Cisco Secure Firewall ASA Container Getting Started Guide, 9.22

First Published: 2024-09-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CHAPTER 1

Deploy the ASA Container in a Docker Environment

You can deploy the ASA container (ASAc) in an open source Docker environment running on any cloud platform.

Overview

A container is a software package that bundles up code and associated requirements such as system libraries, system tools, default settings, runtime, and so on, to ensure that the application runs successfully in a computing environment. From Secure Firewall ASA version 9.22, you can deploy the ASA container (ASAc) in an open source Docker environment running on any cloud platform.

Guidelines and Limitations to Deploy ASA Container in Docker Environment

- The ASA container (ASAc) solution is validated on open-source Kubernetes and Docker environments only.
- Other Kubernetes frameworks such as EKS, GKE, AKS, OpenShift, are not validated yet.
- The following features are not validated:
    - Upgrade

- High Availability
- Cluster
- IPv6
- Transparent mode

Licenses to Deploy ASA Container in Docker Environment

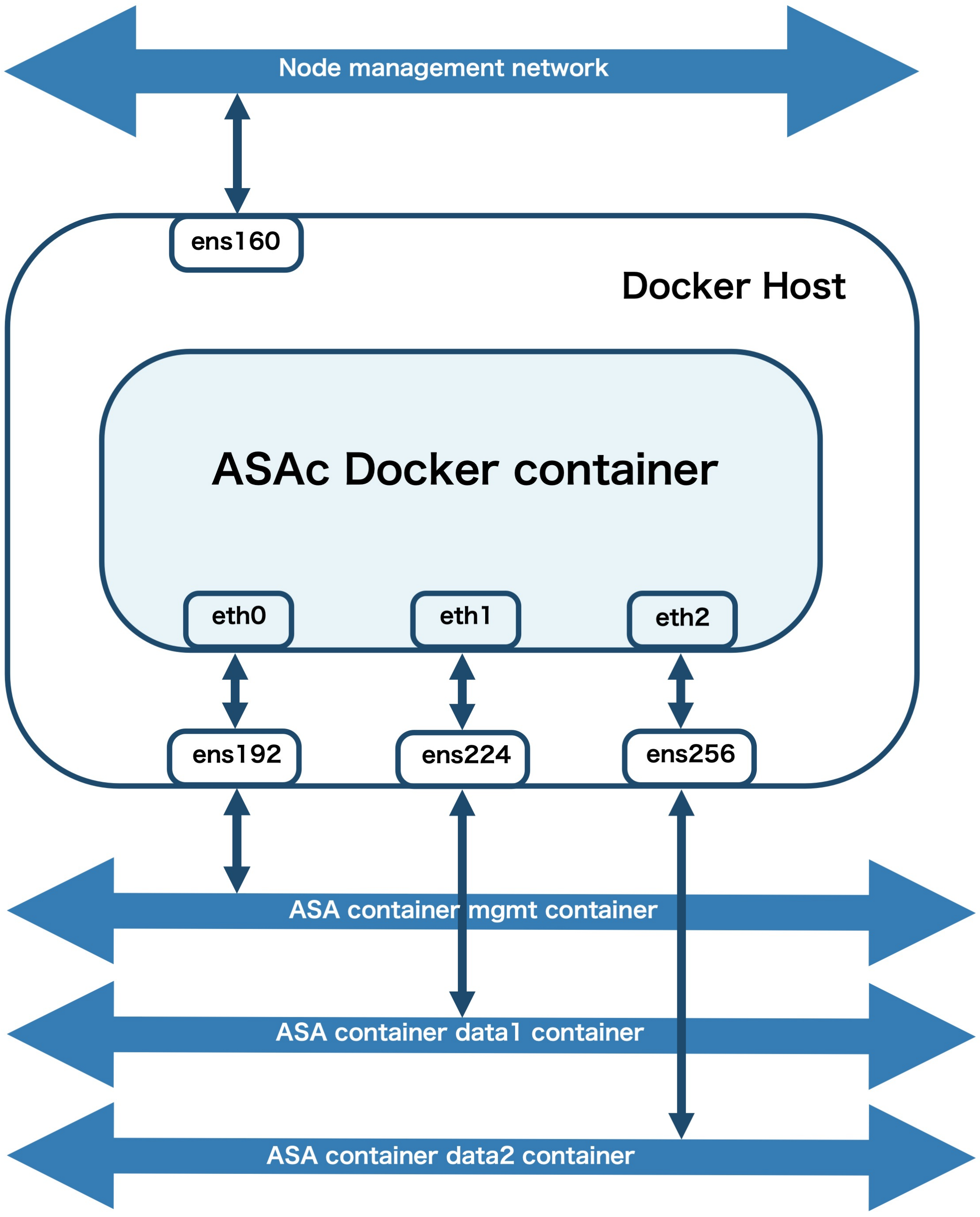
Use one of the following licenses to enable deployment of ASA container on Docker:

- ASAc5 - 1 vCPU, 2 GB RAM, and 100 Mbps rate limit
- ASAc10 - 2 vCPUs, 4 GB RAM, and 1 Gbps rate limit

Components of Solution to Deploy ASA Container in Docker Environment

- Operating system
    - Ubuntu 20.04.6 LTS on docker host
- Macvlan network for configuration validation

Sample Topology to Deploy ASA Container in Docker Environment



In this sample topology, the ASA docker container has three virtual network interfaces –eth0, eth1, and eth2, that are connected to the following interfaces – ens192, ens224, and ens256. These interfaces are mapped to the ASAc mgmt, data1, and data2 networks. The interface ens160 is the node management interface.

