A CLOUD GURU

K8s Networking Architectural Overview

Will Boyd

DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

K8s Networking Architectural Overview

Kubernetes Network Model

Network Model Architecture

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

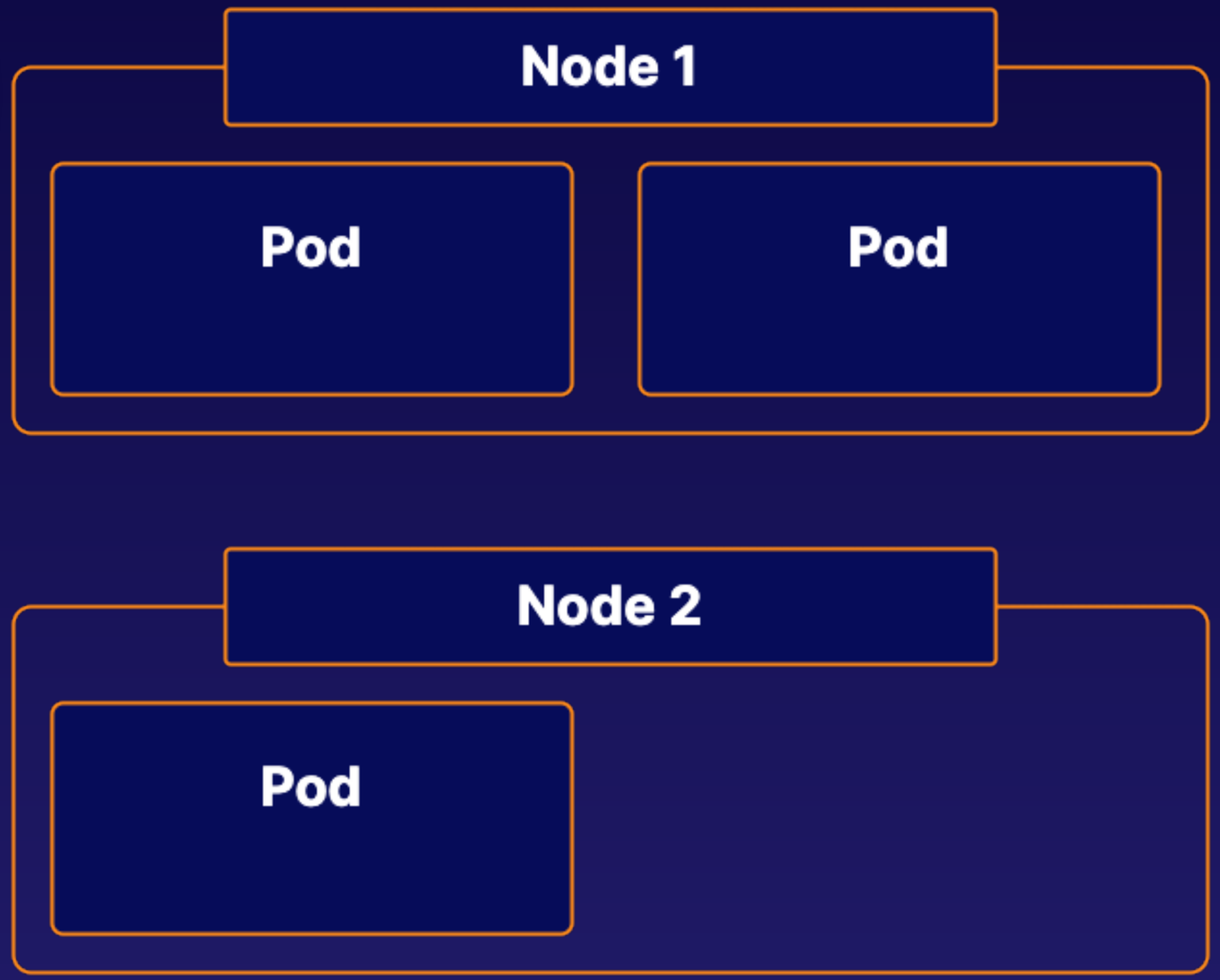
Kubernetes Network Model

The **Kubernetes network model** is a set of standards that define how networking between Pods behaves.

