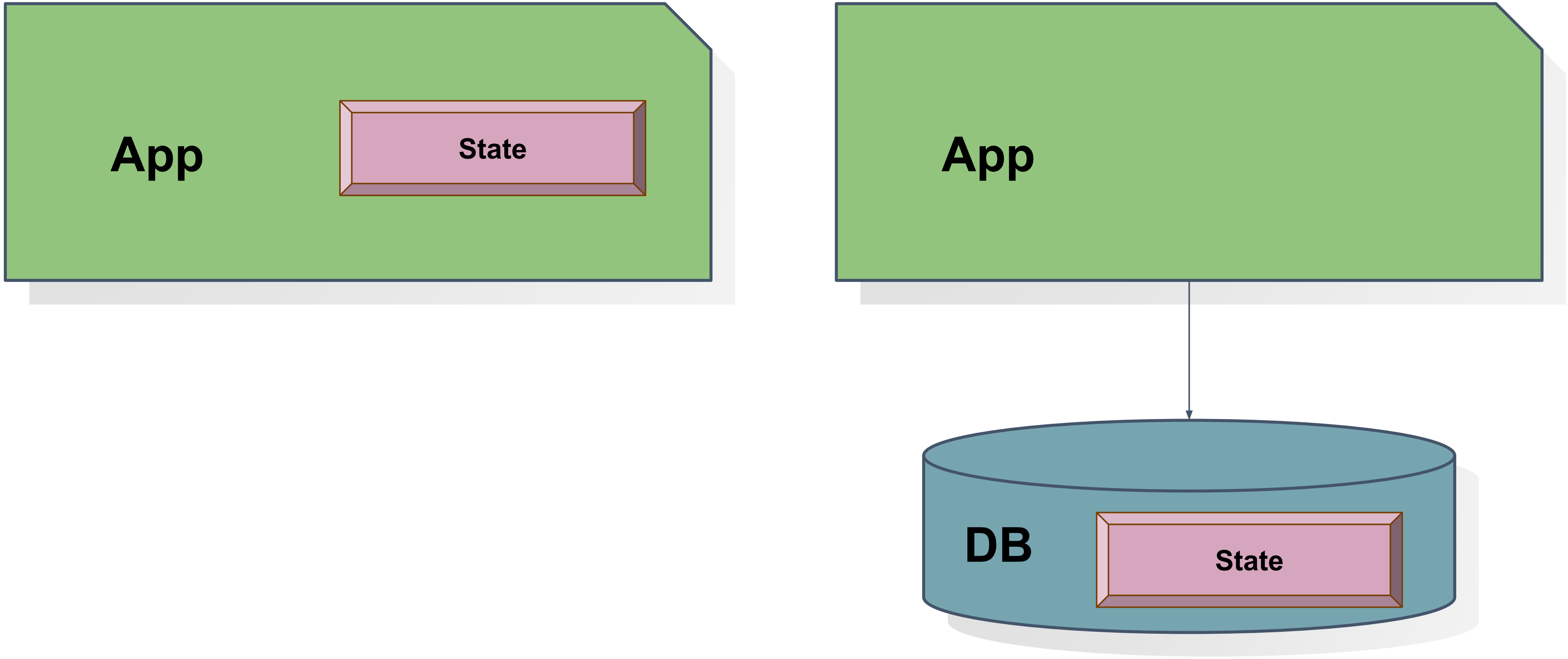
k8s: Persistent Data

Develop
Intelligence

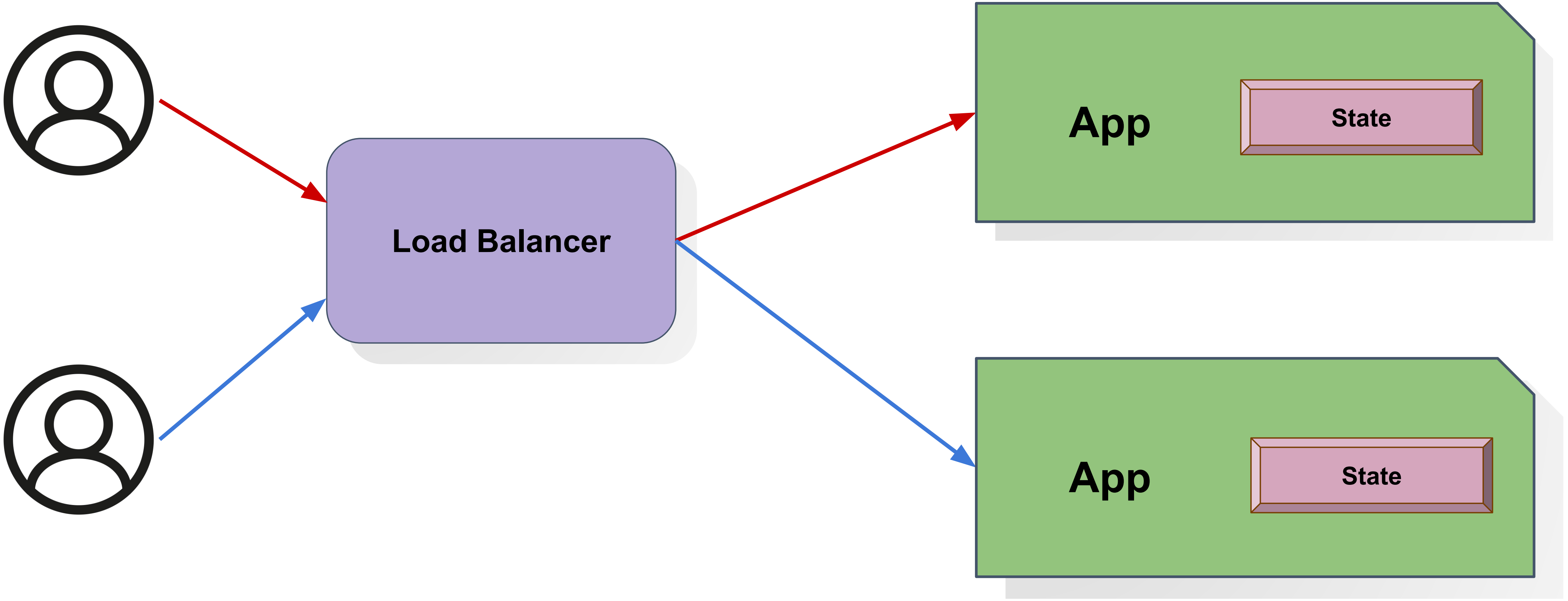
Module Outline

