

Lab 16: Network Policies

Training Goals Covered:

- Observe the "Allow All" default behavior.
- Create a policy to Deny All traffic to a specific app.
- Create an Ingress rule to allow traffic only from a specific "Trusted" Pod.

Steps:

1. Create a deployment as **web-server**, using the label/selector "app=web-server" and the image as **nginx:alpine**.

Shell

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-server
spec:
  selector:
    matchLabels:
      app: web-server
  template:
    metadata:
      labels:
        app: web-server
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
          ports:
            - containerPort: 80
```

2. Expose the deployment as **ClusterIP** service (**web-service**) on **port 80**.

Shell

```
apiVersion: v1
```

```
kind: Service
metadata:
  name: web-service
spec:
  type: ClusterIP
  selector:
    app: web-server
  ports:
    - port: 80
      targetPort: 80
```

3. Deploy 2 pods, the *trusted client* and the *untrusted client*, as defined below.

```
Shell
# The Trusted Client
apiVersion: v1
kind: Pod
metadata:
  name: trusted-frontend
  labels:
    role: frontend
spec:
  containers:
    - name: busybox
      image: busybox
      command: ["sh", "-c", "sleep 3600"]
---
# The Untrusted Client
apiVersion: v1
kind: Pod
metadata:
  name: untrusted-pod
  labels:
    role: unknown
spec:
  containers:
    - name: busybox
      image: busybox
      command: ["sh", "-c", "sleep 3600"]
```

4. Attempt to access the web server from each of the pods defined previously. Is the server reachable?

Shell

```
k exec -it trusted-frontend -- wget -qO- --timeout=2
web-service.default.svc.cluster.local
```

Shell

```
k exec -it untrusted-pod -- wget -qO- --timeout=2
web-service.default.svc.cluster.local
```

5. Deploy a deny all policy for **ingress** access to web-server.

Shell

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-web-server
spec:
  podSelector:
    matchLabels:
      app: web-server
  policyTypes:
    - Ingress
    # Leaving ingress empty means "Deny All"
```

Shell

```
kubectl get netpol
```

6. Test the access again by repeating step 4. Based on this test, which entity or entities are able to reach the web server?
→ All traffic blocked.

7. Add a new network policy to allow access only from trusted frontend.

Shell

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-only
spec:
  podSelector:
    matchLabels:
      app: web-server
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          role: frontend
  ports:
  - protocol: TCP
    port: 80
```

8. Validate that only the frontend pod has access.

→ Execute the same commands of step 4 and 6.