Lesson 3: Kubernetes Pod and Service Networks

Interconnect applications pods inside the same cluster by using Kubernetes services.

The Software-defined Network

- Network Infrastructure management
- Virtual network - abstracting several networking layers
- Not accessible from outside of cluster
- Does not regulate processes on cluster nodes
- Port allocation, IP address leasing and reservation, name resolution, service discovery, load balancing
- Application components can communicate with each other

Shared Network

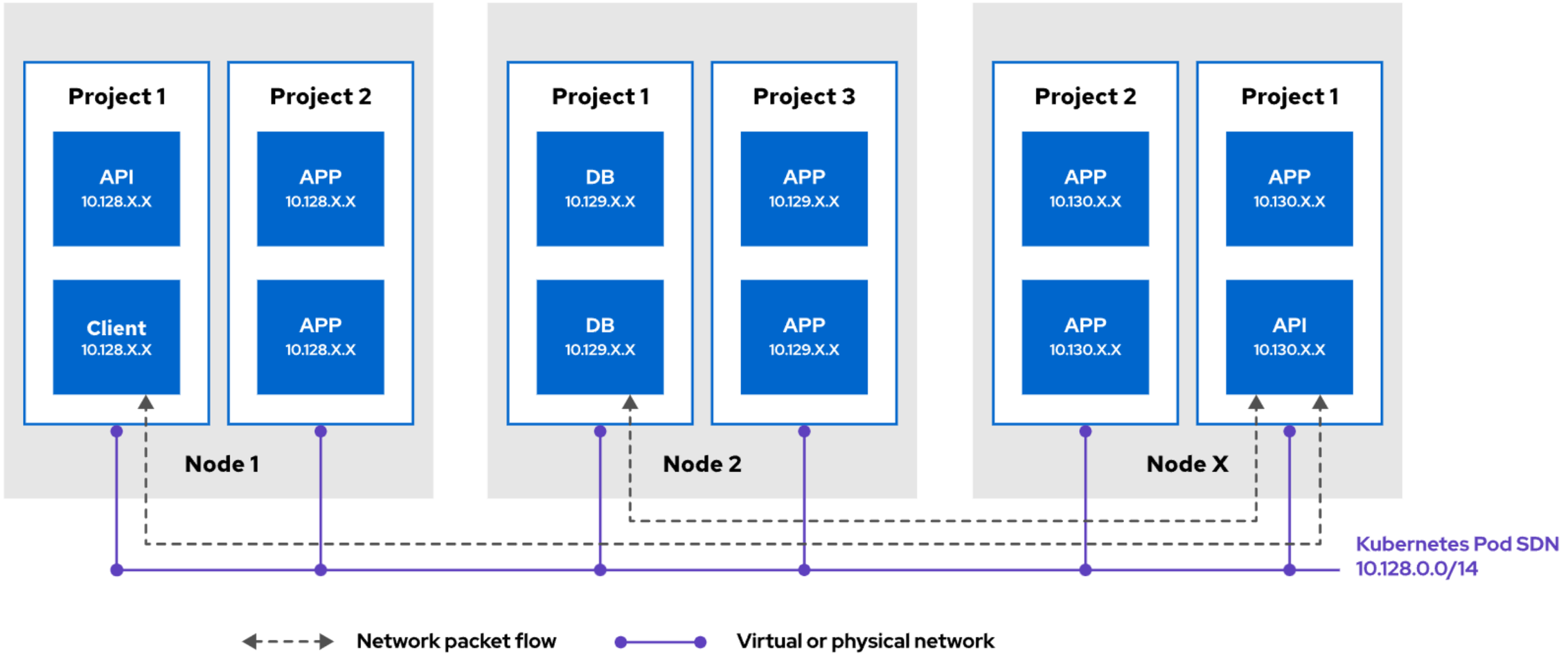
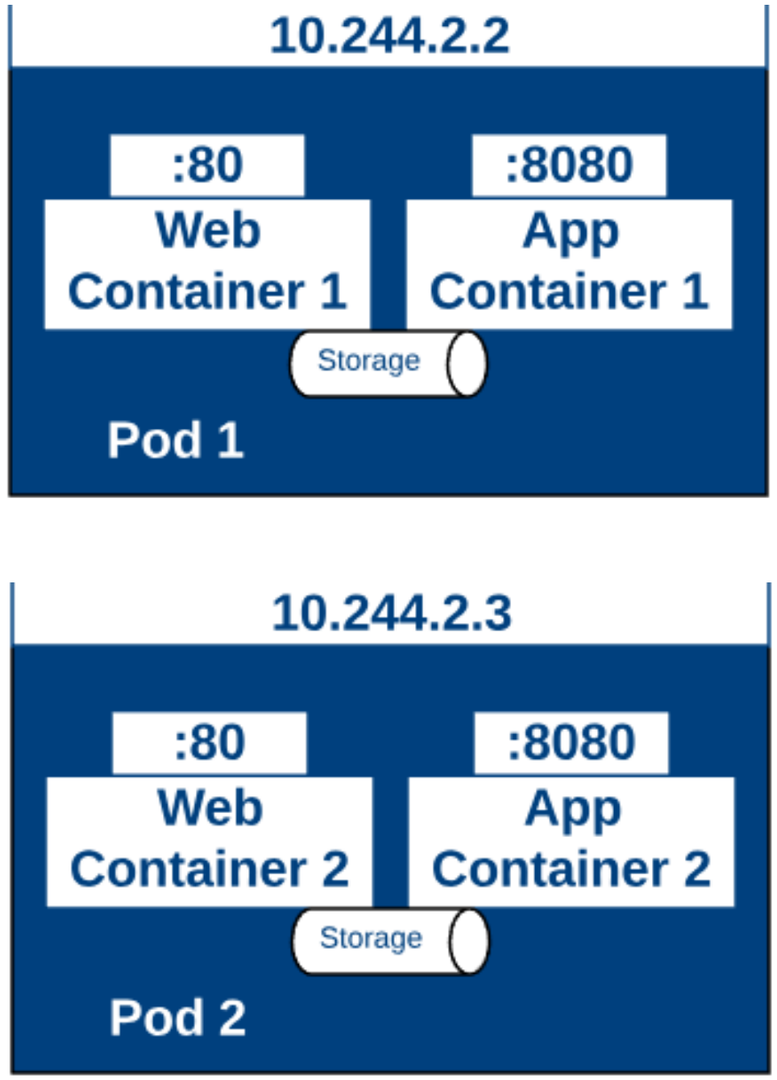


