

OpenShift Github Training Advanced

Trainer: Alexander Kolin

The World's Local Training Provider

Welcome



Alexander Kolin

Software/System Architect

Web-Developer

DevOps-Engineer

Cloud & Migration Consultant

Course objectives

Course objectives

Day 1

- Openshift Architecture
- Network
- Storage
- CRD & Operators
- Kustomize

Course objectives

Day 1

- Openshift Architecture
- Network
- Storage
- CRD & Operators
- Kustomize

Day 2

- Github Actions
- Pod Security
- Storage management
- Advanced Management
- DR & HA Strategies

Course objectives

Day 1

- Openshift Architecture
- Network
- Storage
- CRD & Operators
- Kustomize

Day 2

- Github Actions
- Pod Security
- Storage management
- Advanced Management
- DR & HA Strategies

Day 3

- App & Infrastructure Security
- Monitoring
- Openshift Operators
- DR & HA Velero
- Wrap Up & Questions

Preparations

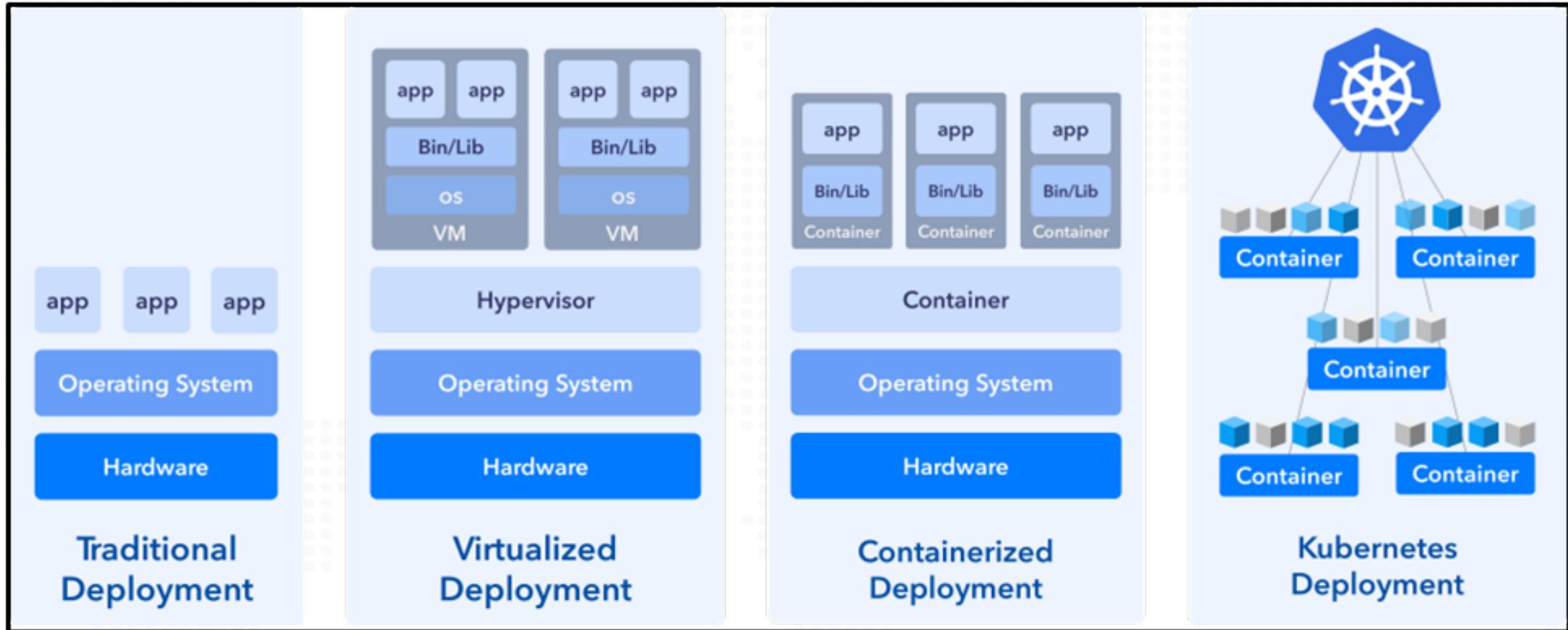
Preparations

Open Terminal

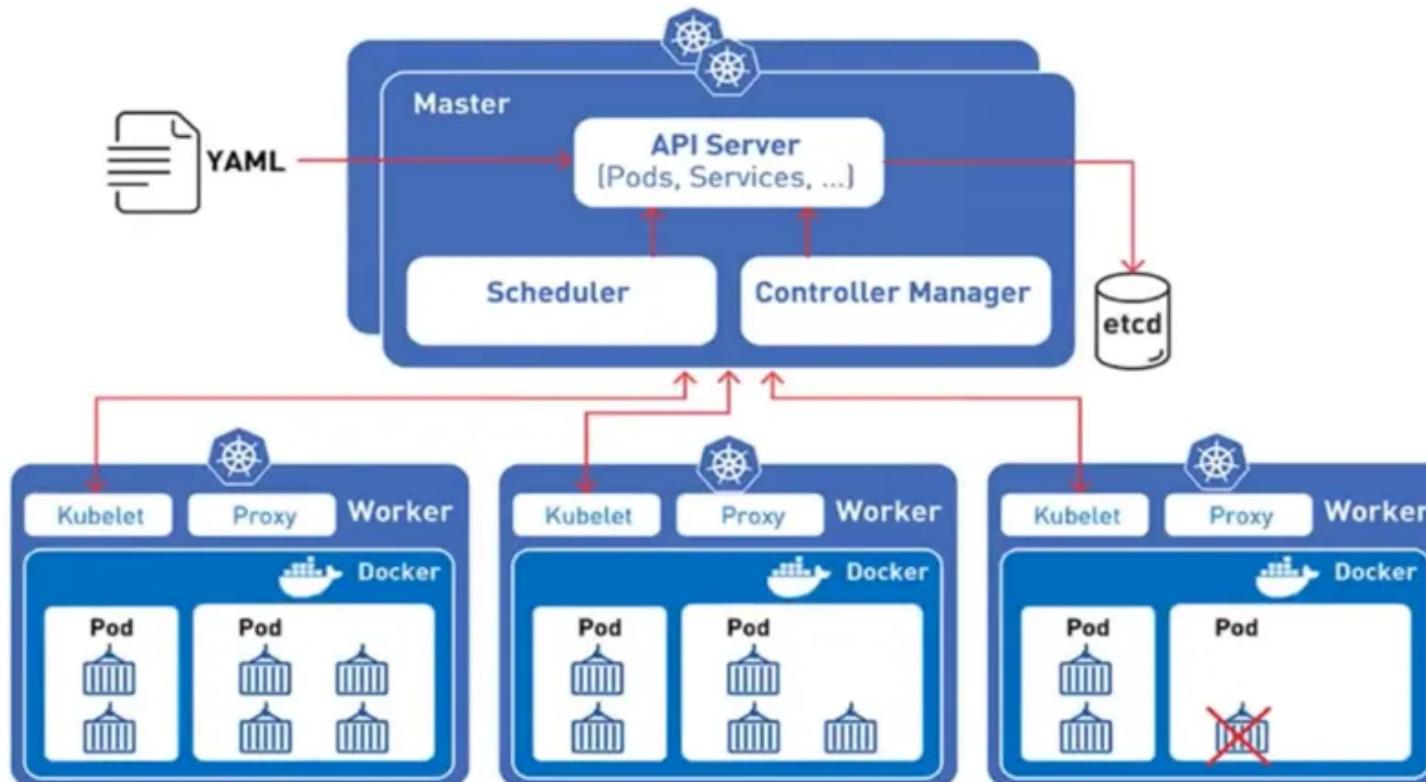
write following commands line by line, run with enter-key:

- cd \$HOME
- git clone <https://github.com/kolinrr/openshift-adv.git>
- crc start -p \$HOME/pull-secret

