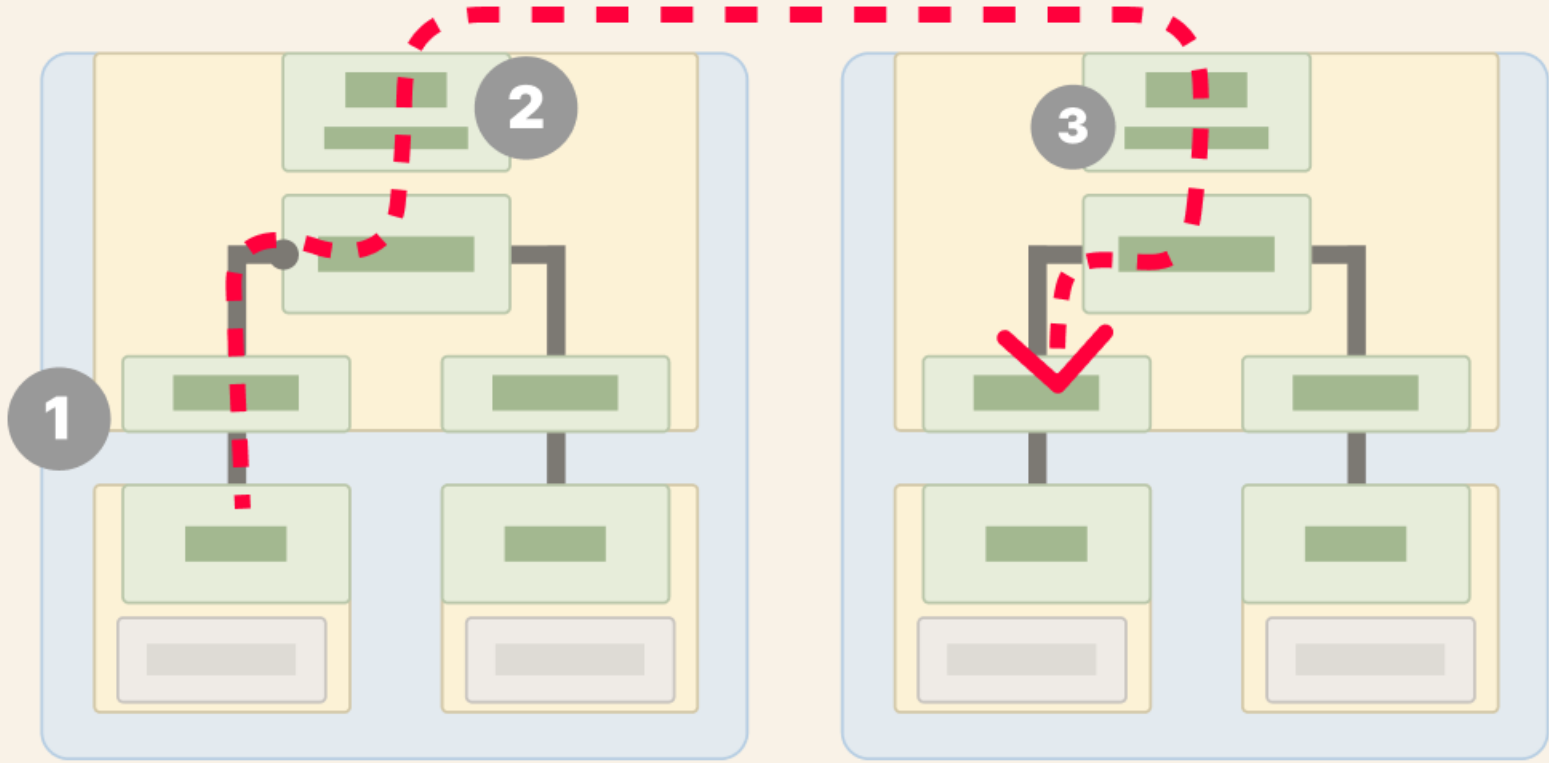
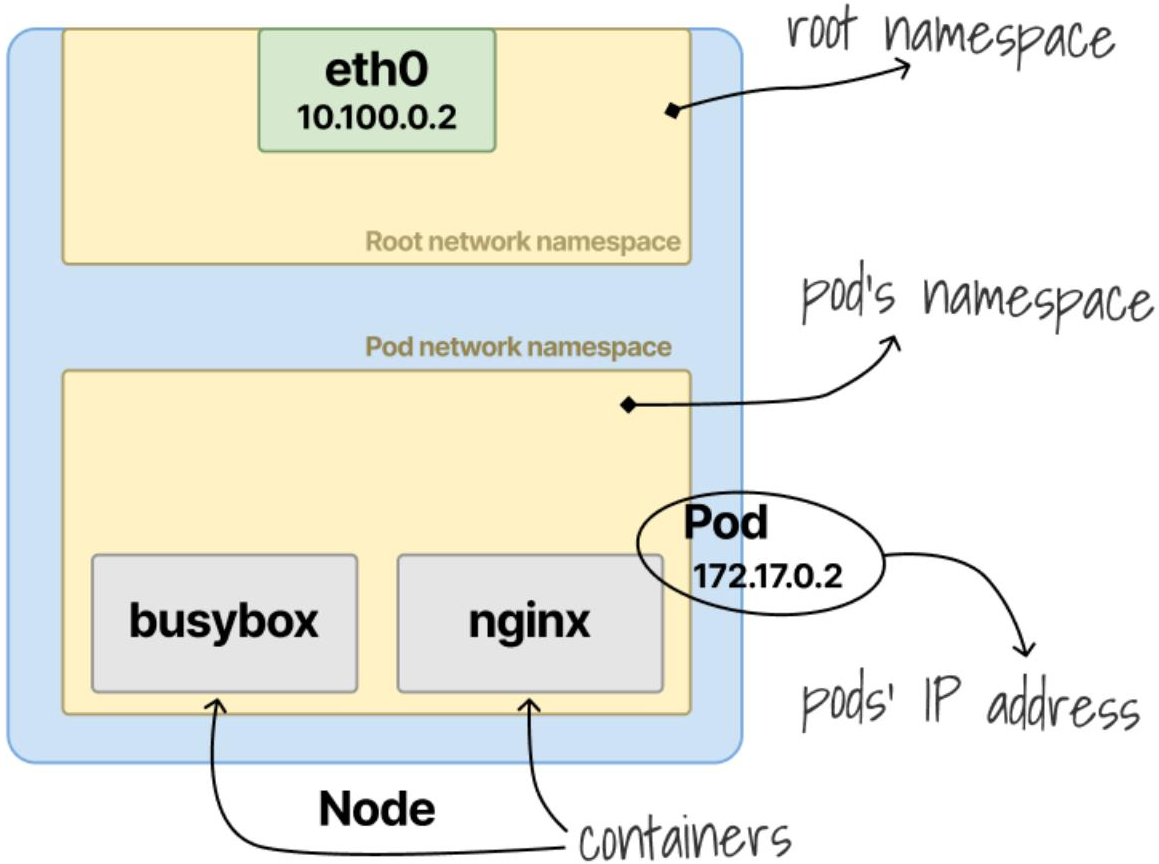
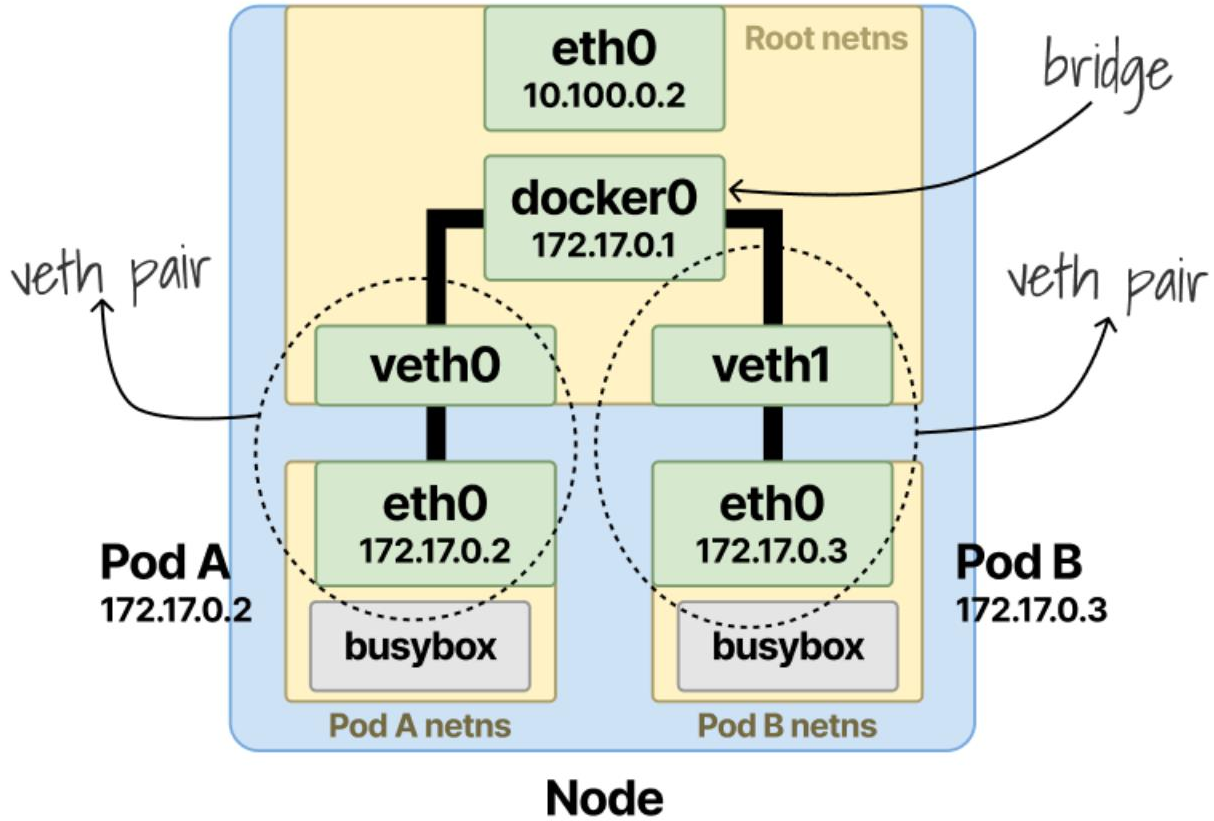
