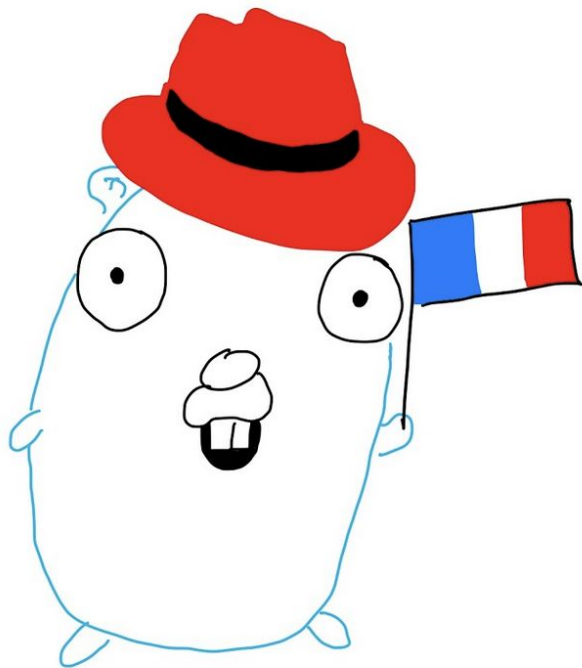
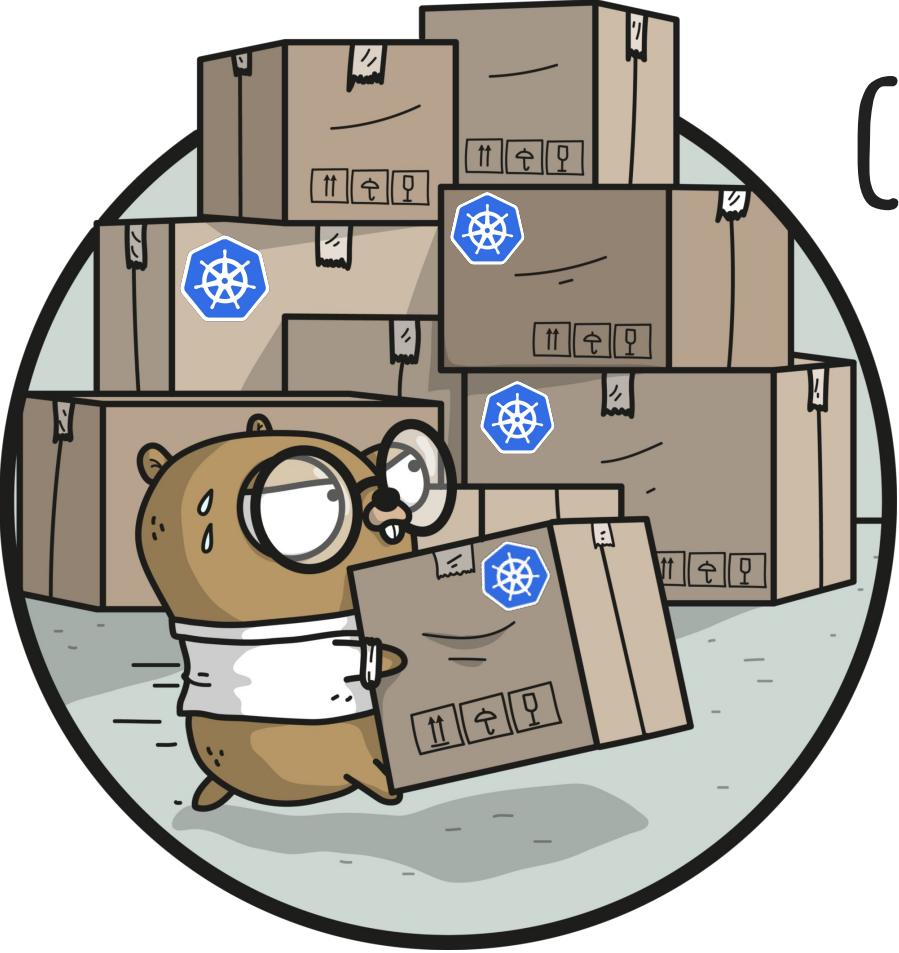




22 JUIN 2021



CREATE & DISTRIBUTE
A PLUGIN FOR
KUBERNETES.
EASY!

AURÉLIE VACHE

@aurelievache

Cloud Engineer at Stack Labs

Google Developer Expert
on Google Cloud

Conferences organizer

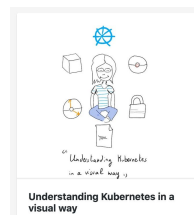
Duchess France Leader & Mentor

Technical articles & sketchnotes writer

... & Retrogaming lover



TOULOUSE
DATA
SCIENCE



Les Productions de MOA



GAËLLE ACAS

@Gaelleacas

SRE at Stack Labs



CNCF Nantes
Co-organizer

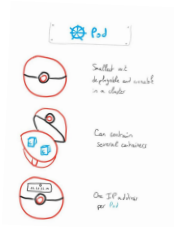


Technical articles writer

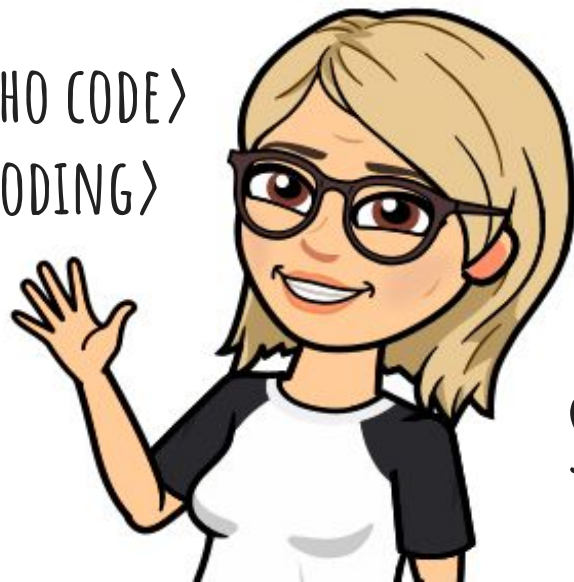


CLOUD
ADDICT

Sketchnote
reviewer



<MUM WHO CODE>
<KIDS CODING>



SPEAKER

WHY THIS TALK?

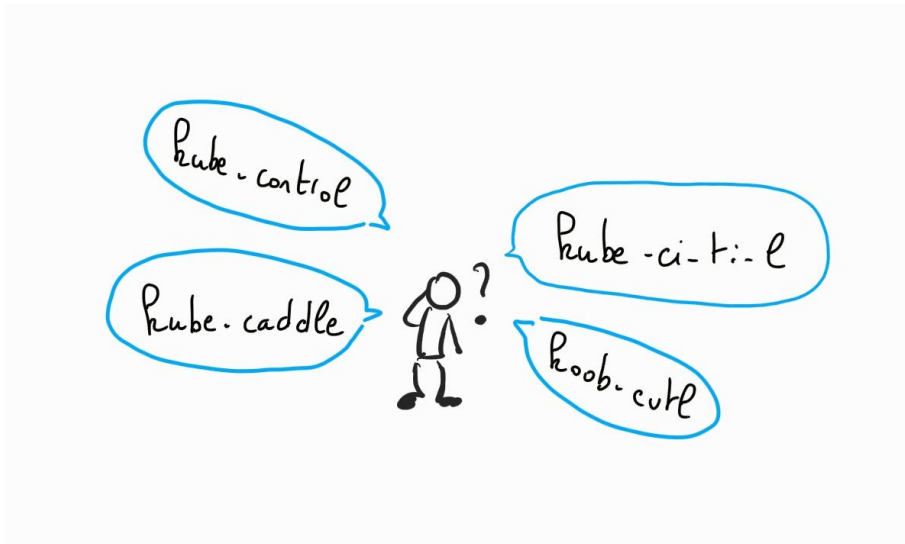
- `$ kubectl` CLI is awesome but not exhaustive
- Painful to extend kubernetes/kubectl CLI
- Easy to create your own plugins
- Possible to spread your plugin worldwide!





