Certified Kubernetes Application Developer: Services and Networking

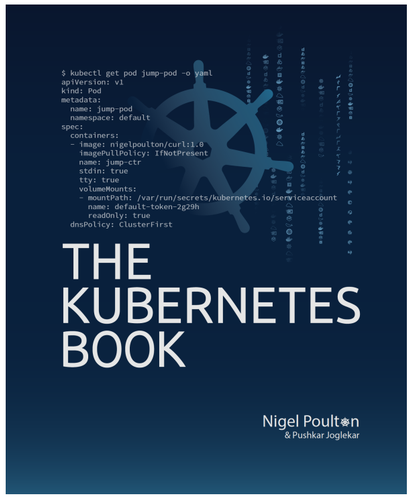
Demonstrate Basic Understanding of NetworkPolicies

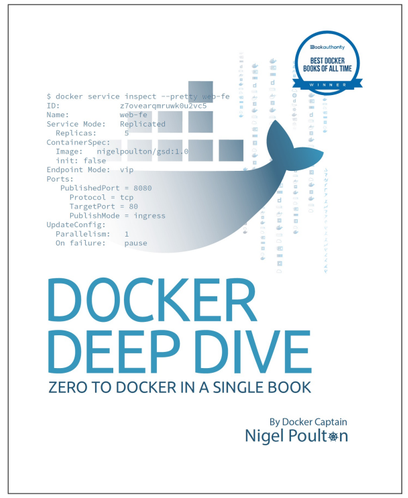
Nigel Poulton
Author & Trainer

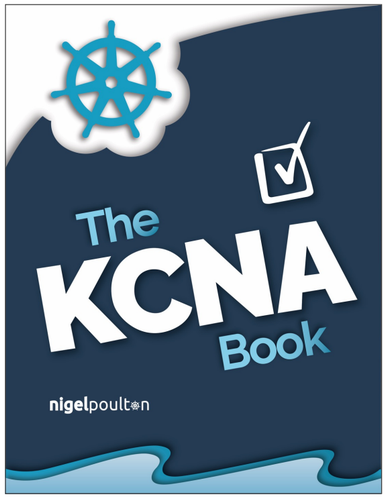
@nigelpoulton nigelpoulton.com

@nigelpoulton

@nigelpoulton







- Docker and Kubernetes: The Big Picture
- Getting Started with Docker
- Getting Started with Kubernetes
- Docker Deep Dive...

Dan Wahlin

Wahlin Consulting

@danwahlin

linkdein.com/in/danwahlin

codewithdan.com

Nigel Poulton

Author & Trainer

@nigelpoulton

linkdein.com/in/nigelpoulton

nigelpoulton.com

Certified Kubernetes Application Developer

Application Design and Build

Application Deployment

Application Observability and Maintenance

Application Environment, Configuration and Security

Services and Networking

About this Course

Demonstrate Basic Understanding of NetworkPolicies

Provide and Troubleshoot Access to Applications via Services

Use Ingress Rules to Expose Applications

Module Agenda

Understanding NetworkPolicies

Working with Network Policies

Exam Scenarios

Recap and Test Yourself

Understanding NetworkPolicies

“By default, Kubernetes networks are like the lawless wild west.”