TRACING POD2POD NETWORK TRAFFIC in KUBERNETES

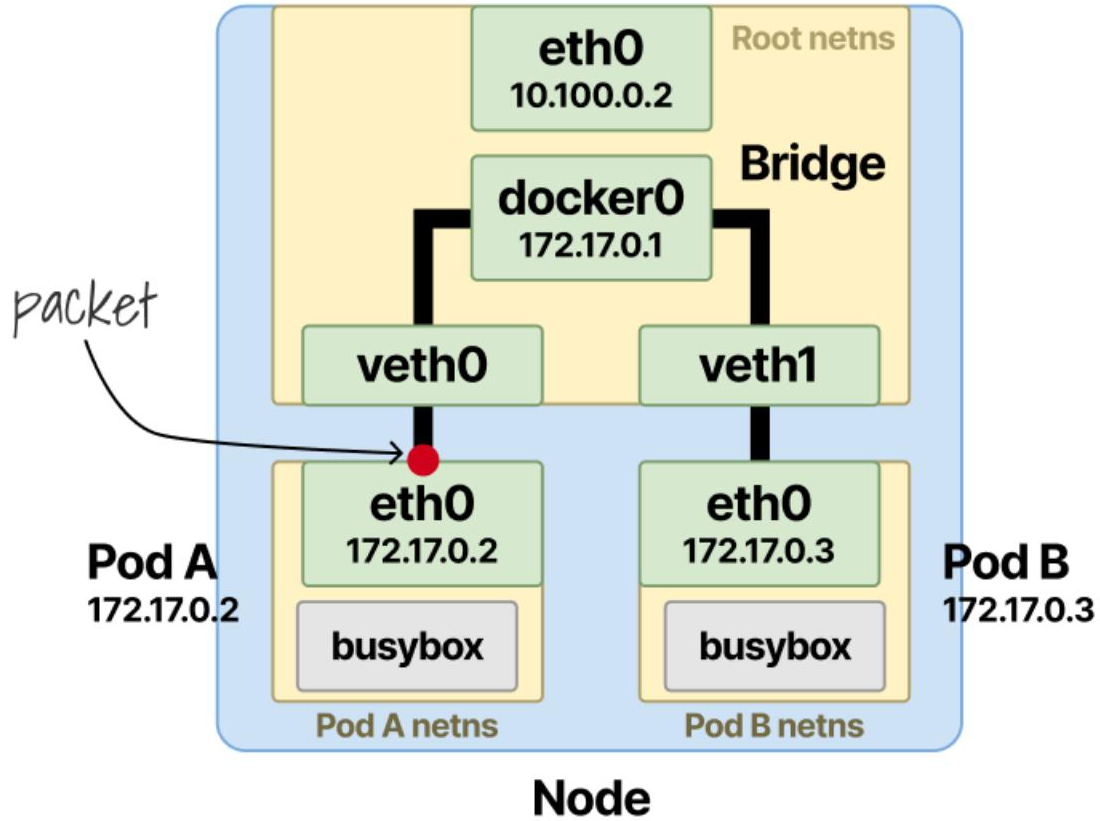




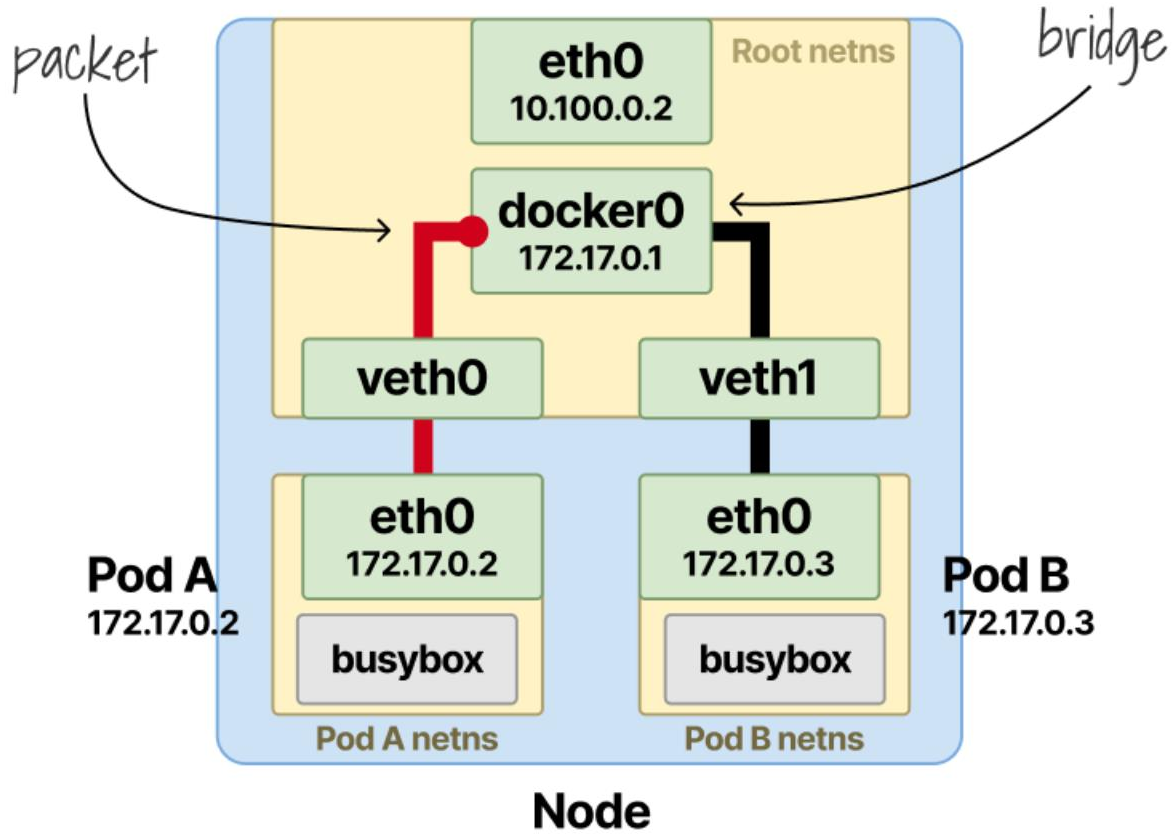
