

```
Text(  
'Section Title',  
style: TextStyle(  
  color: Colors.blue[200],  
),  
),  
),
```

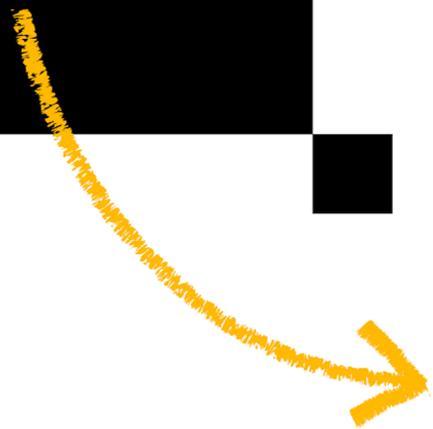
# devfest

```
s.star,  
r: Colors.blue[500],  
Text('23'),
```

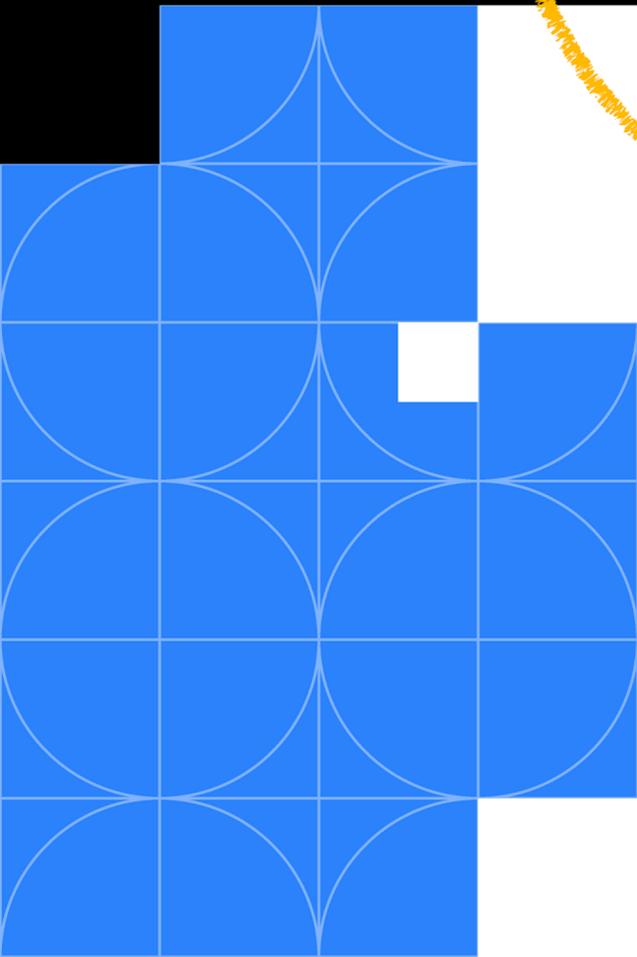


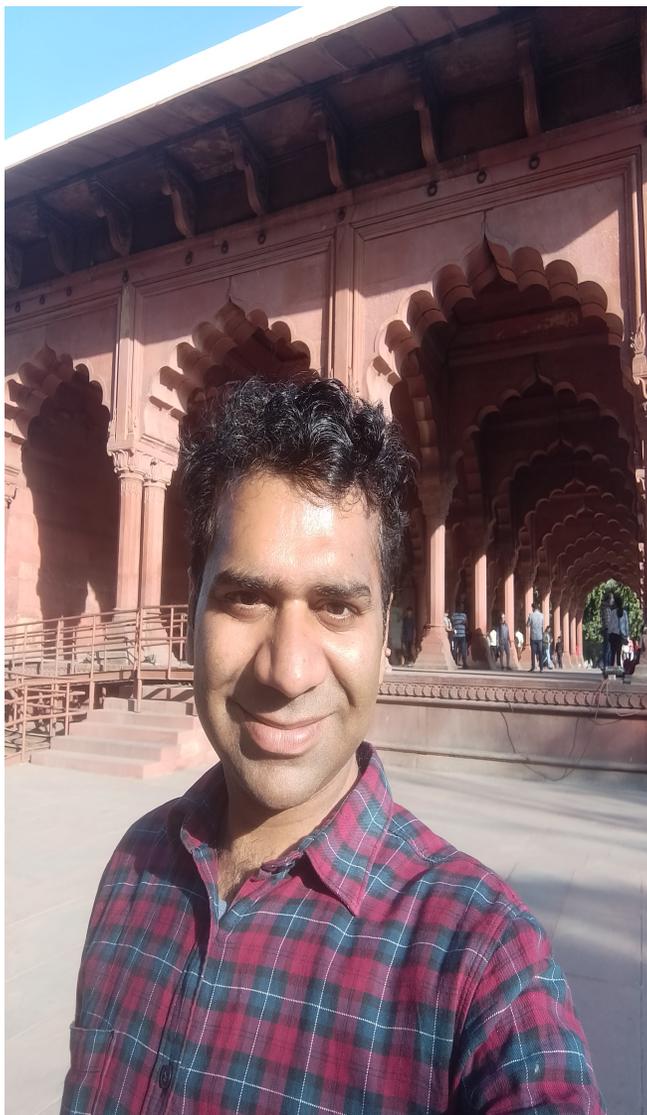
Google Developer Groups

📅 30th Dec, 2023      📍 Noida



