



KodeKloud

Curriculum

Overview

Kustomize Basics

Generators

Imperative Commands

Final Assignment

Kustomize

Installation

syntax

transformers

patches

overlays

components

Edit/Set

CI/CD

generators

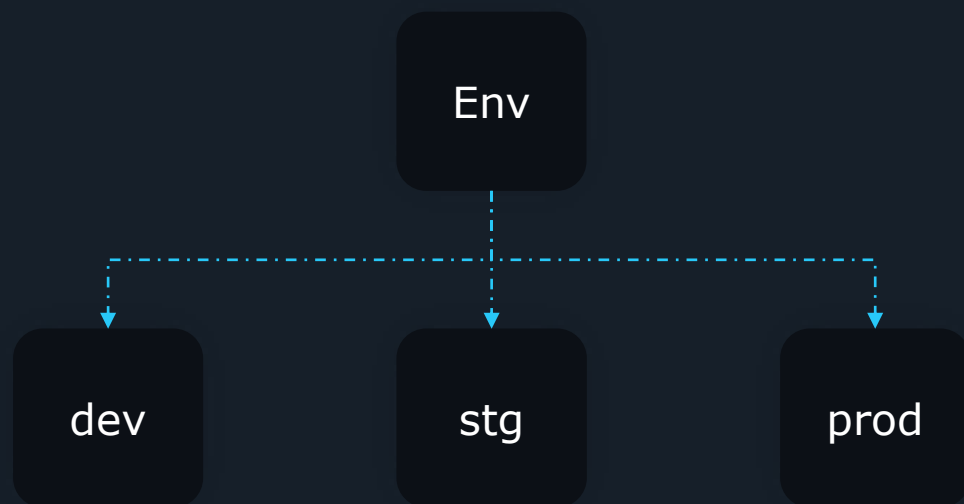


KodeKloud

Kustomize Problem Statement & idealogy



Why Kustomize



nginx-depl.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: nginx
  template:
    metadata:
      labels:
        component: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
```

Why Kustomize

 dev

 nginx-depl.yml

 stg

 nginx-depl.yml

 prod

 nginx-depl.yml

dev/nginx.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: nginx
  template:
    metadata:
      labels:
        component: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
```

stg/nginx.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      component: nginx
  template:
    metadata:
      labels:
        component: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
```

prod/nginx.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 5
  selector:
    matchLabels:
      component: nginx
  template:
    metadata:
      labels:
        component: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
```

Deploying Dev

> _

```
$ kubectl apply -f dev/  
deployment.apps/nginx-deployment created
```



nginx

A diagram showing a container labeled 'nginx'. The container is represented by a rounded rectangle with a light blue border. Inside the container, at the bottom, is a red rounded rectangle with the text 'nginx' in white.