When you deploy a Pod, the following things happen:
1. The pod gets its own **network namespace**.
2. An **IP address** is assigned.
3. Any containers in the pod **share the same networking namespace** and can see each other on localhost.



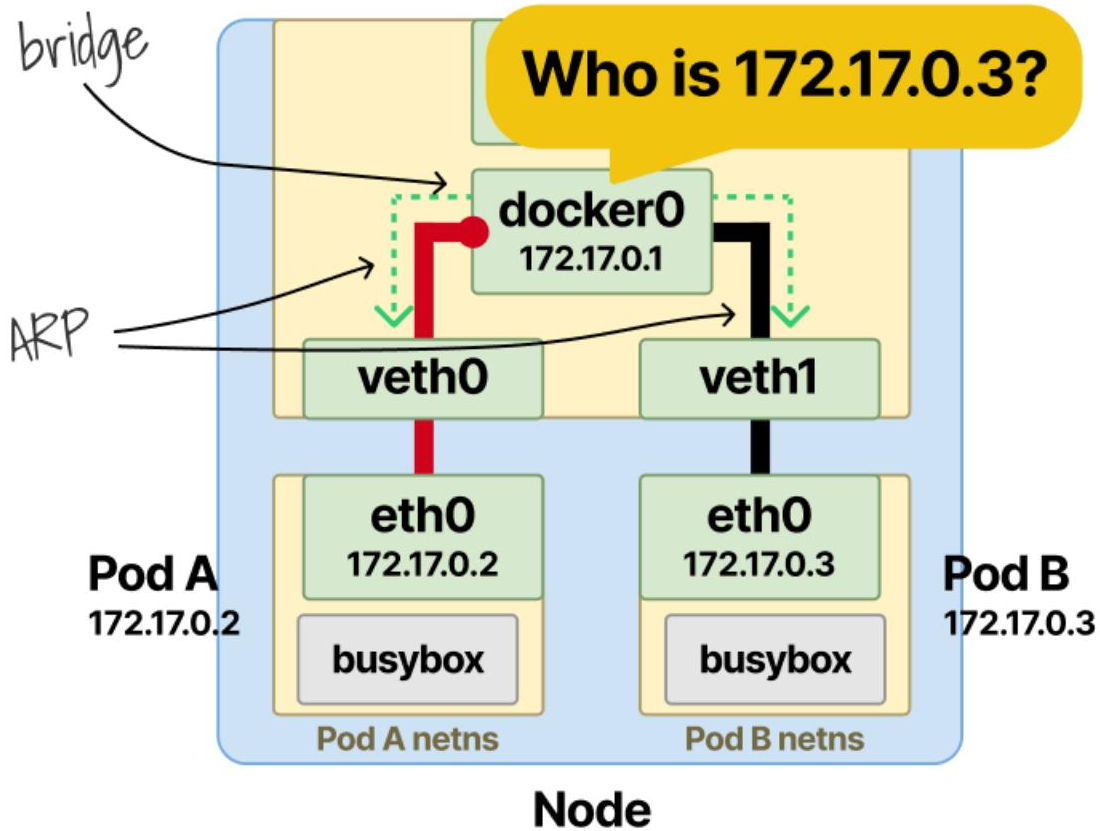
> For a pod to reach other pods it must first have access to the node's root namespace.
