

K8s Workload

Exploring Pods

Lab: Explore Pod Parameters

Objective:

- Create a Pod using a detailed spec
- Observe behavior with environment variables, probes, scheduling, and volumes
- Inspect effects of Pod restart policy and security context

Steps

1. Create a detailed Pod manifest

Create a file called `detailed-pod.yaml`:

```

apiVersion: v1
kind: Pod
metadata:
  name: full-demo-pod
  labels:
    app: demo
spec:
  containers:
  - name: web
    image: nginx
    ports:
    - containerPort: 80
    env:
    - name: DEMO_MODE
      value: "true"
    resources:
      requests:
        cpu: "100m"
        memory: "64Mi"
      limits:
        cpu: "200m"
        memory: "128Mi"
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
  livenessProbe:
    httpGet:
      path: /
      port: 80
    initialDelaySeconds: 5
    periodSeconds: 10
  readinessProbe:
    httpGet:
      path: /
      port: 80
    initialDelaySeconds: 3
    periodSeconds: 5
  securityContext:
    runAsUser: 1000
    allowPrivilegeEscalation: false
  volumes:
  - name: html
    emptyDir: {}
  nodeSelector:
    kubernetes.io/hostname: minikube
  restartPolicy: Always

```

2. Apply the manifest

```
kubectl apply -f detailed-pod.yaml
```

3. Check Pod status

```
kubectl get pod full-demo-pod -o wide
```

4. Verify liveness and readiness

```
kubectl describe pod full-demo-pod | grep -A5 "Liveness"  
kubectl describe pod full-demo-pod | grep -A5 "Readiness"
```

5. Inspect environment variables

```
kubectl exec full-demo-pod -- printenv | grep DEMO_MODE
```

6. Inspect volume mount

```
kubectl exec full-demo-pod -- ls /usr/share/nginx/html
```

7. Check scheduling rules

```
kubectl get pod full-demo-pod -o=jsonpath="{.spec.nodeSelector}"
```

8. View security context

```
kubectl get pod full-demo-pod -o=jsonpath="{.spec.containers[*].securityContext}"
```

9. Cleanup

```
kubectl delete pod full-demo-pod
```