Prerequisites to Deploy ASA Container in Docker Environment

- Ensure that Ubuntu 20.04.6 LTS is installed on the docket host.
- Allocate three virtual interfaces on the docker host for ASA container operations.
- Set up the docker host's management interface to be used for ssh access to the docker host.
- Enable Hugepages on the docker host.
- Set up Docker version 24.0.5 with macvlan network for configuration validation.

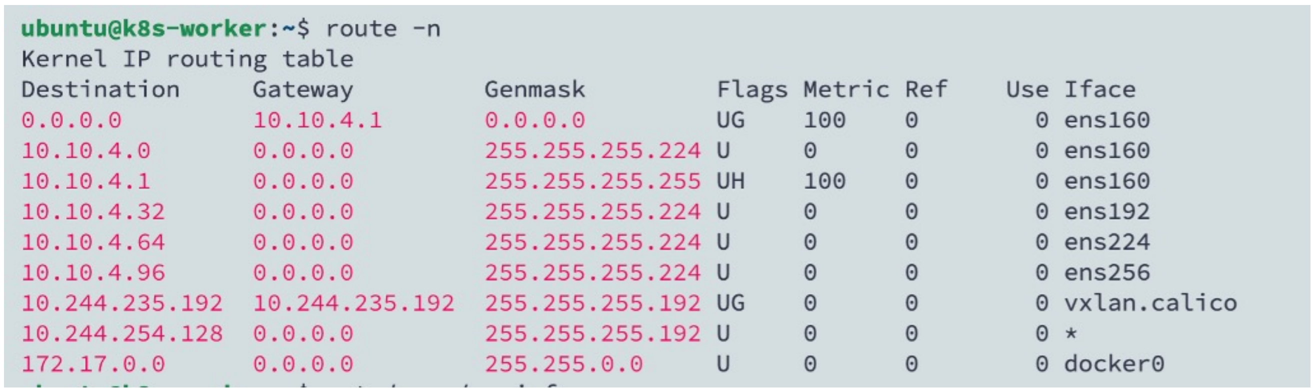
For more information on general Docker operations mentioned in these prerequisites, see Docker documentation.

Deploy ASA Container in Docker Environment

Perform the procedure given below to deploy ASA container (ASAc) in Docker environment.

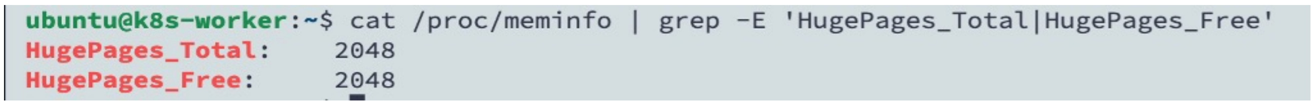
Step 1 Set up the requirements mentioned in the Prerequisites to Deploy ASA Container in Docker Environment.

Step 2 Run the **route -n** command to verify the network interface configuration. In this example, ens160 is the node's management interface. The nodes ens192, ens224, and ens256, are mapped to the ASAc interfaces.



```
ubuntu@k8s-worker:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.10.4.1       0.0.0.0         UG    100    0        0 ens160
10.10.4.0       0.0.0.0         255.255.255.224 U     0      0        0 ens160
10.10.4.1       0.0.0.0         255.255.255.255 UH    100    0        0 ens160
10.10.4.32      0.0.0.0         255.255.255.224 U     0      0        0 ens192
10.10.4.64      0.0.0.0         255.255.255.224 U     0      0        0 ens224
10.10.4.96      0.0.0.0         255.255.255.224 U     0      0        0 ens256
10.244.235.192  10.244.235.192  255.255.255.192 UG    0      0        0 vxlan.calico
10.244.254.128  0.0.0.0         255.255.255.192 U     0      0        0 *
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
```

Step 3 Run the **cat** command given below to verify hugepage configuration.



```
ubuntu@k8s-worker:~$ cat /proc/meminfo | grep -E 'HugePages_Total|HugePages_Free'
HugePages_Total:     2048
HugePages_Free:      2048
```

Step 4 Download the ASAc image from the Cisco docker hub repository - dockerhub.cisco.com/asac.

```
ubuntu@k8s-worker: ~$ docker pull dockerhub.cisco.com/asacdev-docker/asac:9.22.x.x
```

Step 5 Download the templates and other files from the **asac-docker** folder in the ASAc GitHub repository.

Step 6 Run the **docker network create** command to create docker networks. The ASAc needs one management interface and two date interfaces for inside and outside networks. When docker starts, the docker networks are attached to the docker in alphabetical order. We recommend that you name the management interface in such a way that it is the first interface that is attached to the docker.

```
$ docker network create -d macvlan -o parent=ens192 asac_nw1
$ docker network create -d macvlan -o parent=ens224 asac_nw2
$ docker network create -d macvlan -o parent=ens256 asac_nw3
```

Step 7 Run the **docker network ls** command to verify that the networks have been created successfully.

```
$ docker network ls
NETWORK ID    NAME      DRIVER   SCOPE
06f5320016f8  asac_nw1  macvlan  local
258954fa5611  asac_nw2  macvlan  local
3a3cd7254087  asac_nw3  macvlan  local
```

Step 8 Verify the default parameter values present in the **day0-config** file. You can also update these values as per your requirement.

Step 9 Open the **start_docker_asac.sh** script to update configuration values for CPU, memory, container-name, and image repo name, as per your requirement.

Note Default configuration values are provided for the parameters in the start_docker_asac.sh script. Modify them only if required.