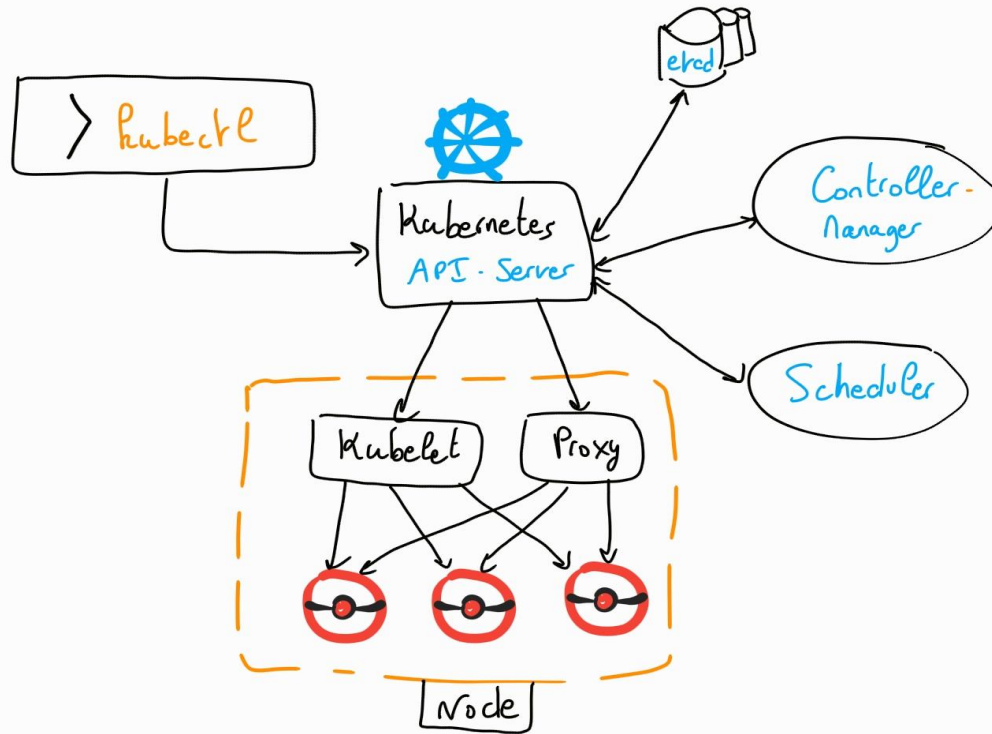
KUBECTL > PRONUNCIATION FIGHT

Disclaimer



Pronounce “kubectl” as you want :-)

KUBECTL > KUBERNETES TOOL/CLI



KUBECTL > A VERY LOGICAL CLI!



I want the list of Pods
in my-namespace namespace



```
$ kubectl get pods -n my-namespace
```

KUBECTL > A VERY LOGICAL CLI!



I want to scale a deployment
my-deploy to 5 replicas



```
$ kubectl scale deploy my-deploy --replicas=5
```

KUBECTL > A VERY LOGICAL CLI!



```
$ kubectl get pods -n my-namespace
$ kubectl describe pod your-pod
$ kubectl logs my-pod
$ kubectl delete super-pod
$ kubectl create ns my-ns
$ kubectl run tmp --image=my-img:tag
$ kubectl exec a-pod -it -- sh
```

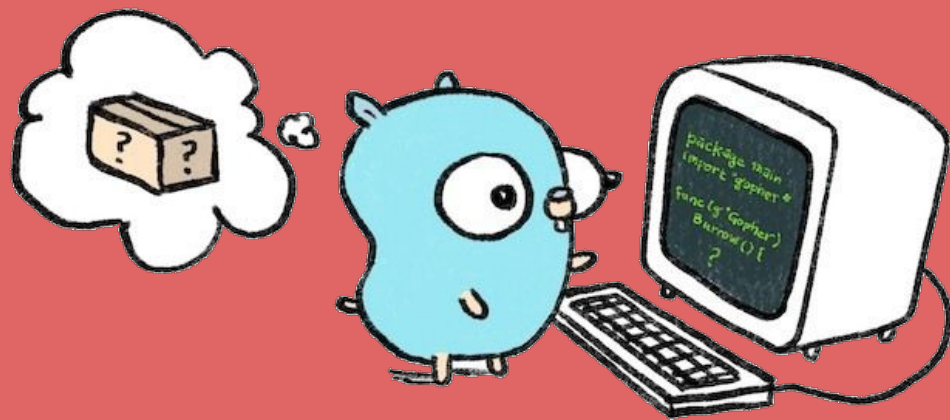
KUBECTL > COOL, BUT CAN I EXTEND IT?



```
$ kubectl get pods,deploy
```

NAME	READY	STATUS	RESTARTS	AGE
pod/my-pod-c9cb9cbf6-r2gnc	1/1	Running	0	2d6h
pod/my-pod-75897978cd-f28j9	1/1	Running	0	3d1h

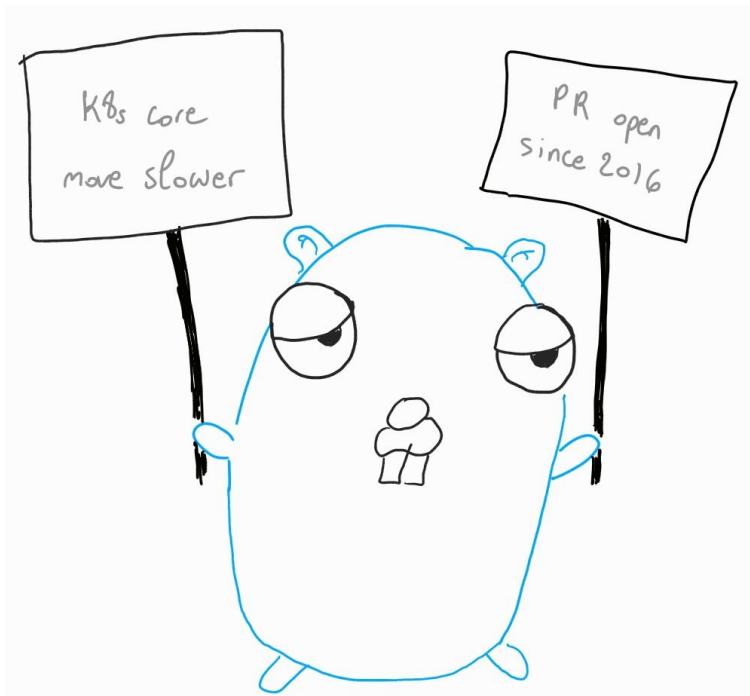




KUBECTL PLUGINS

KUBECTL: TIED TO CORE RELEASE CYCLE

Wants to implement a new feature in Kubectl?

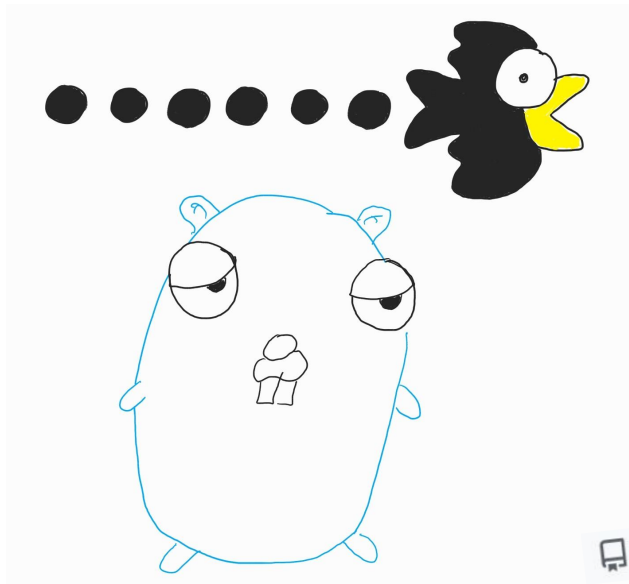


So ... Forget about it and think about plugins instead ;-)

PLUGINS > K[TOOL] OR TOO MANY TOOLS?

Too many opinions about one need

 [wercker / stern](#)



 [johanhaleby / kubetail](#)

 [atombender / ktail](#)

PLUGINS > K[TOOL] OR TOO MANY TOOLS?

