

Kubernetes Networking: Under the Hood and Beyond

Bashir Ahmed Zeeshan
Pakistan

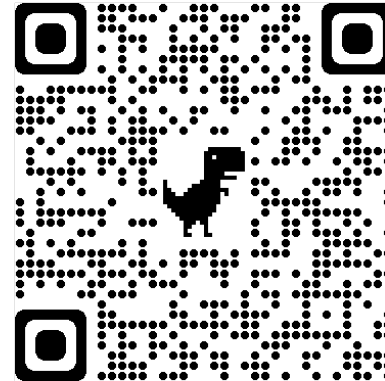
Making complex **technical** concepts, *easy* to grasp



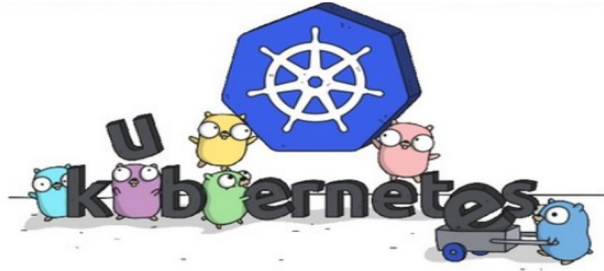
Bashir Ahmed Zeeshan

 [\bashirahmedzeeshan](https://www.linkedin.com/in/bashirahmedzeeshan)

- IP/Telecom & **Cloud** Consultant
- **Technical Trainer** - DevOps | CKAD
- Multi-Cloud Enthusiast
- Supporter of academic & industry liaison



Hello!

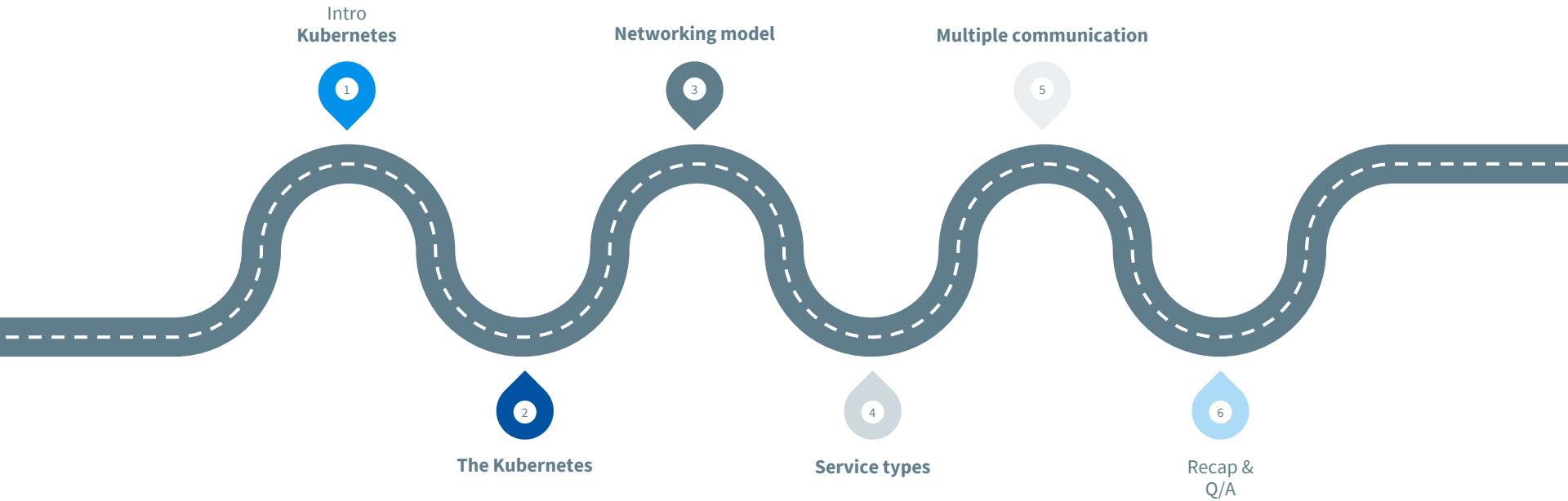


- Explain the **purpose** of Kubernetes Networking.
- **Identify the different types** of K8s Networking parts.
- Use Kubernetes Network to **traffic between pods/Nodes.**



Apply Kubernetes Network to **route Kubernetes applications**

The Kubernetes Network Model 101



World biggest Search Engine is running
Two Billion Containers per week !!!

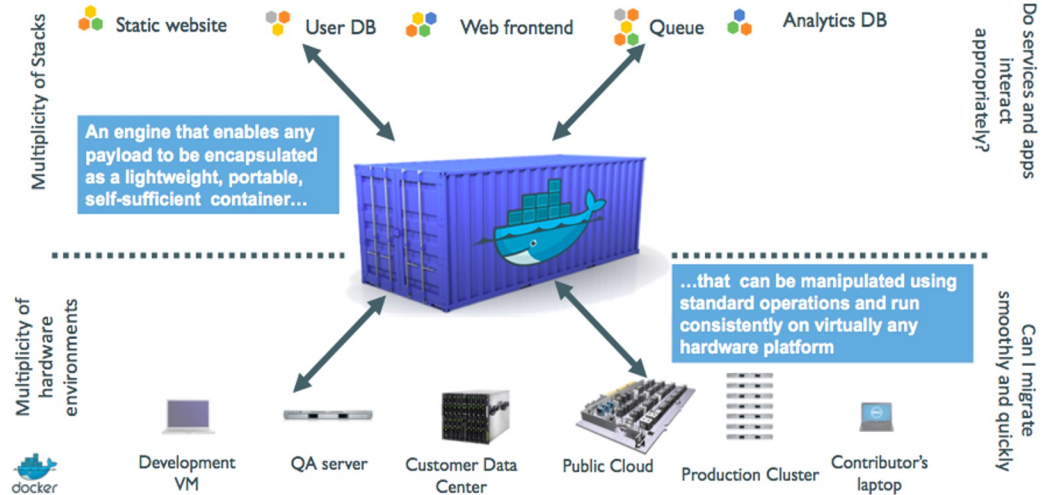
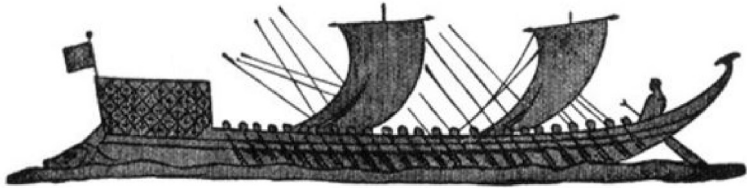


