

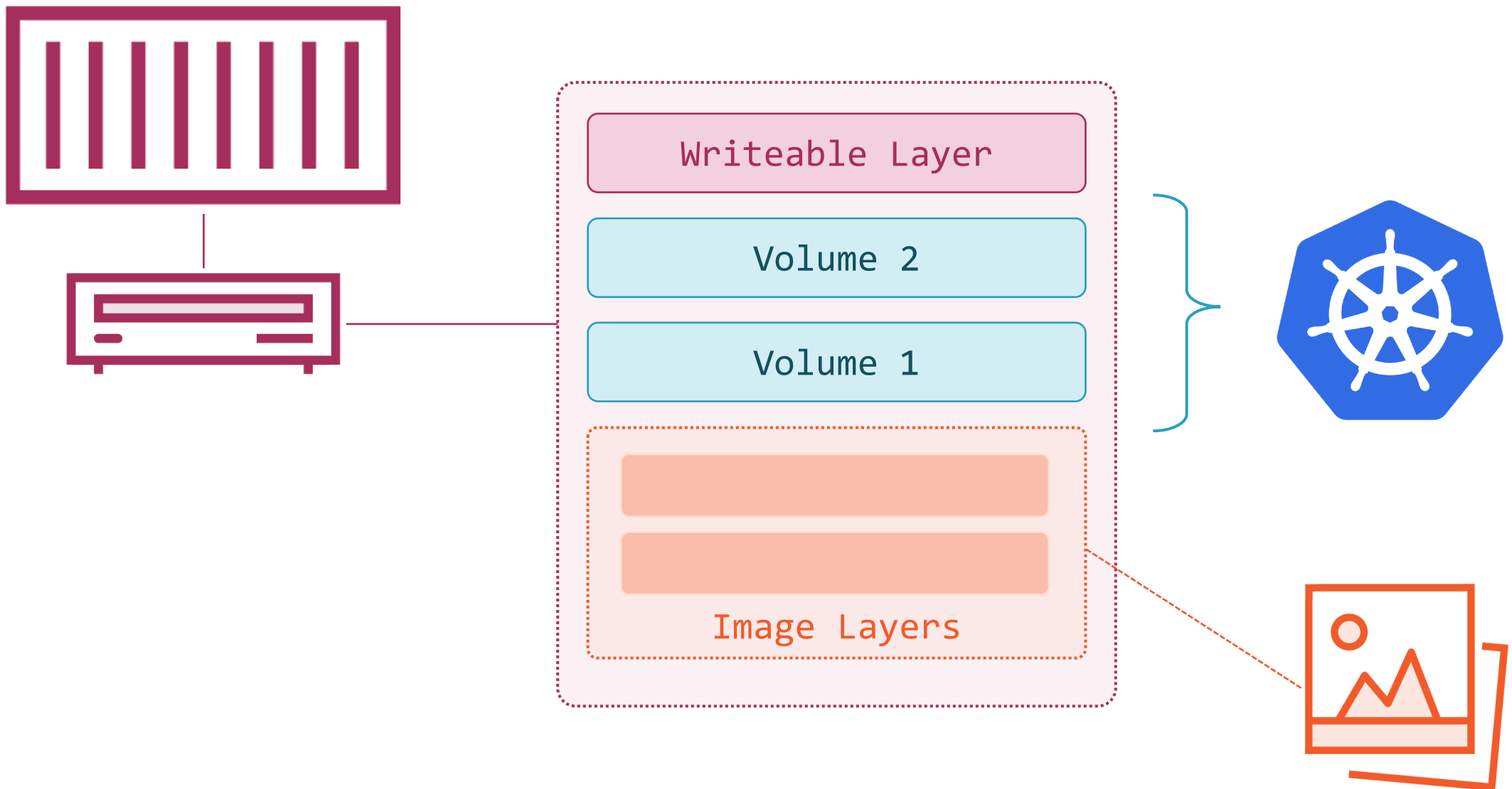
Reading Application Configuration in Kubernetes Volumes

