28d
kube-system	kube-scheduler-localhost	1/1	Running	0	28d
kube-system	kubernetes-dashboard-84999f8b5b-48xjq	1/1	Running	0	28d
kube-system	kubernetes-metrics-scraper-54fbb4d595-rw649	1/1	Running	0	28d
kube-system	local-volume-provisioner-bmkc6	1/1	Running	0	28d
kube-system	virtlet-vk7jl	3/3	Running	0	28d
node-feature-discovery	nfd-master-78nms	1/1	Running	0	28d
node-feature-discovery	nfd-worker-k4d5g	1/1	Running	45	28d
operator	nfn-agent-zlp9g	1/1	Running	0	28d
operator	nfn-operator-b768877d8-vcx7v	1/1	Running	0	28d
operator	ovn4nfv-cni-4c6rx	1/1	Running	0	28d

```
# Verify connectivity to EMCO Cluster
onaplab@localhost:~$ ping 10.121.7.145
PING 10.121.7.145 (10.121.7.145) 56(84) bytes of data.
64 bytes from 10.121.7.145: icmp_seq=1 ttl=64 time=0.457 ms
64 bytes from 10.121.7.145: icmp_seq=2 ttl=64 time=0.576 ms
```

Uninstall Guide

Perform the following steps to remove EMCO and Edge Clusters from the Host Server:

```
sudo virsh destroy amcop-vm-01
sudo virsh undefine amcop-vm-01
sudo virsh pool-destroy amcop-vm-01
sudo virsh pool-undefine amcop-vm-01
sudo rm /var/lib/libvirt/images/amcop-vm-01/amcop-vm-01-cidata.iso
sudo rm /var/lib/libvirt/images/amcop-vm-01/amcop-vm-01.qcow2

sudo virsh destroy edge_k8s-1
sudo virsh undefine edge_k8s-1
sudo virsh pool-destroy edge_k8s-1
sudo virsh pool-undefine edge_k8s-1
sudo rm /var/lib/libvirt/images/edge_k8s-1/edge_k8s-1-cidata.iso
sudo rm /var/lib/libvirt/images/edge_k8s-1/edge_k8s-1.qcow2

sudo virsh destroy edge_k8s-2
sudo virsh undefine edge_k8s-2
sudo virsh pool-destroy edge_k8s-2
sudo virsh pool-undefine edge_k8s-2
sudo rm /var/lib/libvirt/images/edge_k8s-2/edge_k8s-2-cidata.iso
sudo rm /var/lib/libvirt/images/edge_k8s-2/edge_k8s-2.qcow2

sudo rm -rf ~/amcop_deploy
sudo rm -rf ~/aarna_stream
```

License

References

[AMCOP 2.0 Quickstart Guide \(Bare Metal\)](#) - EMC Install Guide by Aarna Networks
[AMCOP 2.0 User Guide](#) - EMC Config Guide by Aarna Networks