

making kubectl plugins mainstream

ahmet alp balkan (@ahmetb)
google cloud

about me

@ahmetb on {twitter, github}

- ported docker.exe, docker maintainer ~2015
- started Azure Container Registry ~2016
- developer advocate @ Google ~2017
 - working on Google Cloud Run, Knative, dev tools
- *kubectx & kubens & kubectl-aliases*
- Google's *microservices-demo*
- kubectl plugin manager [krew.dev](#)

agenda

- user {
 - 1. kubectl plugins 101
 - 2. **krew** (plugin manager) overview
 - 3. why did we need plugins?
- developer {
 - 4. let's develop a plugin
 - 5. plugin runtime
 - 6. why develop a plugin?
 - 7. how to distribute your plugins
 - 8. best practices for writing plugins

USING PLUGINS



kubectl plugins

circa v1.12

*an extension mechanism
that lets you write
your own kubectl subcommands.*



kubectl plugin mechanism

in a nutshell

1. write an executable program
2. name it kubectl-something
3. put it in your \$PATH
4. invoke as "kubectl something"

where do I find plugins?

- github?
- some list?
- package manager?



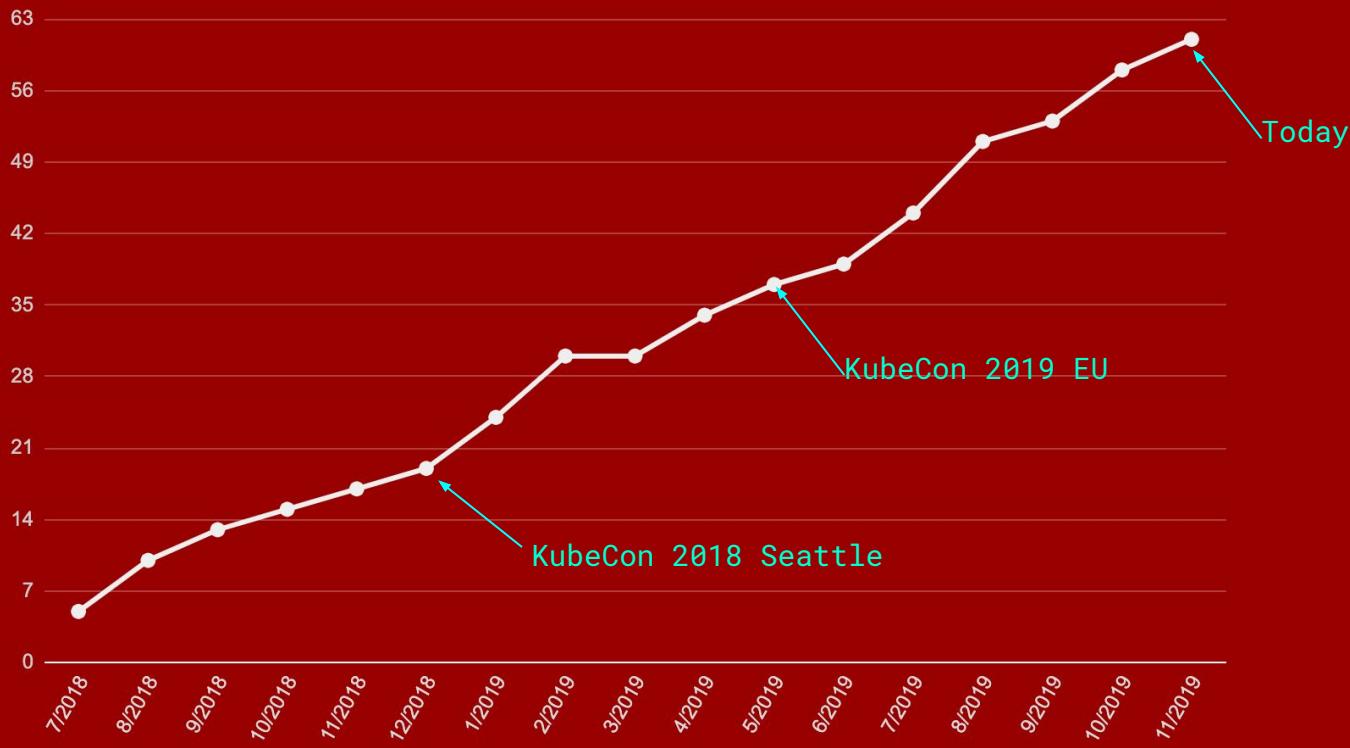
the missing plugin manager for kubectl
Kubernetes SIG CLI sub-project

<https://krew.dev>
(sig.k8s.io/krew)

