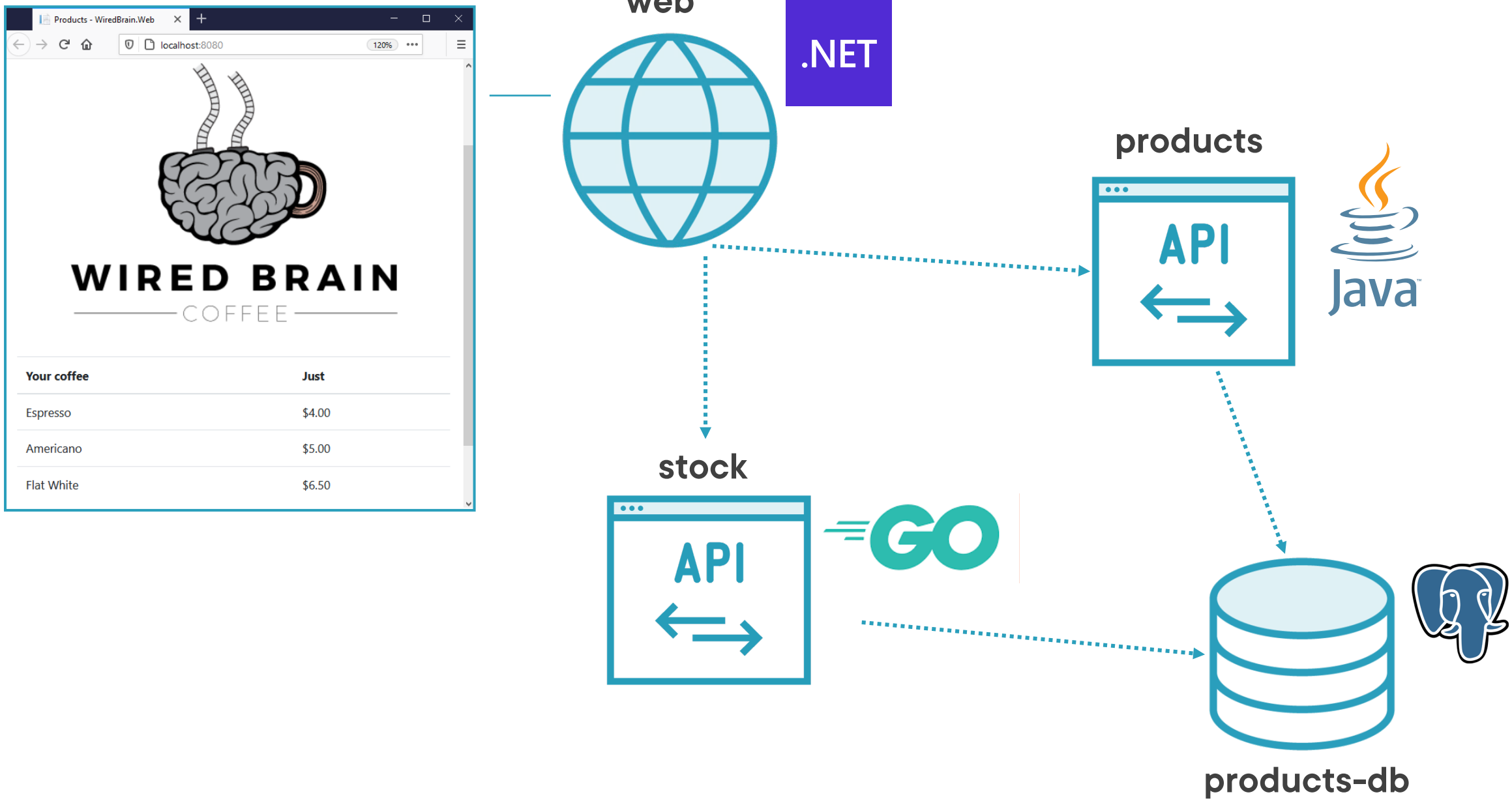
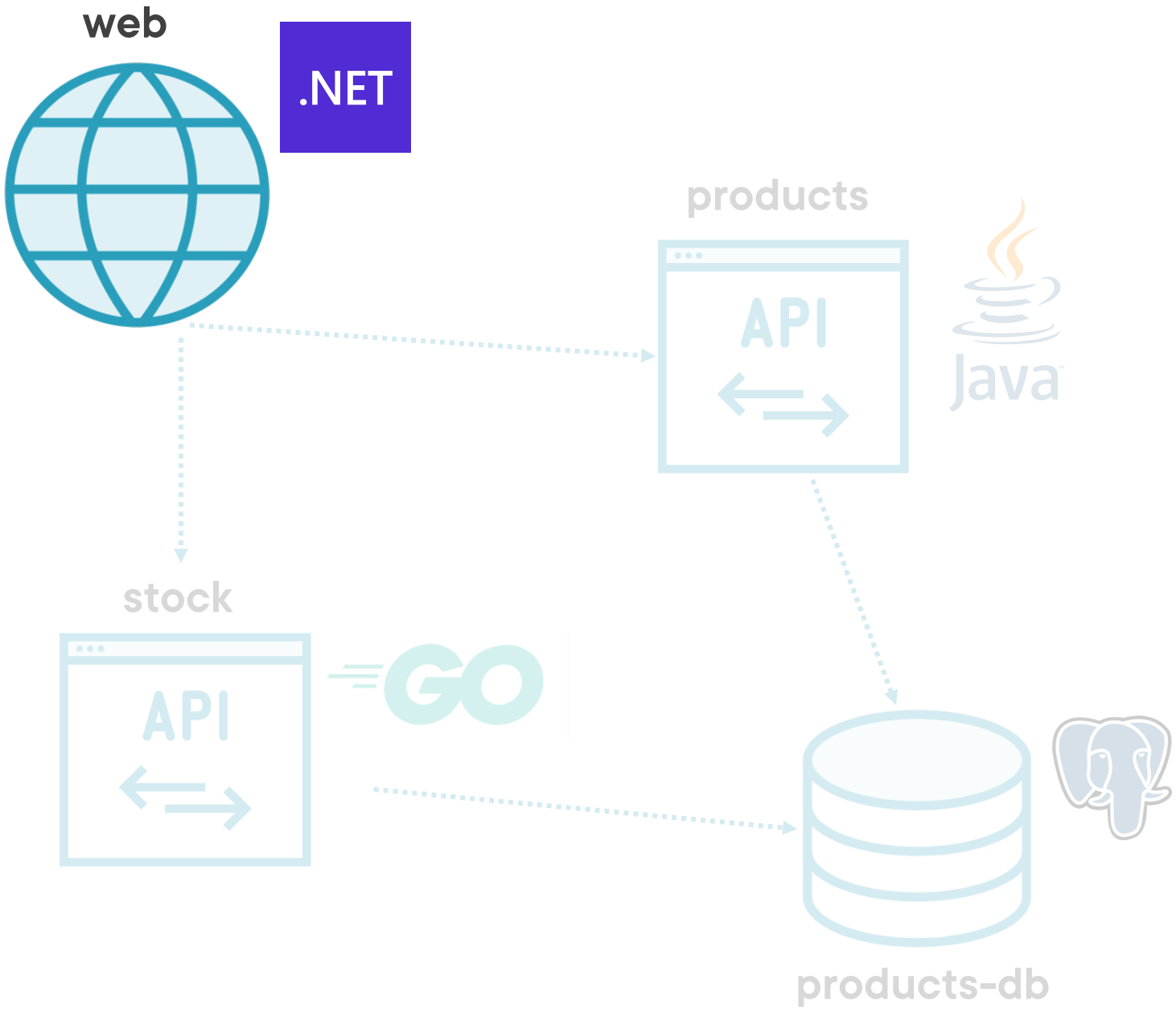
Persisting Data in Kubernetes

Elton Stoneman
Consultant & Trainer

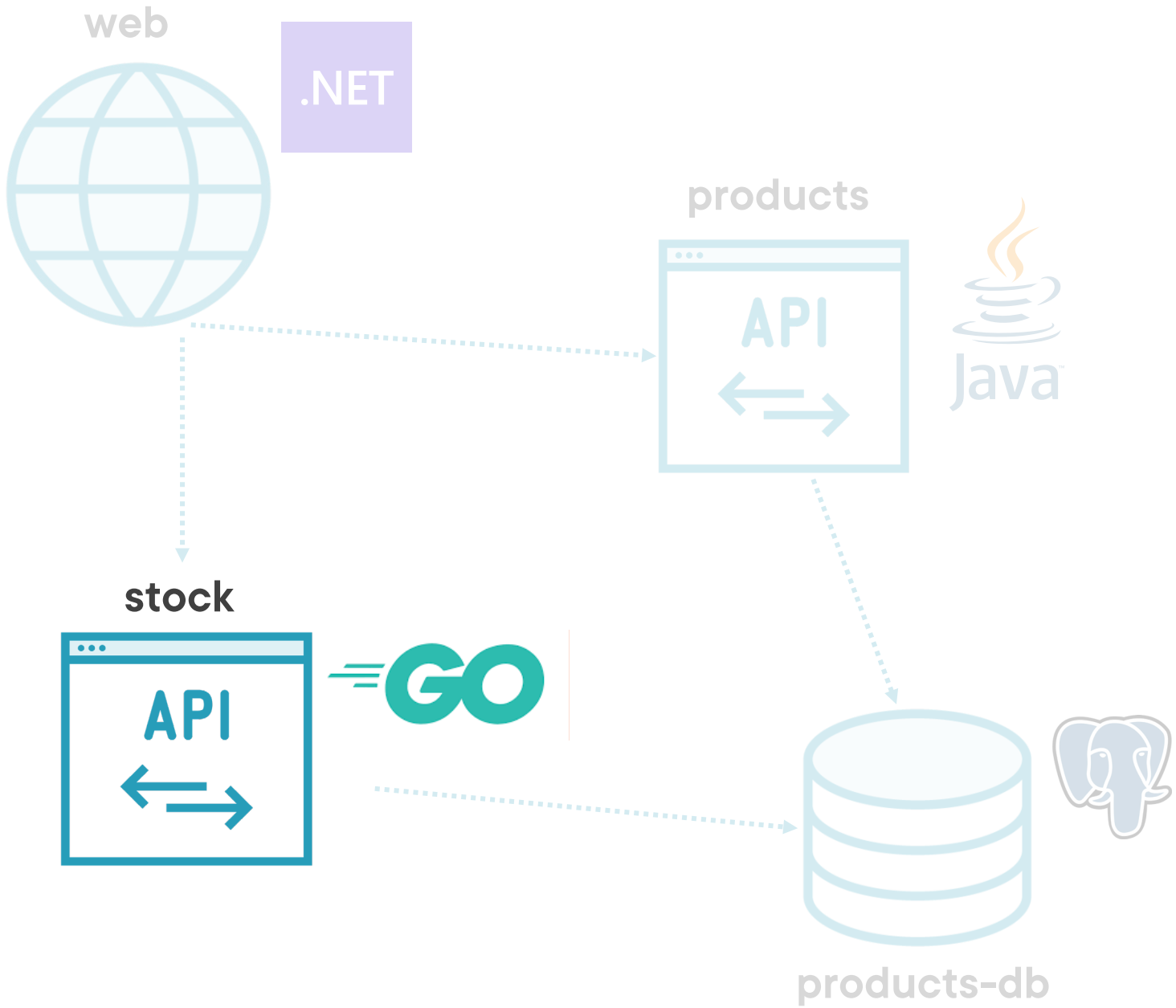
@EltonStoneman blog.sixeyed.com


Products - WiredBrain.Web
localhost:8080
120%
WIRED BRAIN
COFFEE
Your coffee
Just
Espresso
$4.00
Americano
$5.00
Flat White
$6.50
web
.NET
products
API
Java
stock
API
GO
products-db