Not easily understandable

ksort 🤔

kubectl sort-manifest 😎

No common way to install them

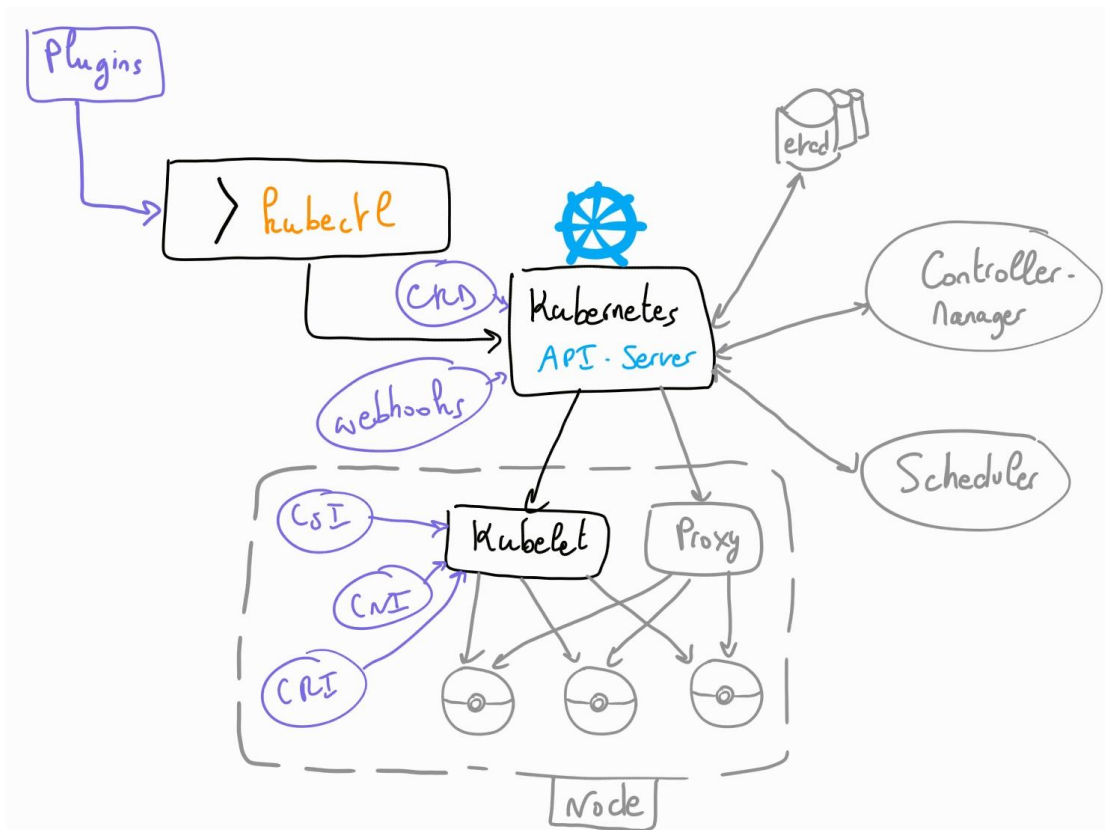
apt-get install

brew install

git clone ...

And... it's not easy to know them!

KUBERNETES, AN EXTENSIBLE WORLD



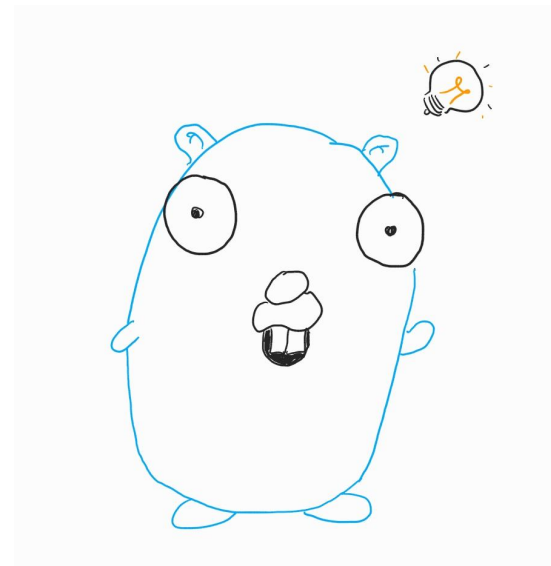
PLUGINS > WHY IT'S USEFUL?

Extend `kubectl` with extra functionality that still feels like `kubectl`

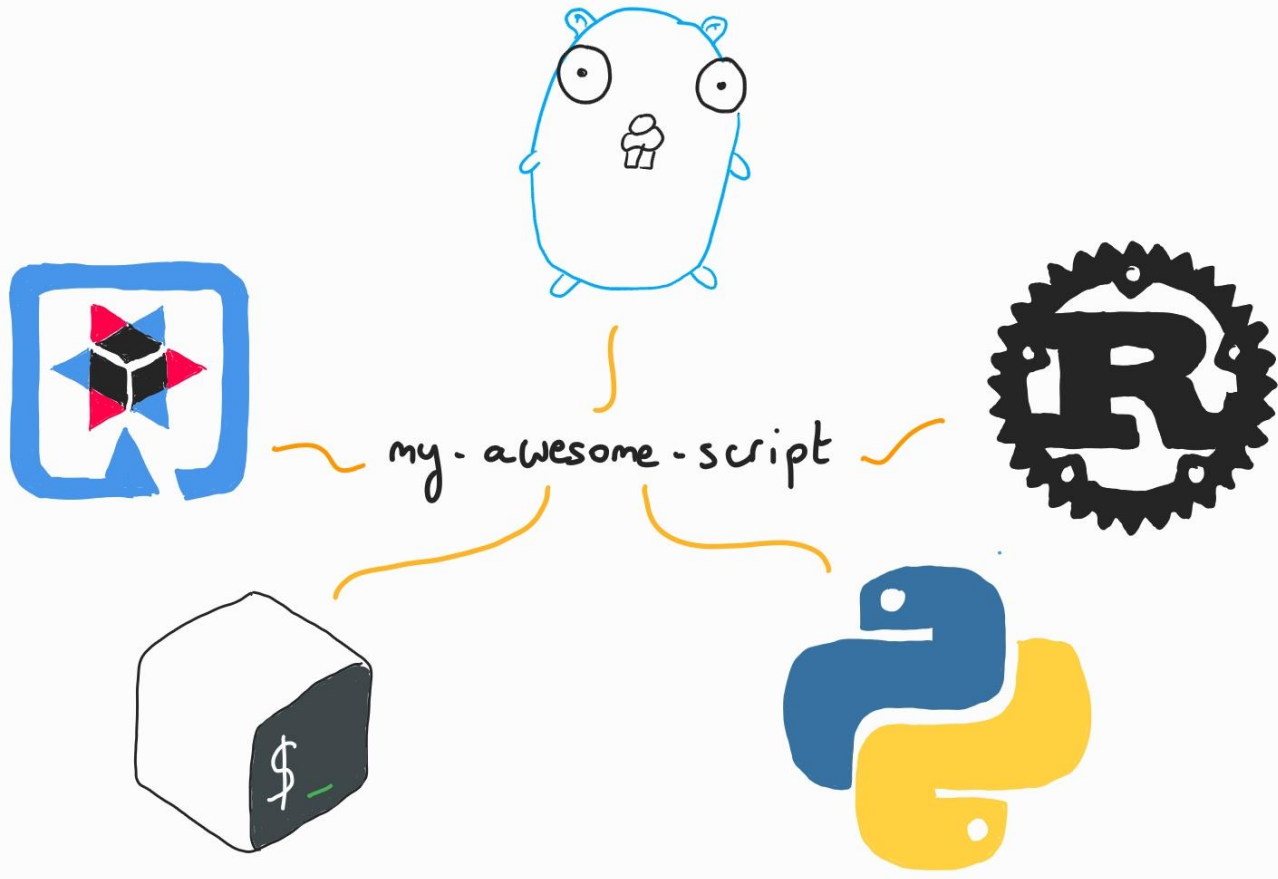
- e.g. `kubectl service-catalog [...]`

Encapsulate repetitive workflow

- e.g. checking a secret resource, check logs

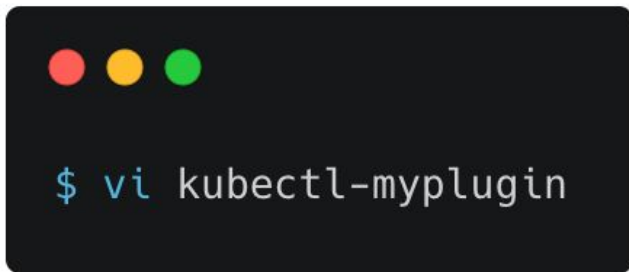


PLUGINS > LANGUAGE AGNOSTIC!

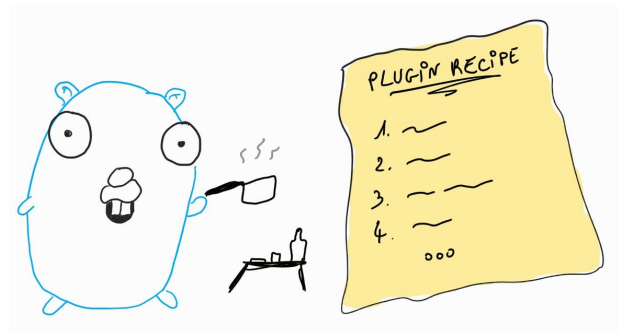


KUBECTL PLUGIN > HOW IT WORKS?