Deploying Stg

> _

```
$ kubectl apply -f stg/  
deployment.apps/nginx-deployment created
```

nginx

nginx

Why Kustomize

 dev

 nginx-depl.yml

 stg

 nginx-depl.yml

 prod

 nginx-depl.yml

Why Kustomize

 dev

 nginx-depl.yml

 service.yml

 stg

 nginx-depl.yml

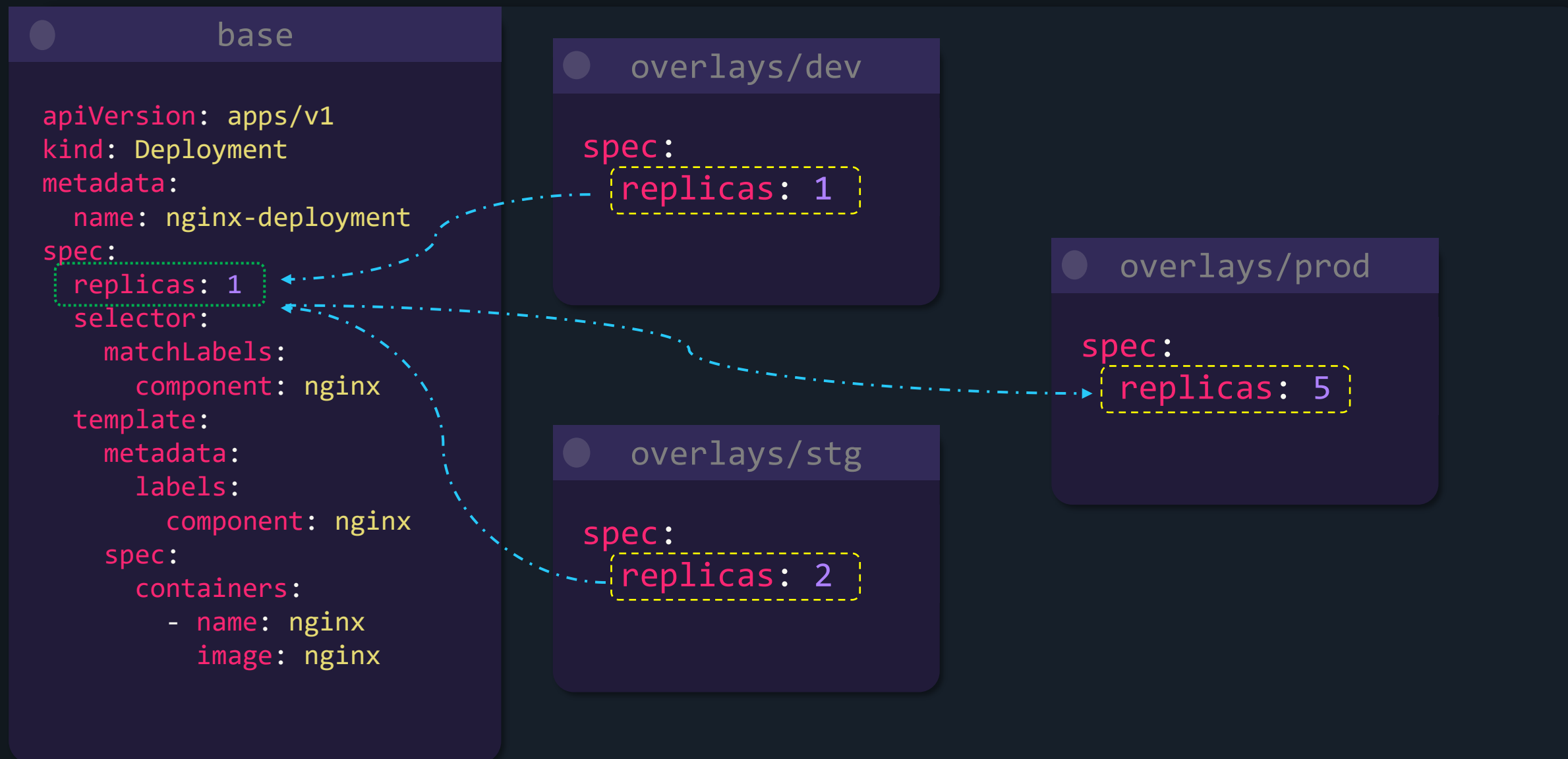
 service.yml

 prod

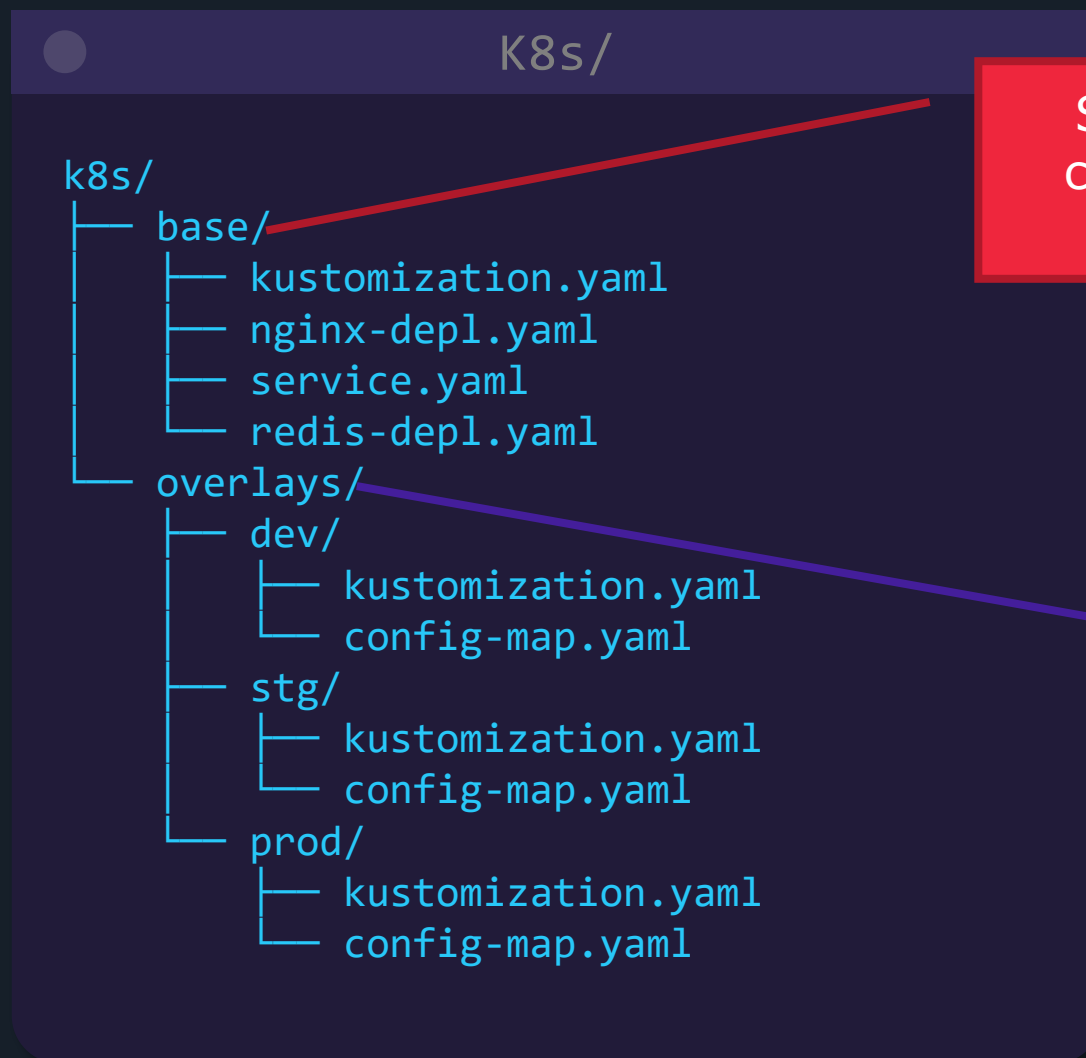
 nginx-depl.yml

 service.yml

Overlays



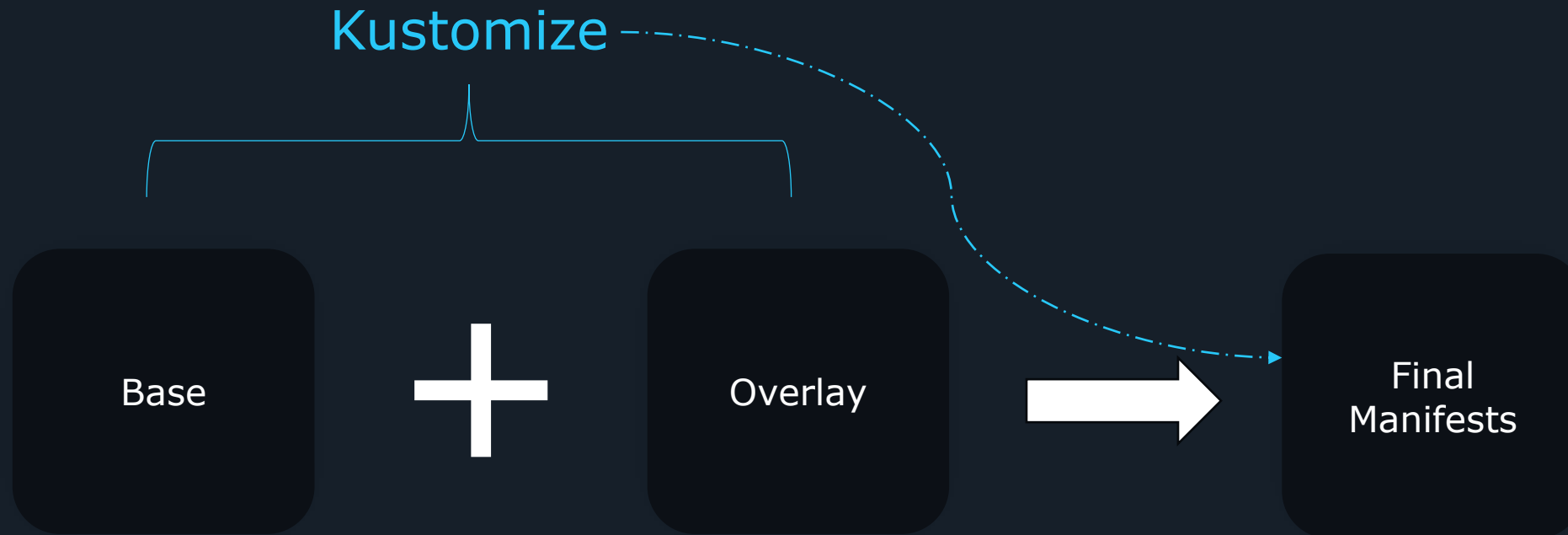
Folder Structure



Share or default
configs across all
environments

Environment specific
configurations that add or
modify the base configs

Kustomize





KodeKloud

Kustomize vs Helm



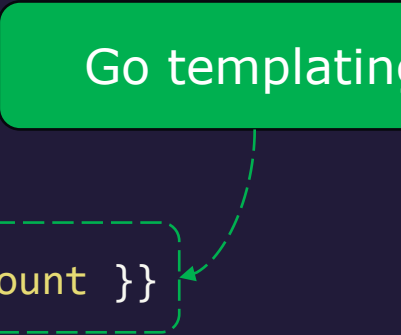
Kustomize vs Helm

Helm makes use of go templates to allow assigning variables to properties.

Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Values.name }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: {{ .Values.name }}
  template:
    metadata:
      labels:
        app: {{ .Values.name }}
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "nginx:{{ .Values.image.tag }}"
```

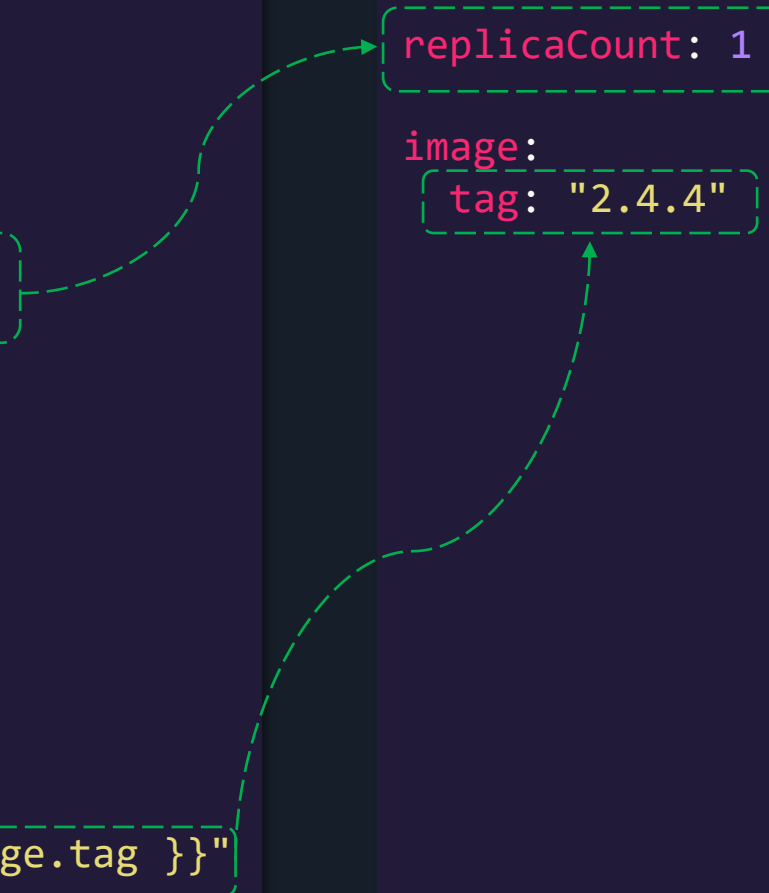
Go templating



Kustomize vs Helm

Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Values.name }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: {{ .Values.name }}
  template:
    metadata:
      labels:
        app: {{ .Values.name }}
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "nginx:{{ .Values.image.tag }}"
```



values.yaml

```
replicaCount: 1
```

```
image:
  tag: "2.4.4"
```

Kustomize vs Helm

Deployment.yaml

```
k8s/
├── environments/
│   ├── values.dev.yaml
│   ├── values.stg.yaml
│   └── values.prod.yaml
└── templates/
    ├── nginx-deployment.yaml
    ├── nginx-service.yaml
    ├── db-deployment.yaml
    └── db-service.yaml
```

Kustomize vs Helm

- Helm is more than just a tool to customize configurations on a per environment basis. Helm is also a package manager for your App
- Helm provides extra features like conditionals, loops, functions, and hooks
- Helm templates are not valid YAML as they use go templating syntax
 - Complex templates become hard to read

Kustomize

- Kustomize comes built-in with kubectl so no other packages need to be installed
 - May still want to install the kustomize cli to get the latest version. Kubectl doesn't come with the latest version
- Does not require learning how to use any complex & hard to read templating systems(like helm)
- Every artifact the Kustomize uses is plain YAML and can be validated and processed as such



KodeKloud

Installation/Setup



Install Kustomize

```
>_  
$ curl -s "https://raw.githubusercontent.com/kubernetes-  
sigs/kustomize/master/hack/install_kustomize.sh" | bash
```

kubectl



Install Kustomize

>_

```
$ kustomize version --short  
{kustomize/v4.4.1 2021-11-11T23:36:27Z }
```



KodeKloud

kustomization.yaml
file



Kustomization.yaml File

 k8s


 nginx-depl.yml

 nginx-service.yml

Kustomization.yaml File

 k8s

 nginx-depl.yml

 nginx-service.yml

 kustomization.yaml

Kustomization.yaml

```
# kubernetes resources to be managed by  
kustomize
```

```
resources:
```

- nginx-deployment.yaml
- nginx-service.yaml

```
#Customizations that need to be made
```

```
commonLabels:
```

```
  company: KodeKloud
```

Kustomize Build

```
> _
```

```
$ kustomize build k8s/
```

```
kustomize build k8s/
apiVersion: v1
kind: Service
metadata:
  labels:
    company: KodeKloud
  name: nginx-loadbalancer-service
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3000
  selector:
    company: KodeKloud
    component: nginx
  type: LoadBalancer
```

service

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    company: KodeKloud
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      company: KodeKloud
      component: nginx
  template:
    metadata:
      labels:
        company: KodeKloud
        component: nginx
    spec:
      containers:
        - image: nginx
          name: nginx
```

nginx

Kustomize

- Kustomize looks for a kustomization file which contains:
 - List of all the Kubernetes manifests kustomize should manage
 - All of the customizations that should be applied
- The *kustomize build* command combines all the manifests and applies the defined transformations
- The *kustomize build* command does not apply/deploy the Kubernetes resources to a cluster
 - The output needs to be redirected to the *kubectl apply* command

Kustomize CLI

>_

```
$ kustomize build k8s/
```

```
$ kubectl kustomize k8s/
```




KodeKloud

Kustomize Output



Apply Kustomize Configs

>_