There are a variety of different implementations of this model — including the **Calico network plugin**, which we have been using throughout this course.

Network Model Architecture

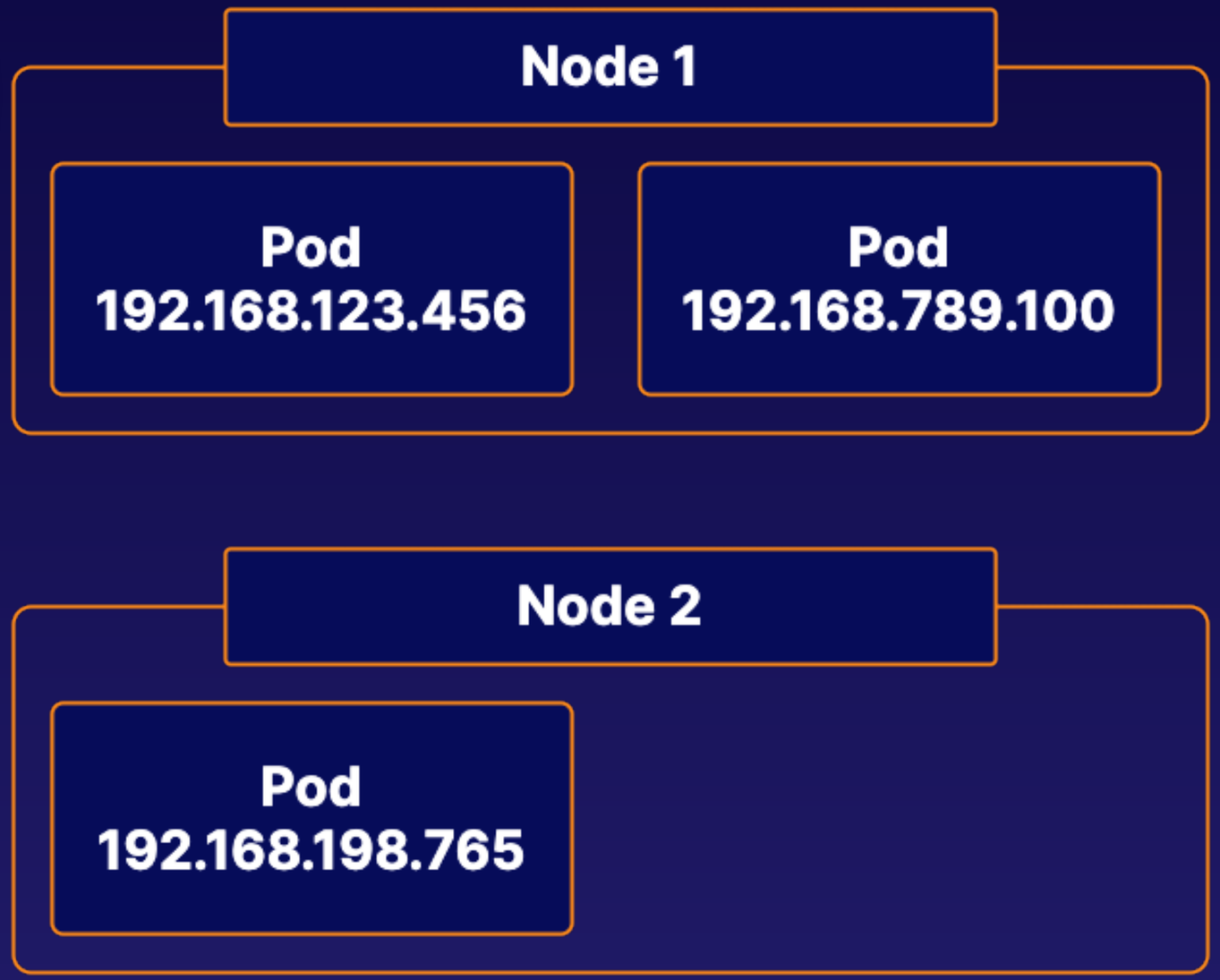
The Kubernetes network model defines how Pods communicate with each other, regardless of which Node they are running on.



Network Model Architecture

The Kubernetes network model defines how Pods communicate with each other, regardless of which Node they are running on.