1. Create a file named `kubectl-myplugin`



```
$ vi kubectl-myplugin
```

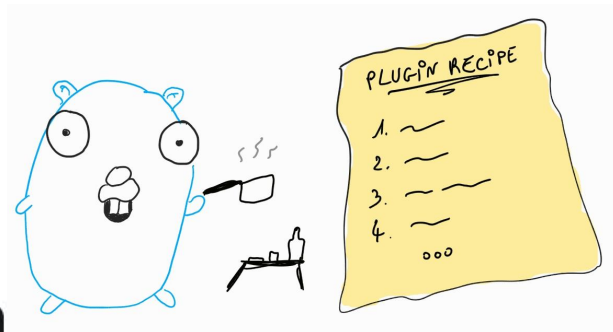


KUBECTL PLUGIN > HOW IT WORKS?

1. Create a file named `kubectl-myplugin`
2. **Make it executable**

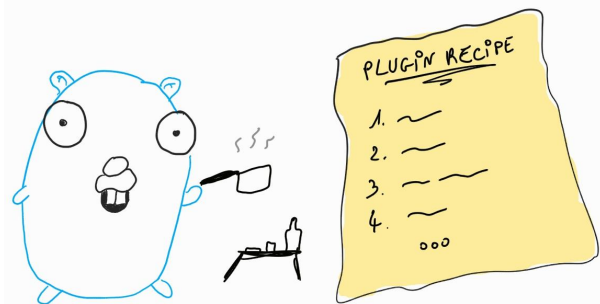


```
$ chmod +x kubectl-myplugin
```



KUBECTL PLUGIN > HOW IT WORKS?

1. Create a file named `kubectl-myplugin`
2. Make it executable
3. **Put it in your PATH**



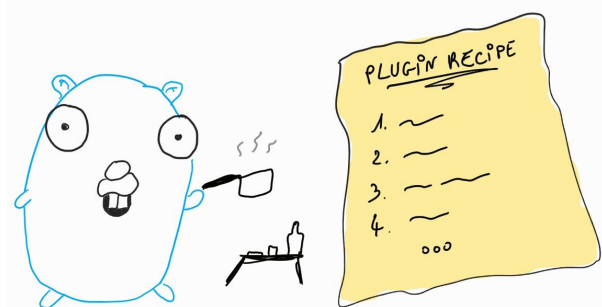
```
$ cp kubectl-myplugin /usr/local/bin/kubectl-myplugin
```

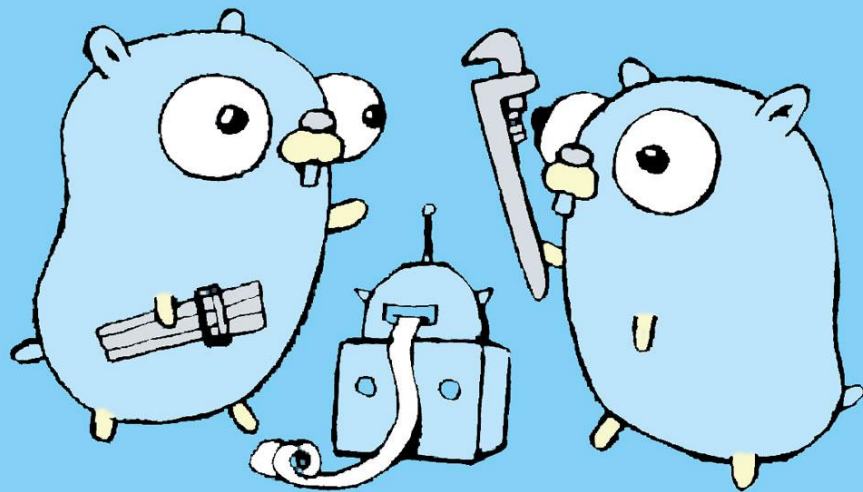
KUBECTL PLUGIN > HOW IT WORKS?

1. Create a file named `kubectl-myplugin`
2. Make it executable
3. Put it in your PATH
4. Run “`kubectl myplugin`”



```
$ kubectl myplugin  
Welcome in our awesome plugin!
```

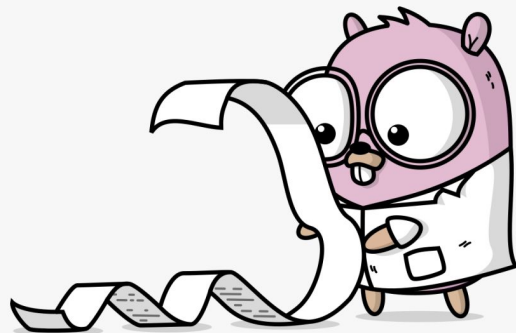




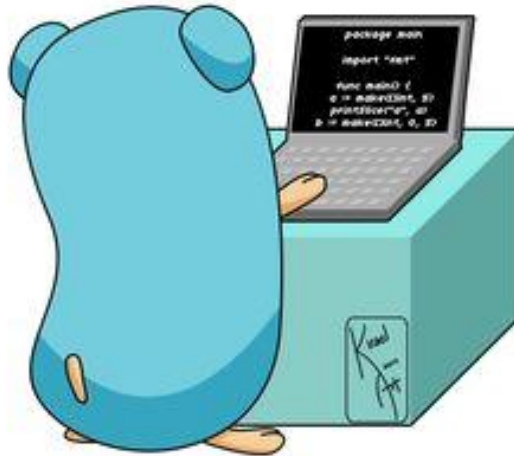
LET'S CREATE A PLUGIN!

WHAT DO WE WANT?

- Display seasonal emojis in “kubectl get” command
- In a simple and easy plugin
- In BASH



LET'S DO IT, IN LIVE!



OUR PLUGIN > STEPS

1. Create the script



```
$ vi kubectl-season
```

OUR PLUGIN > STEPS

2. Add executable permission



```
$ chmod +x kubectl-season
```

OUR PLUGIN > STEPS

3. Initialize our script



```
#!/bin/bash
```

OUR PLUGIN > STEPS

4. Define a list of emojis

```
● ● ●

NOW=`date +%m`

case $NOW in

    01|02|03)
        EMOJIS=(🧊 🍪 🏂 🌲 ❄️ 🧸 🛷 🧤)
    ;;
    04|05)
        EMOJIS=(🐰 🐣 🍫 🥚 🔔 🍪 🌻)
    ;;
    #...

    12)
        EMOJIS=(🌲 🎁 🧑🎄 🦌 🍪 🧸 ❄️ ✨)
    ;;
esac
```

OUR PLUGIN > STEPS

5. Get a randomized emoji



```
SELECTED_EMOJI=${EMOJIS[$(( ( RANDOM % ${#EMOJIS[@]} ) ))]};
```

OUR PLUGIN > STEPS

6. Define & display an usage/help

```
## Usage
n=$'\n'
usage="Display seasonal emoji randomly just before resources name.${n}${n}Usage:${n}    kubectl season
KIND${n}${n}Examples:${n}    kubectl season pod${n}    kubectl season pod,svc${n}    kubectl season deploy -A"

# MAIN
# Usage
if [[ "$1" == "" || "$1" == "-h" || "$1" == "--help" ]]
then
    echo "$usage"
    exit 1
fi
```


OUR PLUGIN > STEPS

7. Execute `kubectl get` command & add seasonal emojis



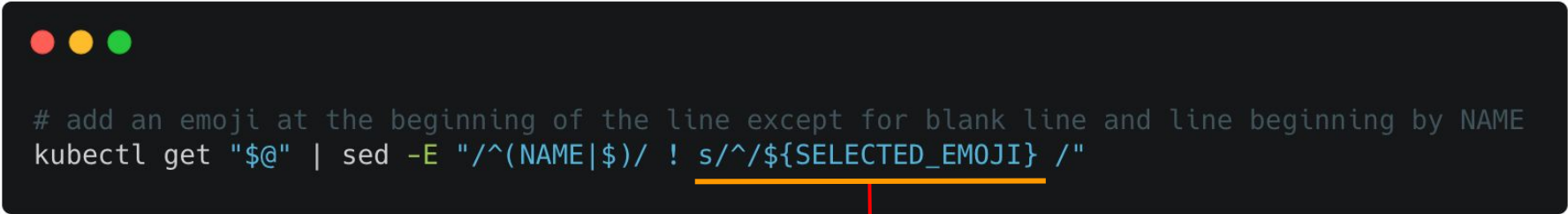
```
# add an emoji at the beginning of the line except for blank line and line beginning by NAME  
kubectl get "$@" | sed -E "/^(NAME|$)/ ! s/^/${SELECTED_EMOJI} /"
```



execute **kubectl**
get command

OUR PLUGIN > STEPS

7. Execute `kubectl get` command & add seasonal emojis



```
# add an emoji at the beginning of the line except for blank line and line beginning by NAME  
kubectl get "$@" | sed -E "/^(NAME|$)/ ! s/^/${SELECTED_EMOJI} /"
```