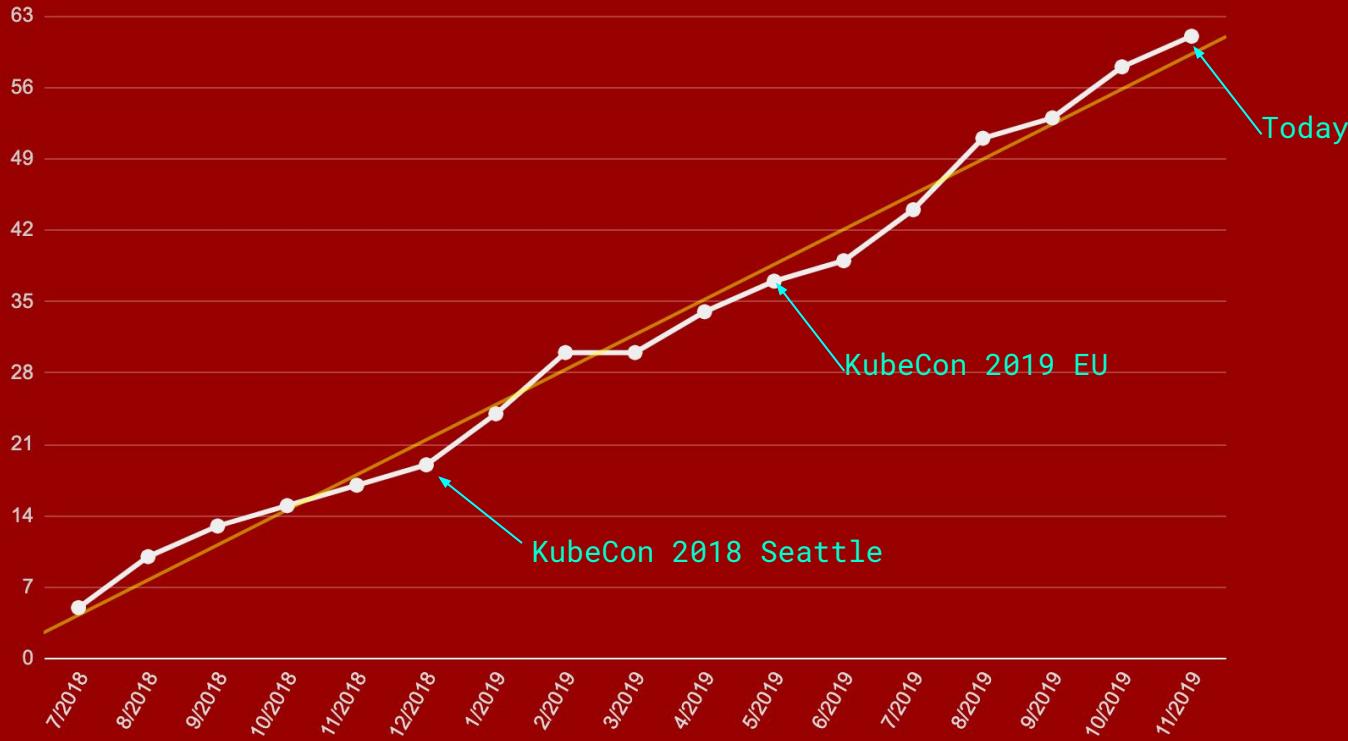
What is Krew?

- Functionality
 - discover plugins
 - install plugins
 - update plugins
 - remove plugins
- Started in 2018 as an intern project at Google Cloud.
- Donated to Kubernetes project in May 2019.
- Not an actual package manager
 - Doesn't handle dependencies.

kubectl plugins on Krew



kubectl plugins on Krew



demo: let's try

krew

why did we need
kubectl plugins?

1.
people want to add
new sub-commands
to kubectl

is:issue label:area/kubectl label:kind/feature

⌚ Feature request: Support OAuth2 authorization flow in kubectl

#54201 by tamalsaha was closed on Jun 5, 2018

area/kubectl kind/feature lifecycle/rotten sig/cli

❗ [Feature request] Get resource usage per namespace using kubectl

#55046 opened on Nov 2, 2017 by gheon

area/kubectl kind/feature lifecycle/stale sig/cli

⌚ kubectl get pods should report last restart time?

#54491 by smarterclayton was closed on Jul 25, 2018

area/kubectl kind/feature lifecycle/rotten sig/cli

⌚ Feature request: Support reading secrets from terminal in kubectl

#54200 by tamalsaha was closed on Jun 5, 2018

area/kubectl kind/feature lifecycle/rotten sig/cli

⌚ kubectl top node/pod should have option like "--watch"

#52963 by u2takey was closed on Aug 16, 2018

area/kubectl kind/feature lifecycle/rotten priority/awaiting-more-evidence sig/cli

❗ FR: New kubectl command kubectl debug

#45922 opened on May 16, 2017 by verb

area/kubectl kind/feature priority/important-longterm sig/cli

⌚ need an API for a user to know which actions they can perform

#31292 by deads2k was closed on Sep 11, 2017 v1.8

area/kubectl kind/feature priority/important-soon sig/auth

⌚ PROPOSAL: Subcommand for easier editing of configmap keys

#37965 by juanvallejo was closed on Feb 27, 2018 updated on Feb 27, 2018

lifecycle/rotten

sig/cli

⌚ Create kubectl lint command

#39699 by pwittrock was closed on May 17, 2018

area/kubectl kind/feature lifecycle/rotten priority/i

❗ By default, merge more .kubeconfig files

#9298 opened on Jun 4, 2015 by ghodss

area/kubectl kind/feature lifecycle/frozen priority/ba

⌚ kubectl diff command

#6284 by fgrzadkowski was closed on Feb 13, 2018

area/app-lifecycle area/kubectl help wanted kind/fe

⌚ kubectl export and clone

#13038 by bgrant0607 was closed on Jan 16, 2016

area/app-lifecycle area/kubectl kind/feature priority

⌚ [WIP]exec: support exec as a specific user

#81883 opened on Aug 23 by max88991

area/kubectl area/kubelet cncf-cla: yes do-not-merge

⌚ kubectl port-forward should allow forwarding to a Service

#15180 by thockin was closed on Feb 15, 2018

area/kubectl help wanted kind/feature priority/important-soon sig/cli

⌚ kubectl login should support the Google OAuth device flow

#35906 by justinsb was closed on Mar 11, 2018

area/kubectl kind/feature lifecycle/rotten priority/backlog sig/auth sig/cli

@ahmetb

2.
everybody has
opinions

 [johanhaleby / kubetail](#)