- **Statefulness**
- ConfigMaps & Secrets
- Persistent Volumes
- Demo

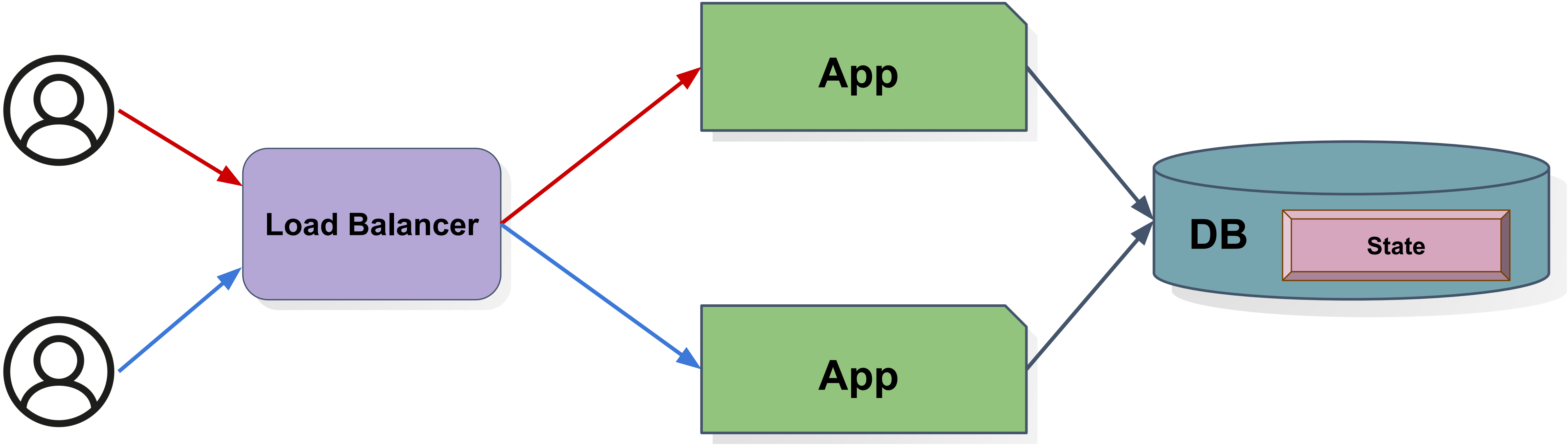
Stateful vs Stateless



