



KubeCon



CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

“Why Can’t Kubernetes Devs Just Add This New Feature? Seems So Easy!”

Understanding the Feature Lifecycle in Kubernetes

Carlos Tadeu Panato Jr - Chainguard
Ricardo Pchevuzinske Katz - VMware

Understanding the Feature Lifecycle in Kubernetes



BUILDING FOR THE ROAD AHEAD

DETROIT 2022



KubeCon



CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021



Carlos Panato
@comedordexis
Software Engineer
Chainguard

Ricardo Katz
@rp Katz
Software Engineer
VMware

Do you know how much time it takes for a feature to be implemented in Kubernetes?

An easier way to rollout ConfigMap

Facilitate ConfigMap rollouts / management #22368

Open

bgrant0607 opened this issue on Mar 2, 2016 · 220 comments



bgrant0607 commented on Mar 2, 2016 · edited

Member



To do a rolling update of a ConfigMap, the user needs to create a new ConfigMap, update a Deployment to refer to it, and delete the old ConfigMap once no pods are using it. This is similar to the orchestration Deployment does for ReplicaSets.

One solution could be to add a ConfigMap template to Deployment and do the management there.

Another could be to support garbage collection of unused ConfigMaps, which is the hard part. That would be useful for Secrets and maybe other objects, also.

cc @kubernetes/sig-apps-feature-requests



Or some "simple" startup ordering

Support startup dependencies between containers on the same Pod

Open

nmittler opened this issue on Jun 26, 2018 · 105 comments



nmittler commented on Jun 26, 2018 · edited

Tip

/kind feature
/sig node

This is a proposed kubernetes feature to address an issue we're seeing on Istio. I'm definitely open for alternative approaches, but I feel that this may be a generally useful feature for Kubernetes.

Background:

Istio pods have a sidecar container (called "istio-proxy") which runs an Envoy proxy. All traffic in and out of the pod is redirected (via iptables) through Envoy. Kubernetes starts both the application and istio-proxy at the same time, and as soon as the application shows "healthy", it can start receiving traffic. However, if Envoy has not yet received its initial configuration from Pilot (the configuration service), the service will experience temporary traffic loss until Envoy is configured.

We're currently recommending that developers solve this problem themselves by running a startup script on their application container which delays application startup until Envoy has received its initial configuration. However, this is a bit of a hack and requires changes to every one of the developer's containers.

Proposal:

In my mind, this situation would be better if we could make the startup of one container on the Pod dependent on a healthy status of another container. This way, we can provide an Envoy health check that confirms that it's received configuration from Pilot, thus triggering the startup of the application container automatically. No custom scripts, or manual steps at all.

331 7 5

Ass

No

Lak

ki

si

Pro

No

Mil

No

Dev

No

No

You

- ReadinessGate?
- Some new ordering field?
- A LOT OF DISCUSSION
- Some closed issues



DualStack? Everybody needs IPv6 this days!

DualStack: Add IPv4/IPv6 Dual-Stack support and awareness #6282

✓ Closed

leblancd opened this issue on Apr 18, 2018 · 29 comments



leblancd commented on Apr 18, 2018 · edited

😊 ⚠️ Tip ...

Is this a **BUG REPORT** or **FEATURE REQUEST**?:

Uncomment only one, leave it on its own line:

/kind bug

/kind feature

What happened:

For Kubernetes Release 1.10 or older, clusters can only be run in either IPv4-only, IPv6-only, or in a VERY limited dual-stack configuration. The current dual-stack support is limited by the following restrictions:

- Pods can have dual-stack addresses (if a CNI plugin is used that supports dual-stack, e.g. bridge or calico), but Kubernetes is only aware of 1 address per pod
- Nodes can have dual-stack addresses
- Kube system pods (api server, controller manager, etc.) can only have 1 address per pod (all pods have IPv4 or all pods have IPv6)
- Only 1 service CIDR can be configured for the cluster, so service IPs are all IPv4 or all IPv6
- Kube-proxy only supports 1 family of iptables at a time (choose iptables or ip6tables)
- Endpoints for services are all IPv4 or all IPv6
- Kube-dns is capable of running dual-stack, but it is currently only made aware of either all IPv4 or all IPv6 service addresses

What you expected to happen:

Ass



Lat



si

Pro

Noi

Mil

No

Dev

No

No

Yol



DualStack? Everybody needs IPv6 this days!