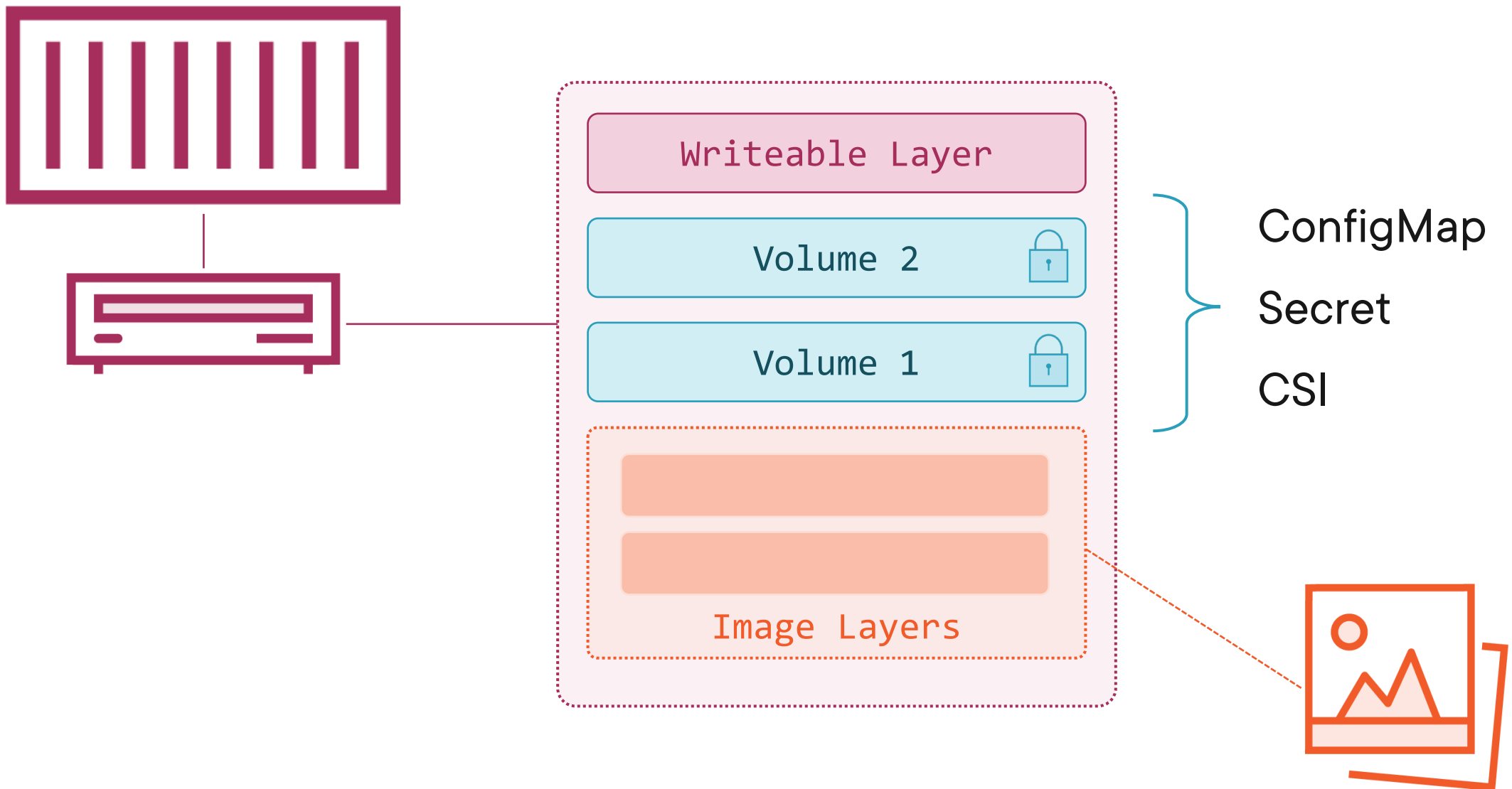


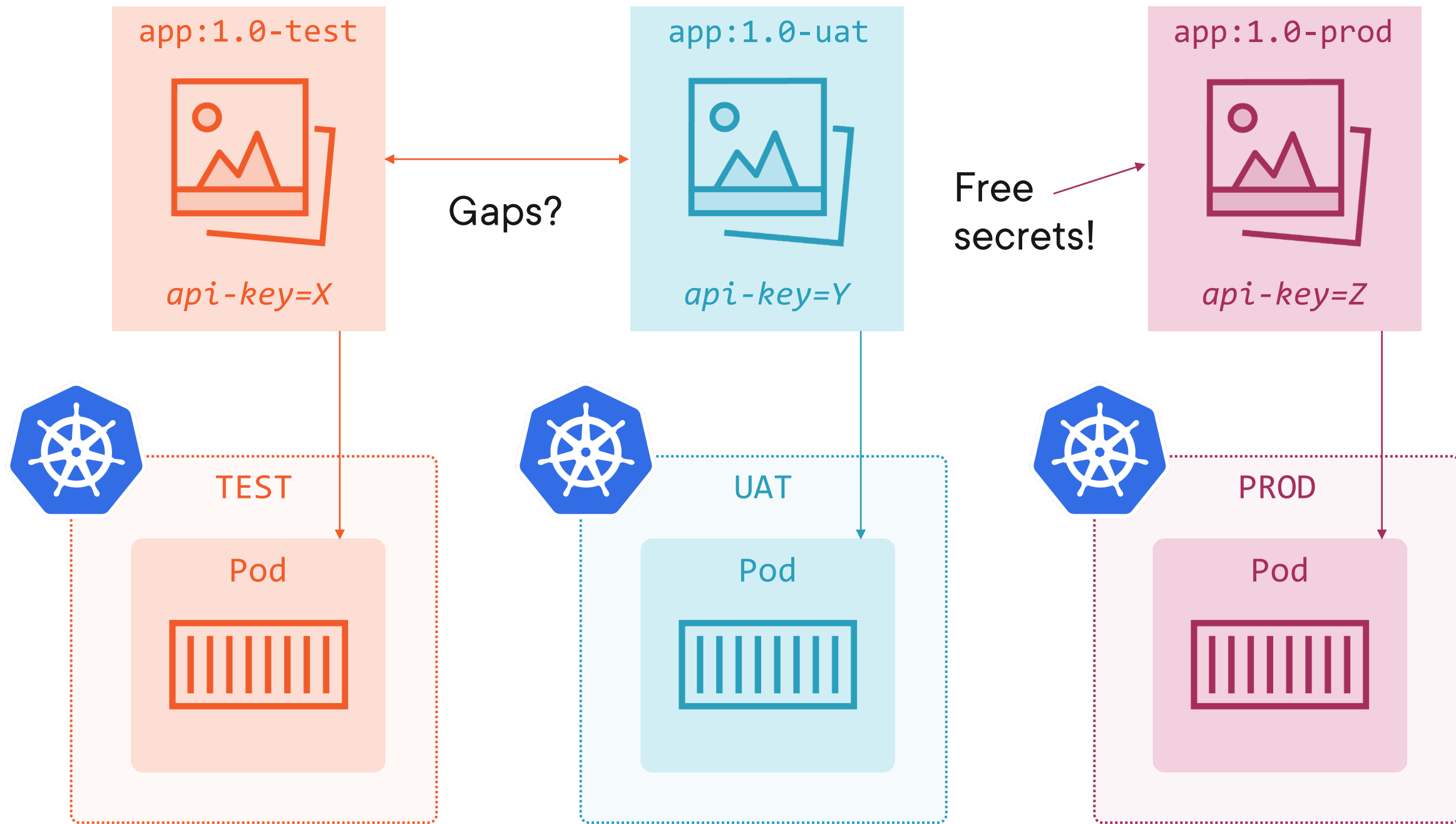
Elton Stoneman

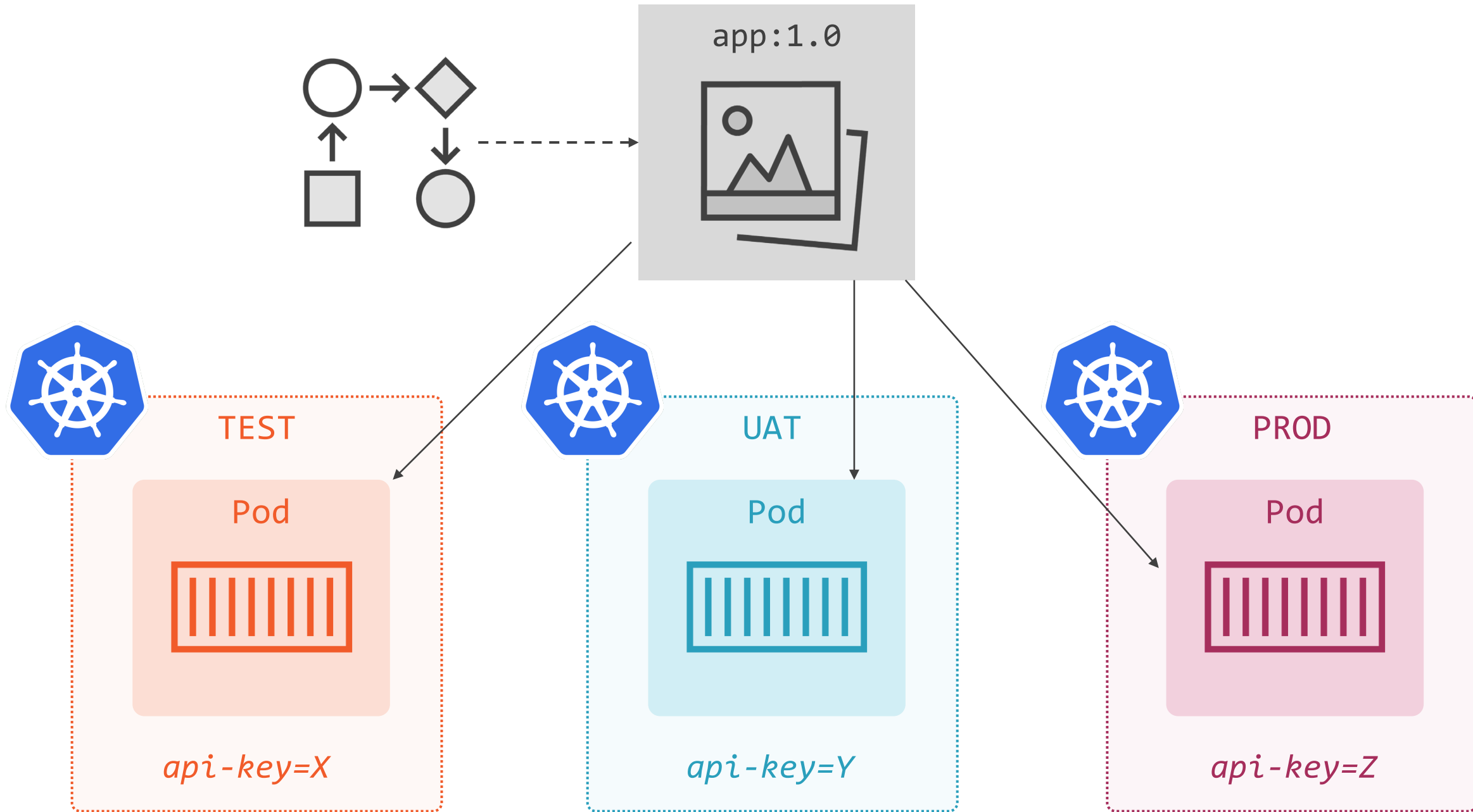
Consultant & Trainer

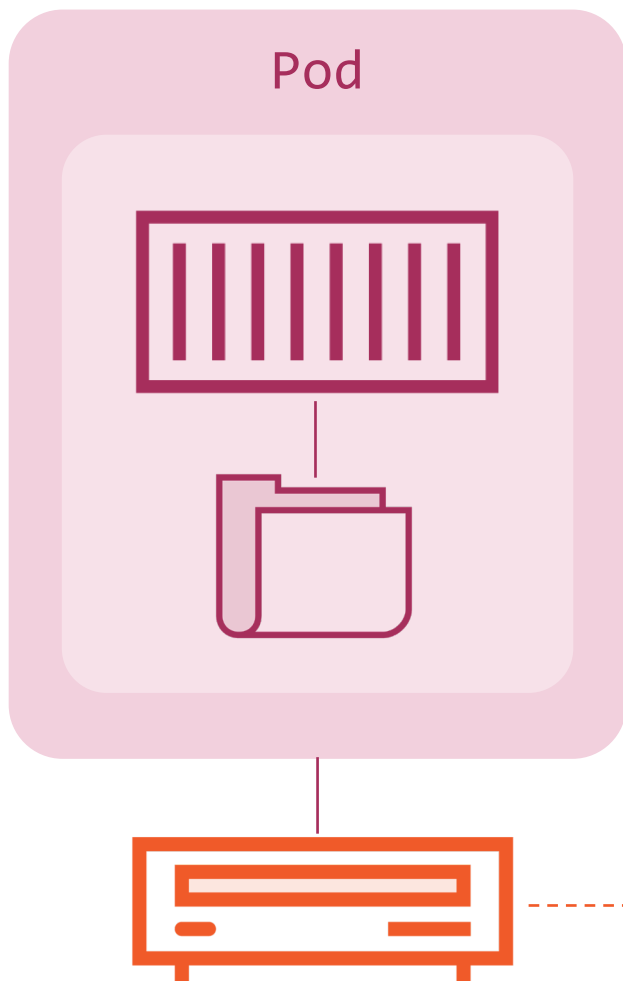
@EltonStoneman blog.sixeyed.com











```
apiVersion: v1
kind: Pod
metadata:
  name: whoami
spec:
  containers:
    - name: app
      image: wiredbrain/web
      volumeMounts:
        - name: config
          mountPath: "/conf"
          readOnly: true
  volumes:
    - name: config
      configMap:
        name: web-config
```



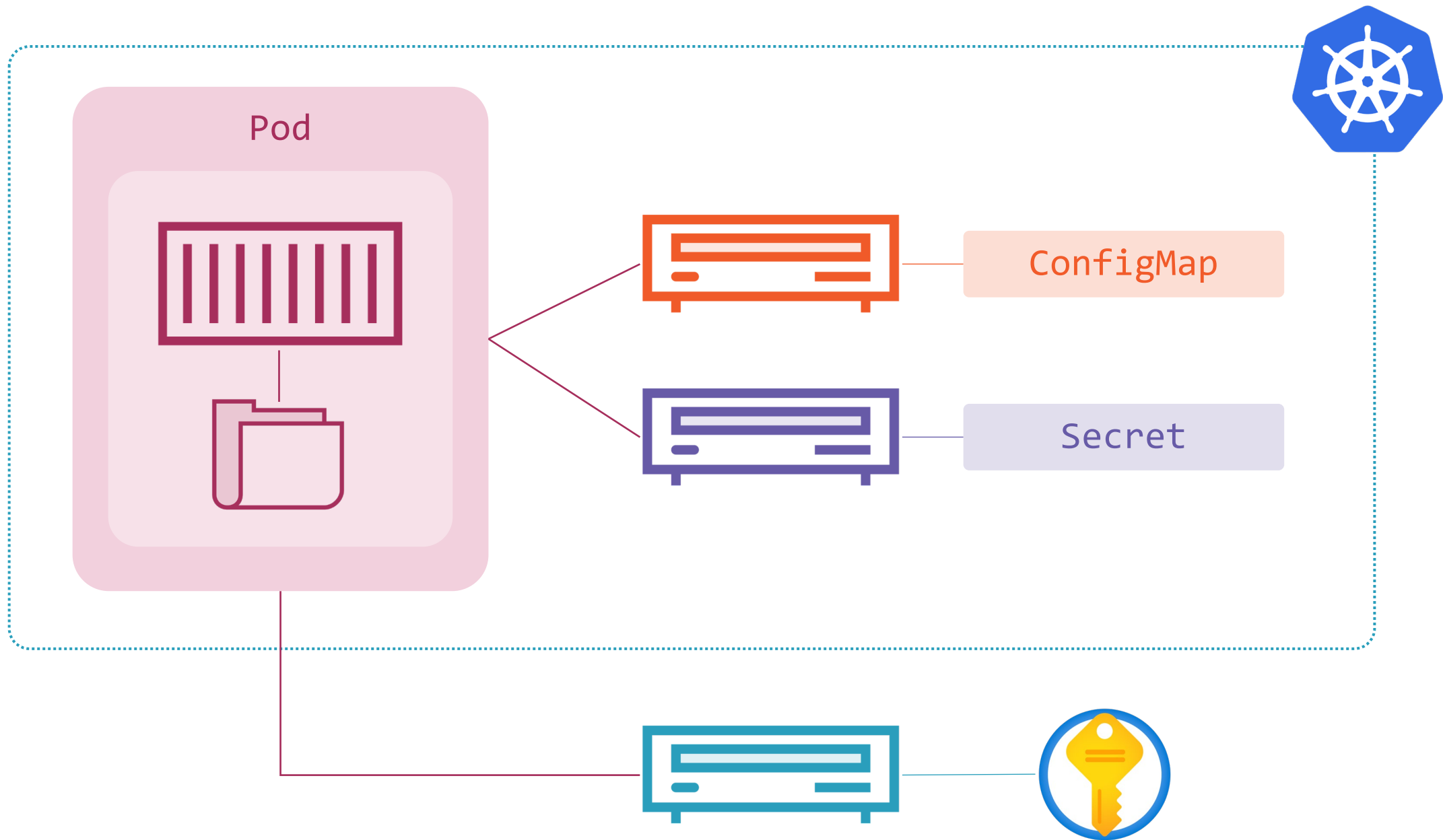
volumeMounts:

- name: config
mountPath: "/config"
- name: secret
mountPath: "/secret"
- name: keyvault
mountPath: "/creds"