> This is achieved using a virtual eth pair connecting the 2 namespaces: pod and root.
> **The bridge allows traffic to flow between virtual pairs and traverse through the common root namespace.**



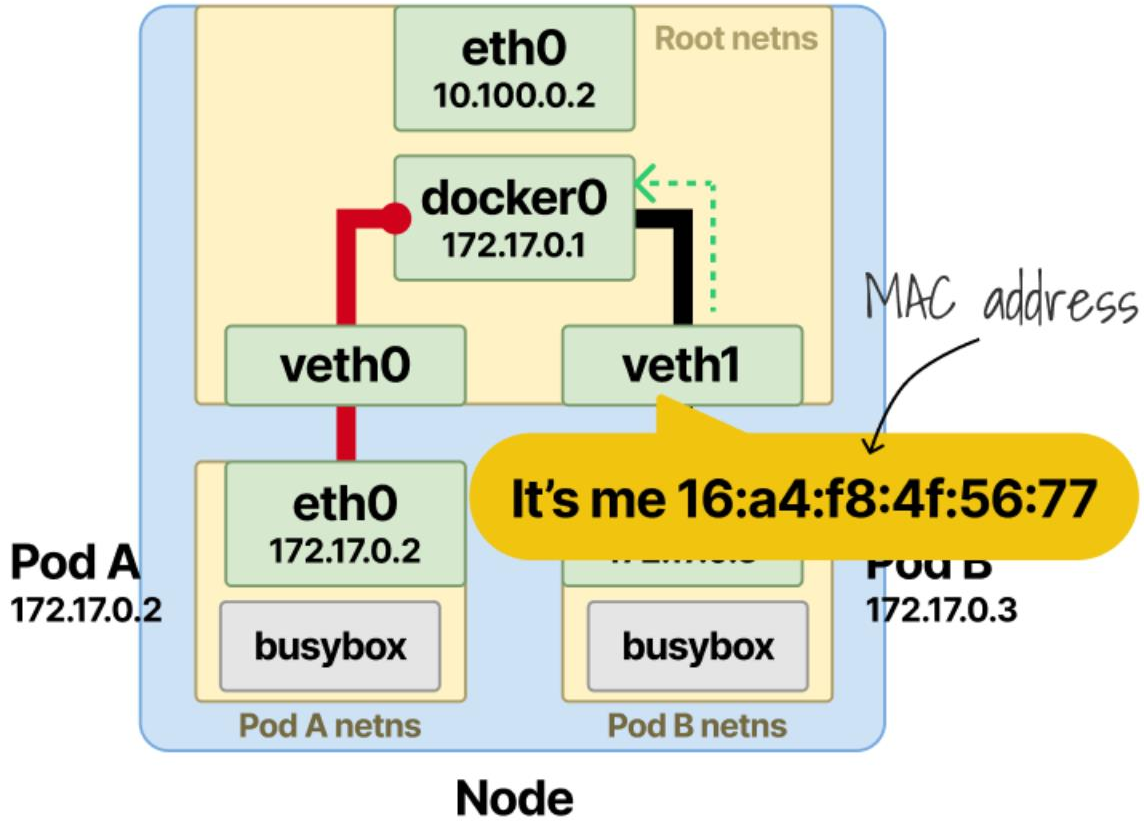
> *So what happens when Pod-A wants to send a message to Pod-B?*
> Since the destination isn't one of the containers in the namespace Pod-A sends out a packet to its default interface eth0.