add **randomized emoji**
at the beginning of
the line

OUR PLUGIN > STEPS

7. Execute `kubectl get` command & add seasonal emojis



```
# add an emoji at the beginning of the line except for blank line and line beginning by NAME  
kubectl get "$@" | sed -E "/^(NAME|$)/ ! s/^/${SELECTED_EMOJI} /"
```



except for line
starting by **NAME** &
blank line

OUR PLUGIN > FINAL! (1/2)

```
#!/bin/bash

NOW=`date +%m`

case $NOW in
  01|02|03)
    EMOJIS=(🌊 🍷 🏊 🌲 ❄️ 🧊 🏠 🌱 🍀)
    ;;
  04|05)
    EMOJIS=(🐉 🍵 🎴 🥚 🔔 🍷 🌻)
    ;;
  06|07|08)
    EMOJIS=(😎 🍌 🍷 🌴 🍷 🌞 🕒)
    ;;
  9)
    EMOJIS=(🍷 🗡️ 🚗 🏠 📺 🍷 ✂️)
    ;;
  10)
    EMOJIS=(🍷 🍷 🍷 🍷 🍷 🍷 🍷 🍷)
    ;;
  11)
    EMOJIS=(🍷 🍷 🍷 🍷 🍷 🍷 🍷 🍷)
    ;;
  12)
    EMOJIS=(🍷 🍷 🍷 🍷 🍷 🍷 🍷 🍷)
    ;;
esac
```

<https://github.com/scraly/kubectl-season/>

OUR PLUGIN > FINAL! (2/2)

```
SELECTED_EMOJI=${EMOJIS[$(( ( RANDOM % ${#EMOJIS[@]} ) ))]};

## Usage
n=${'\n'}
usage="Display seasonal emoji randomly just before resources name.${n}${n}Usage:${n}    kubectl season
KIND${n}${n}Examples:${n}    kubectl season pod${n}    kubectl season pod,svc${n}    kubectl season deploy -A"

# MAIN
if [[ "$1" == "" || "$1" == "-h" || "$1" == "--help" ]]
then
    echo "$usage"
    exit 1
fi

# add an emoji at the beginning of the line except for blank line and line beginning by NAME
kubectl get "$@" | sed -E "/^(NAME|$)/ ! s/^{SELECTED_EMOJI} /"
```

<https://github.com/scraly/kubectl-season/>

OUR PLUGIN > IN KUBECTL?



Cool ... but, how can I
add it in `kubectl`
command?

OUR PLUGIN > IN KUBECTL!



```
$ cp kubectl-season /usr/local/bin/kubectl-season
```

OUR PLUGIN > RESULTS



```
$ kubectl season -h
```

Display seasonal emoji randomly just before resources name.

Usage:

```
    kubectl season KIND
```

Examples:

```
    kubectl season pod
```

```
    kubectl season pod,svc
```

```
    kubectl season deploy -A
```


OUR PLUGIN > RESULTS



```
$ kubectl season pod
```

NAME	READY	STATUS	RESTARTS	AGE
 pod	1/1	Running	0	3h10m
 a-pod-f9fd979d6-6tnbs	1/1	Running	2	26d

OUR PLUGIN > RESULTS

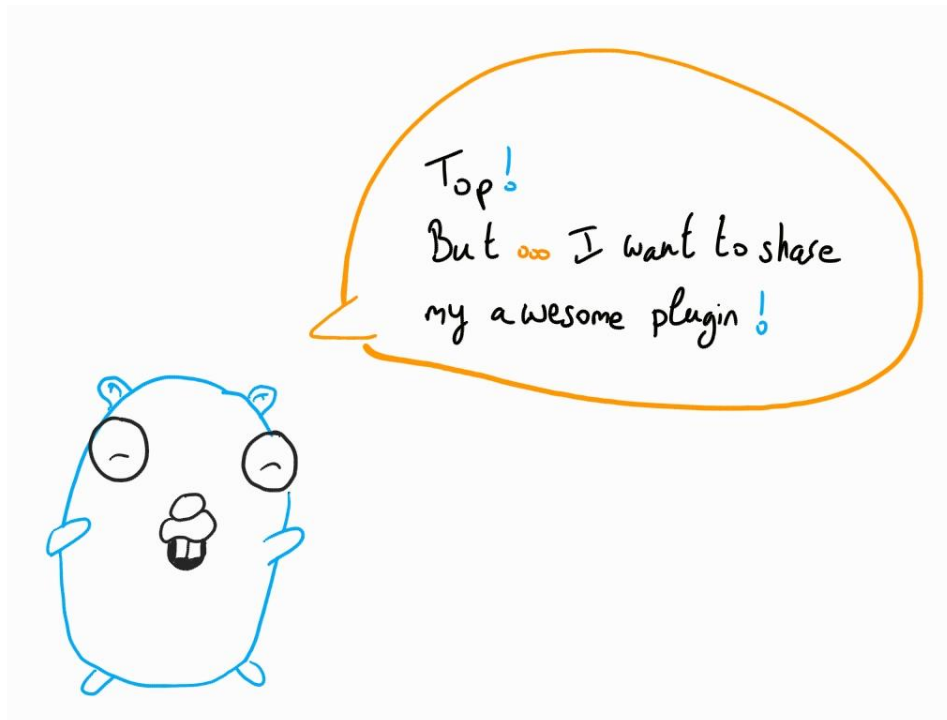


```
$ kubectl season pod,deploy
```

NAME	READY	STATUS	RESTARTS	AGE
🍪 pod/pod	1/1	Running	0	3h12m
🍪 pod/a-pod-f9fd979d6-6tnbs	1/1	Running	2	26d

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
🍪 deployment.apps/my-deploy	2/2	2	2	26d
🍪 deployment.apps/a-deploy	1/1	1	1	26d

OUR PLUGIN > HOW TO SHARE IT?





Krew

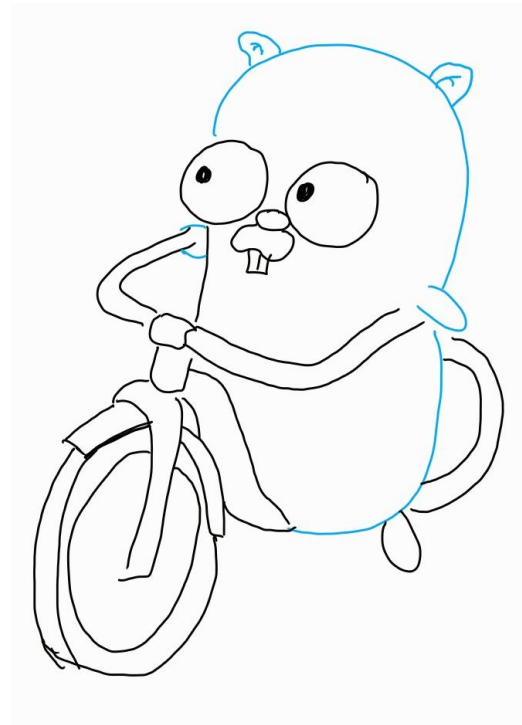
WHAT IS KREW?

- A **package manager for kubectl plugins**
 - Like “brew” but for kubectl :)
 - Developed summer of 2018 by Google intern Luk Burchard ([@lbb](#))
- **Krew is open sourced, try it:**
<https://github.com/kubernetes-sigs/krew>



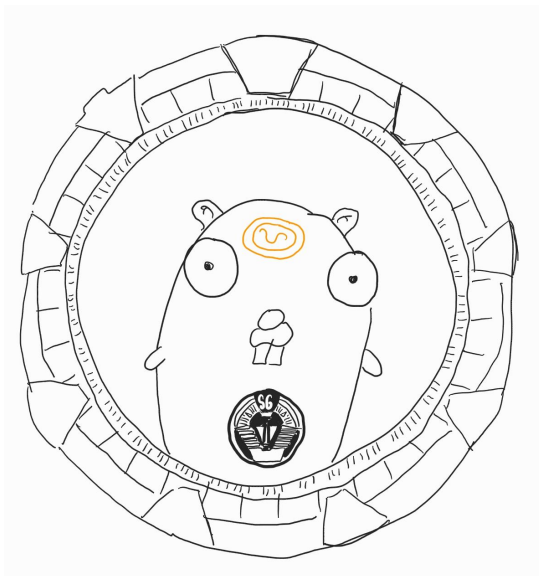
KREW > USEFUL FOR USERS

- Discover kubectl plugins
- Easy way to install plugins
- Update easily plugins



KREW > USEFUL FOR DEVELOPERS

- Make their plugins discoverable
- Easy way to package their plugins for multiplatforms (Win, Linux, MacOS)



KREW

```
$ kubectl krew -h
krew is the kubectl plugin manager.
You can invoke krew through kubectl: "kubectl krew [command]..."
```

Usage:

```
  kubectl krew [command]
```

Available Commands:

help	Help about any command
index	Manage custom plugin indexes
info	Show information about an available plugin
install	Install kubectl plugins
list	List installed kubectl plugins
search	Discover kubectl plugins
uninstall	Uninstall plugins
update	Update the local copy of the plugin index
upgrade	Upgrade installed plugins to newer versions
version	Show krew version and diagnostics

Flags:

-h, --help	help for krew
-v, --v Level	number for the log level verbosity

Use "kubectl krew [command] --help" for more information about a command.

