

Kubernetes Essential

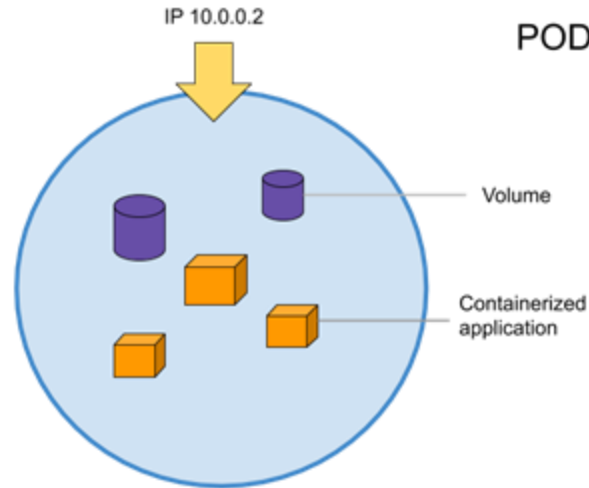


Agenda

- Assignment Review & Guides
- Kubernetes network model
- Network Topology
- Pod networking
- Container Network Interface
- CoreDNS

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

A *Pod* is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

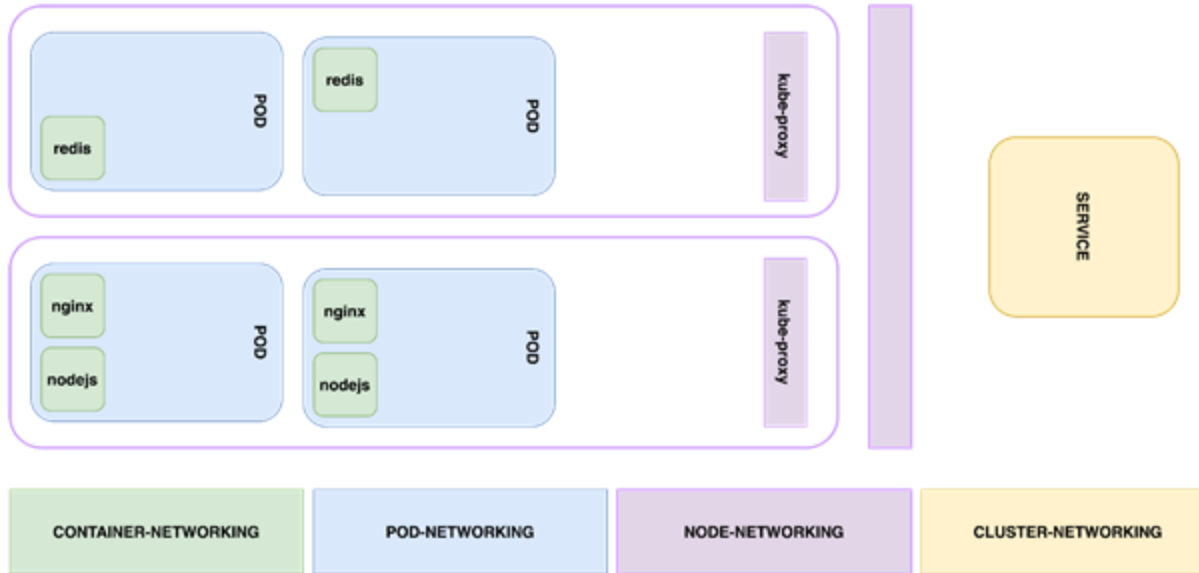


Kubernetes Networking Model

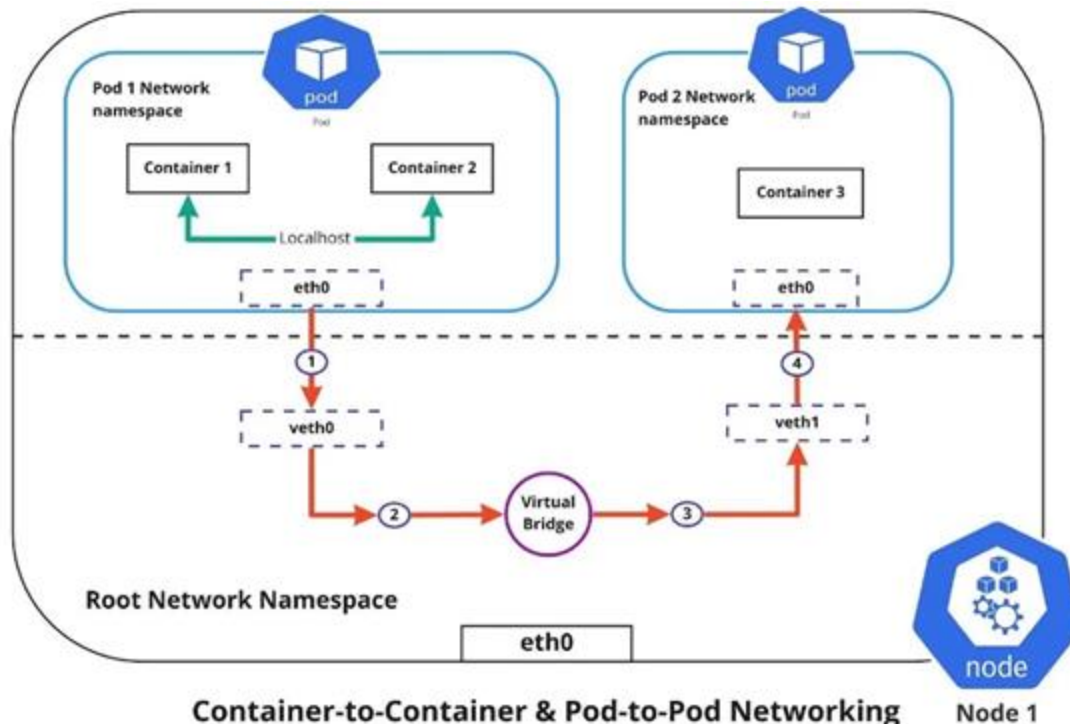
- Every Pod gets its own unique cluster-wide IP address: There should be no need to create links between Pods and no need to map container ports to host ports.
- Pods on a node should be able to communicate with all Pods on all nodes without NAT.
- Agents get all-access passes: Agents on a node (system daemons, Kubelet) can communicate with all the Pods in that node.



