How can we fine-tune Network Policy using selectors?

Kubernetes - Beginners | Intermediate | Advanced

How can we fine-tune Network Policy using selectors?

There are endless scenarios where you want to allow or deny traffic from specific or multiple sources. The same thing holds for which destination you want to allow traffic to. Kubernetes NetworkPolicy resource provides you with a rich set of selectors that you can use to secure your network paths the way you want. :

There are four kinds of selectors that can be specified in an ingress from section or egress to section:

- podSelector: This selects particular Pods in the same namespace as the NetworkPolicy which should be allowed as ingress sources or egress destinations.
- namespaceSelector: This selects particular namespaces for which all Pods should be allowed as ingress sources or egress destinations.

- namespaceSelector and podSelector: A single to/from entry that specifies both namespaceSelector and podSelector selects particular Pods within particular namespaces. Be careful to use correct YAML syntax; this policy:

```
...
ingress:
- from:
  - namespaceSelector:
      matchLabels:
        user: alice
    podSelector:
      matchLabels:
        role: client
...
```

contains a single from element allowing connections from Pods with the label role=client in namespaces with the label user=alice. But this policy:

```
...
ingress:
- from:
  - namespaceSelector:
      matchLabels:
        user: alice
  - podSelector:
      matchLabels:
        role: client
...
```

ipBock: here you can define which IP CIDR blocks are the source or the destination of the selected pods' connections. Traditionally, those IPs are external to the cluster because pods always have short-lived IPs that can change at any moment. You should be aware that, depending on the network plugin in use, the source IP address may change before the packet gets analyzed by the NetworkPolicy rules. An example scenario is when the cloud provider's Load Balancer replaces the source IP of the packet with its own. ipBlock can also be used to block specific IPs from an allowed range. This can be done using the except keyword. For example, we can allow all traffic from 182.213.0.0/16 but deny 182.213.50.43. The snippet for such a configuration may look as follows:

```
- ipBlock:
      cidr: 182.213.0.0/16
      except:
      - 182.213.50.43/24
```

Join KubeDaily

kubelabs is maintained by collabnix.

This page was generated by GitHub Pages.