 **thockin** moved this from **Beta gated (merged) to GA (merged, gate not removed)** in SIG-Network KEPs on **Sep 24, 2021**



Priyankasaggu1... commented on Nov 8, 2021 • edited ▾

Member 😊 ...

Hello @lachie83 🙌

Checking in once more as we approach 1.23 code freeze at **6:00 pm PST on Tuesday, November 16.**

Please ensure the following items are completed:

- All PRs to the [Kubernetes repo](#) that are related to your enhancement are linked in the above issue description (for tracking purposes).
- All PRs are fully merged by the code freeze deadline.

As always, we are here to help should questions come up.

Thank you so much! 😊



 **bridgetkromhout** mentioned this issue on Nov 17, 2021

v1.23 blog post: Dual-stack IPv4/IPv6 Networking Reaches GA [kubernetes/website#30538](#)

 3 tasks

 Merged

It took "only"
3 years!

Also we do "suffer"

[Feature] - Support printing HTTP Liveness check return #61622

✓ Closed

rikatz opened this issue on Mar 23, 2018 · 5 comments



rikatz commented on Mar 23, 2018 · edited ▾

Member



Is this a BUG REPORT or FEATURE REQUEST?: Feature Request

/kind feature

What happened:

It would be great if Liveness Checks supports returning the HTTP response.

The case here is that we have a liveness check that returns not only a different RC when something else fails, but also what have failed (DB integration, some integration failure).

Having this in Describe POD would be great so operations team can troubleshoot what happened to a POD when it is restarting.

I've tested this, and describe only returns the HTTP code, but not the content.

This could be a flag in API, exposing if the content should or shouldn't be returned.

Thanks



There are a lot of more
examples!

But why so much time and
some features are refused?

The feature solves a
wide problem or is this
too niche?

Does the feature bring
(or solve) a security
concern?

Does the feature bring
(or solve) a
performance concern?

Is the feature a
breaking change?

Breaking changes in GA
are not allowed!

Was this feature
discussed before?

What are the conclusions
of previous discussions?

Meet the KEP!



- A Kubernetes Enhancement Proposal (KEP) is a way to propose, communicate and coordinate on new efforts for the Kubernetes project.
- This process is still in a beta state and is mandatory for all enhancements beginning release 1.14



Discussions everywhere!

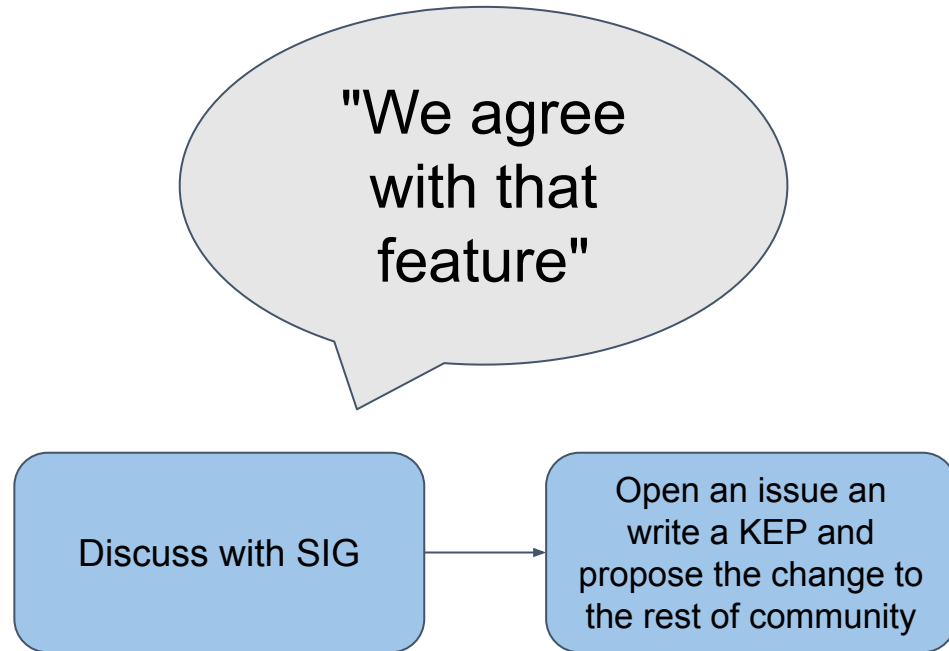
SIGs?