Each Pod has its own **unique IP address** within the cluster.

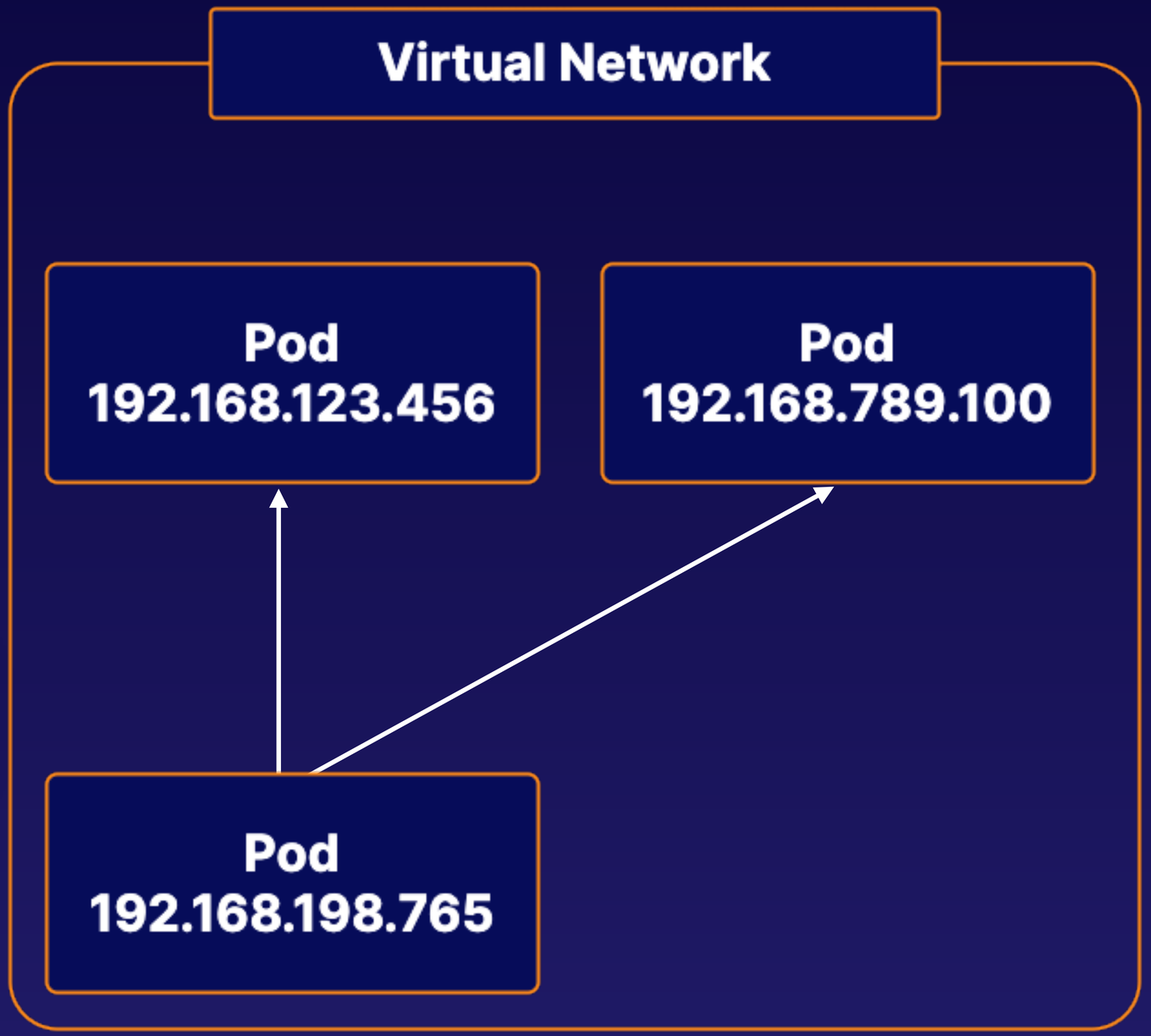


Network Model Architecture

The Kubernetes network model defines how Pods communicate with each other, regardless of which Node they are running on.

Each Pod has its own **unique IP address** within the cluster.

Any Pod can reach any other Pod using that Pod's IP address. This creates a **virtual network** that allows Pods to easily communicate with each other, regardless of which node they are on.



