KUBERNETES PRODUCTION TEMPLATES – MASTER SHEET

1. Namespace

Used to logically isolate applications.

```
---
apiVersion: v1
kind: Namespace
metadata:
name: prod-app
labels:
environment: production
---
```

2. ConfigMap

Stores non-sensitive configuration data.

```
---
apiVersion: v1
kind: ConfigMap
metadata:
name: app-config
namespace: prod-app
data:
APP_MODE: "production"
LOG_LEVEL: "info"
---
```

3. Secret

Stores sensitive credentials securely.

```
---
apiVersion: v1
kind: Secret
metadata:
name: app-secret
namespace: prod-app
type: Opaque
stringData:
DB_PASSWORD: "StrongPassword123"
---
```

4. Deployment

Defines how application pods are created and managed.

Includes probes, resource limits, annotations.

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
name: web-app
namespace: prod-app
labels:
app: web-app
spec:
replicas: 3
revisionHistoryLimit: 5
selector:
matchLabels:
app: web-app
template:
metadata:
labels:
app: web-app
annotations:
prometheus.io/scrape: "true"
prometheus.io/port: "80"
spec:
serviceAccountName: web-sa
containers:
- name: nginx
image: nginx:1.25
ports:
- containerPort: 80
env:
- name: APP_MODE
valueFrom:
configMapKeyRef:
name: app-config
key: APP_MODE
```

```
- name: DB_PASSWORD
valueFrom:
secretKeyRef:
name: app-secret
key: DB_PASSWORD
resources:
requests:
cpu: "200m"
memory: "256Mi"
limits:
cpu: "500m"
memory: "512Mi"
livenessProbe:
httpGet:
path: /
port: 80
initialDelaySeconds: 15
periodSeconds: 10
readinessProbe:
httpGet:
path: /
port: 80
initialDelaySeconds: 5
periodSeconds: 5
volumeMounts:
- name: web-storage
mountPath: /usr/share/nginx/html
volumes:
- name: web-storage
persistentVolumeClaim:
claimName: app-pvc
```

---

5. Service

Exposes deployment internally via ClusterIP.

```
apiVersion: v1
kind: Service
```

```
metadata:
name: web-service
namespace: prod-app
spec:
type: ClusterIP
selector:
app: web-app
ports:
- name: http
port: 80
targetPort: 80
```

---

6. Ingress

Routes external traffic to the service.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
name: web-ingress
namespace: prod-app
annotations:
nginx.ingress.kubernetes.io/rewrite-target: /
spec:
rules:
- host: webapp.company.com
http:
paths:
- path: /
pathType: Prefix
backend:
service:
name: web-service
port:
number: 80
```

---

7. PersistentVolume

Provision storage for application data.

```yaml
---
apiVersion: v1
kind: PersistentVolume
metadata:
name: app-pv
spec:
capacity:
storage: 5Gi
accessModes:
- ReadWriteOnce
storageClassName: standard
hostPath:
path: "/data/app"
```

---

8. PersistentVolumeClaim

Application requests storage via PVC.

```yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: app-pvc
namespace: prod-app
spec:
accessModes:
- ReadWriteOnce
storageClassName: standard
resources:
requests:
storage: 5Gi
```

---

9. Horizontal Pod Autoscaler

Scales pods based on CPU/memory.

```yaml
---
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
```

```
metadata:
name: web-hpa
namespace: prod-app
spec:
scaleTargetRef:
apiVersion: apps/v1
kind: Deployment
name: web-app
minReplicas: 3
maxReplicas: 10
metrics:
- type: Resource
resource:
name: cpu
target:
type: Utilization
averageUtilization: 60
---
```

10. ServiceAccount, Role, RoleBinding

Manages RBAC permissions.

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
name: web-sa
namespace: prod-app
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
name: app-role
namespace: prod-app
rules:
- apiGroups: [""]
resources: ["pods", "configmaps"]
verbs: ["get", "list", "watch"]
```

```yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: app-role-binding
namespace: prod-app
subjects:
- kind: ServiceAccount
name: web-sa
roleRef:
kind: Role
name: app-role
apiGroup: rbac.authorization.k8s.io
```