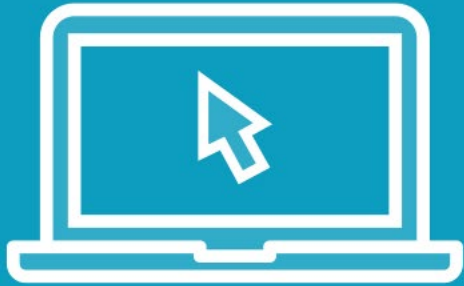


volumes:

- name: config
configMap:
 name: web-config
- name: secret
secret:
 secretName: web-urls
- name: keyvault
csi:
 driver: secrets-store.csi.k8s.io
 readOnly: true
 volumeAttributes:
 secretProviderClass: kv01

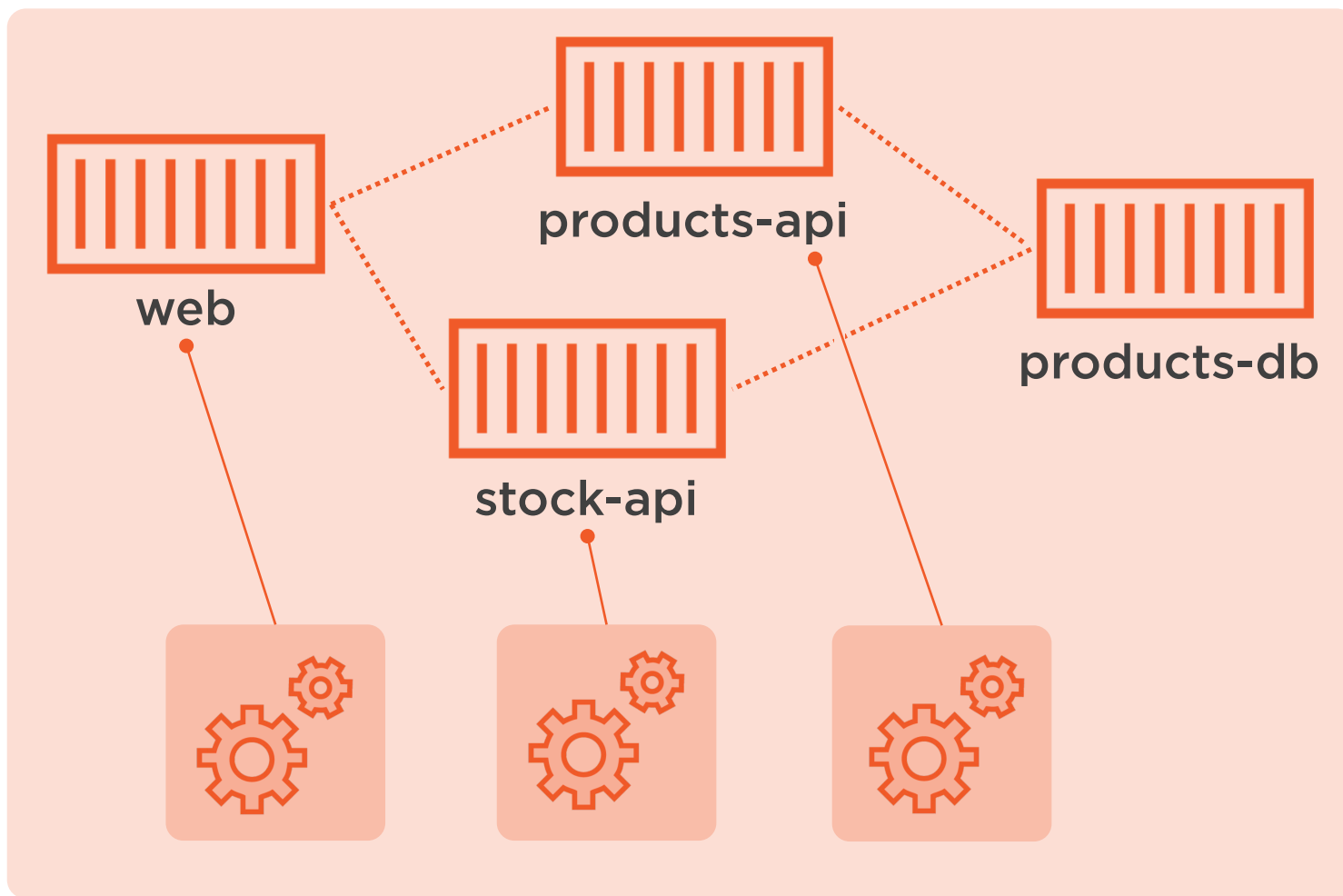
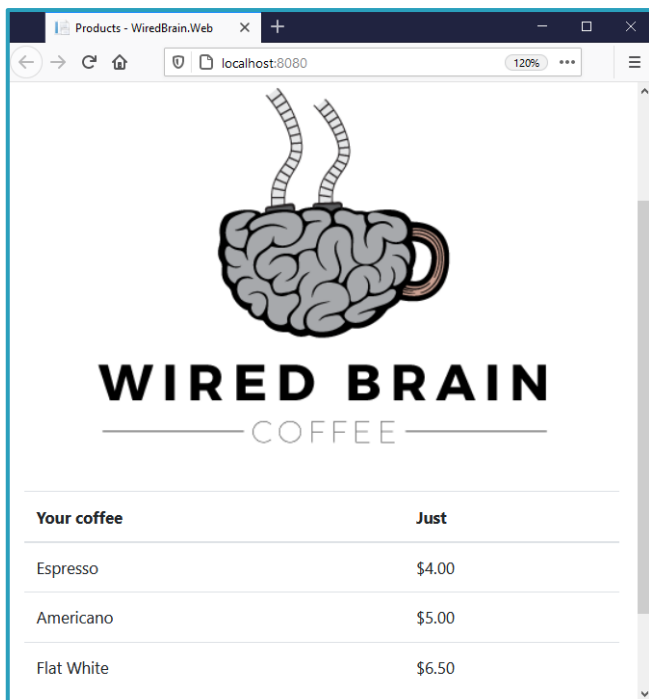


Demo



ConfigMaps in Kubernetes

- Default settings in the image
- Applying settings with ConfigMaps
- Understanding change propagation



Pod

in the Deployment template

spec:

containers:

- name: app

image: wiredbrain/products-api:22.03

volumeMounts:

- name: db-properties

mountPath: "/app/config/db"

readOnly: true

volumes:

- name: db-properties

configMap:


name: products-api-config-db

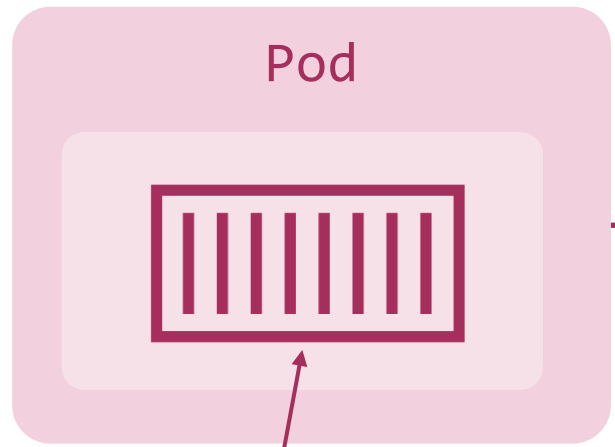
- Mount files or folders
- Project multiple volumes
- Can't project to a file