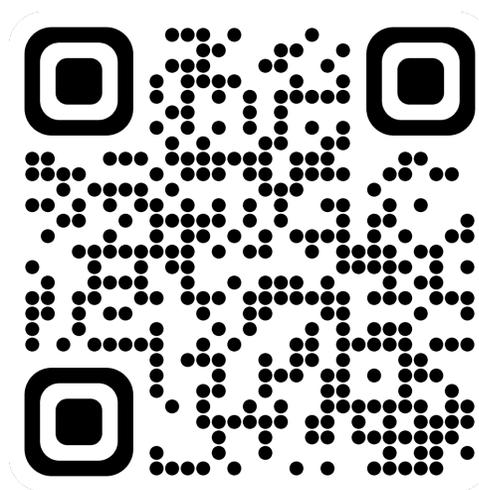
# Kubernetes Networking





## Ankit Gupta

Manager - Cloud Engineering @ KPMG  
5x GCP certified



# Agenda



Background...



Kubernetes network model



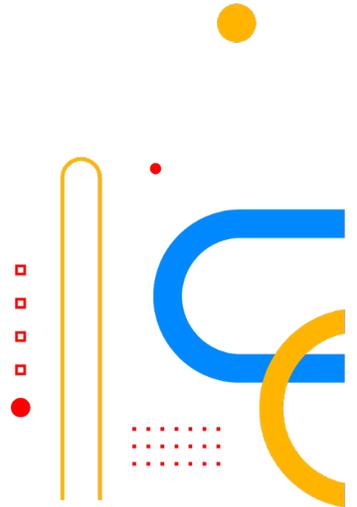
Pod-Node communication



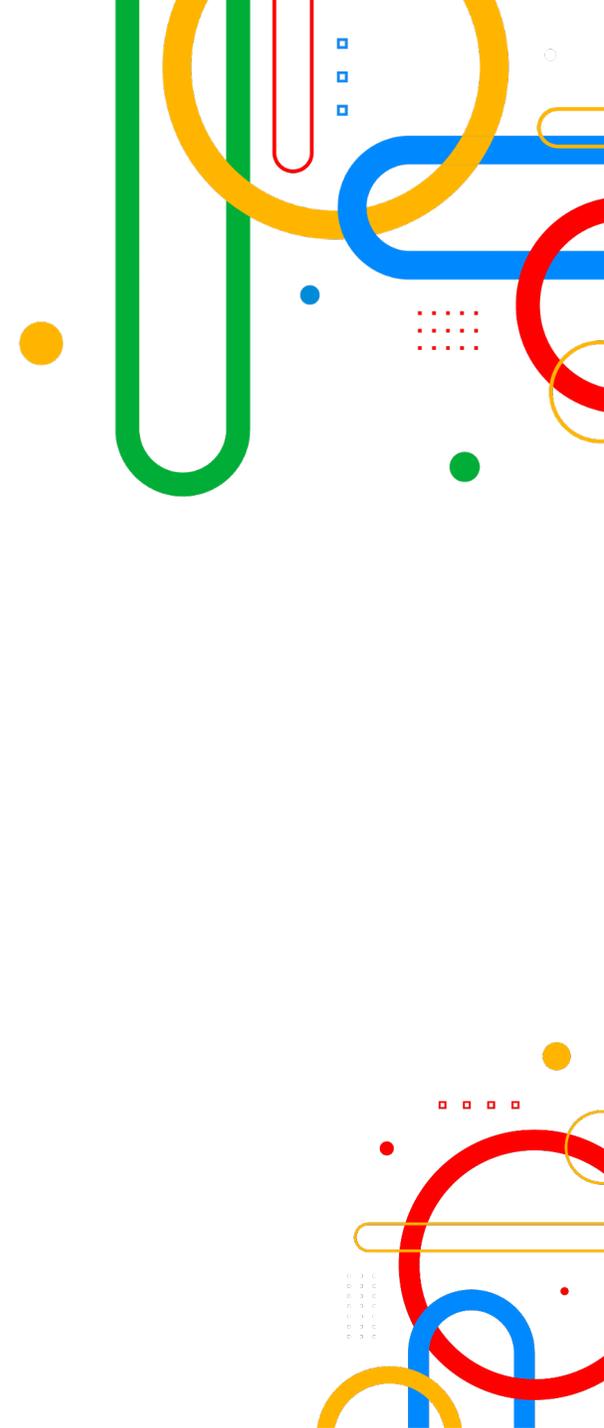
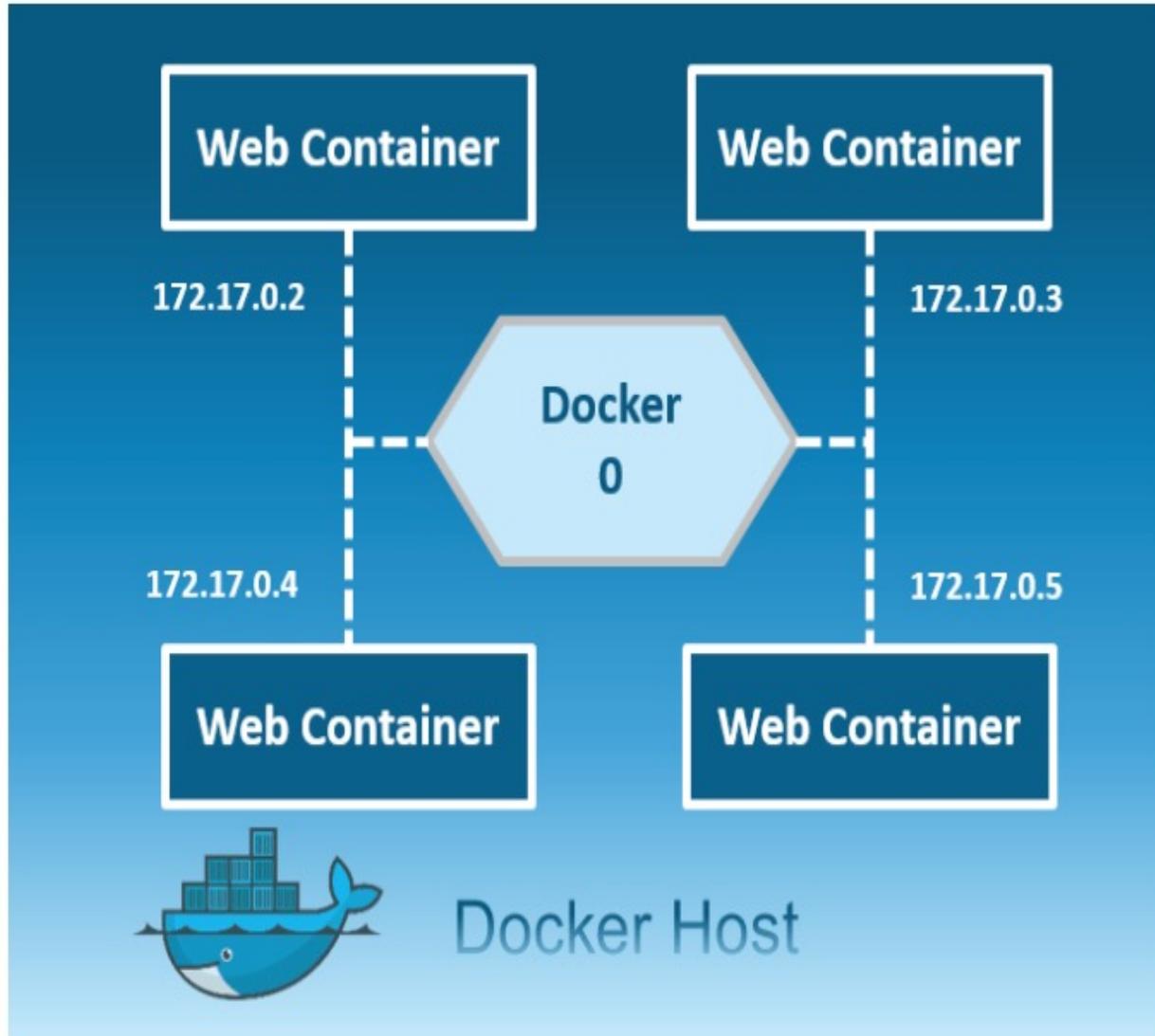
Service, Ingress