Nigel Poulton

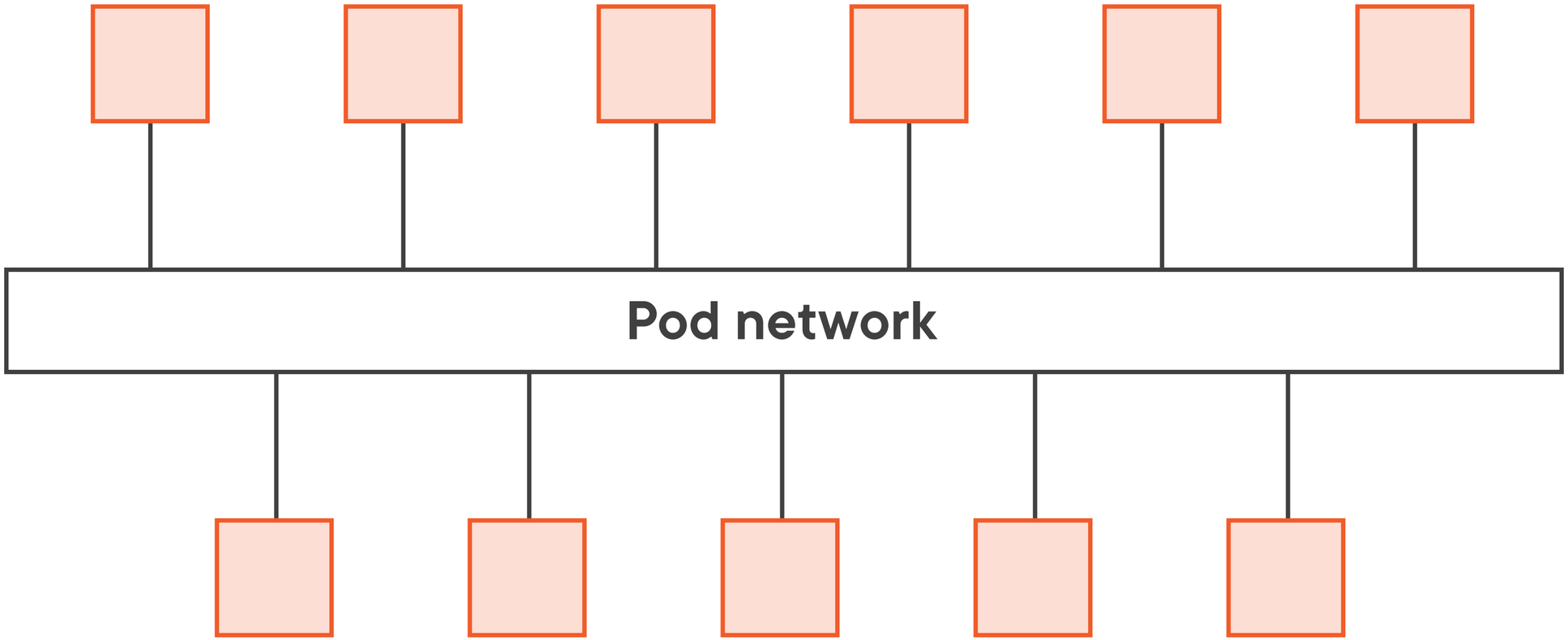
“By default, Kubernetes networks are not fit for production!”