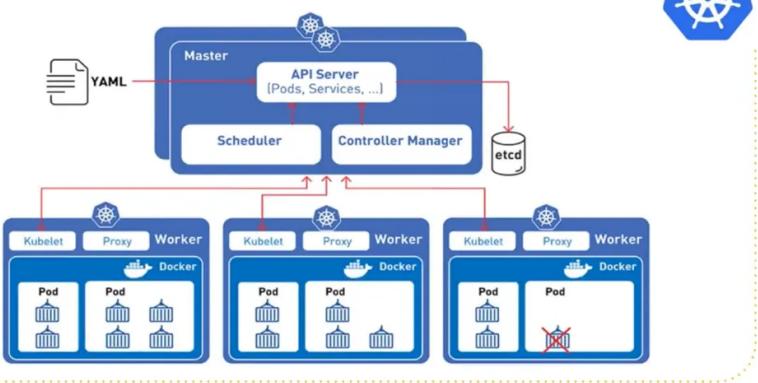
Cloud solutions



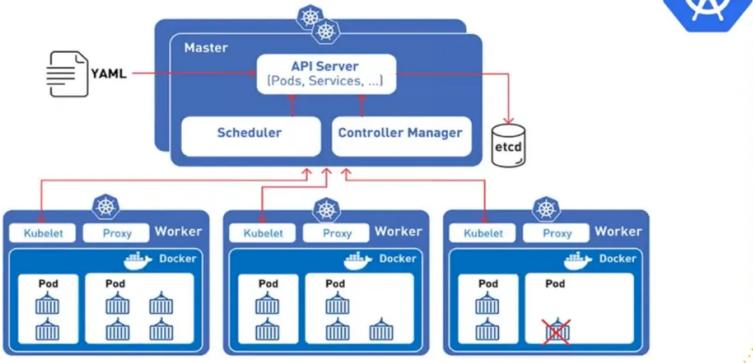
Kubernetes Architecture



Kubernetes Architecture



Kubernetes Architecture



Control Plane

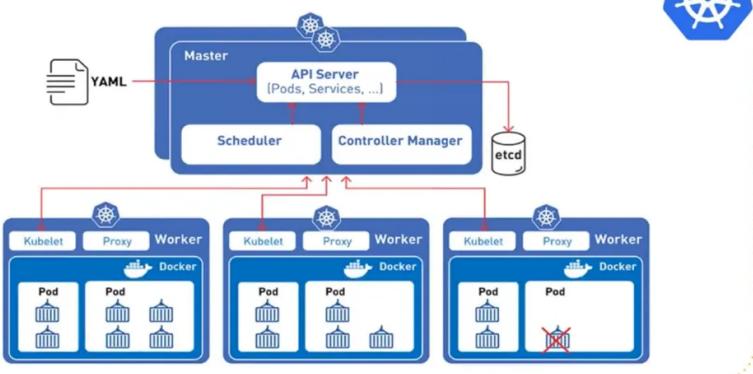
Api Server: control interface for k8s and clients

Controller Manager: cluster monitoring, change management (replicas), restarting pods

Scheduler: pods deployment on nodes

Etcdb: persistent internal k8s key-value storage

Kubernetes Architecture



Control Plane

Api Server: control interface for k8s and clients

Controller Manager: cluster monitoring, change management (replicas), restarting pods

Scheduler: pods deployment on nodes

Etcdb: persistent internal k8s key-value storage

Nodes

Kubelet: Container agent, starts the pod

Container runtime: Docker, CRI-O, containerd

Kube Proxy: For network - traffic to service

Networking SDN

Networking SDN

Software Defined Network

Provided by Container Network Interfaces (CNI) like OpenShift SDN, Calico, Cilium, Flannel..

Non-Physical network control level

Each Pod gets own IP-address

Networking SDN

CNI Benchmark August 2020 infraBuilder	Config	Performances (bandwidth)				Resources consumption (cpu/ram)				Security features					
		MTU	Pod to Pod		Pod to Service		Idle	Pod to Pod		Pod to Service		Network Policies	Encryption		
	setting	setting	TCP	UDP	TCP	UDP	none	TCP	UDP	TCP	UDP	in	out	activation	Performance
Antrea	auto	Very fast	Very fast	Very fast	Slow	Low	Low	Low	Low	Low	Low	yes	yes	at deploy time	Slow
Calico	manual	Very fast	Very fast	Very fast	Fast	Low	Very low	Very low	Very low	Very low	Very low	yes	yes	anytime	Very fast
Canal	manual	Very fast	Very fast	Very fast	Very fast	Low	Very low	Very low	Very low	Very low	Very low	yes	yes	no	n/a
Cilium	auto	Fast	Very fast	Very fast	Very fast	High	High	High	High	High	High	yes	yes	at deploy time	Slow
Flannel	auto	Very fast	Very fast	Very fast	Very fast	Very low	Very low	Very low	Very low	Very low	Very low	no	no	no	n/a
Kube-OVN	auto	Fast	Very slow	Fast	Very slow	High	High	High	High	High	High	yes	yes	no	n/a
Kube-router	none	Slow	Very slow	Slow	Very slow	Low	Very low	Low	Very low	Low	Low	yes	yes	no	n/a
Weave Net	manual	Very fast	Very fast	Very fast	Fast	Very low	Low	Low	Low	Low	Low	yes	yes	at deploy time	Slow

(c) <https://itnext.io/benchmark-results-of-kubernetes-network-plugins-cni-over-10gbit-s-network-updated-august-2020-6e1b757b9e49>
 CIDR - Classless Inter-Domain Routing

Networking Openshift SDN

Feature	OpenShift SDN	OVN-Kubernetes
Egress IPs	Supported	Supported
Egress firewall [1]	Supported	Supported
Egress router	Supported	Supported [2]
IPsec encryption	Not supported	Supported
IPv6	Not supported	Supported [3]
Kubernetes network policy	Partially supported [4]	Supported
Kubernetes network policy logs	Not supported	Supported
Multicast	Supported	Supported

Network Policies

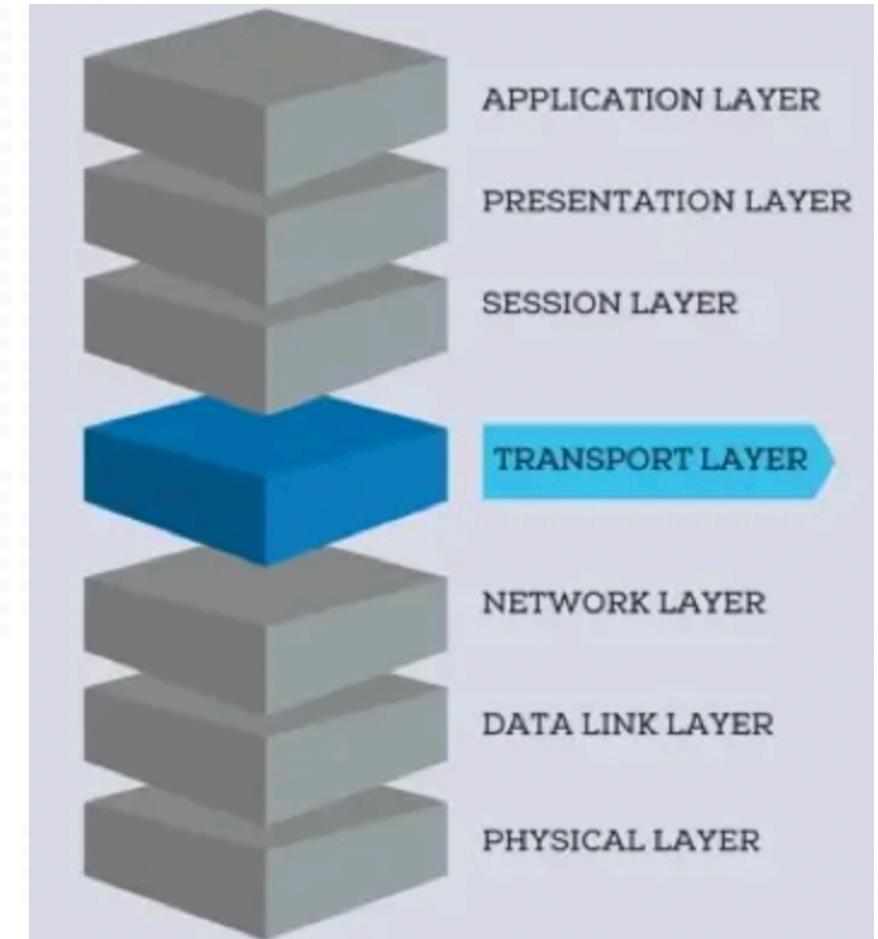
Network Policies

Objects to control/restrict ingress and egress traffic

Network Policies

Objects to control/restrict ingress and egress traffic

OSI Model



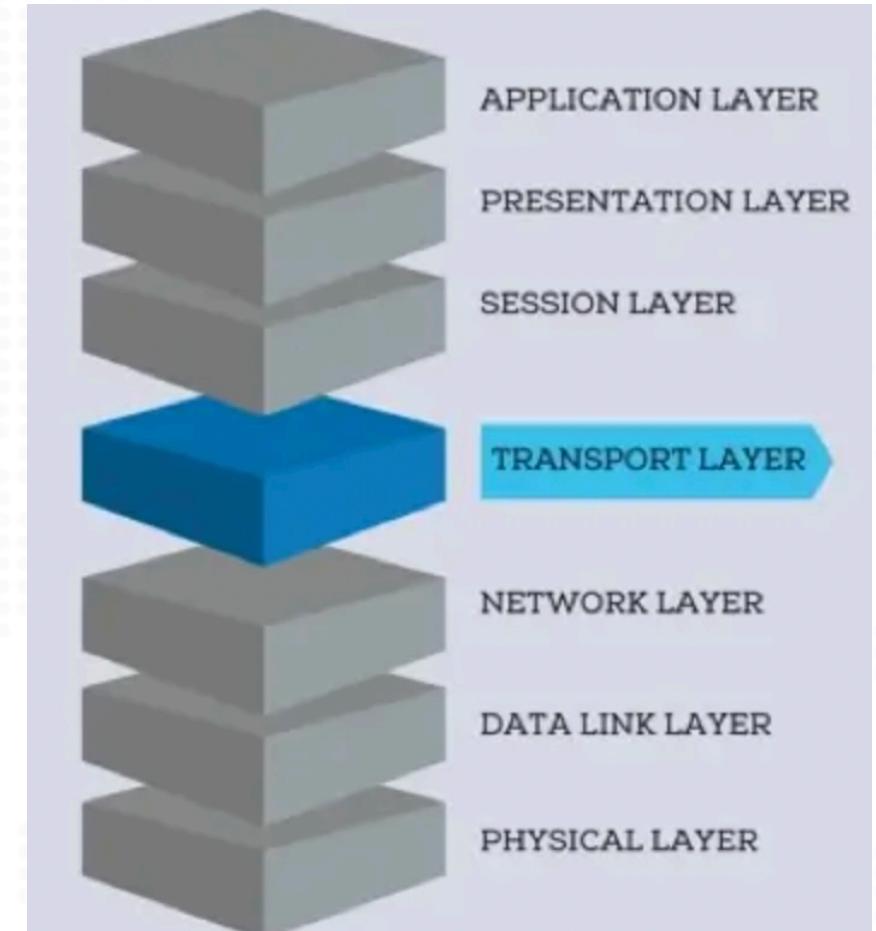
(c) <https://www.whatismyip.com/transport-layer/>