ConfigMap

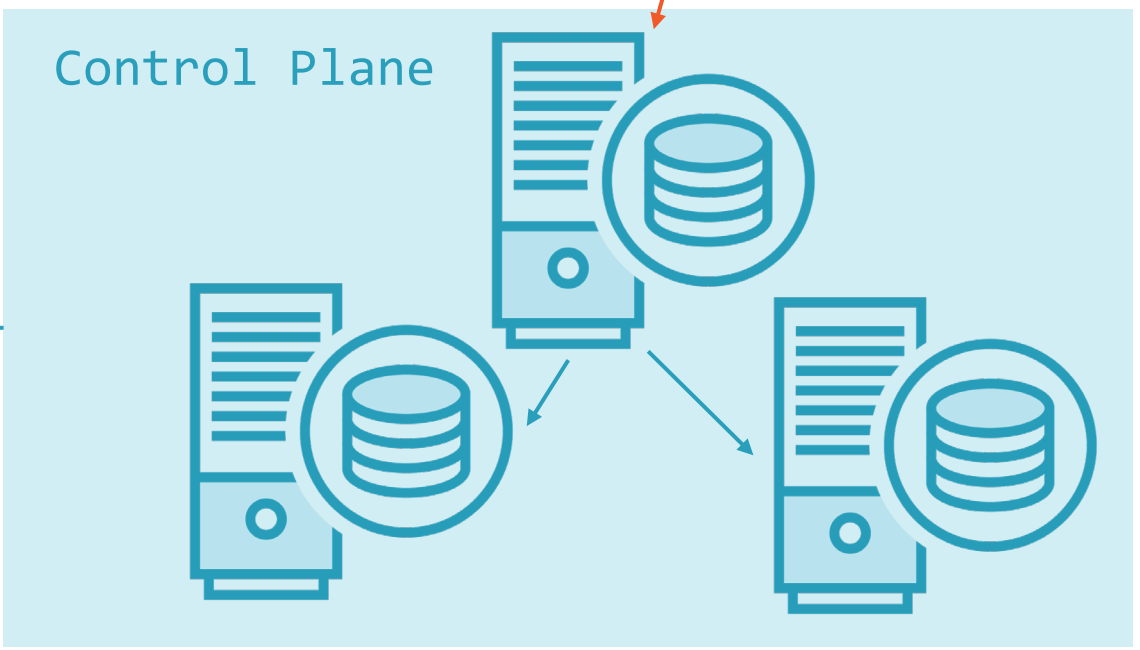
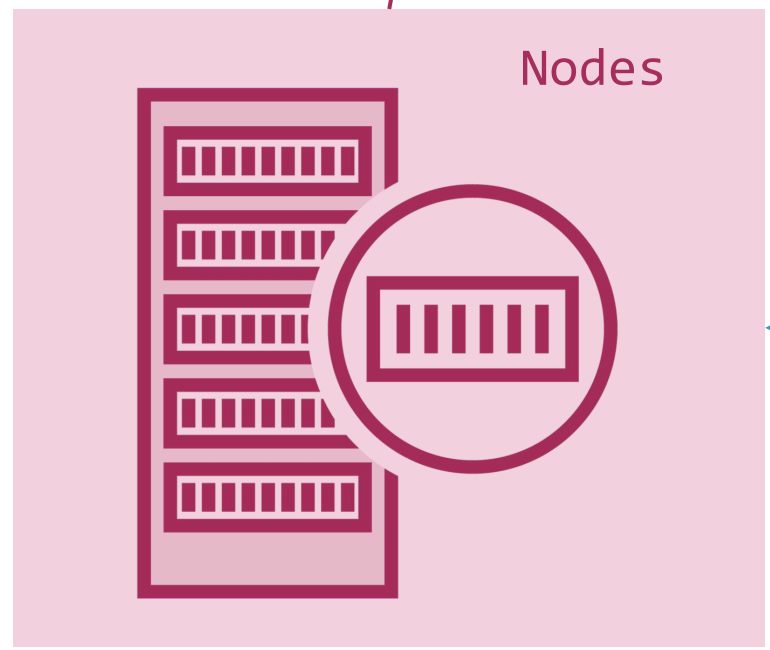
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: products-api-config-db
  labels:
    app: wiredbrain
    component: products-api
data:
  application.properties: |-
    spring.datasource.platform=postgres
    spring.datasource.username=postgres
    spring.datasource.password=wired
    spring.datasource.url=jdbc:postgresql://products-db-cl:5432/postgres
```

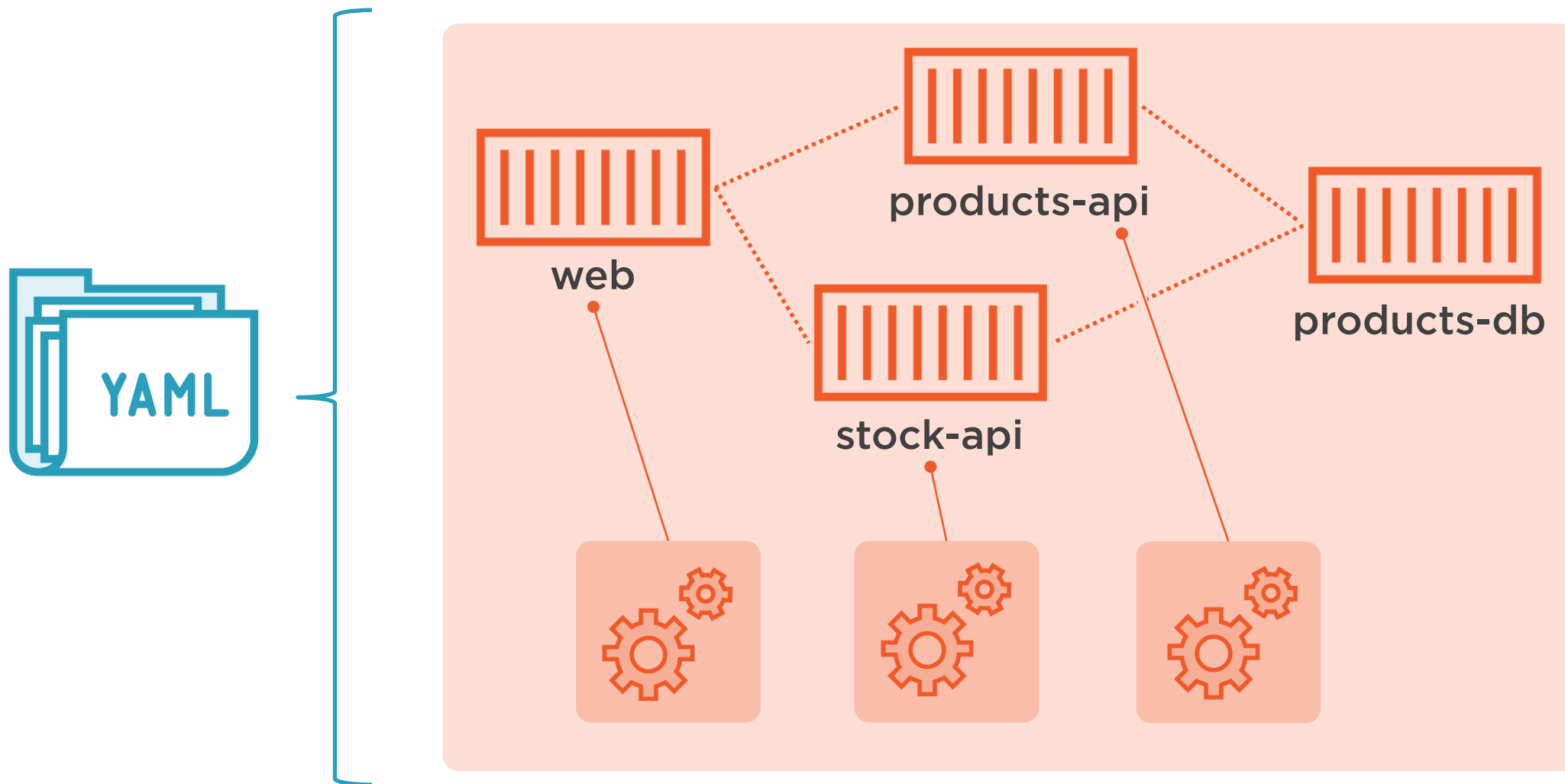
- Static
- Sensitive
- Environmental

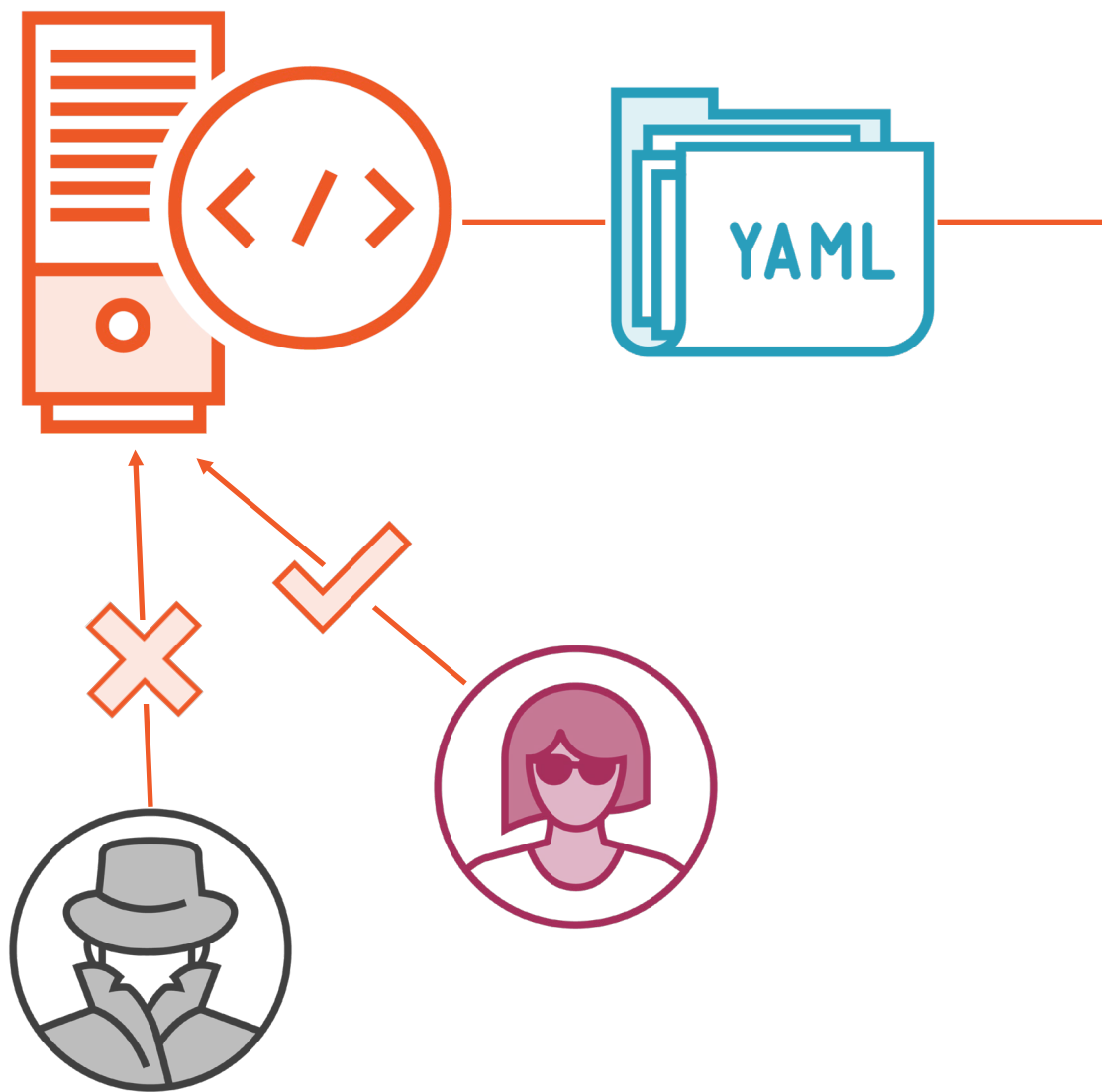




symlink

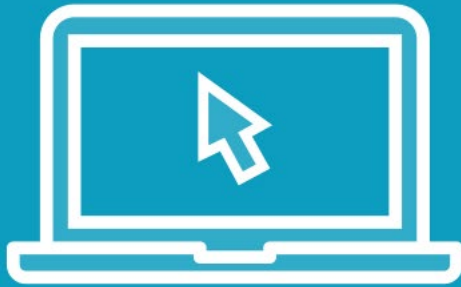






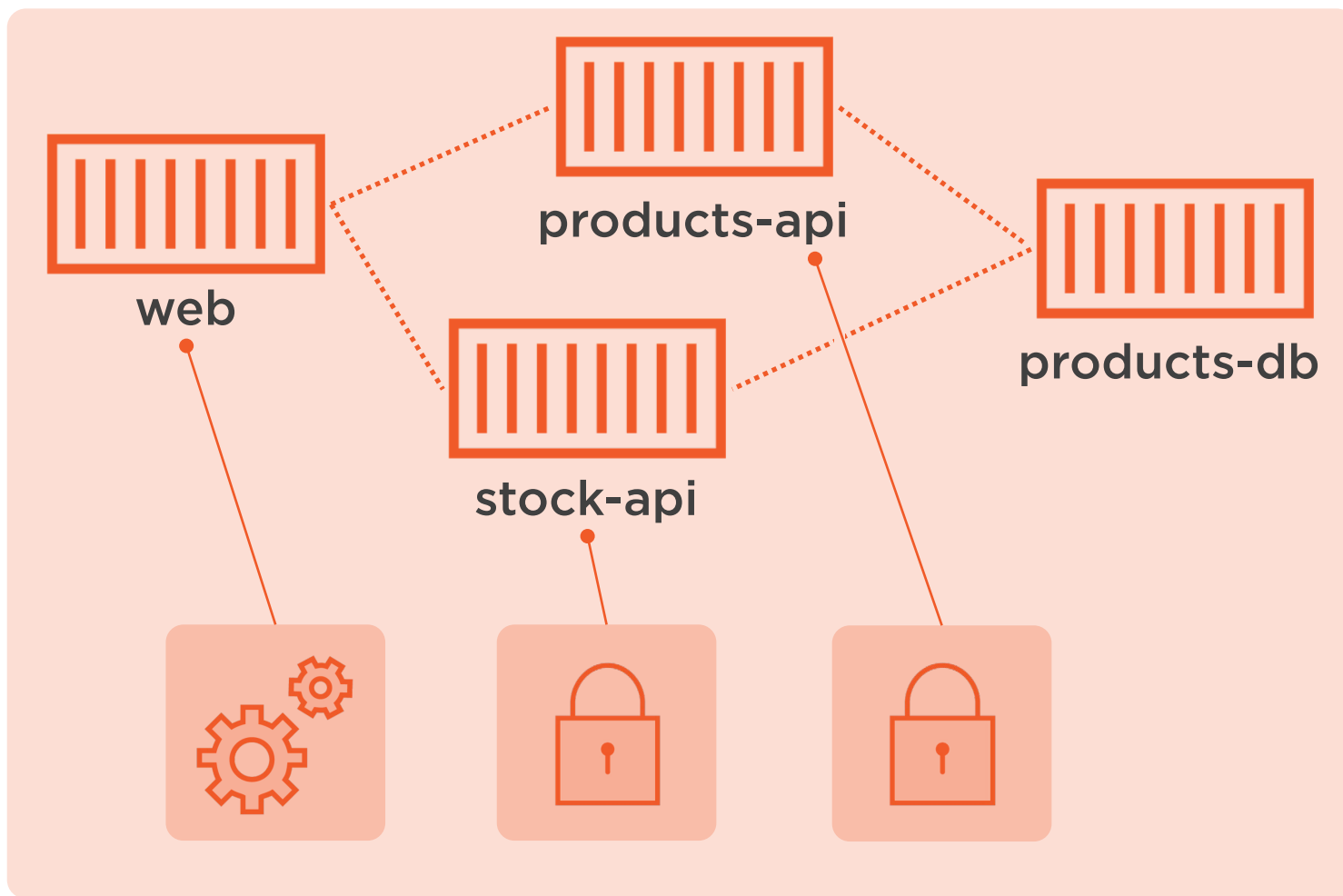
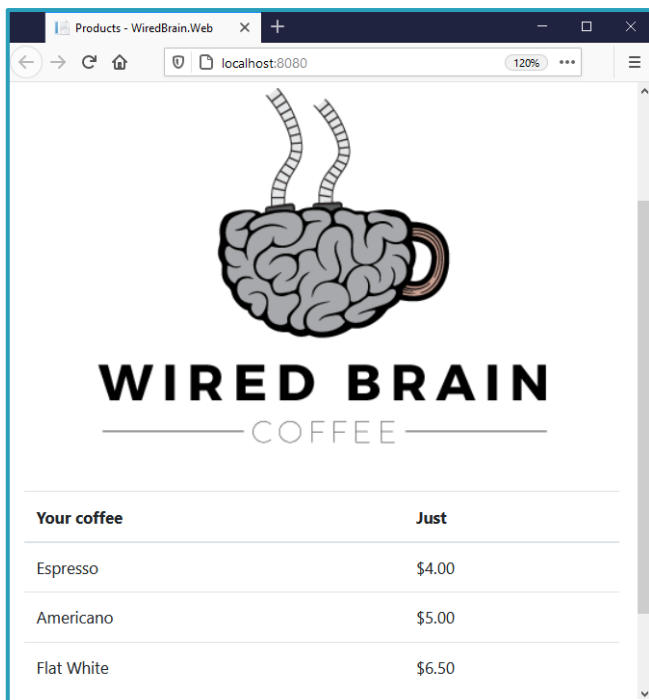
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: products-api-config-db
  labels:
    app: wiredbrain
    component: products-api
data:
  application.properties: |-
    spring.jpa.database=POSTGRESQL
    spring.datasource.platform=postgres
    spring.datasource.url=...
    spring.datasource.username=postgres
    spring.datasource.password=wired
```