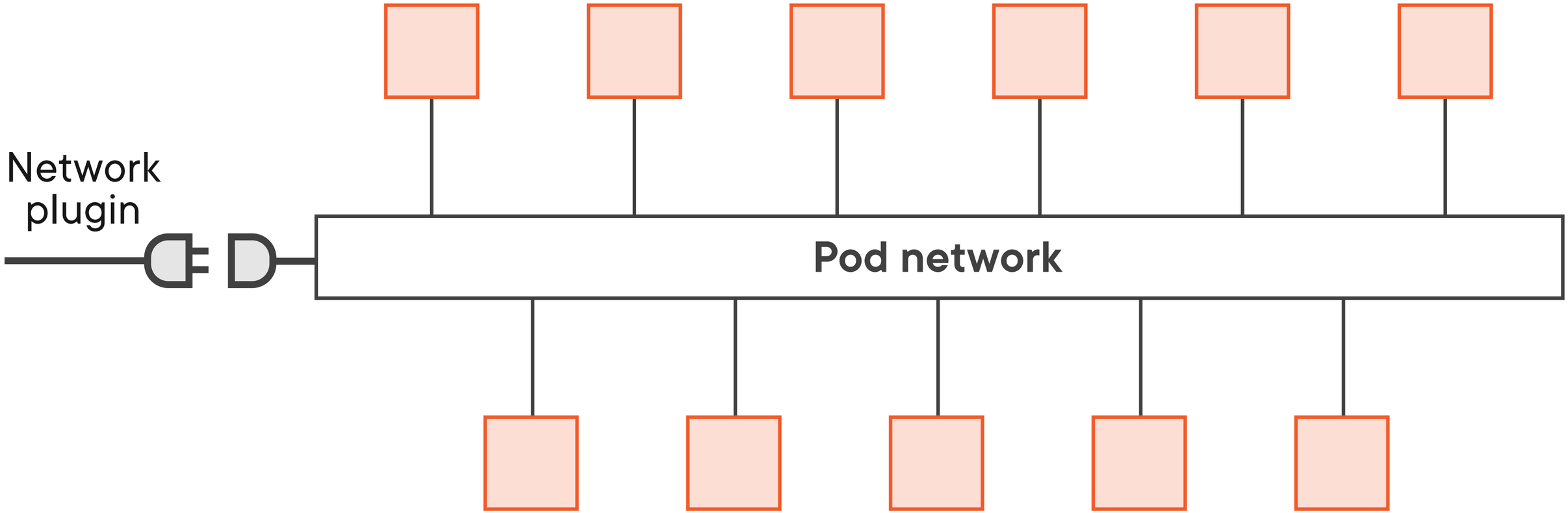
Nigel Poulton

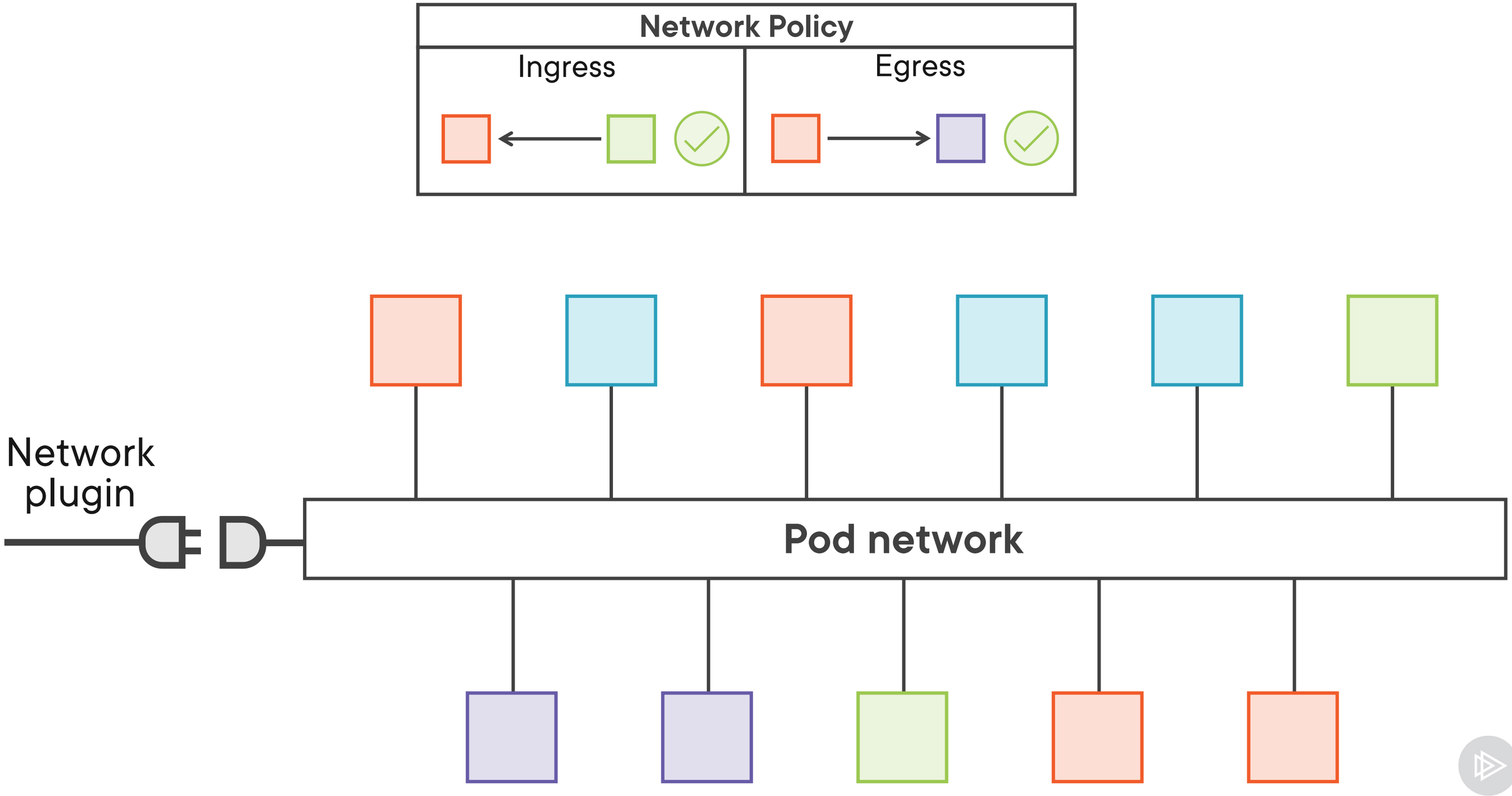
“By default, Kubernetes networks are not fit for production! NetworkPolicies come to the rescue...”

Nigel Poulton

It’s the **plugin** that implements network policies


Pod network


Network
plugin
Pod network


Network Policy
Ingress
Egress
Network plugin
Pod network

Ingress

Traffic coming in to a Pod

Ingress

Traffic coming in to a Pod

Egress

Traffic going out from a Pod

Traffic not in a policy is implicitly denied

Policies aggregate

All rules are “allow” rules

Single FROM rule

```
- from:
  - podSelector:
      matchLabels:
        project: ckad
    namespaceSelector:
      matchLabels:
        kubernetes.io/metadata.name: ps
```

(AND)

Two FROM rules

```
- from:
  - podSelector:
      matchLabels:
        project: ckad
  - namespaceSelector:
      matchLabels:
        kubernetes.io/metadata.name: ps
```

(OR)

Working with NetworkPolicies

```
ingress:
   - from:
       - podSelector:
           matchLabels:
             project: ckad
       - namespaceSelector:
           matchLabels:
             kubernetes.io/metadata.name: pluralsight
```

Exam Scenarios

Task weight: 4%

```
Cluster: k8s001
Namespace: default
Doc links: NetworkPolicies, Pods, Services
```

You're in the process of deploying an application.