Network Policies

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-cross-namespace
  namespace: dev
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress
  ingress:
    - from:
        - podSelector: {}
  egress:
    - to:
        - podSelector: {}
```

OSI Model



(c) <https://www.whatismyip.com/transport-layer/>

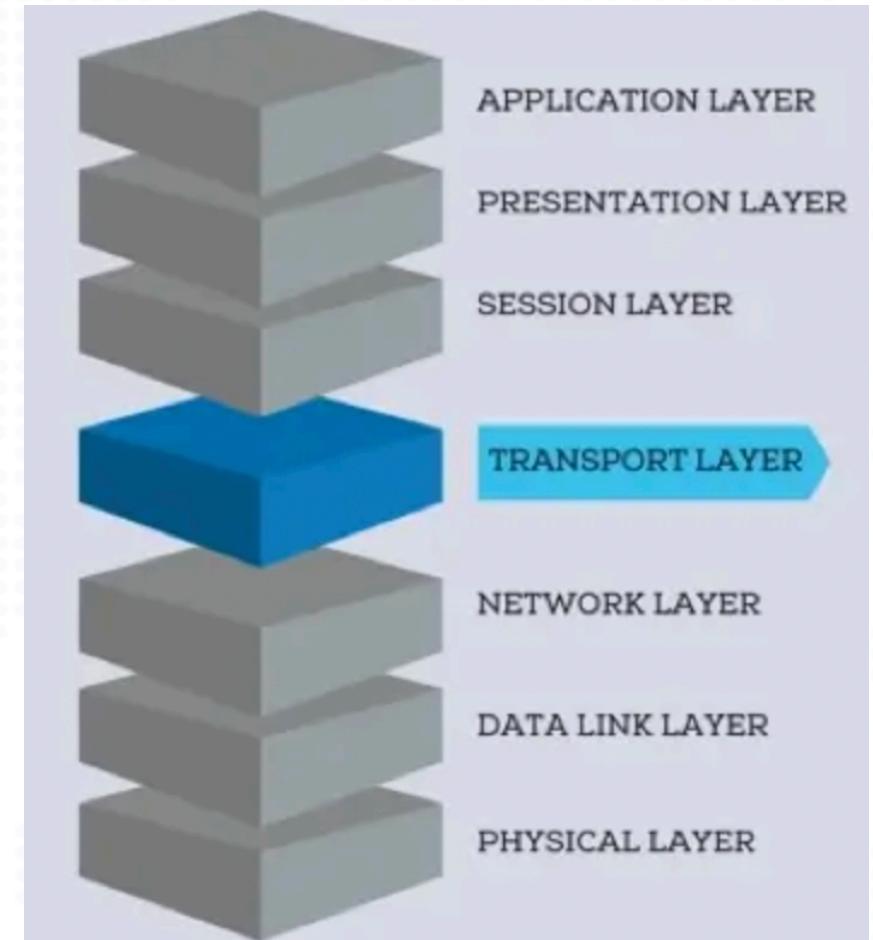
Network Policies

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-cross-namespace
  namespace: dev
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress
  ingress:
    - from:
        - podSelector: {}
  egress:
    - to:
        - podSelector: {}
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-cross-namespace
  namespace: dev
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress
  ingress:
    - from:
        - podSelector: {}
        - namespaceSelector:
            matchLabels:
              project: monitoring
  egress:
    - to:
        - podSelector: {}
```

OSI Model



(c) <https://www.whatismyip.com/transport-layer/>

Network Policies - Openshift

Network Policies - Openshift

Objects to control/restrict ingress and egress traffic

Network Policies - Openshift

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-to-own-namespace
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
  policyTypes:
    - Ingress
```

Network Policies - Openshift

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-to-own-namespace
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
  policyTypes:
    - Ingress
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-monitoring-access
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
        - namespaceSelector:
            matchLabels:
              name: monitoring
  policyTypes:
    - Ingress
```

Network Policies - Openshift

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-to-own-namespace
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
  policyTypes:
    - Ingress
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-monitoring-access
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
        - namespaceSelector:
            matchLabels:
              name: monitoring
  policyTypes:
    - Ingress
```

```
apiVersion: k8s.ovn.org/v1
kind: EgressFirewall
metadata:
  name: allow-specific-egress
  namespace: dev
spec:
  egress:
    - type: Allow
      to:
        dnsName: example.com
    - type: Deny
      to:
        cidrSelector: 0.0.0.0/0
  priority: 100
```

Network Policies - OpenShift

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: restrict-to-own-namespace
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
  policyTypes:
    - Ingress
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-monitoring-access
  namespace: dev
spec:
  podSelector: {}
  ingress:
    - from:
        - podSelector: {}
        - namespaceSelector:
            matchLabels:
              name: monitoring
  policyTypes:
    - Ingress
```

```
apiVersion: k8s.ovn.org/v1
kind: EgressFirewall
metadata:
  name: allow-specific-egress
  namespace: dev
spec:
  egress:
    - type: Allow
      to:
        dnsName: example.com
    - type: Deny
      to:
        cidrSelector: 0.0.0.0/0
  priority: 100
```

* only in OVN-Kubernetes - newer SDN in OpenShift

Objects to control/restrict ingress and egress traffic

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress # not in OpenShift
```

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress # not in OpenShift
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: frontend
  ports:
    - protocol: TCP
      port: 1337
```

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress # not in OpenShift
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: frontend
  ports:
    - protocol: TCP
      port: 1337
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-backend-to-db
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: database
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: backend
  ports:
    - protocol: TCP
      port: 3306
```

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress # not in OpenShift
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: frontend
  ports:
    - protocol: TCP
      port: 1337
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-backend-to-db
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: database
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: backend
  ports:
    - protocol: TCP
      port: 3306
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-ingress-to-frontend
  namespace: myapp
spec:
  podSelector:
    matchLabels:
      app: frontend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - ipBlock:
            cidr: 0.0.0.0/0
  ports:
    - protocol: TCP
      port: 8080
```

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
policyTypes:
  - Ingress
  - Egress # not in OpenShift
```

```
apiVersion: network.openshift.io/v1
kind: EgressNetworkPolicy
metadata:
  name: deny-all-egress
  namespace: myapp
spec:
  egress:
    - type: Deny
      to:
        cidrSelector: 0.0.0.0/0
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: frontend
  ports:
    - protocol: TCP
      port: 1337
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-backend-to-db
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: database
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: backend
  ports:
    - protocol: TCP
      port: 3306
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-ingress-to-frontend
  namespace: myapp
spec:
  podSelector:
    matchLabels:
      app: frontend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - ipBlock:
            cidr: 0.0.0.0/0
  ports:
    - protocol: TCP
      port: 8080
```

Network Policies - Openshift

Objects to control/restrict ingress and egress traffic

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: myapp
spec:
  podSelector: {}
policyTypes:
  - Ingress
  - Egress # not in OpenShift
```

```
apiVersion: network.openshift.io/v1
kind: EgressNetworkPolicy
metadata:
  name: deny-all-egress
  namespace: myapp
spec:
  egress:
    - type: Deny
      to:
        cidrSelector: 0.0.0.0/0
```