```
$ kustomize build k8s/
```

Doesn't
apply/deploy
config

```
$ kustomize build k8s/ | kubectl apply -f -
```

```
service/nginx-loadbalancer-service created  
deployment.apps/nginx-deployment created
```

Redirect

```
$ kubectl apply -k k8s/
```

```
service/nginx-loadbalancer-service created  
deployment.apps/nginx-deployment created
```

kubernetes



Service

Delete with Kustomize

>_

```
$ kustomize build k8s/ | kubectl delete -f -
```

```
service "nginx-loadbalancer-service" deleted  
deployment.apps "nginx-deployment" deleted
```

```
$ kubectl delete -k k8s/
```

```
service "nginx-loadbalancer-service" deleted  
deployment.apps "nginx-deployment" deleted
```

kubernetes



Service



KodeKloud

Kustomize ApiVersion & Kind



apiVersion & kind



kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
```

```
# kubernetes resources to be managed by
kustomize
```

```
resources:
```

- nginx-depl.yaml
- nginx-service.yaml

```
#Customizations that need to be made
```

```
commonLabels:
```

```
  company: KodeKloud
```



KodeKloud

Managing Directories



Multiple Directories

 k8s

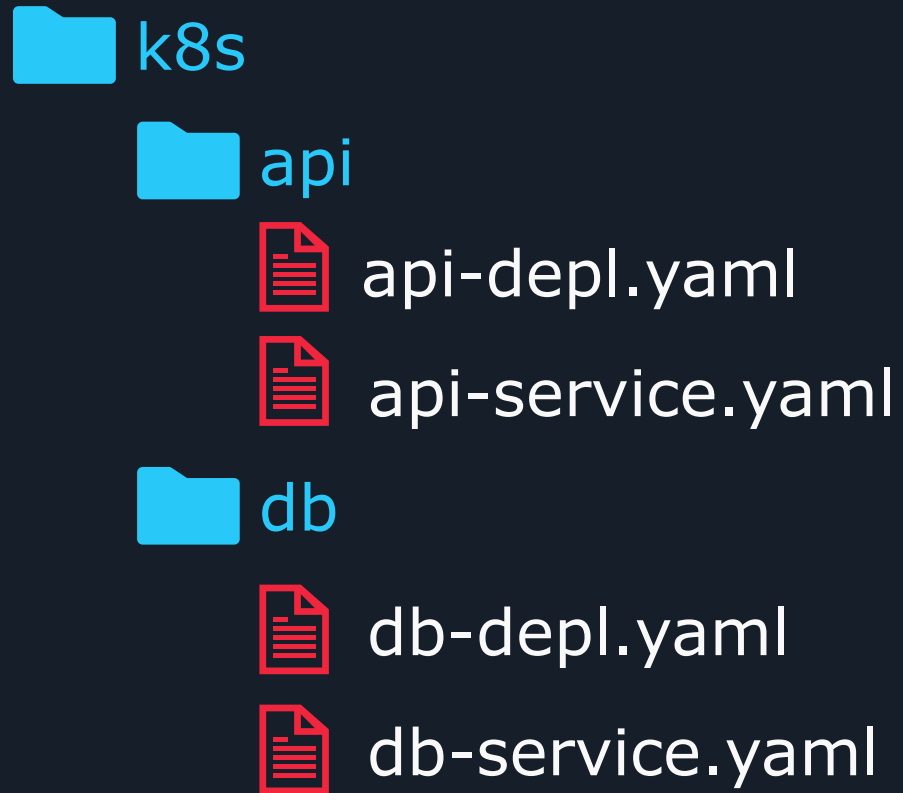
 api-depl.yaml

 api-service.yaml

 db-depl.yaml

 db-service.yaml

Multiple Directories



Multiple Directories

>_

```
$ kubectl apply -f k8s/api/
```

```
$ kubectl apply -f k8s/db/
```

 k8s api api-depl.yaml api-service.yaml db db-depl.yaml db-service.yamlAPI
DeplAPI
ServiceDB
DeplDB
Service

Multiple Directories

k8s



kustomization.yaml



api



api-depl.yaml



api-service.yaml



db



db-depl.yaml



db-service.yaml

Kustomization.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
```

```
# kubernetes resources to be managed by
kustomize
```

```
resources:
```

- api/api-depl.yaml
- api/api-service.yaml
- db/db-depl.yaml
- db/db-service.yaml

Apply Kustomize Configs

>_

```
$ kustomize build k8s/ | kubectl apply -f -
```

```
service/api-service created  
service/db-service created  
deployment.apps/api-deployment created  
deployment.apps/db-deployment created
```

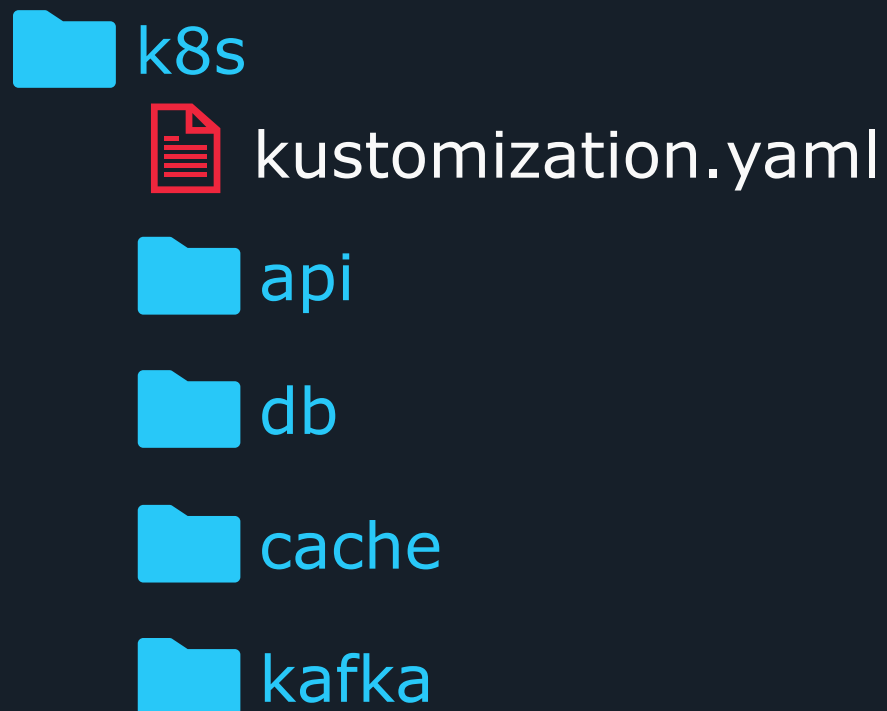
```
$ kubectl apply -k k8s/
```

```
service/api-service created  
service/db-service created  
deployment.apps/api-deployment created  
deployment.apps/db-deployment created
```

kubernetes

API
DeplAPI
ServiceDB
DeplDB
Service

Multiple Directories



```
Kustomization.yml

apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

# kubernetes resources to be managed by
kustomize
resources:
  - api/api-depl.yaml
  - api/api-service.yaml
  - db/db-depl.yaml
  - db/db-service.yaml
  - cache/redis-depl.yaml
  - cache/redis-service.yaml
  - cache/redis-config.yaml
  - kafka/kafka-depl.yaml
  - kafka/kafka-service.yaml
  - kafka/kafka-config.yaml
```

Multiple Directories

 k8s

kustomization.yaml



api



kustomization.yaml



db



kustomization.yaml



cache



kustomization.yaml



kafka



kustomization.yaml

k8s/kustomization.yaml

resources:

- api/
- db/
- cache/
- kafka/

k8s/db/kustomization.yaml

resources:

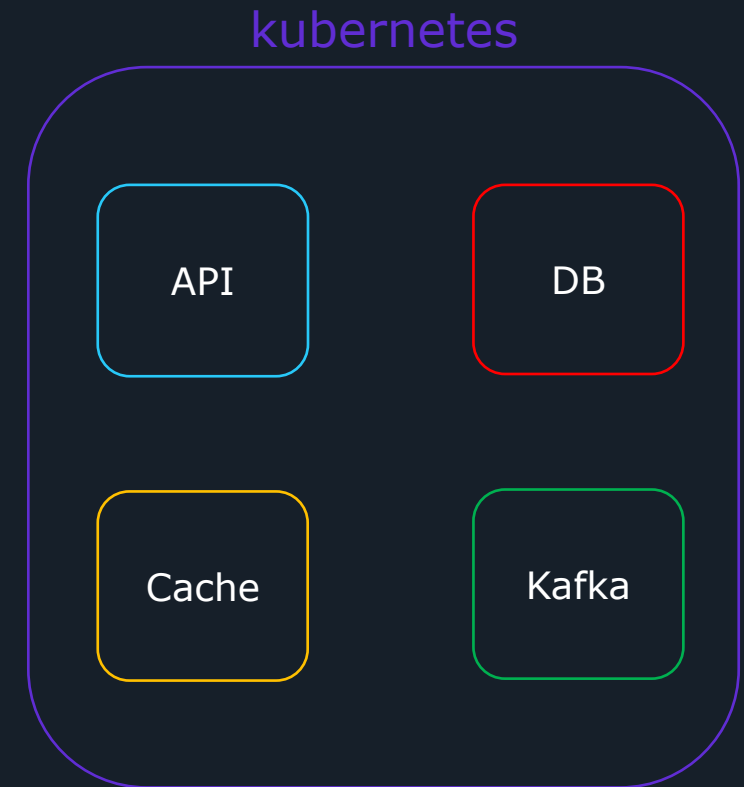
- db-depl.yaml
- db-service.yaml

Apply Kustomize Configs

```
>_
```

```
$ kustomize build k8s/ | kubectl apply -f -
```

```
$ kubectl apply -k k8s/
```





KodeKloud

Common Transformers



Transformers

db-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: db-service
spec:
  selector:
    component: db-depl
  ports:
    - protocol: "TCP"
      port: 27017
      targetPort: 27017
  type: LoadBalancer
```