KREW > SEARCH

```
$ kubectl krew search
```

NAME	DESCRIPTION	INSTALLED
access-matrix	Show an RBAC access matrix for server resources	yes
advise-psp	Suggests PodSecurityPolicies for cluster.	no
apparmor-manager	Manage AppArmor profiles for cluster.	no
auth-proxy	Authentication proxy to a pod or service	no
bd-xray	Run Black Duck Image Scans	no
...		
view-utilization	Shows cluster cpu and memory utilization	yes
virt	Control KubeVirt virtual machines using virtctl	no
warp	Sync and execute local files in Pod	no
who-can	Shows who has RBAC permissions to access Kubern...	no
whoami	Show the subject that's currently authenticated...	no

KREW > LIST



```
$ kubectl krew list
```

PLUGIN	VERSION
access-matrix	v0.4.5
graph	v0.1.1
krew	v0.4.0
tree	v0.4.0
view-secret	v0.6.0
view-utilization	v0.3.3

KREW > INSTALL



```
$ kubectl krew install view-utilization
```

KREW > INSTALL NOT PUBLISHED PLUGINS

Your plugin is not already published/available on krew index?

Not problem, you can also install it through Krew!



```
$ kubectl krew install --manifest=.krew.yaml --archive=dist/kubectl-myplugin_v0.0.1.tar.gz
```

KREW > INSTALL NOT PUBLISHED PLUGINS

Your plugin is not already published/available on krew index?

Not problem, you can also install it through Krew!

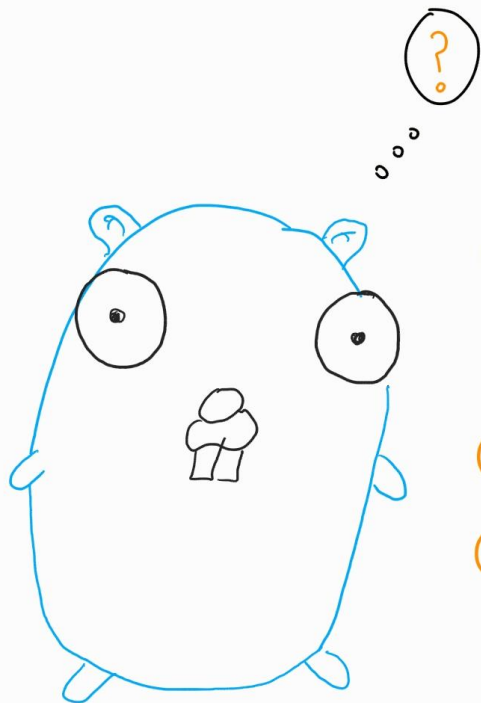


```
$ kubectl krew install --manifest-url=<URL>
```

KREW > LOCAL ARCHITECTURE

```
► tree ~/.krew
/Users/gacas/.krew
├── bin
│   ├── kubectl-krew -> /Users/gacas/.krew/store/krew/v0.4.0/krew
│   └── kubectl-neat -> /Users/gacas/.krew/store/neat/v2.0.1/kubectl-neat
├── index
│   └── default
│       └── plugins.md
├── receipts
│   ├── krew.yaml
│   └── neat.yaml
└── store
    ├── krew
    └── neat
```

PUBLISHING ON KREW



- ① Develop a plugin
- ② Create a Release
- ③ Write a Krew plugin manifest
- ④ Make a Pull Request to Krew index

PUBLISHING ON KREW > MANIFEST FILE

kubectl-season/.krew.yaml

INDEX:

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
        uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
        sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
        bin: kubectl-season-0.0.4/kubectl-season
```


K8s OBJECT



kubectl-season/.krew.yaml

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
version: v0.0.4
description: |
  This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
  just before the resources name. It's a special seasonal mode.
platforms:
- selector:
    matchExpressions:
    - key: os
      operator: In
      values:
      - darwin
      - linux
  uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
  sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
  bin: kubectl-season-0.0.4/kubectl-season
```

kubectl-season/krew.yaml

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
    uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
    sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
    bin: kubectl-season-0.0.4/kubectl-season
```

PLUGIN
INFO



PLATFORMS[]



kubectl-season/.krew.yaml

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
        uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
        sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
        bin: kubectl-season-0.0.4/kubectl-season
```

MATCH
PLATFORM



kubectl-season/.krew.yaml

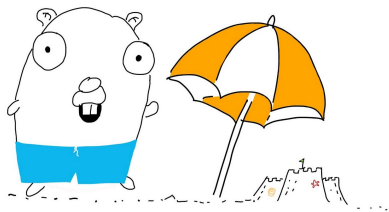
```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
        uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
        sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
        bin: kubectl-season-0.0.4/kubectl-season
```

kubectl-season/krew.yaml

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
    uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
    sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
    bin: kubectl-season-0.0.4/kubectl-season
```

DOWNLOAD





PLUGIN BINARY



kubectl-season/.krew.yaml

```
apiVersion: krew.googlecontainertools.github.com/v1alpha2
kind: Plugin
metadata:
  name: season
spec:
  homepage: https://github.com/scraly/kubectl-season
  shortDescription: Display seasonal emoji randomly before resources name
  version: v0.0.4
  description: |
    This plugin allows you to add randomly a seasonal (halloween, christmas, easter...) emoji
    just before the resources name. It's a special seasonal mode.
  platforms:
    - selector:
        matchExpressions:
          - key: os
            operator: In
            values:
              - darwin
              - linux
        uri: https://github.com/scraly/kubectl-season/archive/v0.0.4.tar.gz
        sha256: 7331819379ca14da988d2537904381b6186c989d527719cb1e0fd212293d749e
        bin: kubectl-season-0.0.4/kubectl-season
```

PUBLISHING ON KREW > PULL REQUEST



Author ▾ Label ▾ Assignee ▾ Sort ▾

-  **add kubectl-creyaml plugin** •
cnf-cla: yes size/S
#886 opened 2 days ago by sahil-lakhwani
-  **Add shovel plugin** •
approved cnf-cla: yes size/M
#883 opened 2 days ago by StupidScience
-  **Redirect plugin list to the site** •
approved cnf-cla: yes do-not-merge/hold size/L
#882 opened 3 days ago by ahmetb
-  **Update tree manifest (remove defaults)** •
approved cnf-cla: yes size/XS
#881 opened 3 days ago by ahmetb
-  **release new version v0.0.37 of modify-secret** ×
cnf-cla: yes needs-rebase size/S
#879 opened 3 days ago by krew-release-bot
-  **Add halloween plugin** ×
cnf-cla: yes size/S
#872 opened 5 days ago by scraly
-  **Add kubectl-config-registry plugin** •
cnf-cla: yes size/M
#857 opened 16 days ago by mumoshu

<https://github.com/kubernetes-sigs/krew-index>

KREW > CREATE YOUR PRIVATE INDEX & PUBLISH INTO IT

New feature: you can create your private index

```
$ kubectl krew index add scaly https://github.com/scaly/krew-index
```

```
$ kubectl krew index list
```

INDEX	URL
default	https://github.com/kubernetes-sigs/krew-index.git
scaly	https://github.com/scaly/krew-index

```
$ kubectl krew search scaly
```

NAME	DESCRIPTION	INSTALLED
scaly/halloween	Display resources with Halloween emojis	yes
scaly/season	Display seasonal emoji randomly before resource...	no

```
$ kubectl krew install scaly/season
```


KREW > UPDATE INDEX



```
$ kubectl krew update
```

KREW > UPGRADE PLUGINS



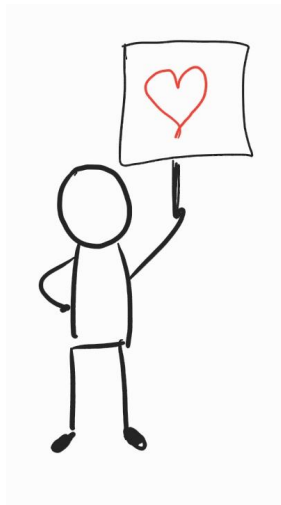
```
// Upgrade all plugins
```

