

**Kustomize:** It's a native k8s container tool, which will help to deploy different different configuration in different different environment without losing original configuration.

Let's take one example: When using Kubernetes, multiple teams use multiple environments, such as development, staging, testing, and production, to deploy applications. These environments use applications with minor configuration changes.

Many organizations deploy a single application to multiple data centers for multiple teams and regions. Depending on the load, the organization needs a different number of replicas for every region. The organization might need various configurations that are specific to a data center or team

All these use cases require a single set of manifests with multiple customizations at multiple environments. Kustomize can support such use cases.

**Use Case:** Deploy some configuration like deployment.yml, secret.yml, configmap.yml, service.yml file in dev test prod environment.

# mkdir cust-dir; cd cust-dir

# mkdir base overlays; cd base

# vim deployment.yml

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: pd-app
5 spec:
6   replicas: 2
7   selector:
8     matchLabels:
9       app: bank
10  template:
11    metadata:
12      name: pd-app
13      labels:
14        app: bank
15    spec:
16      containers:
17        - name: nginx-con
18          image: quay.io/redhattraining/hello-world-nginx:v1.0
```

```
# oc create secret generic axis-secret --from-literal name=rudra --from-literal pass=pass123 -  
oyaml --dry-run=client >> secret.yml
```

```
# oc create cm axis-cm --from-literal name=rudra --from-literal pass=pass123 -oyaml --dry-  
run=client >> configmap.yml
```

```
# vim service.yml
```

```
1 apiVersion: v1  
2 kind: Service  
3 metadata:  
4   name: pd-svc  
5 spec:  
6   selector:  
7     app: bank  
8   type: ClusterIP  
9   ports:  
10    - port: 8080  
11      targetPort: 8080  
12      protocol: TCP  
13
```

```
# vim kustomization.yml
```

```
1 apiVersion: kustomize.config.k8s.io/v1beta1
2 kind: Kustomization
3 metadata:
4   name: axis-kustomize
5 resources:
6   - deployment.yml
7   - secret.yml
8   - configmap.yml
9   - service.yml
```

# cd ../overlays

# mkdir dev test prod

# vim dev/kustomization.yaml

```
1 apiVersion: kustomize.config.k8s.io/v1beta1
2 kind: Kustomization
3 metadata:
4   name: dev-kustomize
5 resources:
6   - ../../base
```

# vim test/kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
metadata:
  name: test-kustomize
resources:
  - ../../base
```

# vim prod/kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
metadata:
  name: prod-kustomize
resources:
- ../../base
```

# tree cust-dir

```
├── base
│   ├── configmap.yml
│   ├── deployment.yml
│   ├── kustomization.yml
│   ├── secret.yml
│   └── service.yml
└── overlays
    ├── dev
    │   └── kustomization.yml
    ├── prod
    │   └── kustomization.yml
    └── test
        └── kustomization.yml
```

For verification kustomize directories run below command

```
# kubectl kustomize overlays/dev -n dev
```

```
# kubectl kustomize overlays/dev -n test
```

```
# kubectl kustomize overlays/dev -n prod
```

Create these configurations on dev test prod environment.

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

```
# kubectl apply -k overlays/test -n test
```

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

### Do verification

Now testing team want to do some modification on deployment and service as well want to add one configmap in their environment.

```
# vim cust-dir/overlays/test/kustomization.yaml
```

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
metadata:
  name: test-kustomize
resources:
- ../../base
patches:
- patch: |-
    - op: replace
      path: /spec/replicas
      value: 4
  target:
    kind: Deployment
    name: pd-app
- patch: |-
    - op: replace
      path: /spec/type
      value: NodePort
  target:
    kind: Service
    name: pd-svc

configMapGenerator:
- name: axis-bank-2
  literals:
    - msg="Hello Class"
    - enable="true"
[student@workstation axis-cust]$
```

```
# kubectl apply -k overlays/test -n test
```

Do Verification on test namespace

- In service conf. type got changed
- In Deployment replicas got changed

- New configmap got created in test namespace

Delete all configuration whatever created by kustomization.yaml file by below command

**# oc delete -k overlay/test -n test**