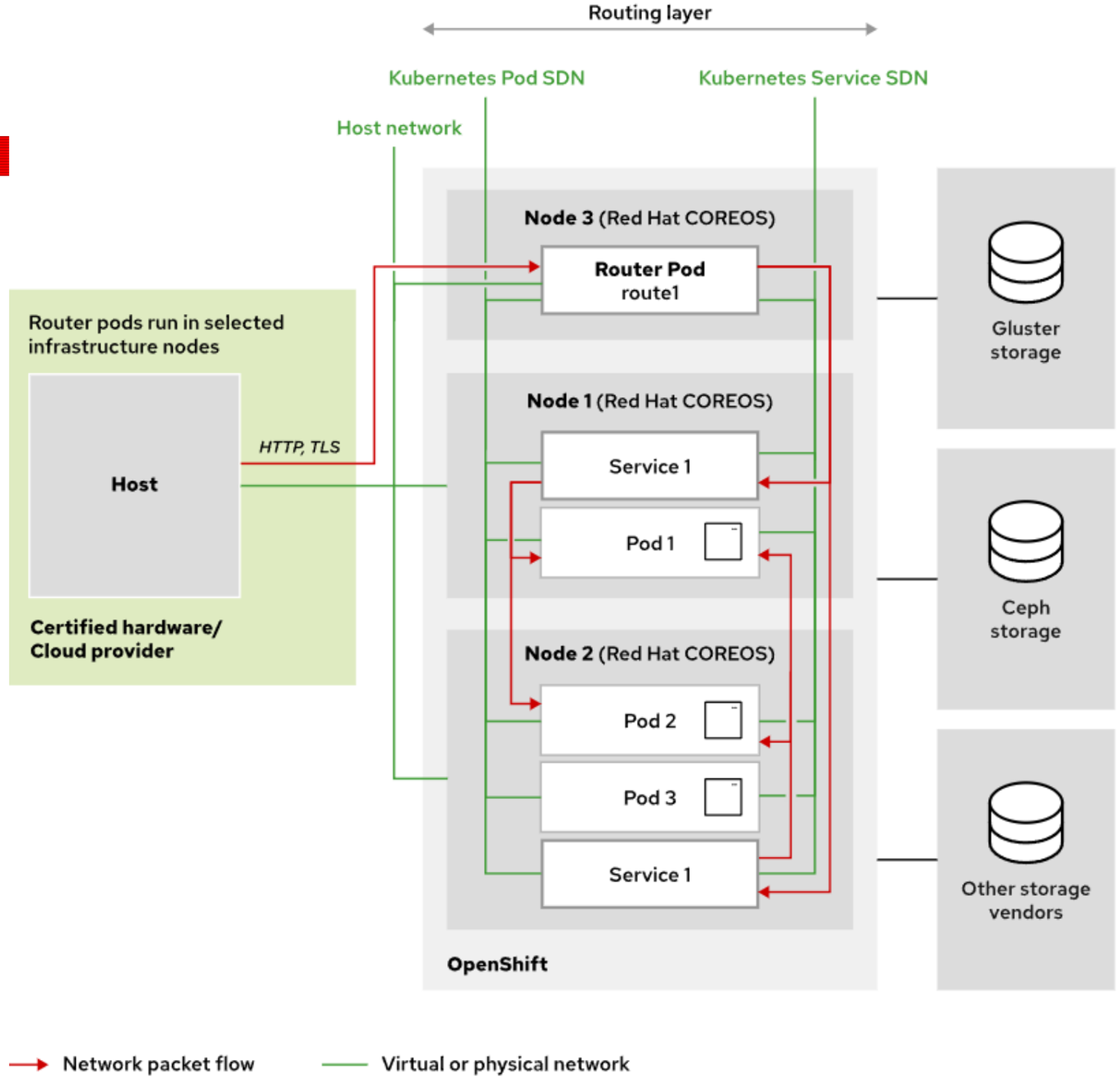
Figure 4.5: How the Kubernetes SDN manages the network