Demo



Secrets in Kubernetes

- **Modelling Secrets**
- **Managing data in Secrets**
- **Securing access with RBAC**



Pod

in the Deployment template

spec:

containers:

- name: app
image: wiredbrain/products-api:22.03

volumeMounts:

- name: db-properties
mountPath: "/app/config/db"
readOnly: true

volumes:

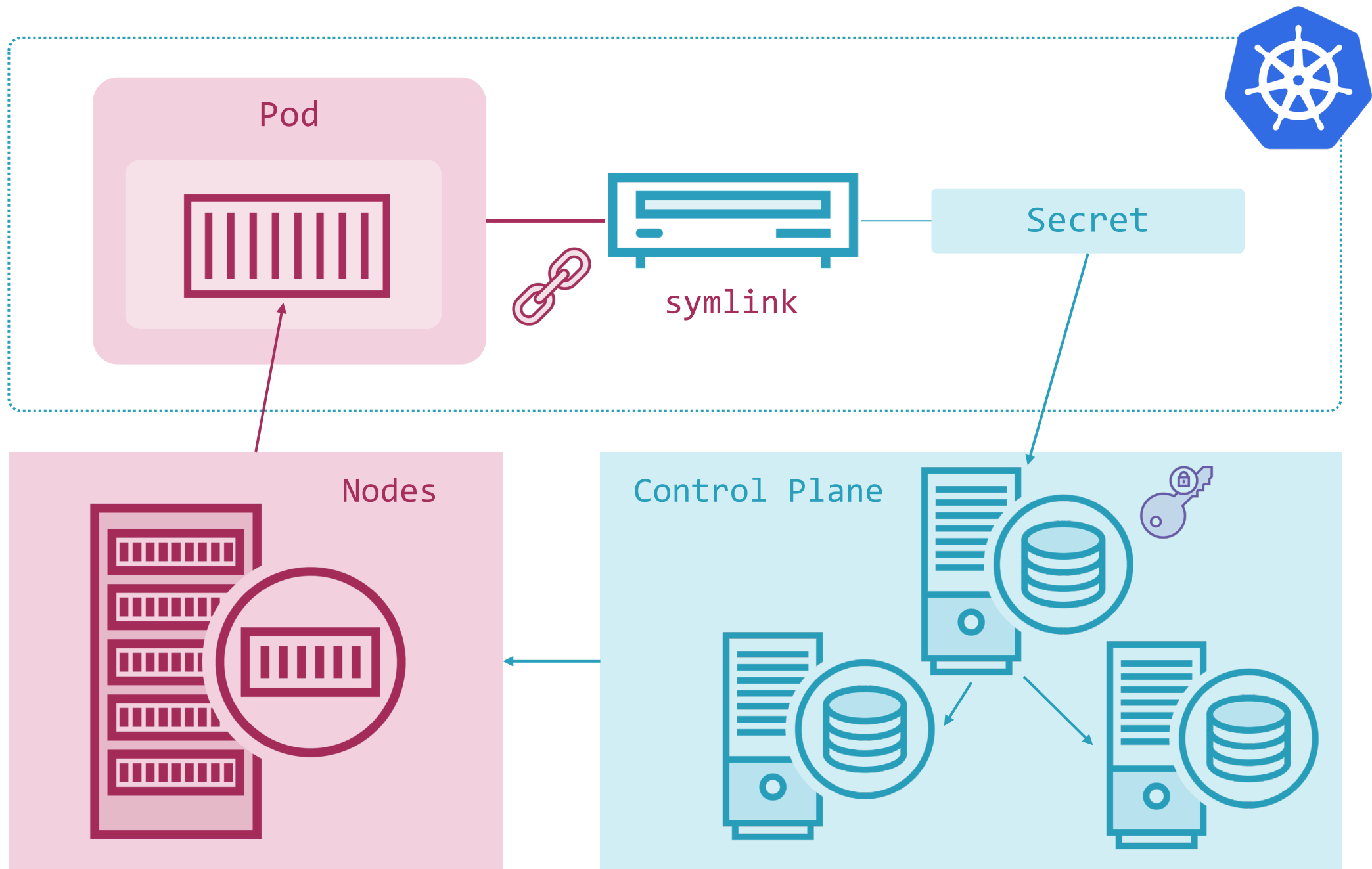
- name: db-properties
secret:
secretName: products-api-config-db

- Mount files or folders
- Project multiple volumes
- Can't project to a file

```
kubectl create secret  
    generic  
    products-api-config-db  
    --from-file application.properties
```

Creating Secrets with Kubectl

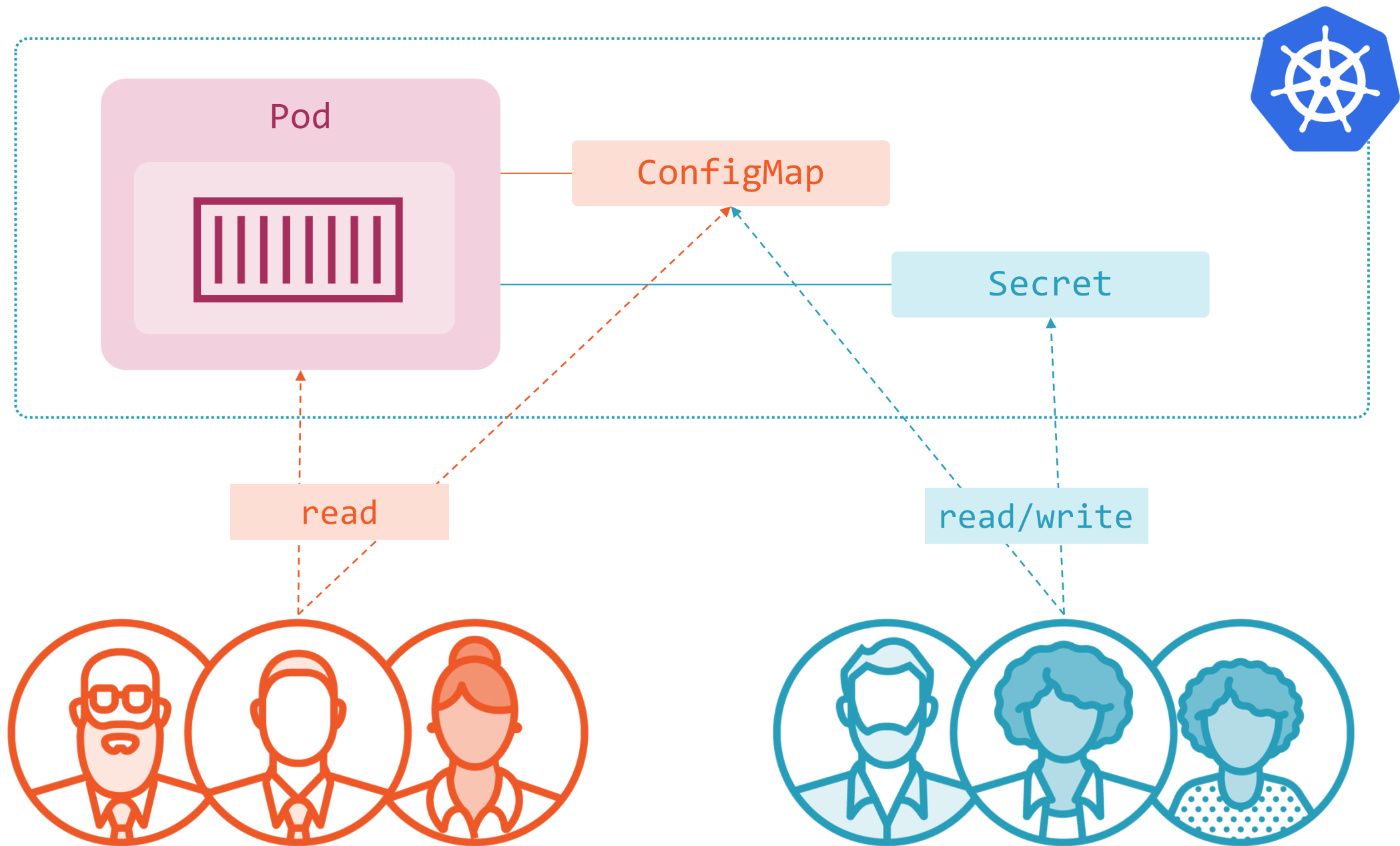
Secrets can also be modelled in YAML



```
kubectl get secret  
products-api-config-db  
-o go-template='{{index  
    .data \"application.properties\"  
    | base64decode}}'
```