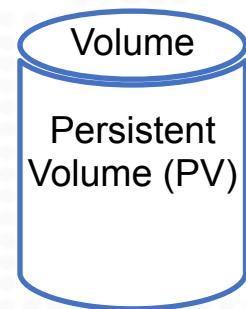
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-frontend-to-backend
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: frontend
  ports:
    - protocol: TCP
      port: 1337
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-backend-to-db
  namespace: dev
spec:
  podSelector:
    matchLabels:
      app: database
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: backend
  ports:
    - protocol: TCP
      port: 3306
```

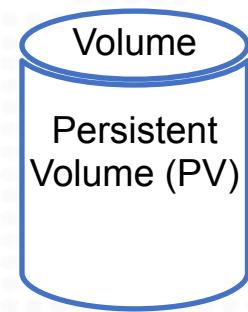
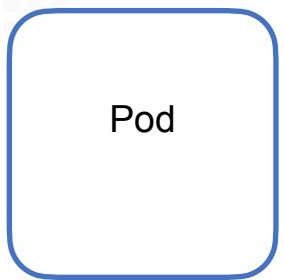
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-ingress-to-frontend
  namespace: myapp
spec:
  podSelector:
    matchLabels:
      app: frontend
  policyTypes:
    - Ingress
  ingress:
    - from:
        - ipBlock:
            cidr: 0.0.0.0/0
  ports:
    - protocol: TCP
      port: 8080
```

