



openssf best practices in progress 97%

kapp

- Website: <https://carvel.dev/kapp>
- Slack: #carvel in Kubernetes slack
- Docs with topics about diff, apply, gitops, config, *blog posts and talks* etc.
- Install: grab prebuilt binaries from the Releases page or Homebrew Carvel tap.
- Backlog: See what we're up to.

kapp (pronounced: **kap**) CLI encourages Kubernetes users to manage resources in bulk by working with “Kubernetes applications” (sets of resources with the same label). It focuses on resource diffing, labeling, deployment and deletion. Unlike tools like Helm, **kapp** considers YAML templating and management of packages outside of its scope, though it works great with tools that generate Kubernetes configuration.

```

$ kapp deploy -a redis -f examples/gitops/redis-with-configmap/ -c
--- create configmap/redis-config-ver-2 (v1) namespace: default
...
2, 2    redis-config: l
3      -    maxmemory 3mb
3      +    maxmemory 4mb
4, 4    maxmemory-policy allkeys-lru
5, 5    kind: ConfigMap
--- update pod/redis (v1) namespace: default
...
37, 37   path: redis.conf
38      -    name: redis-config-ver-1
38      +    name: redis-config-ver-2
39, 39   name: config
40, 40   status:

Changes

Namespace Name          Kind      Conds.  Age  Op      Wait to
default   redis             Pod       4/4 ok  1m  update  reconcile
~         redis-config-ver-2 ConfigMap  -      -    create  reconcile

Op:      1 create, 0 delete, 1 update, 0 noop
Wait to: 2 reconcile, 0 delete, 0 noop

Continue? [yN]: y

9:42:54AM: ---- applying 2 changes [0/2 done] ----
9:42:54AM: create configmap/redis-config-ver-2 (v1) namespace: default
9:42:54AM: update pod/redis (v1) namespace: default
9:43:06AM: ---- waiting on 2 changes [0/2 done] ----
9:43:06AM: waiting on reconcile configmap/redis-config-ver-2 (v1) namespace: default
9:43:06AM: waiting on reconcile pod/redis (v1) namespace: default
9:43:07AM: ---- applying complete [2/2 done] ----
9:43:07AM: ---- waiting complete [2/2 done] ----

Succeeded

```

See <https://carvel.dev/kapp> for detailed example workflow.

Features:

- Works with standard Kubernetes YAMLs
- Focuses exclusively on deployment workflow, not packaging or templating
 - but plays well with tools (such as ytt) that produce Kubernetes configuration
- Converges application resources (creates, updates and/or deletes resources) in each deploy

-
- based on comparison between provided files and live objects in the cluster
 - Separates calculation of changes (diff stage) from application of changes (apply stage)
 - Waits for resources to be “ready”
 - Creates CRDs and Namespaces first and supports custom change ordering
 - Works without admin privileges and does not use custom CRDs
 - making it possible to use kapp as a regular user in a single namespace
 - Records application deployment history
 - Opt-in resource version management
 - for example, to trigger Deployment rollout when ConfigMap changes
 - Optionally streams Pod logs during deploy
 - Works with any group of labeled resources (`kapp -a label:tier=web inspect -t`)
 - Works without server side components
 - GitOps friendly (`kapp app-group deploy -g all-apps --directory .`)

Join the Community and Make Carvel Better

Carvel is better because of our contributors and maintainers. It is because of you that we can bring great software to the community. Please join us during our online community meetings. Details can be found on our Carvel website.

You can chat with us on Kubernetes Slack in the #carvel channel and follow us on Twitter at @carvel_dev.

Check out which organizations are using and contributing to Carvel: Adopter's list

Development

Consult docs/dev.md for build instructions, code structure details.