 Star

1.6k

Bash script to tail Kubernetes logs from multiple pods at the same time

 [wercker / stern](#)

 Star

2.9k

⌚ Multi pod and container log tailing for Kubernetes

 [dtan4 / k8stail](#)

 Star

97

tail -f experience for Kubernetes Pods

 [boz / kail](#) ✓

 Star

888

kubernetes log viewer

 [atombender / ktail](#) ✓

 Star

102

ktail is a tool to easily tail Kubernetes logs.

3.
reaching **consensus**
in open source
is not trivial

Proposal: make `kubectl` respect `KUBECTL_{CONTEXT,NAMESPACE}` environment variables

 Closed

colemickens opened this issue on Jun 13, 2016 · 54 comments

Cole Mickens on Jun 13, 2016 • edited ▾

Member +  ...

I'm curious if there is interest in creating some environment variables (`KUBECTL_CONTEXT` and `KUBECTL_NAMESPACE`) to make `kubectl` a bit easier to use when one has multiple clusters configured in `~/.kube/config` .

The alternative to these environment variables either require a lot of typing (specify `--context` and `--namespace` to all `kubectl` commands) or requires changing the global context, such that it affects other looping `kubectl` invocations. Or as I've done so far, wrapper functions in my shell, but I've been unable to setup my environment such that I can have helper functions without breaking the tab completion for `kubectl` .

Is there interest in this? Or asked differently, would such a PR be accepted?



165



7



15



Proposal: make kubectl respect KUBECTL_{CONTEXT,NAMESPACE} environment variables

Closed

colemickens opened this issue on Jun 13, 2016 · 54 comments

Cole Mickens on Jun 13, 2016 • edited

ahmetb / kubectx

Watch

85

Star

5.7k

Fork

378

I'm
to r

The
to a
inv
such

Switch faster between clusters and namespaces in kubectl <https://kubectx.dev>

kubernetes

kubectl

kubectx-plugins

Is there interest in this? Or asked differently, would such a PR be accepted?



