

Intro to Helm

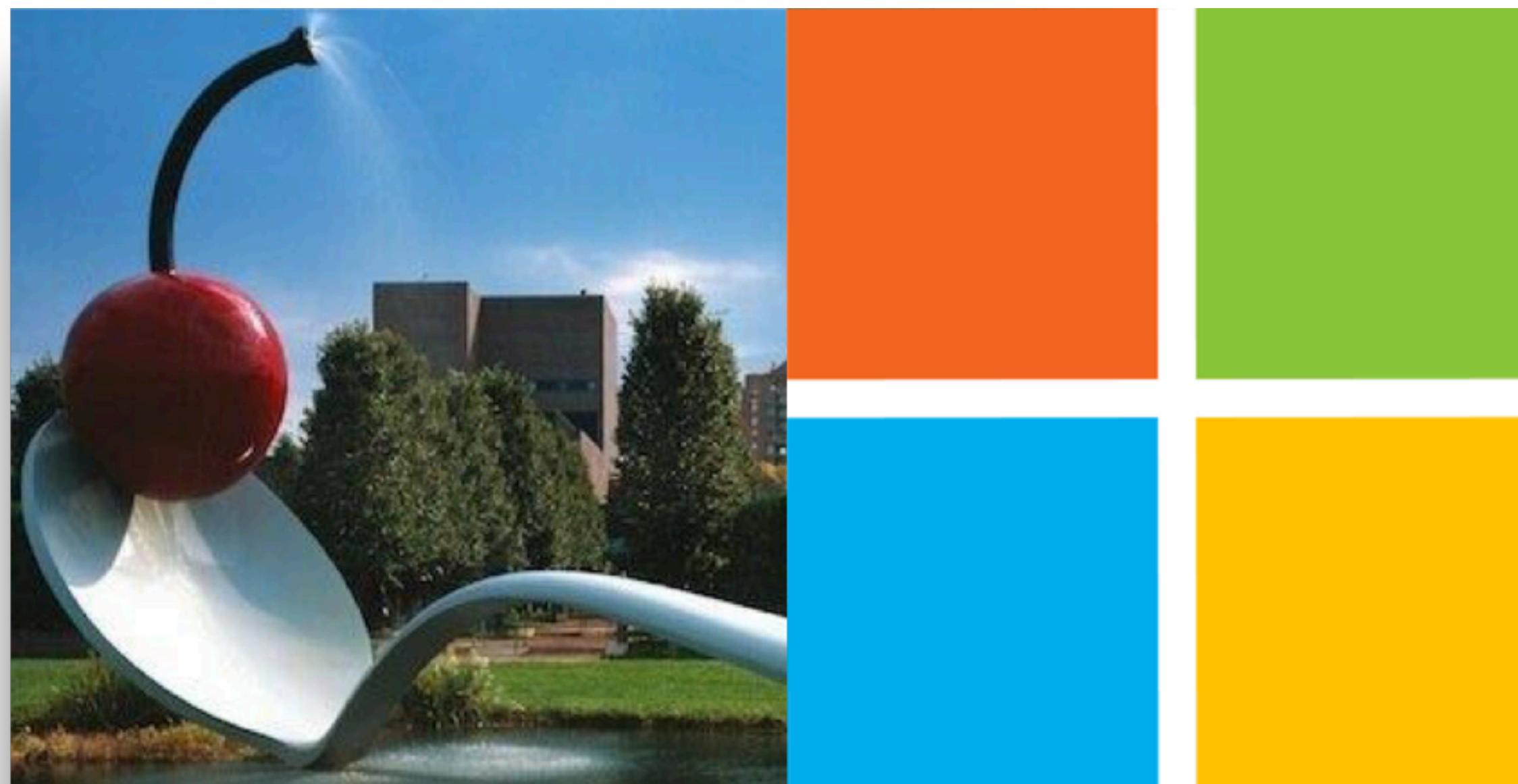


@bridgetkromhout

#kubecon

@marckhouzam

Bridget Kromhout



@bridgetkromhout

Marc Khouzam

```
Users > marckhouzam > git > helm > cmd > helm > completion.go
73
74 zsh := &cobra.Command{
75   Uses:           "zsh",
76   Short:          "generate autocompletions script for zsh",
77   Long:           "zshCompletion",
78   Args:           require.NoArgs,
79   DisableFlagsInUseLines: true,
80   RunE: func(cmd *cobra.Command, args []string) error {
81     return runCompletionZsh(out, cmd)
82   },
83 }
84
85 cmd.AddCommand(bash, zsh)
86
87 return cmd
88 }

89 func runCompletionBash(out io.Writer, cmd *cobra.Command) error {
90   err := cmd.Root().GenBashCompletionV2(out, false)
91
92   // In case the user renamed the helm binary (e.g., to be able to run
93   // both helm2 and helm3), we hook the new binary name to the completion function
94   if binary := filepath.Base(os.Args[0]); binary != "helm" {
95     renamedBinaryHook := "
96       # Hook the command used to generate the completion script
97       # to the helm completion function to handle the case where
98       # the user renamed the helm binary
99       if [[ $1type -t compopt ]]=> "builtin" ]; then
100         complete -o default -F __start_helm $1
101       else
102     fi
103   }
104 }
```



#kubecon

@marckhouzam



1. Overview: fundamental concepts
2. Getting started with charts
3. Migrating from v2 to v3
4. Community





The package manager for Kubernetes

Helm is the best way to find, share, and use software built for Kubernetes.