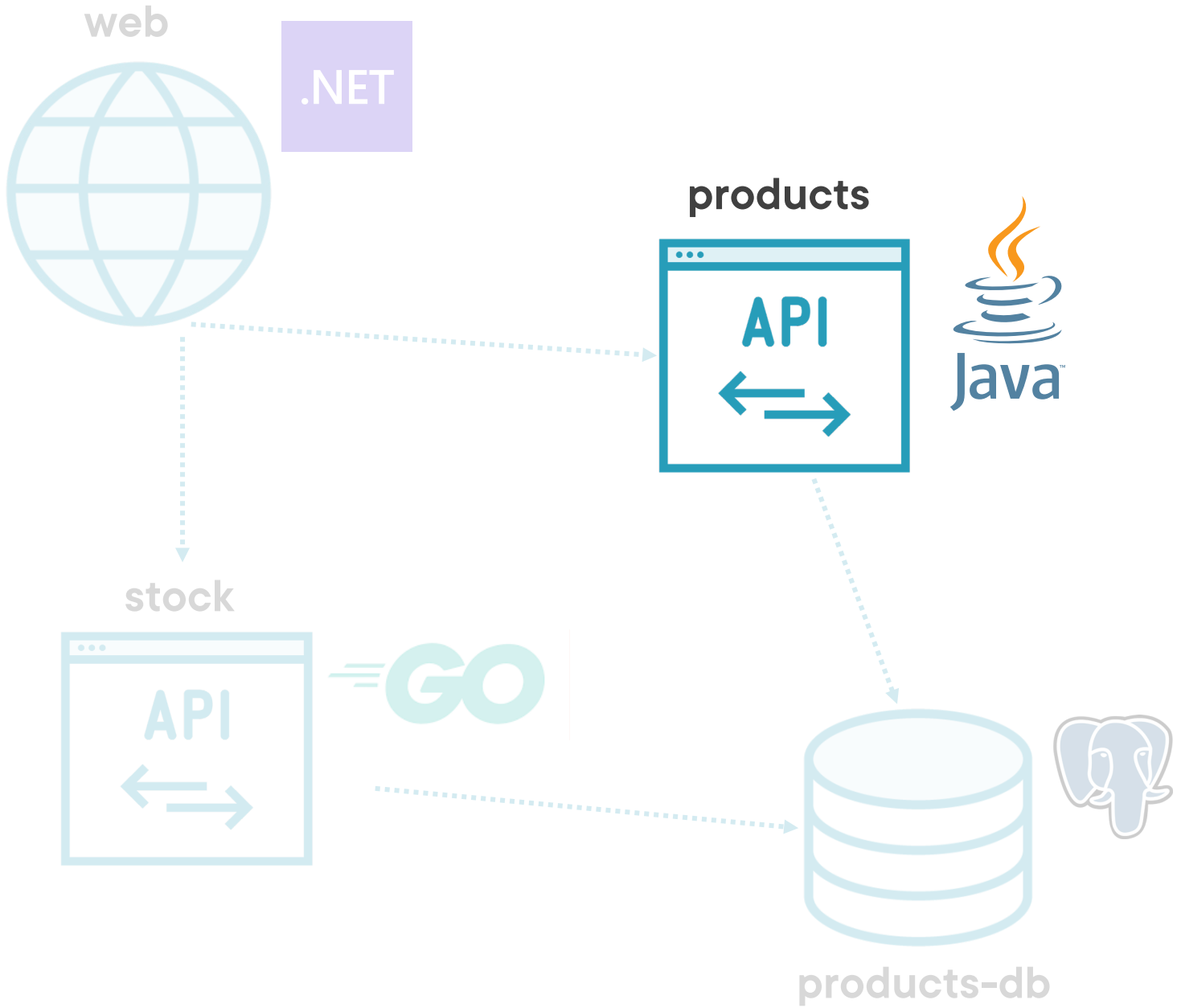
- Stateless
- No storage requirements



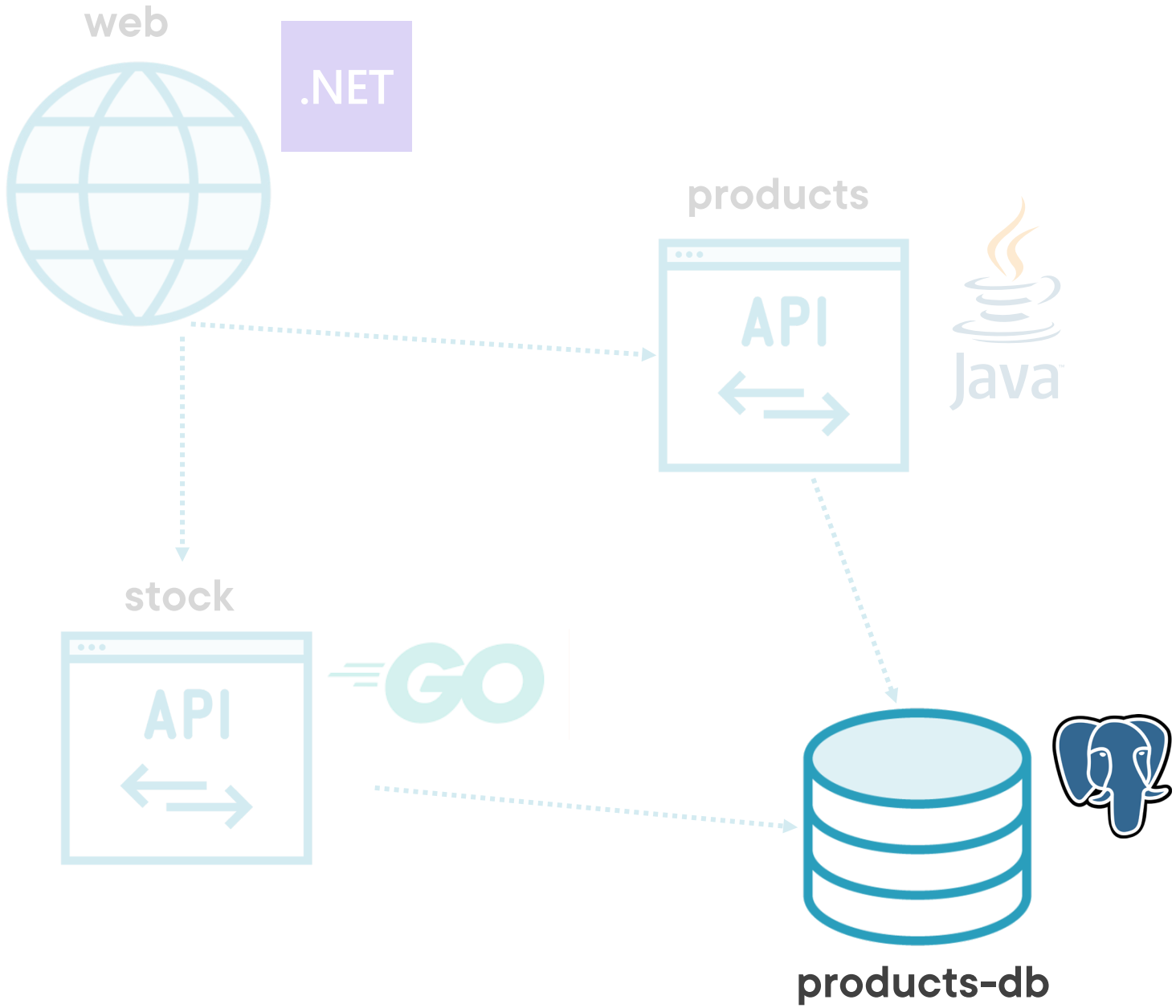
- Local data cache
- Performance boost
- Not persistent

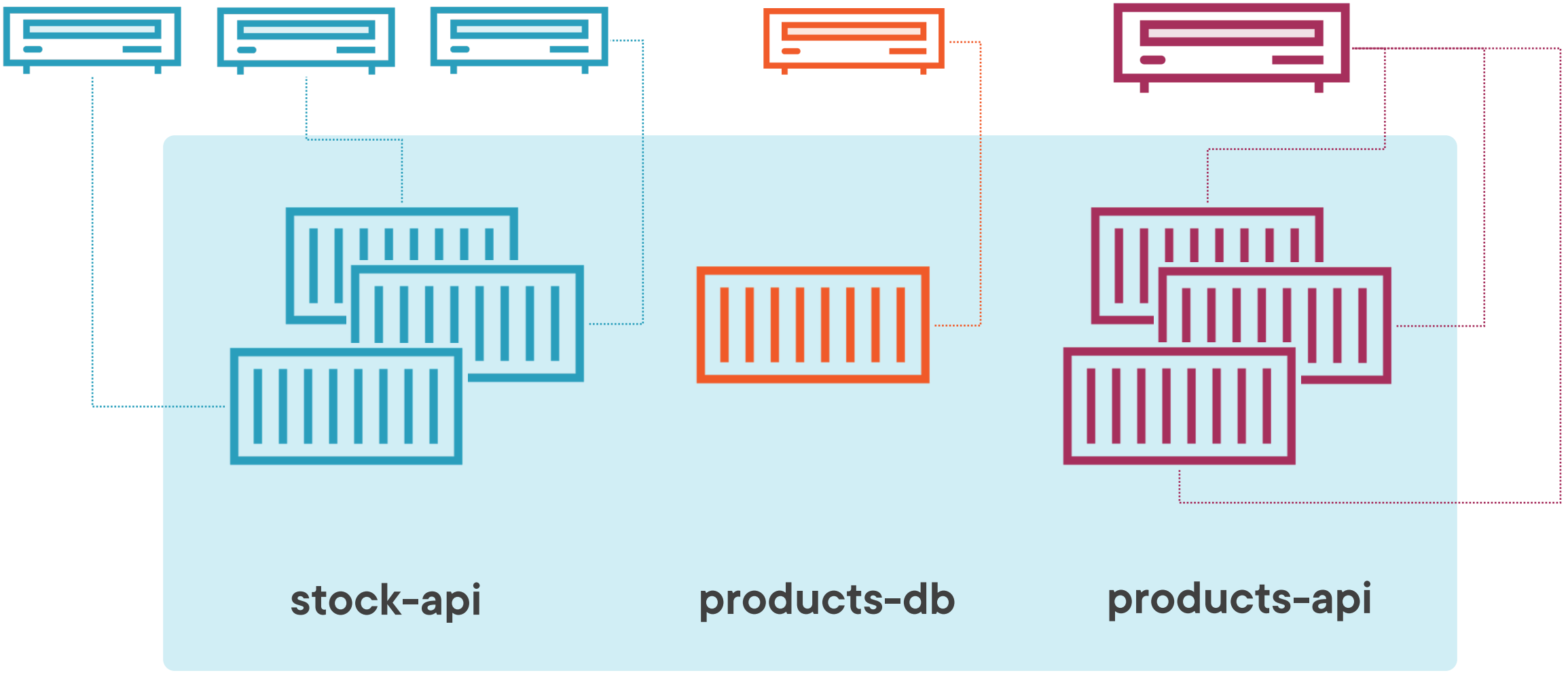


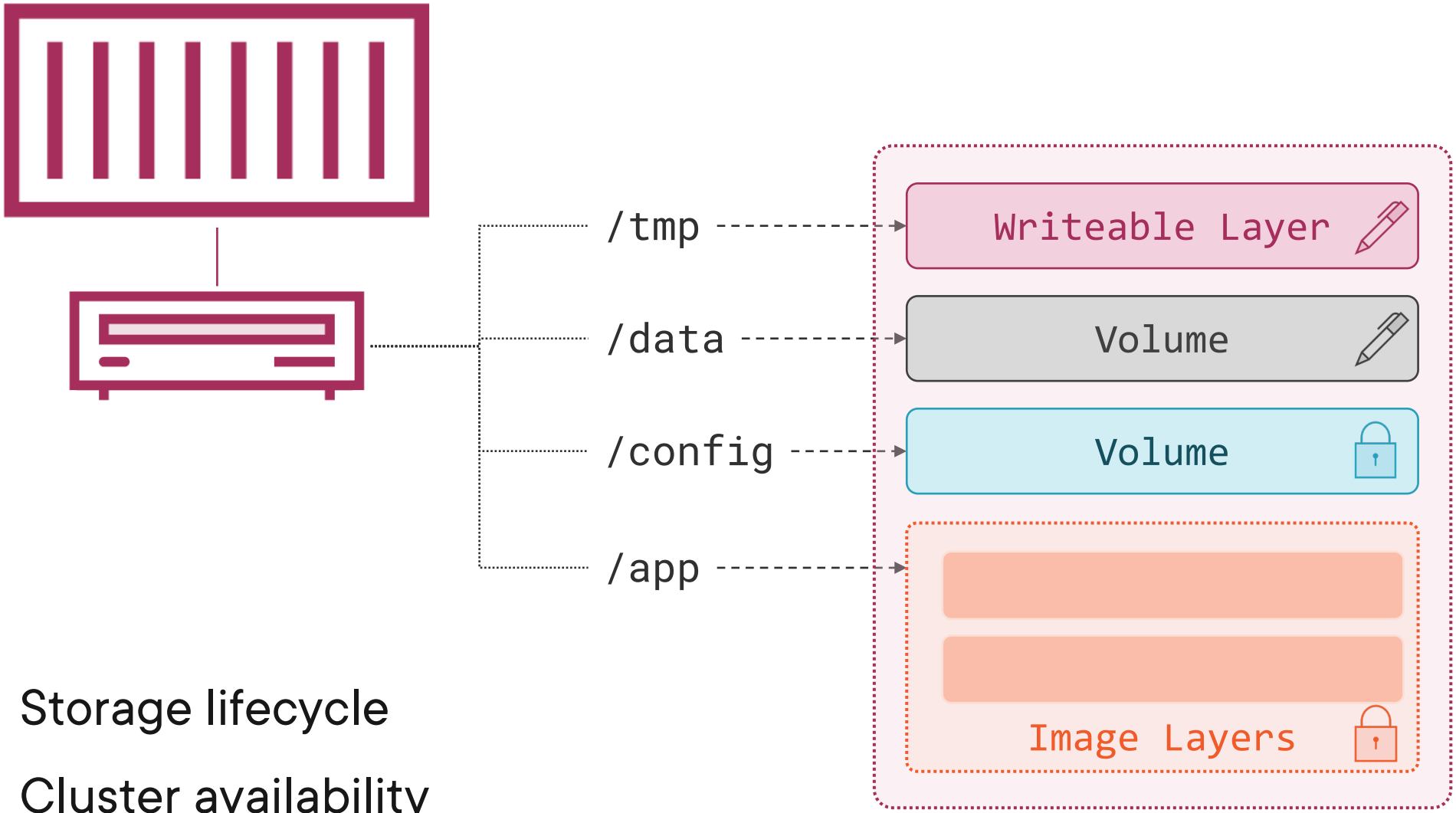
- Logs to file
- Central log location
- Persistent & shared



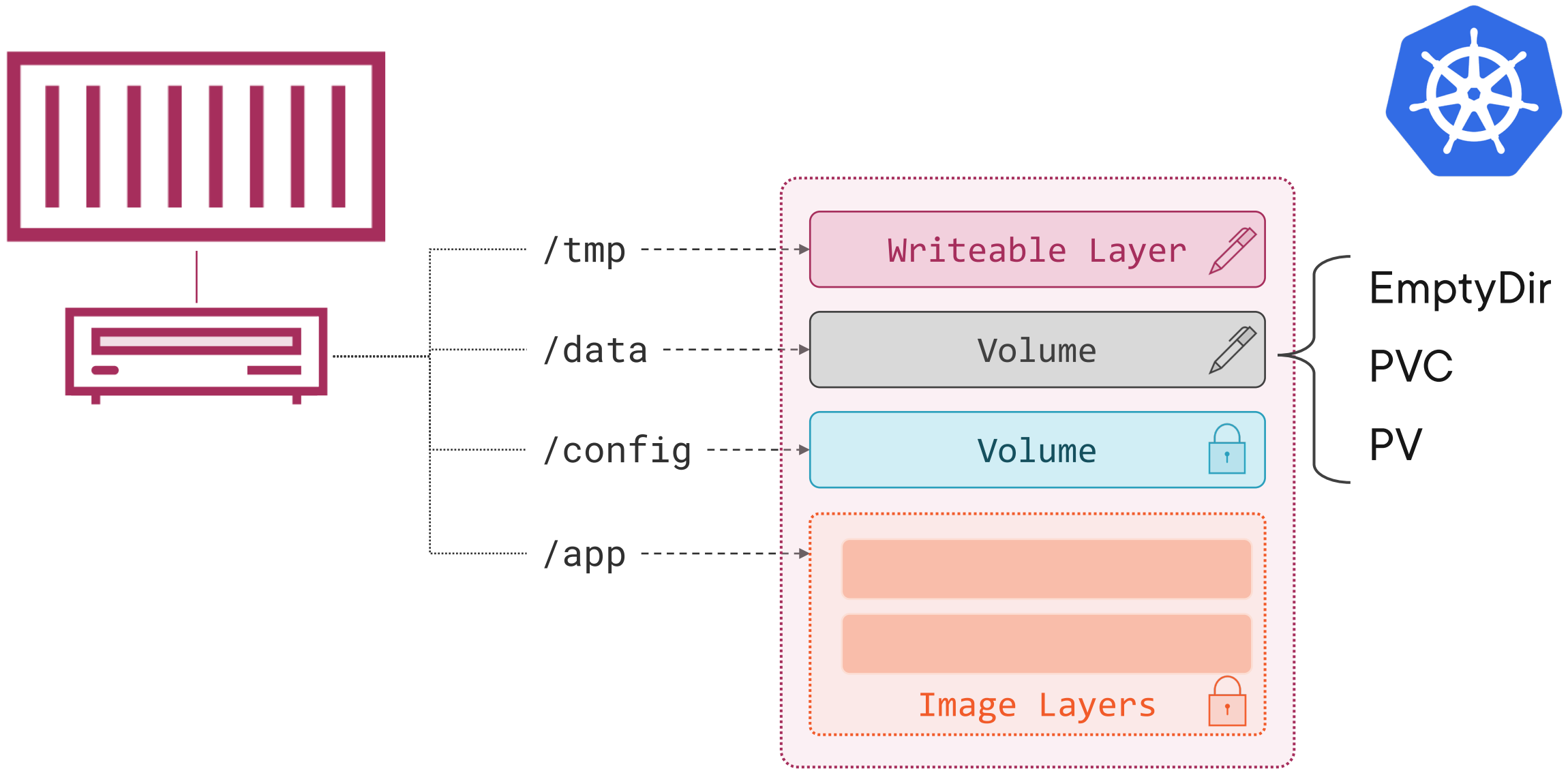
- Local data files
- HA through replication
- Persistent & isolated



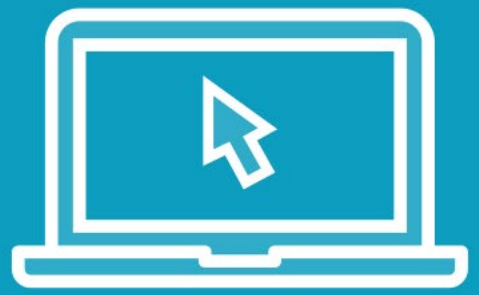

stock-api
products-db
products-api



- Storage lifecycle
- Cluster availability
- Performance & features

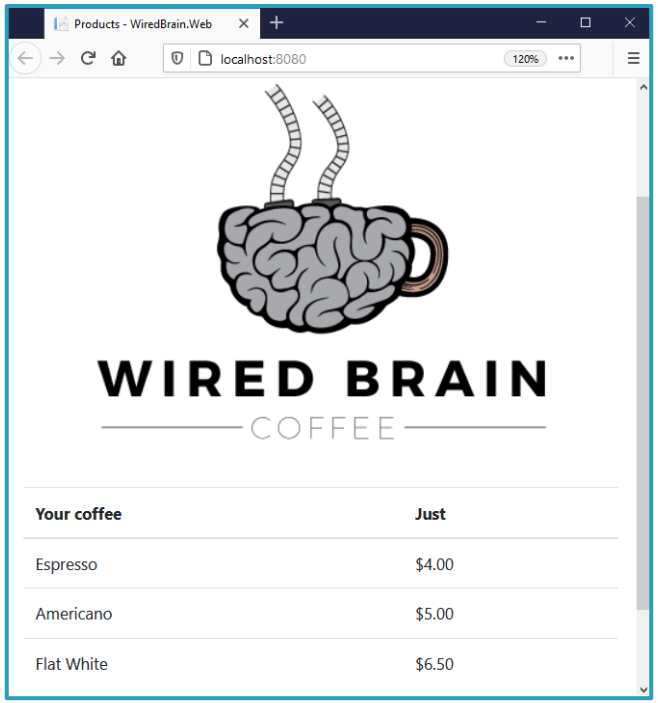

/tmp
/data
/config
/app
Writeable Layer
Volume
Volume
Image Layers
EmptyDir
PVC
PV