165



7

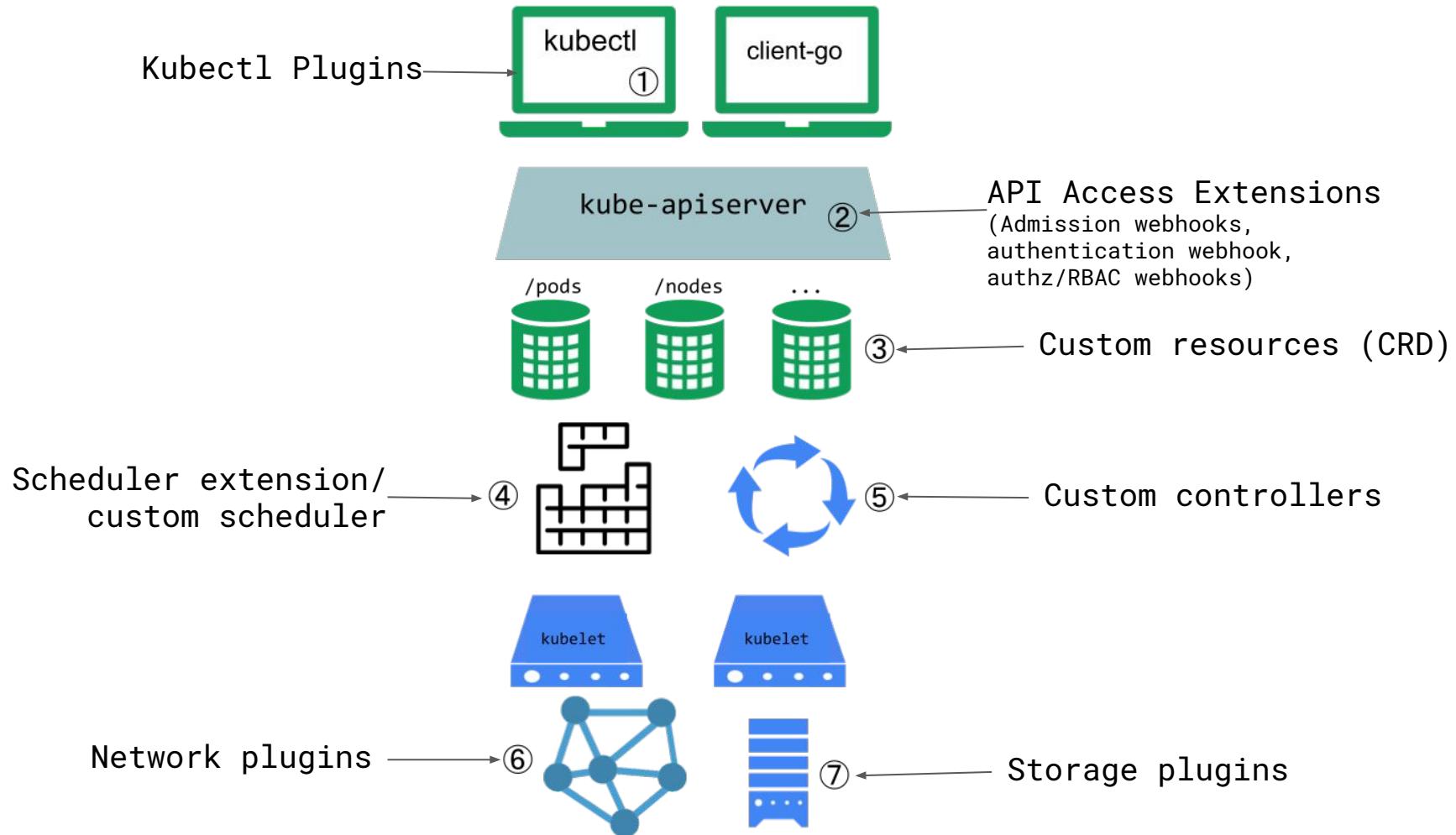


15



@ahmetb

4 .
we cannot solve
everything in the
kubernetes core



5.
kubernetes core
moves slower



official command	plugin
KEP + approval	no approvals
usefulness and stability	no restrictions
hosted in kubectl codebase (Go only)	any language
tied to Kubernetes release cycle	release at your own pace
has to be consistent with kubectl	has room for creativity
takes 0(months)..0(years) from alpha→beta→stable	develop & distribute in 0(hours)

6 .
people are
creative

		NAME	LIST	CREATE	UPDATE	DEL
Namespace	Name	Kind	Ready	Reason		
default	hello	Service	True			
default	hello	Configuration	True			
default	hello-zlr6t	Revision	True			
default	hello-zlr6t-deployment	Deployment	-			
default	hello-zlr6t-deployment-65695556d	ReplicaSet	-			
default	hello-zlr6t-cache	Image	-			
default	hello-zlr6t	PodAutoscaler	False	NoTraffic		
default	hello-zlr6t-metrics	Service	-			
default	hello-zlr6t	Metric	True			
default	hello-zlr6t	ServerlessService	Unknown	NoHealthyBackends		
default	hello-zlr6t-qs5k5	Service	-			
default	hello-zlr6t	Service	-			
default	hello-zlr6t	Endpoints	-			
default	hello-sr94v	Revision	True			
default	hello-sr94v-deployment	Deployment	-			
default	hello-sr94v-deployment-65989ddcc4	ReplicaSet	-			

kui> kubectl kui get pods

NAME	READY	STATUS	RESTARTS	AGE
details-v1-6865b9b99d-rjjvp	1/1	Running	0	12d
productpage-v1-f8c8fb8-x64fn	1/1	Running	0	12d
ratings-v1-77f657f55d-l5wzv	1/1	Running	0	12d
reviews-v1-6b7f6db5c5-gwqrz	1/1	Running	0	12d
reviews-v2-7ff5966b99-sspnq	1/1	Running	0	12d
reviews-v3-5df889bcff-wz8sq	1/1	Running	0	12d

kui> kubectl kui get pods --ui

kui> kubectl kui status https://raw.githubusercontent.com/ahmetb/kube-bookinfo.yaml

NAME	KIND	NAMESPACE	MESSAGE
details	Service	default	
details-v1	Deployment	default	
ratings	Service	default	
ratings-v1	Deployment	default	
reviews	Service	default	
reviews-v1	Deployment	default	
reviews-v2	Deployment	default	
reviews-v3	Deployment	default	
productpage	Service	default	
productpage-v1	Deployment	default	
name	Kind	Namespace	
details-v1-6865b9b99d-rjjvp	Pod	default	
ratings-v1-77f657f55d-l5wzv	Pod	default	
reviews-v1-6b7f6db5c5-gwqrz	Pod	default	
reviews-v2-7ff5966b99-sspnq	Pod	default	
reviews-v3-5df889bcff-wz8sq	Pod	default	
productpage-v1-f8c8fb8-x64fn	Pod	default	

kui> kubectl kui status https://raw.githubusercontent.com/ahmetb/kube-bookinfo.yaml --ui

kui>

get pods details-v1-6865b9b99d-rjjvp

details-v1-6865b9b99d-rjjvp

```

apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/psp: ibm-privileged-psp
  creationTimestamp: 2018-11-30T23:54:47Z
  generateName: details-v1-6865b9b99d-
  labels:
    app: details
    pod-template-hash: "2421656558"
    version: v1
  name: details-v1-6865b9b99d-rjjvp
  namespace: default
  ownerReferences:
  - apiVersion: extensions/v1beta1
    blockOwnerDeletion: true
    controller: true
    kind: Deployment
    name: details-v1-6865b9b99d
    uid: 51bb0fc8-f4fb-11e8-8485-6a5d33db9ea3
  resourceVersion: "1281697"
  selfLink: /api/v1/namespaces/default/pods/details-v1-6865b9b99d-rjjvp
  uid: 51bb0fc8-f4fb-11e8-8485-6a5d33db9ea3
spec:
  containers:
  - image: istio/examples-bookinfo-details-v1:1.8.0
    imagePullPolicy: IfNotPresent
    name: details
    ports:
    - containerPort: 9080
  
```

YAML Status

...even GUI-based plugins

7.
allow safe
experimentation

experimentation

- deprecating a kubectl command is painful.
- let people experiment, graduate successful ones.