LESSON BREAKDOWN

K8s Networking Architectural Overview

Kubernetes Network Model

Network Model Architecture

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

CNI Plugins Overview

Will Boyd

DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

CNI Plugins Overview

CNI Plugins

Selecting a Network Plugin

Installing Network Plugins

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

CNI Plugins

CNI plugins are a type of Kubernetes network plugin. These plugins provide network connectivity between Pods according to the standard set by the Kubernetes network model.

There are many CNI network plugins available.

A CLOUD GURU

Selecting a Network Plugin

Which network plugin is best for you will depend on your specific situation.

Check the Kubernetes documentation for a list of available plugins. You may need to research some of these plugins for yourself, depending on your production use case.

The plugin used in this course is Calico.

A CLOUD GURU

Installing Network Plugins

Each plugin has its own unique installation process. Early in this course, we installed the Calico network plugin.

Note: Kubernetes nodes will remain `NotReady` until a network plugin is installed. You will be unable to run Pods while this is the case.

LESSON BREAKDOWN

CNI Plugins Overview

CNI Plugins

Selecting a Network Plugin

Installing Network Plugins

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Understanding K8s DNS

Will Boyd

DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Understanding K8s DNS

DNS in K8s

Pod Domain Names

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

DNS in K8s

The K8s virtual network uses a **DNS** (Domain Name System) to allow Pods to locate other Pods and Services using domain names instead of IP addresses.

This DNS runs as a Service within the cluster. You can usually find it in the **`kube-system`** namespace.

Kubeadm clusters use **CoreDNS**.

A CLOUD GURU

Pod Domain Names

All Pods in our kubeadm cluster are automatically given a domain name of the following form:

```
pod-ip-address.namespace-name.pod.cluster.local
```

A Pod in the **`default`** namespace with the IP address 192.168.10.100 would have a domain name like this:

```
192-168-10-100.default.pod.cluster.local
```

Hands-On Demonstration

LESSON BREAKDOWN

Understanding K8s DNS

DNS in K8s

Pod Domain Names

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Using NetworkPolicies

Will Boyd

DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Using NetworkPolicies

What Is a NetworkPolicy?