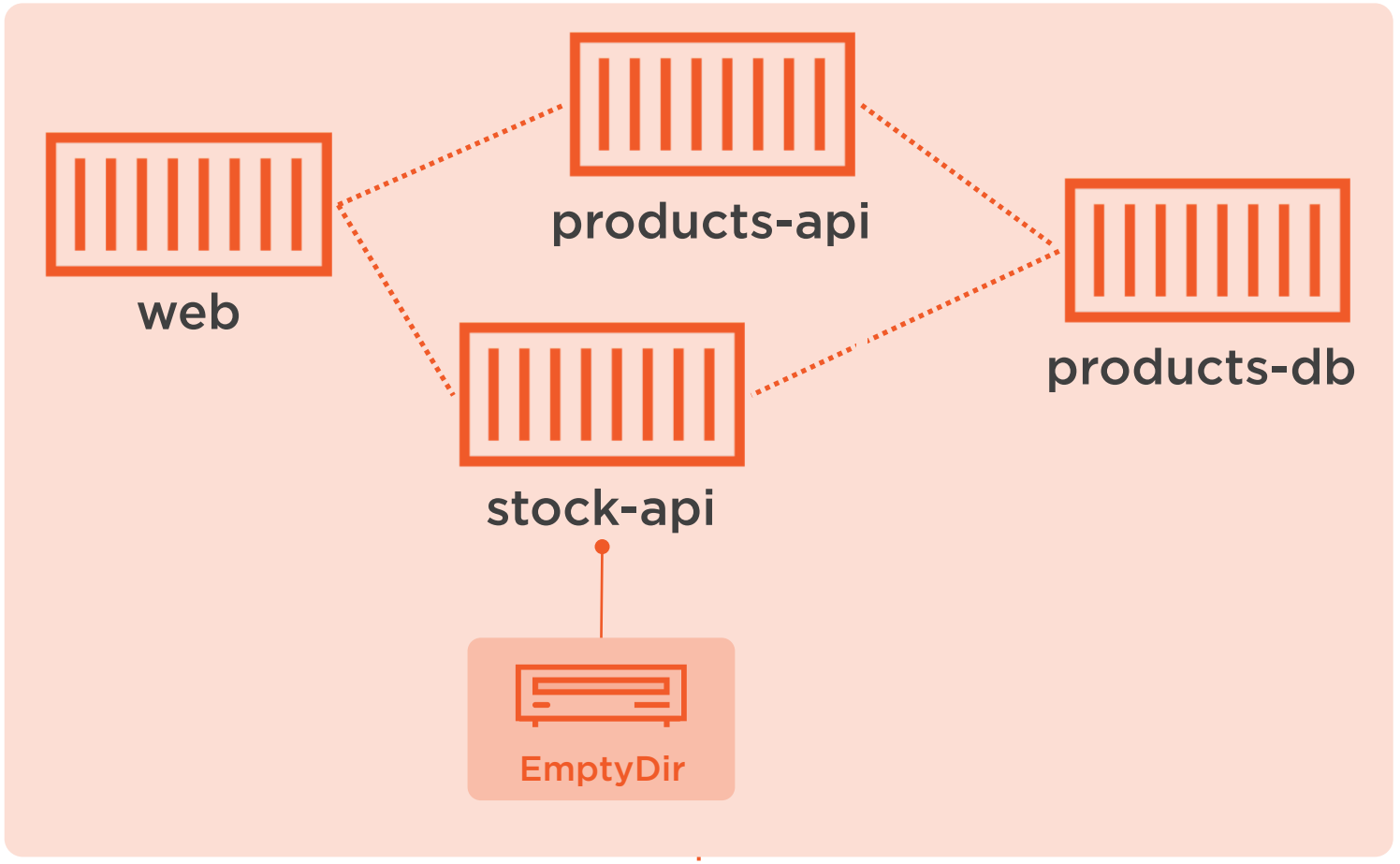
Demo

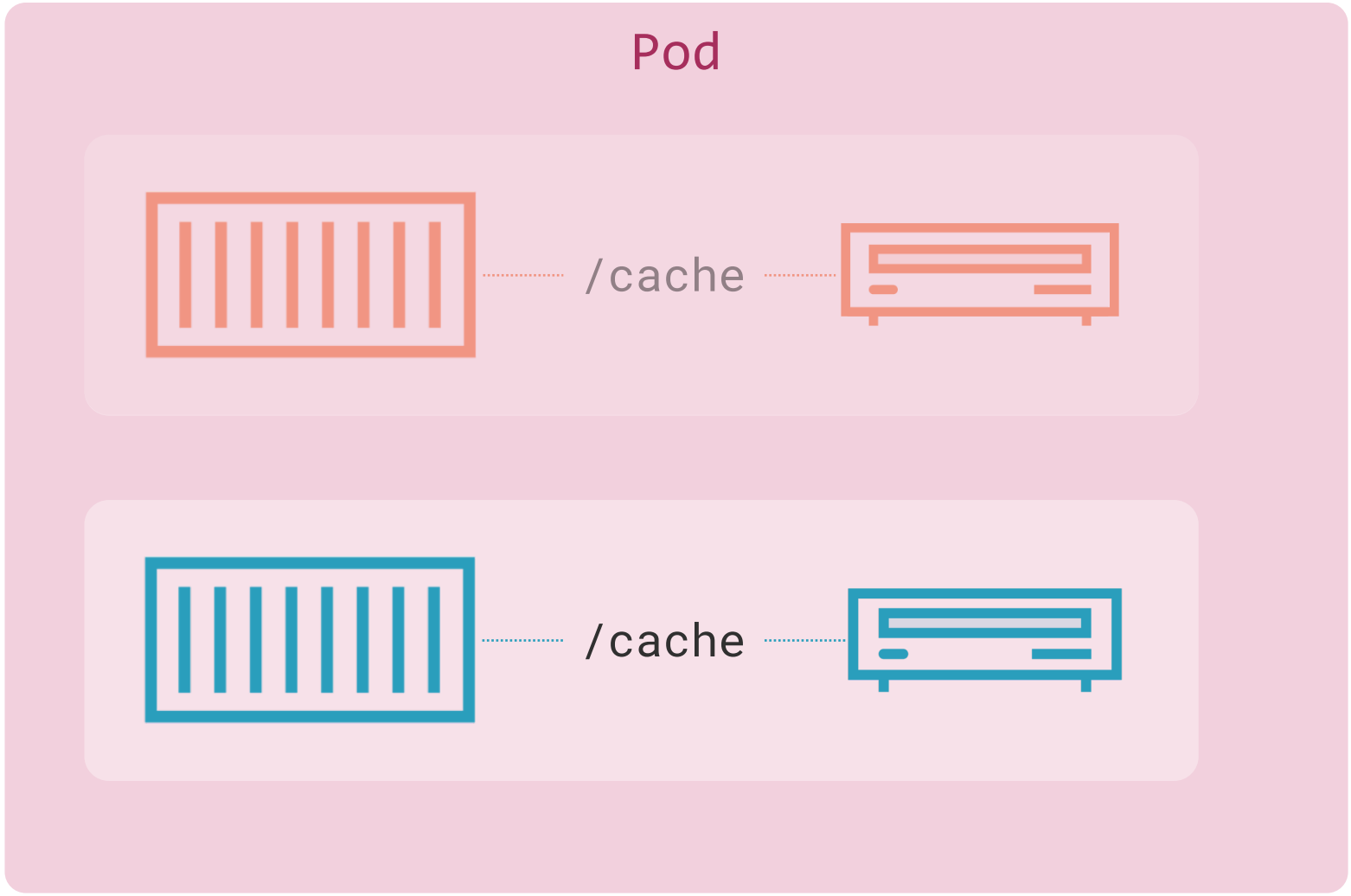


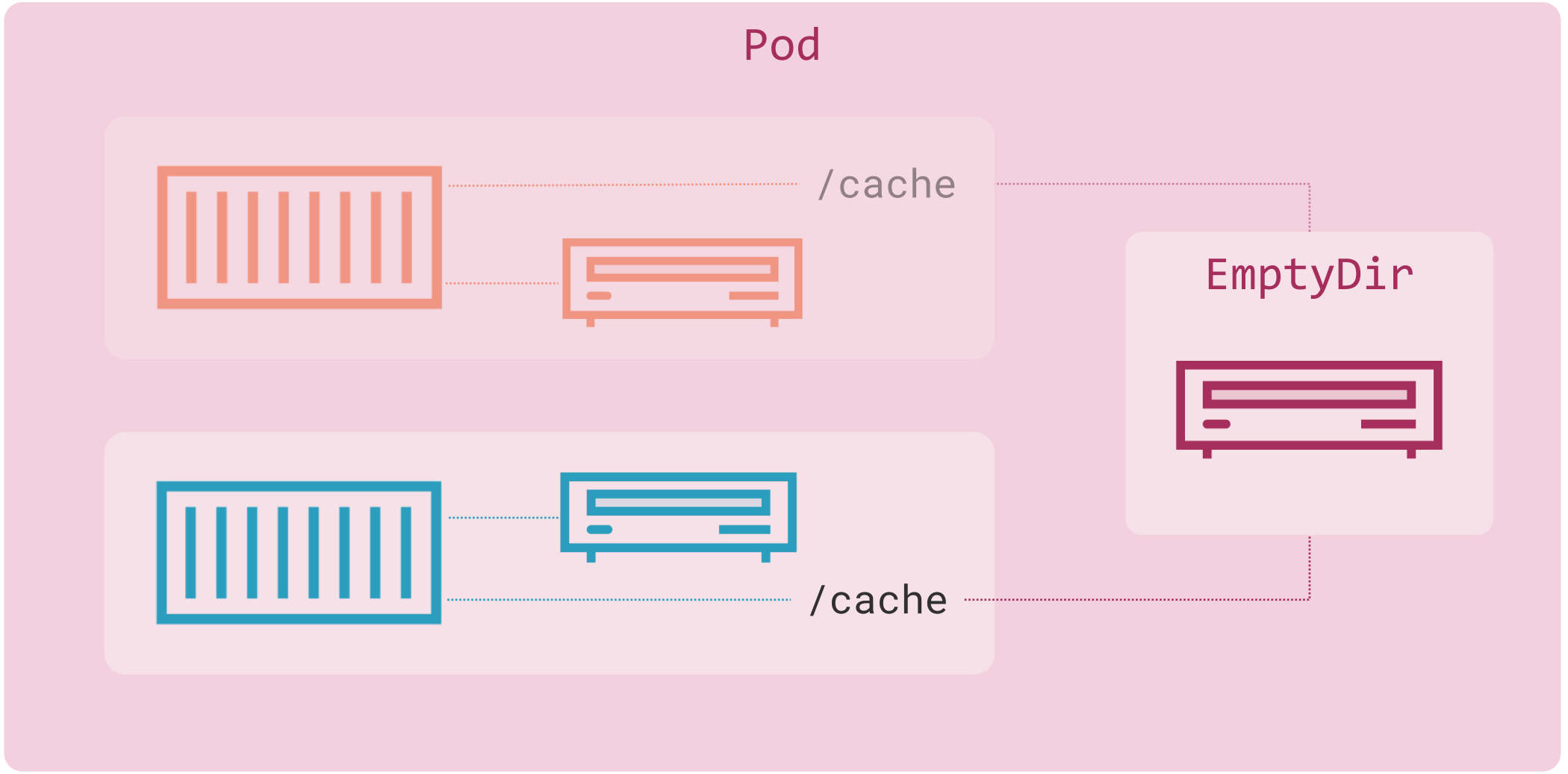
Persistent Storage in Kubernetes

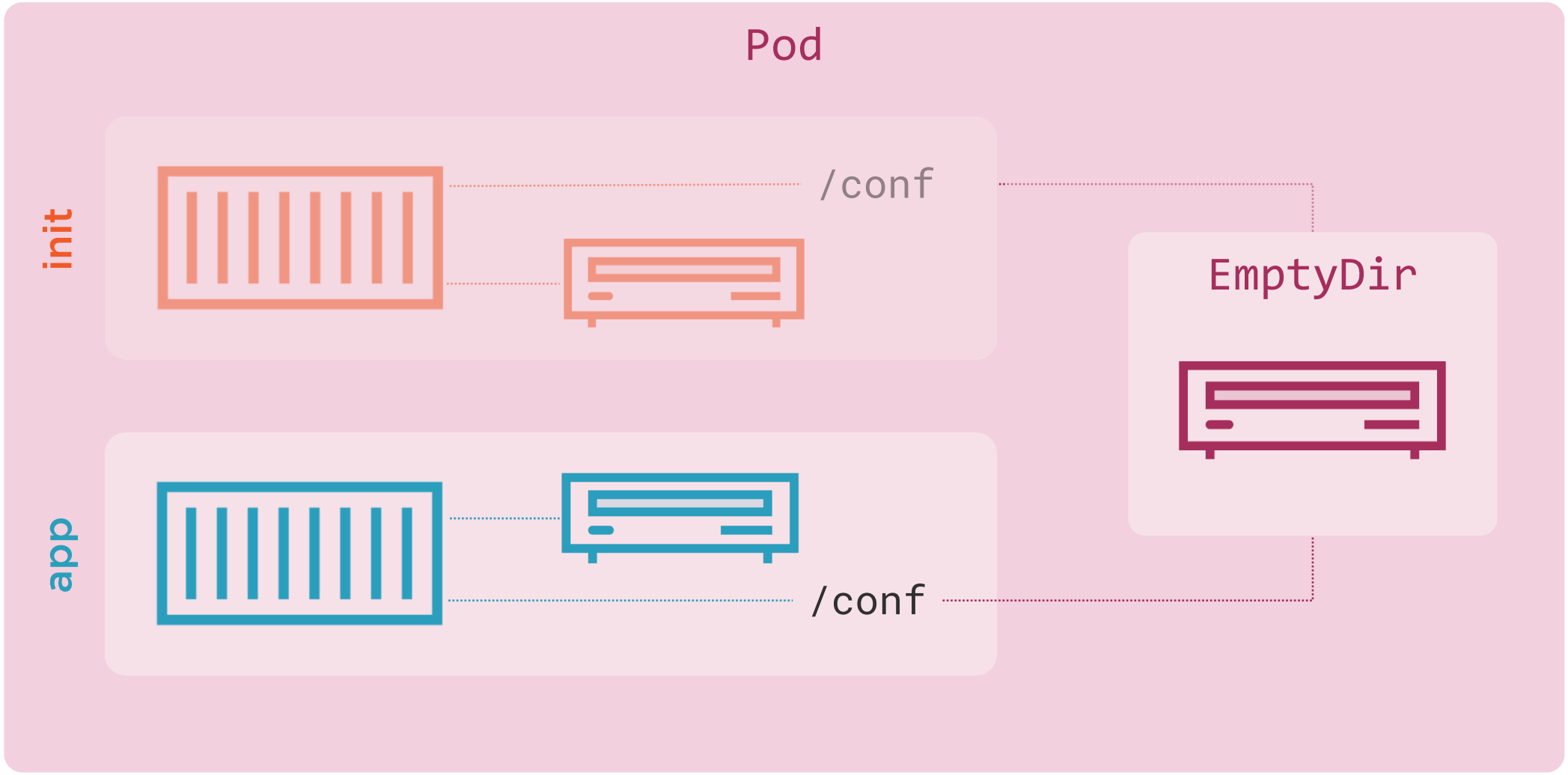
- Data in the container writeable layer
- Pod-level storage with EmptyDir
- Working with read-only filesystems