What is Kubernetes

= A Production-Grade Container Orchestration System

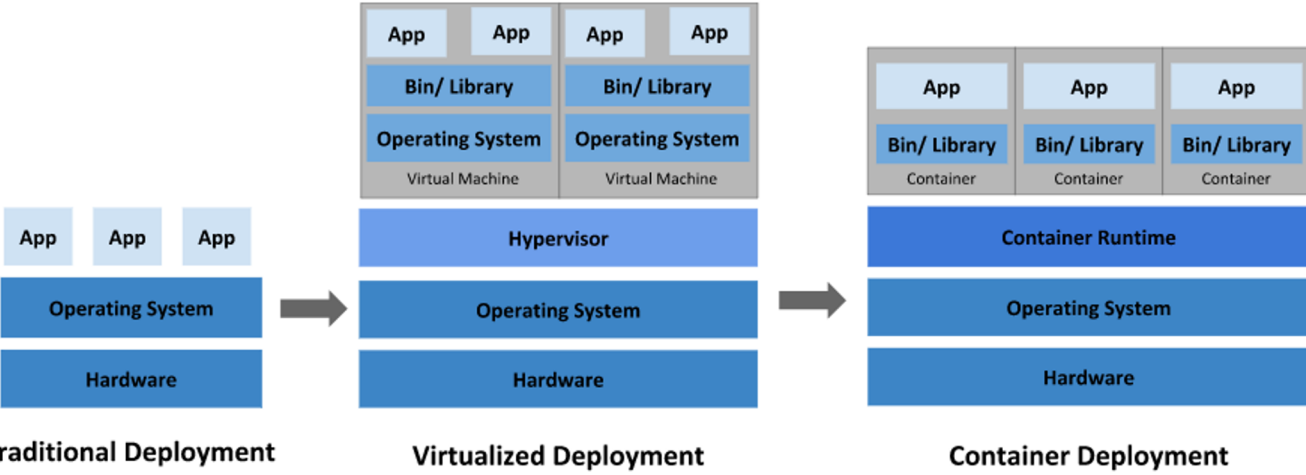


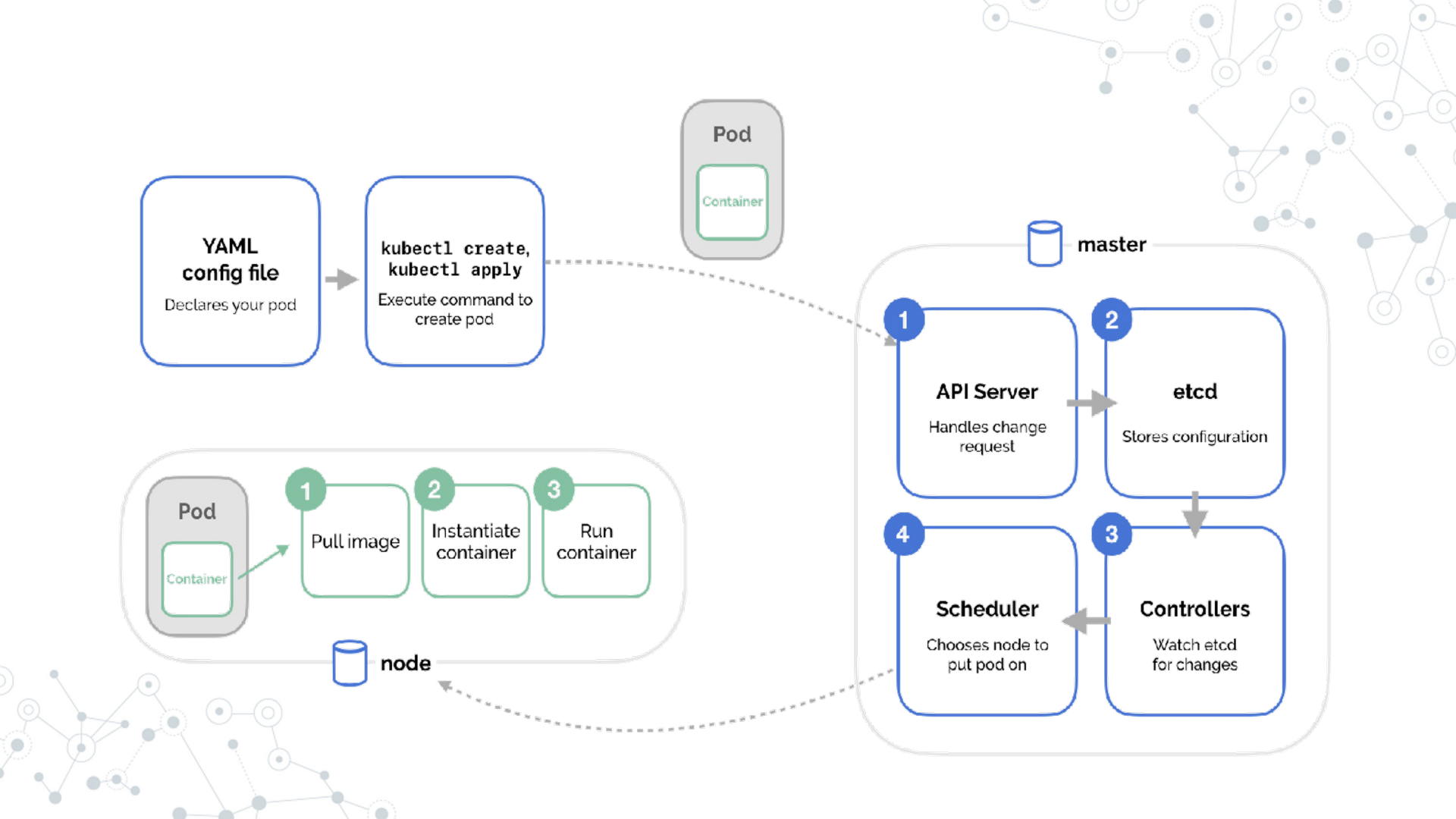
Greek for “pilot” or
“Helmsman of a ship”



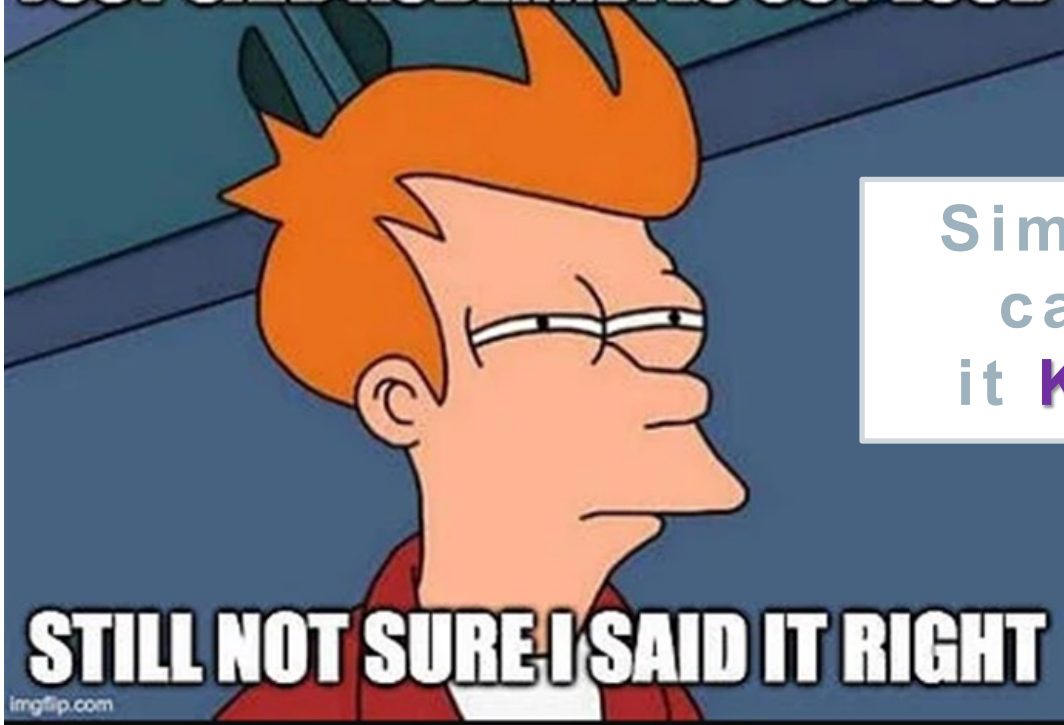
Before Kubernetes

Let's take a look at why Kubernetes is so useful by going back in time.