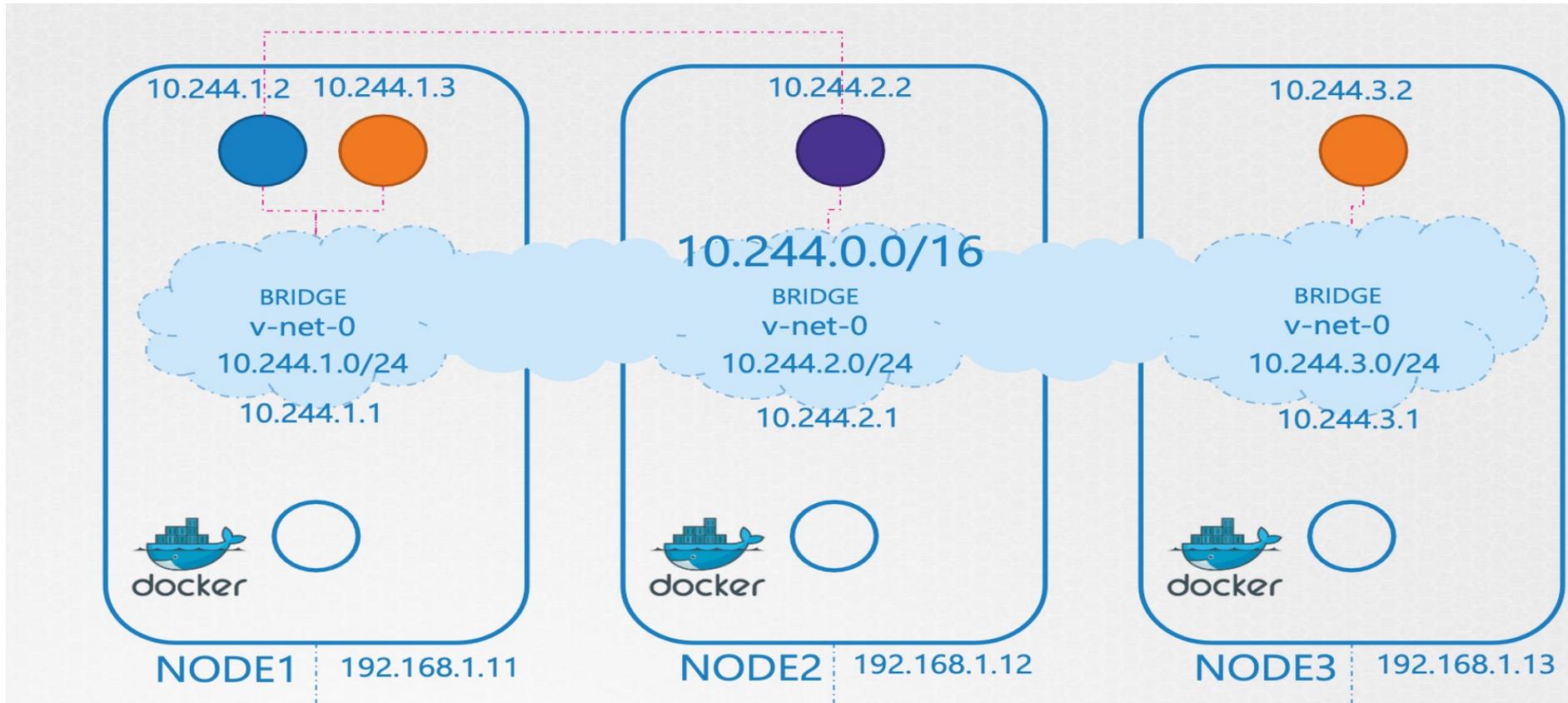
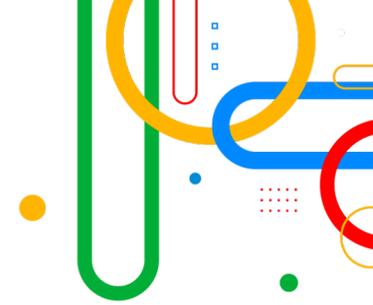
Network Policy



