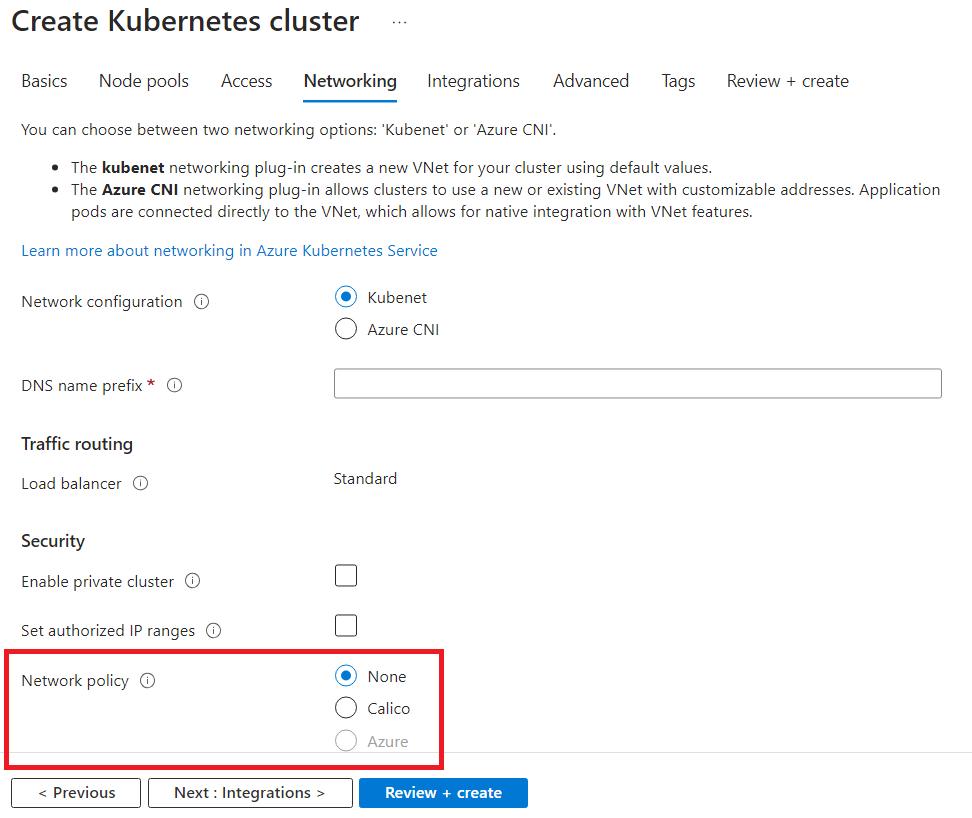
***K8s Network Policy for Restrictions on Pod Connectivity***

Prerequisites: According to the Kubernetes Network Policies [documentation](https://kubernetes.io/docs/concepts/services-networking/network-policies/), Deployed Network Policies will be only effective if a [network plugin](https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/) is installed in the Kubernetes cluster.



A custom label must be included in Deployment YAMLs to identify the resources on which this network policy has to be implemented. For this example, we are using “networkpolicy” custom label. Below is the YAML for the sample pod. To differentiate, we use “needed” and “notneeded” as values for this custom label.

--------------------------------------------------------------------------------------------------------------------------------------

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*name: samplepod1*

*spec:*

*replicas: 1*

*selector:*

*matchLabels:*

*app: samplepod1*

*networkpolicy: needed*

*template:*

*metadata:*

*labels:*

*app: samplepod1*

*networkpolicy: needed*

*spec:*

*containers:*

*- name: nginx*

*image: nginx:latest*

--------------------------------------------------------------------------------------------------------------------------------------

Scenario: Communication happens only between Pod2 and Pod3 while restricting Pod1 from communicating with any other pod.

Below is a sample Kubernetes Network Policy manifest that fulfils this requirement:

--------------------------------------------------------------------------------------------------------------------------------------

*apiVersion: networking.k8s.io/v1*

*kind: NetworkPolicy*

*metadata:*

*name: allow-pod2-pod3-deny-pod1*

*spec:*

*podSelector:*

*matchLabels:*

*networkpolicy: needed*

*policyTypes:*

*- Ingress*

*- Egress*

*ingress:*

*- from:*

*- podSelector:*

*matchLabels:*

*app: samplepod2*

*- podSelector:*

*matchLabels:*

*app: samplepod3*

*egress:*

*- to:*

*- podSelector:*

*matchLabels:*

*app: samplepod2*

*- podSelector:*

*matchLabels:*

*app: samplepod3*

*- to:*

*- ipBlock:*

*cidr: 0.0.0.0/0*

--------------------------------------------------------------------------------------------------------------------------------------

In this Network Policy:

1. `podSelector` selects all the pods with the label `*networkpolicy: needed* `. Make sure to label your pods accordingly.
2. `policyTypes` specifies that both Ingress (incoming) and Egress (outgoing) traffic will be controlled.
3. Under `ingress`, we allow incoming traffic from pods labelled as `app: samplepod2` and `app: samplepod3`, effectively allowing communication between SamplePod2 and SamplePod3.
4. Under `egress`, we allow outgoing traffic to pods labelled as `app: samplepod2` and `app: samplepod3`, ensuring that Pod1 cannot communicate with other pods.
5. The egress section includes a rule that allows traffic to the IP block 0.0.0.0/0, which represents the entire internet.

Apply the manifest using `kubectl apply -f your\_network\_policy.yaml`. With this Network Policy in place, the communication between the pods will be as per our specified requirements.