Network Topology



Networking cluster

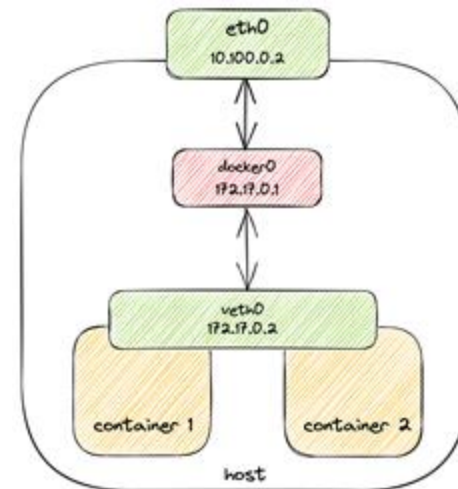


Container-to-Container & Pod-to-Pod Networking



Container-to-container networking

- Both containers will have the same IP address of 172.17.0.2, so the applications inside it can listen to the port on the same IP address and can connect to each other through localhost.
- This also means that two containers cannot listen to the same port, which is a limitation but it is no different from running multiple processes on the same host machine.



Commands

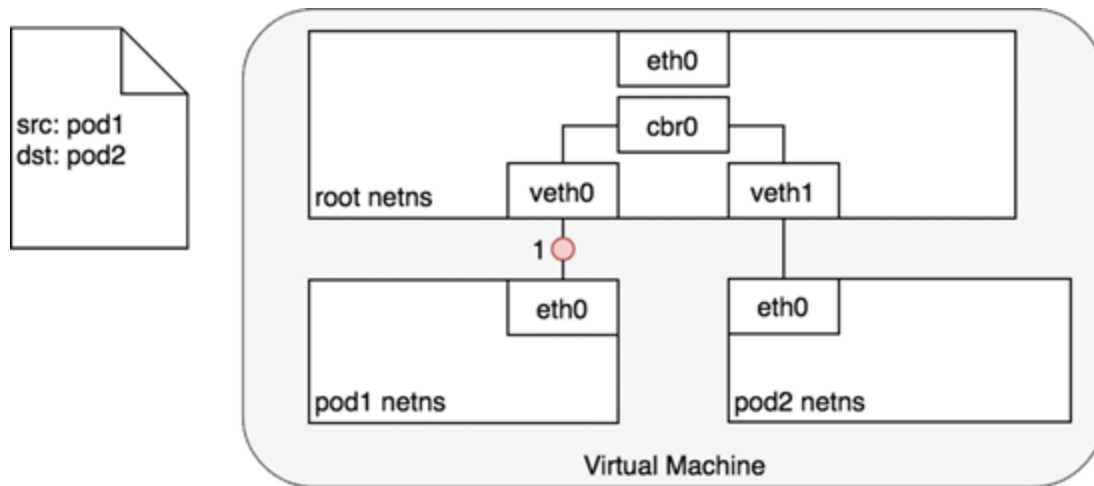
get	Get a list of objects (pod, service, deployments, ...)
describe	Inspect detailed information of an object
create	Create an object
logs	Get logs of a pod
exec	Execute command from a pod
apply	Apply object configurations (-f with file)