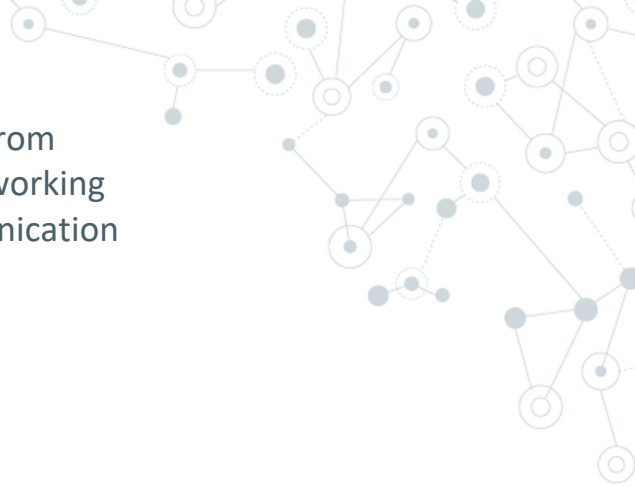
JUST SAID KUBERNETES OUT LOUD



STILL NOT SURE I SAID IT RIGHT

imgflip.com

Simply
call
it **K8s**

A network diagram in the top right corner showing interconnected nodes and lines, representing a network structure.

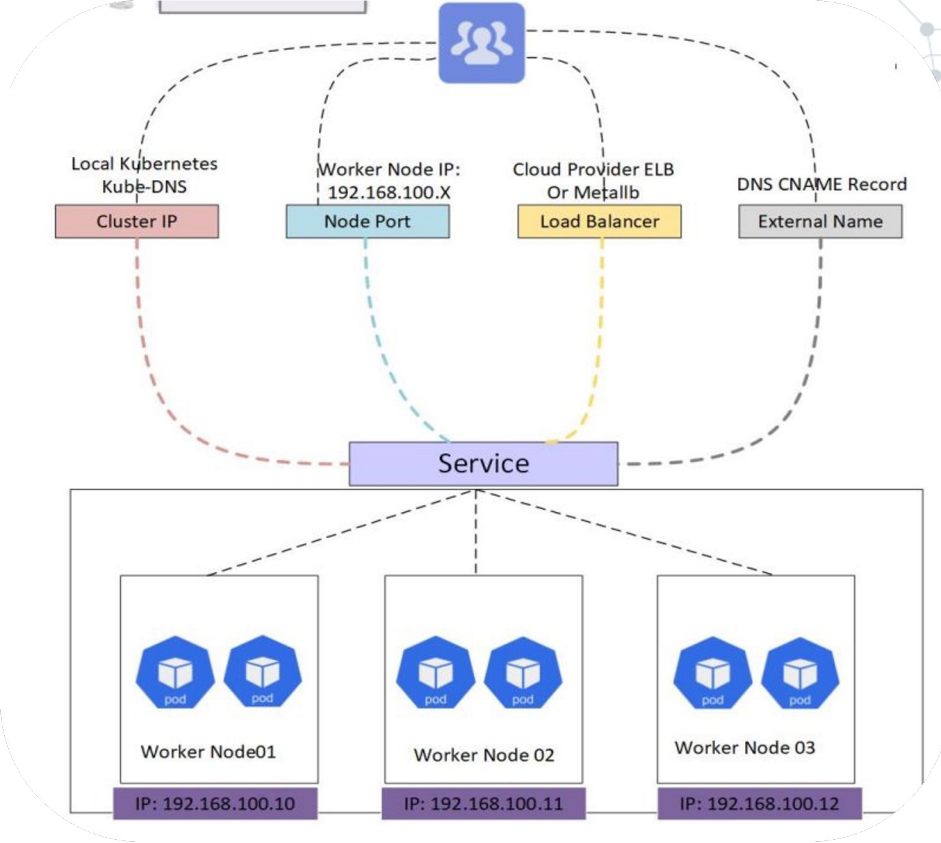
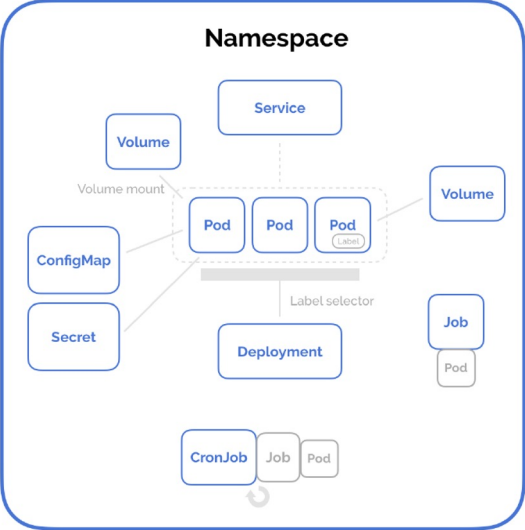
Networking within Kubernetes is so different from networking in the physical world. Remember networking basics, and you'll have no trouble enabling communication between containers, Pods, and Services.

Networking 101

Kubernetes World

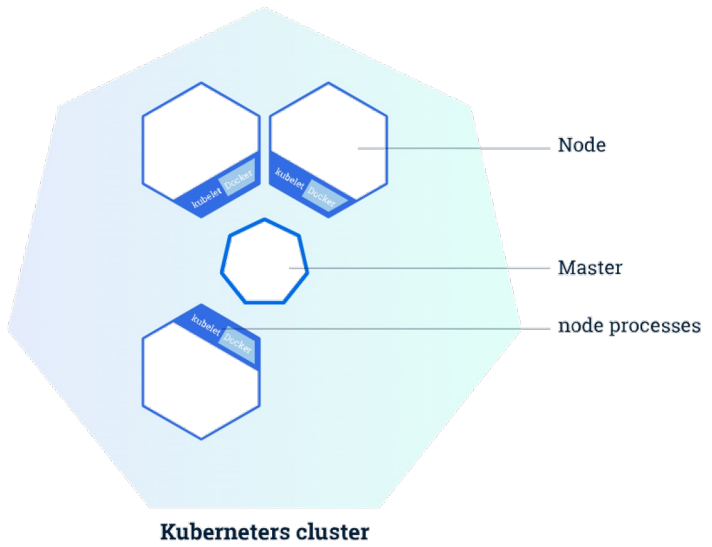
A network diagram in the bottom left corner showing interconnected nodes and lines, representing a network structure.