Kubernetes Networking

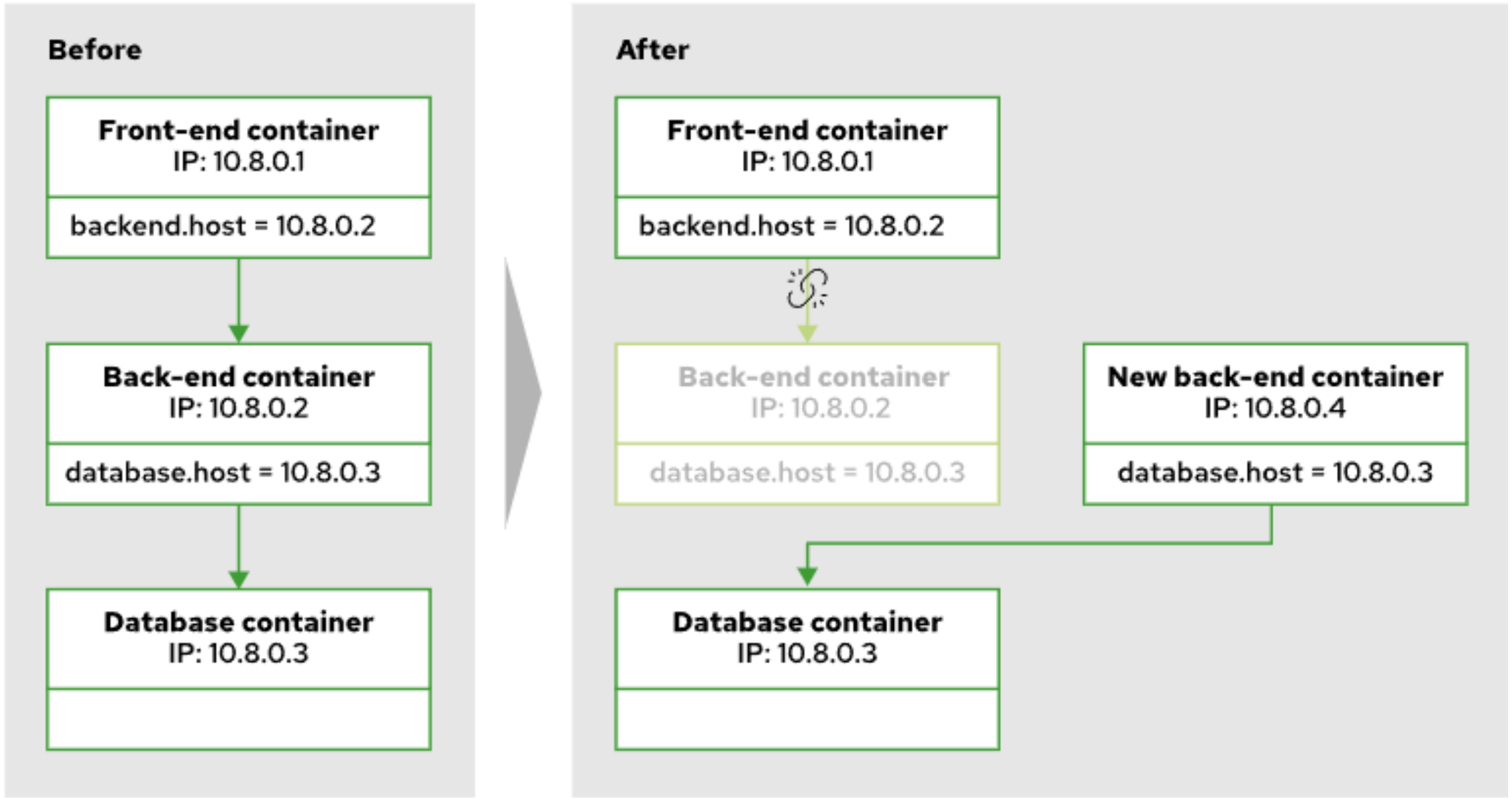
- Is scalable
- Kubernetes networking provides following capabilities:
  a) Highly coupled container-to-container communications
  b) Pod-to-pod communications
  c) Pod-to-service communications
  d) External-to-service communication
- IP addresses are assigned automatically, but unstable
- All container share networking resources
- No need for NAT