Storage Part 1

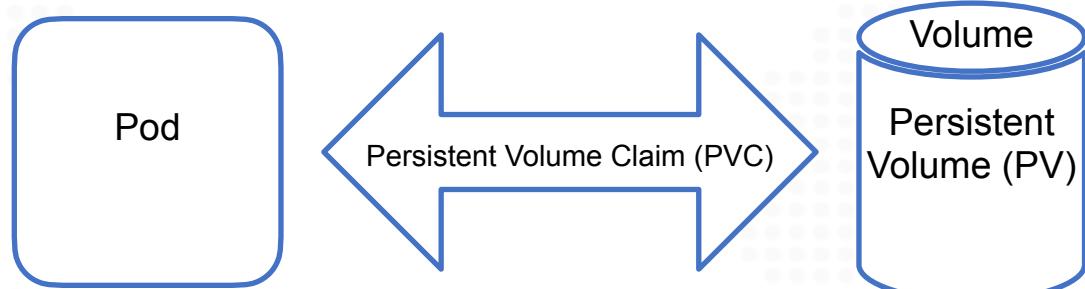
Storage Part 1



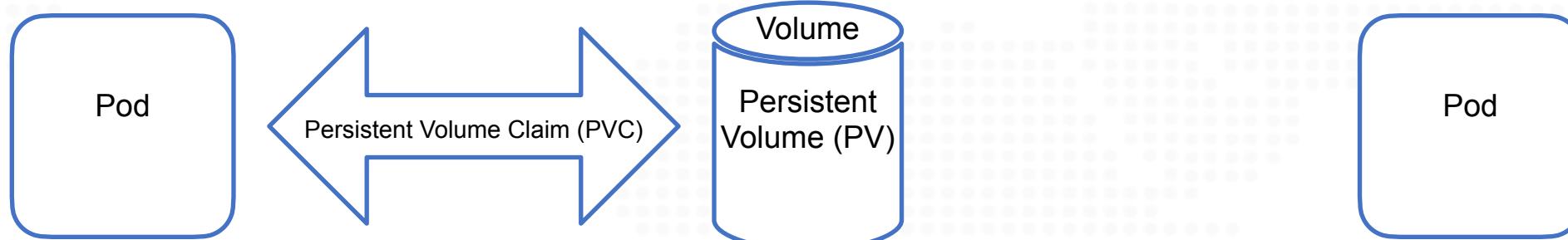
Storage Part 1



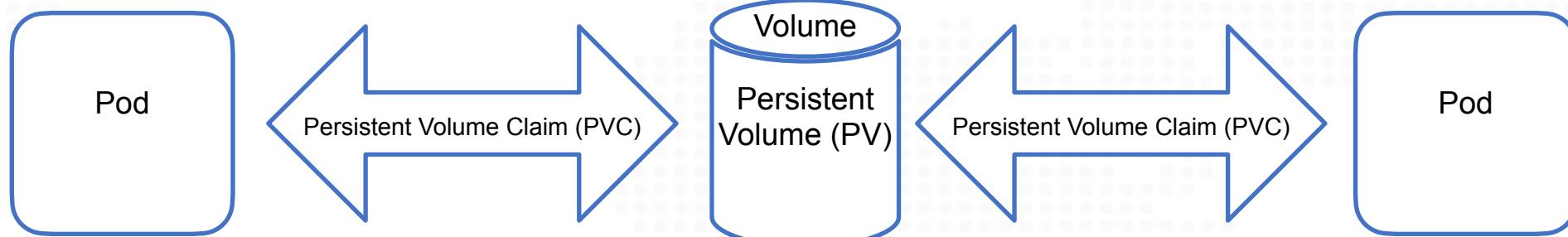
Storage Part 1



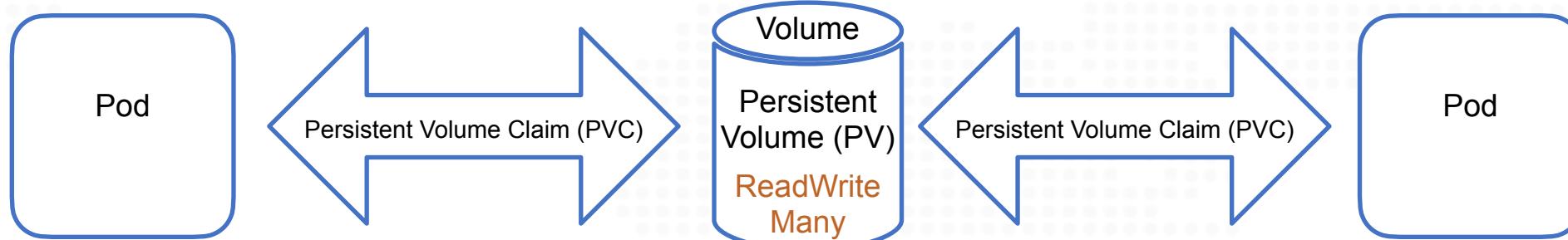
Storage Part 1



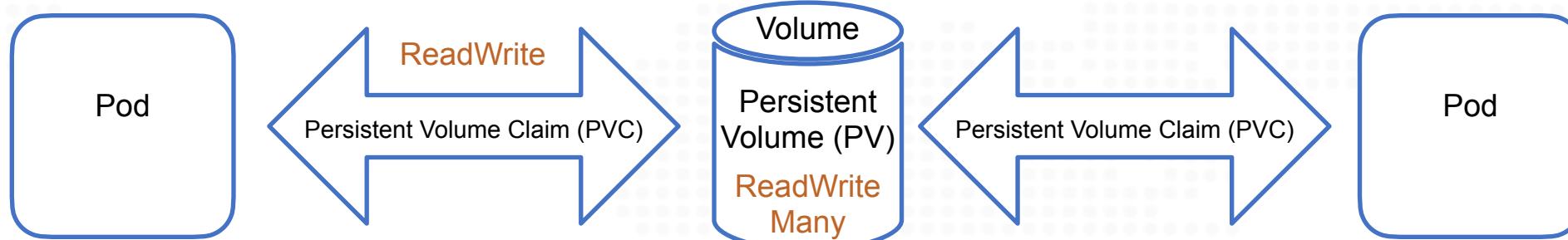
Storage Part 1



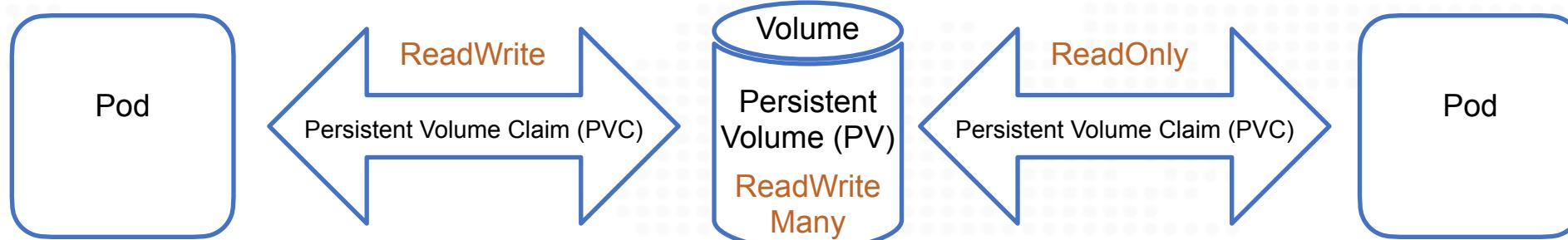
Storage Part 1



Storage Part 1

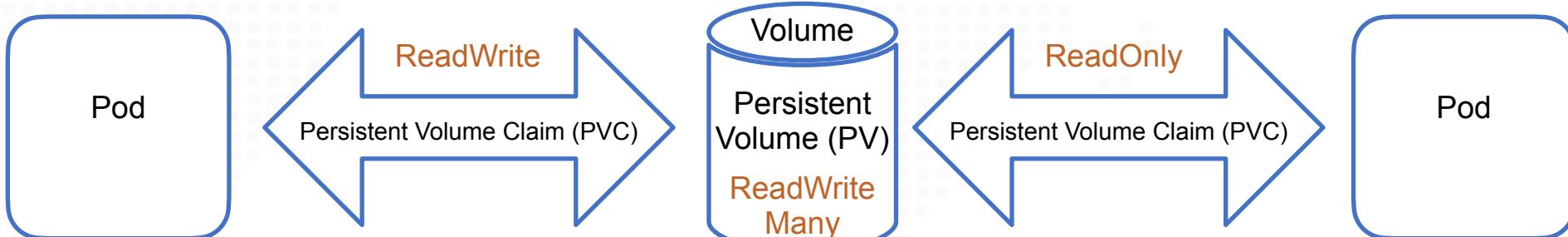


Storage Part 1



PV - Persistent Volume

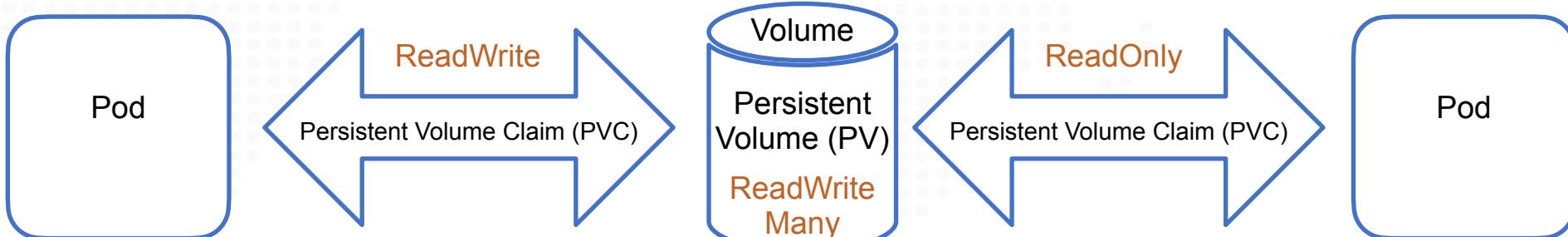
- Provided by Admin or Storage Class
- Independent of pods
- Defines:
 - Capacity
 - Access mode: RO/RWO/RWM
 - Reclaim: after PVC was deleted - retain/delete/recylce



PV - Persistent Volume

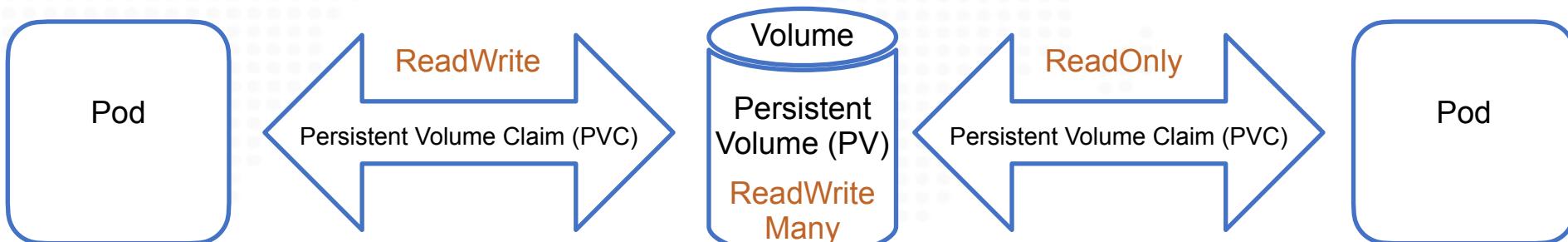
- Provided by Admin or Storage Class
- Independent of pods
- Defines:
 - Capacity
 - Access mode: RO/RWO/RWM
 - Reclaim: after PVC was deleted - retain/delete/recylce

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: "/mnt/data"
```



PVC - Persistent Volume Claim

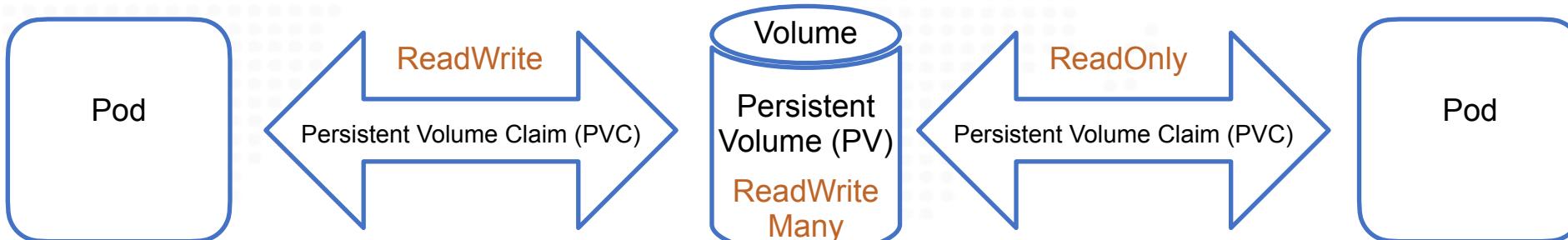
- Provided by User/Developer
- Defines:
 - Requested Capacity from PV
 - Access mode: RO/RWO/RWM



PVC - Persistent Volume Claim

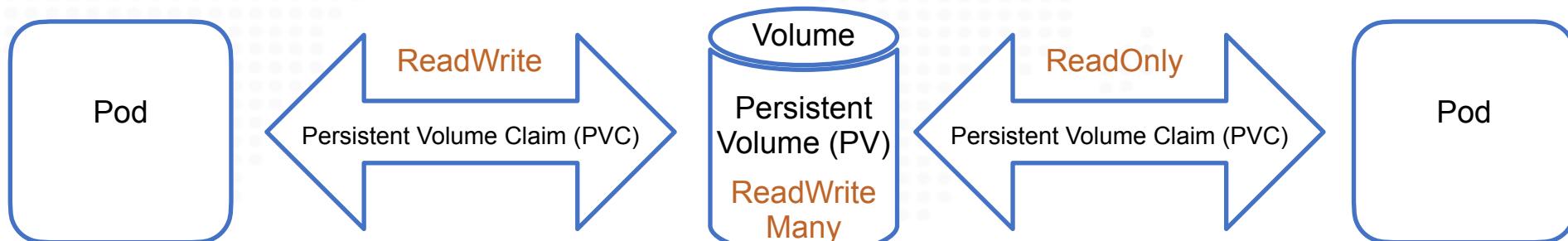
- Provided by User/Developer
- Defines:
 - Requested Capacity from PV
 - Access mode: RO/RWO/RWM

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: ""
```



SC - Storage Class

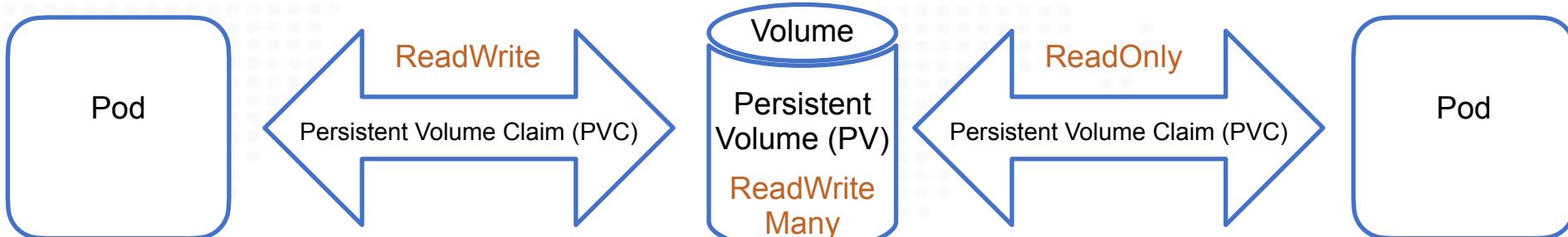
- Provided by Admin
- Allows Dynamic Provisioning of PVs
- Defines:
 - Provisioner: CephFS/NFS/AWS/GCP...
 - Parameters: disk type, performance settings



SC - Storage Class

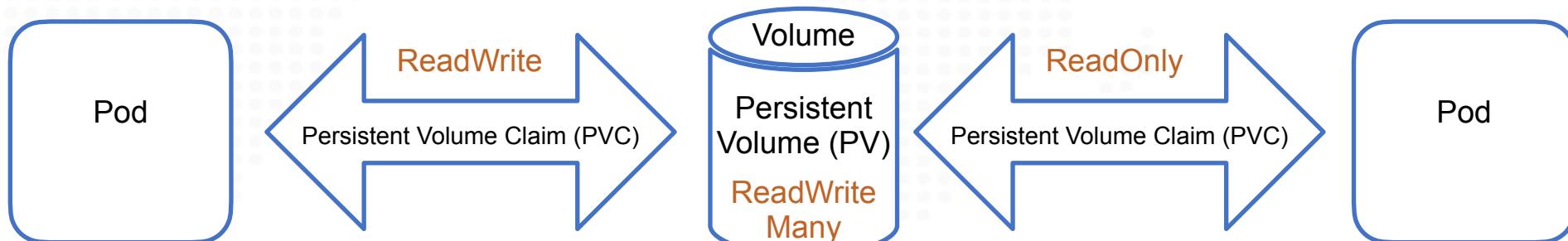
- Provided by Admin
- Allows Dynamic Provisioning of PVs
- Defines:
 - Provisioner: CephFS/NFS/AWS/GCP...
 - Parameters: disk type, performance settings