Networking Plan in K8s



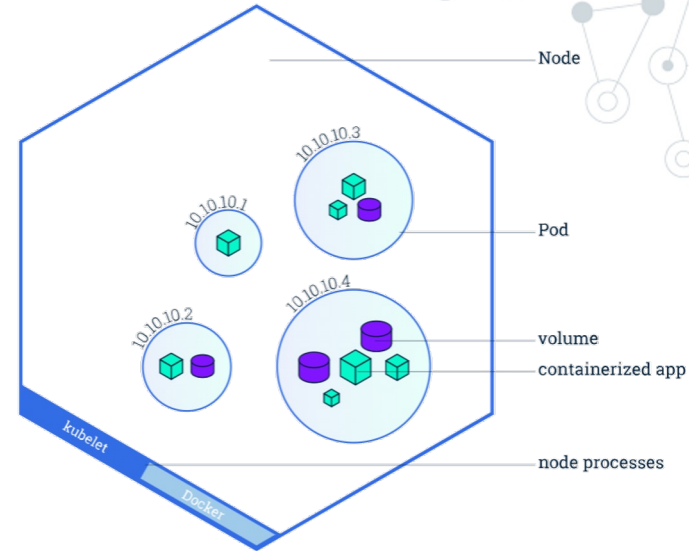
Before Networking

- Nodes
- Master node
- Pods
- Containers



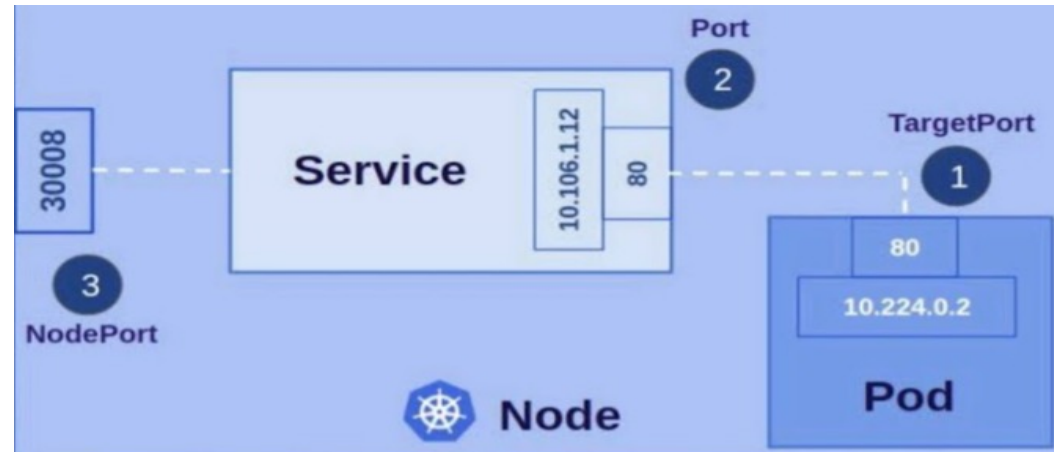
Kubernetes network model - the good

- ✓ **TL,DR:** Our cluster (nodes and pods) is one big flat IP network.
- ✓ In detail: without NAT
 - ✓ all nodes / pods must be able to reach each other,
 - ✓ pods & nodes must be able to reach each other
 - ✓ each pod is aware of its IP address (no NAT)
- ✓ Pods cannot move
- ✓ Not "portable"



Kubernetes network model - less good

- ⊙ Everything can reach everything
 - Security??
- ⊙ Dozens of implementations out there
- ⊙ Pods have L3(IP) connectivity, but services are on L4

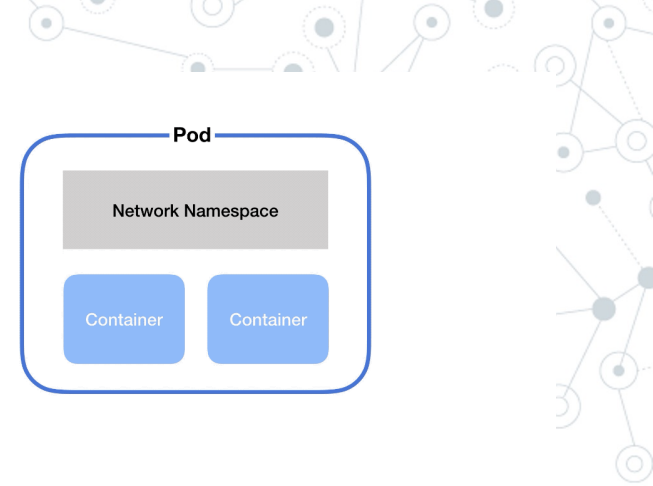


5 Ways to communicate!

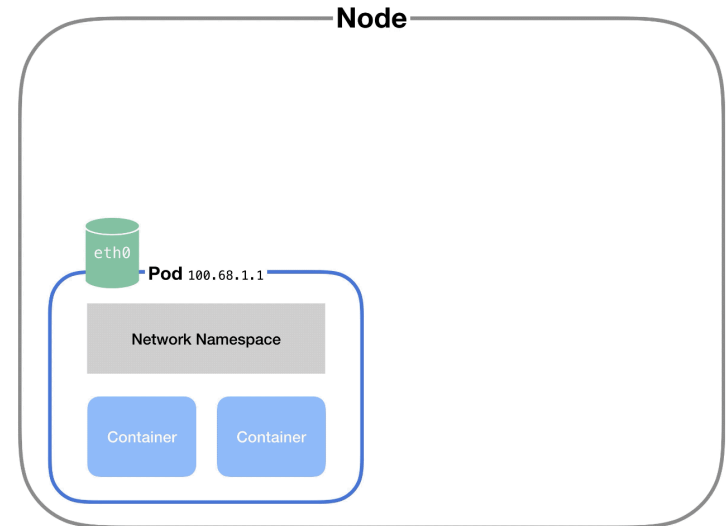
1. Communication between **containers** in the *same pod*
2. Communication between **Pods** on the *same node*
3. Communication between **Pods** on *different nodes*
4. Communication between **Pods** and **services**
5. How do we discover IP addresses?

K8 Networking

- ❑ **Container to container / communication within a pod**
 - Through localhost and the port number



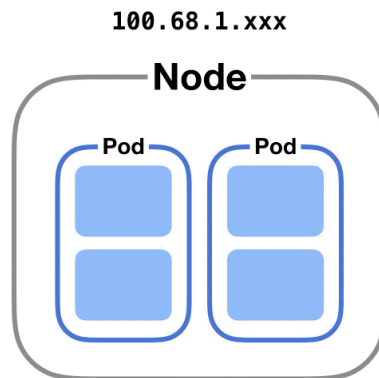
- ❑ **Between Pods on the same node**
 - own network namespace; Pod has its own IP, a veth (virtual ethernet)