# Background...



# Kubernetes network model



# Container Networking Interface (CNI)

1. Create Network Namespace

2. Create Bridge Network/Interface

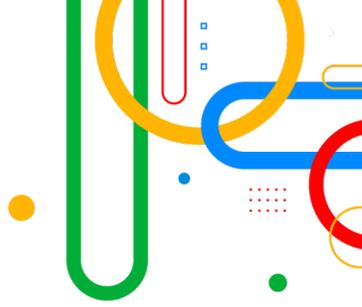
3. Create VETH Pairs (Pipe, Virtual Cable)

4. Attach vEth to Namespace

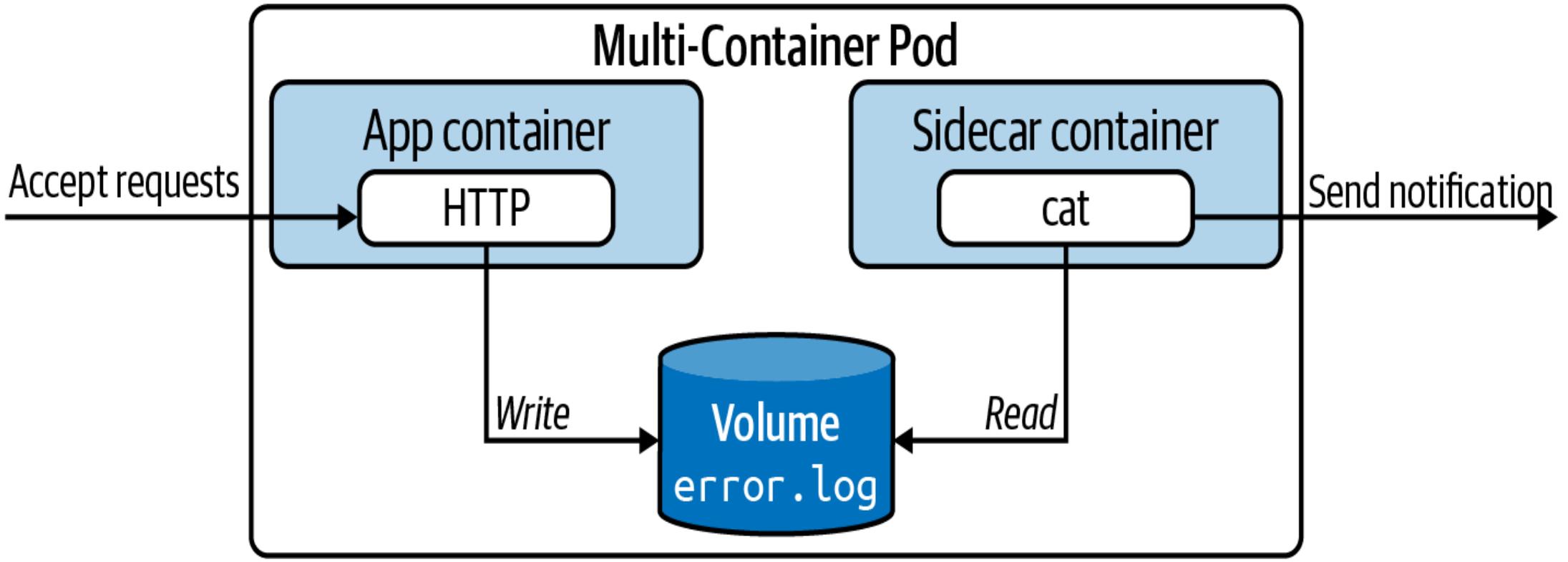
5. Attach Other vEth to Bridge

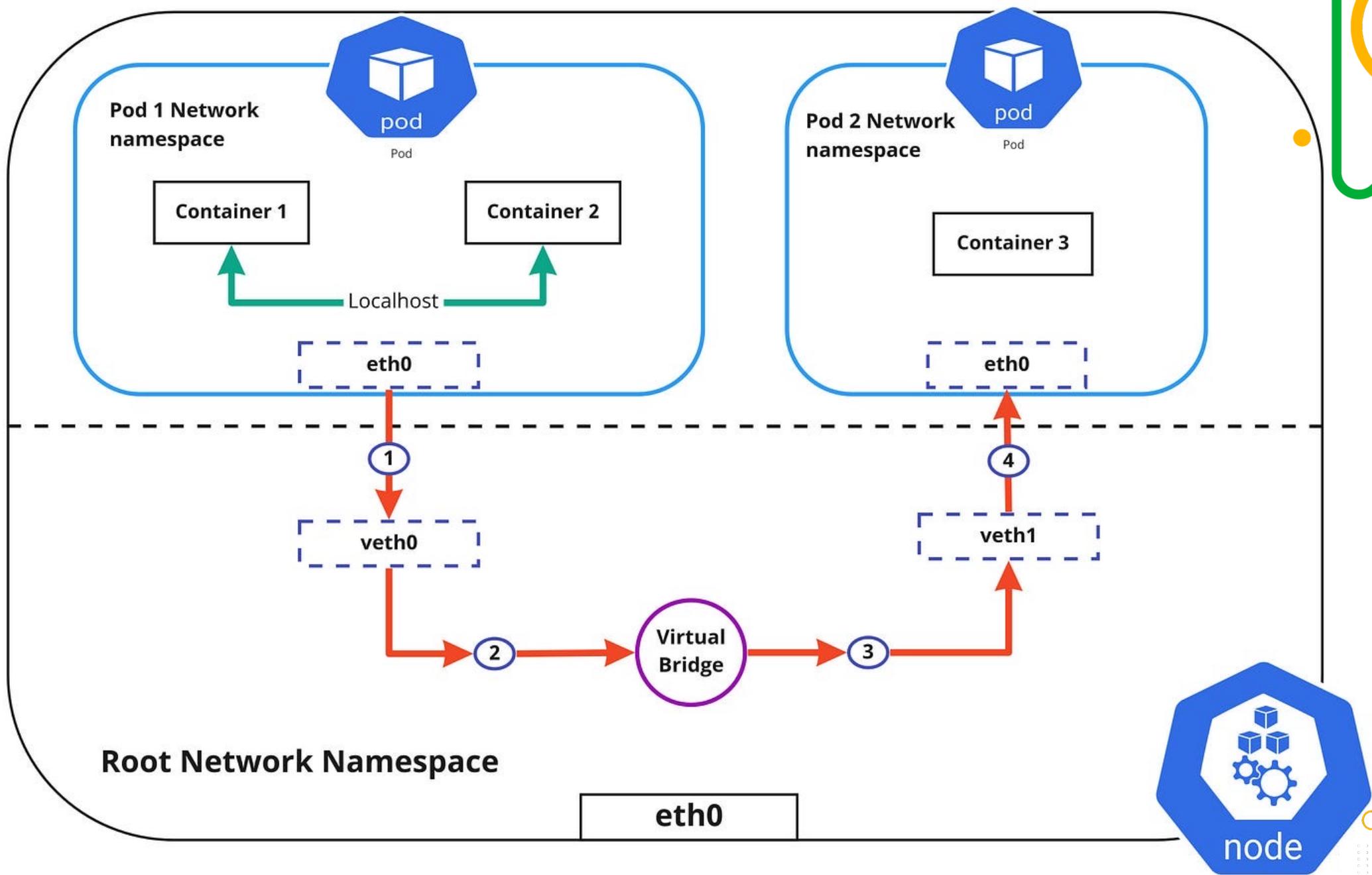
6. Assign IP Addresses

7. Bring the interfaces up



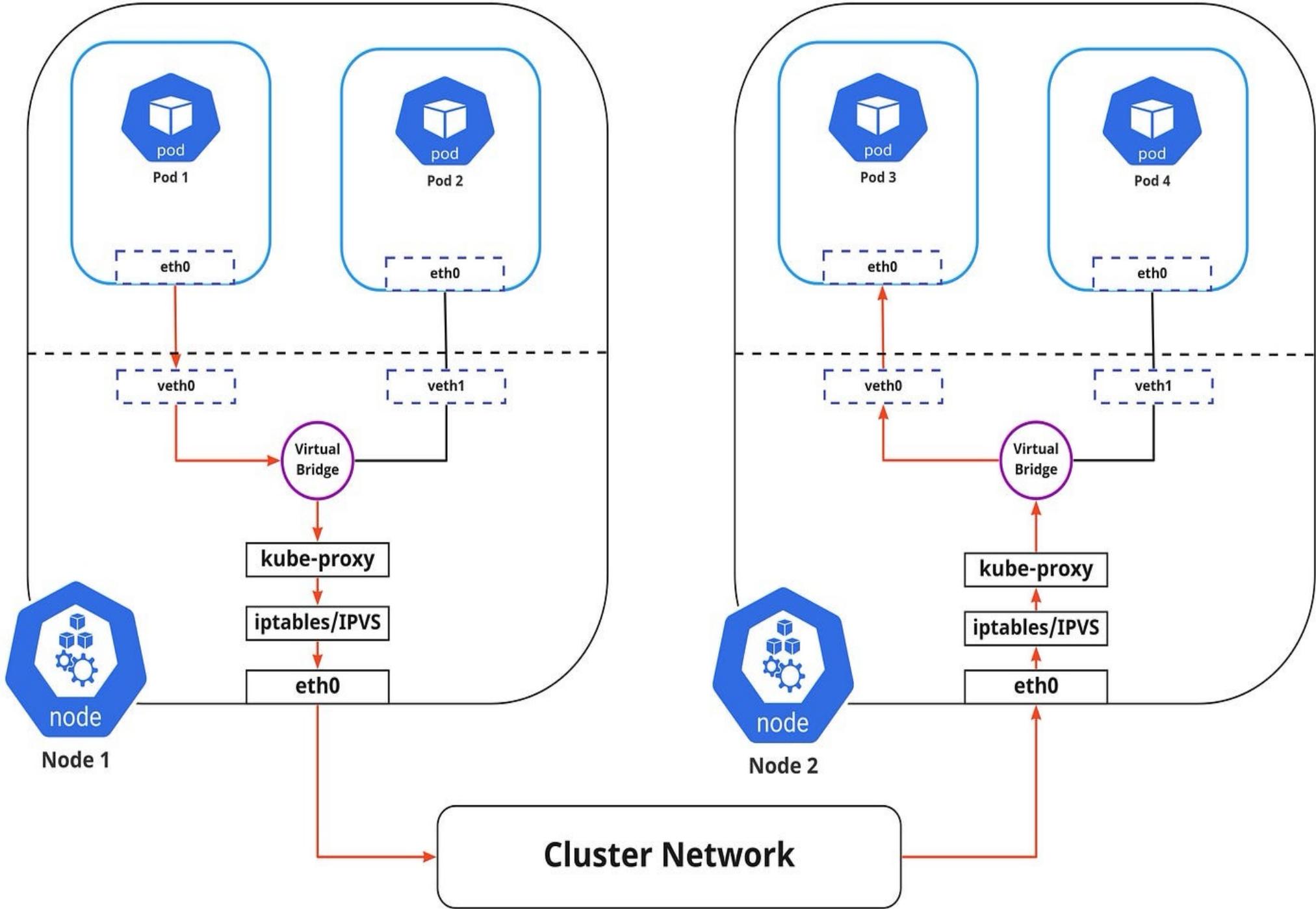