The screenshot shows two GitHub repository cards. The top card is for 'aylei / kubectl-debug' with 909 stars. The bottom card is for 'verb / kubectl-debug' with 27 stars. Both cards feature a star icon and a 'Star' button.

The screenshot shows several GitHub issue cards. At the top is an issue titled 'FR: New kubectl command kubectl debug #45922' with a green 'Open' button and 22 comments. Below it is an issue titled 'Ephemeral Containers #277' with a green 'Open' button and 83 comments. A third issue card for 'Feature Description' is partially visible at the bottom. The GitHub interface includes user avatars, names, and date stamps.

DEVELOPING PLUGINS



kubectl plugin mechanism

in a nutshell

1. write an executable program
2. name it kubectl-something
3. put it in your \$PATH
4. invoke as "kubectl something"

demo: let's develop a plugin!

git.k8s.io/sample-cli-plugin



writing kubectl plugins

plugin lookup

"kubectl a b c"

- is "a" builtin command?
- kubectl-a-b-c ?
- kubectl-a-b ?
- kubectl-a ?

plugin execution runtime

1. look for the plugin executable in PATH
2. execve() syscall
replace kubectl with the plugin process
 - ◆ inherit **environment variables**
 - ◆ inherit **standard streams**
 - ◆ determine **exitcode** of kubectl invocation

why develop a kubectl plugin?

your tool has a weird name and it's not very memorable

rakkess	kubectl access-matrix
ketall	kubectl get-all
ksort	kubectl sort-manifests
kail	kubectl tail



feels more natural

kubectl is missing a feature

⚠ Support the user flag from docker exec in kubectl exec

#30656 opened on Aug 15, 2016 by VikParuchuri ⏲ updated 21 days ago

area/kubectl sig/cli sig/node

⚠ [WIP]exec: supprt exec as a specific user

#81883 opened on Aug 23 by max88991

area/kubectl area/kubelet cncf-cla: yes do-not-merge

```
ahmetb@ahmetb-macbookpro2: ~ — 87X14
$ kubectl krew info exec-as
NAME: exec-as
URI: https://github.com/jordanwilson230/kubectl-plugins/archive/v1.0.2-krew.zip
SHA256: 9bd734b1dc08fc59df3f40418e92b228a830098a27257d454deccc5aa1384cf4
VERSION: v1.0.2-krew
HOMEPAGE: https://github.com/jordanwilson230/kubectl-plugins/tree/krew#kubectl-exec-as
DESCRIPTION:
Run `kubectl exec-as` for usage. Example, `kubectl exec-as -u root zookeeper-0` .
CAVEATS:
 \
| The node which the pod is running on cannot have more than one taint.
/_
```

kubectl is missing a feature

⚠ Need a "whoami" sub-command for kubectl

#65798 by [yue9944882](#) was closed on Jul 5, 2018 [🕒](#)

[kind/feature](#) [sig/cli](#)

● ● ● ahmetb@ahmetb-macbookpro2: ~

```
$ kubectl whoami  
ahmetb@google.com
```

\$

kubectl is missing a feature

kubectl export and clone

#13038 by [bgrant0607](#) was closed on Jan 16, 2016

[area/app-lifecycle](#) [area/kubectl](#) [kind/feature](#) [priority](#)

```
ahmetb@ahmetb-macbookpro2: ~
$ kubectl krew info eksporter
NAME: eksporter
URI: https://github.com/Kyrremann/kubectl-eksporter/releases/download/v1.1.0/eksporter.tar.gz
SHA256: 65124766f929c5bcab1f89b0b052b8e2f64c9dd9cb4daf062123c8c632668d5
VERSION: v1.1.0
HOMEPAGE: https://github.com/Kyrremann/kubectl-eksporter
DESCRIPTION:
This plugin helps you export a resource, and removes a pre-defined set of
fields. This is useful for when you have resource you want to migrate from
one namespace or cluster, to another one, but need to edit/add some fields
before applying.
```

If you just need to move one resource, it's probably easier to just:
`kubectl get -oyaml | kubectl create -f -`

Run the tool with:
`kubectl eksporter <resource> <resource_name>`
Or with piping:
`kubectl get pods -o yaml <resource_name> | kubectl eksporter`

```
ahmetb@ahme
$ kubectl krew info neat
```

```
NAME: neat
URI: https://github.com/itaysk/kubectl-neat/rele
SHA256: 0d8e3761c44edd826034a8e7c7bce36f63db725
VERSION: v1.0.0
HOMEPAGE: https://github.com/itaysk/kubectl-neat
DESCRIPTION:
```

If you try to `kubectl get` resources you have just created,
they be unreadably verbose. `kubectl-neat` cleans that up by
removing default values, runtime information, and other internal fields.