Step 10 Run the command given below to start ASAc in the docker environment.

```
$ ./<script-name> <asac-version> <asac-mgmt-nw> <asac-data1-nw> <asac-data2-nw>
```

```
$ ./start_docker_asac.sh 9.22.x.x asac_nw1 asac_nw2 asac_nw3
Docker networks are provided..
Starting ASA Build Container...
docker create -it --privileged --cap-add=NET_RAW --network asac_nw1 --name asac -e ASAC_CPUS=1 -e
ASAC_MEMORY=2048M -v /dev:/dev -v
/home/ubuntu/standalone-asac/asac-docker/day0-config:/asac-day0-config/day0-
config:Z -v
/home/ubuntu/standalone-asac/asac-docker/interface-config:/mnt/disk0/interface-config/interfaceconfig:
Z -e CORE_SIZE_LIMIT=200MB -e COREDUMP_PATH=/mnt/coredump_repo/ -e ASA_DOCKER=1 -e
ASAC_STANDALONE_MODE=1 -e
ASAC_ROOT_PRIVILEGE=1 --entrypoint /asa/bin/lina_launcher.sh
dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x
Mount Points:
----------------------------------------------------------------------------------------
Host                                                          Container
----                                                          ---------
/dev                                                          /dev
/home/ubuntu/standalone-asac/asac-docker/day0-config          /asac-day0-config/day0-config
/home/ubuntu/standalone-asac/asac-docker/interface-config
/mnt/disk0/interface-config/interface-config
----------------------------------------------------------------------------------------
docker network connect asac_nw2 asac
docker network connect asac_nw3 asac
docker start asac
```

Validate ASA Container Deployment in Docker Environment

Validate successful ASA container deployment by checking the list of containers running on the docker host.

```
$ docker ps -a
CONTAINER ID IMAGE                                                  COMMAND
 CREATED       STATUS        PORTS  NAMES
6e5bff4dbcaf dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x   "/asa/bin/lina_launc…" 3
minutes ago Up 3 minutes         asac
```

Access ASA Container Deployment Logs in Docker Environment

Run the **docker logs asac** command to check the docker logs for troubleshooting any issues that may occur.

```
$ docker logs asac
Skip NVMe Device for ASAc mode
cdrom device /dev/sr0 found
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
Error: Encrypted file system support not in Linux kernel.
nr_overcommit_hugepages set to 128 for virtual platform
info: ASAc SSHd Directory Created
No interface-config file found at /interface-config, using default shared
file: /mnt/disk0/interface-config/interface-config
No day0-config file found at /day0-config, using default shared file:
/asac-day0-config/day0-config
info: ASAc Day 0 configuration installed.
info: ASAc Primay/backup Key installed
info: Running in vmware virtual environment.
....
INFO: Network Service reload not performed.
INFO: Power-On Self-Test in process.
.....................................
INFO: Power-On Self-Test complete.
INFO: Starting SW-DRBG health test...
INFO: SW-DRBG health test passed.
Creating trustpoint "_SmartCallHome_ServerCA" and installing certificate...
Trustpoint CA certificate accepted.
Creating trustpoint "_SmartCallHome_ServerCA2" and installing
certificate...
Trustpoint CA certificate accepted.
User enable_1 logged in to ciscoasa
Logins over the last 1 days: 1.
Failed logins since the last login: 0.
Type help or '?' for a list of available commands.
ciscoasa>
```

Access ASA Container in Docker Environment

Run the **docker attach asac** command to access the CLI of the ASA container (ASAc) and obtain required outputs. In this example, we access the CLI of the ASAc and run the **show version** command.

```
ciscoasa> enable
Password: *********
ciscoasa# sh version
```

```
Cisco Adaptive Security Appliance Software Version 9.22
SSP Operating System Version 82.16(0.216i)
Device Manager Version 7.22
Compiled on Tue 28-Nov-23 14:37 GMT by builders
System image file is "Unknown, monitor mode tftp booted image"
Config file at boot was "startup-config"
ciscoasa up 9 mins 50 secs
Start-up time 36 secs
Hardware: ASAc, 2048 MB RAM, CPU Xeon E5 series 2100 MHz, 1 CPU (1
core)
BIOS Flash Firmware Hub @ 0x1, 0KB
0: Ext: Management0/0 : address is 0242.ac12.0002, irq 0
1: Ext: GigabitEthernet0/0 : address is 0242.ac13.0002, irq 0
2: Ext: GigabitEthernet0/1 : address is 0242.ac14.0002, irq 0
3: Int: Internal-Data0/0 : address is 0000.0100.0001, irq 0
```

CHAPTER 2

Deploy the ASA Container in a Kubernetes Environment

You can deploy the ASA container (ASAc) in an open source Kubernetes environment running on any cloud platform.

Overview

A container is a software package that bundles up code and associated requirements such as system libraries, system tools, default settings, runtime, and so on, to ensure that the application runs successfully in a computing environment. From Secure Firewall ASA version 9.22, you can deploy the ASAc in an open source Kubernetes environment running on any cloud platform. In this solution, the ASAc is integrated with the Container Network Interface (CNI) and is deployed as an Infrastructure-as-Code (IaC) solution. The integration with CNI provides improved flexibility in deployment of network infrastructure.