The database elements are up and running, but the **store-backend** Pod is failing to start.

The following files are in your working directory and are all required to start the application.

`db.yml` `backend.yml` `netpol.yml` `deny.yml`

Task

Troubleshoot and fix the issue. When completed, the store-backend Pod should enter the **Running** phase

Task weight: 8%

```
Cluster: k8s0012
Namespace: ckad
Doc links: NetworkPolicies, Pods, Services
```

You have a multi-tier application deployed to the **ckad** Namespace.

Database Pods (tagged with the **app=db** label) host sensitive data and should only be accessible on **port 5432** by Pods in the same Namespace with the **app=backend** label. The **netdb** NetworkPolicy is designed to enforce this rule.

However, all Pods in the **ckad** Namespace are able to connect to the database Pods on port 5432.

Task

Troubleshoot the issue and ensure only Pods in the **ckad** Namespace with the **app=backend** label can access the database Pods.

When you have resolved the issue, the following test command should work in pod-1 but should timeout in pod-2 ---- **`psql -h db`**

Pod-1

```
kubectl run pod-1 --rm -it --image=postgres:alpine -l app=backend -- /bin/sh
```

Pod-2

```
kubectl run pod-2 --rm -it --image=postgres:alpine -l -- /bin/sh
```

Recap and Test Yourself

Kubernetes networks are wide open.
NetworkPolicies help us secure them.

NetworkPolicies

are Namespaced

NetworkPolicies

require plugin support

NetworkPolicies

have ingress and egress rules

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-policy
  namespace: ckad
spec:
  podSelector:
    matchLabels:
      app: db
  ingress:
    - from:
      - podSelector:
          matchLabels:
            app: front-end
        namespaceSelector:
          matchLabels:
            kubernetes.io…: ckad
```

Apply to these Pods

Ingress rule

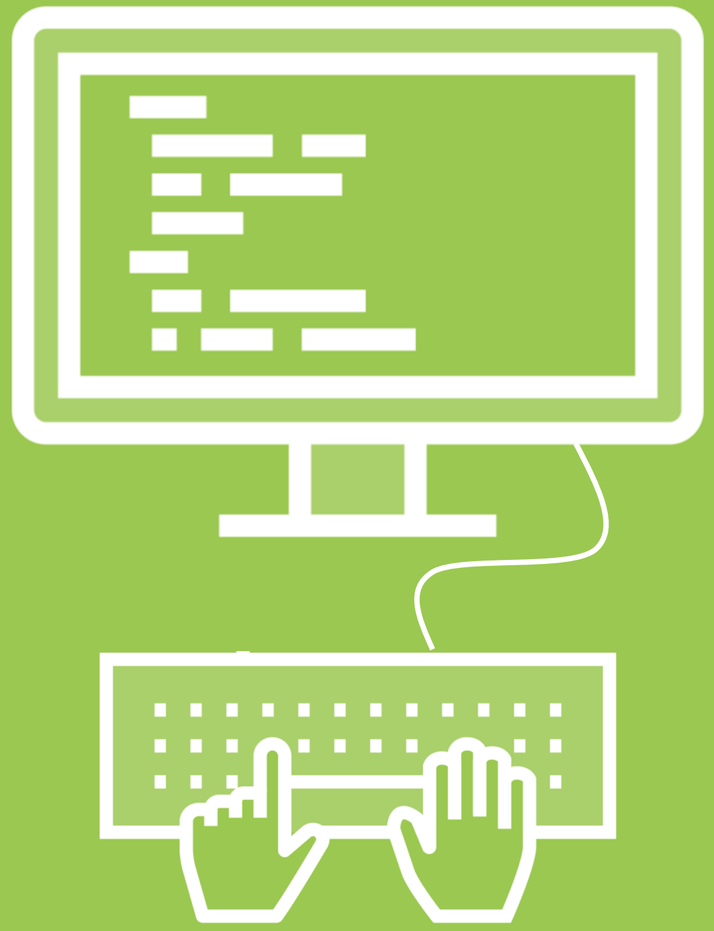
From Pods with app=front-end label

That are also in the ckad Namespace

```
namespaceSelector:
    matchLabels:
        kubernetes.io/metadata.name: ckad
```

Policies aggregate

Traffic not in a policy is implicitly denied



GitHub Repo

https://github.com/nigelpoulton/ckad

Navigate to:

- 5 Services and Networking
- 2 Demonstrate Basic Understanding of NetworkPolicies

Up Next:

Provide and Troubleshoot Access to Applications via Services