



KubeCon



CloudNativeCon

Europe 2022

WELCOME TO VALENCIA





KubeCon



CloudNativeCon

Europe 2022

Kubernetes SIG CLI: Intro and Updates

- Katrina Verrey - Shopify
- Maciej Szulik - Red Hat
- Eddie Zaneski - Chainguard



Who?

Chairs:

- Sean Sullivan (@seans3), Google
- Eddie Zaneski (@eddiezane), Chainguard
- Katrina Verey (@KnVerey), Shopify

Tech Leads:

- Maciej Szulik (@soltys), Red Hat
- Katrina Verey (@KnVerey), Shopify

<https://github.com/kubernetes/community/tree/master/sig-cli#leadership>

What?

We focus on:

- development and standardization of the **CLI framework**
- improving **user experience and usability**

<https://github.com/kubernetes/community/tree/master/sig-cli#cli-special-interest-group>

What?

Subprojects:

- [kubectl](#)
- [kustomize](#)
- [krew](#) + [krew index](#)
- [kui](#)
- [cli-utils](#) + [cli-runtime](#) + [cli-experimental](#)
- **NEW!** [krm-functions](#)

<https://github.com/kubernetes/community/tree/master/sig-cli#subprojects>

Where?

Slack:

[#sig-cli](#)

Mailing list:

<https://groups.google.com/g/kubernetes-sig-cli>

<https://github.com/kubernetes/community/tree/master/sig-cli#contact>

When?

Biweekly meeting:

Wednesday at 06:00 PM CEST / 12:00 PM EDT / 09:00 AM PDT

Biweekly Kustomize/Kubectl bug scrubs:

Wednesday at 06:00 PM CEST / 12:00 PM EDT / 09:00 AM PDT

<https://github.com/kubernetes/community/tree/master/sig-cli#meetings>

Enhancements

[KEP 1441 - kubectl debug](#) - still beta

[KEP 2551 - kubectl exit code normalization](#) - proposal

[KEP 2590 - Add subresource support to kubectl](#) - alpha released in 1.24

[KEP 2953 - Kustomize Plugin Graduation](#) - alpha in progress

[KEP 2299 - Kustomize Plugin Composition API](#) - implementable

[KEP 2906 - Kustomize KRM Plugin Catalog](#) - implementable

[KEP 2985: Public KRM functions registry](#) - alpha in progress

Major changes

[Expose kustomize version in kubectl version](#) (thx Katrina Verrey)

[Beautify kubectl help commands](#) (thx Chok Yip Lau)

Major completion refactorings, more still to come (thx Marc Khouzam)

[Increase default value of cache TTL to 6 hours](#) (thx Manish Kumar)

Major changes

[kubectl create token](#) (thx Jordan Liggitt)

[kubectl logs will now warn and default to the first container in a pod](#) (thx Jian Li)

Intro to KRM Functions

Our newest subproject

KRM Functions Specification

"a standard for client-side functions that operate on Kubernetes declarative configurations"

<https://github.com/kubernetes-sigs/kustomize/blob/master/cmd/config/docs/api-conventions/functions-spec.md>

KRM Functions

- KRM = Kubernetes Resource Model
- small, **interoperable**, and language-independent executable programs
- can be **chained together** as part of a configuration management pipeline
- the best way to build Kustomize and Kpt **extensions**



KubeCon



CloudNativeCon

Europe 2022

```
apiVersion: transformers.example.co/v1
kind: ValueAnnotator
metadata:
  name: my-annotator
value: 'important-data'
```