Guidelines and Limitations to Deploy ASA Container in Kubernetes Environment

- The ASA container solution is validated on open-source Kubernetes and Docker environments only.
- Other Kubernetes frameworks such as EKS, GKE, AKS, OpenShift, are not validated yet.
- The following features are not validated:

- Upgrade
- High Availability
- Cluster
- IPv6
- Transparent mode

Licenses to Deploy ASA Container in Kubernetes Environment

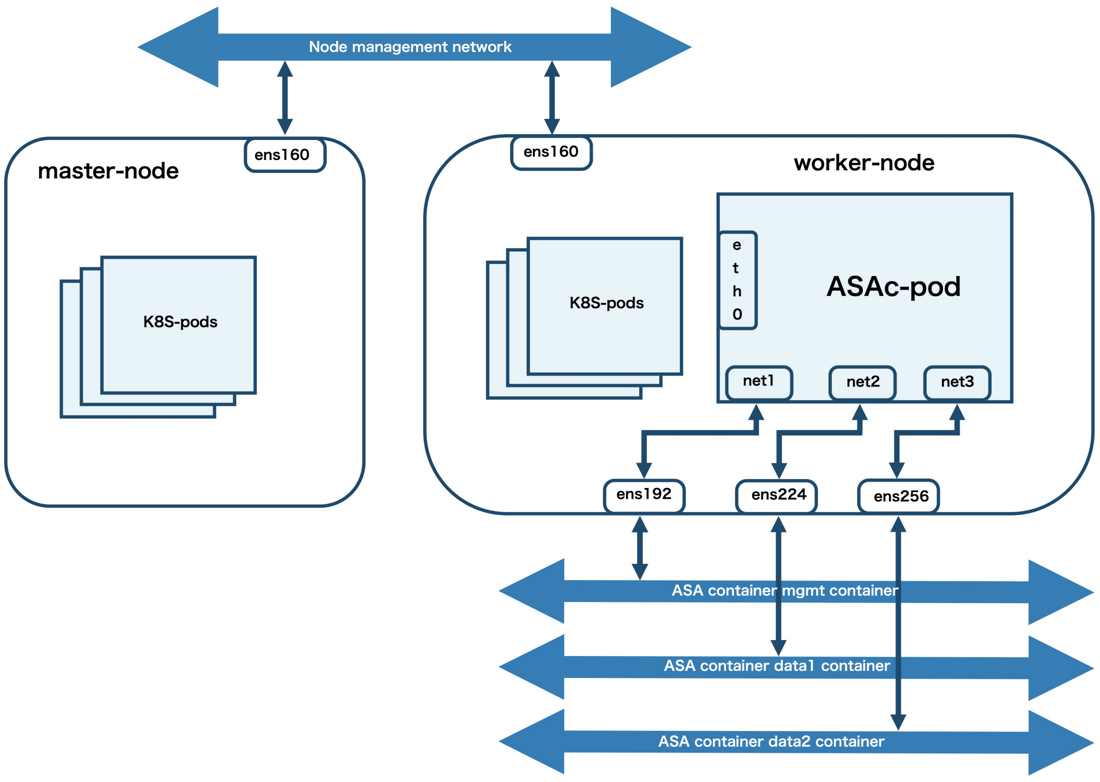
Use one of the following licenses to enable deployment of ASA container on Kubernetes:

- ASAc5 - 1 vCPU, 2 GB RAM, and 100 Mbps rate limit
- ASAc10 - 2 vCPUs, 4 GB RAM, and 1 Gbps rate limit

Components of Solution to Deploy ASA Container in Kubernetes Environment

- Operating system
    - Ubuntu 20.04.6
    - Kubernetes version v1.26
    - Helm version v3.13.1
- Kubernetes cluster nodes – master and worker nodes
- Kubernetes CNI
    - POD management CNI - Calico
    - ASAc network CNI - Multus macvlan
- Helm charts provided as yaml files are used to set up Infrastructure-as-Code (IaC)

Sample Topology to Deploy ASA Container in Kubernetes Environment



In this sample topology, the ASA container (ASAc) pod has three virtual network interfaces – net1, net2, and net3, that are connected to the following worker node interfaces – ens192, ens224, and ens256. The worker node interfaces are mapped to the ASAc mgmt, data1, and data2 networks. The interface ens160 is the node management interface. The interface eth0 is derived from the Calico CNI. The interfaces net1, net2, and net3, are derived from the multus macvlan CNI.

Prerequisites to Deploy ASA Container in Kubernetes Environment

- Ensure that Ubuntu 20.04.6 LTS is installed on both master and worker nodes.
- Allocate three virtual interfaces on the worker node for ASA container (ASAc) operations.
- Set up the worker node’s management interface to be used for ssh access to the worker node.
- Enable Hugepages on the worker node.
- Set up the Calico CNI to be used as POD management.

- Set up Multus with macvlan CNI to be used for managing ASAc interfaces.

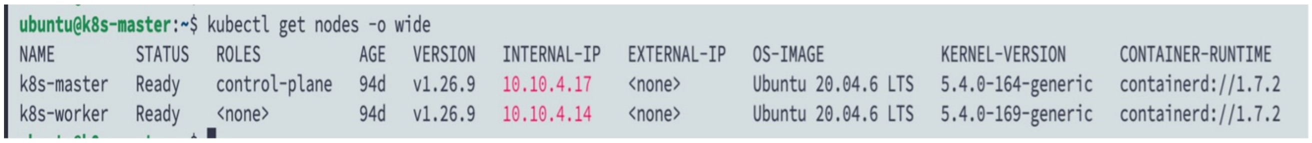
For more information on general Kubernetes operations mentioned in these prerequisites, see Kubernetes documentation.