Common Transformers

db-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment-dev
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
      org: KodeKloud
  template:
    metadata:
      labels:
        component: api
        org: KodeKloud
    spec:
      containers:
        - name: nginx
          image: nginx
```

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    org: KodeKloud
  name: api-service-dev
spec:
  selector:
    component: api
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 3000
  type: LoadBalancer
```

Common Transformations

- `commonLabel` – adds a label to all Kubernetes resources
- `namePrefix/Suffix` – adds a common prefix-suffix to all resource names
- `Namespace` – adds a common namespace to all resources
- `commonAnnotations` – adds an annotation to all resources

CommonLabel Transformer

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    org: KodeKloud
  name: api-service
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 3000
  selector:
    component: api
    org: KodeKloud
  type: LoadBalancer
```

Kustomization.yaml

```
commonLabels:
  org: KodeKloud
```



Namespace Transformer

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    branch: master
  labels:
    org: KodeKloud
    name: api-service
    namespace: lab
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3000
  selector:
    component: api
    org: KodeKloud
  type: LoadBalancer
```

Kustomization.yaml

```
namespace: lab
```


Name Prefix/suffix Transformer

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: KodeKloud-api-service-dev
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3000
  selector:
    component: api
  type: LoadBalancer
```

Kustomization.yaml

```
namePrefix: KodeKloud-
nameSuffix: -dev
```

KodeKloud-api-service-dev

CommonAnnotations Transformer

db-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    branch: master
  labels:
    org: KodeKloud
    name: api-service
    namespace: auth
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3000
  selector:
    component: api
    org: KodeKloud
  type: LoadBalancer
```

Kustomization.yaml

```
commonAnnotations:
  branch: master
```



KodeKloud

Image Transformers



Image Transformer

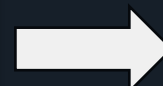
● Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: nginx
```

+

kustomization.yaml

```
images:
- name: nginx
  newName: haproxy
```



● Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: haproxy
```

Image Transformer

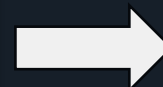
Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: nginx
```

+

kustomization.yaml

```
images:
- name: nginx
  newTag: 2.4
```



Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: nginx:2.4
```

Image Transformer

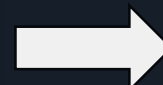
● Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: nginx
```

+

kustomization.yaml

```
images:
- name: nginx
  newName: haproxy
  newTag: 2.4
```



● Web-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: web
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - name: web
          image: haproxy:2.4
```



KodeKloud

Patches Intro



Patches

- Kustomize patches provide another method to modifying Kubernetes configs
- Unlike common transformers, patches provide a more “surgical” approach to targeting one or more specific sections in a Kubernetes resource
- To create a patch 3 parameters must be provided:
 - Operation Type: add/remove/replace
 - Target: What resource should this patch be applied on
 - Kind
 - Version/Group
 - Name
 - Namespace
 - labelSelector
 - AnnotationSelector
 - Value: What is the value that will either be replaced or added with(only needed for add/replace operations)

Patches

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization.yaml

```
patches:
  - target:
      kind: Deployment
      name: api-deployment
    patch: |-
      - op: replace
        path: /metadata/name
        value: web-deployment
```

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

Patches

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

Kustomization

```
patches:
- target:
    kind: Deployment
    name: api-deployment
  patch: |-
    - op: replace
      path: /spec/replicas
      value: 5
```

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 5
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```



KodeKloud

Different Types of Patches



Json 6902 vs Strategic Merge Patch

Json 6902 Patch

```
Kustomization

patches:
- target:
    kind: Deployment
    name: api-deployment
  patch: |-
    - op: replace
      path: /spec/replicas
      value: 5
```

strategic merge patch

```
kustomization

patches:
- patch: |-
    apiVersion: apps/v1
    kind: Deployment
    metadata:
      name: api-deployment
    spec:
      replicas: 5
```

<https://datatracker.ietf.org/doc/html/rfc6902>

Json 6902 patch Inline vs Separate File

Inline

```
● Kustomization

patches:
- target:
    kind: Deployment
    name: api-deployment
  patch: |-
    - op: replace
      path: /spec/replicas
      value: 5
```

Separate File

```
● Kustomization

patches:
- path: replica-patch.yaml
  target:
    kind: Deployment
    name: nginx-deployment
```

```
● replica-patch.yaml

- op: replace
  path: /spec/replicas
  value: 5
```


Strategic merge patch Inline vs Separate File

Inline

```
● Kustomization

patches:
- patch: |-
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: api-deployment
  spec:
    replicas: 5
```

Separate File

The diagram illustrates the 'Separate File' approach. It shows two panels within a dashed blue border. The left panel, titled 'Kustomization', contains a 'patches' list with a single entry '- replica-patch.yaml'. A dashed green arrow points from this entry to the right panel. The right panel, titled 'replica-patch.yaml', contains the following YAML content:

```
● replica-patch.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 5
```



KodeKloud

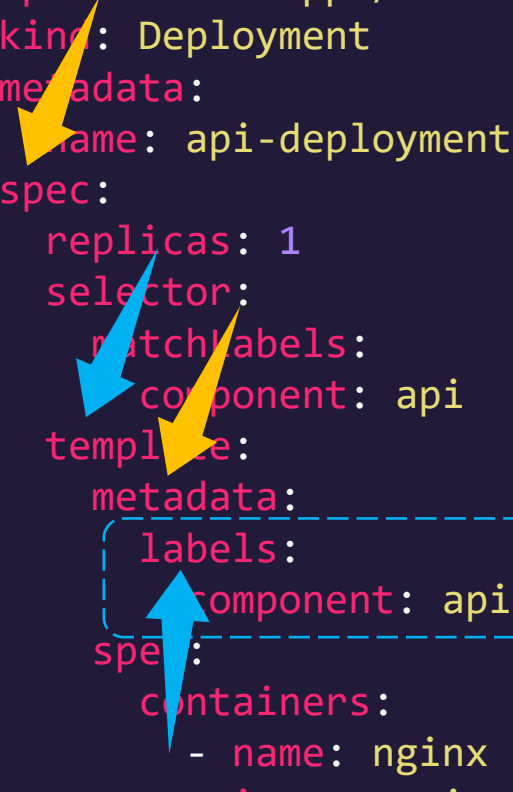
Patches Dictionary



Replace Dictionary Json6902

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```



The diagram illustrates the replacement of the 'component' label value from 'api' to 'web'. A yellow arrow points from the original 'component: api' label in the 'api-depl.yaml' file to the 'path' field of the patch in the 'kustomization' file. Another yellow arrow points from the 'value: web' in the patch to the 'component: web' label in the updated 'api-depl.yaml' file. Blue arrows indicate the flow of the original configuration from the first file to the patch and then to the second file.

kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
      patch: |-
        - op: replace
          path:
            /spec/template/metadata/
            labels/component
          value: web
```



The diagram shows the patch operation. A yellow arrow points from the 'path' field of the patch to the 'component' label in the original 'api-depl.yaml' file. Another yellow arrow points from the 'value: web' in the patch to the 'component: web' label in the updated 'api-depl.yaml' file. A blue arrow points from the 'deployment' field of the patch to the 'api-depl.yaml' file.

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: web
    spec:
      containers:
        - image: nginx
          name: nginx
```



The diagram shows the final state of the deployment. A yellow arrow points from the 'value: web' in the patch to the 'component: web' label in the updated 'api-depl.yaml' file. A blue arrow points from the 'api-depl.yaml' file to the 'kustomization' file.

Replace Dictionary Strategic Merge Patch

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization

```
patches:
- label-patch.yaml
```

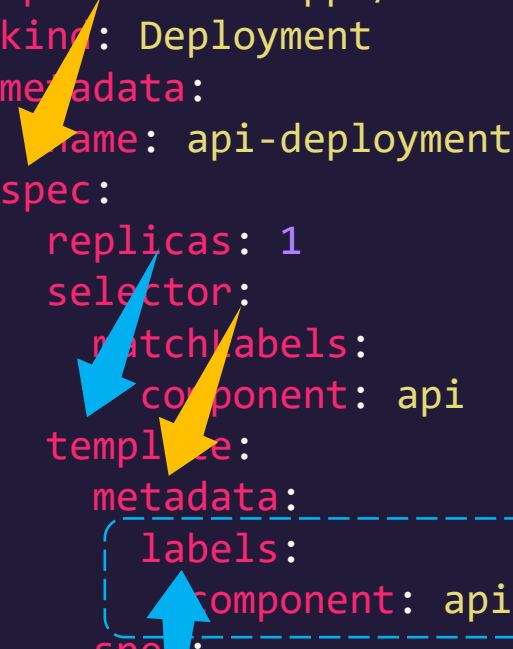
label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    metadata:
      labels:
        component: web
```

Add Dictionary Json6902

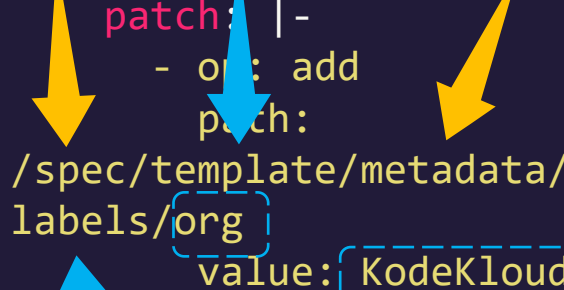
api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```



kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
      patch: |-
        - op: add
          path:
            /spec/template/metadata/
            labels/org
          value: KodeKloud
```



api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
        org: KodeKloud
    spec:
      containers:
        - image: nginx
          name: nginx
```




Add Dictionary Strategic Merge Patch

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization

```
patches:
- label-patch.yaml
```



label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    metadata:
      labels:
        org: kodekloud
```