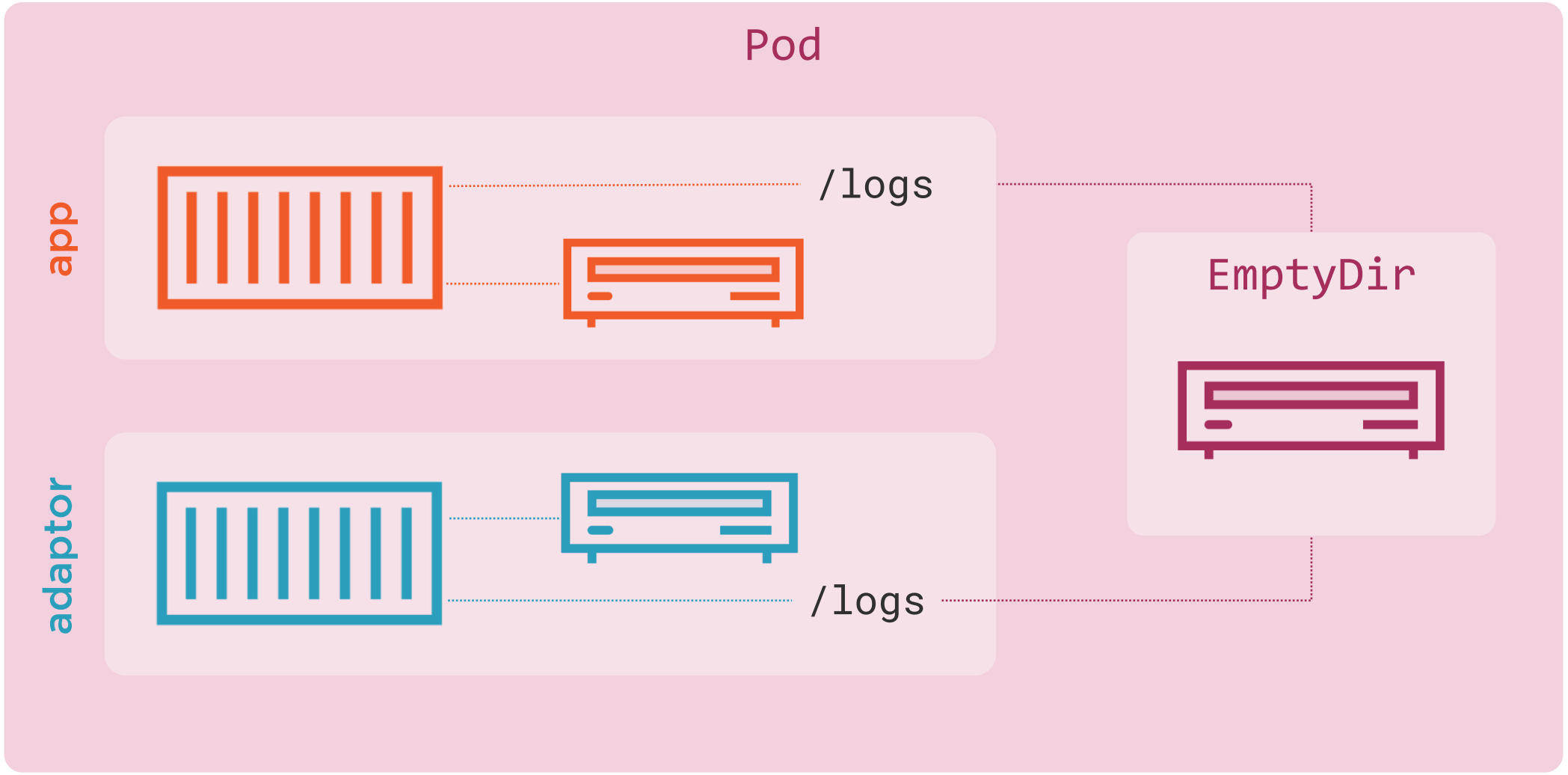

Products - WiredBrain.Web
localhost:8080
120%
WIRED BRAIN
COFFEE
Your coffee
Just
Espresso
$4.00
Americano
$5.00
Flat White
$6.50


web
products-api
products-db
stock-api
EmptyDir


Pod
/cache
/cache


Pod
/cache
EmptyDir
/cache


Pod
init
/conf
EmptyDir
app
/conf


Pod
app
/logs
EmptyDir
adaptor
/logs

Pod

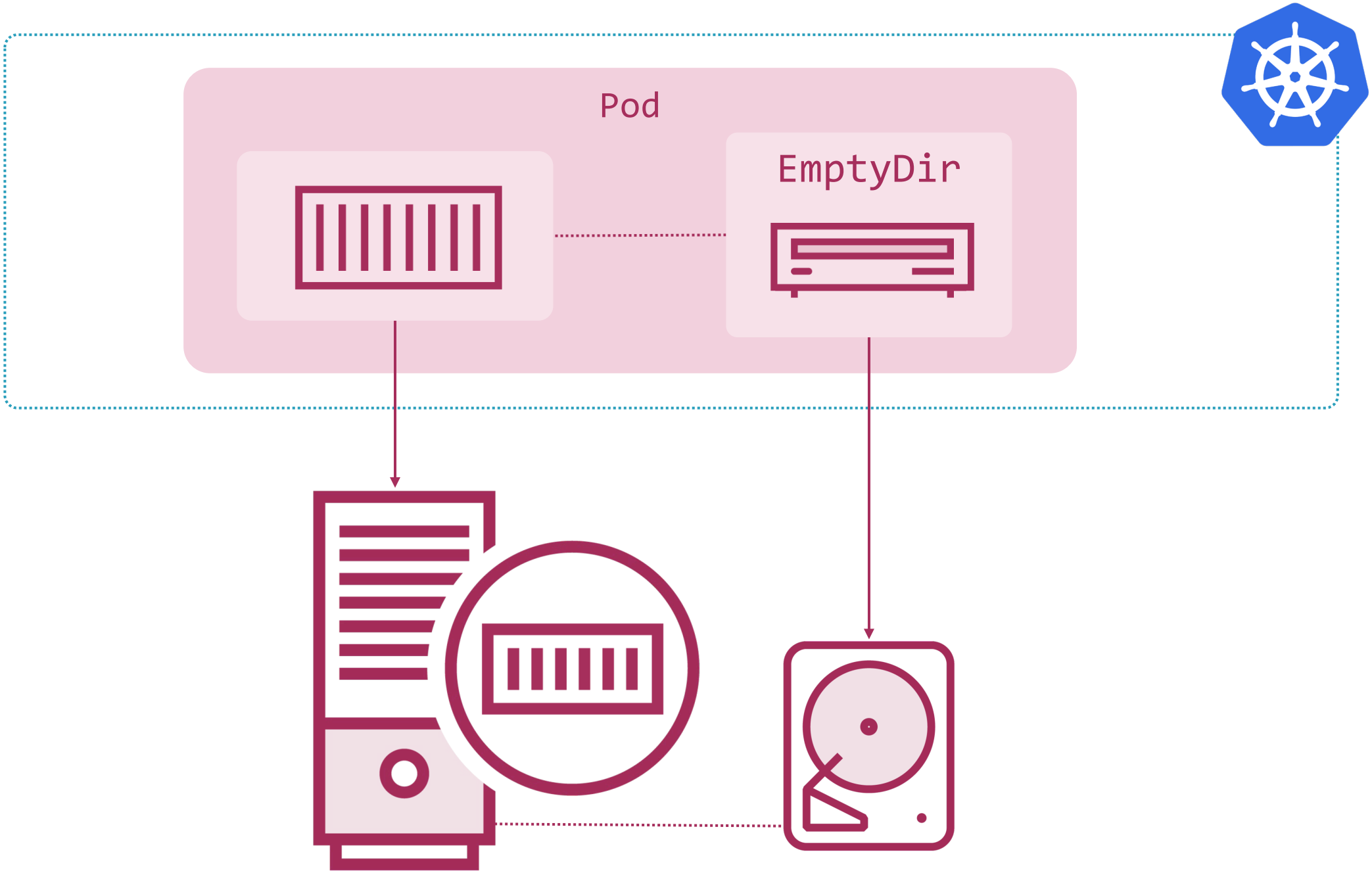
```yaml
# in the Deployment template
spec:
  containers:
    - name: app
      image: wiredbrain/stock-api:22.05
      volumeMounts:
        - name: cache
          mountPath: "/cache"
  volumes:
    - name: cache
      emptyDir: {}
```

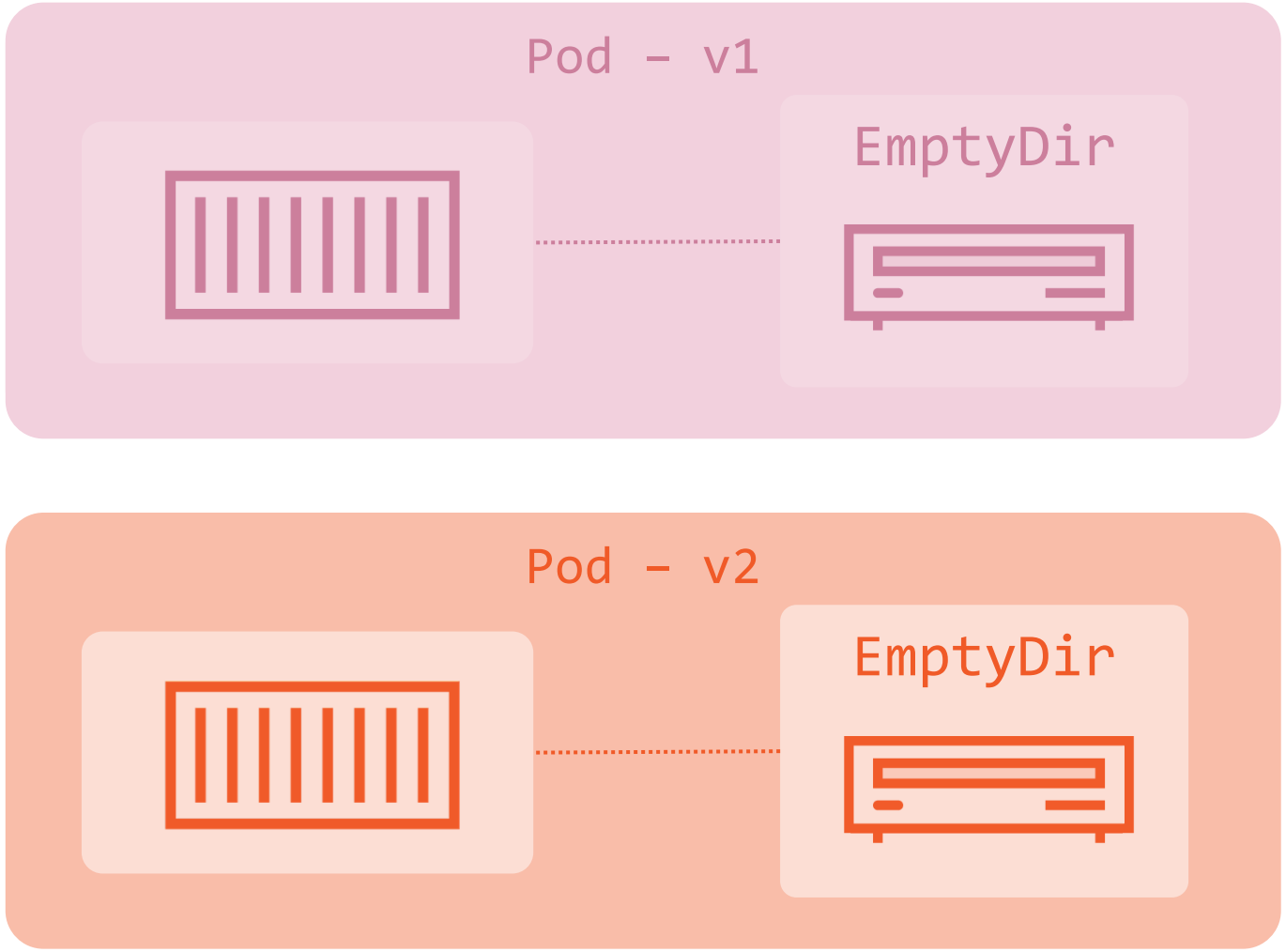
- Pod lifecycle
- Shared between containers
- Concurrently or consecutively

Pod with SecurityContext

```yaml
spec:
  containers:
    - name: app
      image: wiredbrain/stock-api:22.05
      securityContext:
        readOnlyRootFilesystem: true
      volumeMounts:
        - name: cache
          mountPath: "/cache"
  volumes:
    - name: cache
      emptyDir: {}
```

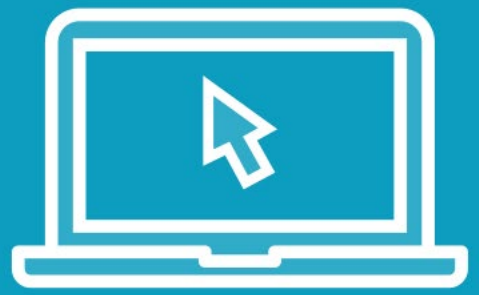
- Read-only filesystem
- Needs app support
- Use EmptyDir for write access


Pod
EmptyDir



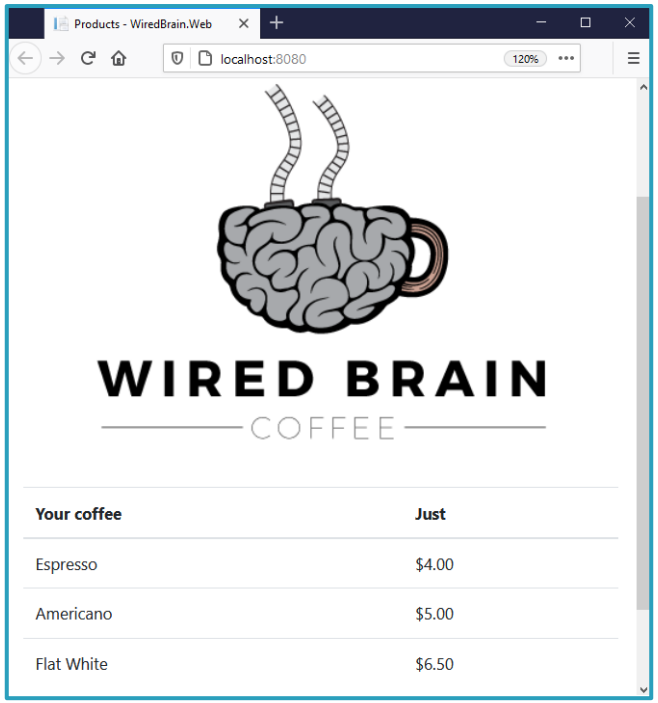
- Environment variables
- Filesystem mounts
- Resources
- Image version

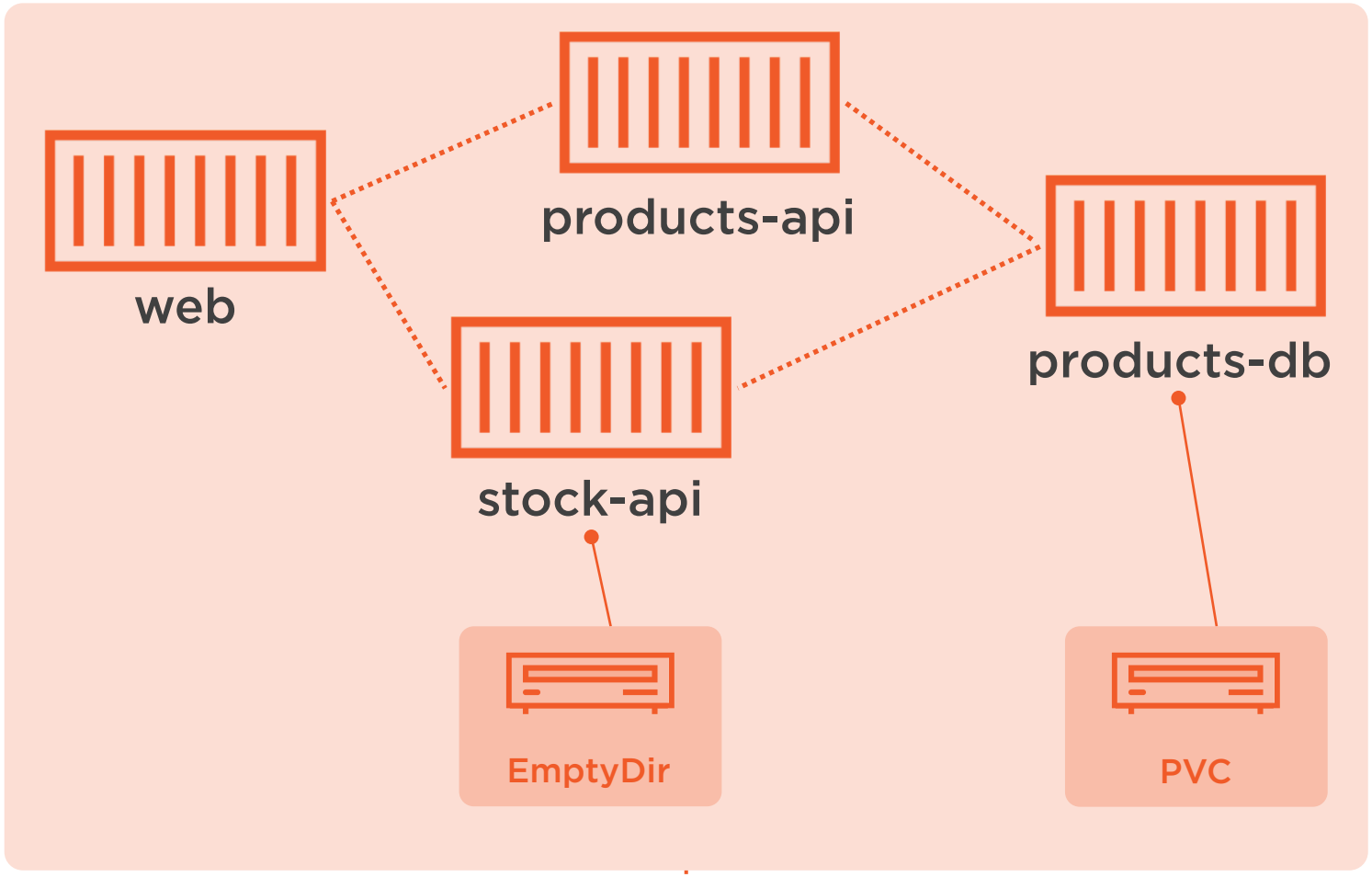
Demo



Persistent Volumes and Claims

- **Losing data during updates**
- **Requesting storage with PVCs**
- **Understanding Storage Classes**