A group of
subject matter
in that part of
Kubernetes

Discuss with SIG

Example: If you want to add colors to kubectl, sig-cli is the right place for you!

Discussions everywhere!

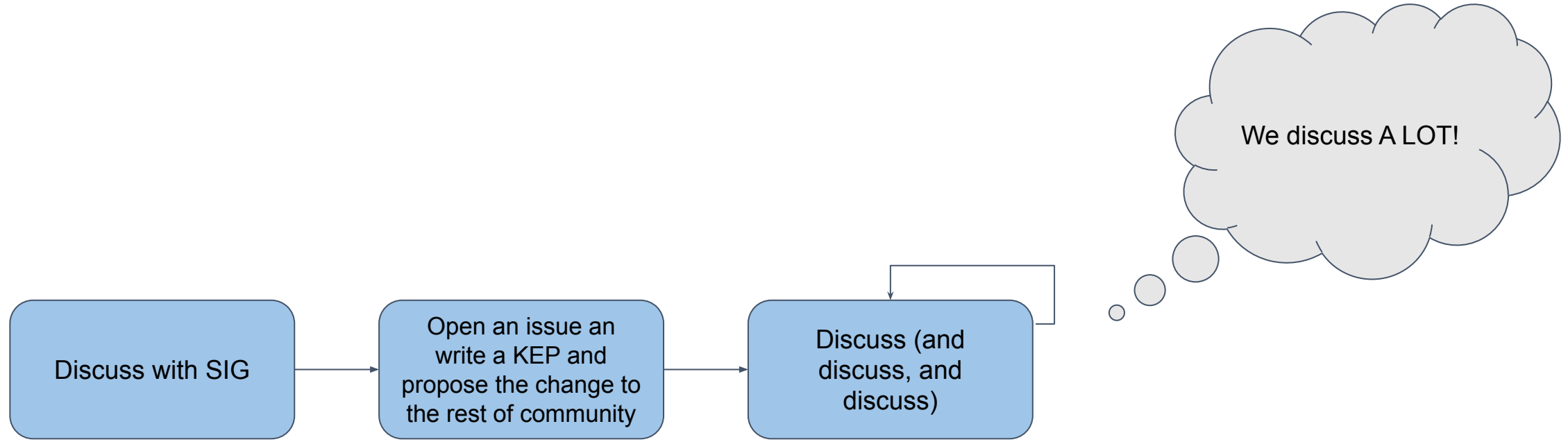


Is my thing an Enhancement?

- Needs a blog post after release?
- Requires more SIGs commenting on it?
- Can follow graduation (alpha -> beta -> GA) steps?
- Redesign something? Needs a big effort? (this is questionable!)
- Impacts User Experience in Kubernetes?
- People can complain / notice about it?



Discussions everywhere!



Discussions everywhere!

Add support for cluster-scoped AdminNetworkPolicy resource #25

 Merged k8s-ci-robot merged 39 commits into `kubernetes:master` from `abhiraut:cnp-kep`  on Feb 3