Remove Dictionary Json6902

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        org: KodeKloud
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
      patch: |-
        - op: remove
          path: /spec/template/metadata/labels/org
```

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - image: nginx
          name: nginx
```


Remove Dictionary Strategic Merge Patch

api-depl.yaml

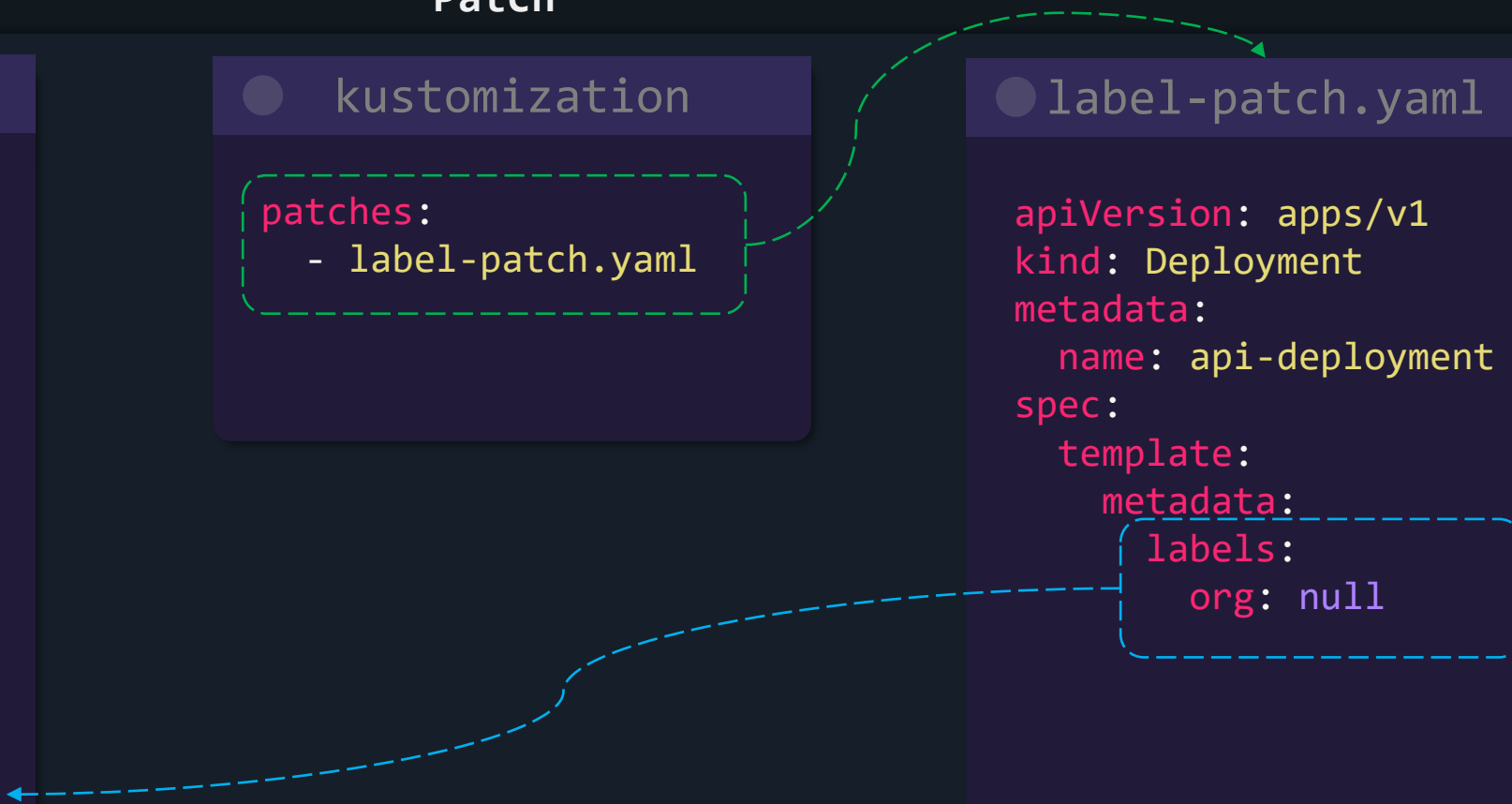
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        org: KodeKloud
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization

```
patches:
- label-patch.yaml
```

label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    metadata:
      labels:
        org: null
```





KodeKloud

Patches list



Replace List Json6902

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

List →

kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
      patch: |-
        - op: replace
          path:
            /spec/template/spec/containers/0
            value:
              name: haproxy
              image: haproxy
```

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - image: haproxy
          name: haproxy
```

Replace List Strategic Merge Patch

api-depl.yaml

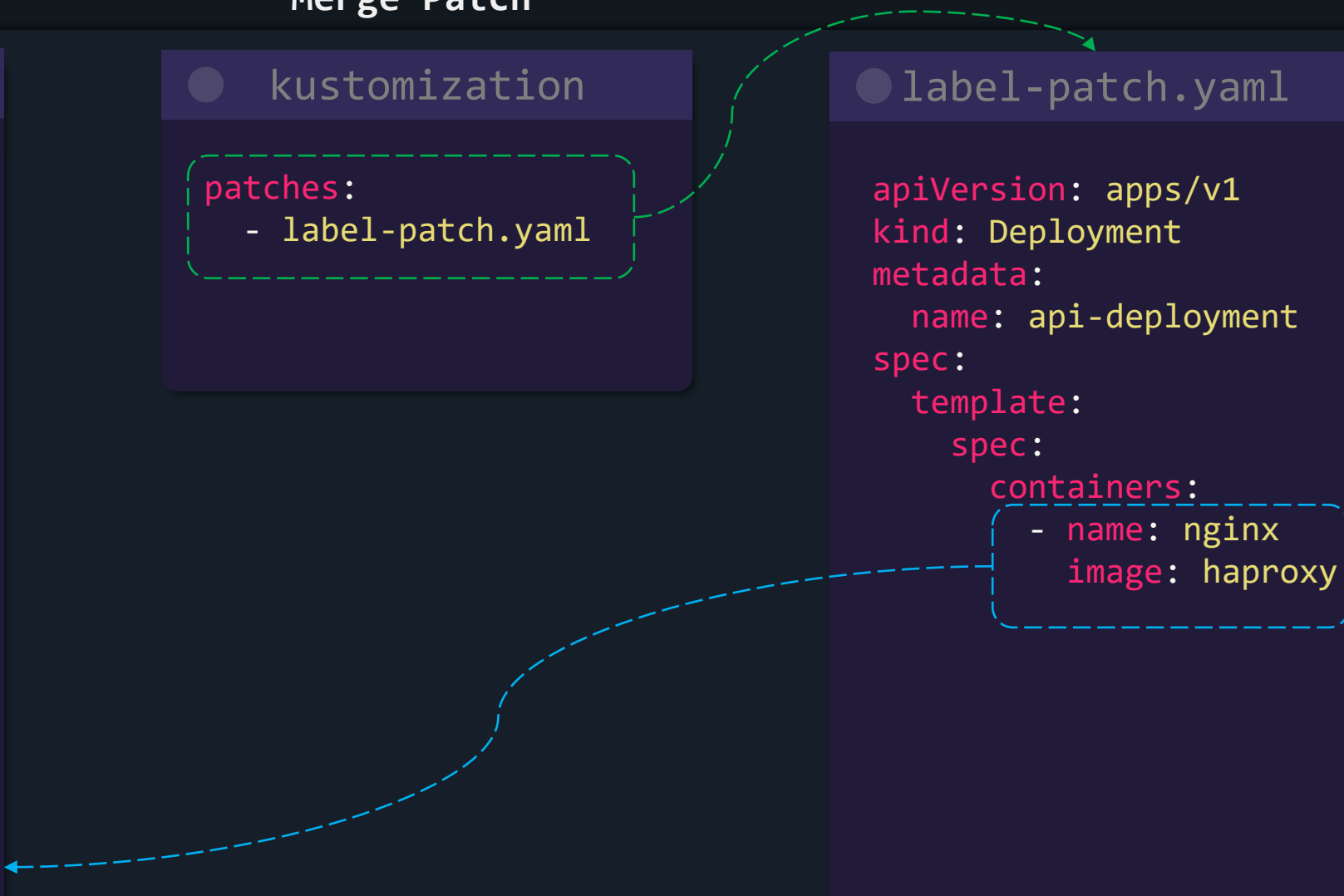
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

kustomization

```
patches:
- label-patch.yaml
```

label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    spec:
      containers:
        - name: nginx
          image: haproxy
```



Add List Json6902

● api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: nginx
          image: nginx
```

● kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
    patch: |-
      - op: add
        path:
          /spec/template/spec/containers/-
        value:
          name: haproxy
          image: haproxy
```

Append to
end of list

● api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - image: nginx
          name: nginx
        - image: haproxy
          name: haproxy
```

Add List Strategic Merge Patch

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: web
          image: nginx
```

kustomization

```
patches:
  - label-patch.yaml
```

label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    spec:
      containers:
        - name: haproxy
          image: haproxy
```

Delete List Json6902

● api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: web
          image: nginx
        - name: database
          image: mongo
```

● kustomization

```
patches:
  - target:
      kind: Deployment
      name: api-
    deployment
    patch: |-
      - op: remove
        path:
          /spec/template/spec/containers/1
```

