Manipulating Kubernetes Object Metadata

Nigel Brown

Freelance Technical Author

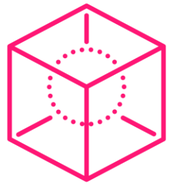
@n_brownuk | @nigelb@fosstodon.org | windsock.io

Module Outline

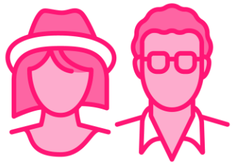
Coming up:

- Setting namespaces for objects
- Object name prefixes and suffixes
- Selective application of annotations
- Working with object labels and selectors

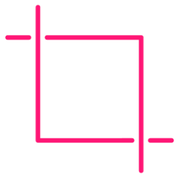
Namespaces in Kubernetes



Namespaces provide a mechanism for isolating Kubernetes objects within a single cluster



Useful for clusters where there are multiple tenants that require workloads to be isolated



To effect namespace isolation, all object definitions require their namespace definition to be set

Namespace Transformer

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: NamespaceTransformer
    metadata:
      name: namespace-transformer
      namespace: dev
```

Namespace Transformer

kustomization.yaml

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

transformers:
  - namespace-transformer.yaml
```

Namespace Transformer

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: NamespaceTransformer
    metadata:
      name: namespace-transformer
      namespace: dev
```

Namespace Transformer

kustomization.yaml

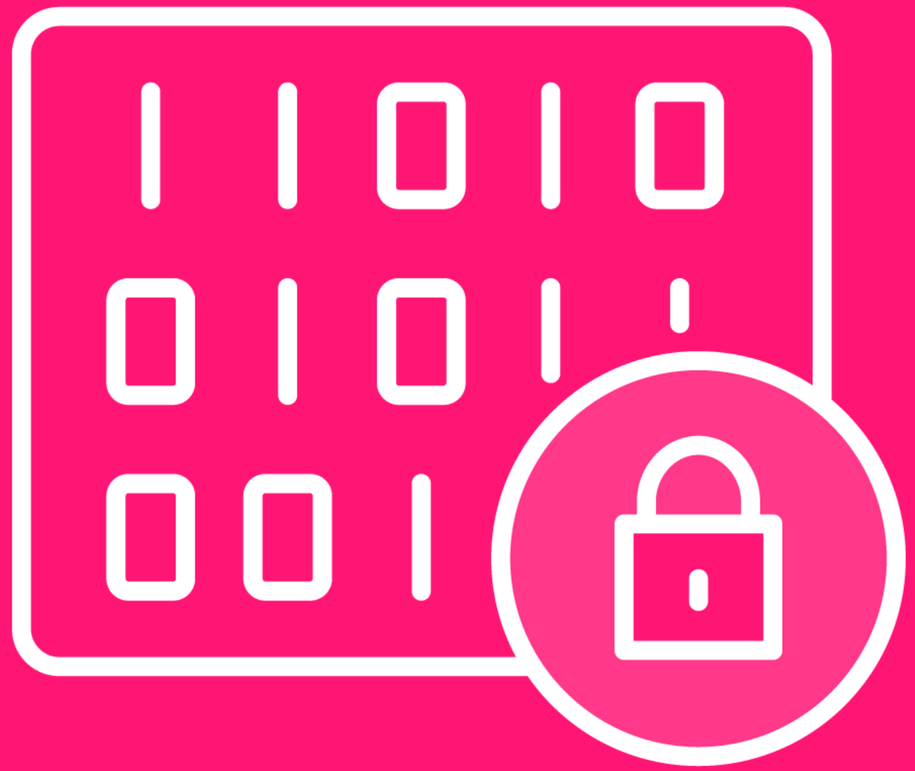
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

transformers:
  - |-
    apiVersion: builtin
    kind: NamespaceTransformer
    metadata:
      name: namespace-transformer
      namespace: dev
    unsetOnly: true
```

Default transformer behavior is to apply namespace to all objects in scope

Namespace Transformer

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: NamespaceTransformer
    metadata:
      name: namespace-transformer
      namespace: dev
    fieldSpecs:
      - kind: Namespace
        path: metadata/name
```



Role Bindings

Kubernetes ‘RoleBinding’ objects contain ‘subject’ references. The ‘subject’ may be a ‘ServiceAccount’ which is scoped to a namespace.

Namespaces for Subjects

Default Namespace

Sets the namespace for subjects to ‘default’ namespace

Update Subjects

Namespace for all subjects is set to the defined value†

Do Nothing

Kustomize ignores the namespace setting of subjects

† Applies to objects of type ServiceAccount only

Updating Subjects

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: NamespaceTransformer
    metadata:
      name: namespace-transformer
      namespace: dev
    setRoleBindingSubjects: allServiceAccounts
```

> Other field values are ‘defaultOnly’ or ‘none’

Namespace

kustomization.yaml

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

namespace: dev
```

Demo

Setting Namespaces in an Overlay

- Create a namespace definition for ‘dev’
- Add to resources field in Kustomization
- Set namespace field value to ‘dev’
- Perform a Kustomize build of overlay

Manipulating Object Names

Name Prefix

An object's name has a designated prefix added to its declared name

Name Suffix

Similarly, an object's name can be transformed with a designated suffix

References to the name in other objects will be updated accordingly.

Prefixes and Suffixes

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: PrefixSuffixTransformer
    metadata:
      name: prefix-suffix-transformer
    prefix: qa-
    suffix: -f8a2ab4
```

Prefixes and Suffixes

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: PrefixSuffixTransformer
    metadata:
      name: prefix-suffix-transformer
    prefix: qa-
    suffix: -f8a2ab4
    fieldSpecs:
      - path: metadata/name
```