Products - WiredBrain.Web
localhost:8080
120%
WIRED BRAIN
COFFEE
Your coffee
Just
Espresso
$4.00
Americano
$5.00
Flat White
$6.50


web
products-api
products-db
stock-api
EmptyDir
PVC

PersistentVolumeClaim

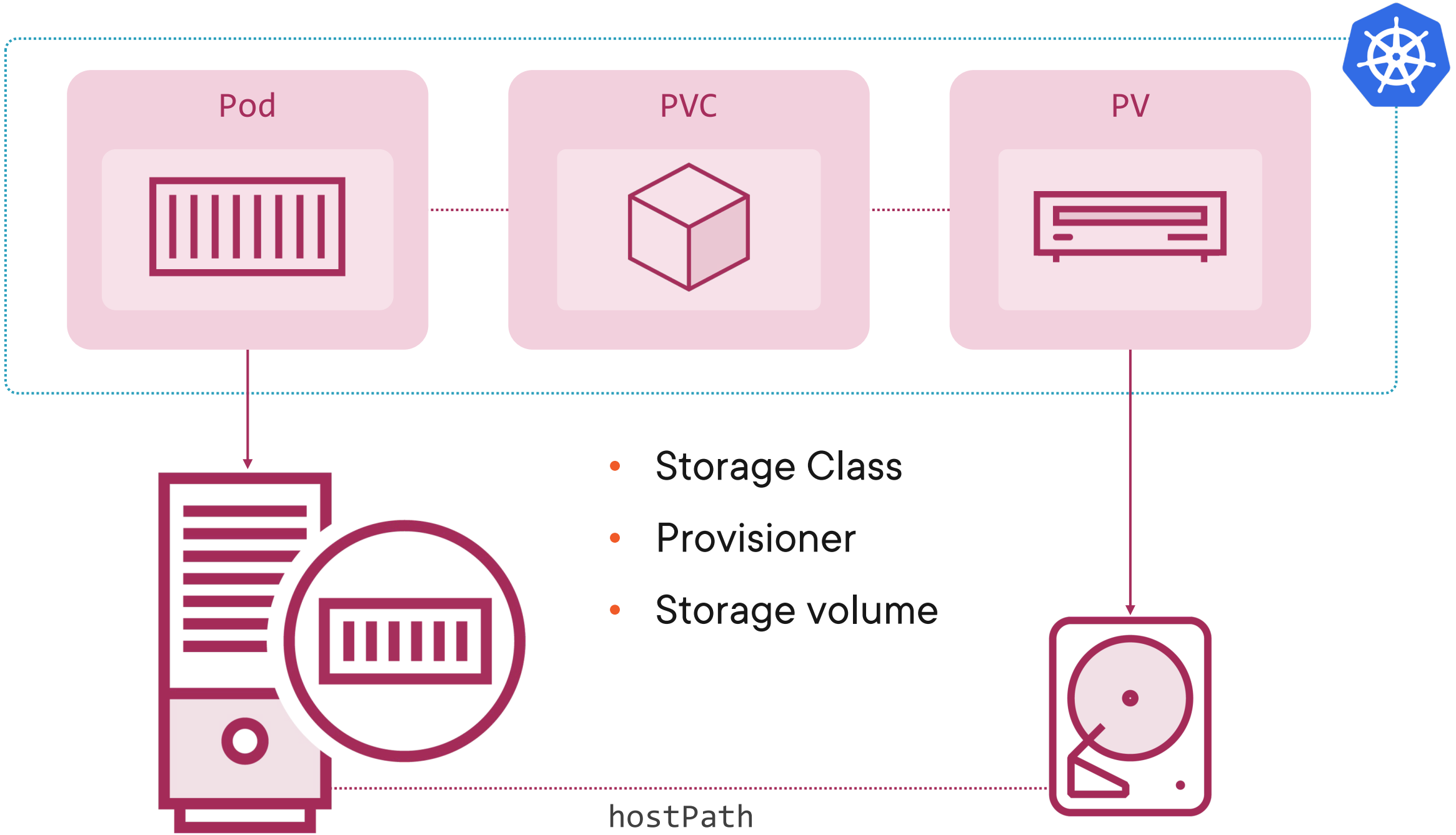
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: products-db-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

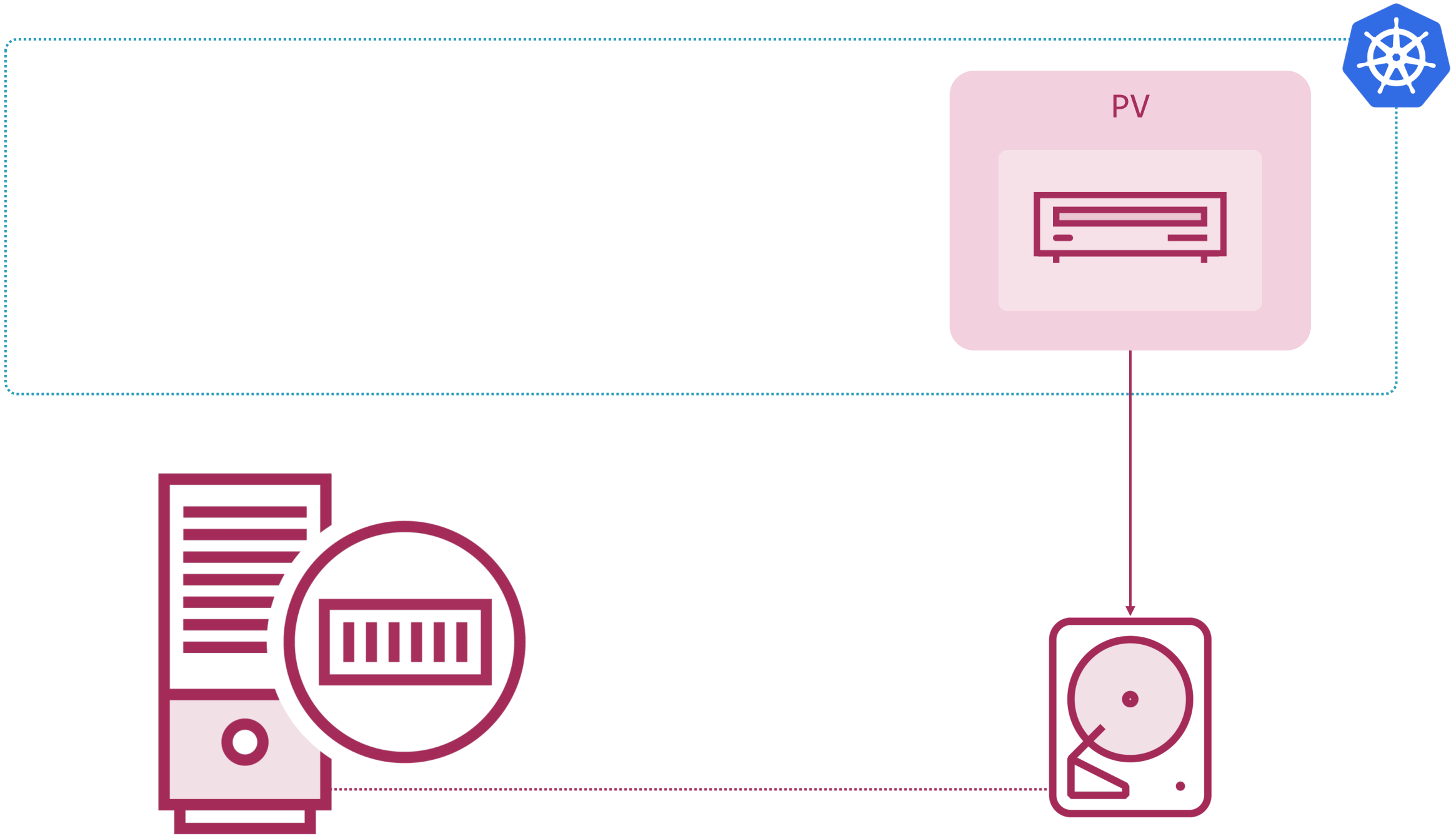
- Request for storage
- Access and size required
- Dynamically provisioned

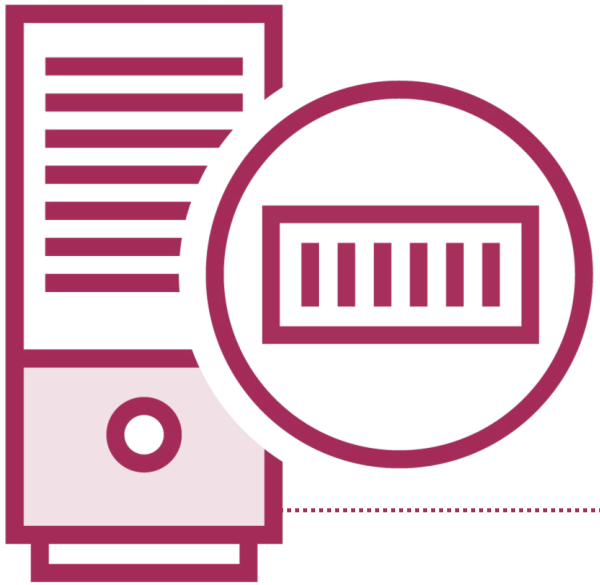
Pod

```
spec:
  containers:
    - name: app
      image: wiredbrain/products-db:22.05
      volumeMounts:
        - name: data
          mountPath: /var/lib/postgresql/data
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: products-db-pvc
```

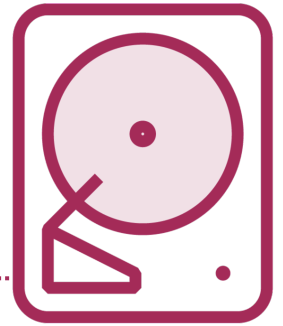
- Standard volume mount
- Binds to PVC
- Storage allocated