Deploy ASA Container in Kubernetes Environment

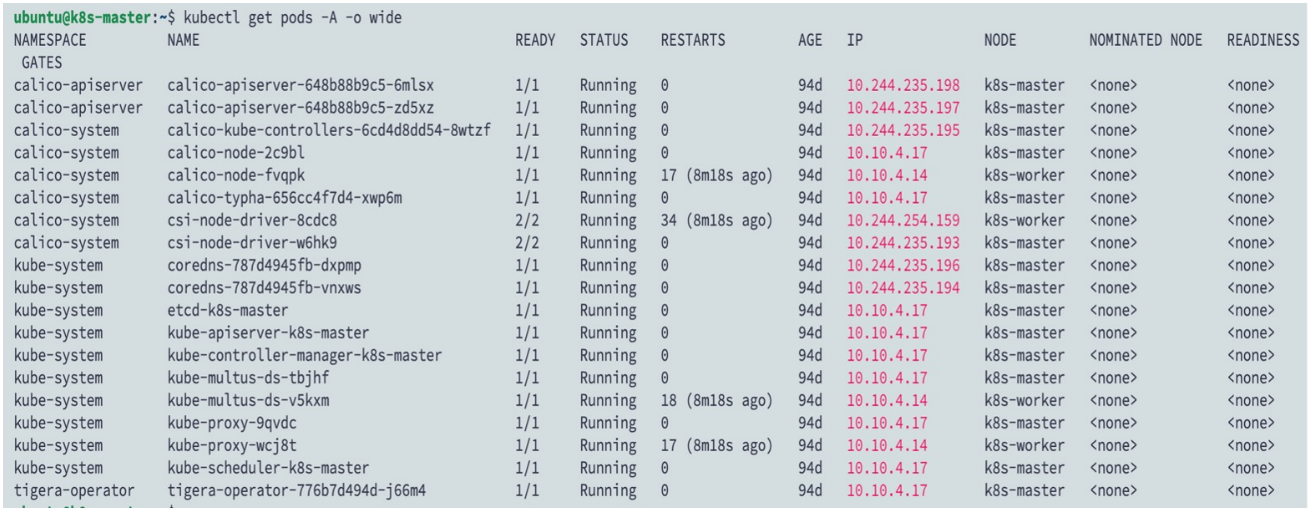
Perform the procedure given below to deploy ASA container (ASAc) in Kubernetes environment.

Step 1 Set up the requirements mentioned in the Prerequisites to Deploy ASA Container in Kubernetes Environment.

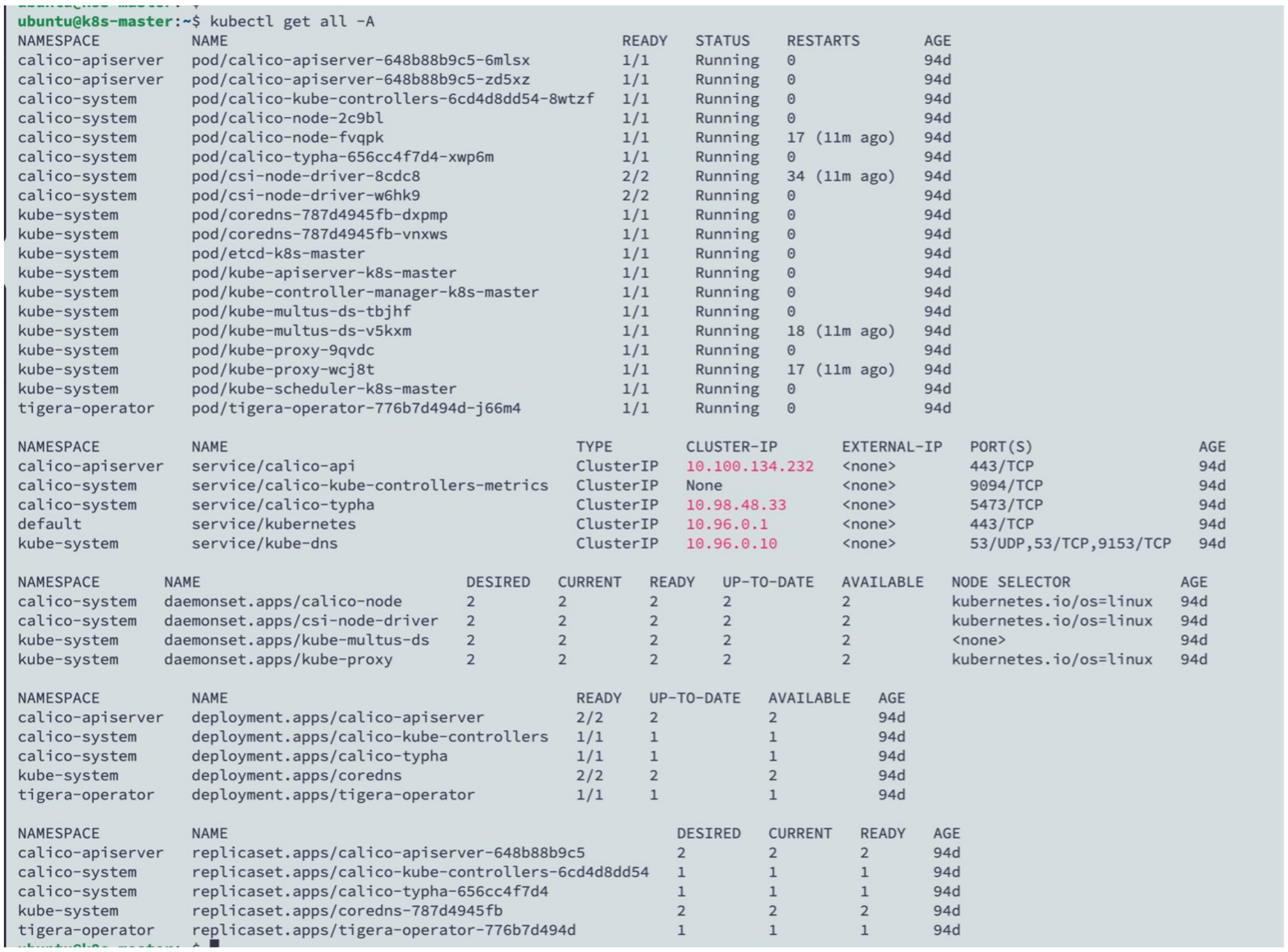
Step 2 Run the **kubectl get nodes**, **kubectl get pods**, and **kubectl get all** commands, to display the status of all nodes, pods, and all resources, respectively.



```
ubuntu@k8s-master:~$ kubectl get nodes -o wide
NAME         STATUS   ROLES           AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
k8s-master   Ready    control-plane   94d   v1.26.9   10.10.4.17    <none>        Ubuntu 20.04.6 LTS   5.4.0-164-generic   containerd://1.7.2
k8s-worker   Ready    <none>          94d   v1.26.9   10.10.4.14    <none>        Ubuntu 20.04.6 LTS   5.4.0-169-generic   containerd://1.7.2
```