Pod Selector

Ingress and Egress

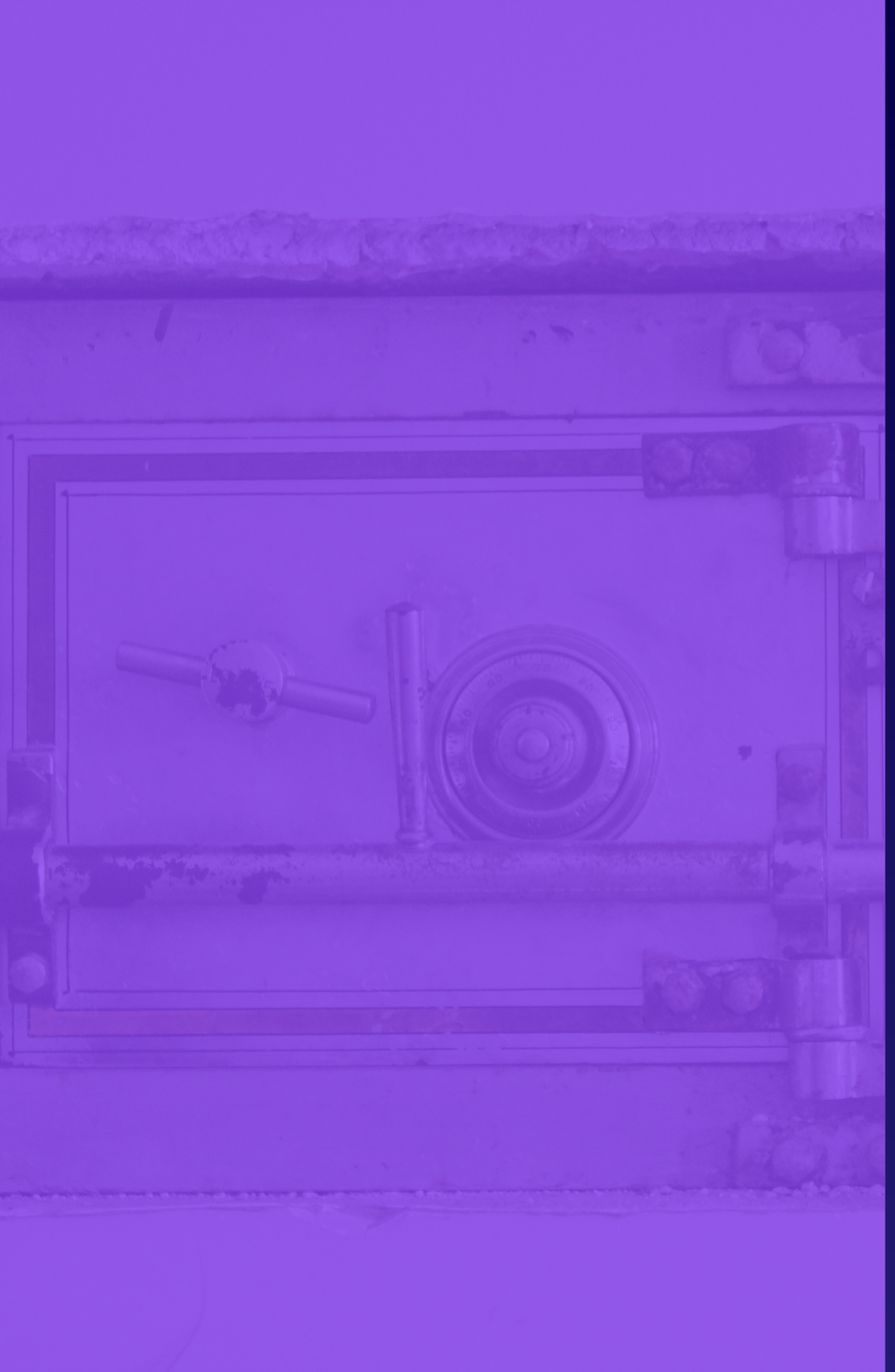
`from` and `to` Selectors

Ports

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU



A CLOUD GURU

What Is a NetworkPolicy?

A K8s **NetworkPolicy** is an object that allows you to control the flow of network communication to and from Pods.

This allows you to build a more secure cluster network by keeping Pods isolated from traffic they do not need.

Pod Selector

podSelector: Determines to which Pods in the namespace the NetworkPolicy applies.

The podSelector can select Pods using Pod labels.

Note: By default, Pods are considered non-isolated and completely open to all communication.

If any NetworkPolicy selects a Pod, the Pod is considered isolated and will only be open to traffic allowed by NetworkPolicies.

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: my-network-policy
spec:
  podSelector:
    matchLabels:
      role: db
```

Ingress and Egress

A NetworkPolicy can apply to Ingress, Egress, or both.

INGRESS

Incoming network traffic coming **into** the Pod from another source.

VS

EGRESS

Outgoing network traffic **leaving** the Pod for another destination.

`from` and `to` Selectors

`from` selector: Selects ingress (incoming) traffic that will be allowed.

`to` selector: Selects egress (outgoing) traffic that will be allowed.

```
spec:
  ingress:
    - from:
      ...
  egress:
    - to:
      ...
```

A CLOUD GURU

`from` and `to` Selectors

podSelector

Select Pods to allow traffic from/to.

```
spec:
  ingress:
    - from:
      - podSelector:
          matchLabels:
            app: db
```

namespaceSelector

Select namespaces to allow traffic from/to.

```
spec:
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            app: db
```

ipBlock

Select an IP range to allow traffic from/to.

```
spec:
  ingress:
    - from:
      - ipBlock:
          cidr: 172.17.0.0/16
```

A CLOUD GURU

Ports

`port`: Specifies one or more ports that will allow traffic.

```
spec:
  ingress:
    - from:
      ports:
        - protocol: TCP
          port: 80
```

Traffic is only allowed if it matches both an allowed port and one of the `from`/`to` rules.

Hands-On Demonstration

LESSON BREAKDOWN

Using NetworkPolicies

What Is a NetworkPolicy?

Pod Selector

Ingress and Egress

`from` and `to` Selectors

Ports

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU