Kustomize Principles

Nigel Brown

Freelance Technical Author

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

Module Outline

Coming up:

- The Kustomization
- Using the Kustomize CLI
- Bases and overlays
- Generators and transformers
- Kustomize resources

Kubernetes Resource Model

The KRM is a declarative means for expressing our intent in Kubernetes clusters

Kustomize adopts the familiar KRM approach for defining operations on configuration

Kustomization

A file called ‘kustomization.yaml’, containing syntax defining how Kubernetes API objects get rendered, is the Kustomization root.

Kustomization File Content

kustomization.yaml

```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
  - github.com/nbrownuk/myapp/deploy/config?ref=v1.1
  - configmap.yaml
generators:
  - generator.yaml
components:
  - ../external_db
transformers:
  - transformer.yaml
validators:
  - validator.yaml
```

```
$ ls *.yaml
deployment.yaml  kustomization.yaml  service.yaml


$ kustomize build . | kubectl apply -f -
```

Building and Applying Configuration

The kustomize CLI can be used in conjunction with the kubectl CLI

```
$ ls *.yaml
deployment.yaml  kustomization.yaml  service.yaml


# Render Kubernetes API objects with kubectl
$ kubectl kustomize .


# Apply rendered Kubernetes API objects with kubectl (shorthand -k)
$ kubectl apply --kustomize .
```

Using the Integrated Version of Kustomize

Kustomize can be invoked using its integration with the kubectl CLI

Version Lag

The version of Kustomize built into the ‘kubectl’ CLI usually lags behind the current release version.

```
# Standalone kustomize binary
$ kustomize version
5.0.1


# Kustomize integrated with kubectl
$ kubectl version --short
Client Version: v1.26.3
Kustomize Version: v4.5.7
Server Version: v1.26.1
```

Finding the Version of Kustomize

Demo

Command Line Familiarization

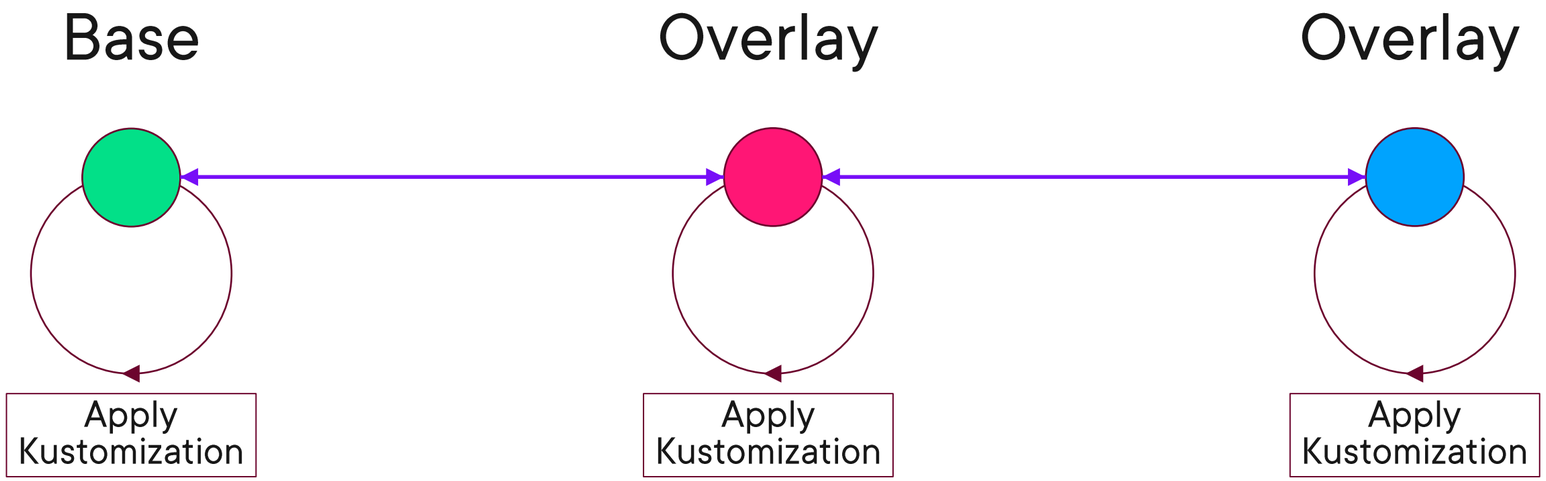
- Standalone Kustomize CLI
- Kustomize built into the Kubernetes CLI

Bases and Overlays

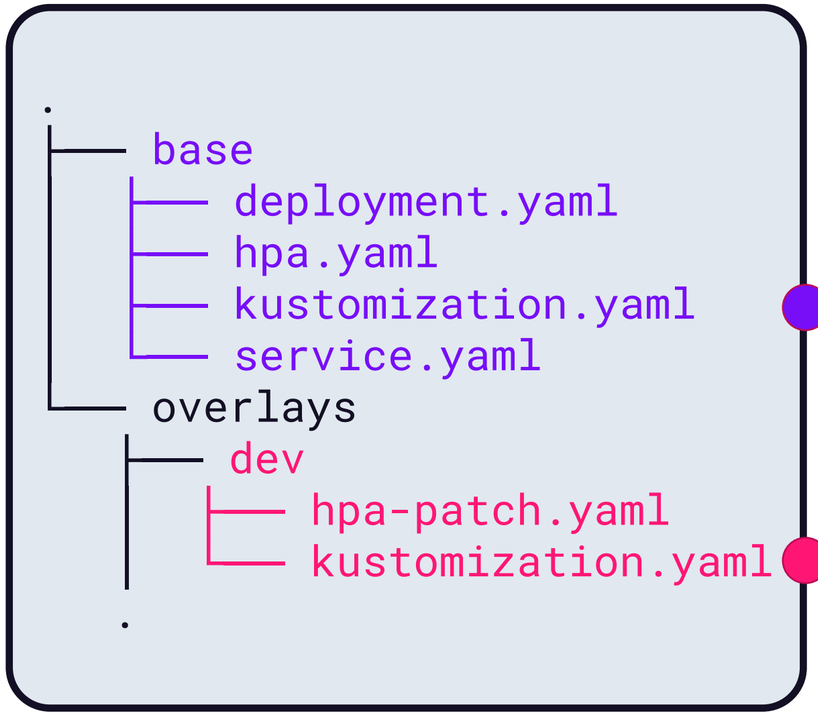
Directory Structure

```
.
├── base
│   ├── deployment.yaml
│   ├── hpa.yaml
│   ├── service.yaml
│   └── kustomization.yaml
└── overlays
    ├── dev
    │   ├── hpa-patch.yaml
    │   └── kustomization.yaml
    ├── production
    │   ├── hpa-patch.yaml
    │   ├── svc-lb-patch.yaml
    │   ├── replica-patch.yaml
    │   └── kustomization.yaml
    └── staging
        ├── hpa-patch.yaml
        ├── svc-np-patch.yaml
        └── kustomization.yaml
```

Inheritance



Creating a Variant with an Overlay




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

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

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

resources:
  - ../../base
patches:
  - hpa-patch.yaml
```

Built-in Generators

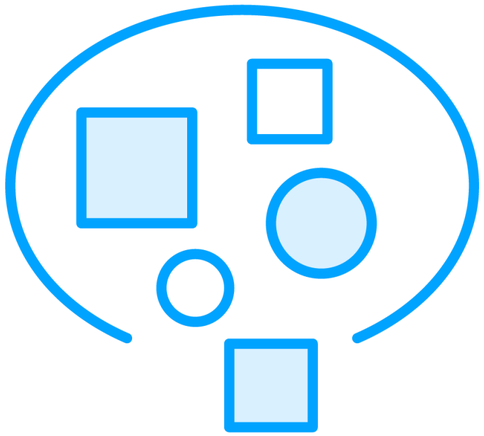
ConfigMap

Creates, replaces or merges one or more ConfigMap objects

Secret

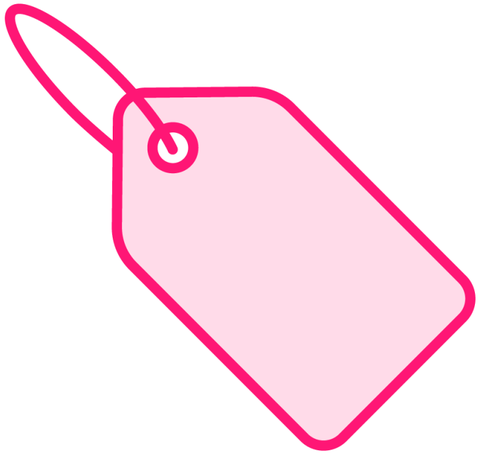
Creates, replaces or merges one or more Secret objects

Built-in Transformers



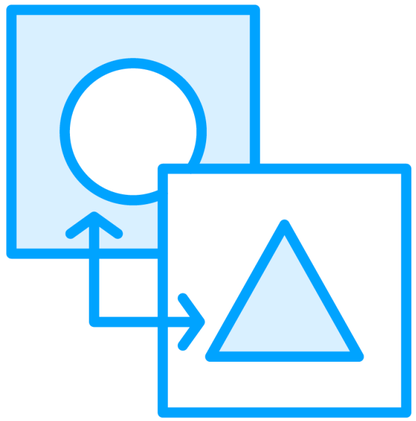
Namespace

Sets namespace for all objects



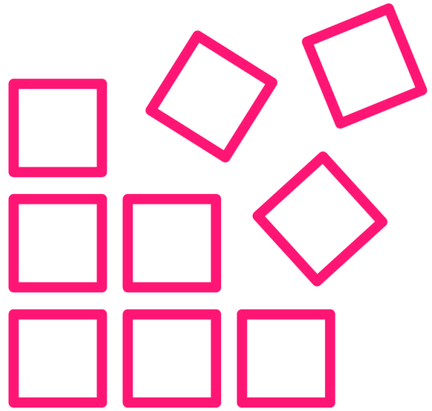
ImageTag

Sets image name, tag or digest



PrefixSuffix

Adds a prefix or suffix to a name



ReplicaCount

Modifies number of replicas

Built-in Metadata Transformers

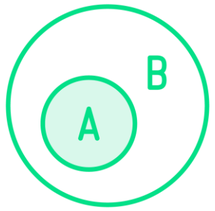
Label

Adds labels to the metadata for objects, and their corresponding selectors

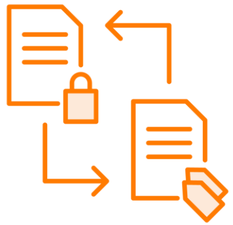
Annotation

Adds annotation key/value pairs to the metadata for all applicable objects

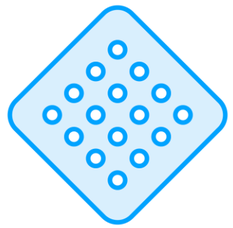
Built-in Patch Transformers



Objects that are selected using the definition of a target, have fields added or amended according to the patch details



Patch content can be defined in a YAML file, or can be provided inline within the Kustomization



Two approaches to patching are supported; ‘Strategic Merge Patch’ method and ‘JSON Patch’ method

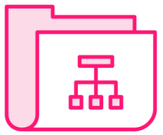
Convenience Fields

Kustomization syntax fields hide the complexity of defining generators and transformers as KRM definitions.

Resources

A field containing a list of resources that enumerate to Kubernetes definitions of API objects

A list entry may be a path to an individual YAML or JSON file with one or more object definitions



Instead of an individual file, the path may refer to another directory containing a Kustomization

A resource path in the list may also refer to remote resources specified using a URL

Local Paths

Paths must be relative to the Kustomization root under consideration

kustomization.yaml

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

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

```
.
├── base
│   ├── deployment.yaml
│   ├── kustomization.yaml
│   └── service.yaml
└── overlay
    └── prod
        ├── deployment.yaml
        ├── service.yaml
        └── kustomization.yaml
```

Local Paths

Paths must be relative to the Kustomization root under consideration

kustomization.yaml

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

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

```
.
├── base
│   ├── deployment.yaml
│   ├── kustomization.yaml
│   └── service.yaml
└── overlay
    └── prod
        ├── deployment.yaml
        ├── service.yaml
        └── kustomization.yaml
```

Local Paths

Paths must be relative to the Kustomization root under consideration

kustomization.yaml

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

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

```
.
├── base
│   ├── deployment.yaml
│   ├── kustomization.yaml
│   └── service.yaml
└── overlay
    └── prod
        ├── deployment.yaml
        ├── service.yaml
        └── kustomization.yaml
```

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

resources:
  - https://github.com/mycorp/myapp//deploy/overlays/dev/?ref=v1.1
```

Remote Paths to Git Repos

Kustomize performs a ‘git clone’ in order to access the Kustomization root content

Delineation of the repo and sub-directory is specified using ‘//’ in the path description

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

resources:
  - git@github.com:mycorp/myapp//deploy/overlays/dev/?ref=v1.1
  - ssh://git@github.com/mycorp/myapp//deploy/overlays/dev/?ref=v1.1
```

Cloning Over SSH Protocol

Kustomize supports both formats of Git's SSH URL syntax

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

resources:
  - https://raw.githubusercontent.com/nbrownuk/myapp/v1.1/deploy/deployment.yaml
```

Remote Files

Remote individual files can also be referenced as resources for a Kustomization

Demo

Using Kustomize to Create an Overlay Variant

- Create a simple Kustomization for a base
- Use an overlay to transform the original config in the base

Up Next:

Manipulating Kubernetes Object Metadata

Module Summary

What we covered:

- Kustomize operates on config defined in bases and overlays
- Desired transformations are defined in Kustomizations
- Kustomize has a set of built-in functions; generators and transformers
- Resources are used to define the config we wish to work with