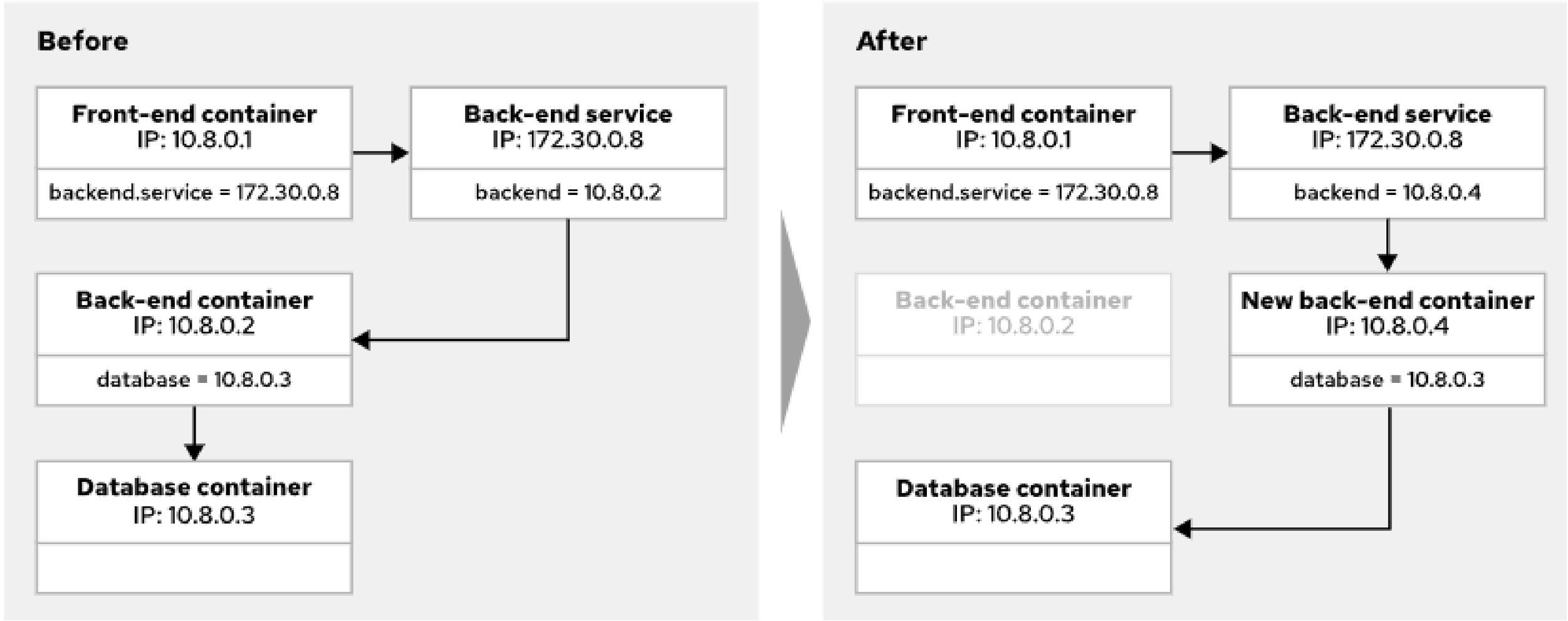
Network Access between pods in cluster



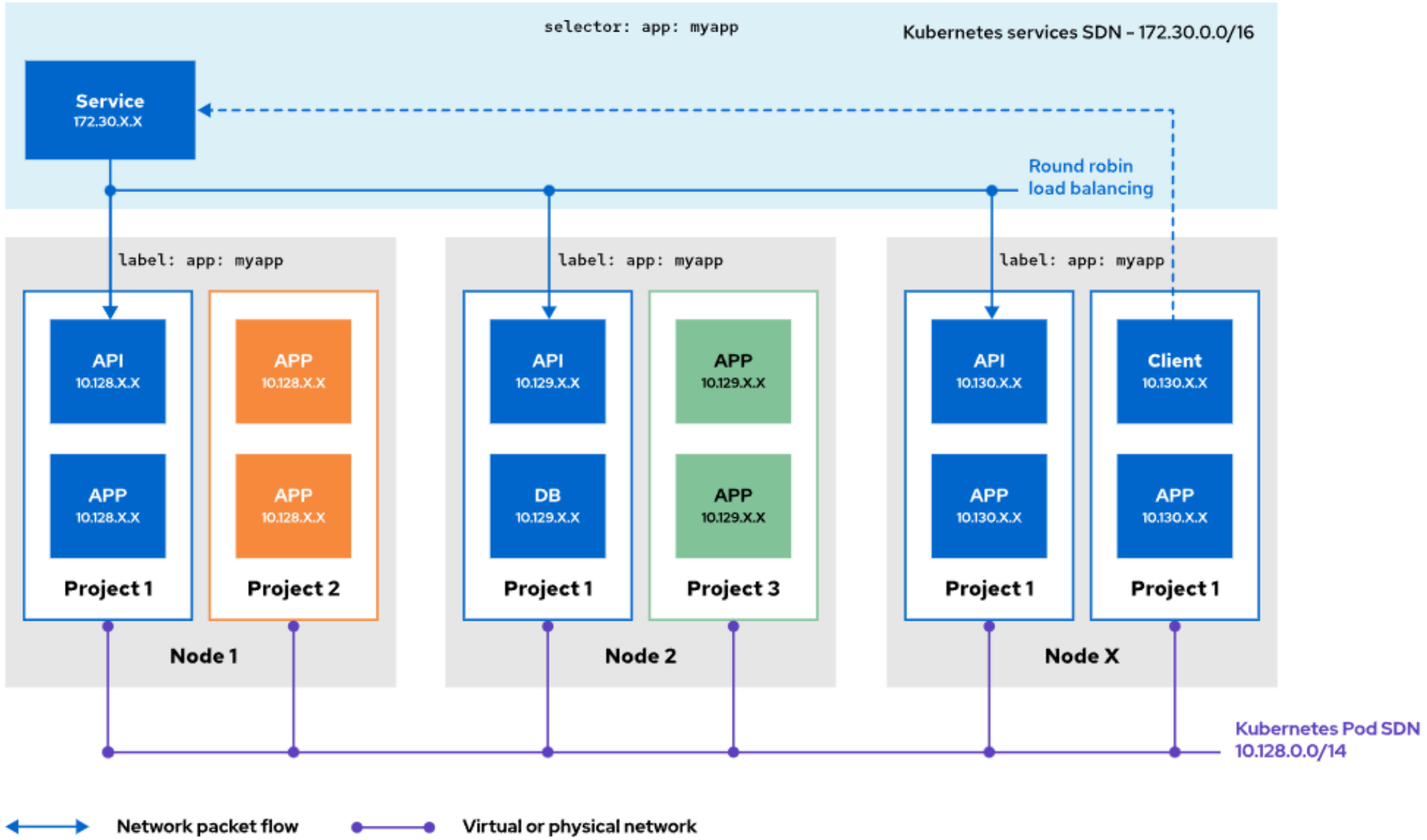
Problem with direct access to pods



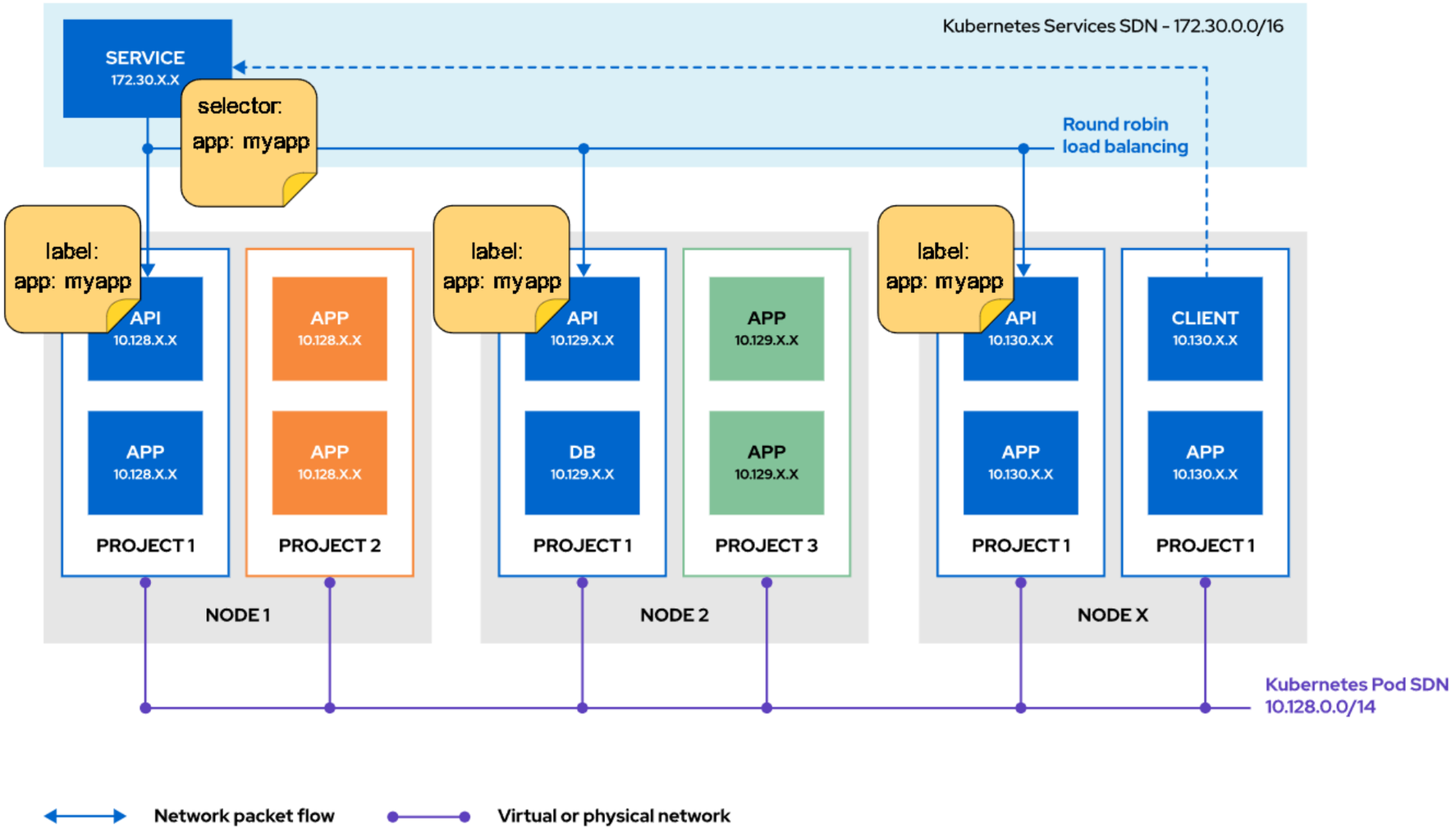
Services resolve pod failure issues



Service with multiple pods across nodes



Service selector match to pod labels



Verify created service

To view the selector that a service uses, use the `-o wide` option with the `oc get` command.

```
[user@host ~]$ oc get service db-pod -o wide
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP PORT(S)     AGE     SELECTOR
db-pod  ClusterIP   172.30.108.92   <none>      3306/TCP    108s    app=db-pod
```

In this example, `db-pod` is the name of the service. Pods must use the `app=db-pod` label to be included in the host list for the `db-pod` service. To see the endpoints that a service uses, use the `oc get endpoints` command.

```
[user@host ~]$ oc get endpoints
NAME     ENDPOINTS                         AGE
db-pod   10.8.0.86:3306,10.8.0.88:3306   27s
```

```
[user@host ~]$ oc describe deployment db-pod
Name:                   db-pod
Namespace:              deploy-services
