# Kubernetes Environment Deployment – Staging/production Environment

## 1. Overview

This deployment uses Kustomize overlays to create a complete Kubernetes application environment.   
The following resources were applied under the staging namespace using:

* kubectl apply -k overlays/staging

## 2. Deployment Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Resource Type | Name | Description | Notes / Verification |
| Namespace | staging | Logical boundary for staging workloads. | kubectl get ns |
| ServiceAccount | myapp-sa | Used by pods for secure API access. | kubectl get sa -n staging |
| Role | myapp-role | Defines RBAC permissions for ConfigMaps, Pods, and PVCs. | kubectl describe role myapp-role -n staging |
| RoleBinding | myapp-rolebinding | Binds myapp-role to myapp-sa. | kubectl describe rolebinding myapp-rolebinding -n staging |
| ConfigMap | myapp-config | Stores non-secret environment configurations. | kubectl get cm myapp-config -n staging -o yaml |
| Secret | myapp-secrets | Stores credentials or sensitive data. | kubectl get secret myapp-secrets -n staging |
| PersistentVolumeClaim | myapp-pvc | Provides persistent storage for the application. | kubectl get pvc -n staging |
| Service | myapp-service | Exposes pods internally using ClusterIP. | kubectl get svc -n staging |
| Deployment | myapp | Manages application pod replicas. | kubectl rollout status deploy/myapp -n staging |
| HorizontalPodAutoscaler | myapp-hpa | Scales pods based on CPU usage. | kubectl get hpa -n staging |
| Ingress | myapp-ingress | Routes external HTTP traffic to myapp-service. | kubectl describe ingress myapp-ingress -n staging |
| NetworkPolicy | myapp-allow-nginx | Allows ingress traffic from NGINX pods only. | kubectl describe netpol myapp-allow-nginx -n staging |

## 3. Cluster Logs and Messages

Below is a sample cluster log output (captured during deployment):

$ kubectl apply -k overlays/staging  
namespace/staging created  
serviceaccount/myapp-sa created  
role.rbac.authorization.k8s.io/myapp-role created  
rolebinding.rbac.authorization.k8s.io/myapp-rolebinding created  
configmap/myapp-config created  
secret/myapp-secrets created  
service/myapp-service created  
persistentvolumeclaim/myapp-pvc created  
deployment.apps/myapp created  
horizontalpodautoscaler.autoscaling/myapp-hpa created  
ingress.networking.k8s.io/myapp-ingress created  
networkpolicy.networking.k8s.io/myapp-allow-nginx created

Warnings observed:

- 'patchesStrategicMerge' and 'commonLabels' are deprecated — use 'kustomize edit fix'  
- 'annotation "kubernetes.io/ingress.class"' is deprecated — use 'spec.ingressClassName' instead.  
A black screen with white text

AI-generated content may be incorrect.

## 4. Post-Deployment Verification

Used these commands to verify functionality and health:

* kubectl get pods -n staging -o wide
* A screen shot of a computer

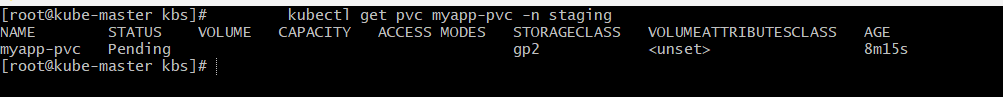
  AI-generated content may be incorrect.
* kubectl rollout status deployment/myapp -n staging

A black background with white text

AI-generated content may be incorrect.

* kubectl logs -l app=myapp -n staging --tail=20
* A screenshot of a computer screen

  AI-generated content may be incorrect.
* kubectl get pvc myapp-pvc -n staging



* kubectl get ingress -n staging
* kubectl describe netpol myapp-allow-nginx -n staging
* A screenshot of a computer

  AI-generated content may be incorrect.

Roles and Role-Bindins

A screenshot of a computer screen

AI-generated content may be incorrect.

Ingress- verification through controller

A computer screen shot of a black screen

AI-generated content may be incorrect.

# Production namespace

A screen shot of a computer

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.