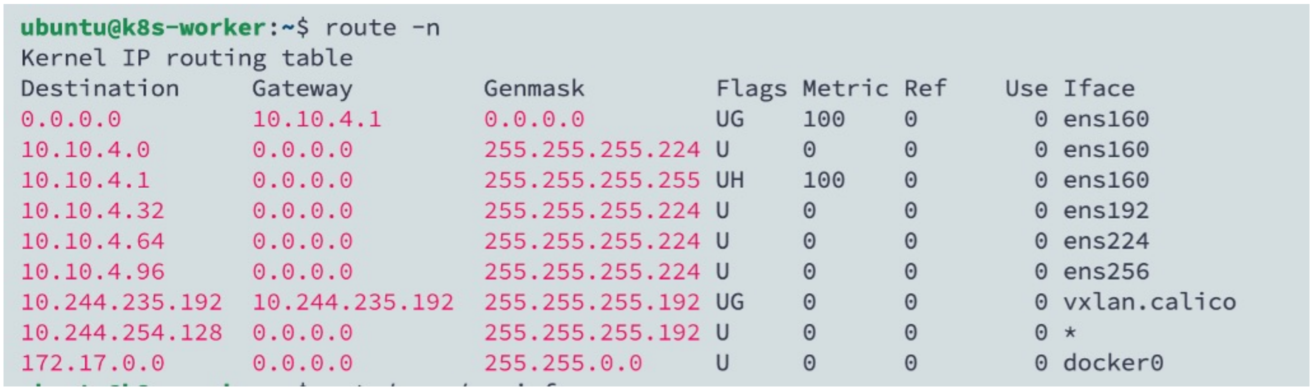


```
ubuntu@k8s-master:~$ kubectl get pods -A -o wide
NAMESPACE          NAME                                       READY   STATUS    RESTARTS        AGE   IP               NODE         NOMINATED NODE   READINESS
 GATES
calico-apiserver   calico-apiserver-648b88b9c5-6mlsx          1/1     Running   0               94d   10.244.235.198   k8s-master   <none>           <none>
calico-apiserver   calico-apiserver-648b88b9c5-zd5xz          1/1     Running   0               94d   10.244.235.197   k8s-master   <none>           <none>
calico-system      calico-kube-controllers-6cd4d8dd54-8wtzf   1/1     Running   0               94d   10.244.235.195   k8s-master   <none>           <none>
calico-system      calico-node-2c9bl                          1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
calico-system      calico-node-fvqpk                          1/1     Running   17 (8m18s ago)  94d   10.10.4.14       k8s-worker   <none>           <none>
calico-system      calico-typha-656cc4f7d4-xwp6m              1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
calico-system      csi-node-driver-8cdc8                      2/2     Running   34 (8m18s ago)  94d   10.244.254.159   k8s-worker   <none>           <none>
calico-system      csi-node-driver-w6hk9                      2/2     Running   0               94d   10.244.235.193   k8s-master   <none>           <none>
kube-system        coredns-787d4945fb-dxpmp                   1/1     Running   0               94d   10.244.235.196   k8s-master   <none>           <none>
kube-system        coredns-787d4945fb-vnxws                   1/1     Running   0               94d   10.244.235.194   k8s-master   <none>           <none>
kube-system        etcd-k8s-master                            1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
kube-system        kube-apiserver-k8s-master                  1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
kube-system        kube-controller-manager-k8s-master         1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
kube-system        kube-multus-ds-tbjhf                       1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
kube-system        kube-multus-ds-v5kxm                       1/1     Running   18 (8m18s ago)  94d   10.10.4.14       k8s-worker   <none>           <none>
kube-system        kube-proxy-9qvdc                           1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
kube-system        kube-proxy-wcj8t                           1/1     Running   17 (8m18s ago)  94d   10.10.4.14       k8s-worker   <none>           <none>
kube-system        kube-scheduler-k8s-master                  1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
tigera-operator    tigera-operator-776b7d494d-j66m4           1/1     Running   0               94d   10.10.4.17       k8s-master   <none>           <none>
```