```
<snip>

fieldSpecs:
  - group: apps/v1
    kind: Deployment
    path: metadata/name
```

FieldSpecs

A ‘fieldSpec’ specifies a field in a Kubernetes API object for transforming

```
<snip>

fieldSpecs:
  - group: apps/v1
    kind: Deployment
    path: metadata/name
  - group: networking.k8s.io/v1
    kind: Ingress
    path: metadata/name
```

FieldSpecs

A ‘fieldSpec’ specifies a field in a Kubernetes API object for transforming

```
<snip>

fieldSpecs:
  - group: apps/v1
    kind: Deployment
    path: metadata/name
```

FieldSpecs

A ‘fieldSpec’ specifies a field in a Kubernetes API object for transforming

```
<snip>

fieldSpecs:
  - path: metadata/name
    create: true
```

FieldSpecs

A ‘fieldSpec’ specifies a field in a Kubernetes API object for transforming

Convenience Fields

Kustomize provides the namePrefix and nameSuffix convenience fields for object name transformations.

Example Object Definition

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend

<snip>
```

Adding Prefixes and Suffixes

If an object has the name ‘backend’ in a definition contained in a resource …

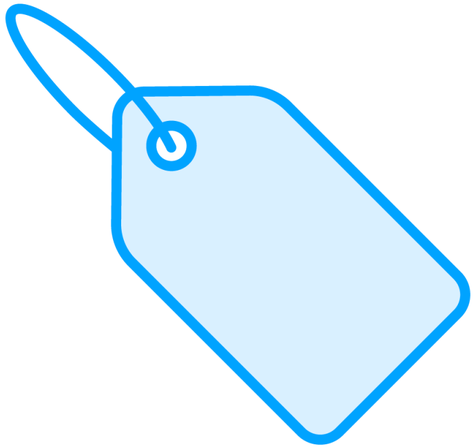
kustomization.yaml

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

resources:
  - deployment.yaml
namePrefix: qa-
nameSuffix: -f8a2ab4

<snip>
```

Build output

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: qa-backend-f8a2ab4

<snip>
```

Adding Metadata to Objects



Labels

Important in Kubernetes when attached to objects for identification purposes

Annotations

Commonly used to imbue important, non-identifying information to Kubernetes objects

Annotations

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: AnnotationsTransformer
    metadata:
      name: annotations-transformer
    annotations:
      prometheus.io/port: "9102"
```

Annotations

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: AnnotationsTransformer
    metadata:
      name: annotations-transformer
    annotations:
      prometheus.io/port: "9102"
    fieldSpecs:
      - group: apps
        version: v1
        kind: Deployment
        path: spec/template/metadata/annotations
        create: true
```

Labels

kustomization.yaml

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

transformers:
  - |-
    apiVersion: builtin
    kind: LabelTransformer
    metadata:
      name: label-transformer
    labels:
      app.kubernetes.io/name: server
    fieldSpecs:
      - path: metadata/labels
        create: true
```

Complicated?

It’s nice to have the full expression of each of these metadata transformers available when it’s needed, but could life be easier?

Common Annotations

kustomization.yaml

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

commonAnnotations:
  app.kubernetes.io/env: production
```

Pod Templates

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    app.kubernetes.io/env: production
  name: myapp

<snip>

spec:
  template:
    metadata:
      annotations:
        app.kubernetes.io/env: production

<snip>
```

Labels

kustomization.yaml

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

labels:
  - pairs:
      app.kubernetes.io/name: server
      app.kubernetes.io/part-of: app
```

Labels

kustomization.yaml

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

labels:
  - pairs:
      app.kubernetes.io/name: server
      app.kubernetes.io/part-of: app
    includeTemplates: true
```

Labels

kustomization.yaml

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

labels:
  - pairs:
      app.kubernetes.io/name: server
      app.kubernetes.io/part-of: app
    includeTemplates: true
    includeSelectors: true
```

Common Labels

kustomization.yaml

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

resources:
  - deployment.yaml
  - service.yaml

commonLabels:
  app.kubernetes.io/name: server
  app.kubernetes.io/part-of: app
```

Label Transformation

Before

```yaml
apiVersion: v1
kind: Service
metadata:
  name: server
spec:
  ports:
    - port: 3000
      targetPort: 3000
  type: NodePort
```

After

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: server
    app.kubernetes.io/part-of: app
  name: server
spec:
  ports:
    - port: 3000
      targetPort: 3000
  selector:
    app.kubernetes.io/name: server
    app.kubernetes.io/part-of: app
  type: NodePort
```

Label Transformation

Before

```yaml
apiVersion: v1
kind: Service
metadata:
  name: server
spec:
  ports:
    - port: 3000
      targetPort: 3000
  type: NodePort
```

After

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: server
    app.kubernetes.io/part-of: app
  name: server
spec:
  ports:
    - port: 3000
      targetPort: 3000
  selector:
    app.kubernetes.io/name: server
    app.kubernetes.io/part-of: app
  type: NodePort
```

Updating Selectors

Don’t attempt to update the selectors associated with workloads and services once deployed to a cluster.

Which Field to Use?

Kustomize has multiple fields for transforming labels, which is the correct one to use?

Kustomization Field Use

labels

Use when labels need adding to objects, and optionally templates

commonLabels

Best for when both labels and selectors need adding to objects

transformer

Full transformer for complex scenarios, including selective object targets

Demo

Applying Labels to Objects in an Overlay

- Define a simple ‘env’ label for objects
- Use a common label for the app version
- Perform builds for the base and overlay
- Make a comparison of the build output

Up Next:

Generating ConfigMaps and Secrets

Module Summary

What we covered:

- Namespaces can be changed in overlays
- Addition of prefixes and/or suffixes to object names
- Object selection using ‘fieldSpecs’
- Non-identifying metadata addition to objects as annotations
- Different techniques for adding labels