Secrets are not... secret

Users with access can read plain-text



RBAC

```
apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRole
```

```
metadata:
```

```
  name: product-tester
```

```
rules:
```

```
- apiGroups: [""]
```

```
  resources: ["pods", "pods/log", "services"]
```

```
  verbs: ["get", "list"]
```

```
- apiGroups: [""]
```

```
  resources: ["pods/exec"]
```

```
  verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
```

- Grant-only
- No Secret access
- Beware exec

```
kubectl exec
```

```
products-api-pod
```

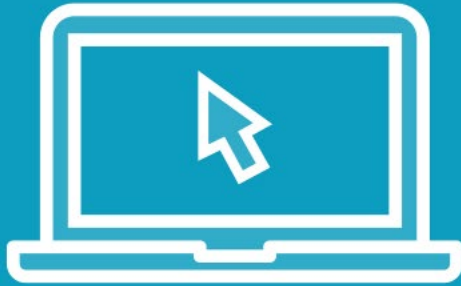
```
--
```

```
cat /app/config/db/application.properties
```

Consider filesystem access

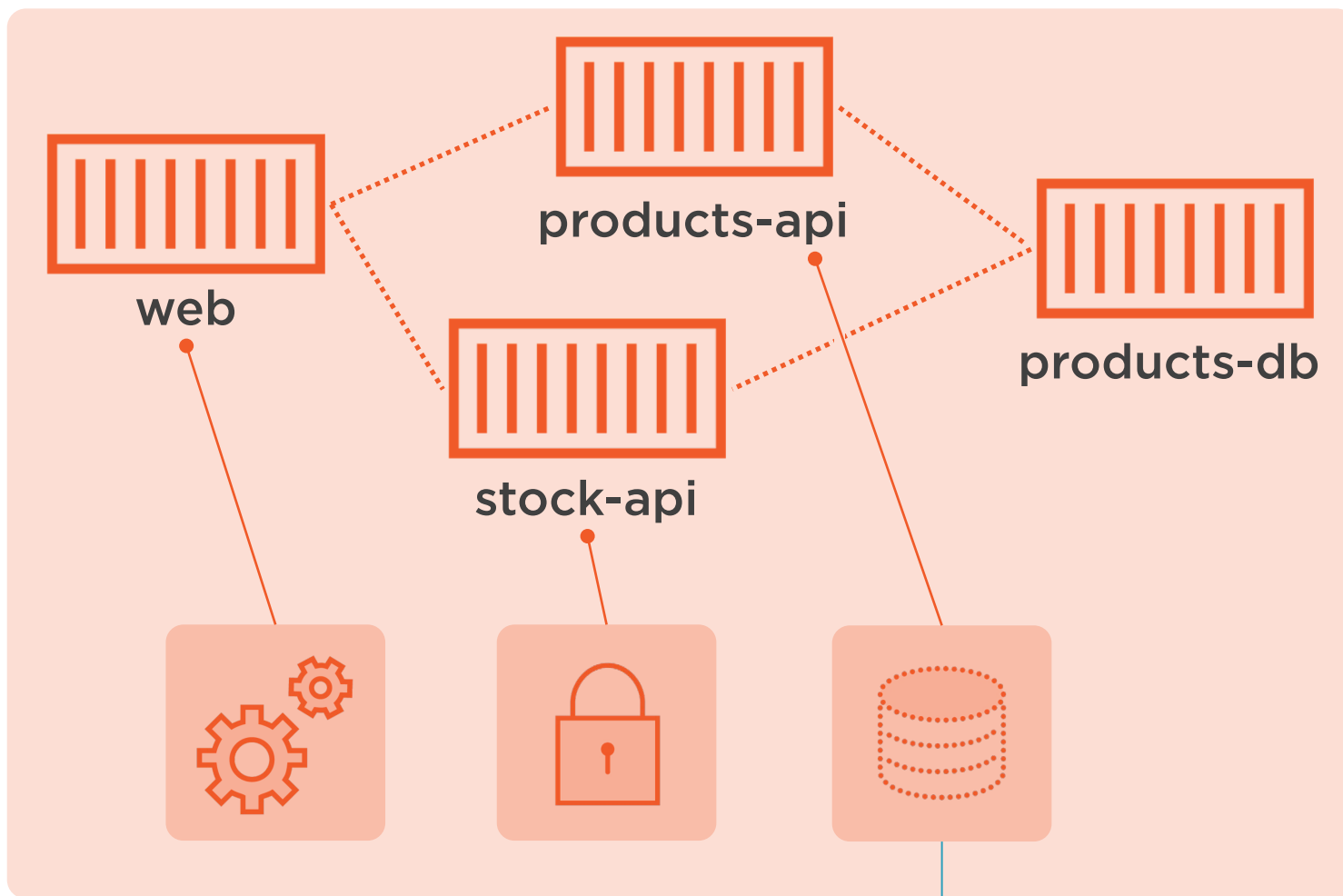
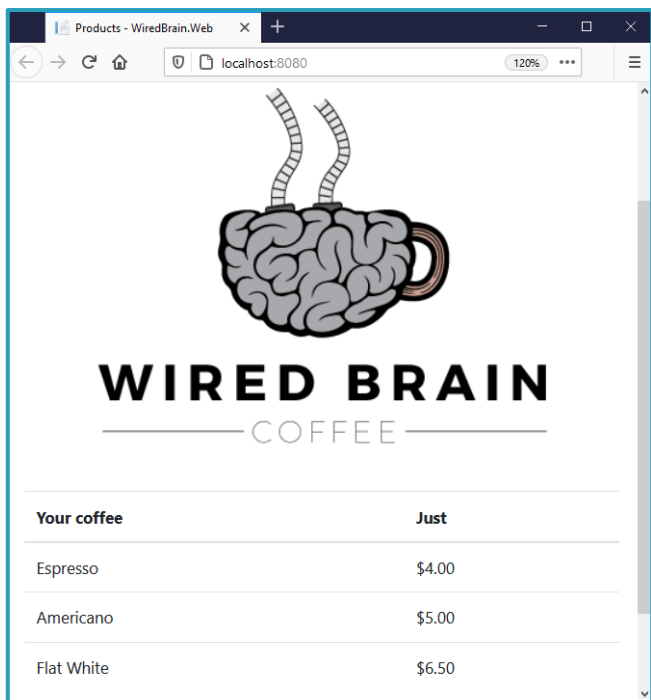
Users with exec or create Pod permissions can read Secrets

Demo



Secret Store CSI Driver

- External secure storage
- Kubernetes SecretProviderClass CRD
- Mounting external secrets into Pods



SecretProviderClass

```
apiVersion: secrets-store.csi.x-k8s.io/v1
```

```
kind: SecretProviderClass
```

```
metadata:
```

```
  name: keyvault-ps-kv01
```

```
spec:
```

```
  provider: azure
```

```
  parameters:
```

```
    useVMManagedIdentity: "true"
```

```
    userAssignedIdentityID: identity-id
```

```
    tenantId: tenant-id
```

```
  keyvaultName: ps-kv01
```

```
  objects: |
```

```
    array:
```

```
      - |
```

```
        objectName: products-api-config
```

```
        objectType: secret
```

```
        objectAlias: application.properties
```

- Multiple providers
- Clouds & Vault
- SPC is provider-specific

Pod

in the Pod spec

containers:

- name: app
image: wiredbrain/products-api:22.03

volumeMounts:

- name: db-properties
mountPath: "/app/config/db"
readOnly: true

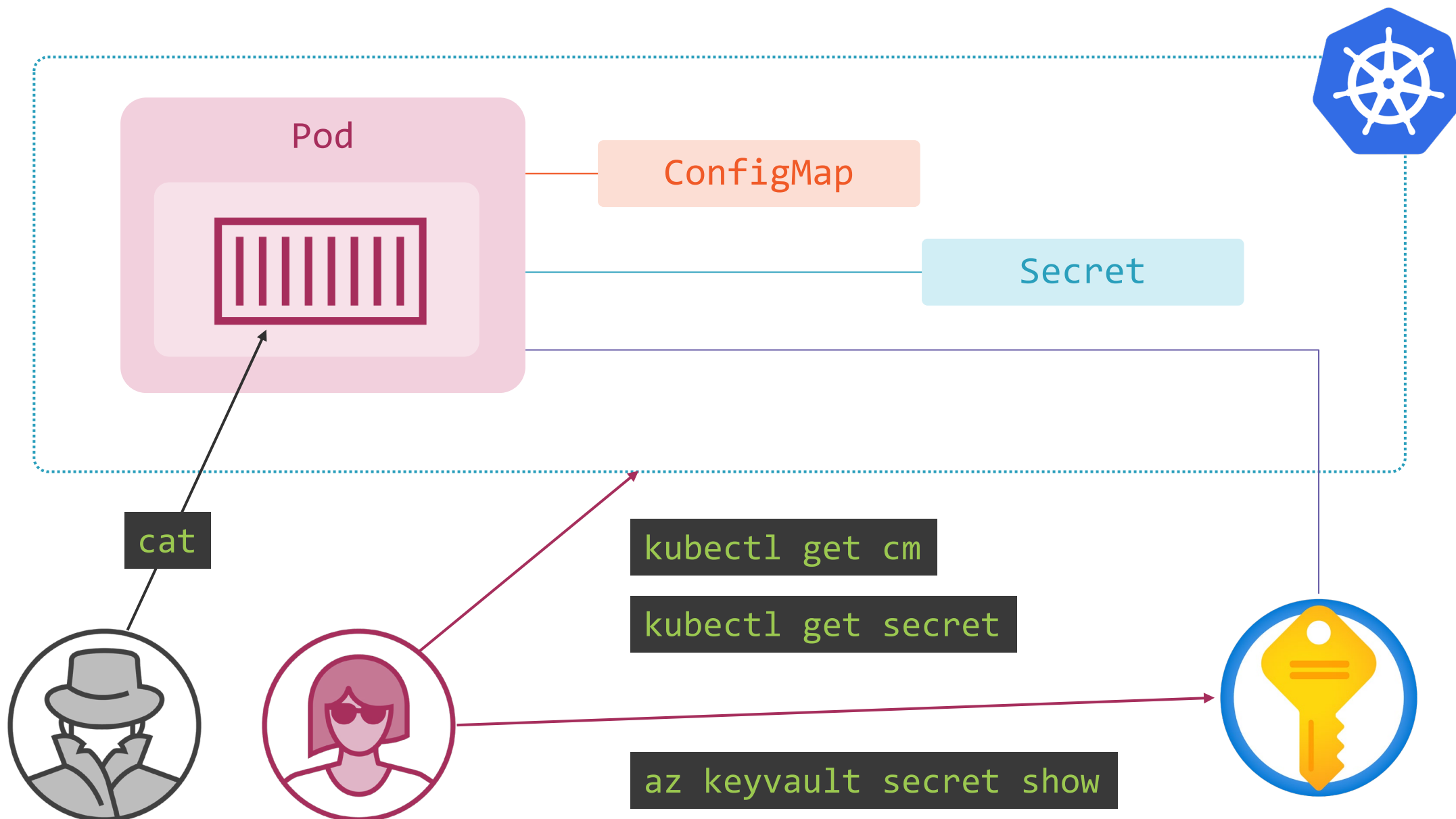
volumes:

