

# Helm Deep Dive

Did you just tell me to go Chart myself ?

<https://sched.co/Zx4K>



# Scott Rigby

Hello! 🙌

- Helm org maintainer
- Also maintains charts and related tooling (chart-testing, chart-releaser, helm actions, and the helm hub [hub.helm.sh](https://hub.helm.sh))
- Find me:
  - Twitter: @r6by
  - GitHub: @scottrigby
  - Kubernetes Slack: @scottrigby



# Paul Czarkowski

Hello! 🖐️

- Helm Charts maintainer
- DevRel @ VMware
- Host of Tanzu.TV
- Find me at
  - Twitter: @pczarkowski
  - Github: @paulczar
  - K8s Slack: @paulczar

# Agenda

- ~~Who Are We?~~
- Helm is Helm 3
- Announcements
- Things You Wish You Knew About Helm
- Tools That Extend Helm
- Security
- Get Involved

# Helm Is Helm 3

In case you were living under a rock



# If you're on Helm 2, time to upgrade

- Helm 3 released Nov 2019
- 🙌 easy to upgrade from Helm 2
- 🙌 for the curious, see the [Overview of Helm 3 Changes](#)
- In summary, time to upgrade
- The focus of this presentation is Helm 3

# Some Important Notes Helm 2 vs 3



- Helm 2 charts still work in Helm 3
- Difference is [mostly] architectural
  - (e.g. no Tiller)
- How do I migrate my cluster from Helm 2 to Helm 3?
  - Helm 2to3 plugin
  - Strangler vs In-place

## 2to3 plugin example

```
$ helm3 plugin list
NAME      VERSION DESCRIPTION
2to3     0.1.3   migrate and cleanup Helm v2 configuration and releases in-place to Helm v3
```

```
$ helm3 2to3 convert postgres
Release "postgres" will be converted from Helm 2 to Helm 3.
[Helm 3] Release "postgres" will be created.
[Helm 3] ReleaseVersion "postgres.v1" will be created.
[Helm 3] ReleaseVersion "postgres.v1" created.
[Helm 3] Release "postgres" created.
Release "postgres" was converted successfully from Helm 2 to Helm 3. Note: the v2 releases still
remain and should be removed to avoid conflicts with the migrated v3 releases.
```



# Helm 2to3 Plugin Demo

- ...

# Announcements



**CNCF Graduation!**

# Who Got Us Here

- Helm core maintainers
- Helm charts maintainers
  - Tooling (CI/CD, Testing, Releasing)
- Helm project maintainers
  - Chartmuseum
  - Hub / Monocular
  - Plugins
- Helm org maintainers
- And the rest of the community



Thanks everyone!

# C19: Helm 2 Deprecation Timeline Extended

- Helm 2 core [timeline](#)
- *stable* and *incubator* Charts repos [timeline](#)
- TL;dr, they're the same dates:
  - Bug fixes extended to Aug 13, 2020
  - Deprecation Nov 13, 2020

# Where are community charts moving to?

- Self hosted Chart Repositories, indexed by
  - Helm Hub <http://hub.helm.sh> - currently integrated with *helm search*
  - Artifact Hub <https://artifacthub.io> - the new hotness for CNCF artifacts
- Many charts are being adopted by the maintainers of the software they install (<https://github.com/helm/charts/issues/21103>)
- If you care about a chart YOU can adopt it.
- Close friend the the charts community **Bitnami** has an excellent collection of curated (and supported!) Helm Charts.
- The Charts community can focus on tooling and best practices, such as ...

# Chart Testing | Chart Releaser

## Chart Testing

<https://github.com/helm/chart-testing>

*Automated testing for Helm Chart git Repositories.*

- Automatically detects charts changed against the target branch
- Lints and tests pull requests

## Chart Releaser

<https://github.com/helm/chart-releaser/>

*Helps Turn GitHub Repositories into Helm Chart Repositories*

- Creates Chart Package
- Uploads to github release
- Updates index in gh-pages



# Helm Chart GitHub Actions

- Chart Testing Action
  - a. On Pull Request will ...
    - Lint chart
    - Start KIND cluster
    - Install chart
    - Run tests
- Chart Releaser Action
  - a. On Merge to Main will ...
    - Create a github release
    - Update gh-pages branch with a new index.yaml

# GitHub Actions demo

- <https://github.com/helm/charts-repo-actions-demo/>

# More announcements

- Stay up to date at: <https://helm.sh/blog/>

Things You Wish  
You Knew About  
Helm

# Chart updates: Chart.yaml

```
apiVersion: v2
name: foo
description: A Helm chart for Kubernetes

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that define release artifacts to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer. They're included as
# a dependency of application charts to inject those utilities and functions into the rendering
# pipeline. Library charts cannot be deployed.
type: application

# This is the chart version. This version number should be incremented each time you make changes
# to the chart and its templates, including the app version.
version: 0.1.0

# This is the version number of the application being deployed. This version number should be
# incremented each time you make changes to the application.
appVersion: 1.16.0

dependencies:
- name: mysql
  version: "1.3.2"
  repository: "https://kubernetes-charts.storage.googleapis.com"
```