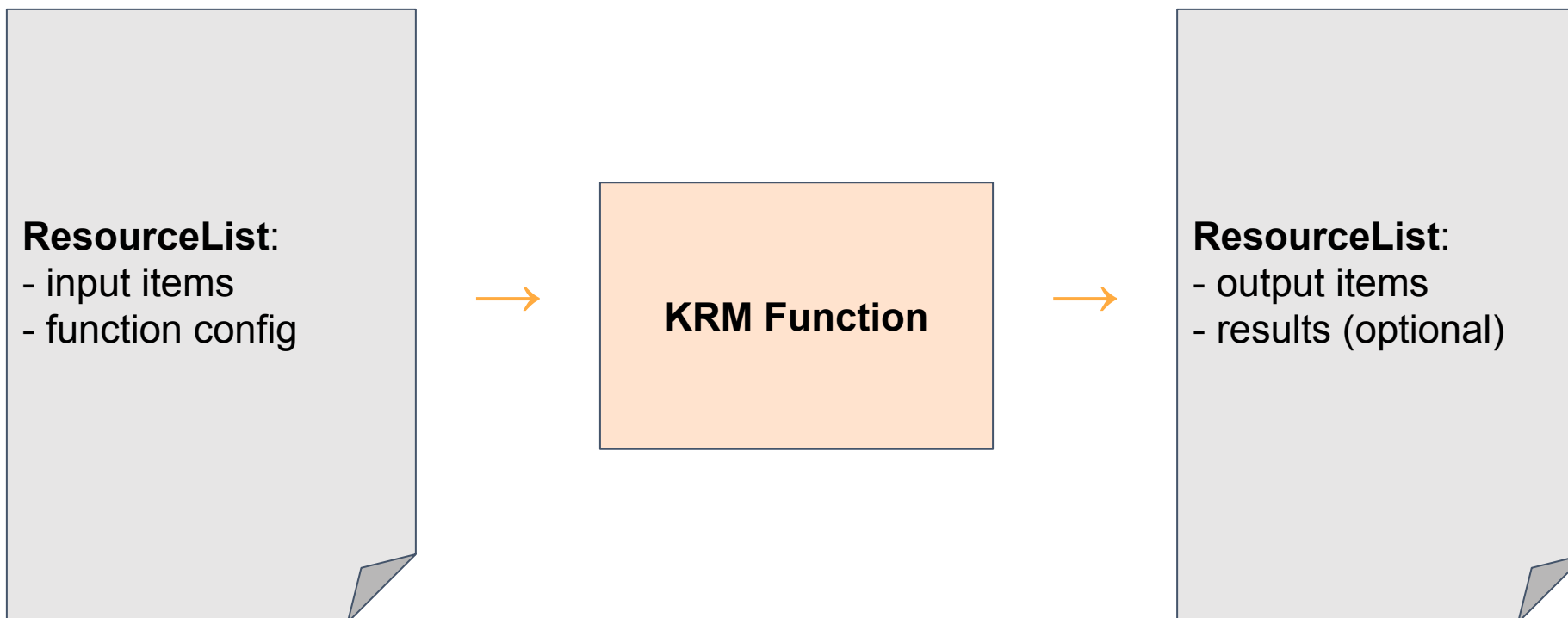


KubeCon



CloudNativeCon

Europe 2022





KubeCon



CloudNativeCon

Europe 2022

```
apiVersion: v1
kind: ResourceList
items:
- apiVersion: v1
  kind: ConfigMap
  metadata:
    name: tester
  data:
    foo: bar
functionConfig:
  apiVersion: transformers.example.co/v1
  kind: ValueAnnotator
  metadata:
    name: my-annotator
  value: 'important-data'
```




KubeCon



CloudNativeCon

Europe 2022

```
apiVersion: v1
kind: ResourceList
items:
- apiVersion: v1
  kind: ConfigMap
  metadata:
    name: tester
  data:
    foo: bar
functionConfig:
  apiVersion: transformers.example.co/v1
  kind: ValueAnnotator
  metadata:
    name: my-annotator
  value: 'important-data'
```

Transform



```
kind: ResourceList
items:
- kind: ConfigMap
  metadata:
    name: tester
    annotations:
      custom.io/the-value: important-data
  data:
    foo: bar
```



KubeCon



CloudNativeCon

Europe 2022

KRM Functions in Kustomize

```
# $MYAPP/annotator.yaml
apiVersion: transformers.example.co/v1
kind: ValueAnnotator
metadata:
  name: notImportantHere
  annotations:
    config.kubernetes.io/function: |
      container:
        image: example.docker.com/my-functions/valueannotator:1.0.0
value: 'important-data'
```

```
# $MYAPP/kustomization.yaml
resources:
- service.yaml
transformers:
- annotator.yaml
```

KRM Functions in Kustomize

[KEP 2953: Kustomize Plugin Graduation](#)

This is the overarching plan for Kustomize extensions. It is based on the promotion of "KRM Functions".

[KEP 2906: Kustomize KRM Plugin Catalog](#)

A proposal for how to manage sets of KRM Functions in Kustomize to provide seamless, secure user experiences for extensibility.

[KEP 2299: Kustomize Plugin Composition API](#)

A proposal for a new way to combine extensions in Kustomize to make them easier to use, particularly when they represent abstractions.



KubeCon



CloudNativeCon

Europe 2022

KRM Functions in Kpt

```
# wordpress/mysql/Kptfile
apiVersion: kpt.dev/v1
kind: Kptfile
metadata:
  name: mysql
pipeline:
  mutators:
    - image: set-labels:v0.1
      configPath: labels.yaml
```

```
# wordpress/mysql/labels.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: labels
data:
  tier: mysql
```

KRM Functions Registry

[KEP 2985: Public KRM functions registry](#)

A proposal to maintain a SIG-CLI-sponsored registry of KRM Functions that can be used in any compliant tool (e.g. Kustomize, Kpt).

<https://github.com/kubernetes-sigs/krm-functions-registry>

Biweekly meeting:

Wednesday at 07:30 PM CEST / 13:30 PM EDT / 10:30 AM PDT

<https://github.com/kubernetes/community/tree/master/sig-cli#meetings>

Declarative vs. Imperative workflows

Reminder about best practices

Imperative commands

- `kubectl create`
- `kubectl run`
- `kubectl expose`
- `kubectl set`
- `kubectl autoscale`
- `kubectl patch`
- `kubectl replace`
- `kubectl scale`
- `kubectl rollout undo`
- ...

Declarative commands

- `kubectl apply`

Declarative commands

- `kubectl apply`

Please use apply



KubeCon



CloudNativeCon

Europe 2022

Mixing `kubectl apply` with the imperative object configuration commands `create` and `replace` is not supported. This is because `create` and `replace` do not retain the `kubectl.kubernetes.io/last-applied-configuration` that `kubectl apply` uses to compute updates.

<https://kubernetes.io/docs/tasks/manage-kubernetes-objects/declarative-config/>

Management technique	Operates on	Recommended environment	Supported writers	Learning curve
Imperative commands	Live objects	Development projects	1+	Lowest
Imperative object configuration	Individual files	Production projects	1	Moderate
Declarative object configuration	Directories of files	Production projects	1+	Highest

Management techniques

Warning: A Kubernetes object should be managed using only one technique. Mixing and matching techniques for the same object results in undefined behavior.

Top of mind

- Refactoring kubectl commands
- How do we better handle flags
- Project reliability ([Reliability Bar KEP](#), [Open letter](#))
- [Kustomize documentation revamp](#)
- New contributors!



KubeCon



CloudNativeCon

Europe 2022

Thank you!

Open discussion / Q&A