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: fast-ssd
provisioner: kubernetes.io/aws-ebs
parameters:
  type: io1
  iopsPerGB: "10"
  fsType: ext4
```



Dynamic Provisioning

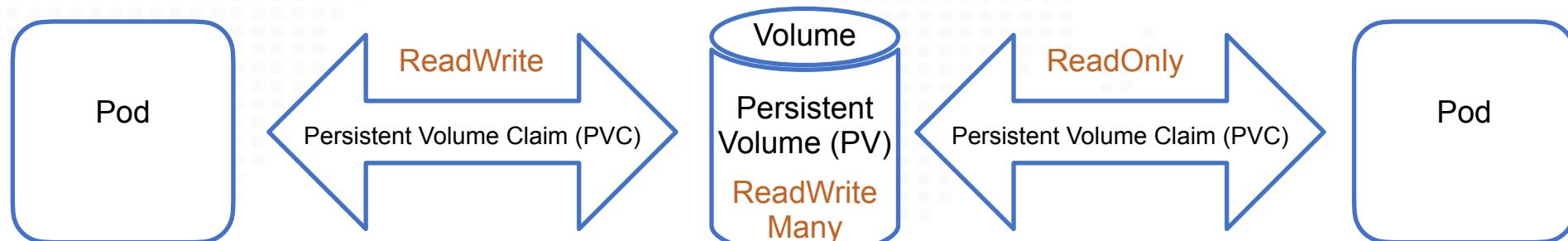
- No need of Admin to provide PV
- Defining of PVC will trigger the creation of the PV



Dynamic Provisioning

- No need of Admin to provide PV
- Defining of PVC will trigger the creation of the PV

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-dynamic-pvc
spec:
  accessModes:
    - ReadWriteOnce
resources:
  requests:
    storage: 20Gi
storageClassName: fast-ssd
```



Storage Lab

- pvc.yaml
- pv.yaml

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/basic
```

Storage Lab

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/basic
```

- pvc.yaml
- pv.yaml

```
apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: my-pv  
spec:  
  capacity:  
    storage: 5Gi  
  accessModes:  
    - ReadWriteOnce  
  persistentVolumeReclaimPolicy: Retain  
  hostPath:  
    path: "/mnt/data"
```

Storage Lab

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/basic
```

- pvc.yaml

- pv.yaml

```
apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: my-pv  
spec:  
  capacity:  
    storage: 5Gi  
  accessModes:  
    - ReadWriteOnce  
  persistentVolumeReclaimPolicy: Retain  
  hostPath:  
    path: "/mnt/data"
```

- oc apply -f pvc.yaml
- oc get pvc

Storage Lab

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/basic
```

- pvc.yaml

```
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
  name: my-pvc  
  namespace: default  
spec:  
  accessModes:  
    - ReadWriteOnce  
resources:  
  requests:  
    storage: 1Gi  
storageClassName: ""
```

- pv.yaml

```
apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: my-pv  
spec:  
  capacity:  
    storage: 5Gi  
  accessModes:  
    - ReadWriteOnce  
  persistentVolumeReclaimPolicy: Retain  
  hostPath:  
    path: "/mnt/data"
```

- oc apply -f pvc.yaml
- oc get pvc

Storage Lab

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/basic
```

- pvc.yaml

```
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
  name: my-pvc  
  namespace: default  
spec:  
  accessModes:  
    - ReadWriteOnce  
resources:  
  requests:  
    storage: 1Gi  
storageClassName: ""
```

- pv.yaml

```
apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: my-pv  
spec:  
  capacity:  
    storage: 5Gi  
  accessModes:  
    - ReadWriteOnce  
  persistentVolumeReclaimPolicy: Retain  
  hostPath:  
    path: "/mnt/data"
```

- oc apply -f pvc.yaml
- oc get pvc

- oc apply -f pv.yaml
- oc get pv
- oc get pvc

Storage Lab

- pvc.yaml
- sc.yaml
- pvc2.yaml

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/1.storage/storageclass
```

Storage Lab

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath
provisioner: kubevirt.io/hostpath-provisioner
reclaimPolicy: Delete
parameters:
  storagePool: local
volumeBindingMode: Immediate
```

- pvc.yaml
- sc.yaml

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/1.storage/storageclass
```

- pvc2.yaml

Storage Lab

- sc.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath
provisioner: kubevirt.io/hostpath-provisioner
reclaimPolicy: Delete
parameters:
  storagePool: local
volumeBindingMode: Immediate
```

- pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

- pvc2.yaml

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/1.storage/storageclass
```

Storage Lab

- sc.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath
provisioner: kubevirt.io/hostpath-provisioner
reclaimPolicy: Delete
parameters:
  storagePool: local
volumeBindingMode: Immediate
```

- pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

- pvc2.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-2
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/1.storage/storageclass
```

Storage Lab

- sc.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath
provisioner: kubevirt.io/hostpath-provisioner
reclaimPolicy: Delete
parameters:
  storagePool: local
volumeBindingMode: Immediate
```

- pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/1.storage/storageclass
```

- oc apply -f sc.yaml
- oc get pv

- pvc2.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-2
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

Storage Lab

- sc.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: hostpath
provisioner: kubevirt.io/hostpath-provisioner
reclaimPolicy: Delete
parameters:
  storagePool: local
volumeBindingMode: Immediate
```

- pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/1.storage/storageclass
```

- oc apply -f sc.yaml
- oc get pv

- pvc2.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc-2
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: hostpath
```

- oc apply -f pvc.yaml
- oc apply -f pvc2.yaml
- oc get pvc
- oc get pv

CRD

- CRD - Custom Resource Definition
- Creates new objects and extends K8s

CRD

- CRD - Custom Resource Definition
- Creates new objects and extends K8s

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: mycrds.mygroup.example.com
spec:
  group: mygroup.example.com
  names:
    kind: MyCRD
    plural: mycrds
    scope: Namespaced
  versions:
    - name: v1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                secretName:
                  type: string
```

CRD

- CRD - Custom Resource Definition
- Creates new objects and extends K8s

```
apiVersion: mygroup.example.com/v1
kind: MyCRD
metadata:
  name: example-cr
spec:
  secretName: my-secret
  someConfig: my-config
  whatever: iwant
```

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: mycrds.mygroup.example.com
spec:
  group: mygroup.example.com
  names:
    kind: MyCRD
    plural: mycrds
  scope: Namespaced
  versions:
    - name: v1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                secretName:
                  type: string
```

Operator

- Is like the APP
- Written in GO
- „CR-Watcher“

Operator

- Is like the APP
- Written in GO
- „CR-Watcher“

```
...
type MyCR struct {
    Spec struct {
        Value string `json:"value"`
    } `json:"spec"`
}
...

func main() {
    mgr, err := manager.New(ctrl.GetConfigOrDie(), manager.Options{})
    if err != nil {
        fmt.Println("Failed to create manager", err)
        os.Exit(1)
    }

    k8sClient := mgr.GetClient()
    ctx := context.Background()

    for {
        cr := &MyCR{}
        err := k8sClient.Get(ctx, client.ObjectKey{Name: "my-cr", Namespace: "default"}, cr)
        if err == nil {
            fmt.Println("CR Value:", cr.Spec.Value)
        }
    }
}
```

Operator

- Is like the APP
- Written in GO
- „CR-Watcher“

```
apiVersion: example.com/v1
kind: MyCR
metadata:
  name: my-cr
  namespace: default
spec:
  value: "Hello, Operator!"
```

```
...
type MyCR struct {
    Spec struct {
        Value string `json:"value"`
    } `json:"spec"`
}
...

func main() {
    mgr, err := manager.New(ctrl.GetConfigOrDie(), manager.Options{})
    if err != nil {
        fmt.Println("Failed to create manager", err)
        os.Exit(1)
    }

    k8sClient := mgr.GetClient()
    ctx := context.Background()

    for {
        cr := &MyCR{}
        err := k8sClient.Get(ctx, client.ObjectKey{Name: "my-cr", Namespace: "default"}, cr)
        if err == nil {
            fmt.Println("CR Value:", cr.Spec.Value)
        }
    }
}
```

CRD & Operator

- CRD: Defines new K8s resource types
- CR: Custom instance of a CRD
- Operator: Run logic for CR
- Use Case: do anything based on CR-Configuration

CRD & Operator

- CRD: Defines new K8s resource types
- CR: Custom instance of a CRD
- Operator: Run logic for CR
- Use Case: do anything based on CR-Configuration ... or not

CRD & Operator LAB

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/2.crd
```

- Open in your Browser:
<https://github.com/kolinrr/openshift-adv/Day1/2.crd>
- Follow the README.md