> This interface is tied to the veth pair and **packets are forwarded to the root namespace.**



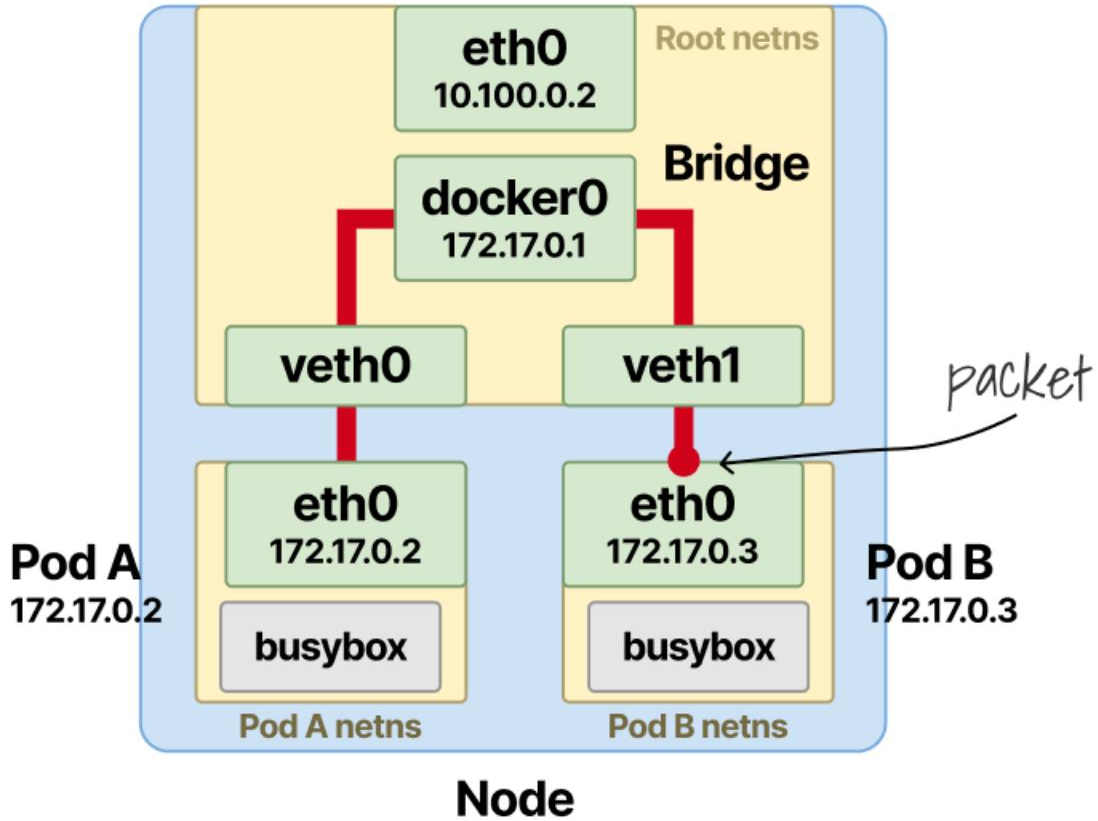
> The ethernet bridge, acting as a virtual switch, has to somehow resolve the destination pod IP (Pod-B) to its MAC address.



