Deploy network policy in k8s

This page is out of date.
Please see the relevant page for the latest release: e.g. Release I - Run in Kubernetes

network policy is used to control the traffic flow between endpoints in k8s cluster.

This is a brief instruction on how to apply network policy to NONRTRIC. This page deploy NONRTRIC and apply a simple network policy to it.

Prerequisite

- *Docker*
- *minikube v1.22.0*
- *cilium v1.10.2*

Installation

configure minukube resources, the NONRTRIC deployment may cost more resource than default settings in your env:

configure minikube

```
minikube config set cpus 4
minikube config set memory 16384
```

start minukube and enable network plugin:

start minikube and enable network plugin

```
minikube start --network-plugin=cni
```

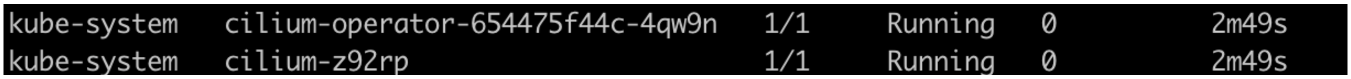
install network plugin, eg: cilium

download cilium and run command, it will detect the minikube cluster automatically and install the network plugin in the cluster:

install cilium

```
cilium install
```

After successful installation, we should see the pods:

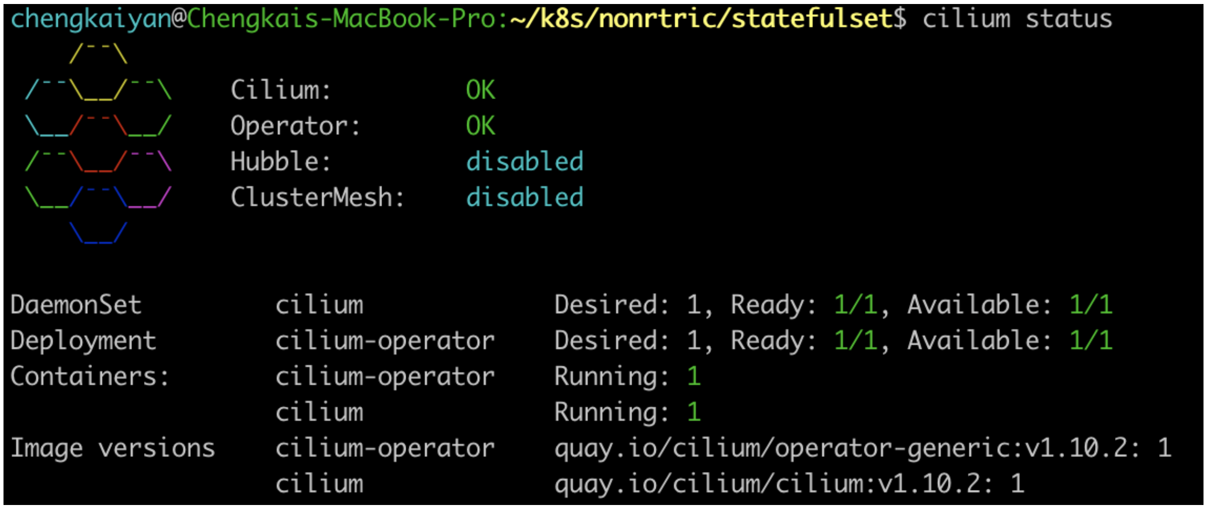


```
kube-system   cilium-operator-654475f44c-4qw9n   1/1     Running   0          2m49s
kube-system   cilium-z92rp                       1/1     Running   0          2m49s
```

Check cilium status:

cilium status

```
cilium status
```



```
chengkaiyan@Chengkais-MacBook-Pro:~/k8s/nonrtric/statefulset$ cilium status
    /¯¯\
 /¯¯\__/¯¯\    Cilium:         OK
 \__/¯¯\__/    Operator:       OK
 /¯¯\__/¯¯\    Hubble:         disabled
 \__/¯¯\__/    ClusterMesh:    disabled
    \__/

DaemonSet         cilium             Desired: 1, Ready: 1/1, Available: 1/1
Deployment        cilium-operator    Desired: 1, Ready: 1/1, Available: 1/1
Containers:       cilium-operator    Running: 1
                  cilium             Running: 1
Image versions    cilium-operator    quay.io/cilium/operator-generic:v1.10.2: 1
                  cilium             quay.io/cilium/cilium:v1.10.2: 1
```

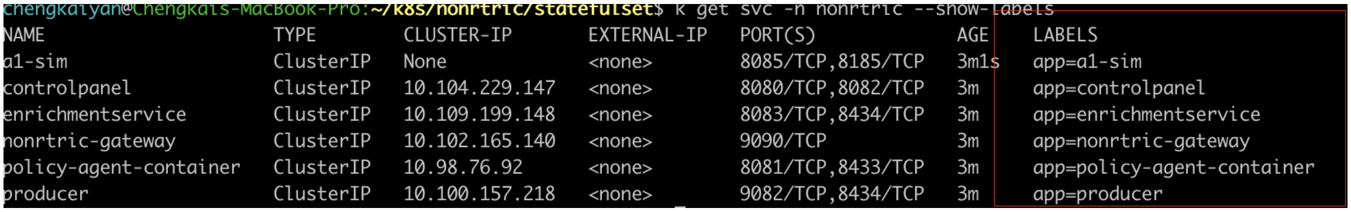
Now we have network plugin enabled in k8s cluster, we can then apply network policy to NONRTRIC.