CONGRATS

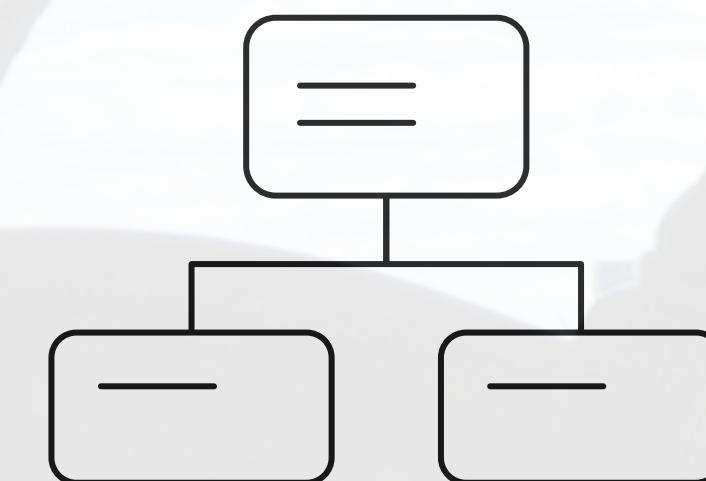
CLASS OF 2020

Love, CNCF





Find, share, and use software built for k8s



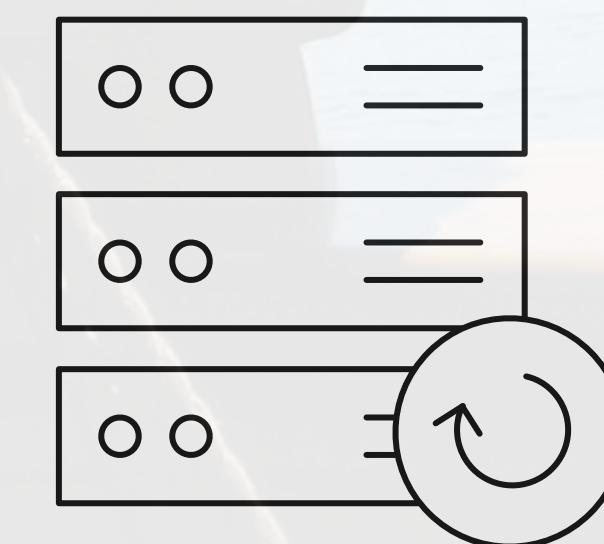
Manage complexity



Simple sharing



Easy updates



Rollbacks



demo: getting started with Helm



@bridgetkromhout

#kubecon

@marckhouzam



Finding charts
decentralized search & repo addition

hub.helm.sh
artifacthub.io

and more...



demo: charts

@bridgetkromhout

#kubecon

@marckhouzam



Customizing and making charts:

[helm.sh/docs/chart template guide/getting started/](https://helm.sh/docs/chart_template_guide/getting_started/)



Why Helm v3?

- v3 based on community best practices
- dramatic simplification of architecture
- security and production use as priorities



Helm v3 replacing v2

- bugfixes for v2 have ended
- security patches end in November 2020



what's different in Helm v3?

- using Kubernetes API server instead of Helm's own custom server (Tiller)
- making Helm more k8s-native
- inheriting security controls from kubeconfig
- using k8s RBAC to limit access & resources



--generate-name

- old style: auto-generated unless overridden
- new style: error unless auto-generation requested



library chart support

- shared by other charts
- does not create any release artifacts of its own
- a library chart's templates can only declare “define” elements
- allows simpler code reuse



farewell, Tiller

- simpler & more flexible architecture, security, & upgrades
- now using Kubernetes API directly
- rendering Charts client-side; storing in release
- lowering the barrier of entry for contributors



upgrade vs convert

- we now default to the same namespace for a release and objects (no longer using the Tiller namespace)
- can't upgrade a Helm v2 release with Helm v3 (but you can convert it with `helm-2to3`, then upgrade)



demo: Helm 2to3



what did we just see?

- Helm v3 cannot see Helm v2 resources
- if you just update from v2 to v3 and do nothing else, it will seem like everything is missing!
- essential to migrate and convert as applicable



tl;dr: Make sure you're using Helm 3 today.
We're here to help!

helm.sh/docs/faq/

cncf.io/webinars/a-conversation-about-helm-3/

cncf.io/webinars/cncf-project-webinar-charting-your-voyage-to-helm-3/



Community!

- helm.sh/docs (contributions welcome!)
- github.com/helm/helm (issues, PRs, learning!)
- github.com/helm/community (Twitter, Slack, mailing lists, weekly call)



Happy Helming!

@bridgetkromhout

#kubecon

@marckhouzam