K8 Networking

Between **Pods** on the **different node**

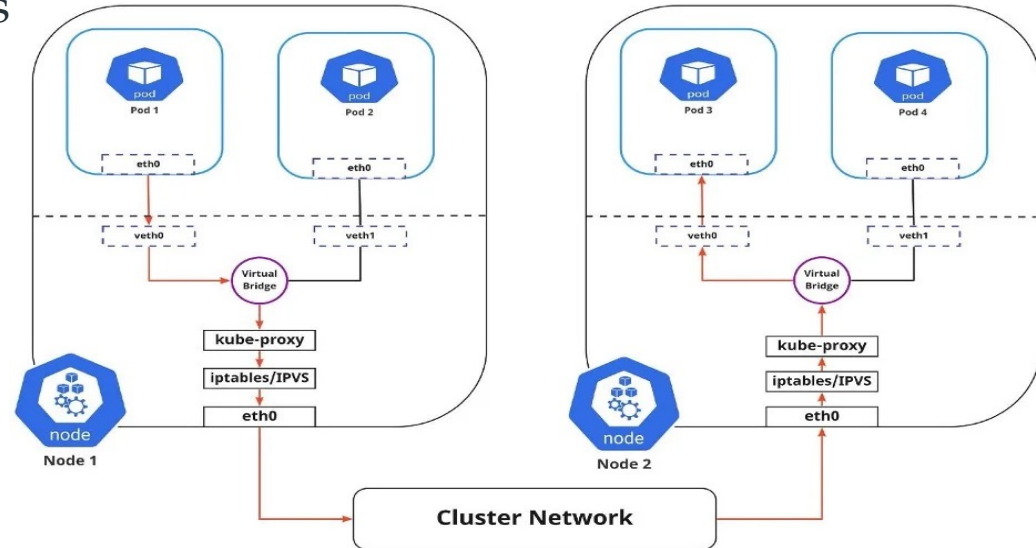
- bridge falls back to the default gateway.
- table that maps IP address ranges to various node
- provides communication between pods and nodes
- generally implemented with CNI plugins



K8 Networking

Pod-To-Service communication

- Pods are Dynamic! Scaling, Recreation .. IP Changed!
- Here comes , Services !
 - Assign static virtual IP addres
 - Load Balancing
 - Keeps tracking Pod IP

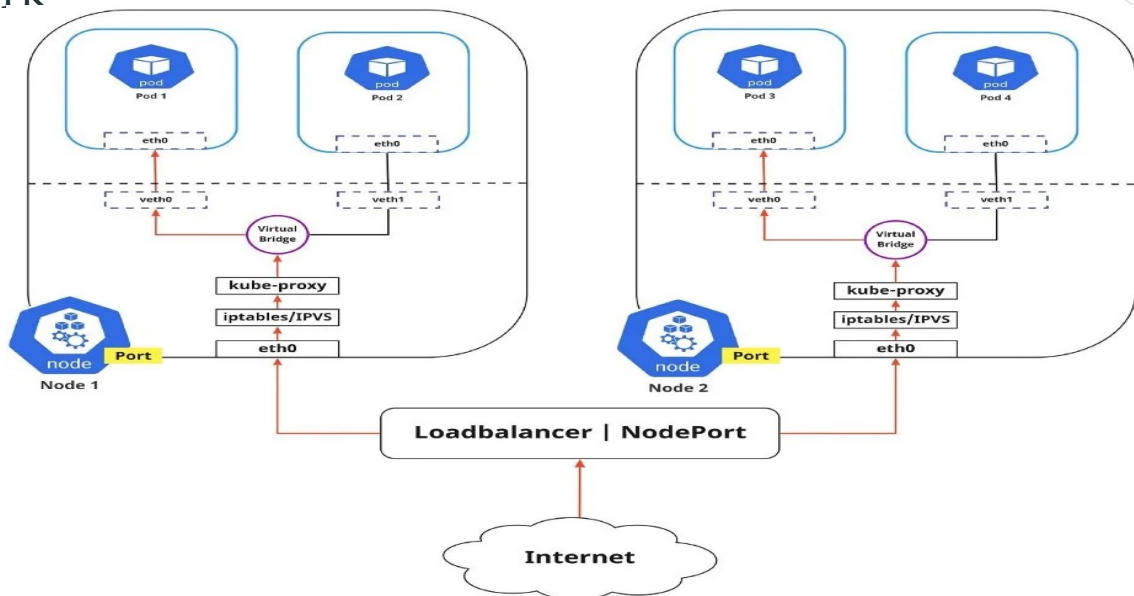


K8 Networking

❑ Internet-to-Service networking

- Using LoadBalancer, NodePort
- Expose to external network
 - Egress
 - Ingress

A cluster-aware DNS server, such as CoreDNS, watches the Kubernetes API for new Services and creates a set of DNS records for each one.



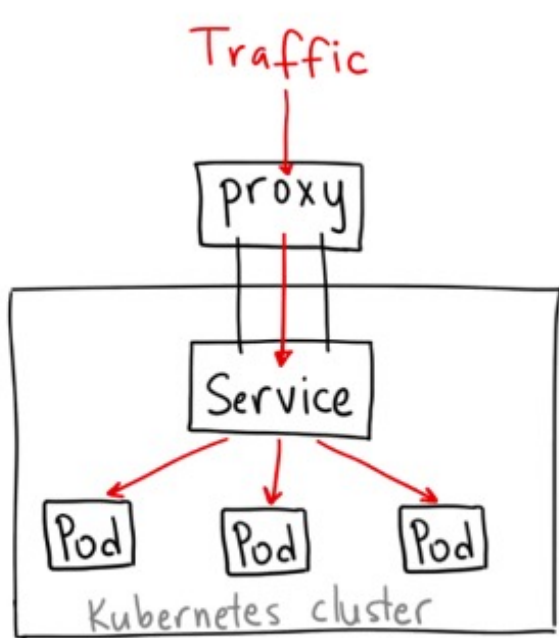
K8 Networking

❑ Discovering Services

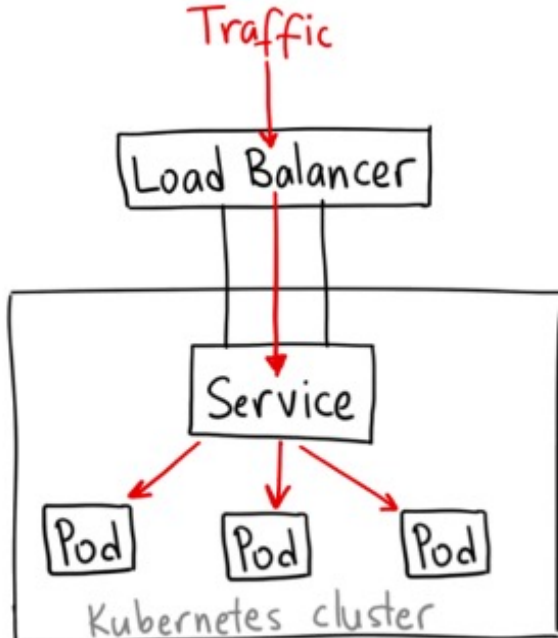
- Environment Variables
- **CoreDNS**
 - Built-in DNS
 - Automatically resolve service name

Services

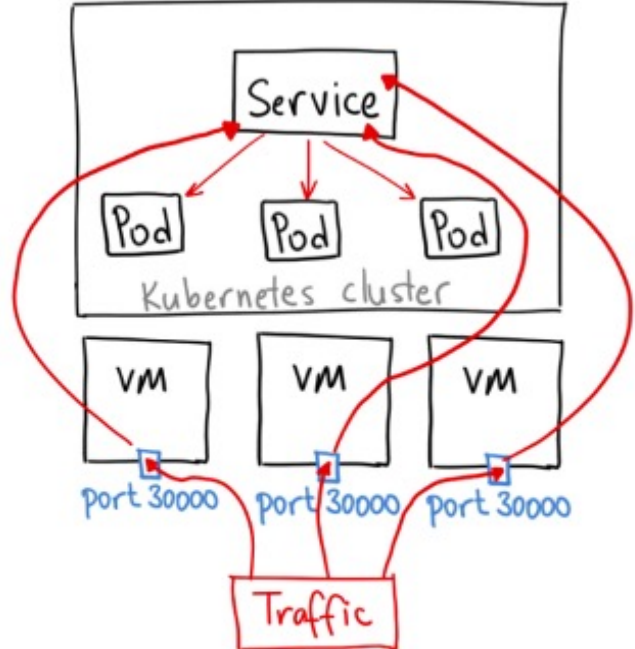
- Services provide you with a way of accessing a group of Pods