Examples:

```
`$ kubectl get pod mypod -o yaml | kubectl neat`  

`$ kubectl neat pod mypod -o yaml`
```

address shortcomings of kubectl

```
ahmetb@ahmetb-r: ~ % $ kubectl get all --namespace=default  
NAME          TYPE      CLUSTER-IP        AGE  
service/kubernetes   ClusterIP   10.4.0.1    154d  
$ [REDACTED]
```

kubectl get all:
lies to you

```
ahmetb@ahmetb-macbookpr: ~ % $ kubectl get-all --namespace default  
NAME          NAMESPACE  AGE  
endpoints/kubernetes  default  154d  
limitrange/limits  default  154d  
resourcequota/gke-resource-quotas  default  154d  
secret/default-token-g4lxh  default  154d  
secret/istio.default  default  19d  
serviceaccount/default  default  154d  
service/kubernetes  default  154d  
$ [REDACTED]
```

plugin: tells you the truth

enhances existing kubectl command

```
ahmetb@ahmetb-macbookpro2: ~ — 48x9
$ kubectl auth can-i delete serviceaccounts
yes
$ kubectl auth can-i create namespaces
yes
$ kubectl auth can-i list users
yes
$ kubectl auth can-i delete bindings
yes
```

builtin: query one
resource at a time

NAME	DESCRIPTION	INSTALLED			
		LIST	CREATE	UPDATE	DELETE
access-matrix	Show an RBAC access matrix for server resources	yes			
\$ kubectl access-matrix					
adapters.config.istio.io		✓	✓	✓	✓
apikeys.config.istio.io		✓	✓	✓	✓
apiservices.apiregistration.k8s.io		✓	✓	✓	✓
attributemanifests.config.istio.io		✓	✓	✓	✓
authorizations.config.istio.io		✓	✓	✓	✓
backendconfigs.cloud.google.com		✓	✓	✓	✓
bindings			✓		
bypasses.config.istio.io		✓	✓	✓	✓
certificates.certmanager.k8s.io		✓	✓	✓	✓
certificates.networking.internal.knative.dev		✓	✓	✓	✓
certificatesigningrequests.certificates.k8s.io		✓	✓	✓	✓
challenges.certmanager.k8s.io		✓	✓	✓	✓

plugin: shows a matrix
of all capabilities

enhances existing kubectl command

```
ahmetb@ahmetb-macbookpro2: ~
$ kubectl auth can-i delete namespace --as tom@google.com
no
$ kubectl auth can-i delete namespace --as omar@google.com
no
$ kubectl auth can-i delete namespace --as jeff@google.com
yes
$ kubectl auth can-i delete namespace --as ahmetb@google.com
yes
$
zsh
```

builtin: query one user at a time

```
ahmetb@ahmetb-macbookpro2: ~
$ kubectl who-can delete namespaces
No subjects found with permissions to delete namespaces assigned through RoleBindings

CLUSTERROLEBINDING                                     SUBJECT          TYPE
cluster-admin                                         system:masters   Group
cluster-admin-binding                                 ahmetb@google.com User
jeff-is-admin                                         jeff@google.com  User
knative-serving-controller-admin                     controller       ServiceAccount
sundar-is-admin                                       sundar@google.com User
system:controller:clusterrole-aggregation-controller clusterrole-aggregation-controller ServiceAccount
system:controller:generic-garbage-collector           generic-garbage-collector ServiceAccount
system:controller:namespace-controller                namespace-controller ServiceAccount
$
```

plugin: queries for all subjects in the cluster

so you have a
plugin, now what?

- packaging
- distribution
- release

plugin developer problems

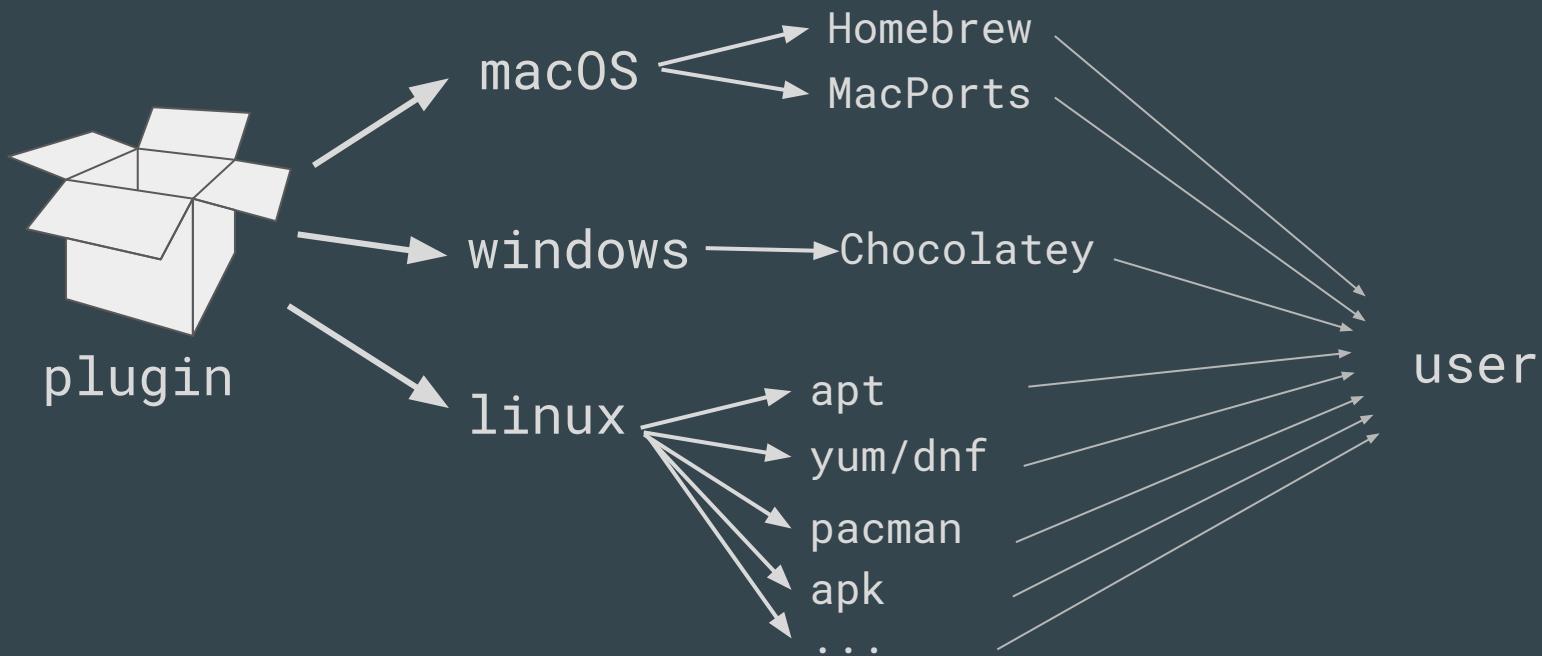
- How can people **hear** about my plugin?
- How can I **package** my plugin for others to install?
- How can I **deliver updates**?