Hotspots



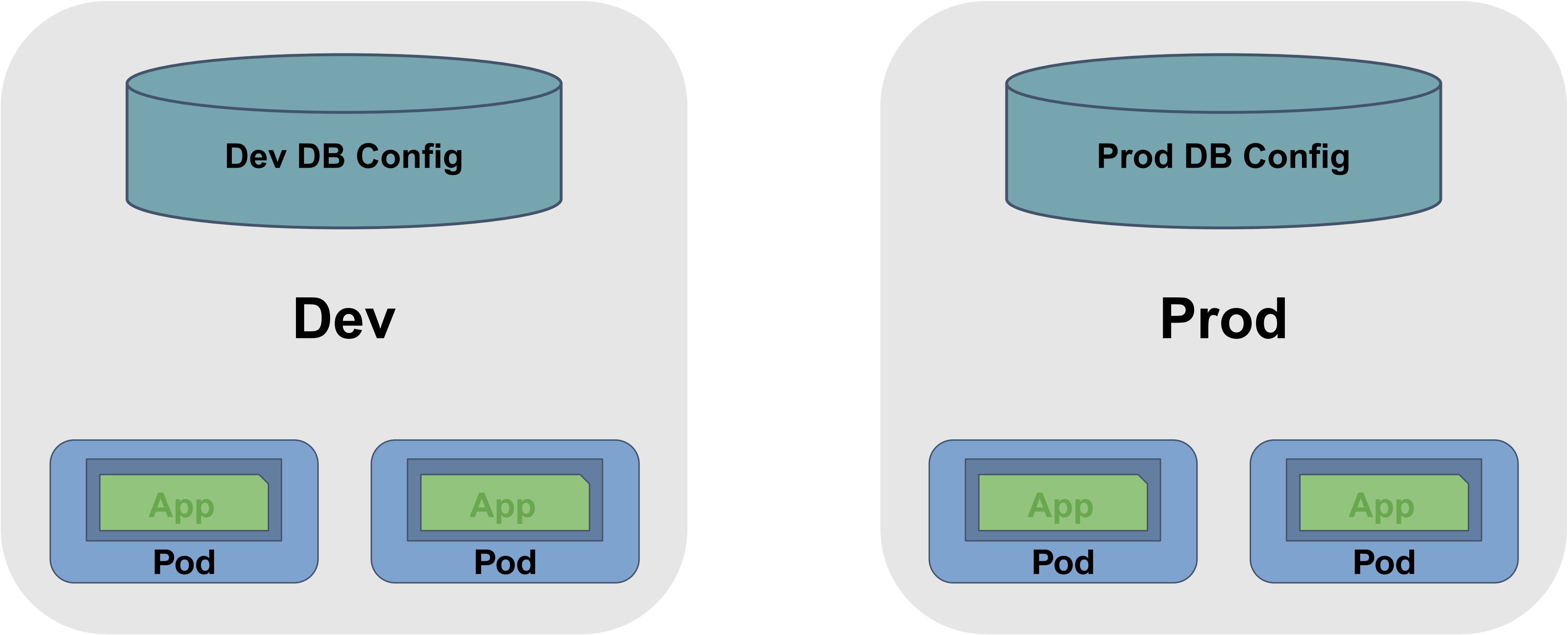
Break it out



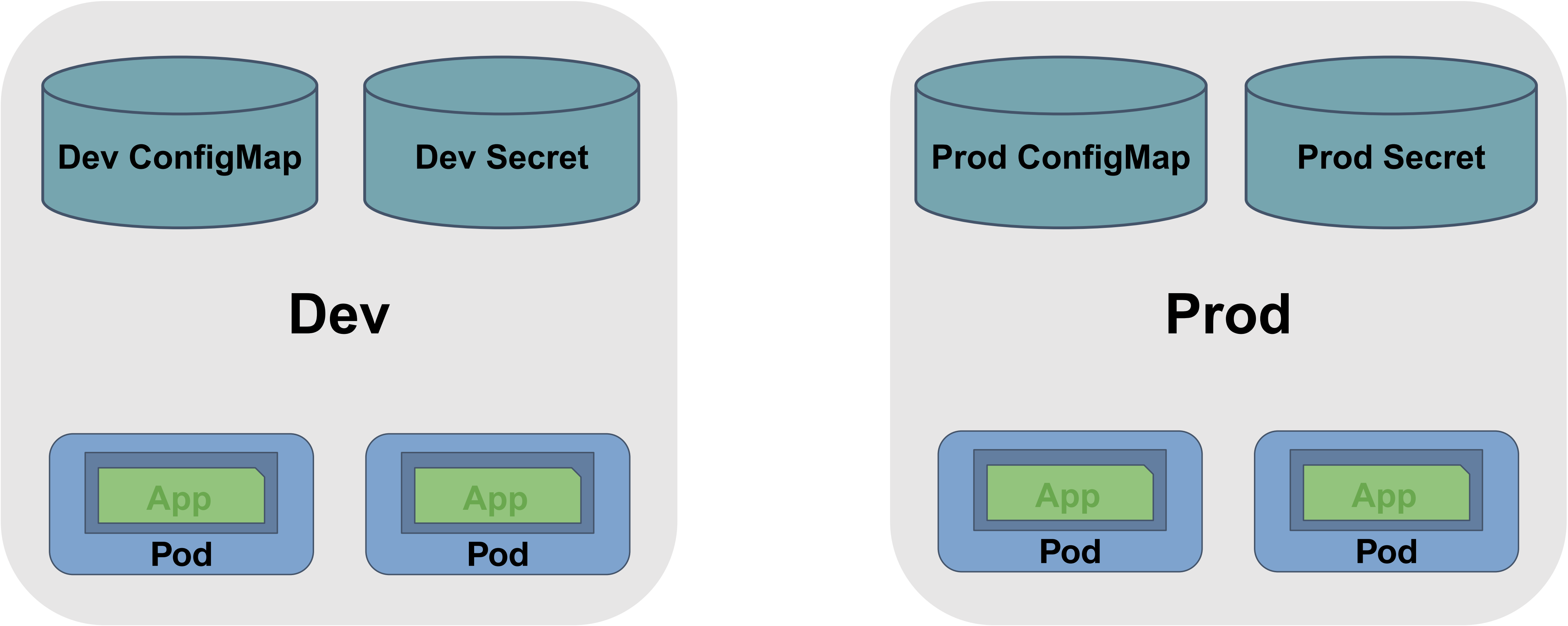
Module Outline

- Statefulness