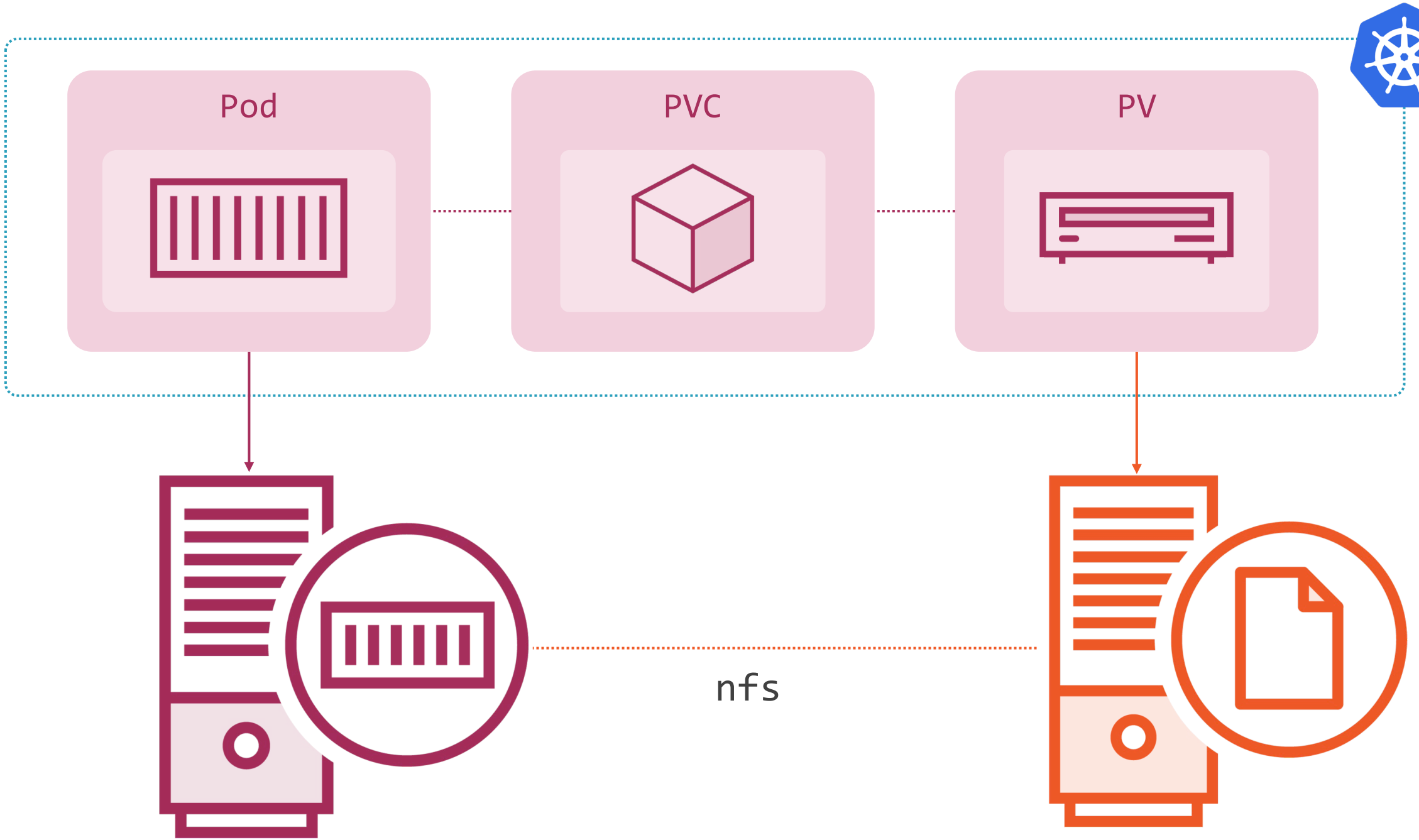

Pod
PVC
PV
Storage Class
Provisioner
Storage volume
hostPath

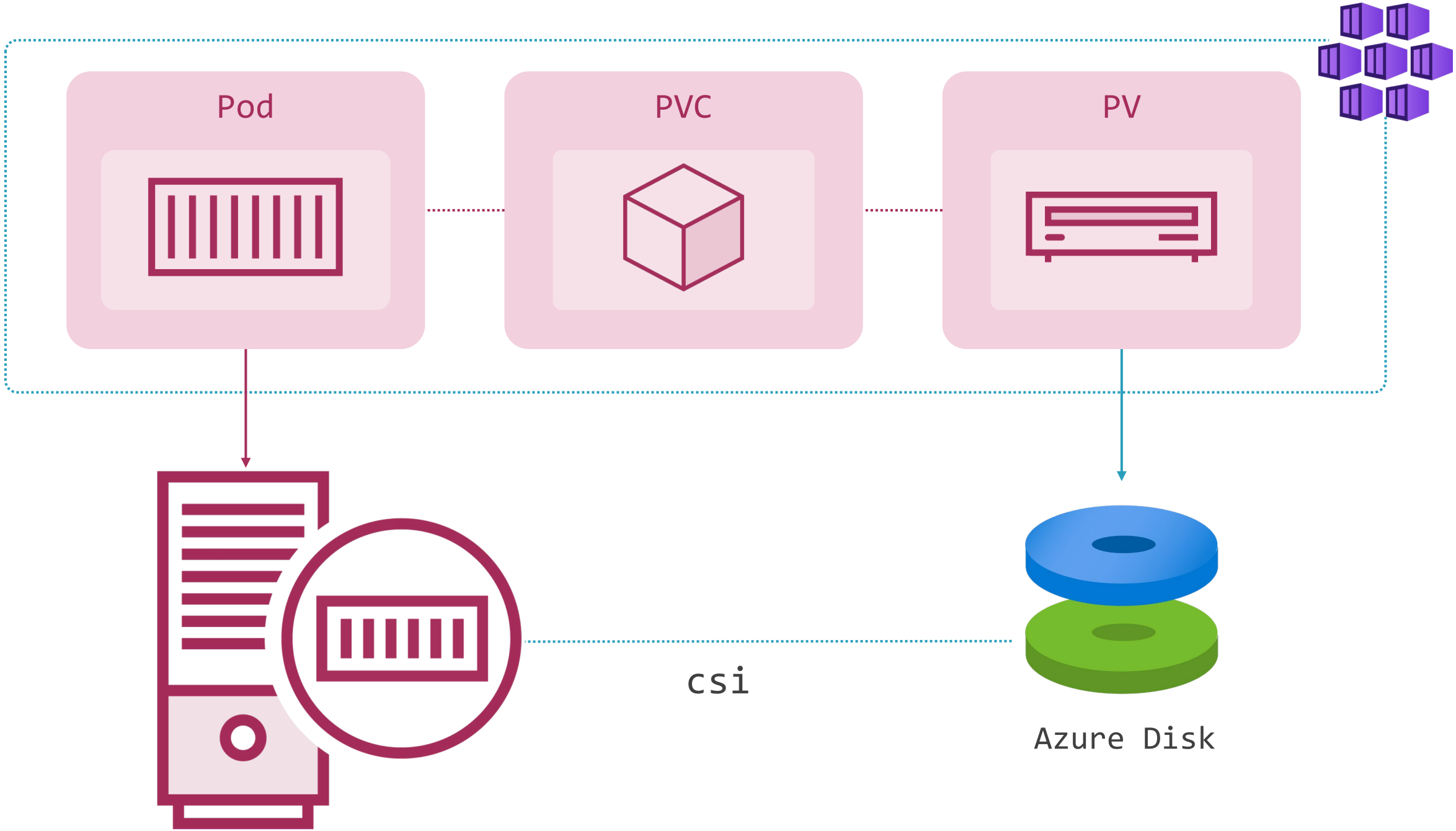

PV

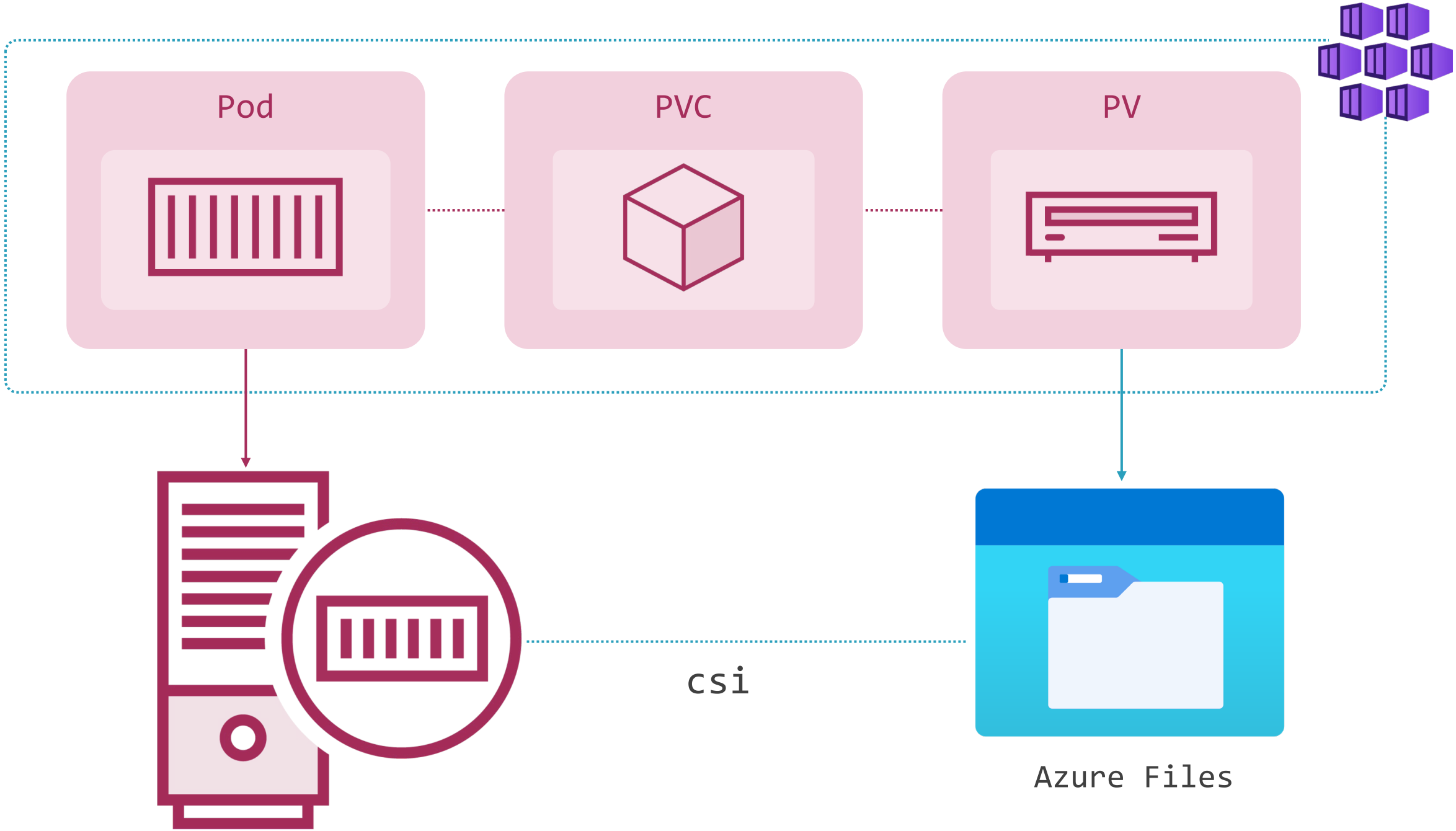


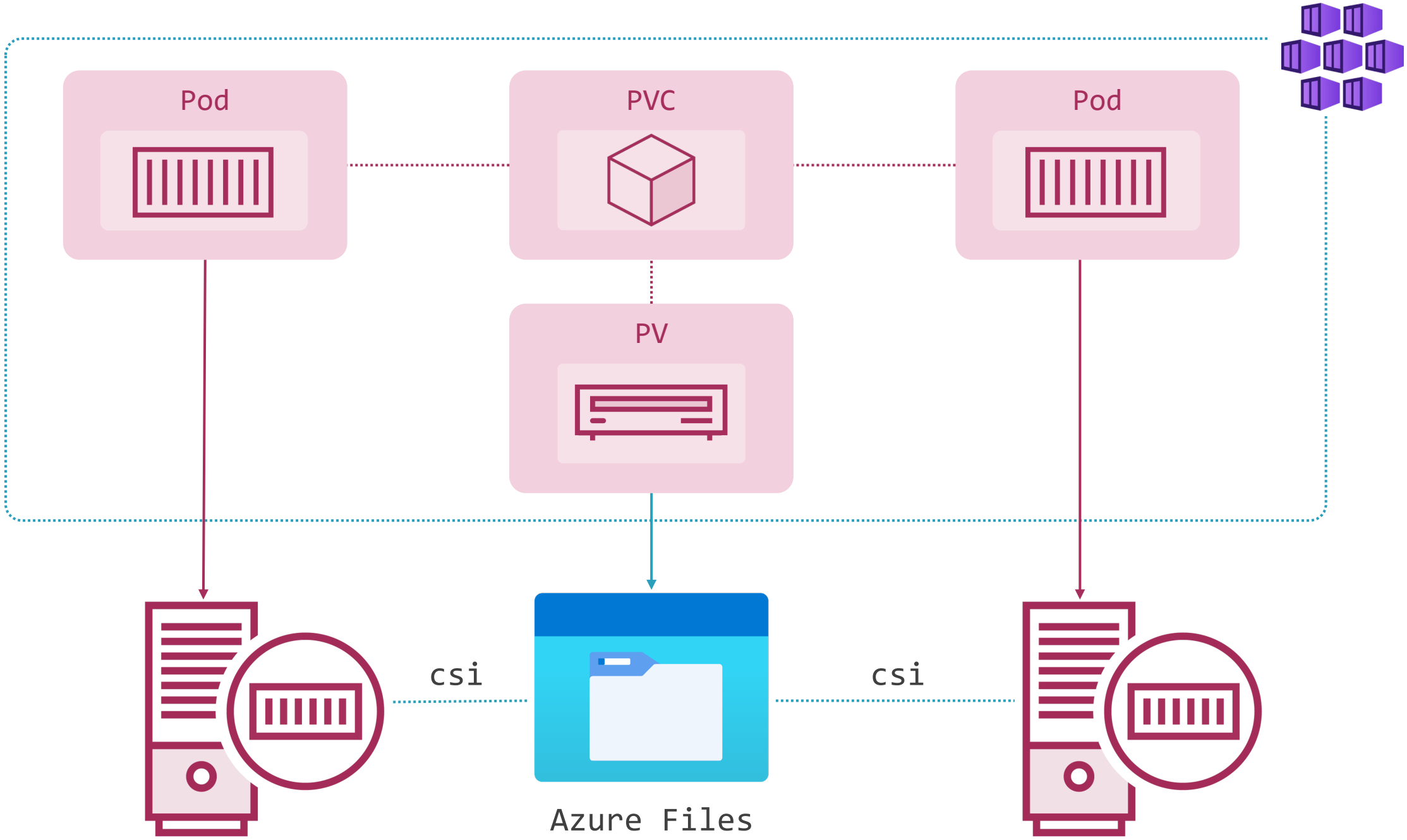
- Reclaim Policy
- Delete (default)
- Retain