If you're a kubectl plugin developer,
Krew solves these problems.

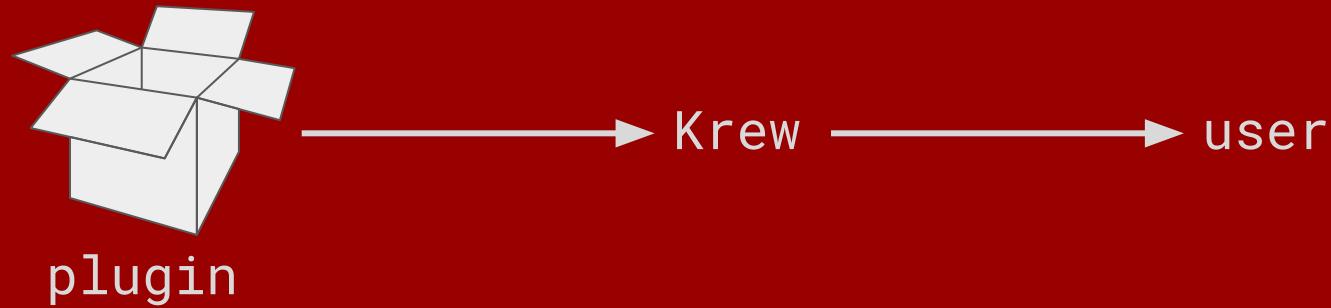
packaging/distribution



packaging/distribution



Krew to rescue!



Without krew

Installation

Installation via krew

```
$ kubectl krew install grep
```

how to update?

how to uninstall?

Manual Installation

```
$ export KUBECTL_VERSION="v1.2.1"
$ wget https://github.com/guessi/kubectl-grep/releases/download/${KUBECTL_VERSION}/kubectl-grep-`uname -s`-`uname -m`.tar.gz
$ tar zxvf kubectl-grep-`uname -s`-`uname -m`.tar.gz
$ mv kubectl-grep /usr/local/bin
$ chmod +x /usr/local/bin/kubectl-grep
```

Krew helps...

...users to:

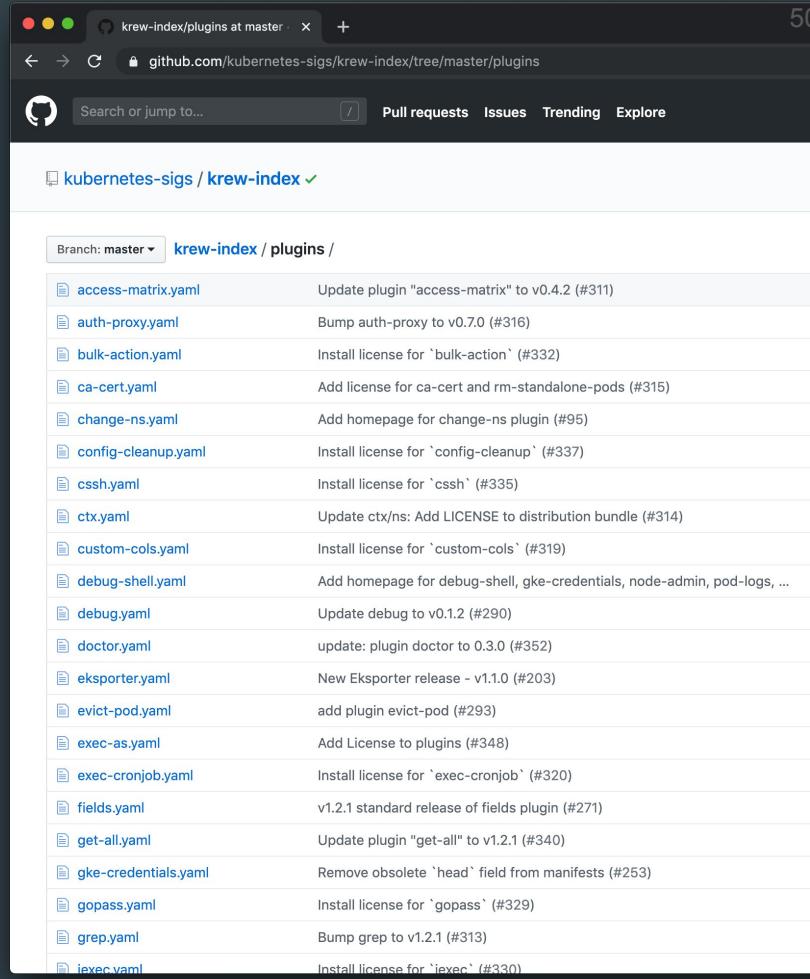
- install plugins
- keep them up to date
- remove plugins cleanly