- **ConfigMaps & Secrets**
- Persistent Volumes
- Demo

What about configs?

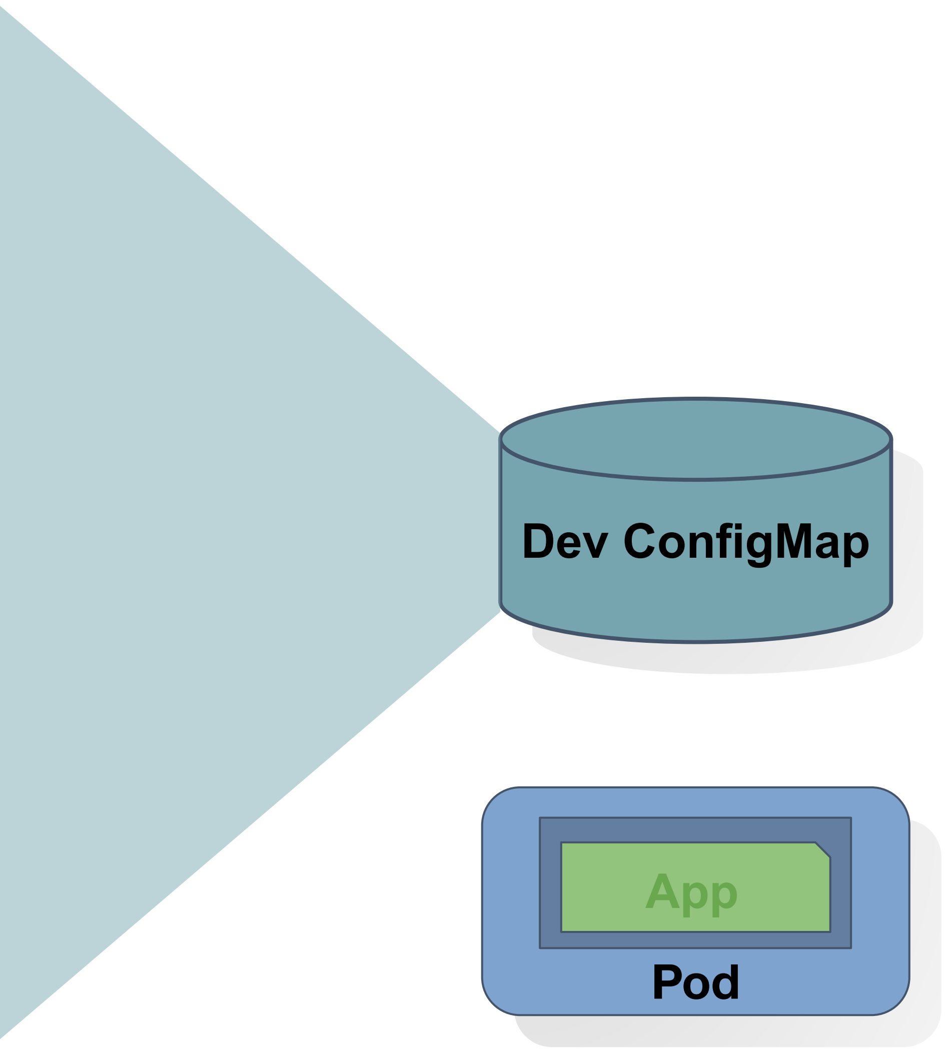


What about configs?