# Kubernetes multi-container communication





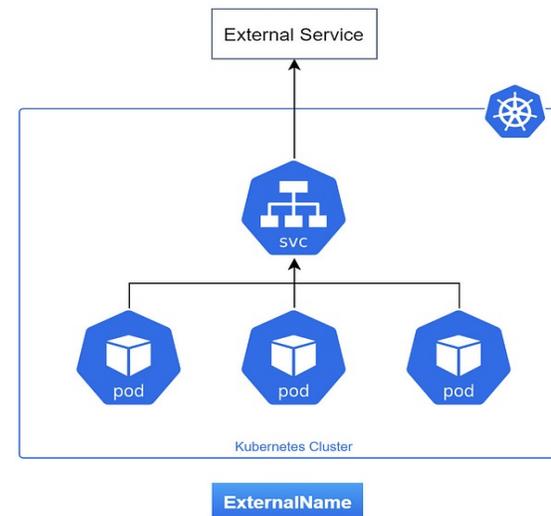
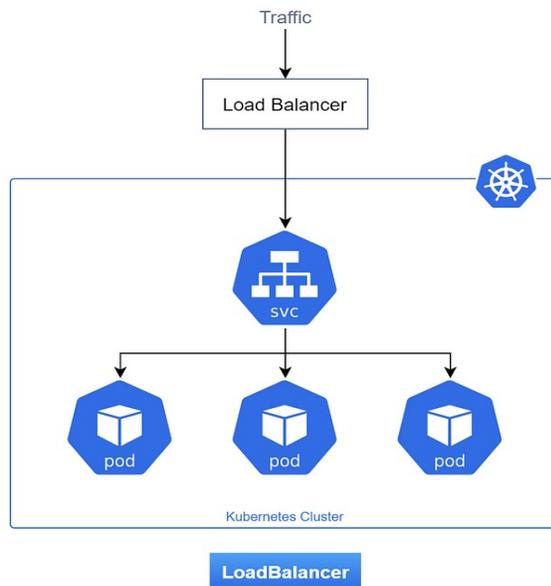
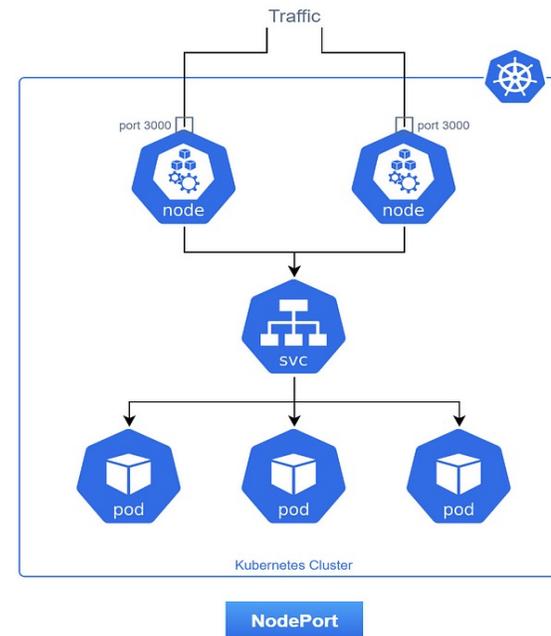
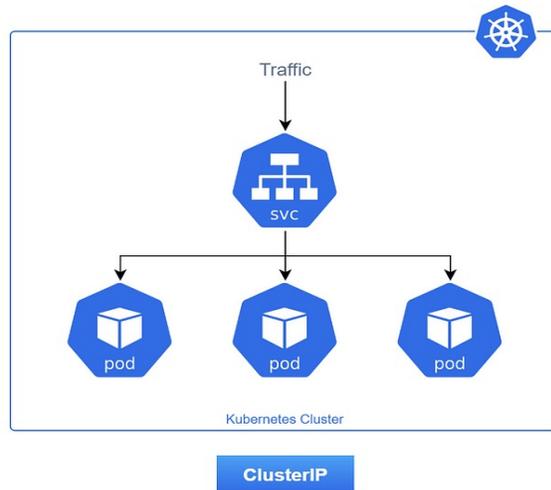
**Container-to-Container & Pod-to-Pod Networking**