ClusterIP



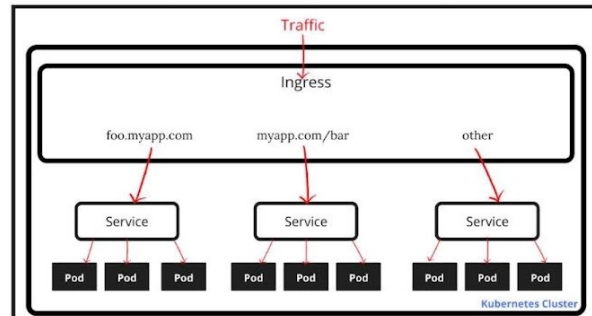
Loadbalancer



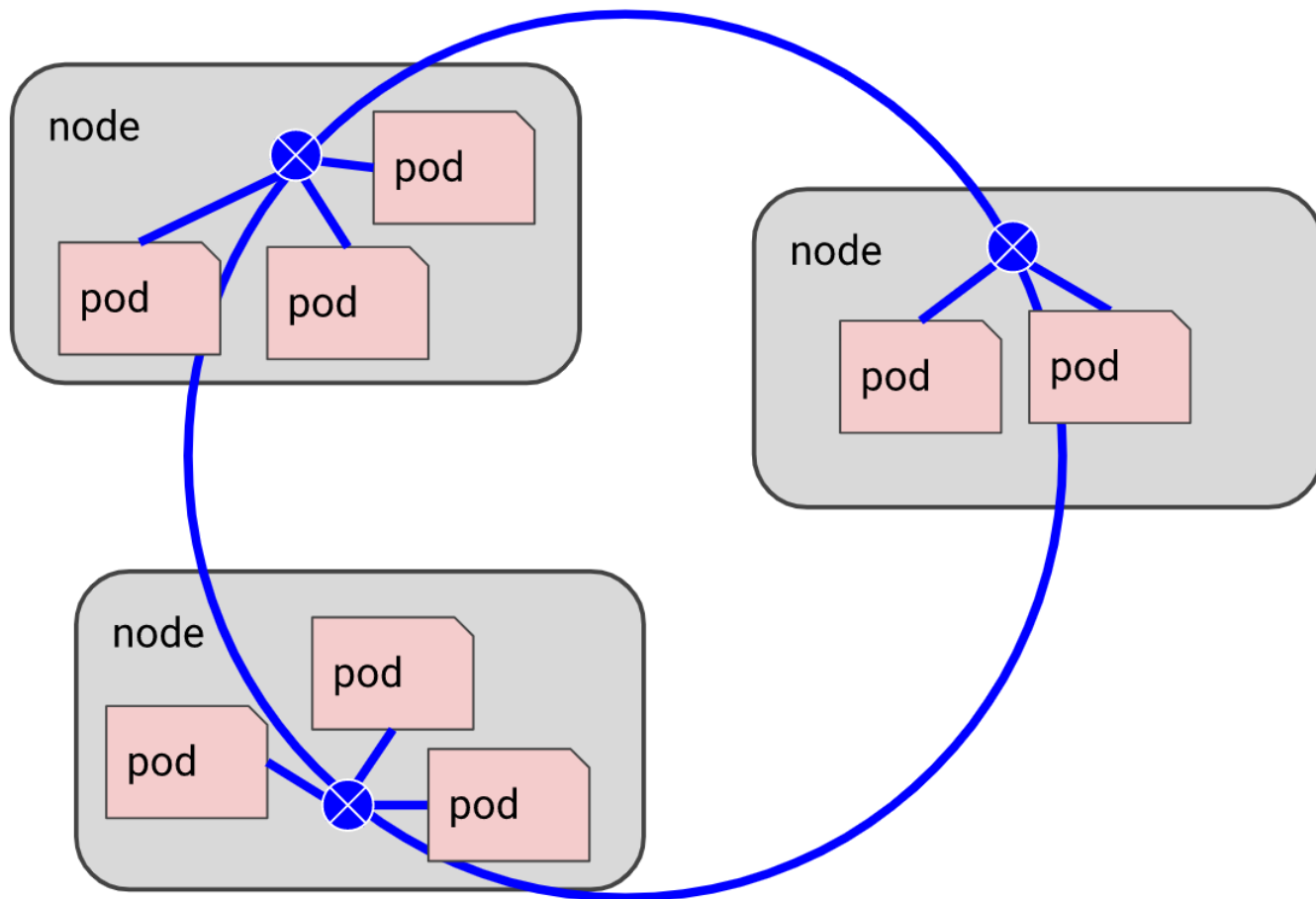
NodePort

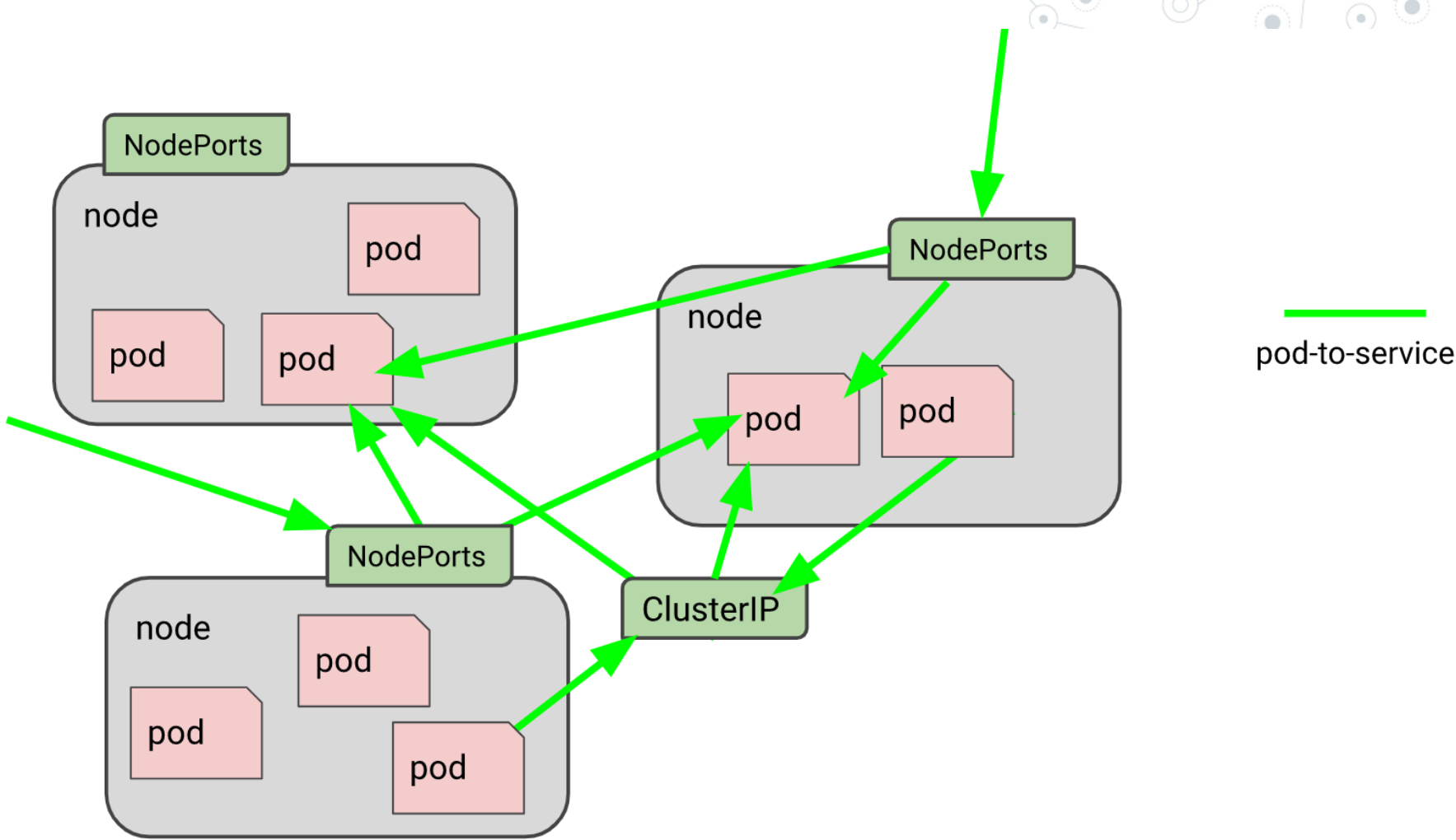
Services in Kubernetes

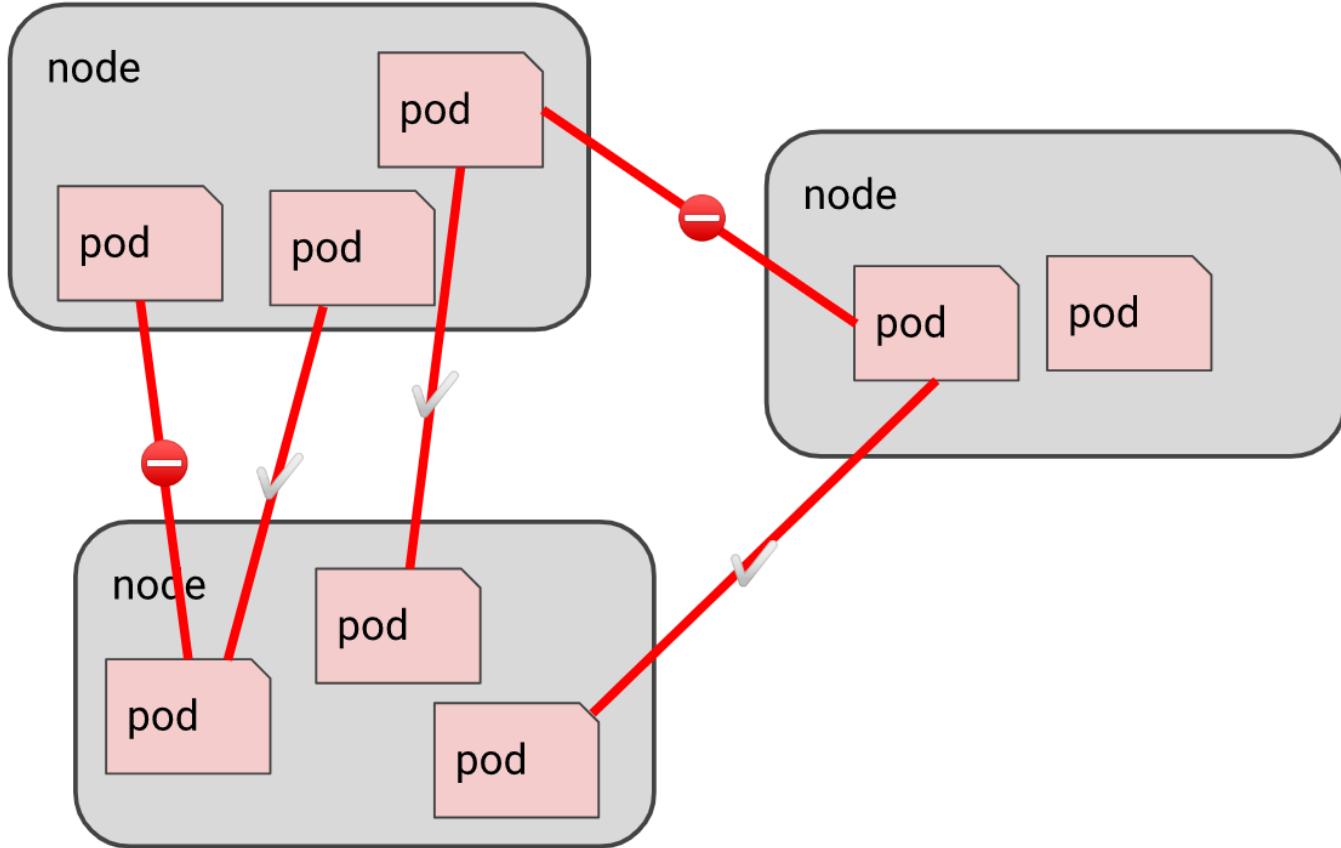
- ❑ **ClusterIP:** default one for internal communications, external traffic can access the default service through a proxy.
- ❑ **NodePort:** opens ports on the nodes or virtual machines, and traffic is forwarded from the ports to the service.
- ❑ **LoadBalancer:** standard way to connect a service externally to the internet. a network LB forwards all external traffic to a service. Each service gets its own IP address.