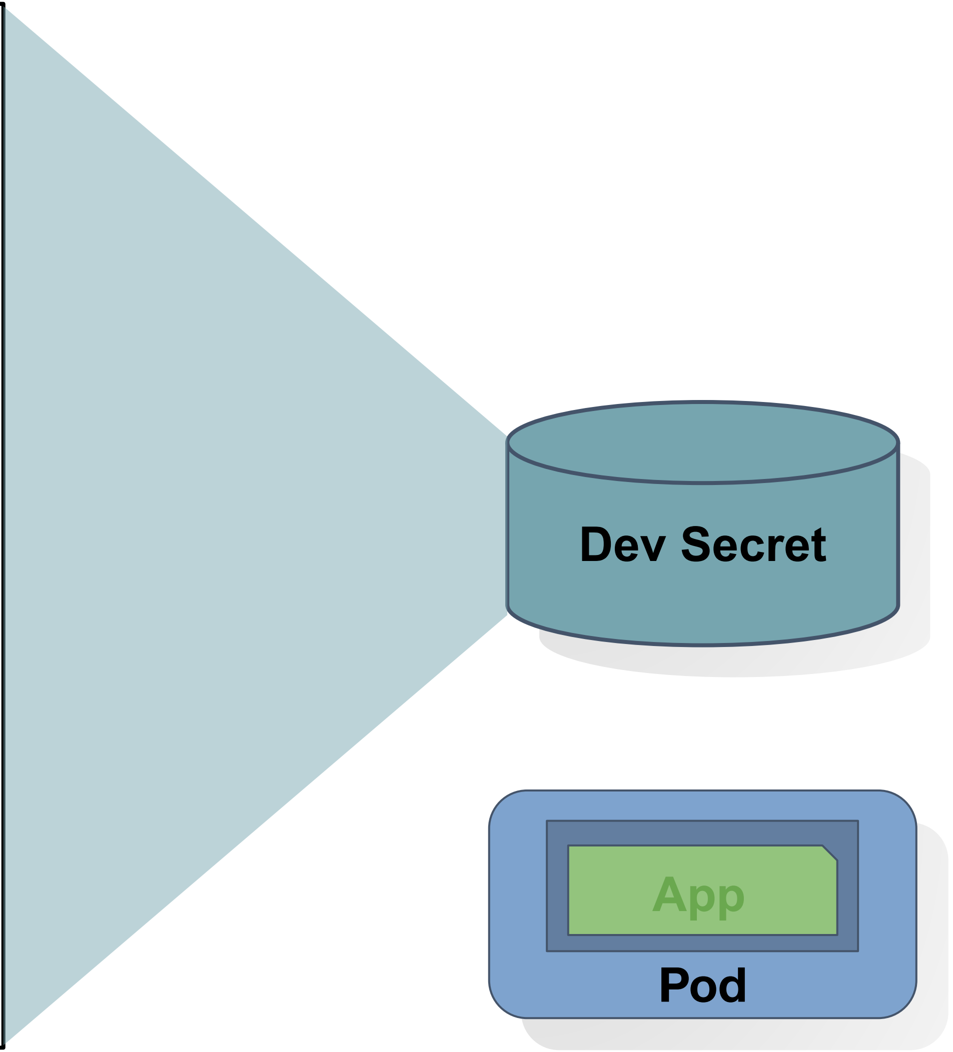
ConfigMap

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: example-configmap
data:
  database: mongodb
  database_uri: mongodb://localhost:27017
```



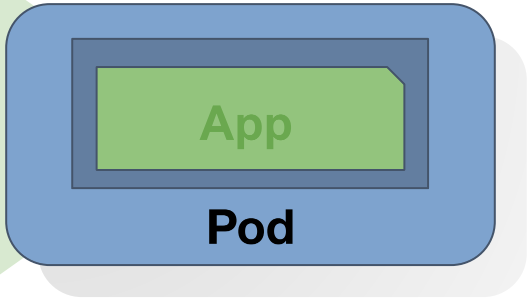
Secret

```
kind: Secret
apiVersion: v1
metadata:
  name: example-secret
data:
  db_username: YWRtaW4=
  db_password: amlraTg5M2tkam5zZDlz
```

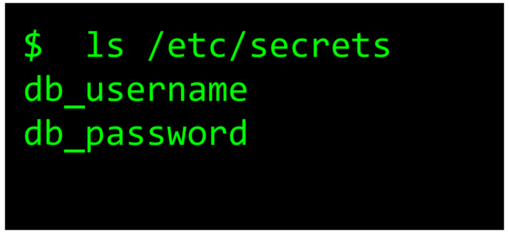


Consuming Secrets as volumes