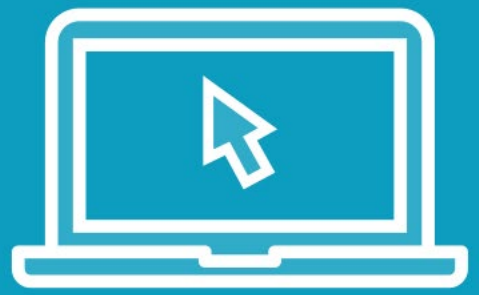

Pod
PVC
PV
nfs


Pod
PVC
PV
csi
Azure Disk


Pod
PVC
PV
csi
Azure Files

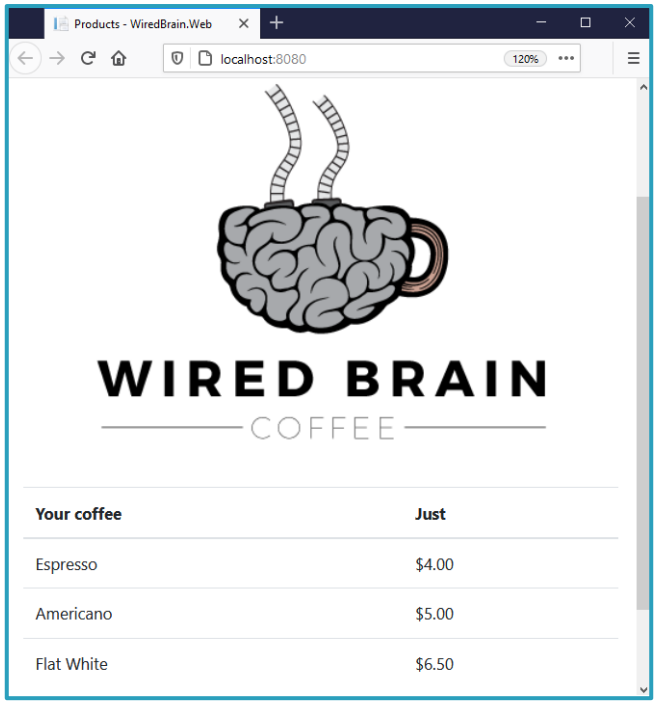

Pod
PVC
Pod
PV
csi
csi
Azure Files

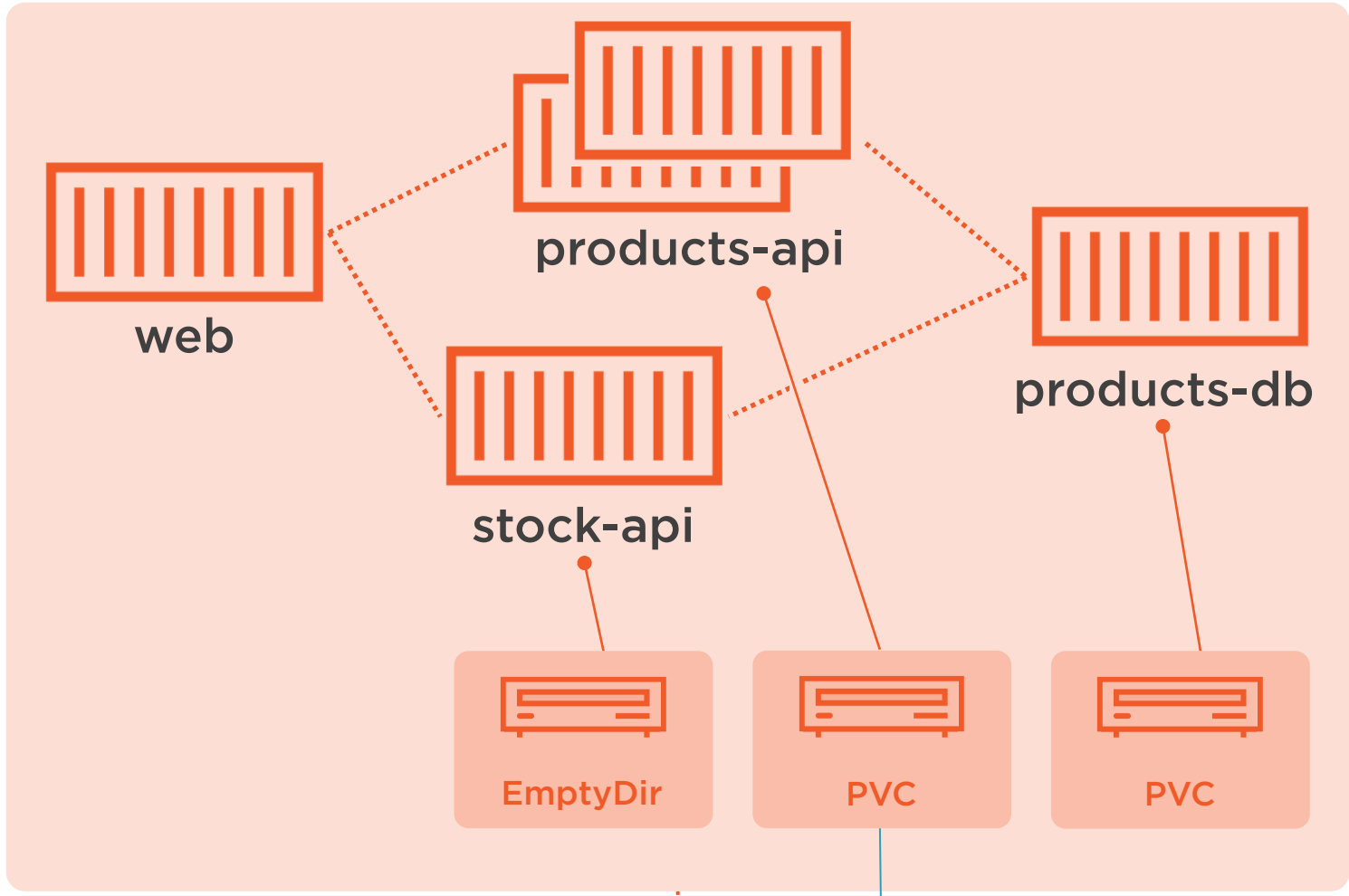
Demo

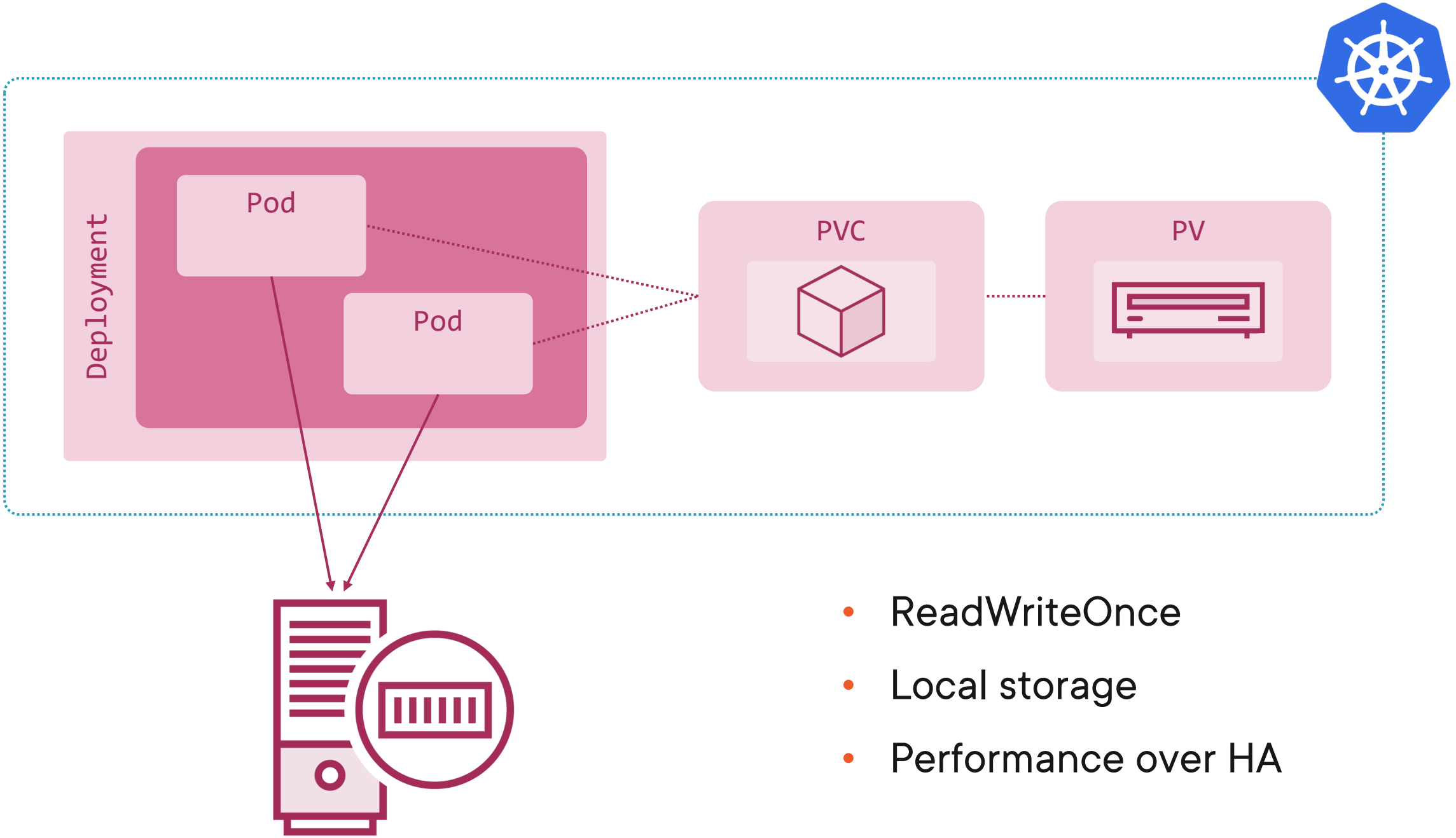


Managing Persistent Volumes

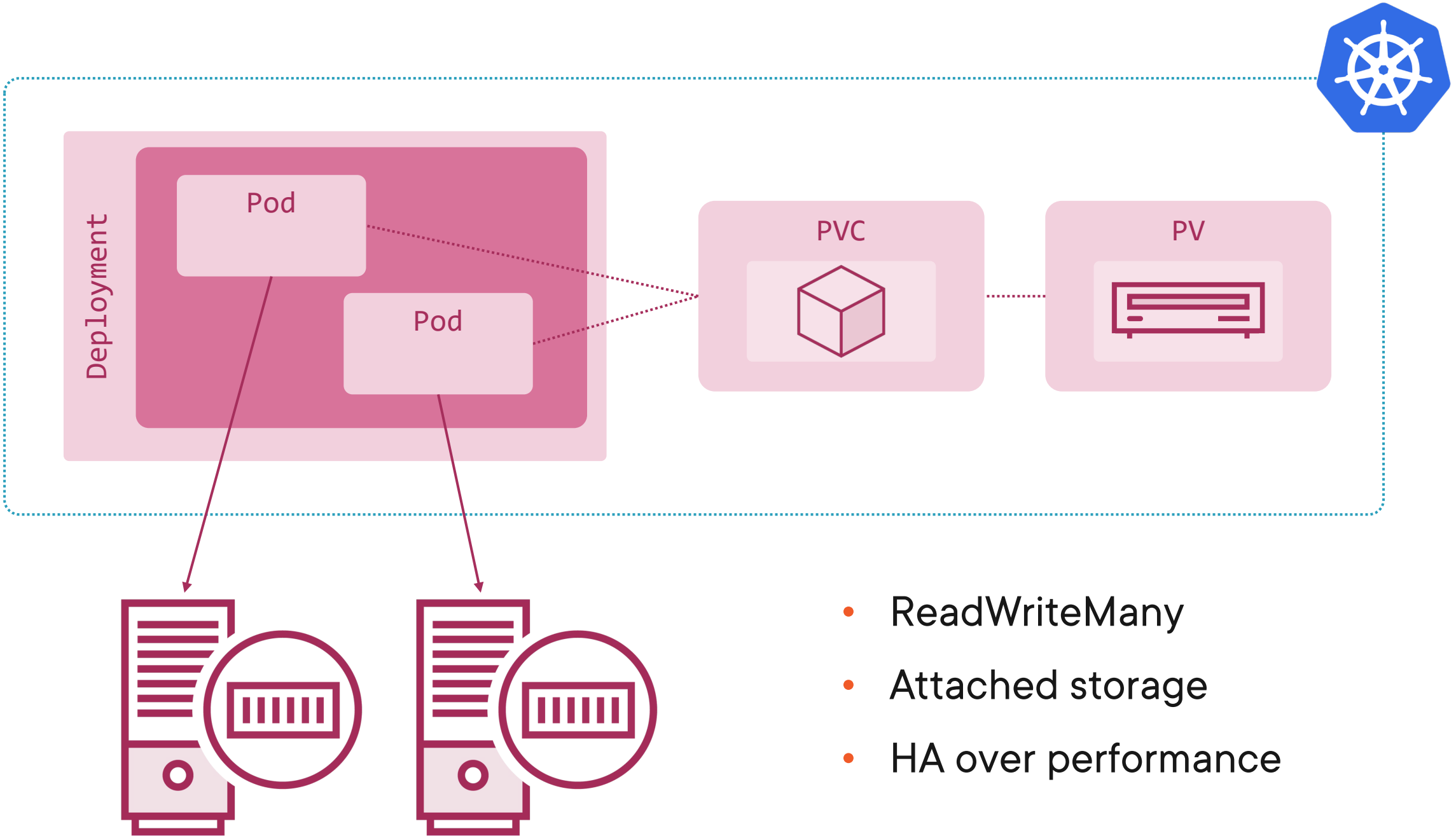
- Explicitly specifying PVs
- Configuring PVs without a provisioner
- Utilizing cluster-wide storage


Products - WiredBrain.Web
localhost:8080
120%
WIRED BRAIN
COFFEE
Your coffee
Just
Espresso
$4.00
Americano
$5.00
Flat White
$6.50


web
products-api
products-db
stock-api
EmptyDir
PVC
PVC



- ReadWriteOnce
- Local storage
- Performance over HA



- ReadWriteMany
- Attached storage
- HA over performance

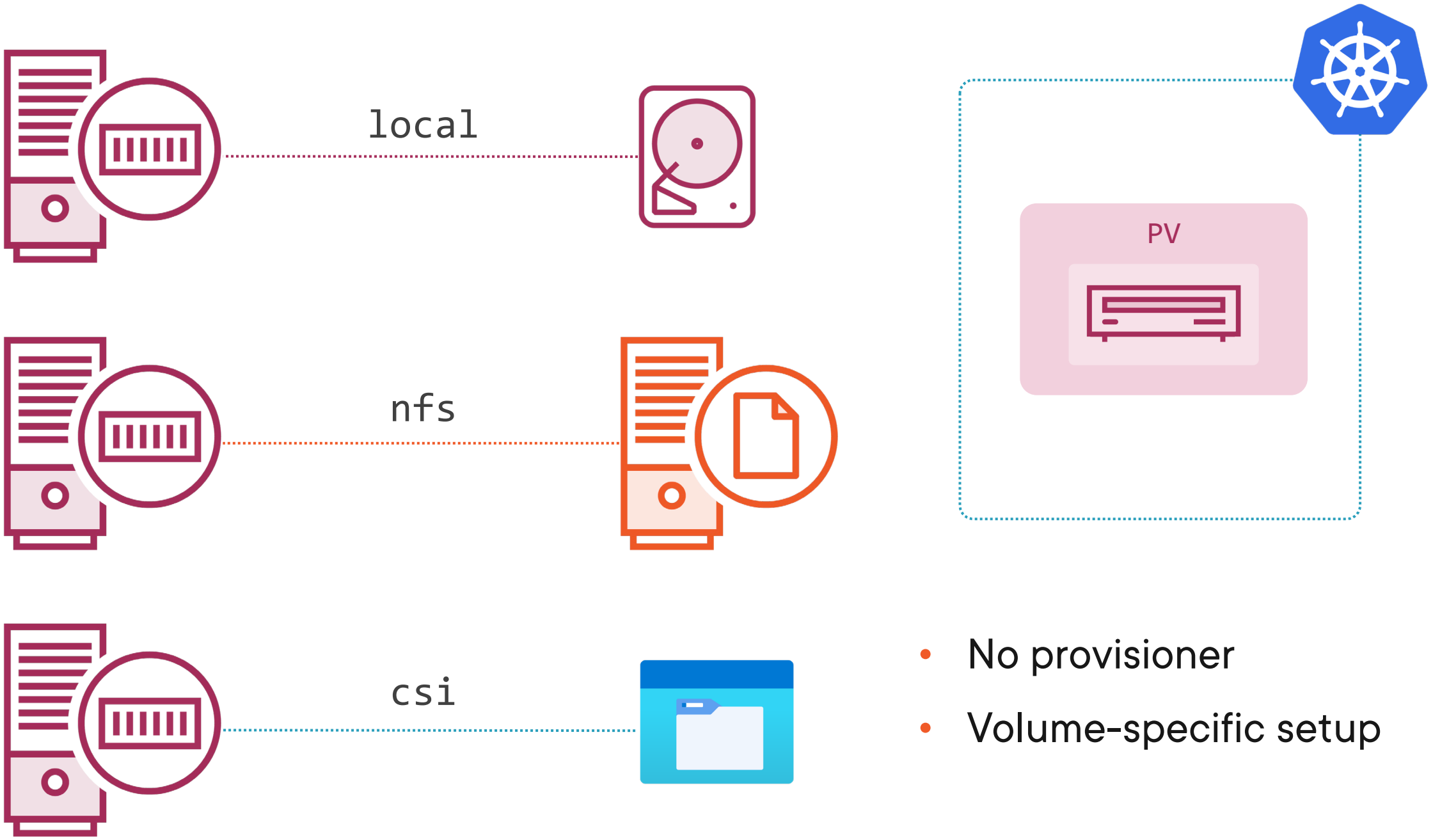
PersistentVolume

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: products-db-pv
spec:
  capacity:
    storage: 500Mi
  accessModes:
    - ReadWriteOnce
  local:
    path: /volumes/products-db
```

PersistentVolume

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: products-db-pv
spec:
  capacity:
    storage: 500Mi
  accessModes:
    - ReadWriteOnce
  local:
    path: /volumes/products-db
```