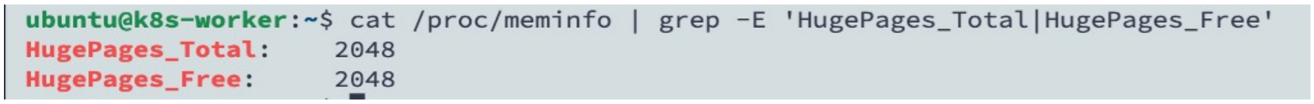
```
ubuntu@k8s-master:~$ kubectl get all -A
NAMESPACE          NAME                                         READY   STATUS    RESTARTS       AGE
calico-apiserver   pod/calico-apiserver-648b88b9c5-6mlsx        1/1     Running   0              94d
calico-apiserver   pod/calico-apiserver-648b88b9c5-zd5xz        1/1     Running   0              94d
calico-system      pod/calico-kube-controllers-6cd4d8dd54-8wtzf 1/1     Running   0              94d
calico-system      pod/calico-node-2c9bl                        1/1     Running   0              94d
calico-system      pod/calico-node-fvqpk                        1/1     Running   17 (11m ago)   94d
calico-system      pod/calico-typha-656cc4f7d4-xwp6m            1/1     Running   0              94d
calico-system      pod/csi-node-driver-8cdc8                    2/2     Running   34 (11m ago)   94d
calico-system      pod/csi-node-driver-w6hk9                    2/2     Running   0              94d
kube-system        pod/coredns-787d4945fb-dxpmp                 1/1     Running   0              94d
kube-system        pod/coredns-787d4945fb-vnxws                 1/1     Running   0              94d
kube-system        pod/etcd-k8s-master                          1/1     Running   0              94d
kube-system        pod/kube-apiserver-k8s-master                1/1     Running   0              94d
kube-system        pod/kube-controller-manager-k8s-master       1/1     Running   0              94d
kube-system        pod/kube-multus-ds-tbjhf                     1/1     Running   0              94d
kube-system        pod/kube-multus-ds-v5kxm                     1/1     Running   18 (11m ago)   94d
kube-system        pod/kube-proxy-9qvdc                         1/1     Running   0              94d
kube-system        pod/kube-proxy-wcj8t                         1/1     Running   17 (11m ago)   94d
kube-system        pod/kube-scheduler-k8s-master                1/1     Running   0              94d
tigera-operator    pod/tigera-operator-776b7d494d-j66m4         1/1     Running   0              94d

NAMESPACE          NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
calico-apiserver   service/calico-api                        ClusterIP   10.100.134.232   <none>        443/TCP                  94d
calico-system      service/calico-kube-controllers-metrics   ClusterIP   None             <none>        9094/TCP                 94d
calico-system      service/calico-typha                      ClusterIP   10.98.48.33      <none>        5473/TCP                 94d
default            service/kubernetes                        ClusterIP   10.96.0.1        <none>        443/TCP                  94d
kube-system        service/kube-dns                          ClusterIP   10.96.0.10       <none>        53/UDP,53/TCP,9153/TCP   94d

NAMESPACE       NAME                             DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
calico-system   daemonset.apps/calico-node       2         2         2       2            2           kubernetes.io/os=linux   94d
calico-system   daemonset.apps/csi-node-driver   2         2         2       2            2           kubernetes.io/os=linux   94d
kube-system     daemonset.apps/kube-multus-ds    2         2         2       2            2           <none>                   94d
kube-system     daemonset.apps/kube-proxy        2         2         2       2            2           kubernetes.io/os=linux   94d

NAMESPACE          NAME                                      READY   UP-TO-DATE   AVAILABLE   AGE
calico-apiserver   deployment.apps/calico-apiserver          2/2     2            2           94d
calico-system      deployment.apps/calico-kube-controllers   1/1     1            1           94d
calico-system      deployment.apps/calico-typha              1/1     1            1           94d
kube-system        deployment.apps/coredns                   2/2     2            2           94d
tigera-operator    deployment.apps/tigera-operator           1/1     1            1           94d

NAMESPACE          NAME                                                 DESIRED   CURRENT   READY   AGE
calico-apiserver   replicaset.apps/calico-apiserver-648b88b9c5          2         2         2       94d
calico-system      replicaset.apps/calico-kube-controllers-6cd4d8dd54   1         1         1       94d
calico-system      replicaset.apps/calico-typha-656cc4f7d4              1         1         1       94d
kube-system        replicaset.apps/coredns-787d4945fb                   2         2         2       94d
tigera-operator    replicaset.apps/tigera-operator-776b7d494d           1         1         1       94d
```

Step 3 Run the **route -n** command to verify the network interface configuration. In this example, ens160 is the node's management interface. The nodes ens192, ens224, and ens256, are mapped to the ASAc interfaces.



```
ubuntu@k8s-worker:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.10.4.1       0.0.0.0         UG    100    0        0 ens160
10.10.4.0       0.0.0.0         255.255.255.224 U     0      0        0 ens160
10.10.4.1       0.0.0.0         255.255.255.255 UH    100    0        0 ens160
10.10.4.32      0.0.0.0         255.255.255.224 U     0      0        0 ens192
10.10.4.64      0.0.0.0         255.255.255.224 U     0      0        0 ens224
10.10.4.96      0.0.0.0         255.255.255.224 U     0      0        0 ens256
10.244.235.192  10.244.235.192  255.255.255.192 UG    0      0        0 vxlan.calico
10.244.254.128  0.0.0.0         255.255.255.192 U     0      0        0 *
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
```

Step 4 Run the **cat** command given below to verify hugepage configuration.



```
ubuntu@k8s-worker:~$ cat /proc/meminfo | grep -E 'HugePages_Total|HugePages_Free'
HugePages_Total:    2048
HugePages_Free:     2048
```

Step 5 Download the ASAc image from the Cisco docker hub repository - dockerhub.cisco.com/asac.

```
ubuntu@k8s-worker: ~$ docker pull dockerhub.cisco.com/asacdev-docker/asac:9.22.x.x
```

Step 6 Download the templates and other files from the **asac-helm** folder in the ASAc GitHub repository.