Index of
container
to delete

● api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - image: nginx
          name: web
```


Delete List Strategic Merge Patch

api-depl.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      component: api
  template:
    metadata:
      labels:
        component: api
    spec:
      containers:
        - name: web
          image: nginx
        - name: database
          image: mongo
```

kustomization

```
patches:
- label-patch.yaml
```

label-patch.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: api-deployment
spec:
  template:
    spec:
      containers:
        - $patch: delete
          name: database
```

Delete Directive specifies which container to delete

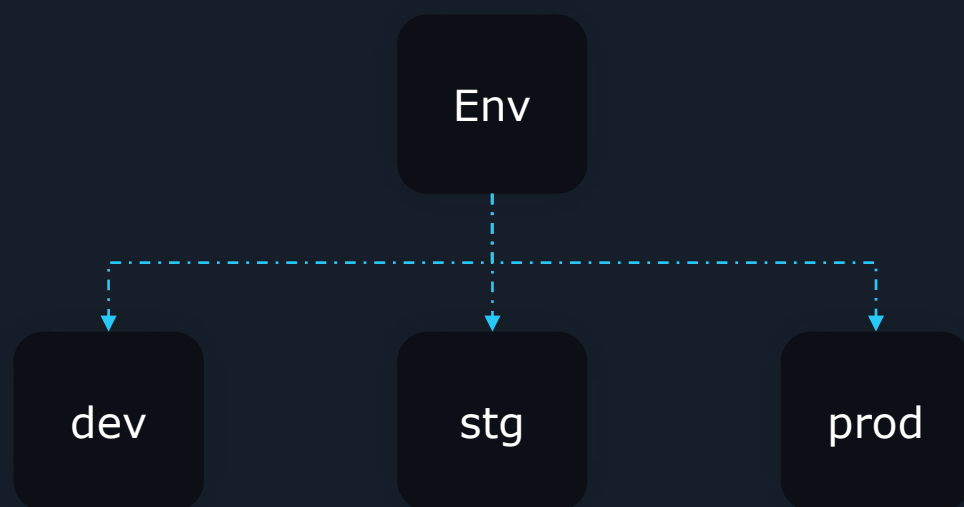


KodeKloud

Overlays



Overlays

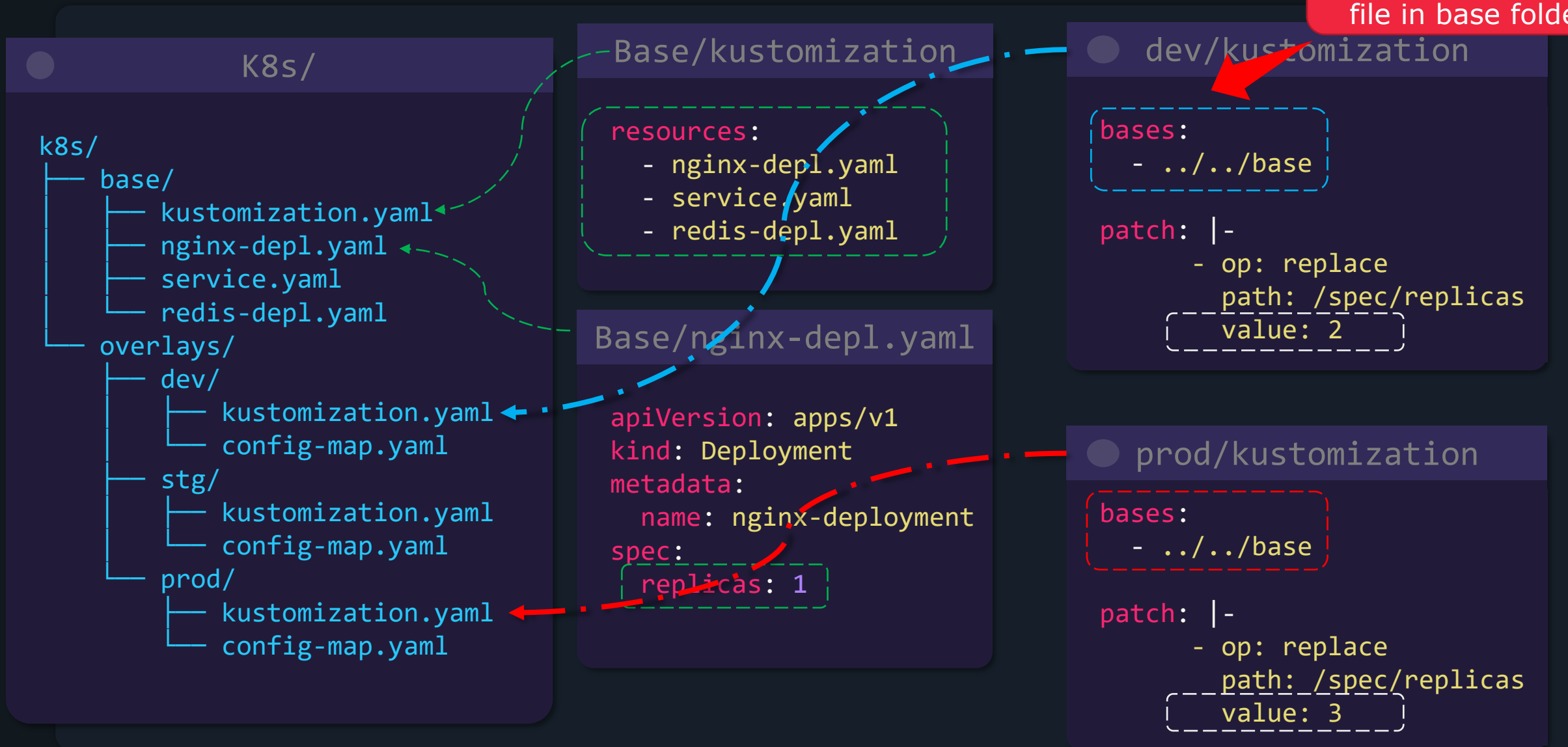


Environment specific configurations that add or modify base configs

```
k8s/  
├── base/  
│   ├── kustomization.yaml  
│   ├── nginx-depl.yaml  
│   ├── service.yaml  
│   └── redis-depl.yaml  
└── overlays/  
    ├── dev/  
    │   ├── kustomization.yaml  
    │   └── config-map.yaml  
    ├── stg/  
    │   ├── kustomization.yaml  
    │   └── config-map.yaml  
    └── prod/  
        ├── kustomization.yaml  
        └── config-map.yaml
```

Share or default configs across all environments

Overlays



Overlays

```
k8s/
├── base/
│   ├── kustomization.yaml
│   ├── nginx-depl.yaml
│   ├── service.yaml
│   └── redis-depl.yaml
└── overlays/
    ├── dev/
    │   ├── kustomization.yaml
    │   ├── config-map.yaml
    │   └── volume.yaml
    ├── stg/
    │   ├── kustomization.yaml
    │   └── config-map.yaml
    └── prod/
        ├── kustomization.yaml
        ├── config-map.yaml
        └── grafana-depl.yaml
```