```yaml
...
spec:
  containers:
  - name:  myapp
    image: appv1
    volumeMounts:
      - name: myvolume
        mountPath: "/etc/secrets"
        readOnly: true
  volumes:
  - name: myvolume
    secret:
      secretName: example-secret
```



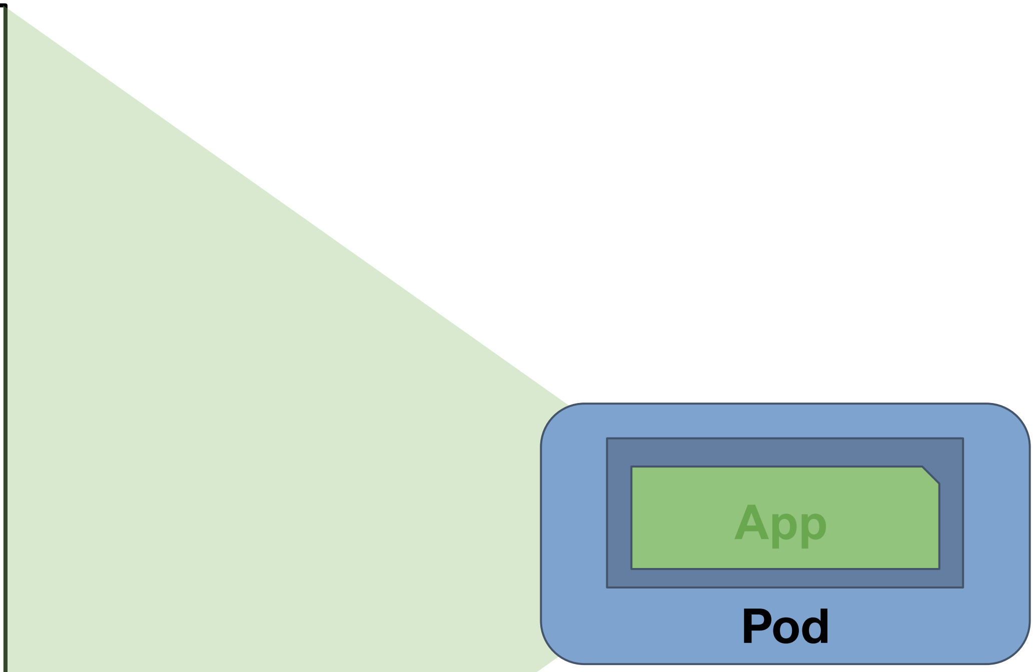
From within container:



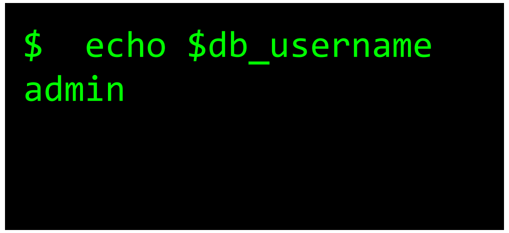
```
$  ls /etc/secrets
db_username
db_password
```

Consuming Secrets as volumes

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
spec:
  containers:
  - name:  myapp
    image: appv1
    envFrom:
      - secretKeyRef:
        name: example-secret
...
```