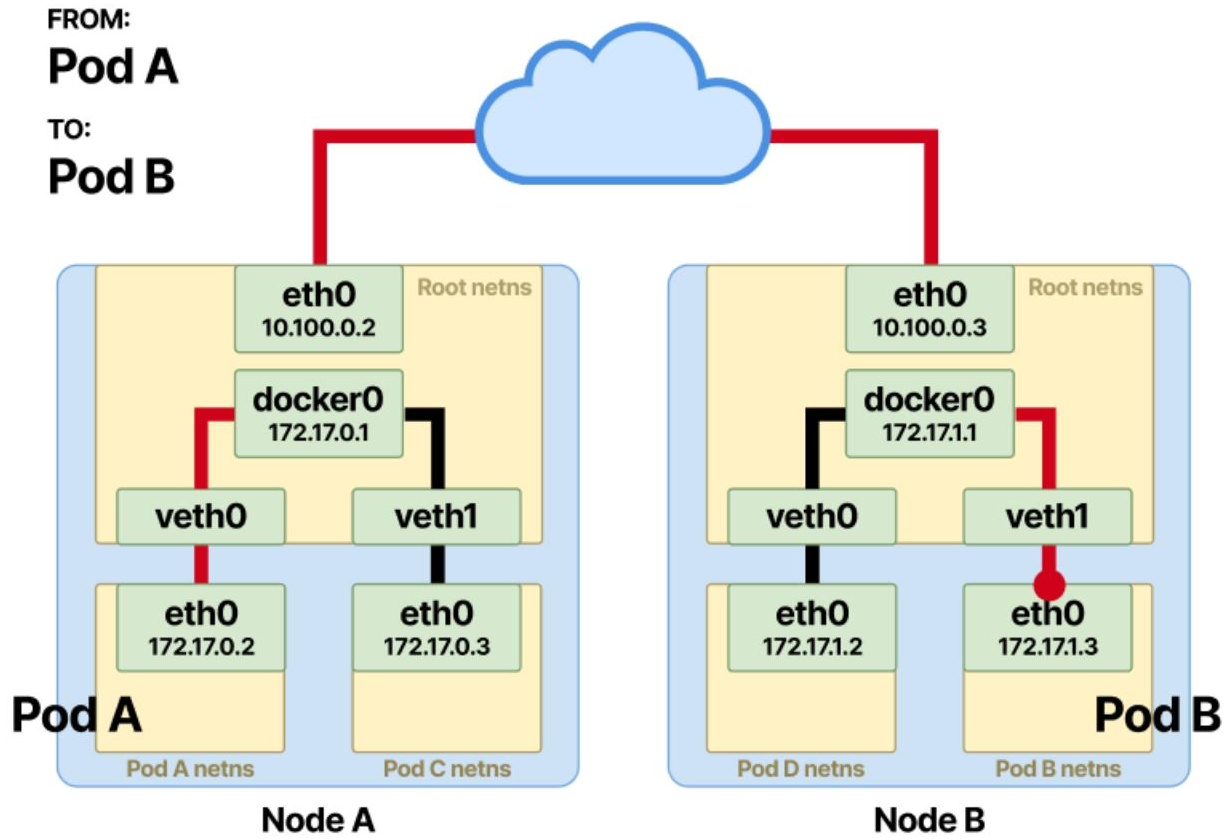
> The ARP protocol comes to the rescue.
> An ARP broadcast is sent on all connected devices when the frame reaches the bridge.
> The bridge shouts "Who has Pod-B IP address?"



A reply is received with the MAC address of the interface that connects Pod-B, then this information is stored in the bridge ARP cache (lookup table).



> Once the IP and MAC address mapping is stored, the bridge looks up in the table and forwards the packet to the correct endpoint. The packet reaches Pod-B veth in the root namespace, and from there, it quickly reaches the eth0 interface inside the Pod-B namespace.



> With this, the communication between Pod-A and Pod-B has been successful.
> For pods that need to communicate across different nodes, an additional hop is required.
> If you are interested in learning more, check out this article: https://learnk8s.io/kubernetes-network-packets

That's it! Thanks for reading this far!

I post deep-dives like this on a regular basis. Follow me if you don't want to miss the next story!