Node 1



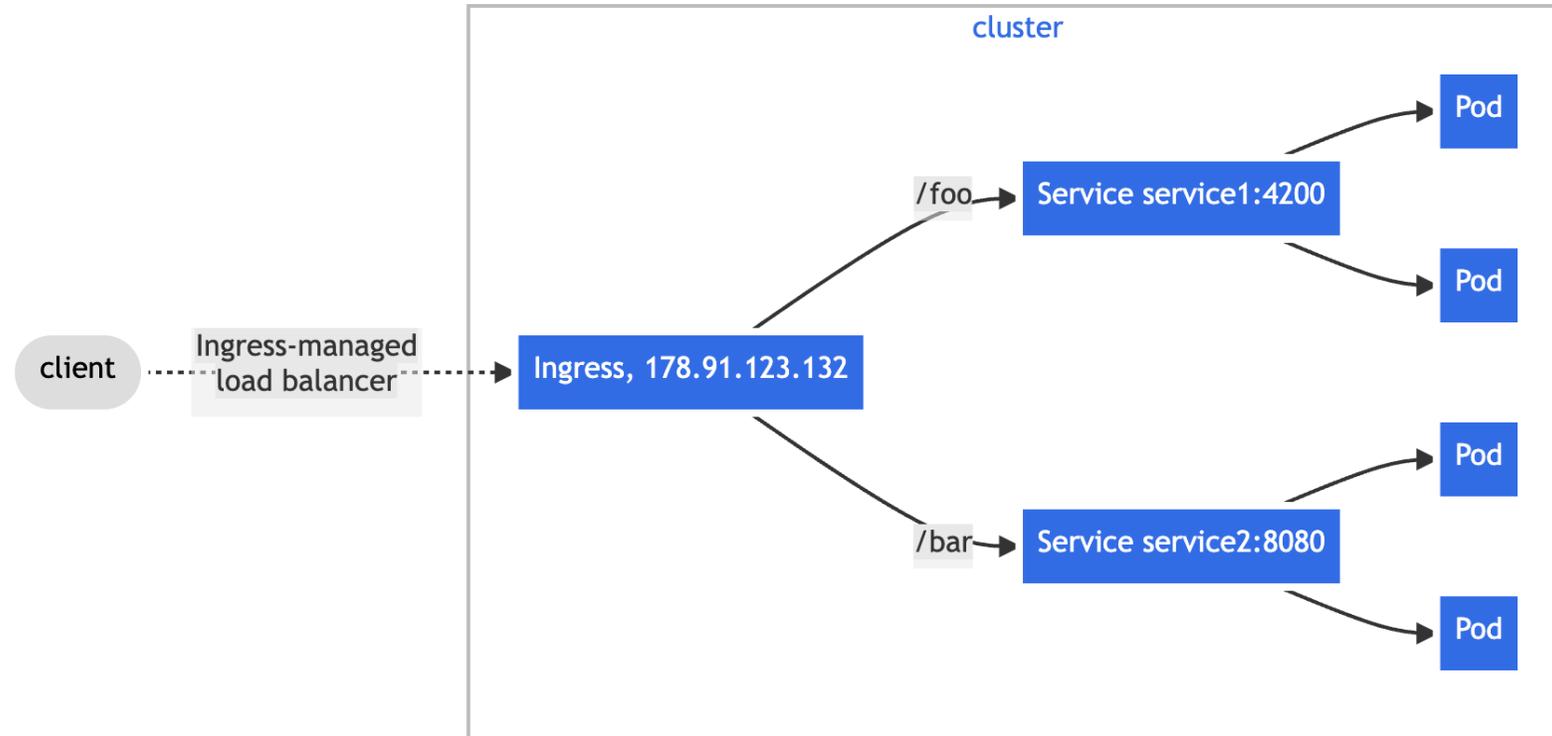
# Service

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

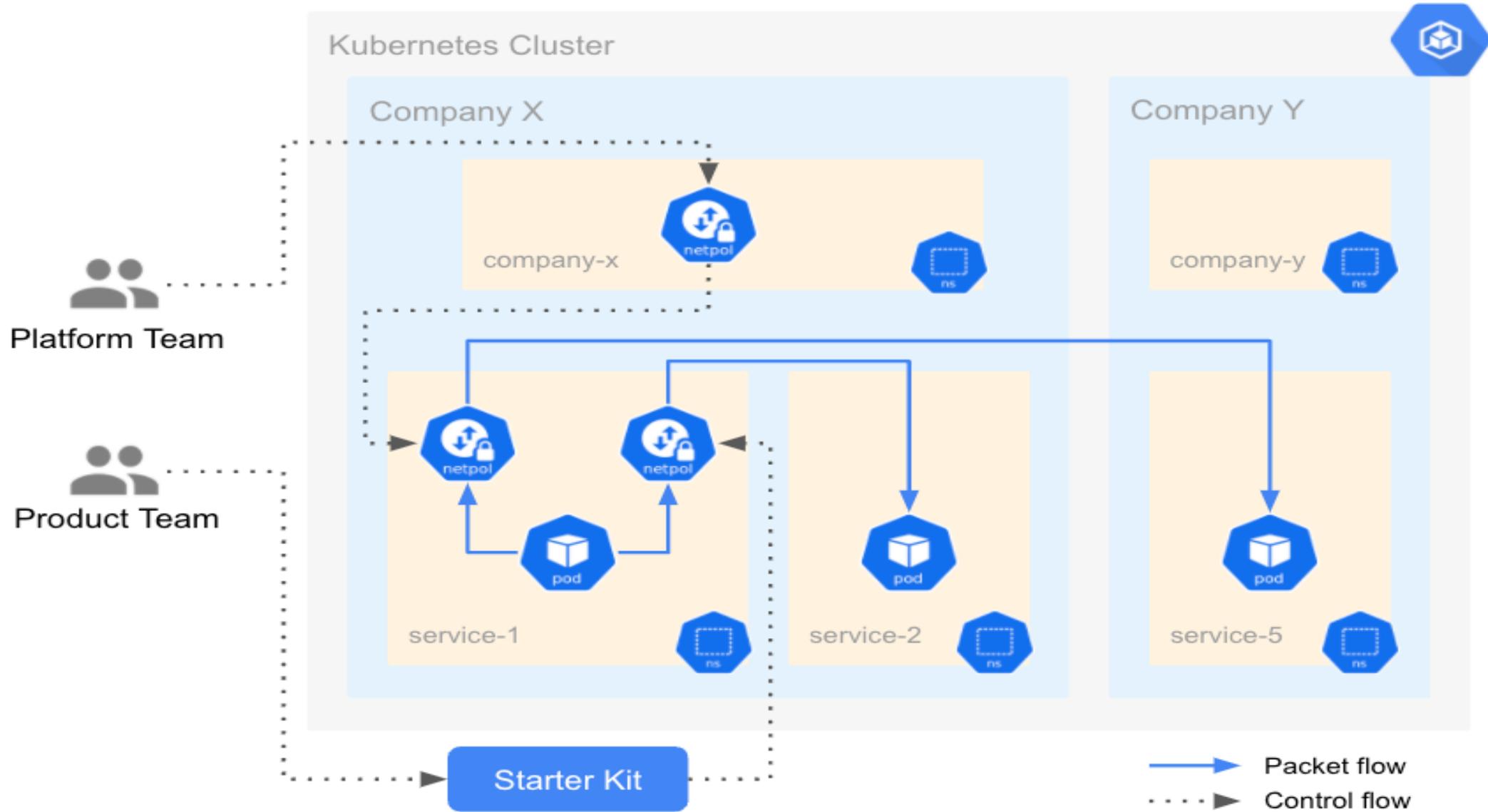


# Ingress

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.



# Network Policy



```
Text('Section Title',  
style: TextStyle(  
color: Colors.green[200],  
),  
),  
),  
),
```

# devfest

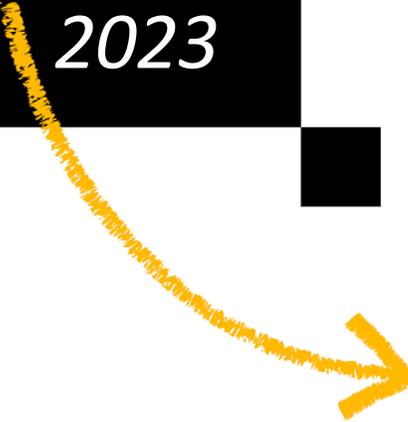
```
s.star,  
r: Colors.green[500],  
Text('23'),
```

2023



Google Developer Groups

Noida



# The End

Ankit Gupta

[gupta.smile@gmail.com](mailto:gupta.smile@gmail.com)