pod-to-pod

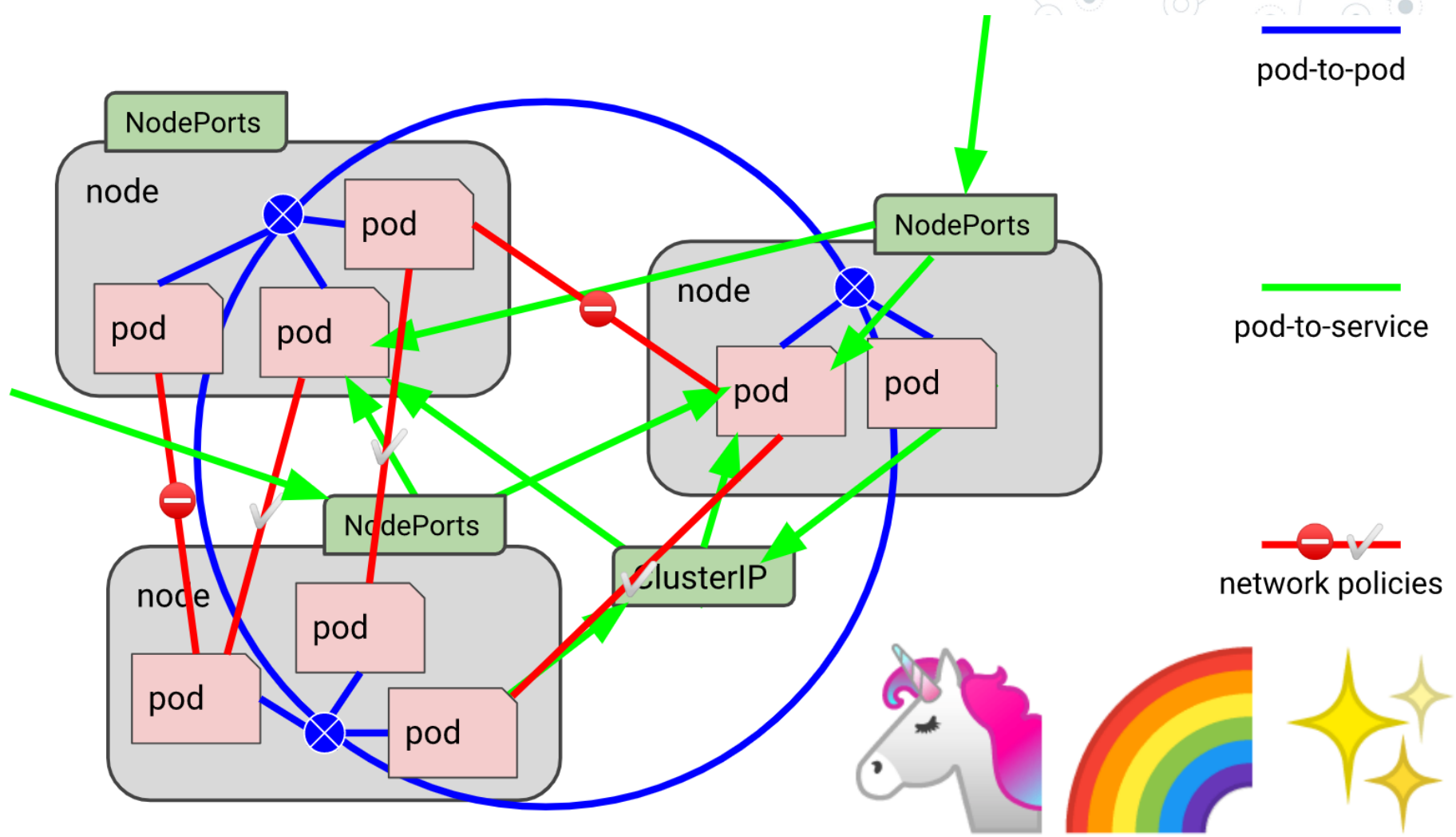






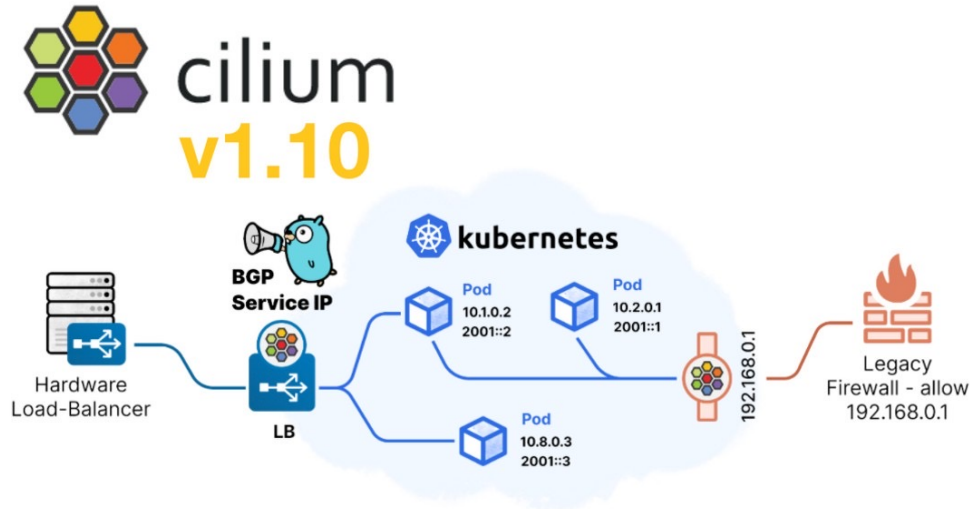

network policies





Cilium

- ◎ Open Source
- ◎ enables networking , security & Observability
- ◎ Provides High performance networking, multi-cluster and multi-cloud



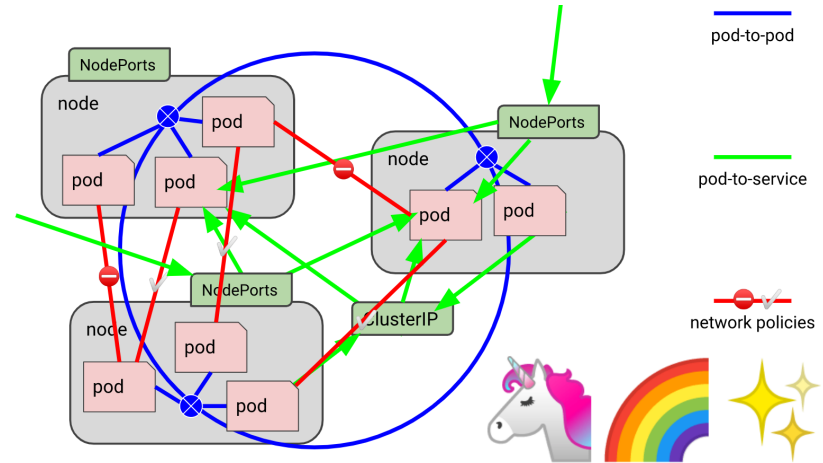
Some of Best practice

- ❖ Structuring Your Network for Scalability
- ❖ Effective Use of Service Discovery
- ❖ High Availability and Redundancy
- ❖ Security and Network Policies
- ❖ Monitoring and Performance Optimization



Recap

- ❖ Kubernetes
- ❖ Networking
- ❖ Key Items
- ❖ Services

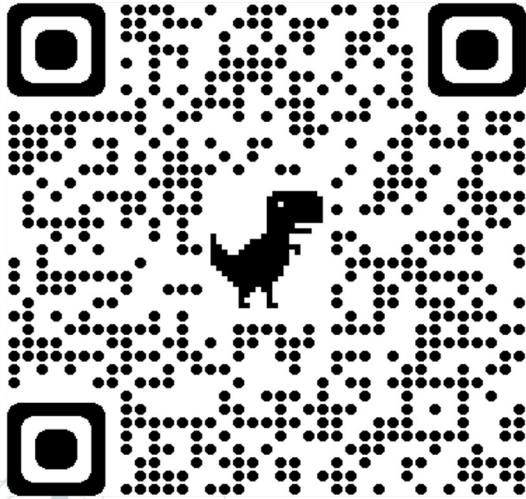


Thanks!



You can find me at:

<https://www.linkedin.com/in/bashirahmedzeeshan/>

A professional banner for Bashir Ahmed Zeeshan. It features several logos at the top: Oracle, AWS, and others. The text reads: **Bashir Ahmed Zeeshan**, IP & Telecom professional, Multi-Cloud Enthusiast, Lead Trainer & Technical Speaker. Below the text are icons for Cloud, Telecommunication, and IP Networks. At the bottom right, there is a photo of him speaking at a podium with a 'STUD' logo. Social media icons for Facebook, LinkedIn, Instagram, and Twitter are also present, along with the text 'ENGINEERBAZ'.