From within container:



```
$  echo $db_username
admin
```

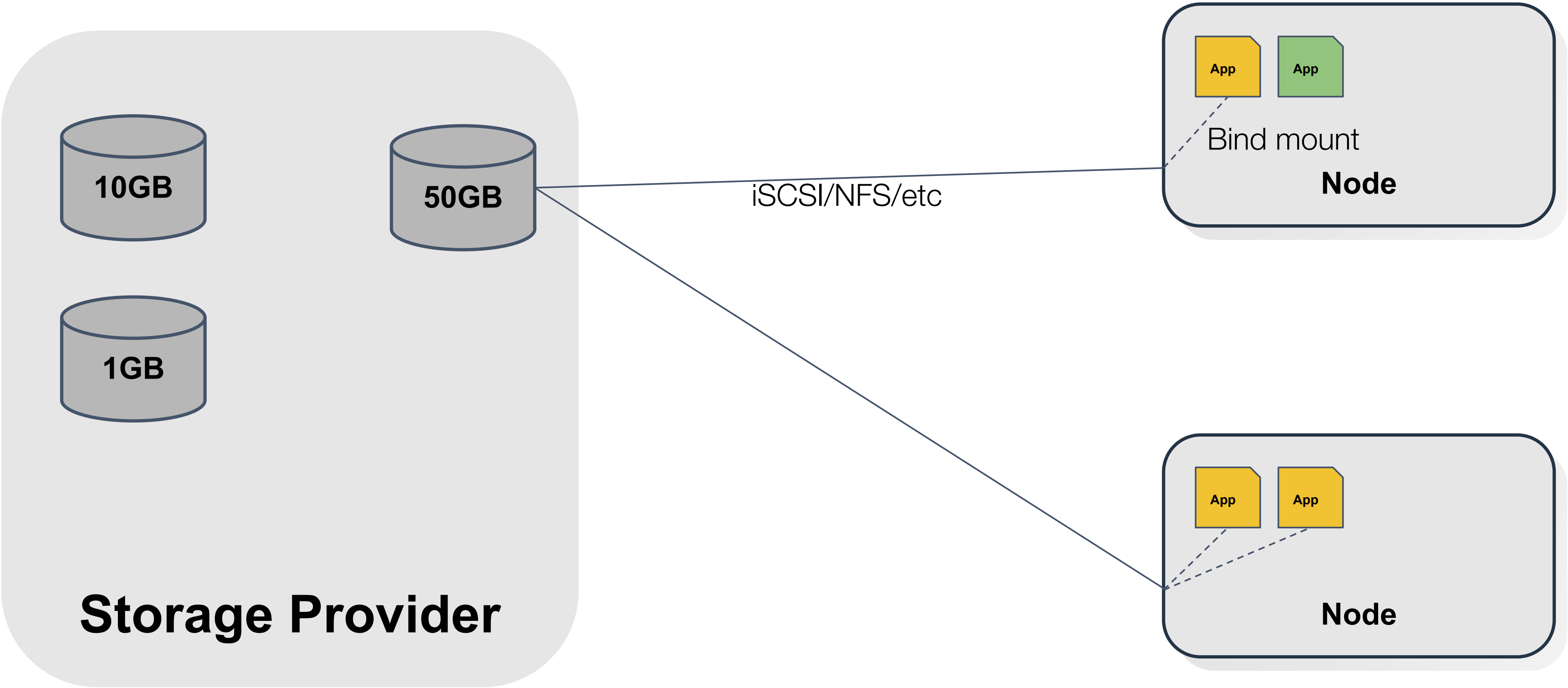
Module Outline

- Statefulness
- ConfigMaps & Secrets
- **Persistent Volumes**
- Demo

Types of Volumes

- EmptyDir - Scratch space for the pod (backed by Node storage or tmpfs)
- HostPath - Access directories on the hosts filesystem
- PersistentVolume

Persistent Volumes (PV)



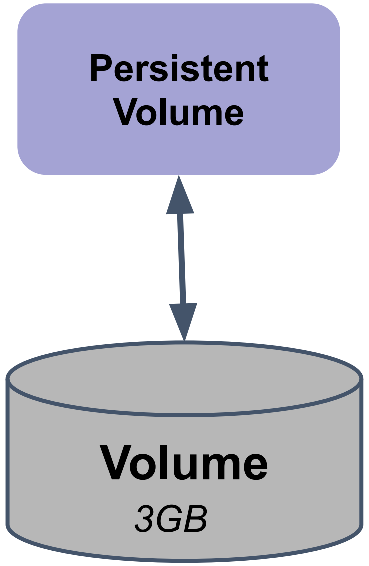
A Note on Volume Provisioning

Static Provisioning = Volumes are pre-created by an admin

Dynamic Provisioning = Volumes are created by kubernetes storage plugins when requested (via PersistentVolumeClaim)

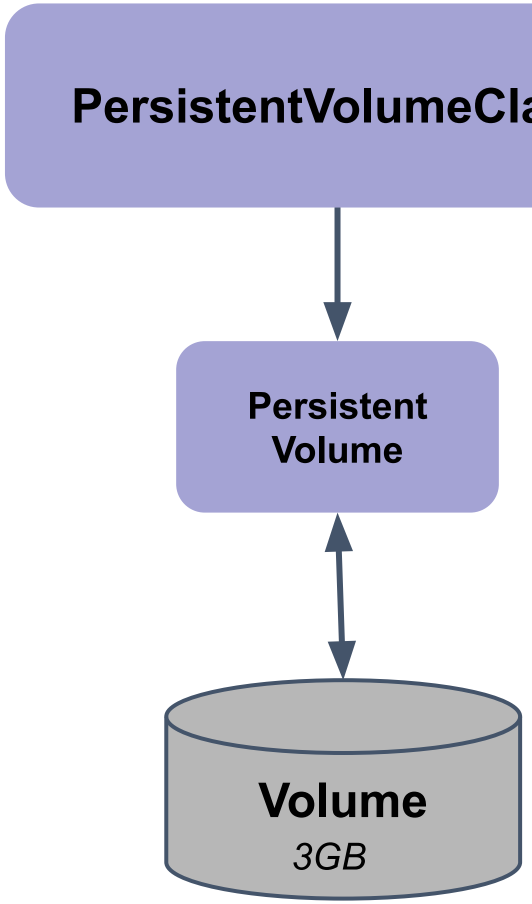
PersistentVolume

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: pv-nfs-0001
spec:
  persistentVolumeReclaimPolicy: Retain
  accessModes:
    - ReadWriteOnce
    - ReadWriteMany
    - ReadOnlyMany
  capacity:
    storage: 3Gi
  nfs:
    server: 192.168.1.3
    path: "/shares/0001"
```



Requesting a new PV (via PersistentVolumeClaims)

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: example-pvc
spec:
  storageClassName: slow
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```



StorageClass

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: slow
annotations:
  storageclass.kubernetes.io/is-default-class: "true"
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-standard
  replication-type: none
```