```
prod/kustomization
```

```
bases:
- ../../base

resources:
- grafana-depl.yaml

patch: |-
- op: replace
  path: /spec/replicas
  value: 2
```

```
k8s/
├── base/
│   ├── kustomization.yaml
│   └── db/
│       ├── db-depl.yaml
│       ├── db-service.yaml
│       └── kustomization.yaml
│   └── api/
│       ├── api-depl.yaml
│       ├── api-service.yaml
│       └── kustomization.yaml
└── overlays/
    ├── dev/
    │   ├── kustomization.yaml
    │   └── db/
    │       ├── db-patch.yaml
    │       └── kustomization.yaml
    │   └── api/
    │       ├── api-patch.yaml
    │       └── kustomization.yaml
    └── prod/
        ├── kustomization.yaml
        ├── db/
        │   ├── db-patch.yaml
        │   └── kustomization.yaml
        └── api/
            ├── api-patch.yaml
            └── kustomization.yaml
```



KodeKloud

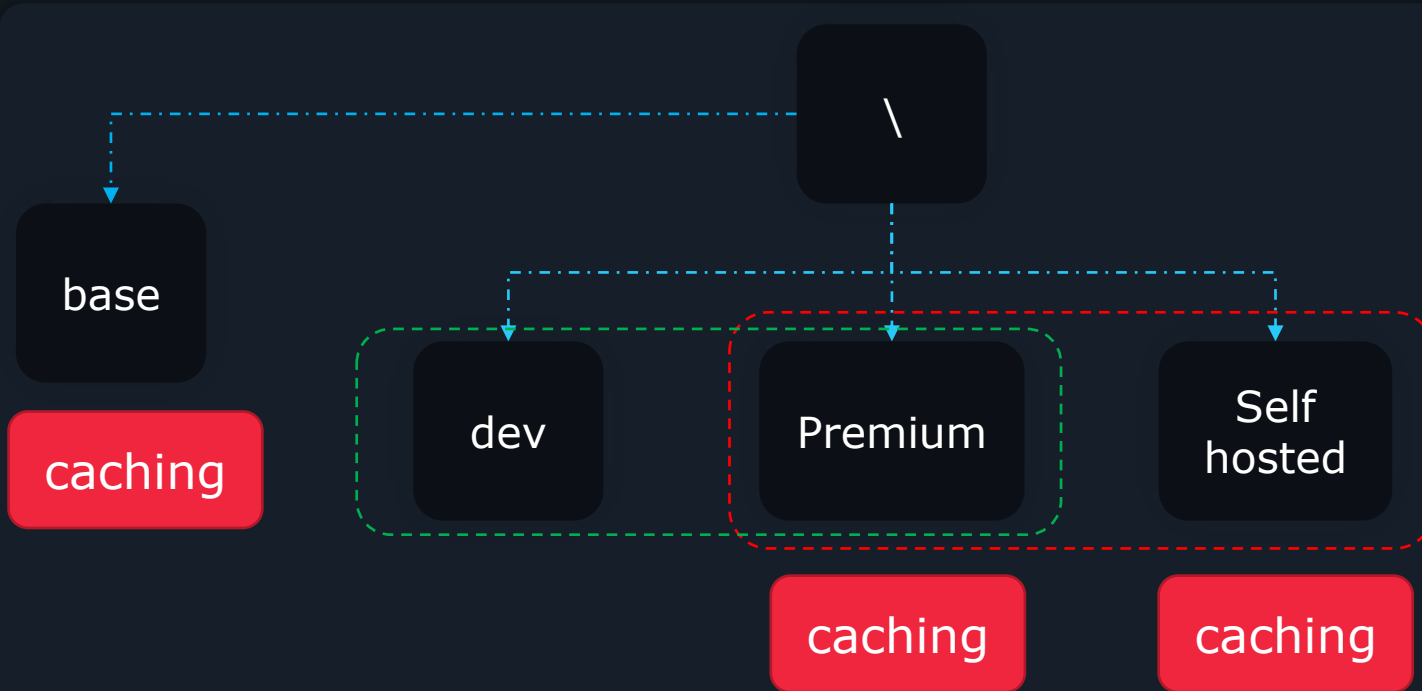
Components



Components

- Components provide the ability to define reusable pieces of configuration logic(resources + patches) that can be included in multiple overlays
- Components are useful in situations where applications support multiple optional features that need to be enabled only in a subset of overlays

Components



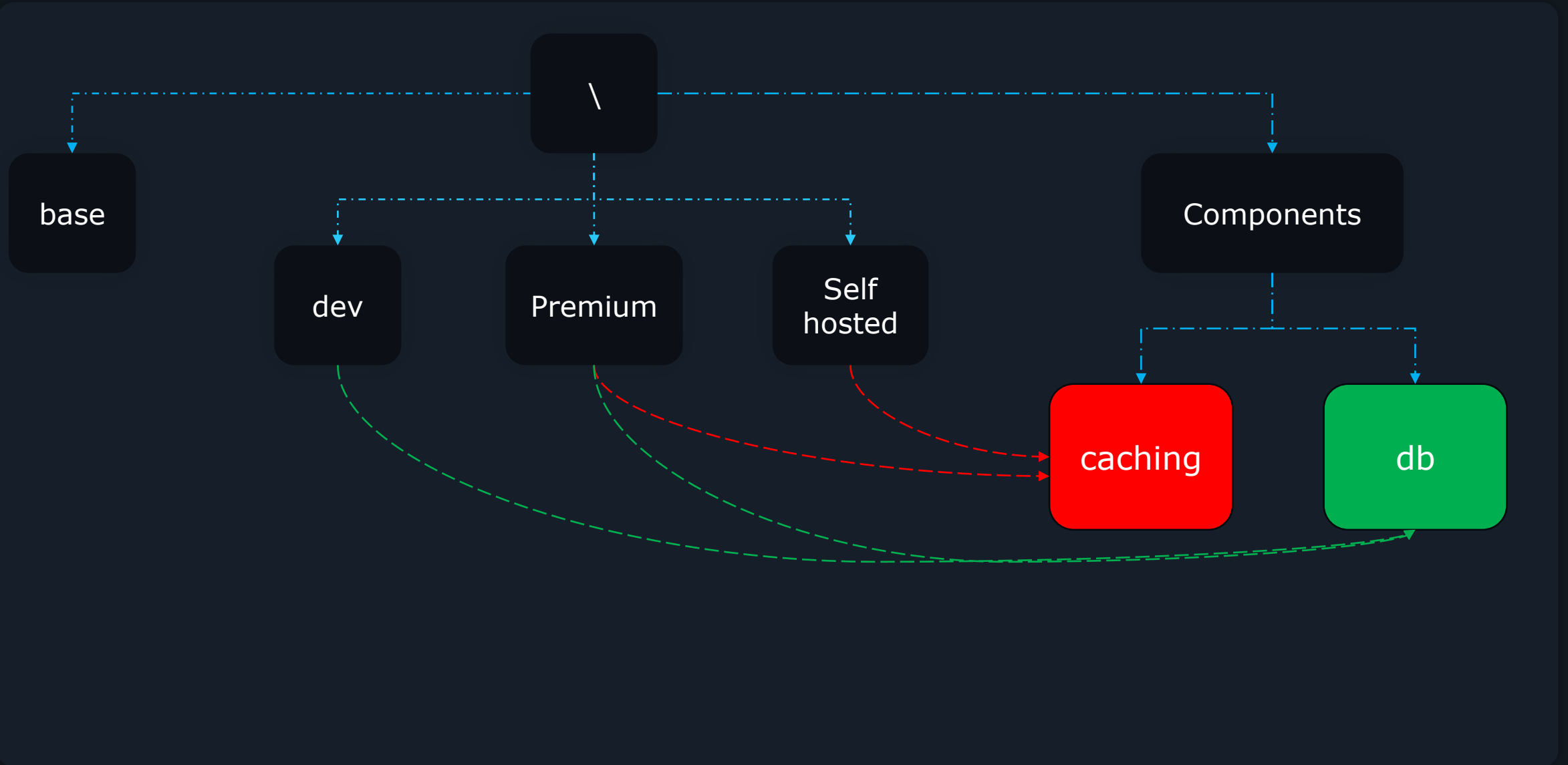
Caching

- Premium
- Self Hosted

External DB

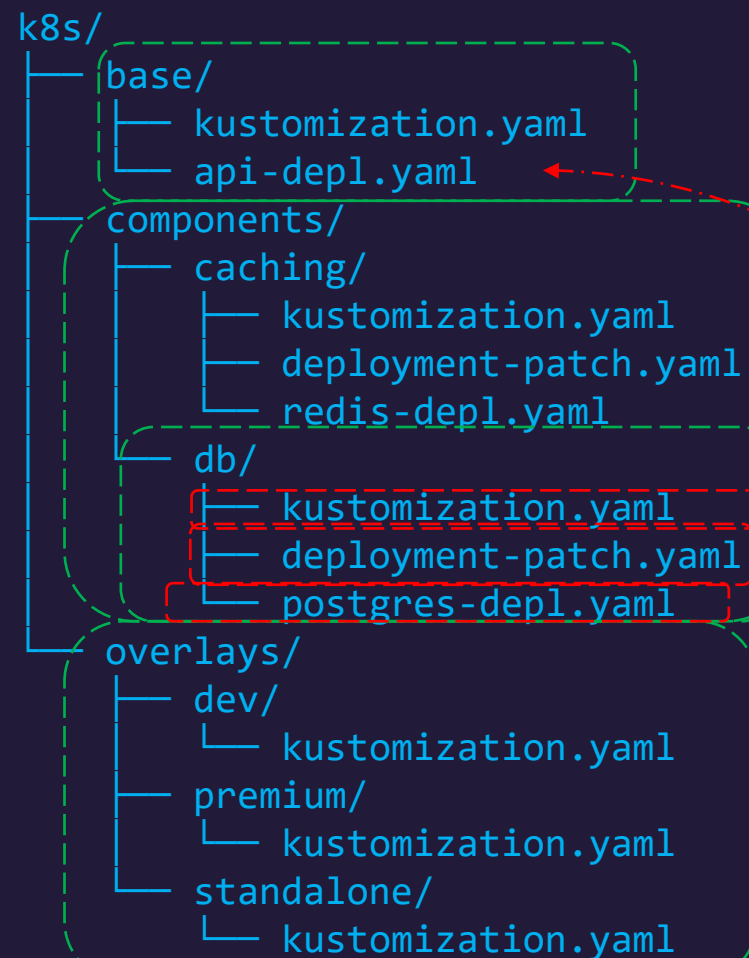
- Dev
- Premium

Components



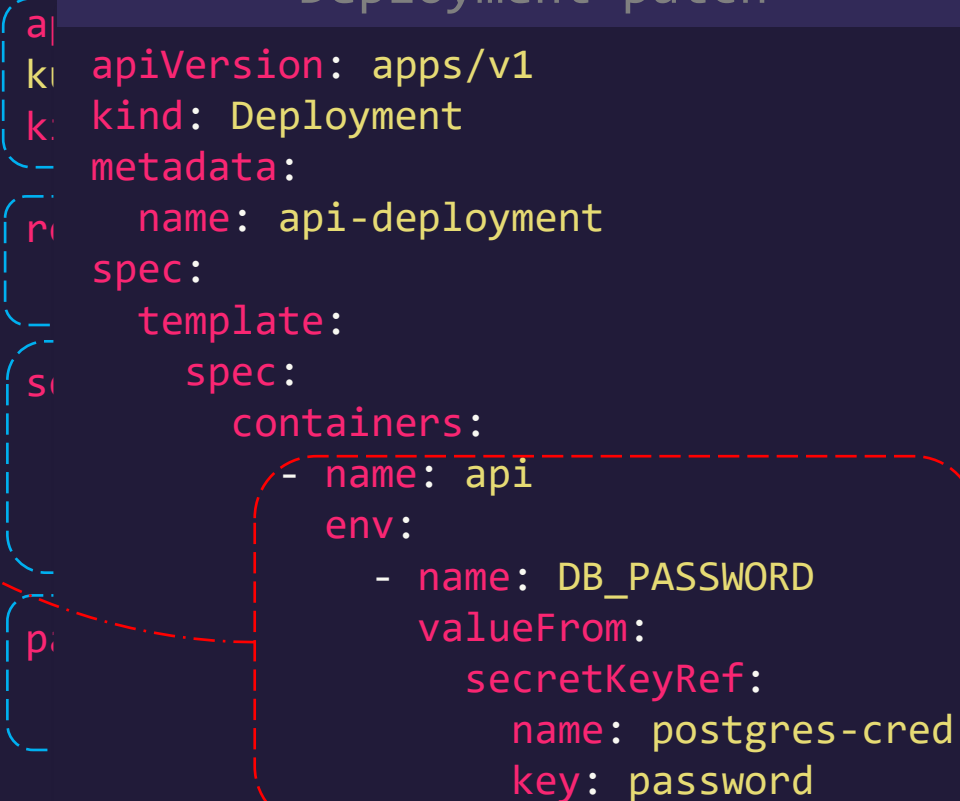
Components