 Conversation 617  Commits 39  Checks 0  Files changed 8



abhiraut commented on Feb 18, 2021 • edited ▼

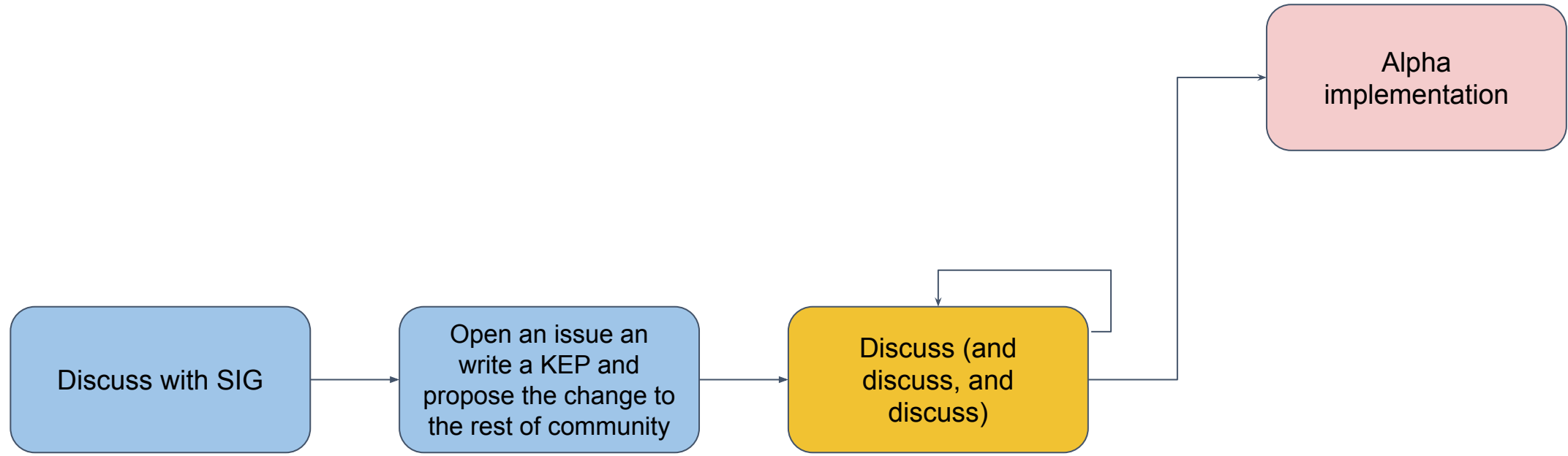
Member  ...

This KEP aims to introduce new resources to the `netpol.networking.k8s.io` group to secure traffic at a cluster level aimed at solving cluster administrator's use cases, i.e. Cluster scoped AdminNetworkPolicy for administrators.

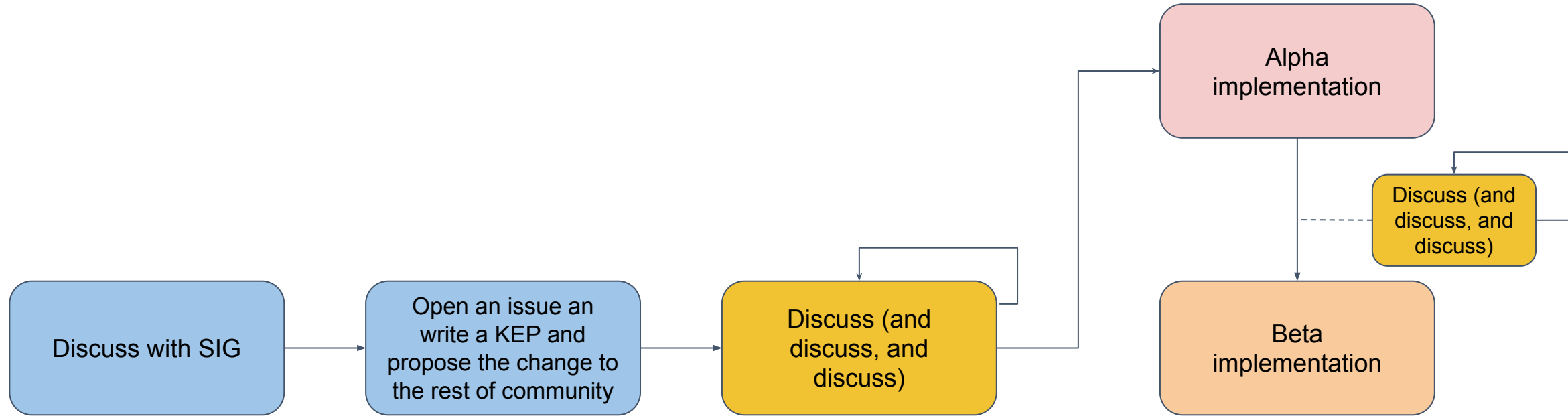
Enhancement Issue: [#2091](#)



Discussions everywhere!



