Using Network Security Policies to Restrict Cluster Level Access

1. Introduction (5 Minutes):

•Briefly explain Network Policies (NPs) and their role in controlling pod-to-pod communication.

•Highlight the security benefits of using NPs to restrict access within the cluster.

2. Demonstration (20 Minutes):

a. Prerequisites:

•Ensure your Kubernetes cluster is up and running with a CNI (Container Network Interface) plugin that supports Network Policies (e.g., Calico, Cilium).

•Have kubectl configured with appropriate permissions to create and manage Network Policies.

b. Creating a basic Network Policy (5 minutes):

1.**Create a YAML file:**

apiVersion: networking.k8s.io/v1 kind: NetworkPolicy metadata: name: deny-all-ingress spec: podSelector: {} policyTypes: - Ingress ingress: [] ```

```
* This policy denies all incoming traffic (ingress) to all pods in the cluster.
```

2.**Apply the policy:**

Bash

kubectl apply -f deny-all-ingress.yaml

Use code with caution.

3.**Verify the policy:**

Bash

kubectl get networkpolicies

Use code with caution.

c. Allowing specific traffic (10 Minutes):

1.**Modify the YAML file to allow specific traffic:**

YAML

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-web-traffic
spec:
  podSelector:
    matchLabels: app: webserver # Only applies to pods with label app=webserver
  policyTypes:

```
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels: app: database # Allow traffic only from pods with label
app=database
    - ports:
      - port: 80  # Allow traffic on port 80 (HTTP)
```

Use code with caution.