...developers to:

- make their plugins discoverable by users
- package their plugins for multiple platforms
- deliver updates

centralized discovery

Krew currently operates with a centralized package index.
sig.sigs.k8s.io/krew-index



A screenshot of a GitHub repository page for 'kubernetes-sigs/krew-index'. The repository has 50 stars. The 'Branch: master' dropdown is selected. The 'krew-index/plugins' folder contains 33 files, each with a brief description of its purpose and the pull request number it was merged from. The files include access-matrix.yaml, auth-proxy.yaml, bulk-action.yaml, ca-cert.yaml, change-ns.yaml, config-cleanup.yaml, cssh.yaml, ctx.yaml, custom-cols.yaml, debug-shell.yaml, debug.yaml, doctor.yaml, eksporter.yaml, evict-pod.yaml, exec-as.yaml, exec-cronjob.yaml, fields.yaml, get-all.yaml, gke-credentials.yaml, gopass.yaml, grep.yaml, and iexec.yaml.

File	Description	Pull Request
access-matrix.yaml	Update plugin "access-matrix" to v0.4.2 (#311)	#311
auth-proxy.yaml	Bump auth-proxy to v0.7.0 (#316)	#316
bulk-action.yaml	Install license for `bulk-action` (#332)	#332
ca-cert.yaml	Add license for ca-cert and rm-standalone-pods (#95)	#95
change-ns.yaml	Add homepage for change-ns plugin (#95)	#95
config-cleanup.yaml	Install license for `config-cleanup` (#337)	#337
cssh.yaml	Install license for `cssh` (#335)	#335
ctx.yaml	Update ctx/ns: Add LICENSE to distribution bundle (#314)	#314
custom-cols.yaml	Install license for `custom-cols` (#319)	#319
debug-shell.yaml	Add homepage for debug-shell, gke-credentials, node-admin, pod-logs, ...	#319
debug.yaml	Update debug to v0.1.2 (#290)	#290
doctor.yaml	update: plugin doctor to 0.3.0 (#352)	#352
eksporter.yaml	New Eksporter release - v1.1.0 (#203)	#203
evict-pod.yaml	add plugin evict-pod (#293)	#293
exec-as.yaml	Add License to plugins (#348)	#348
exec-cronjob.yaml	Install license for `exec-cronjob` (#320)	#320
fields.yaml	v1.2.1 standard release of fields plugin (#271)	#271
get-all.yaml	Update plugin "get-all" to v1.2.1 (#340)	#340
gke-credentials.yaml	Remove obsolete `head` field from manifests (#253)	#253
gopass.yaml	Install license for `gopass` (#329)	#329
grep.yaml	Bump grep to v1.2.1 (#313)	#313
iexec.yaml	Install license for `iexec` (#330)	#330

packaging & distributing with Krew

1. make your plugin available as a `.tar.gz` or `.zip` archive
2. write a **plugin manifest**
3. submit a pull request to **krew-index**
4. party! 

plugin manifest

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: "access-matrix"
spec:
  version: "v0.4.0"
  homepage: "https://github.com/corneliusweig/rakkess"
  platforms:
    - <PLATFORM1>
    - <PLATFORM2>
    - <PLATFORM3>
    - ...
```

plugin manifest - platform

```
... # continued
spec:
  platforms:
    - selector:
        matchLabels:
          os: linux
          arch: amd64
      uri: https://github.com/corneliusweig/rakkess/releases/v0.4.0/bundle.tar.gz
      sha256: 7a16c61dfc4e2924fdedc894d59db7820bc4643a58d9a853c4eb83eadd4deee8
    files:
      - from: ./rakkess_linux-amd64
        to: ./rakkess
      - from: ./LICENSE
        to: "."
bin: ./rakkess
```

recipes for success

1. name your plugin for clarity
(see *Krew Naming Guide*)
2. reuse flags/options and KUBECONFIG detection from
`k8s.io/cli-runtime/pkg/genericclioptions`
3. import auth plugins for cloud clusters
`import _ "k8s.io/client-go/plugin/pkg/client/auth"`

wrap up



Join us make kubectl plugins better!

- Become a kubectl plugin developer,
submit your plugins to [Krew.dev](#).
- Join developing the [**Krew**](#) project.
 - you don't need to be a Kubernetes expert
 - clean and young codebase
 - low bug count, high creativity
- Bring your ideas to [**Kubernetes SIG CLI**](#).

thanks!

sigs.k8s.io/krew

git.k8s.io/sample-cli-plugin

twitter.com/ahmetb

github.com/ahmetb