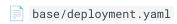
# Lab: Using Kustomize with Kubernetes

### Prerequisites

- Kubernetes cluster (Minikube, Kind, or any cluster)
- kubectl installed
- kustomize installed (or use kubectl kustomize)

## Directory Structure

## Create the Base Configuration



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
   matchLabels:
      app: nginx
  template:
   metadata:
      labels:
       app: nginx
   spec:
      containers:
        - name: nginx
          image: nginx:1.21
          ports:
            - containerPort: 80
```

base/service.yaml

```
apiVersion: v1
kind: Service
metadata:
   name: nginx
spec:
   selector:
   app: nginx
ports:
   - protocol: TCP
     port: 80
     targetPort: 80
```

base/kustomization.yaml

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

Create Overlays

**Dev Overlay** 

overlays/dev/kustomization.yaml resources: - ../../base patches: - path: patch.yaml overlays/dev/patch.yaml apiVersion: apps/v1 kind: Deployment metadata: name: nginx spec: replicas: 1 **Prod Overlay** overlays/prod/kustomization.yaml resources: - ../../base patches: - path: patch.yaml

#### overlays/prod/patch.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
 name: nginx
spec:
 replicas: 4

Apply the Configurations

Dev

```
kubectl apply -k overlays/dev
```

#### Prod

```
kubectl apply -k overlays/prod
```

## Verify

```
kubectl get deployments
kubectl get services
```

You should see an nginx deployment with the correct number of replicas depending on the environment.

## Clean Up

```
kubectl delete -k overlays/dev
# or
kubectl delete -k overlays/prod
```

August 23, 2025