# Chart Values validation

- Place *values.schema.json* in chart folder
- Validation on commands:
  - *helm install*
  - *helm upgrade*
  - *helm template*
  - *helm lint*

```
"$schema": "http://json-schema.org/draft-07/schema#",
"properties": {
  "image": {
    "description": "Container Image",
    "properties": {
      "repo": {
        "type": "string"
      },
      "tag": {
        "type": "string"
      }
    },
    "type": "object"
  },
  "name": {
    "description": "Service name",
    "type": "string"
  },
  "port": {
    "description": "Port",
    "minimum": 0,
    "type": "integer"
  },
  "protocol": {
    "type": "string"
  }
},
"required": [
  "protocol",
  "port"
],
"title": "Values",
"type": "object"
```

# Test framework

- Chart test layout (/ci/test-values.yaml)
- Chart-testing automatically runs your tests ... if you have them.
- Resources with helm test hook annotations are automatically installed on test and cleaned up for you!

# OCI registry support

- **Experimental until clear messaging from OCI**
- Helm 3 enables charts to be stored/shared in OCI-based registries
- Experimental feature: *export HELM\_EXPERIMENTAL\_OCI=1*
- Check out *helm help chart* and *helm help registry*

<https://github.com/jdolitsky/community/blob/oci-support-updates/proposals/oci-support-updates.md>



# Release storage

- Stored as *Secrets* by default
- Stored in the namespace of the release
- Helm “Release” object has changed
- Release name prefixed with *sh.helm.release.v1*

# CRDs

- CRDs are installed only – no modifications or delete
- CRDs in the *crds/* directory are automatically installed
- Installation can be skipped with *--skip-crds*
- *crd-install* hooks are ignored with a warning in Helm 3

# Go SDK

- Things have been restructured for a better experience
- The CLI consumes the same SDK
- Packages to care about
  - *pkg/action*
  - *pkg/{chart,chartutil}*
  - *pkg/cli* and its subpackages

# Go SDK: Example

```
package main

import (
    "log"
    "os"

    "helm.sh/helm/v3/pkg/action"
    "helm.sh/helm/v3/pkg/cli"
)

func main() {
    settings := cli.New()

    actionConfig := new(action.Configuration)
    if err := actionConfig.Init(settings.RESTClientGetter(), settings.Namespace(),
os.Getenv("HELM_DRIVER"), log.Printf); err != nil {
        log.Printf("%+v", err)
        os.Exit(1)
    }

    client := action.NewList(actionConfig)
    // Only list deployed
    client.Deployed = true
    results, err := client.Run()
    if err != nil {
        log.Printf("%+v", err)
        os.Exit(1)
    }

    for _, rel := range results {
        log.Printf("%+v", rel)
    }
}
```

# Post Renderer (*--post-renderer*)

- Allows you to pipe the rendered templates through a script
  - Add labels / annotations to particular resources using kustomize or ytt
  - Add secrets/keys from Hashicorp Vault
  - Security audit resources before they ever hit the cluster
- Docs:
  - <https://helm.sh/docs/topics/advanced/#post-rendering>
- Demo repo:
  - <https://github.com/thomastaylor312/advanced-helm-demos/tree/master/post-render>
- Another example:
  - <https://github.com/k14s/ytt/tree/develop/examples/helm-ytt-post-renderer>

# Post Renderer (*--post-renderer*) - DEMO

# Tools That Extend Helm

# Tools That Extend Helm

A [in]complete list -

- Helm Diff Plugin <https://github.com/databus23/helm-diff>
- Helmfile <https://github.com/roboll/helmfile/>
- Weaveworks Helm Controller <https://github.com/fluxcd/helm-controller>
  - There are also other Helm controllers. Or you can write your own!
- Conftest <https://github.com/instrumenta/helm-conftest>
  - Example: <https://github.com/paulczar/conftest-helm-example>
- <insert your tool here>



Security

# Helm 3 security

## Kubernetes

- RBAC
- PodSecurityPolicy
- NetworkPolicy
- APIServer cert management
- User management
- Service account management
- Auditing
- Container signing

## Helm

- Chart provenance
- ~~Tiller cert management~~
- ~~Tiller RBAC management~~

In other words...

Helm inherits your RBAC\*

\* *It's [just] a client*

Get Involved

# Ways to help

- Help migrate / adopt community charts
  - Tracking issue: <https://github.com/helm/charts/issues/21103>
- Host your own repos
  - Chart Testing | Chart Releaser
  - ^ GitHub Actions
- Helm [core] maintainers
- Write a Helm blog post!
- Update Docs

★😊 *Whatever you're passionate about* 😊★

# Summary

# Summary

- Helm is Awesome. Use it.

# Helpful links

- This slide deck: <https://sched.co/Zx4K>
- Helm 3 docs: <https://v3.helm.sh/docs/>
- Helm 2 to 3 migration: [https://v3.helm.sh/docs/topics/v2\\_v3\\_migration/](https://v3.helm.sh/docs/topics/v2_v3_migration/)
- Helm 3 changes: <https://v3.helm.sh/docs/faq/#changes-since-helm-2>
- Helm Hub: <https://hub.helm.sh/>
- Distributed Helm repo tools:
  - <https://github.com/helm/chart-testing>
  - <https://github.com/helm/chart-releaser>
  - <https://github.com/helm/charts-repo-actions-demo>



Thank you!