```
$ kubectl krew upgrade
```

```
// OR Upgrade only one plugin
```

```
$ kubectl krew upgrade season
```

KREW > OUR FAV PLUGINS

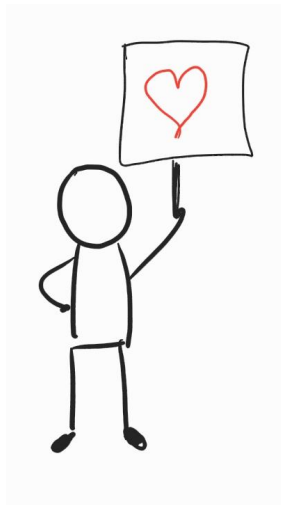


> **view-secret**



```
$ kubectl view-secret mysecret -a
```

KREW > OUR FAV PLUGINS



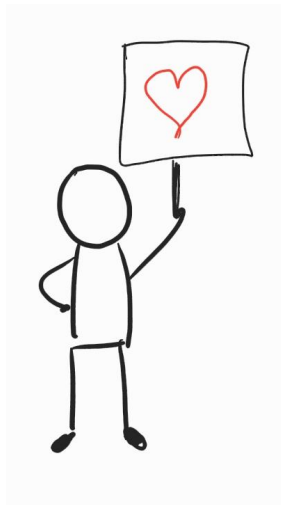
> view-secret

> **view-cert**



```
$ kubectl view-cert -A
```

KREW > OUR FAV PLUGINS

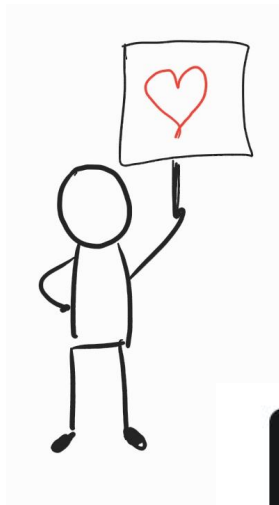


- > view-secret
- > view-cert
- > **view-utilization**



```
$ kubectl view-utilization
```

KREW > OUR FAV PLUGINS



- > view-secret
- > view-cert
- > view-utilization
- > **neat**



