Network Policy

In Kubernetes pods can reach any pods by default. Then how pods can be secured? The answer is network-policy. Network policy is a Kubernetes resource like pod, service, ingress and etc. It defines who are all can reach the pod(ingress) and whom the pod can reach(egress).

Prerequisites:

Network polices are implemented by the network plugin, so you must be using a network solution which supports Network Policy. Simply creating the resource without a controller to implement it will have no effect.

Network policy logically can be divided two sections. The first section will identify the pod(s) where the Network policy would be applied. The second section will define the ingress and egress rules for the selected pod(s).

Pod(s) selection:

How the pods would be selected? Yes. You are right. Pod(s) are identified using labels.

**podSelector:**

**matchLabels:**

**role: db**

In the above example the network policy would be applied to the pods which has the label “role: db”.

Ingress and egress rules for group of pod(s):

The second section defines the policy types for the selected pod(s). Policy type can be ingress or egress or both. Ingress is the default policy type. policy identifies the network endpoint where the selected pod(s) can communicate. Network endpoint can be ip address block or pod(s) (all pods or group of pods) in a namespace or selected pods in the same namespace. Ingress network-endpoint has to be defined in the “from” section. Egress network-endpoint has to be defined in the “to” section.

**policyTypes:**

**- Ingress**

**- Egress**

**ingress:**

**- from:**

**- ipBlock:**

**cidr: 172.17.0.0/16**

**except:**

**- 172.17.1.0/24**

**- namespaceSelector:**

**matchLabels:**

**project: myproject**

**- podSelector:**

**matchLabels:**

**role: frontend**

**egress:**

**- to:**

**- ipBlock:**

**cidr: 10.0.0.0/24**

In the above example,

1. The ingress network points are
   1. 172.17.0.0/16 and port except 172.17.1.0/24
   2. All the pods in namespaces which has the label “project: myproject”.
   3. Pods which has the label “role: frontend”
2. The egress network points are
   1. 10.0.0.0/24

Is there any way to select few pods from namespaces instead of all pods in the namespaces? Yes. It can be specified in the namespaceSelector. “namespaceSelector” can have podSelector.

When namespaceSelector has podSelector, network endpoint would be pods with matching labels in the selected namespaces.

The below example shows that allowing connections from pods with label role=client in namespaces with the label user=alice. Please be aware to use correct yaml syntax.

**...**

**ingress:**

**- from:**

**- namespaceSelector:**

**matchLabels:**

**user: alice**

**podSelector:**

**matchLabels:**

**role: client**

**...**

So far it is fine. Still there is a security concern. Is there any way to specify ports for ingress and egress? Yes. As part of the policy it can be mentioned. If it is not mentioned it applies to all ports. Ports in ingress says that selected pod(s) can allow traffic for the specified ports. Ports in egress says that selected pod(s) can send traffic to specified ports.

Previous example along with port specifications

**policyTypes:**

**- Ingress**

**- Egress**

**ingress:**

**- from:**

**- ipBlock:**

**cidr: 172.17.0.0/16**

**except:**

**- 172.17.1.0/24**

**- namespaceSelector:**

**matchLabels:**

**project: myproject**

**- podSelector:**

**matchLabels:**

**role: frontend**

**ports:**

**- protocol: TCP**

**port: 6379**

**egress:**

**- to:**

**- ipBlock:**

**cidr: 10.0.0.0/24**

**ports:**

**- protocol: TCP**

**port: 5978**

The above network policy says that all ingress network endpoint can reach selected pod(s) tcp port 6379 and selected pod(s) can reach all egress network endpoint’s tcp port 5978.

The rest of the traffic would be blocked.

Sample network-policy

**apiVersion: networking.k8s.io/v1**

**kind: NetworkPolicy**

**metadata:**

**name: mydb**

**spec:**

**podSelector:**

**matchLabels:**

**role: db**

**policyTypes:**

**- Ingress**

**- Egress**

**ingress:**

**- from:**

**- ipBlock:**

**cidr: 172.17.0.0/16**

**except:**

**- 172.17.1.0/24**

**- namespaceSelector:**

**matchLabels:**

**project: myproject**

**- podSelector:**

**matchLabels:**

**role: frontend**

**ports:**

**- protocol: TCP**

**port: 6379**

**egress:**

**- to:**

**- ipBlock:**

**cidr: 10.0.0.0/24**

**ports:**

**- protocol: TCP**

**port: 5978**

kubectl create -f mydb-netpol.yaml

kubectl get netpol

NAME POD-SELECTOR AGE

mydb role=db 3m5s

kubectl describe netpol mydb

Name: mydb

Namespace: default

Created on: 2019-06-30 07:41:18 -0700 PDT

Labels: <none>

Annotations: <none>

Spec:

PodSelector: role=db

Allowing ingress traffic:

To Port: 6379/TCP

From:

IPBlock:

CIDR: 172.17.0.0/16

Except: 172.17.1.0/24

From:

NamespaceSelector: project=myproject

From:

PodSelector: role=frontend

Allowing egress traffic:

To Port: 5978/TCP

To:

IPBlock:

CIDR: 10.0.0.0/24

Except:

Policy Types: Ingress, Egress