Arguments

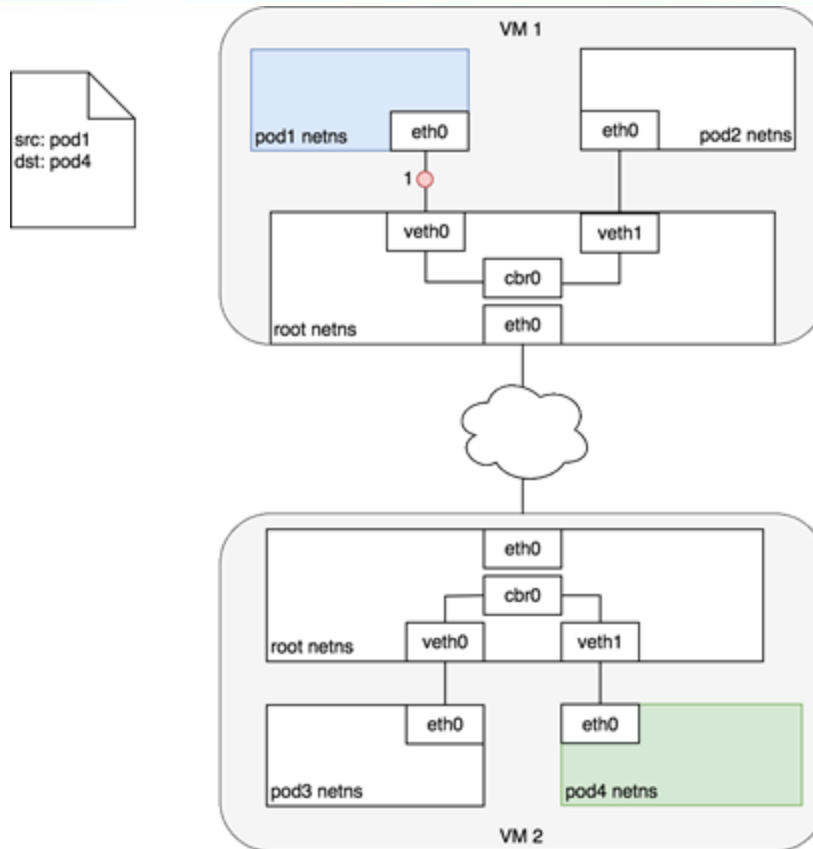
-n	Specify namespace (-A for all namespaces)
-o	Specify output format (YAML, wide, ...)
--dry-run=client	Does not apply changes

Pod Networking (Pod-to-Pod)

- There is a designated CIDR range of IPs for Pods to ensure that every Pod has a unique IP address that other Pods in the cluster can see.
- Pod-to-Pod communication happens using real IPs



Pod Networking (Pod-to-Pod across Nodes)



Issues:

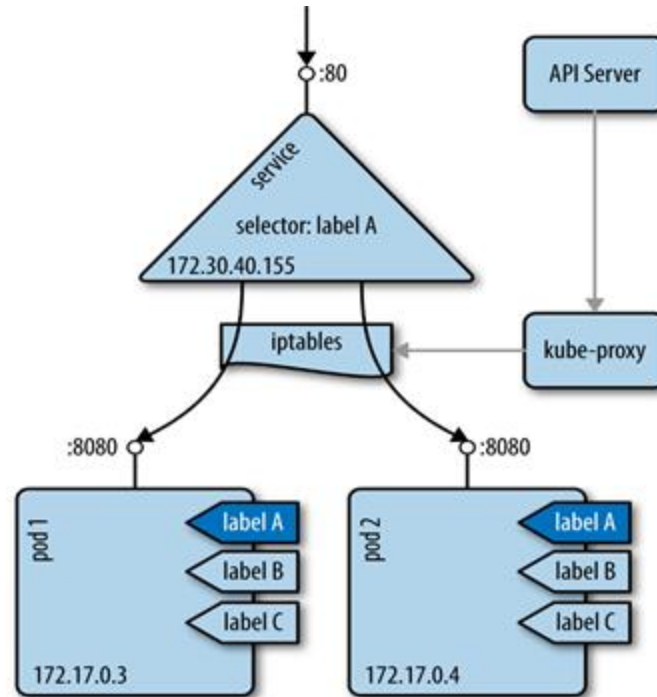
- Pod may need to scale up or down based on demand
- Pod may be created again in case of an application crash or a node failure
- These events cause a Pod's IP address to change

Kubernetes solves this problem by using the Service function, which does the following:

- Assigns a static virtual IP address in the frontend to connect any backend Pods associated with the Service.
- Load-balances any traffic addressed to this virtual IP to the set of backend Pods.
- Keeps track of the IP address of a Pod, such that even if the Pod IP address changes



Pod Networking (Pod-to-Service)



Kubernetes can optionally use DNS to avoid having to hard-code a Service's cluster IP address into your application