CRD & Operator LAB

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/2.crd
```

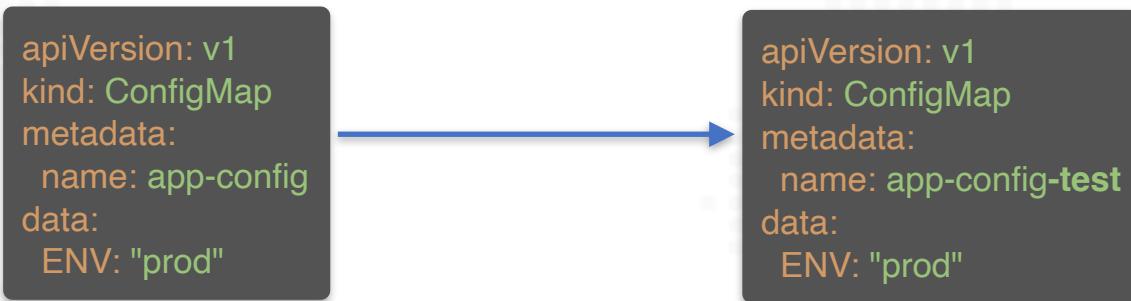
- Open in your Browser:
<https://github.com/kolinrr/openshift-adv/Day1/2.crd>
- Follow the README.md

- Open in your browser:
<https://github.com/kolinrr/openshift-adv/Day1/2.crd>
- Follow the README.m

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

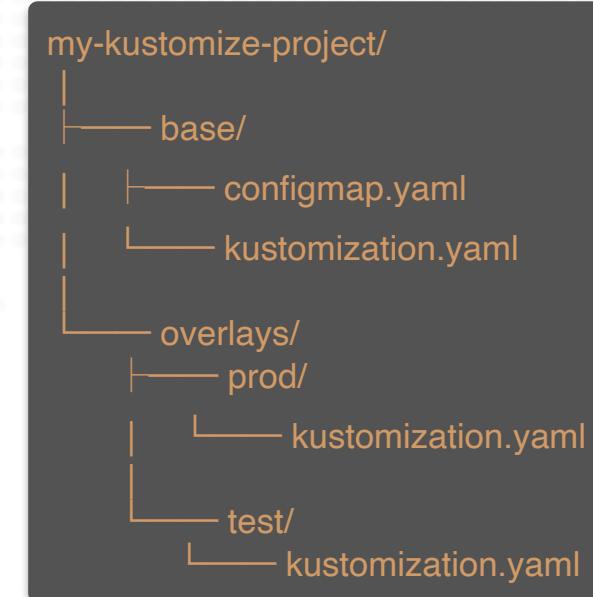
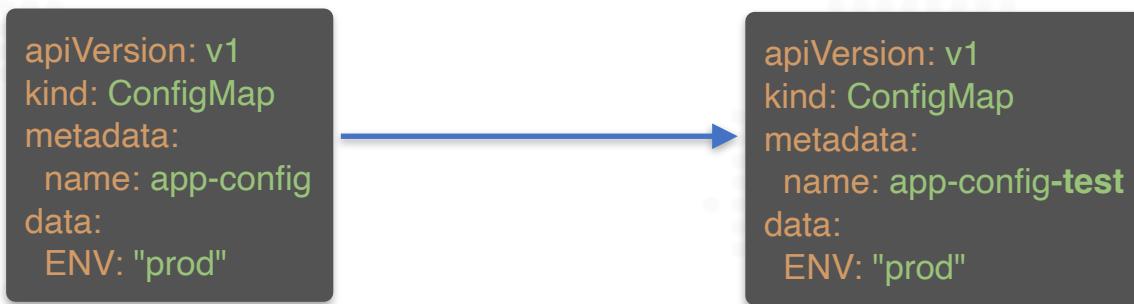
- Template manipulation of objects



Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects



Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
#overlays/test/kustomization.yaml  
bases:  
- ../../base  
nameSuffix: "-test"
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

oc kustomize overlays/prod

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
#overlays/test/kustomization.yaml  
bases:  
- ../../base  
nameSuffix: "-test"
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
#overlays/test/kustomization.yaml  
bases:  
- ../../base  
nameSuffix: "-test"
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
#overlays/test/kustomization.yaml  
bases:  
- ../../base  
nameSuffix: "-test"
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

oc apply -k overlays/prod

Kustomize 1

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/3.kustomize
```

- Template manipulation of objects

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config  
data:  
  ENV: "prod"
```



```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: app-config-test  
data:  
  ENV: "prod"
```

```
#base/kustomization.yaml  
resources:  
- configmap.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

```
#overlays/prod/kustomization.yaml  
bases:  
- ../../base
```

```
#overlays/test/kustomization.yaml  
bases:  
- ../../base  
nameSuffix: "-test"
```

```
my-kustomize-project/  
|  
|   base/  
|   |   configmap.yaml  
|   |   kustomization.yaml  
|  
|   overlays/  
|   |   prod/  
|   |   |   kustomization.yaml  
|  
|   |   test/  
|   |   |   kustomization.yaml
```

oc apply -k overlays/prod

oc apply -k overlays/test

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/4.kustomize
```

Kustomize 2

Extended template manipulation of objects

- Deploy 2 more application for prod- and test-environment
With different resource limits

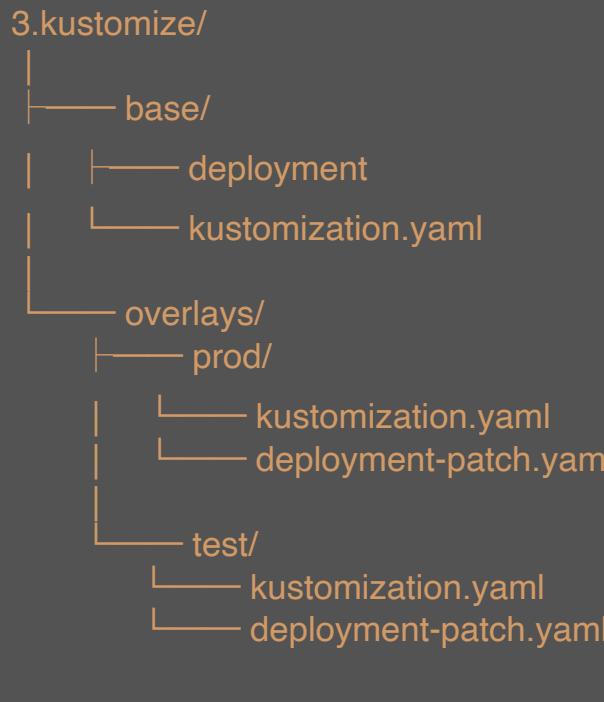
```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/4.kustomize
```

Kustomize 2

Extended template manipulation of objects

- Deploy 2 more application for prod- and test-environment

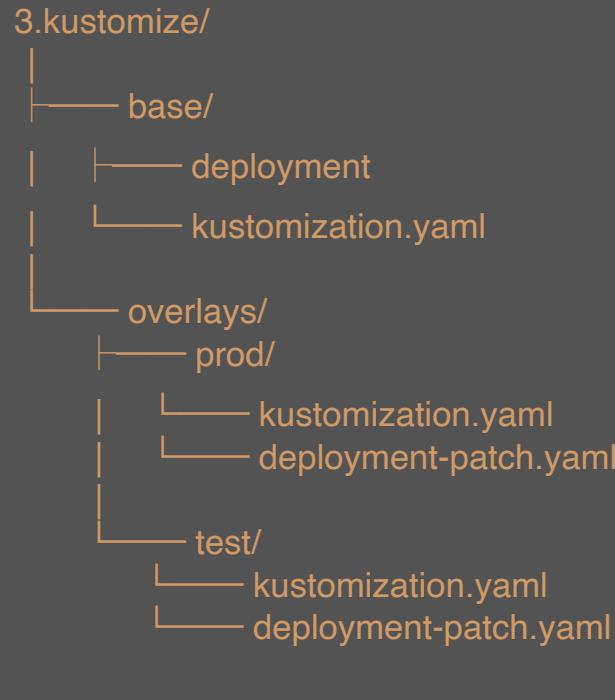
resource limits



Kustomize 2

Extended template management

- Deploy 2 more applications in prod- and test-environment



```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/4.kustomize
```

Kustomize 2

Extended template management

- Deploy 2 more applications in prod- and test-environment

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

    └── test/
      └── kustomization.yaml
      └── deployment-patch.yaml
```

```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/4.kustomize
```

```
#base/kustomization.yaml

resources:
  - deployment.yaml
```

Kustomize 2

Extended template management

- Deploy 2 more applications in prod- and test-environment

```
3.kustomize/
  └── base/
    └── deployment
    └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

  └── test/
    └── kustomization.yaml
    └── deployment-patch.yaml
```

```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
#base/kustomization.yaml

resources:
  - deployment.yaml
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/4.kustomize

#base/overlays/prod/deployment-patch.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

Kustomize 2

Extended template management

- Deploy 2 more applications in prod- and test-environment

```
3.kustomize/
  └── base/
    └── deployment
    └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