Deploy NONRTRIC

deploy nonrtric

```
kubectl apply -f https://raw.githubusercontent.com/yanhuanwang/k8s/master/statefulset/nosdnc.yml
```

after deployment, you should be able to see nonrtric services/pods are up and running:



```
chengkaiyan@Chengkais-MacBook-Pro:~/k8s/nonrtric/statefulset$ k get svc -n nonrtric --show-labels
NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE    LABELS
a1-sim                   ClusterIP   None             <none>        8085/TCP,8185/TCP   3m1s   app=a1-sim
controlpanel             ClusterIP   10.104.229.147   <none>        8080/TCP,8082/TCP   3m     app=controlpanel
enrichmentservice        ClusterIP   10.109.199.148   <none>        8083/TCP,8434/TCP   3m     app=enrichmentservice
nonrtric-gateway         ClusterIP   10.102.165.140   <none>        9090/TCP            3m     app=nonrtric-gateway
policy-agent-container   ClusterIP   10.98.76.92      <none>        8081/TCP,8433/TCP   3m     app=policy-agent-container
producer                 ClusterIP   10.100.157.218   <none>        9082/TCP,8434/TCP   3m     app=producer
```

pay attention to the labels above, in this demo we will use labels to define the network-policy rules.

Apply network-policy

```
$ cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: nonrtric
spec:
  podSelector:
    matchLabels:
      app: a1-sim
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: policy-agent-container
  - from:
    - podSelector:
        matchLabels:
          app: a1-sim
EOF
```

The example policy above applies a rule on endpoint with lable "a1-sim", it only allows traffic coming from pod with labels "policy-agent-container" and "a1-sim".

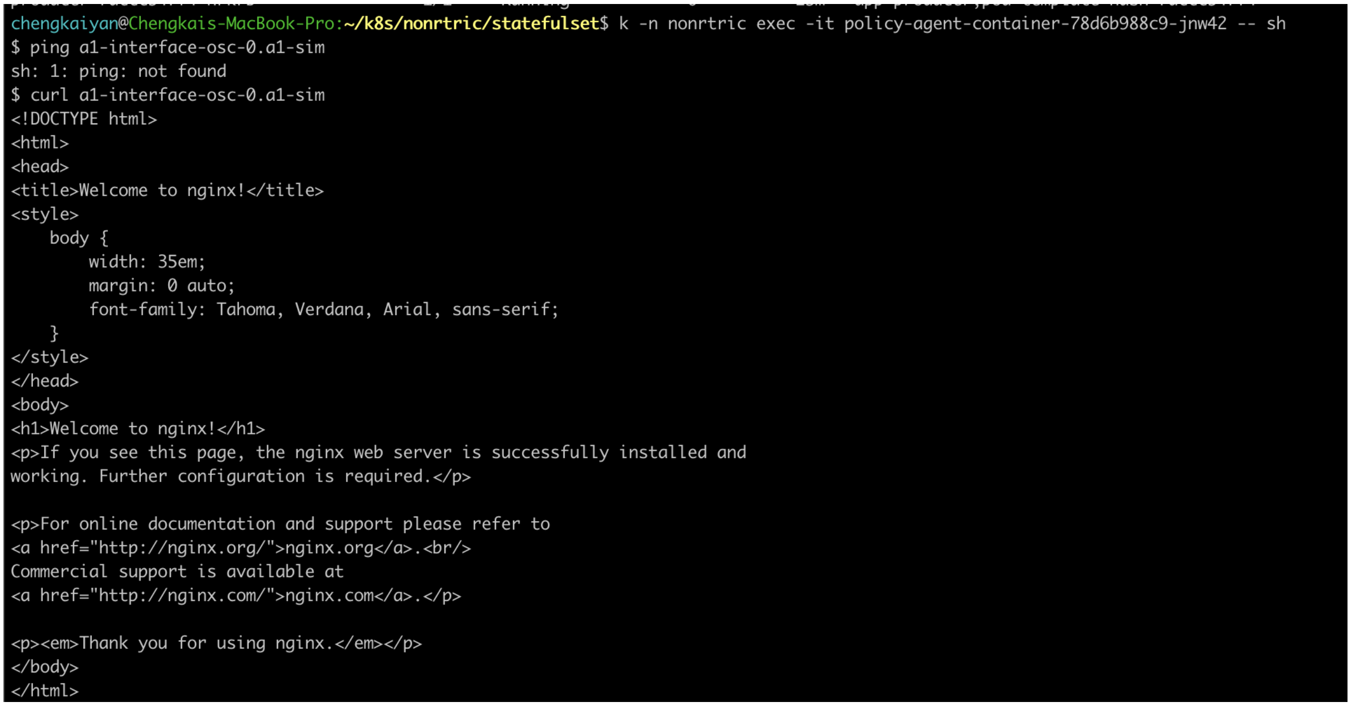
Feel free to change the labels and apply it.

After successfully applying above policy, we login to pod "policy-agent-container-xxxxxx", command:

```
kubectl -n nonrtric exec -it policy-agent-container-78d6b988c9-jnw42 -- sh

curl a1-interface-osc-0.a1-sim
```

We should be able to see:



```
chengkaiyan@Chengkais-MacBook-Pro:~/k8s/nonrtric/statefulset$ k -n nonrtric exec -it policy-agent-container-78d6b988c9-jnw42 -- sh
$ ping a1-interface-osc-0.a1-sim
sh: 1: ping: not found
$ curl a1-interface-osc-0.a1-sim
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

If we update labels in the above policy, for exemple:

```
$ cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: nonrtric
spec:
  podSelector:
    matchLabels:
      app: a1-sim
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: policy-xxxx-container
  - from:
    - podSelector:
        matchLabels:
          app: a1-sim
EOF
```

After applying this changed policy, we cannot access "a1-sim" endpoints from "policy-agent-container" anymore because the labels do not match.

curl/ping command can no longer reach "a1-sim".