```
k8s/  
├── base/  
│   ├── kustomization.yaml  
│   └── api-depl.yaml  
├── components/  
│   ├── caching/  
│   │   ├── kustomization.yaml  
│   │   ├── deployment-patch.yaml  
│   │   └── redis-depl.yaml  
│   └── db/  
│       ├── kustomization.yaml  
│       ├── deployment-patch.yaml  
│       └── postgres-depl.yaml  
└── overlays/  
    ├── dev/  
    │   └── kustomization.yaml  
    ├── premium/  
    │   └── kustomization.yaml  
    └── standalone/  
        └── kustomization.yaml
```



Components/db/kustomization.yaml Deployment-patch

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: api-deployment  
spec:  
  template:  
    spec:  
      containers:  
        - name: api  
          env:  
            - name: DB_PASSWORD  
              valueFrom:  
                secretKeyRef:  
                  name: postgres-cred  
                  key: password
```



Components

```
k8s/  
├── base/  
│   ├── kustomization.yaml  
│   └── api-depl.yaml  
├── components/  
│   ├── caching/  
│   │   ├── kustomization.yaml  
│   │   ├── deployment-patch.yaml  
│   │   └── redis-depl.yaml  
│   └── db/ ←  
│       ├── kustomization.yaml  
│       ├── deployment-patch.yaml  
│       └── postgres-depl.yaml  
└── overlays/  
    ├── dev/  
    │   └── kustomization.yaml  
    ├── premium/  
    │   └── kustomization.yaml  
    └── standalone/  
        └── kustomization.yaml
```

● Overlay/premium/kustomization

```
bases:  
- ../../base  
  
components:  
- ../../components/db
```



KodeKloud

Secret/Config Generator



Generators

Generator

```
configMap
apiVersion: v1
kind: ConfigMap
metadata:
  name: db-cred-jj26gh
data:
  password: "password1"
```

```
api-depl.yaml
env:
- name: DB_PASSWORD
  valueFrom:
    configMapKeyRef:
      name: db-cred-jj26gh
      key: password
```

Modify config/secret

Generator

```
configMap
apiVersion: v1
kind: ConfigMap
metadata:
  name: db-cred-a477b
data:
  password: "password2"
```

```
api-depl.yaml
env:
- name: DB_PASSWORD
  valueFrom:
    configMapKeyRef:
      name: db-cred-a477b
      key: password
```

configMap name changed
resulting in a new
deployment

Generators

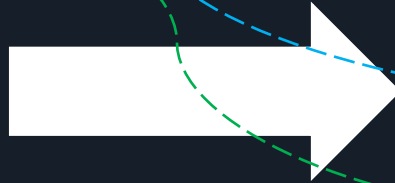
kustomization

```
configMapGenerator:
```

```
- name: db-cred
```

```
  literals:
```

```
  - password=password1
```



configMap

```
apiVersion: v1
```

```
kind: ConfigMap
```

```
metadata:
```

```
  name: db-cred-jj26gh
```

```
data:
```

```
  password: "password1"
```

Generators

kustomization

```
configMapGenerator:  
- name: nginx-config  
  files:  
  - nginx.conf
```

Nginx.conf

```
server {  
  listen 80;  
  server_name example.com;  
  location / {  
  }  
}
```

configMap

```
apiVersion: v1  
data:  
  nginx.conf: "server  
{\r\n  listen  
80;\r\n  server_name  
example.com;\r\n  location  
  / {\r\n  }\r\n}"  
kind: ConfigMap  
metadata:  
  name: nginx-config-jj26gh
```

Secret Generators

● kustomization

```
secretGenerator:  
- name: db-cred  
  literals:  
  - password=password1
```



● secret

```
apiVersion: v1  
data:  
  password: cGFzc3dvcmQx  
kind: Secret  
metadata:  
  name: db-cred-dd6525th4g  
type: Opaque
```

Secret Generators

kustomization

```
secretGenerator:  
  - name: nginx-secret  
    files:  
      - nginx-conf
```

nginx.conf

```
server {  
  listen 80;  
  server_name example.com;  
  location / {  
  }  
}
```



secret

```
apiVersion: v1  
data:  
  nginx.conf: |  
    c2VydmVyIHsNCiAgICBsaXN0ZW4gO  
    DA7DQogICAgc2VydmVyX25hbWUgZXh  
    BsZS5jb2  
    07DQogICAgbG9jYXRpb24gLyB7DQo  
    gICAgfQ0KfQ==  
kind: Secret  
metadata:  
  name: nginx-config-mfbhtbf6bd  
type: Opaque
```

Generator Behavior

● base/kustomization

```
configMapGenerator:  
- name: db-config  
  literals:  
    - password=password1  
    - username=postgres
```

● prod/kustomization

```
configMapGenerator:  
- name: db-config  
  behavior: merge  
  literals:  
    - password=prodPassword  
    - port=5432
```



● Base/kustomization

```
apiVersion: v1  
data:  
  password: prodPassword  
  port: "5432"  
  username: postgres  
kind: ConfigMap  
metadata:  
  name: db-config-k6m98g6mk7
```

Generator Behavior

● base/kustomization

```
configMapGenerator:  
- name: db-config  
  literals:  
    - password=password1  
    - username=postgres
```

● prod/kustomization

```
configMapGenerator:  
- name: db-config  
  behavior: replace  
  literals:  
    - password=prodPassword  
    - port=5432
```



● Base/kustomization

```
apiVersion: v1  
data:  
  password: prodPassword  
  port: "5432"  
kind: ConfigMap  
metadata:  
  name: db-config-mm8mbmccf7
```



KodeKloud

Imperative commands



Imperative Commands

- Kustomize provides the ability to imperatively update kustomization.yaml files using the *edit* subcommand
- Pretty much anything that can be done declaratively, can also be done imperatively as well
 - Adding labels/prefixes/namespaces
 - Modifying images/replicas

Kustomize Edit

```
>_
```

```
$ kustomize edit --help
```

Examples:

```
# Adds a configmap to the kustomization file
kustomize edit add configmap NAME --from-literal=k=v
```

```
# Sets the nameprefix field
kustomize edit set nameprefix <prefix-value>
```

```
# Sets the namesuffix field
kustomize edit set namesuffix <suffix-value>
```

Available Commands:

add	Adds an item to the kustomization file.
alpha-list-builtin-plugin	[Alpha] List the builtin plugins
fix	Fix the missing fields in kustomization file
remove	Removes items from the kustomization file.
set	Sets the value of different fields in

kustomization file.

Kustomize Edit Set

>_

```
$ kustomize edit set image nginx=nginx:1.2.2

$ kustomize edit set namespace staging

$ kustomize edit set label org:KodeKloud env:staging

$ kustomize edit set replicas nginx-deployment=5
```

```
● kustomization
images:
- name: nginx
  newTag: 1.2.2
namespace: staging
commonLabels:
  env: staging
  org: KodeKloud
replicas:
- count: 5
  name: nginx-deployment
```

Kustomize edit add

>_

```
$ kustomize edit add configmap db-creds --from-  
literal=password=password1 --from-  
literal=username=root
```

```
$ kustomize edit add secret my-secret --from-  
literal=login=user1 --from-  
literal=password=mypassword
```

```
$ kustomize edit add resource db/db-depl.yaml
```

● kustomization

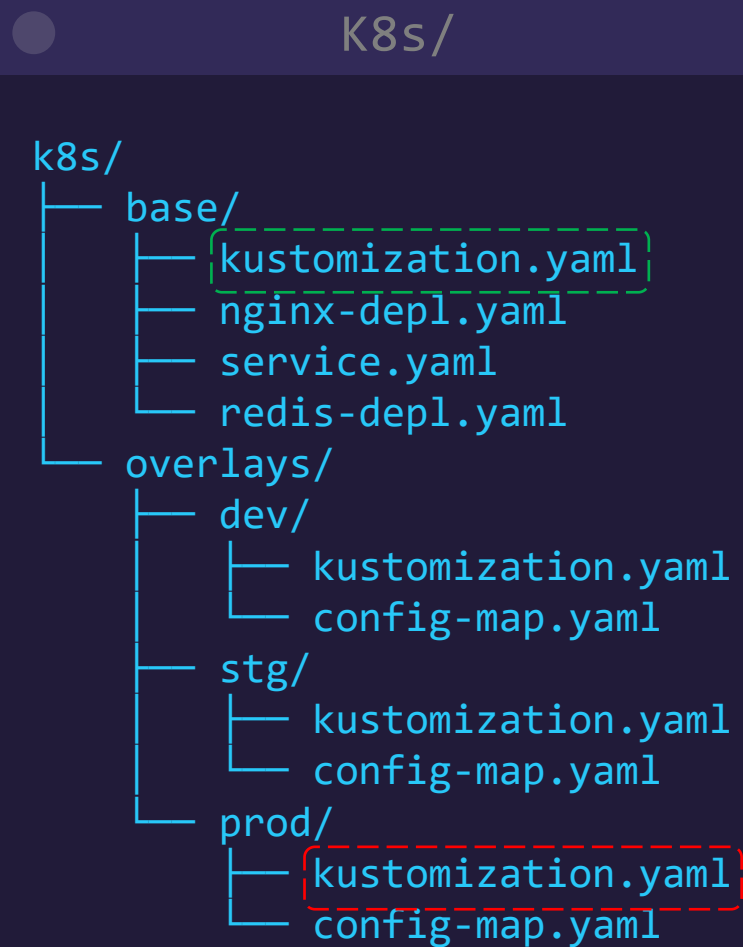
```
configMapGenerator:  
- literals:  
  - password=password1  
  - username=root  
  name: db-creds  
secretGenerator:  
- literals:  
  - login=user1  
  - password=mypassword  
  name: my-secret  
  type: Opaque  
resources:  
- db/db-depl.yaml
```

Kustomize Edit

> _

```
$ cd k8/base && kustomize edit set label org:KodeKloud
```

```
$ cd k8/overlays/prod && kustomize edit set namespace prod
```





KodeKloud

Edit CI/CD Use-Case



Edit Command CI/CD Pipeline