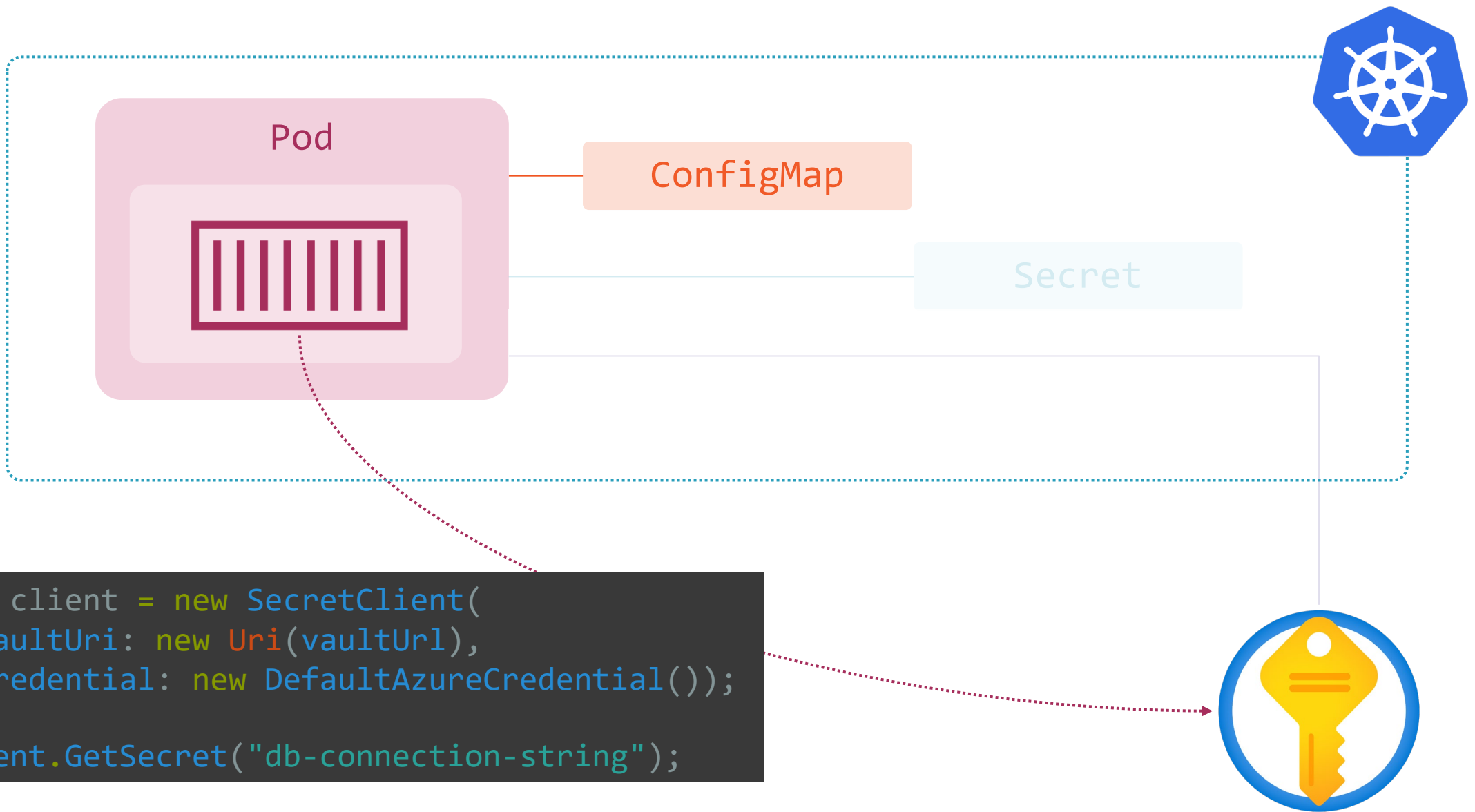
- name: db-properties

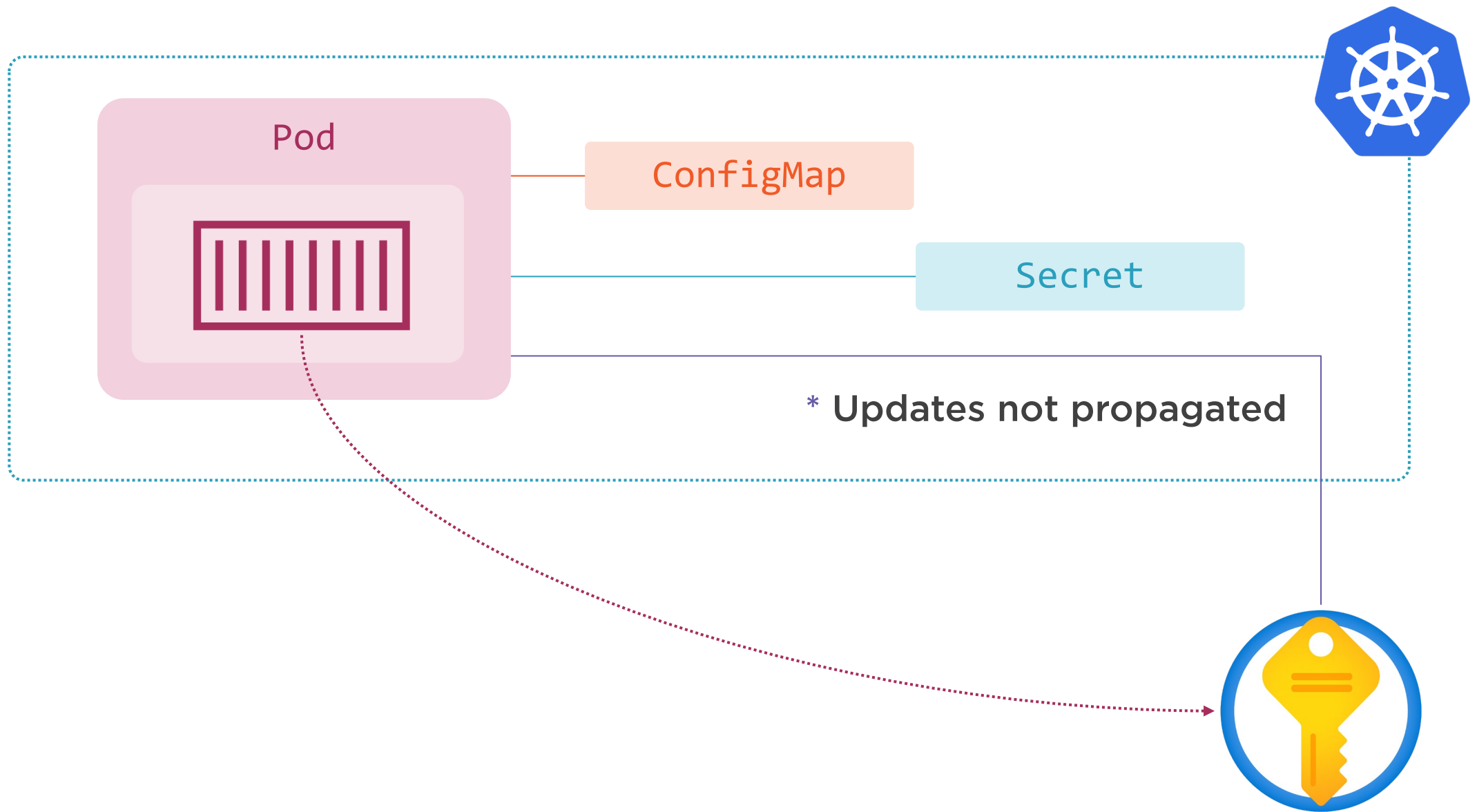
csi:

driver: secrets-store.csi.k8s.io
readOnly: true
volumeAttributes:
secretProviderClass: keyvault-ps-kv01