StorageClass

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
annotations:
  storageclass.kubernetes.io/is-default-class: "false"
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
  replication-type: none
```

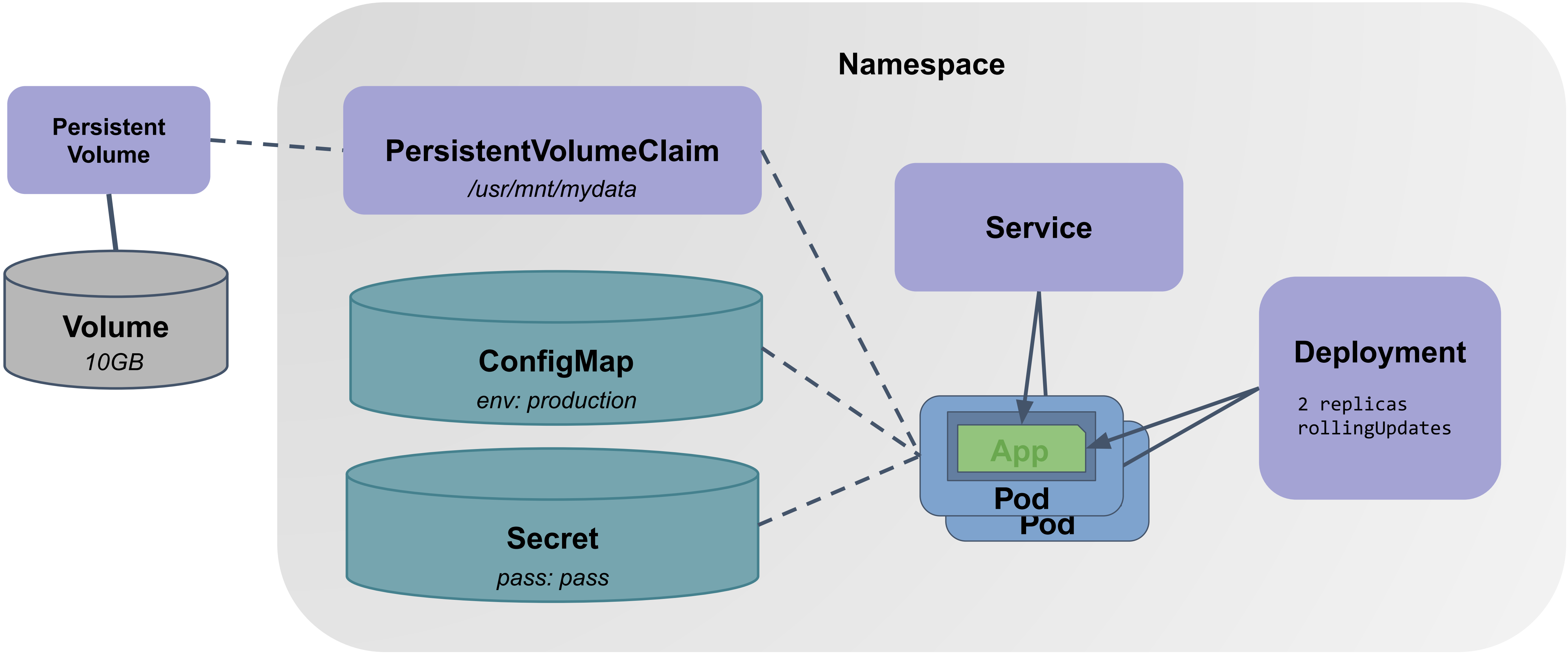
Consuming a PV

```
kind: Pod
apiVersion: v1
metadata:
  name: example-pod
spec:
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
      - mountPath: "/var/www/html"
        name: myvolume
  volumes:
    - name: myvolume
      persistentVolumeClaim:
        claimName: myclaim
```

Upcoming PV features

- Snapshots (v1.12 Alpha)
- Raw Block Volumes (v1.13 beta)
- Volume Expansion (v1.14 Alpha)
- Topology Support (v1.14 Alpha)

Review


Namespace
Persistent Volume
Volume
10GB
PersistentVolumeClaim
/usr/mnt/mydata
Service
ConfigMap
env: production
Secret
pass: pass
App
Pod
Pod
Deployment
2 replicas
rollingUpdates

Demo

Using PVs and ConfigMaps

In what ways can my application consume information from a Secret?

1. Shell environment variables

2. As volumes, with the filenames equal to the keys and their contents equal to the values.

3. As arguments to the command that runs my application.

What type of data should be stored in a PV?

1. Log files

2. Application source code

3. Application binaries

4. State shared across pods

5. Static content for webpages (large amounts)

Additional Resources

- StorageClasses - https://kubernetes.io/docs/concepts/storage/storage-classes/

- PersistentVolumes - https://kubernetes.io/docs/concepts/storage/persistent-volumes/

- Secrets - https://kubernetes.io/docs/concepts/configuration/secret/