SECURITY

Security

Network Security Policy Example

The use of policies has become stable, noted with the **`v1 apiVersion`**. The example below narrows down the policy to affect the default namespace.

Only Pods with the label of **`role: db`** will be affected by this policy, and the policy has both Ingress and Egress settings.

The **`ingress`** setting includes a **`172.17`** network, with a smaller range of **`172.17.1.0`** IPs being excluded from this traffic.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ingress-egress-policy
  namespace: default
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
```

These rules change the namespace for the following settings to be labeled **`project: myproject`**.

The affected Pods also would need to match the label **`role: frontend`**. Finally, TCP traffic on port 6379 would be allowed from these Pods.

The egress rules have the **`to`** settings, in this case the **`10.0.0.0/24`** range TCP traffic to port 5978.

The use of empty ingress or egress rules denies all type of traffic for the included Pods, though this is not suggested. Use another dedicated **`NetworkPolicy`** instead.

Note that there can also be complex **`matchExpressions`** statements in the spec, but this may change as **`NetworkPolicy`** matures.

```
podSelector:
  matchExpressions:
    - {key: inns, operator: In, values: ["yes"]}
```