CreationTimestamp:      Wed, 18 Jan 2023 17:46:03 -0500
Labels:                 app=db-pod
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=db-pod
...output omitted...
```

Kubernetes DNS for Service Discovery

- DNS operator
  a) creates default cluster DNS
  b) assign FQDN to services
  c) provides name resolution
  d) Implements DNS API from operator.openshift.io API group
- Assign FQDN to services using following format:
  - SVC-NAME.PROJECT-NAME.svc.CLUSTER-DOMAIN
- Example
  - db-pod.deploy-services.svc.cluster.local
- PODs are configured to point to the DNS

```
[user@host ~]$ cat /etc/resolv.conf
search deploy-services.svc.cluster.local svc.cluster.local ...
nameserver 172.30.0.10
options ndots:5
```

Kubernetes Networking Drivers

- Container Network Interface (CNI) plug-ins
  a) a framework for dynamically configuring networking resources.
  b) configuring the network, provisioning IP addresses, and maintaining connectivity with multiple hosts.
- RHOCP clusters offer following CNI plug-ins:
  a) OVN-Kubernetes: default in RHOCP v4.10
  b) OpenShift SDN: legacy in RHOCP v3.x
  c) Kuryr: Integrate and performance on OpenStack
- Other supported CNI plug-ins:
  - Flannel, Calico, WeaveNet, Cilium, Canal, and Multus

The OpenShift Cluster Network Operator (CNO)

- Configures OpenShift cluster networking
- Loads and configure CNI plug-ins
- Observe status of the CNO

```
[user@host ~]$ oc get -n openshift-network-operator deployment/network-operator
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
network-operator   1/1     1            1           41d
```

- Cluster admin can only modify CNO during installation

```
[user@host ~]$ oc describe network.config/cluster
Name: cluster
...output omitted...
Spec:
Cluster Network:
Cidr: 10.8.0.0/14 (1)
Host Prefix: 23
External IP:
Policy:
Network Type: OVNKubernetes
Service Network:
172.30.0.0/16 (2)
...output omitted...
```

1. The Cluster Network CIDR defines the range of IPs for all pods in the cluster.
2. The Service Network CIDR defines the range of IPs for all services in the cluster.

Guided Exercise: Kubernetes Pod and Service Networks

You should be able to:

- Deploy a database server,
- Access it indirectly through a Kubernetes service
- Access it directly pod-to-pod for troubleshooting.