    └── test/
      └── kustomization.yaml
      └── deployment-patch.yaml
```

```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
#base/kustomization.yaml

resources:
  - deployment.yaml
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/4.kustomize

#base/overlays/prod/deployment-patch.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#overlays/test/kustomization.yaml

resources:
  - ../../base

apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

Kustomize 2

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

  └── test/
    └── kustomization.yaml
    └── deployment-patch.yaml
```

```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
#base/kustomization.yaml

resources:
  - deployment.yaml
```

```
git clone https://github.com/kolinrr/openshift-adv.git
cd openshift-adv/Day1/4.kustomize

#base/overlays/prod/deployment-patch.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#overlays/test/kustomization.yaml

resources:
  - ../../base

apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

Kustomize 2

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

  └── test/
    └── kustomization.yaml
    └── deployment-patch.yaml
```

```
#base/deployment.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/4.kustomize
```

```
#base/overlays/prod/deployment-patch.yaml

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#base/kustomization.yaml

resources:
  - deployment.yaml
```

```
#overlays/test/kustomization.yaml

resources:
  - ../../base

apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

oc kustomize overlays/prod

Kustomize 2

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

  └── test/
    └── kustomization.yaml
    └── deployment-patch.yaml
```

```
#base/deployment.yaml
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/4.kustomize
```

```
#base/kustomization.yaml
```

```
resources:
  - deployment.yaml
```

```
#base/overlays/prod/deployment-patch.yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#overlays/test/kustomization.yaml
resources:
  - ../../base
```

```
apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

Kustomize 2

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

  └── test/
    └── kustomization.yaml
    └── deployment-patch.yaml
```

```
#base/deployment.yaml
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/4.kustomize
```

```
#base/kustomization.yaml
```

```
resources:
  - deployment.yaml
```

```
#base/overlays/prod/deployment-patch.yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#overlays/test/kustomization.yaml
resources:
  - ../../base
```

```
apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

oc apply -k overlays/prod

Kustomize 2

```
3.kustomize/
  └── base/
    └── deployment
      └── kustomization.yaml

  └── overlays/
    └── prod/
      └── kustomization.yaml
      └── deployment-patch.yaml

    └── test/
      └── kustomization.yaml
      └── deployment-patch.yaml
```

```
#base/deployment.yaml
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: mysql
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_TCP_PORT
              value: "3306"
          ports:
            - containerPort: 3306
          imagePullPolicy: IfNotPresent
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/4.kustomize
```

```
#base/kustomization.yaml
```

```
resources:
  - deployment.yaml
```

```
#base/overlays/prod/deployment-patch.yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  template:
    spec:
      containers:
        - name: mysql
          resources:
            requests:
              cpu: "500m"
              memory: "512Mi"
            limits:
              cpu: "1000m"
              memory: "1024Mi"
```

```
#overlays/test/kustomization.yaml
resources:
  - ../../base
```

```
apiVersion: kustomize.config.k8s.io/
kind: Kustomization
patches:
  - path: deployment-patch.yaml
```

oc kustomize overlays/prod

oc kustomize overlays/test

oc apply -k overlays/prod

oc apply -k overlays/test

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

Kustomize 3

Extended template manipulation of objects

- Deploy 2 more application for prod- and test-environment with different configmaps and secrets

Kustomize 3

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

Kustomize 3

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

```
oc kustomize overlays/prod
```

Kustomize 3

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

```
oc kustomize overlays/prod
```

```
oc kustomize overlays/test
```

Kustomize 3

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

```
oc kustomize overlays/prod
```

```
oc kustomize overlays/test
```

```
oc apply -k overlays/prod
```

Kustomize 3

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/5.kustomize
```

```
oc kustomize overlays/prod
```

```
oc kustomize overlays/test
```

```
oc apply -k overlays/prod
```

```
oc apply -k overlays/test
```

```
git clone https://github.com/kolinrr/openshift-adv.git
```

```
cd openshift-adv/Day1/6.kustomize
```

Kustomize 4

Deploy CRD + Operator on 3 new namespaces

Kustomize 4

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/6.kustomize
```

Kustomize 4

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/6.kustomize
```

```
oc kustomize overlays/dev
```

Kustomize 4

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/6.kustomize
```

```
oc kustomize overlays/dev
```

```
oc kustomize overlays/test
```

Kustomize 4

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/6.kustomize
```

`oc kustomize overlays/dev`

`oc kustomize overlays/test`

`oc kustomize overlays/prod`

Kustomize 4

```
git clone https://github.com/kolinrr/openshift-adv.git  
cd openshift-adv/Day1/6.kustomize
```

`oc kustomize overlays/dev`

`oc kustomize overlays/test`

`oc kustomize overlays/prod`

`oc apply -k ...` will not work.

Namespaces are missing,
No images in namespace-Registry
No Roles & Rights...

Github Actions & Openshift Deployment Lab

- Register / Login Github <https://github.com/> *avoid caps in name
- Fork <https://github.com/kolinrr/openshift-adv/fork>

Openshift Sandbox

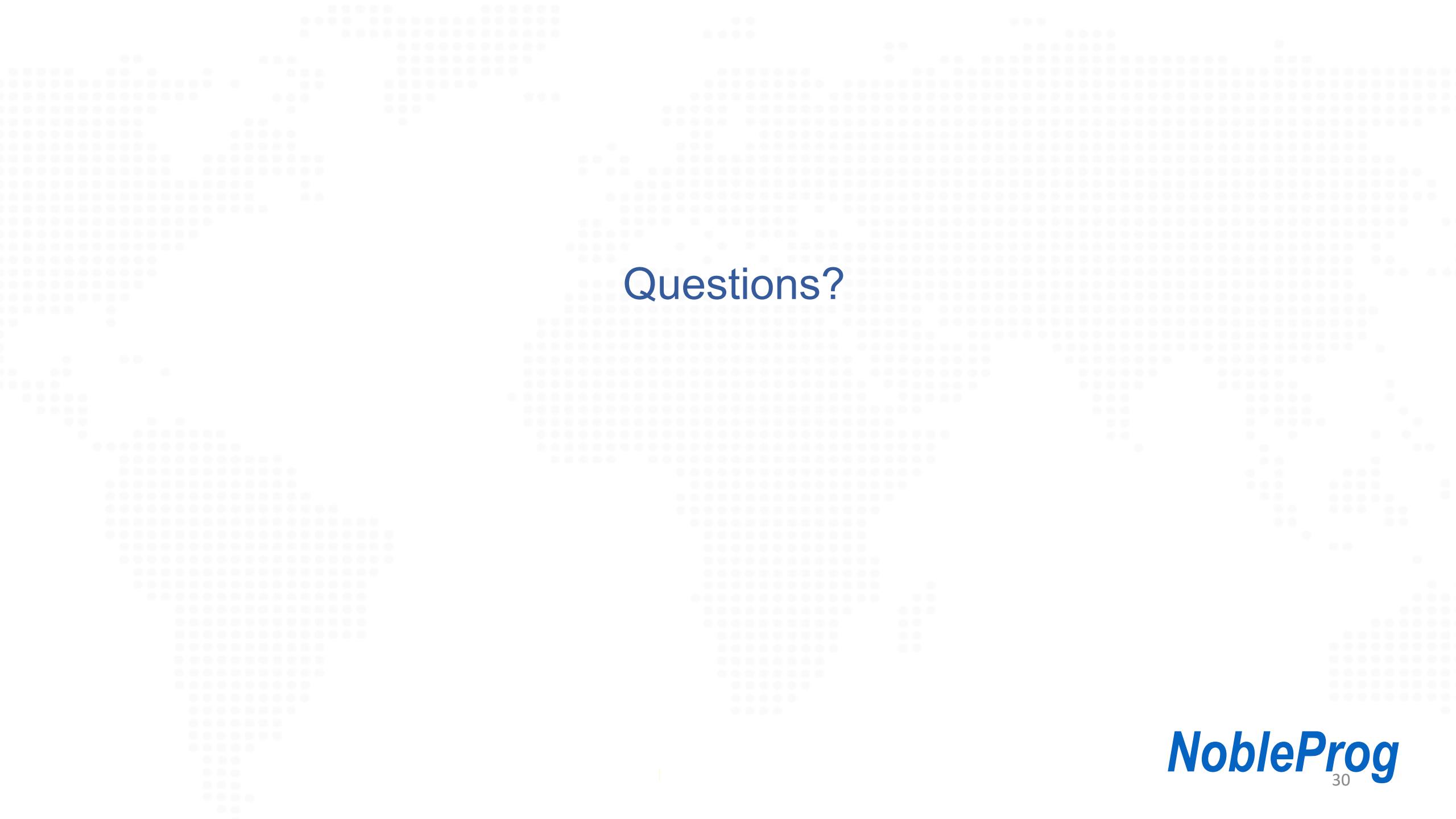
- Get own sandbox: <https://developers.redhat.com/developer-sandbox>

Learn containers, Kubernetes, and OpenShift in your browser

Start exploring in the Developer Sandbox for free

Try Red Hat's products and technologies without setup or configuration.

Start your sandbox for free



Questions?