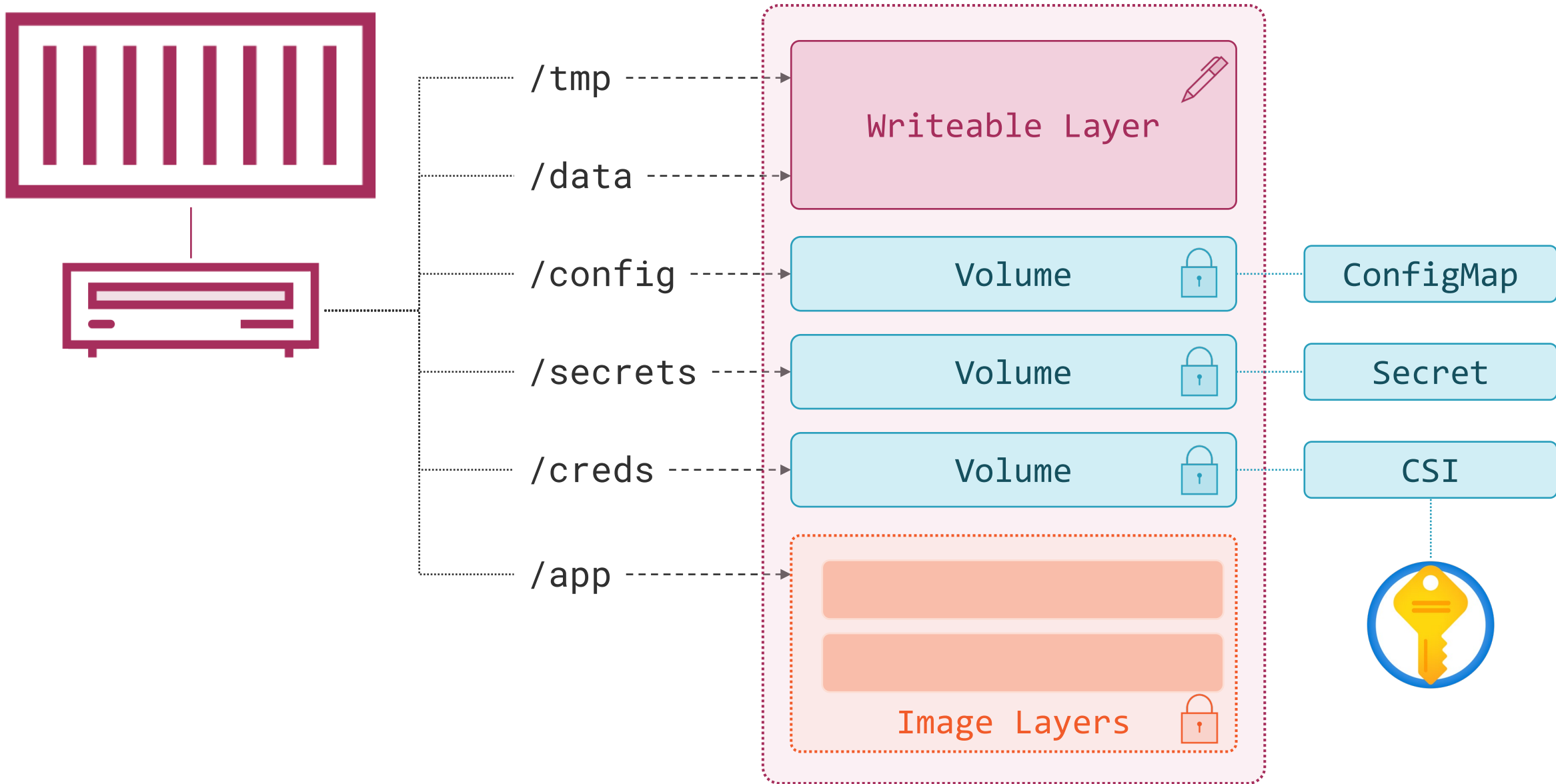
- Container Storage Interface
- Volume abstraction
- Filesystem projection







Module Summary





volumeMounts:

- name: config
mountPath: "/config"
- name: secret
mountPath: "/secret"
- name: keyvault
mountPath: "/creds"

volumes:

- name: config
configMap:
 name: web-config
- name: secret
secret:
 secretName: web-urls
- name: keyvault
csi:
 driver: secrets-store.csi.k8s.io
 readOnly: true
 volumeAttributes:
 secretProviderClass: kv01



Module Summary



VolumeMounts

- Load volumes into directories
- Files surfaced as symlinks

ConfigMaps

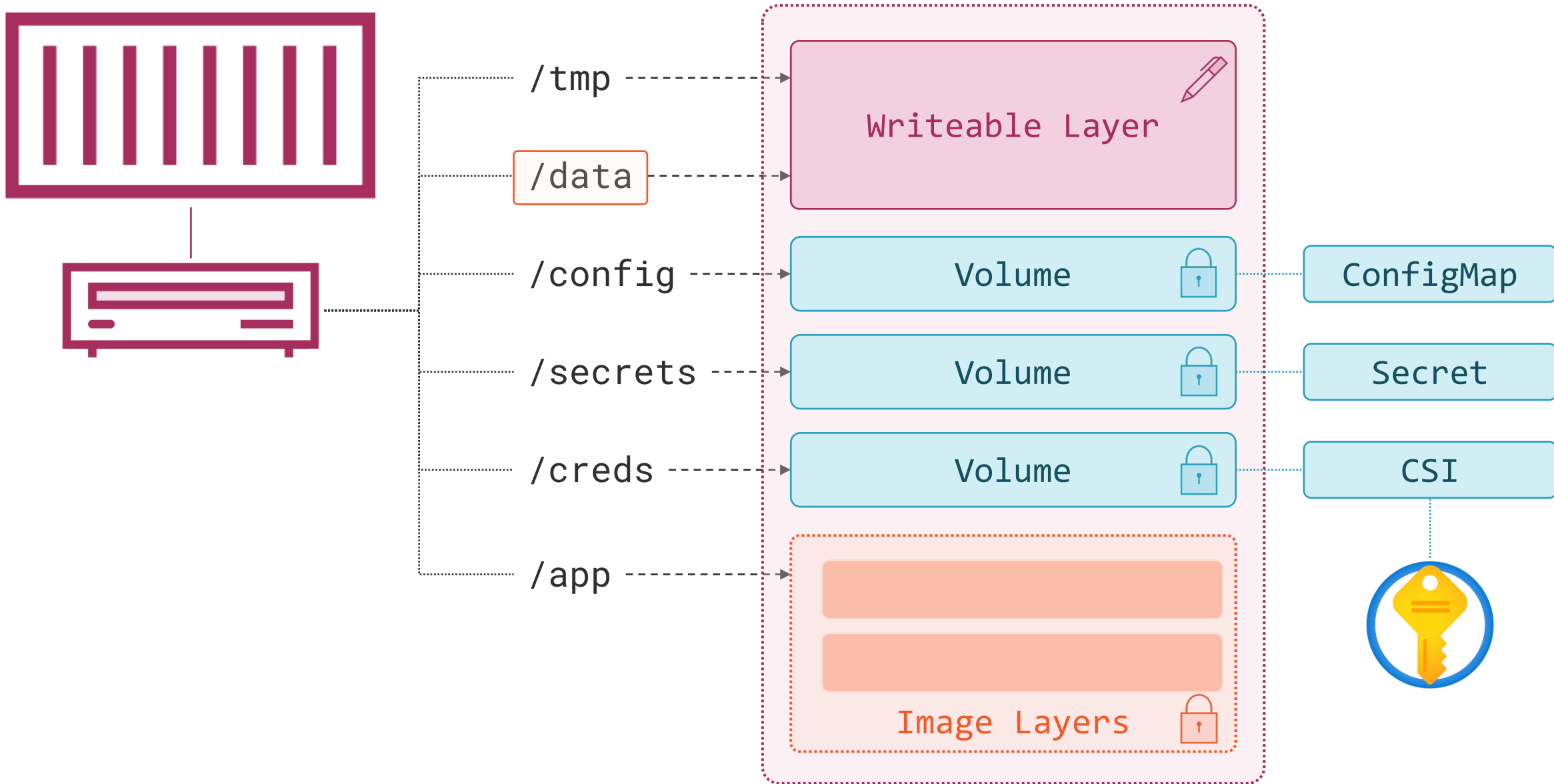
- Small data items
- Stored & accessed as plain text

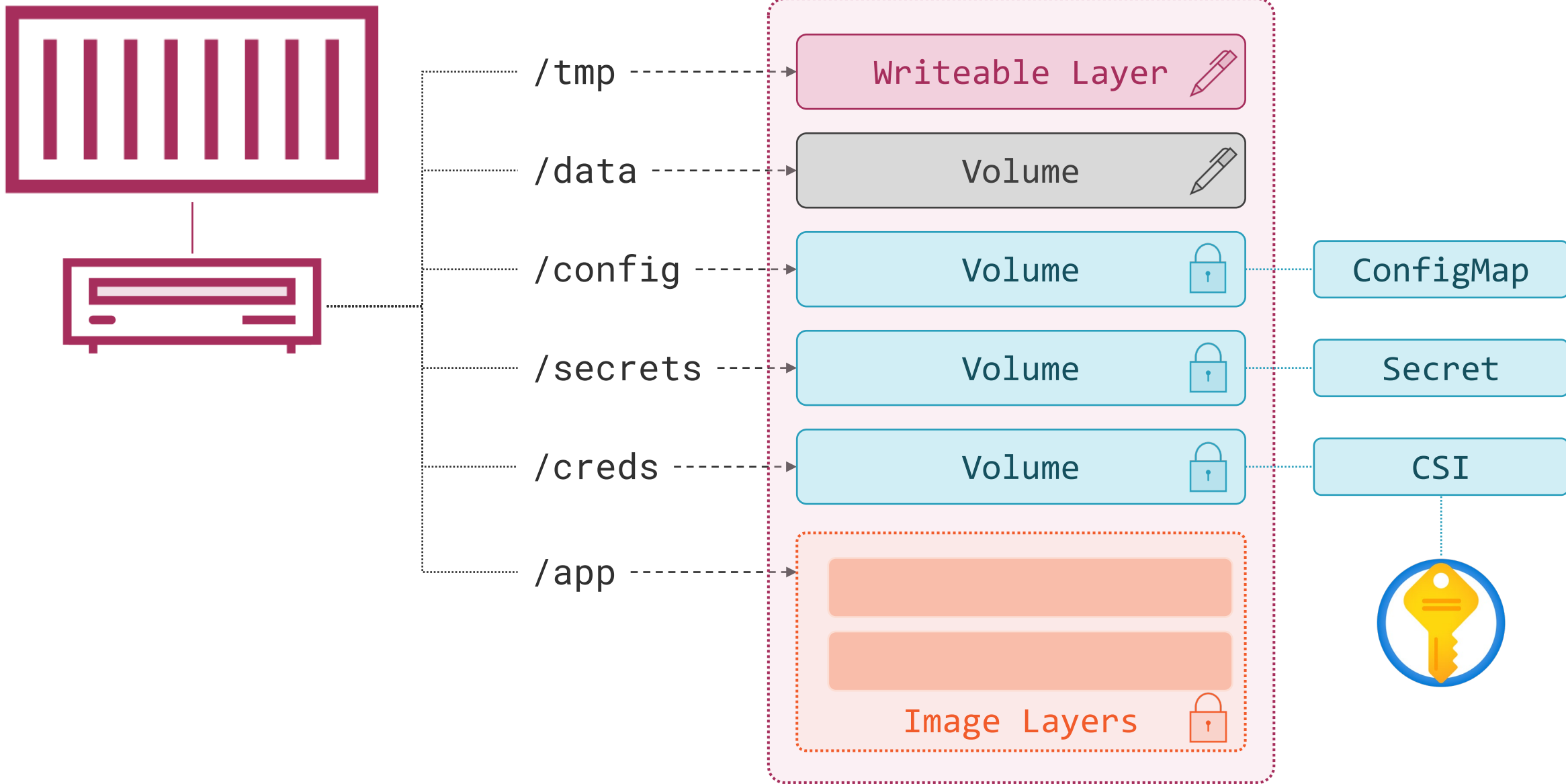
Secrets

- Separately permissioned
- Stored & accessed as Base64

Secrets Store CSI Driver

- Remote secure store
- Clouds & Vault
- Updates not propagated





Up Next:

Persisting Data in Kubernetes