Step 7 Enter the required parameter values in the values.yaml file.

```
Default values for asac-helm.
This is a YAML-formatted file.
Declare variables to be passed into your templates.
replicas: 1
image:
repository: dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x
#repository: localhost:5000/asac_9.22.x.x
persistVolPath: /home/ubuntu/pod-path
asacMgmtInterface: "ens192"
asacInsideInterface: "ens224"
asacOutsideInterface: "ens256"
```

The parameter names along with descriptions for the parameters in the values.yaml file are given below.

Variable Name	Description
repository	ASAc image path from Cisco's docker hub link.
persistVolPath	Valid path from the worker node in which the persistent configuration file from the ASAc is stored.
asacMgmtInterface	Name of the worker node interface that is used as the ASAc management interface.
asacInsideInterface	Name of the worker node interface that is used as the ASAc inside data interface.
asacOutsideInterface	Name of the worker node interface that is used as the ASAc outside data interface.

Step 8 Verify the default parameter values present in the **day0-config** file. You can also update these values as per your requirement.

Step 9 Run the **helm install** command to deploy the helm charts and deploy ASAc in the Kubernetes framework.

```
$ helm install test-asac asac-helm
NAME: test-asac
LAST DEPLOYED: Sun Jan 21 07:41:03 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Step 10 Run the **helm list -all** command to list the deployed resources and check the status of the ASAc deployment.

```
$ helm list -all
NAME        NAMESPACE    REVISION   UPDATED                                    STATUS     CHART
         APP VERSION
test-asac   default      1          2024-01-21 07:41:03.175728953 +0000 UTC   deployed
asac-helm-0.1.0   1.16.0
```

Validate ASA Container Deployment in Kubernetes Environment

Validate successful ASA container (ASAc) deployment by checking the status of the helm chart, ASAc pod, and by going through the pod events.

```
ubuntu@k8s-master:~$ helm status test-asac
NAME: test-asac
LAST DEPLOYED: Sun Jan 21 07:41:03 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None


ubuntu@k8s-master:~$ kubectl get pod
NAME                     READY   STATUS    RESTARTS   AGE
asac-5d8c4d547f-6k479   1/1     Running   0          43m



ubuntu@k8s-master:~$ kubectl events asac-5d8c4d547f-6k479
LAST SEEN   TYPE     REASON                  OBJECT                                MESSAGE
52m         Normal   SuccessfulCreate        ReplicaSet/asac-5d8c4d547f            Created pod:
 asac-5d8c4d547f-6k479
52m         Normal   ScalingReplicaSet       Deployment/asac                       Scaled up
replica set asac-5d8c4d547f to 1
52m         Normal   WaitForFirstConsumer    PersistentVolumeClaim/local-pvc       waiting for
first consumer to be created before binding
51m         Normal   Scheduled               Pod/asac-5d8c4d547f-6k479             Successfully
 assigned default/asac-5d8c4d547f-6k479 to k8s-worker
51m         Normal   AddedInterface          Pod/asac-5d8c4d547f-6k479             Add eth0
[10.244.254.160/32] from k8s-pod-network
51m         Normal   AddedInterface          Pod/asac-5d8c4d547f-6k479             Add net1 []
from default/macvlan-mgmt-bridge
51m         Normal   AddedInterface          Pod/asac-5d8c4d547f-6k479             Add net2 []
from default/macvlan-in-bridge
51m         Normal   AddedInterface          Pod/asac-5d8c4d547f-6k479             Add net3 []
from default/macvlan-out-bridge
51m         Normal   Pulling                 Pod/asac-5d8c4d547f-6k479             Pulling image
 "dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x"
50m         Normal   Pulled                  Pod/asac-5d8c4d547f-6k479             Successfully
 pulled image "dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x" in 1m10.641397525s
(1m10.641428591s including waiting)
50m         Normal   Created                 Pod/asac-5d8c4d547f-6k479             Created
container asac
50m         Normal   Started                 Pod/asac-5d8c4d547f-6k479             Started
container asac
```

Access ASA Container Deployment Logs in Kubernetes Environment

Check the pod logs and container logs for troubleshooting any issues that may occur.

To display pod logs:

```
ubuntu@k8s-master:~$ kubectl describe pod asac-5d8c4d547f-6k479
```

To display container logs:

```
ubuntu@k8s-master:~$ kubectl logs asac-5d8c4d547f-6k479
```

Access the ASA Container Pod in Kubernetes Environment

Run the **kubectl attach** command to access the CLI of the ASA container (ASAc) pod and obtain required outputs. In this example, we access the CLI of the ASAc pod and run the **show version** command.

```
ubuntu@k8s-master:~$ kubectl attach -it asac-5d8c4d547f-6k479
If you don't see a command prompt, try pressing enter.
ciscoasa> show version
Cisco Adaptive Security Appliance Software Version 9.22
SSP Operating System Version 82.16(0.179i)
Device Manager Version 7.20
Compiled on Thu 02-Nov-23 13:30 GMT by builders
System image file is "Unknown, monitor mode tftp booted image"
Config file at boot was "startup-config"
ciscoasa up 55 mins 53 secs
Start-up time 12 secs
Hardware: ASAc, 2048 MB RAM, CPU Xeon E5 series 2100 MHz, 1 CPU (1 core)
BIOS Flash Firmware Hub @ 0x0, 0KB
0: Ext: Management0/0 : address is ae15.c291.86b1, irq 0
1: Ext: GigabitEthernet0/0 : address is faff.65b8.73a9, irq 0
2: Ext: GigabitEthernet0/1 : address is be89.078a.a560, irq 0
3: Int: Internal-Data0/0   : address is 0000.0100.0001, irq 0
```