```
$ kubectl get deploy mydeployment -o yaml | kubectl neat
```



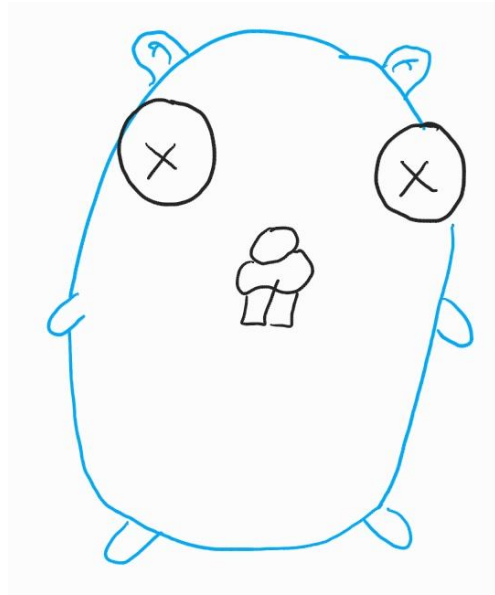
BEST PRACTICES

PLUGIN CREATION > BEST PRACTICES

Naming is important

- **DON'T:** `kubectl-version`

Not possible to create plugins that overwrite existing kubectl commands



PLUGIN CREATION > BEST PRACTICES

Naming is important

- **DON'T:** `kubectl service`

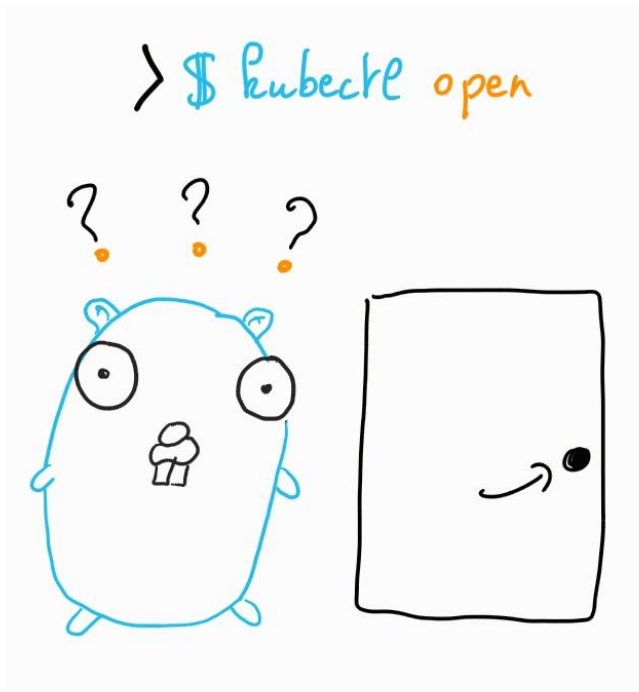
Unclear what the plugin is doing with service

- **DON'T:** `kubectl open`

Unclear what it is opening

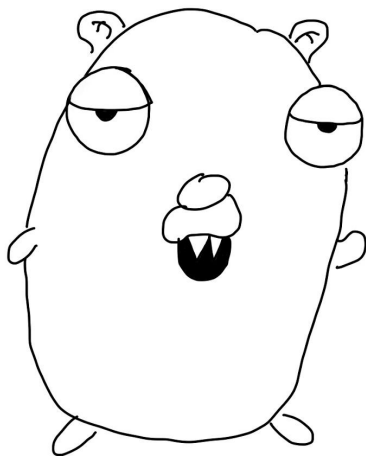
- **DO:** `kubectl open-svc`

The plugin will open a service :-)



PLUGIN CREATION > BEST PRACTICES

Golang

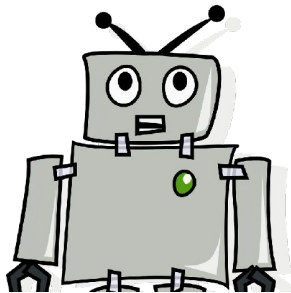
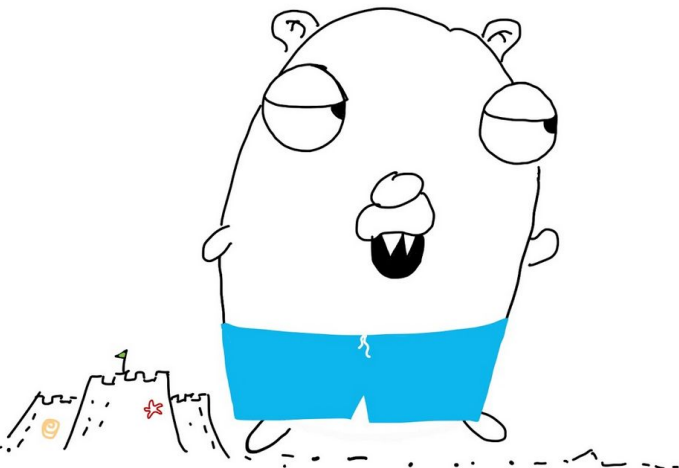


<https://github.com/kubernetes/sample-cli-plugin>

PLUGIN CREATION > BEST PRACTICES

Golang

krew-release-bot



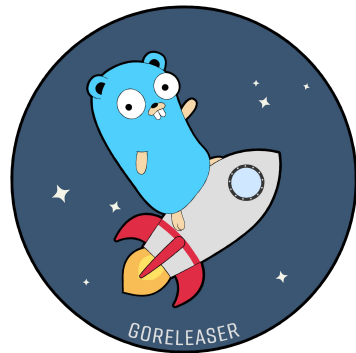
<https://github.com/rajatjindal/krew-release-bot>

PLUGIN CREATION > BEST PRACTICES

Golang

knew-release-bot

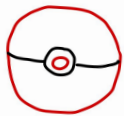
Go releaser



<https://goreleaser.com/>



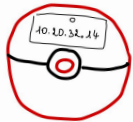
Pod



Smallest unit
deployable & runnable
in a cluster



Can contain
several containers



One IP address
per Pod



Debugging / Troubleshooting
Istio

404

503

When errors occurs, debugging Istio
can be very tricky

1 Check services's ports name

Istio can route correctly the traffic if it knows the protocol.
Service's port name must match with protocol (suffix)



http

grpc

http2-mysql

http2

mongo

redis

tcp

udp



web

mysql

http-mysql

app-http

...

spec:

ports:

- port: 5555

targetPort: 5555

name: grpc-app



Alpha

Since Kubernetes 1.18, you can use new appProtocol field
in EndpointPort or ServicePort API instead of ServicePort's name



Docker images

→ Images are identified by a name & a tag

repository : tag



Imagine images like a cake with layers



Layer 5a1bfe8

Layer 8b2ac42

Layer 9c41bd7

Layer 4b1c1d7f

Layer 7c13f4c12

→ An image consists of everything needed to run an
application : code, binaries, tools, runtimes, dependencies...

→ Layers can be reused by images

<http://gumroad.com/aurelievache> & <http://dev.to/aurelievache>

UNDERSTANDING IN A VISUAL WAY

 **Aurelie Vache**
434 abonnés

ACCUEIL VIDÉOS PLAYLISTS CHAÎNES DISCUSSION À PROPOS

Vidéos en ligne ► TOUT REGARDER

Understanding Kubernetes in a visual way - 12 - ...
151 vues • il y a 1 semaine
Sous-titres

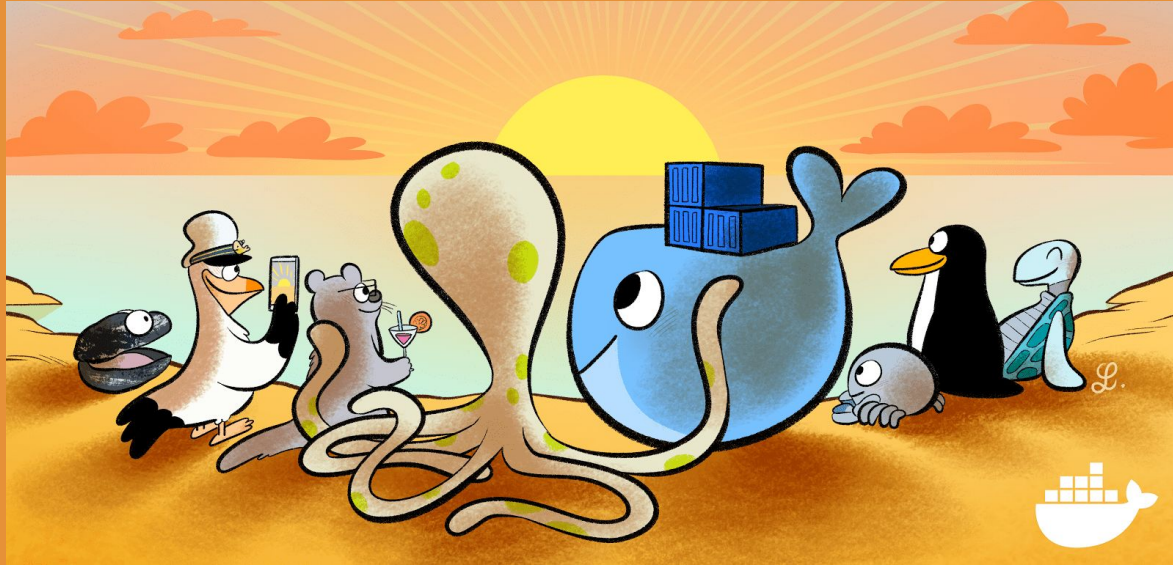
Understanding Kubernetes in a visual way - 11 - ...
191 vues • il y a 2 semaines
Sous-titres

Understanding Kubernetes in a visual way - 10 - Kubec...
170 vues • il y a 3 semaines
Sous-titres

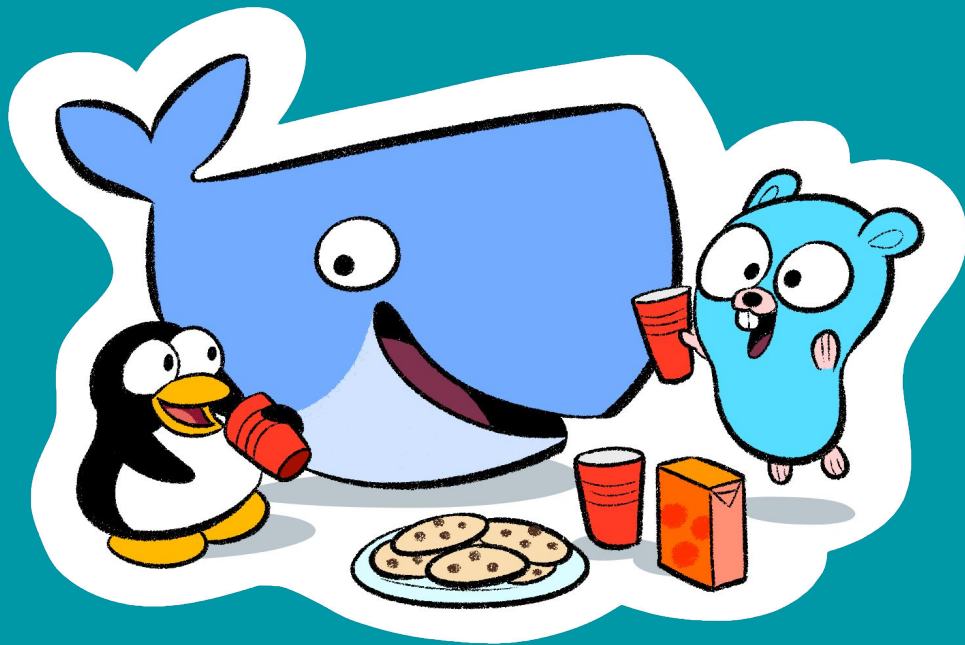
Understanding Kubernetes in a visual way - 09 - La...
156 vues • il y a 1 mois
Sous-titres

<https://www.youtube.com/c/AurelieVache>

UNDERSTANDING IN A VISUAL WAY 



CONCLUSION



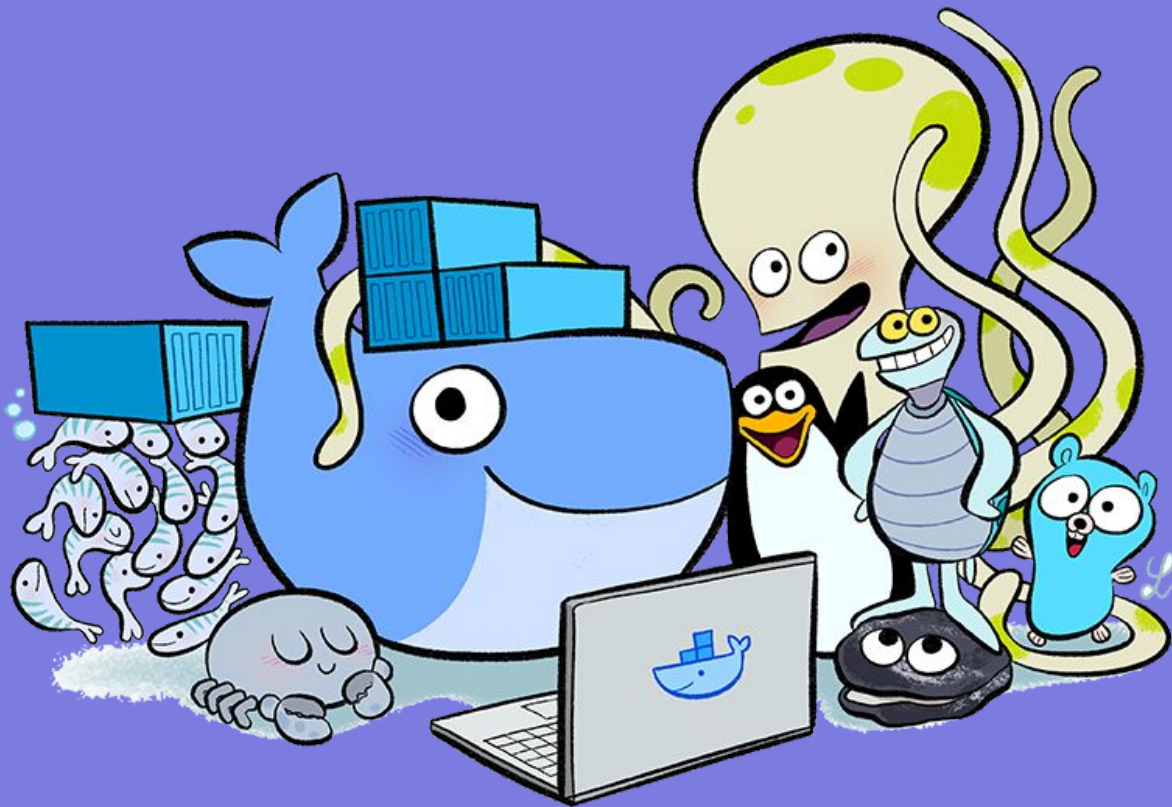
<http://bit.ly/kubernetes-plugin-devnation>

\$ KUBECTL THANK-YOU



<http://bit.ly/vote-devnation>

@aurelievache | @Gaelleacas



\$ KUBECTL APPLY -F QUESTIONS.YAML