- Abstract storage unit
- Capabilities – access & size
- Includes volume spec

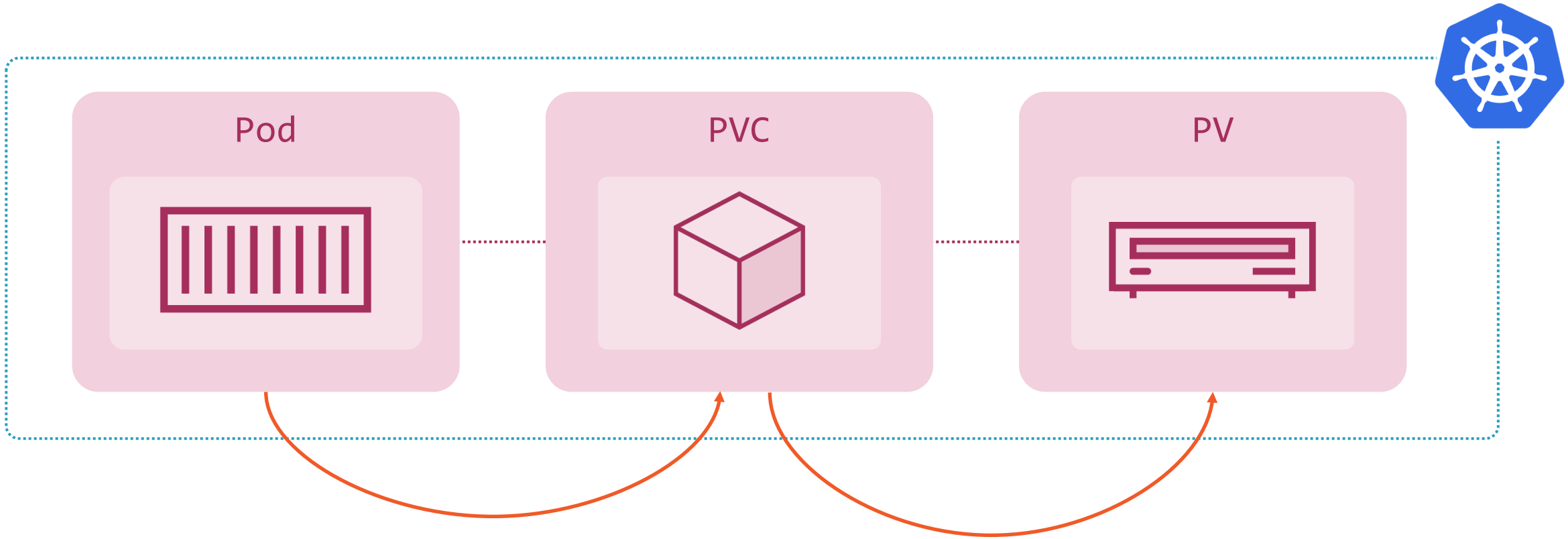


- No provisioner
- Volume-specific setup

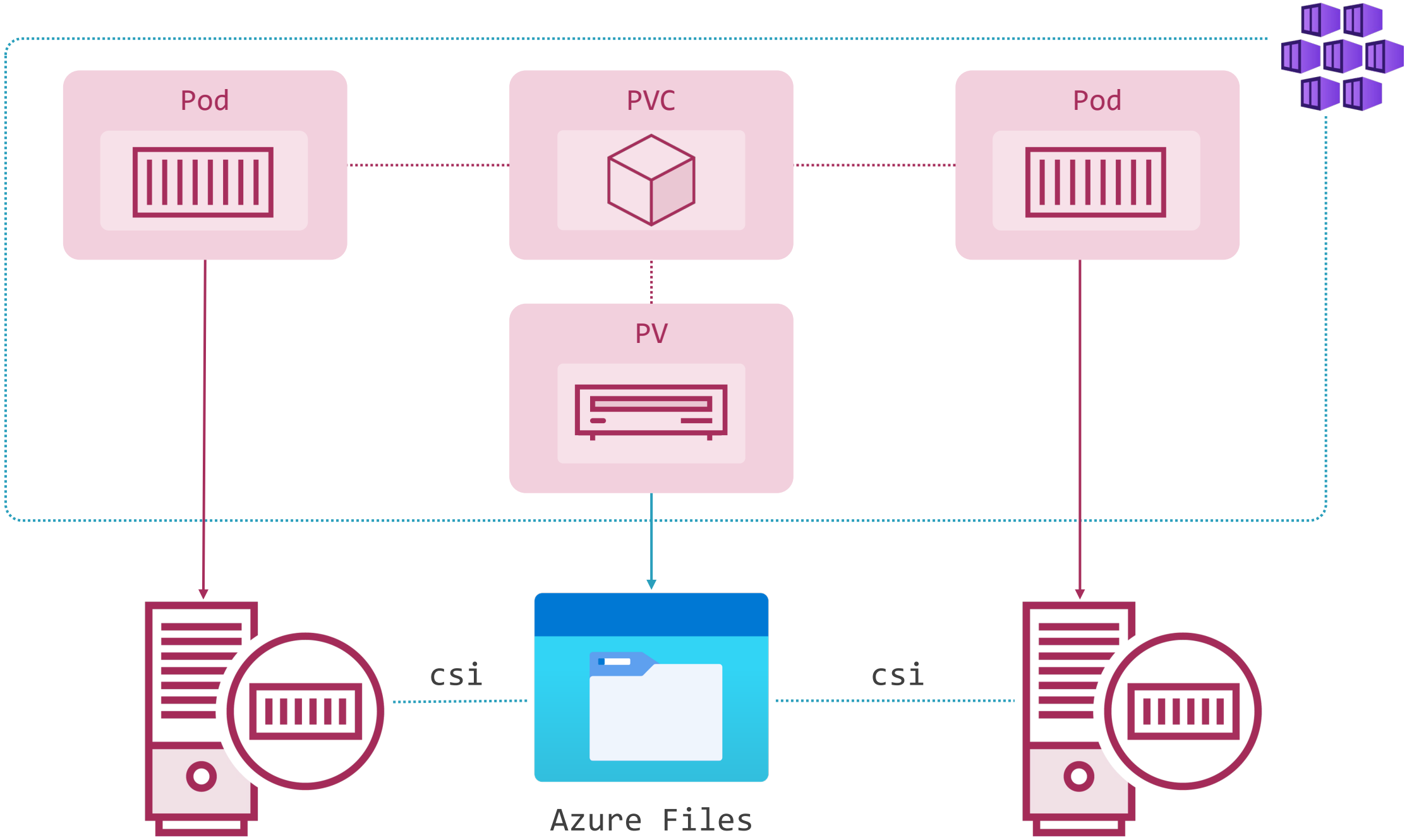
PersistentVolumeClaim

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: products-db-pvc-manual
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: ""
  volumeName: products-db-pv
```

- Explicit PV name
- No Storage Class
- PV capabilities must match



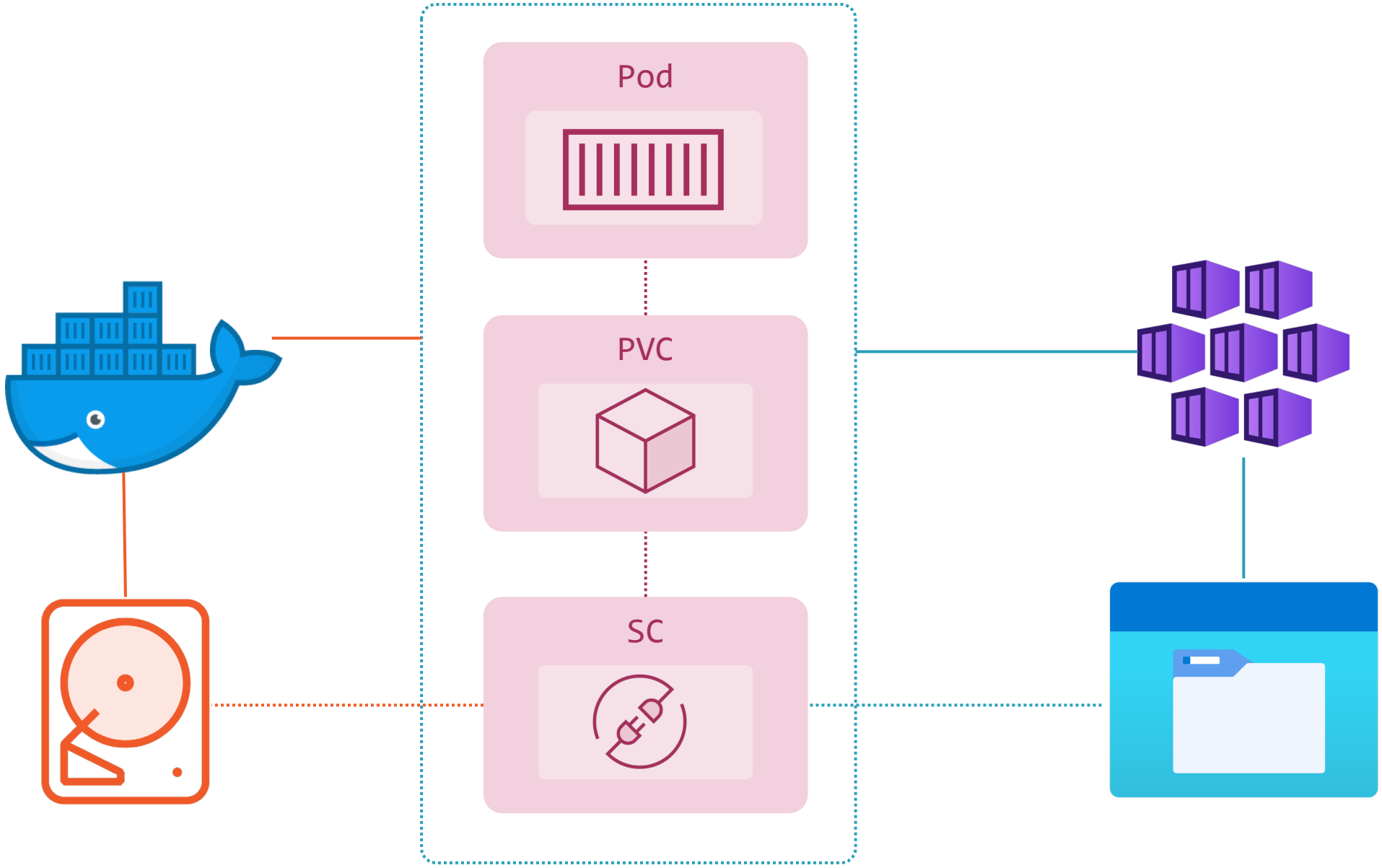
- Objects bound
- Volume mount may fail
- Standard retry logic


Pod
PVC
Pod
PV
csi
csi
Azure Files

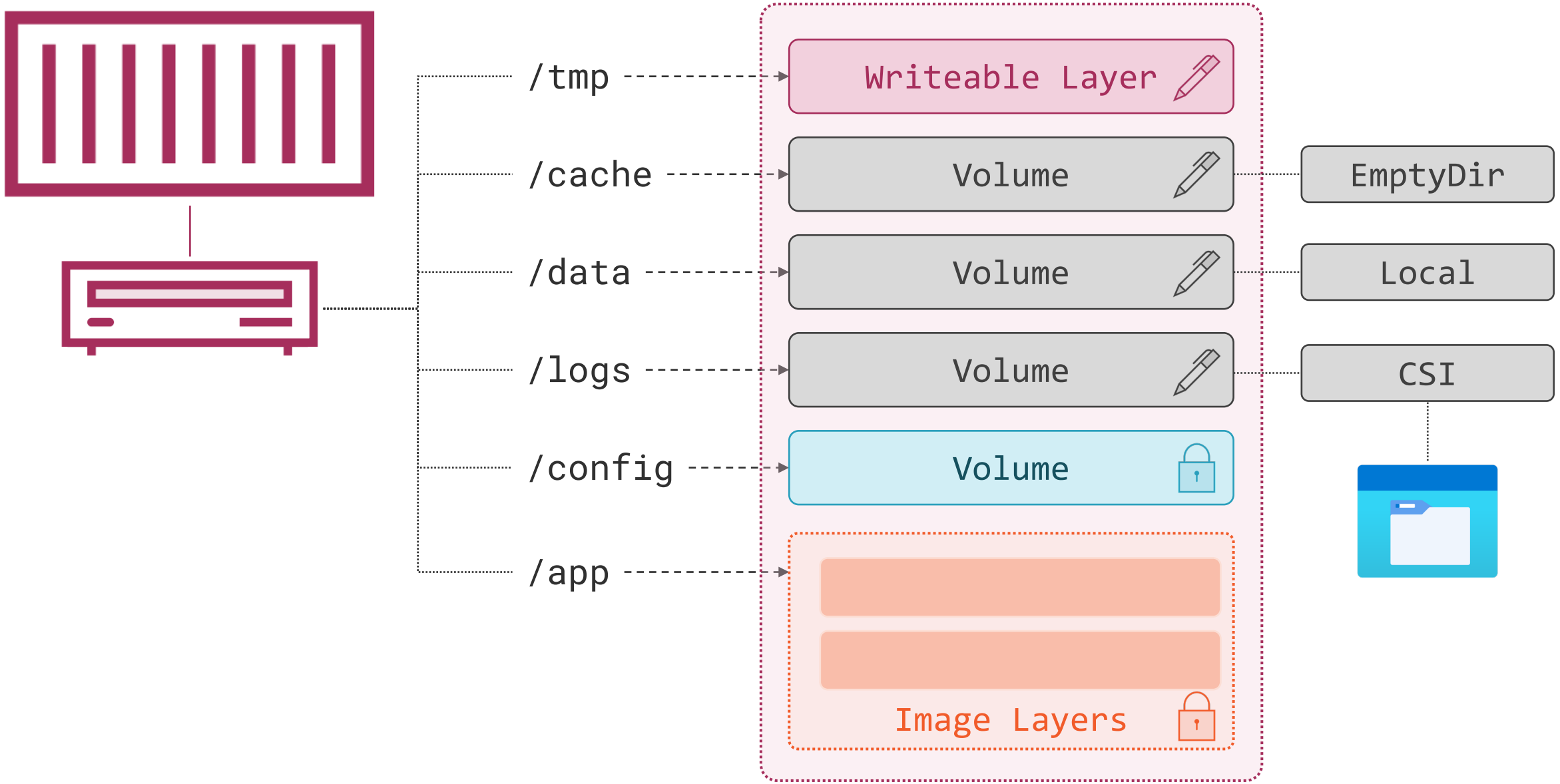
PersistentVolumeClaim

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: products-api-logs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 500Mi
  storageClassName: azurefile-csi
```

- Pods on many nodes
- Explicit Storage Class
- Abstract persistence model


Pod
PVC
SC

Module Summary


/tmp
/cache
/data
/logs
/config
/app
Writeable Layer
Volume
Volume
Volume
Volume
Image Layers
EmptyDir
Local
CSI

```yaml
volumeMounts:
  - name: cache
    mountPath: "/cache"
  - name: data
    mountPath: "/data"
  - name: logs
    mountPath: "/logs"
```

```yaml
volumes:
  - name: cache
    emptyDir: {}
  - name: data
    persistentVolumeClaim:
      claimName: pvc-data-local
  - name: logs
    persistentVolumeClaim:
      claimName: pvc-logs-nfs
```

Module Summary

Persistent Volume Claims

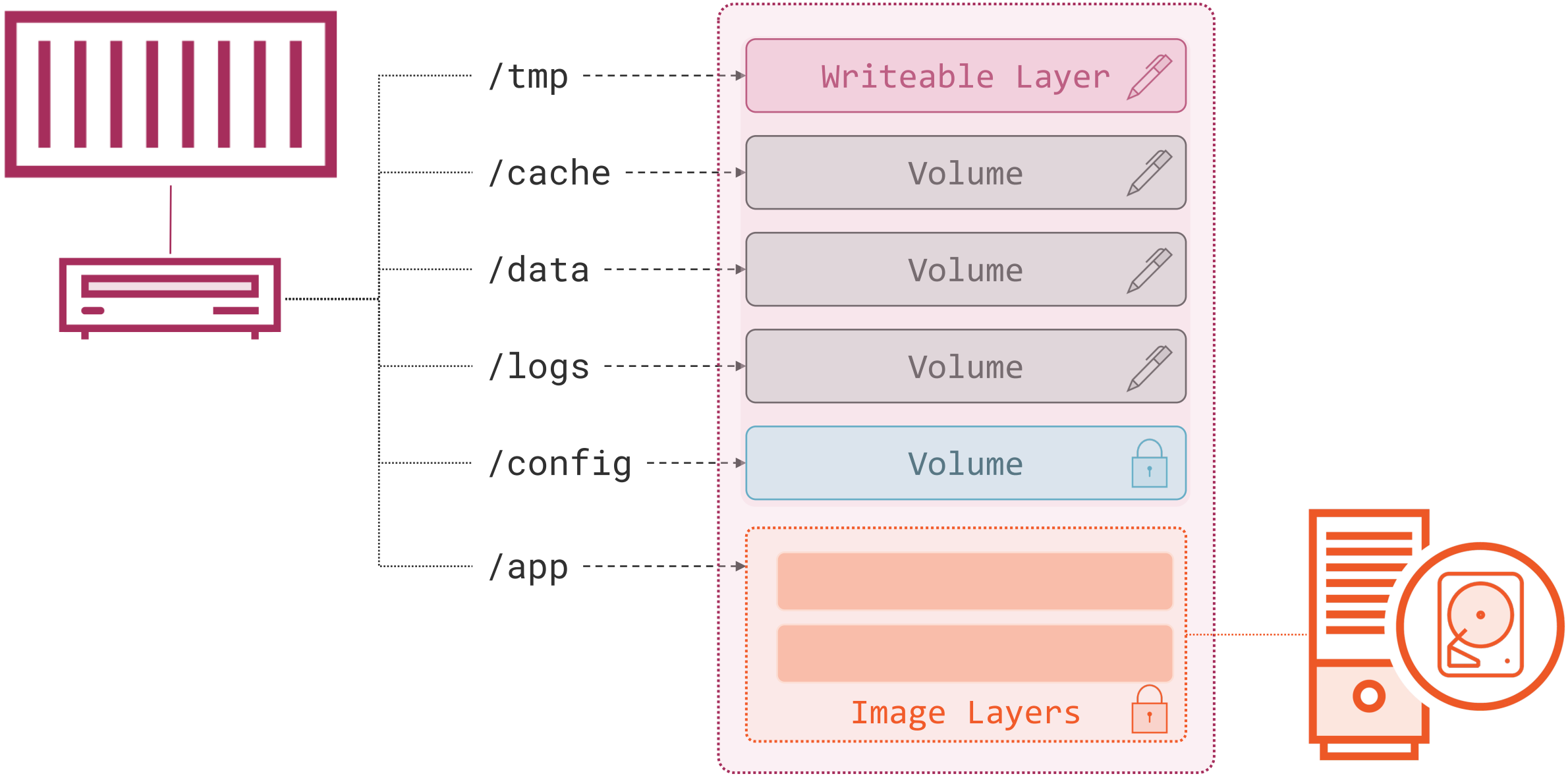
- Abstract storage request
- Dynamically provisioned
- Named Storage Class

Persistent Volumes

- Storage unit
- Physical implementation details
- Separate lifecycle

Storage Classes

- Cluster capabilities
- Single node or cluster-wide
- Allows abstraction


/tmp
/cache
/data
/logs
/config
/app
Writeable Layer
Volume
Volume
Volume
Volume
Image Layers

Up Next:
Managing Storage on Servers and Registries