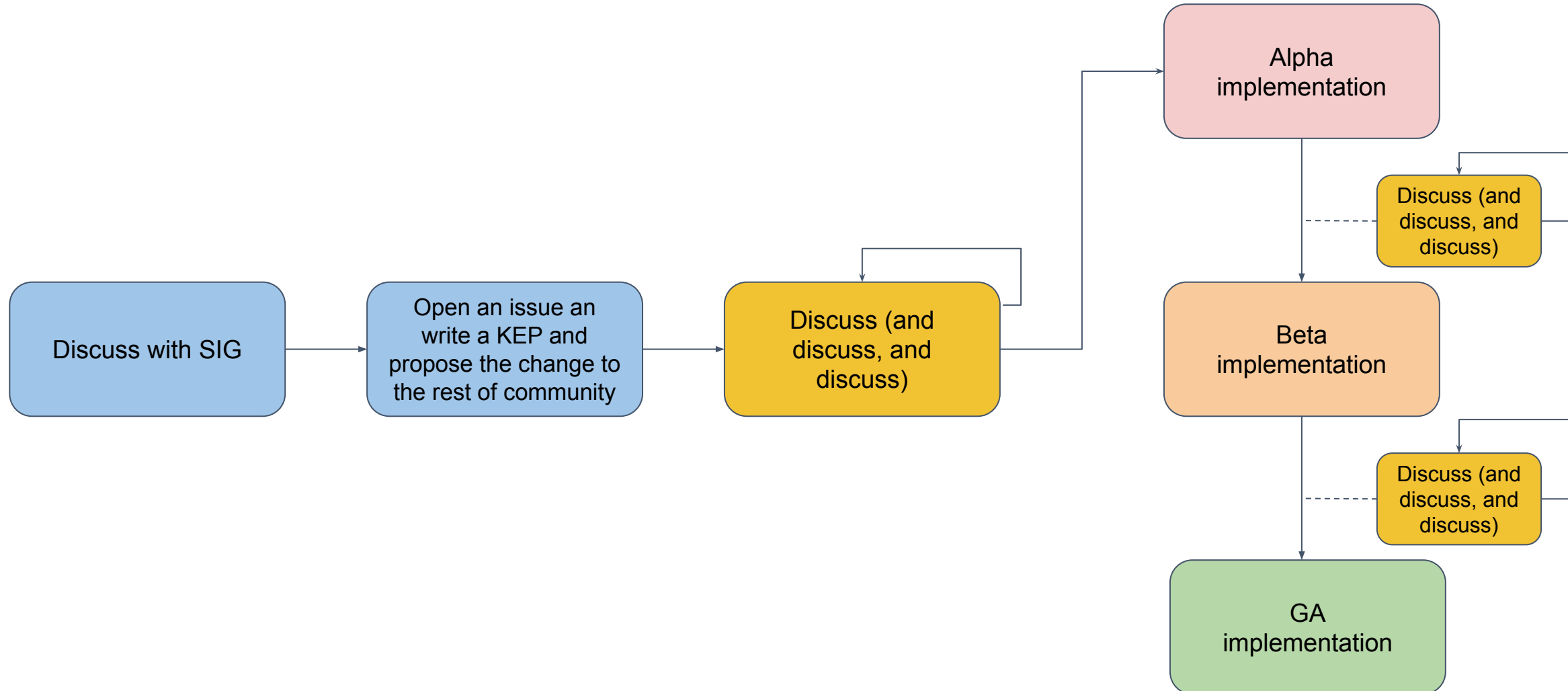
Discussions everywhere!



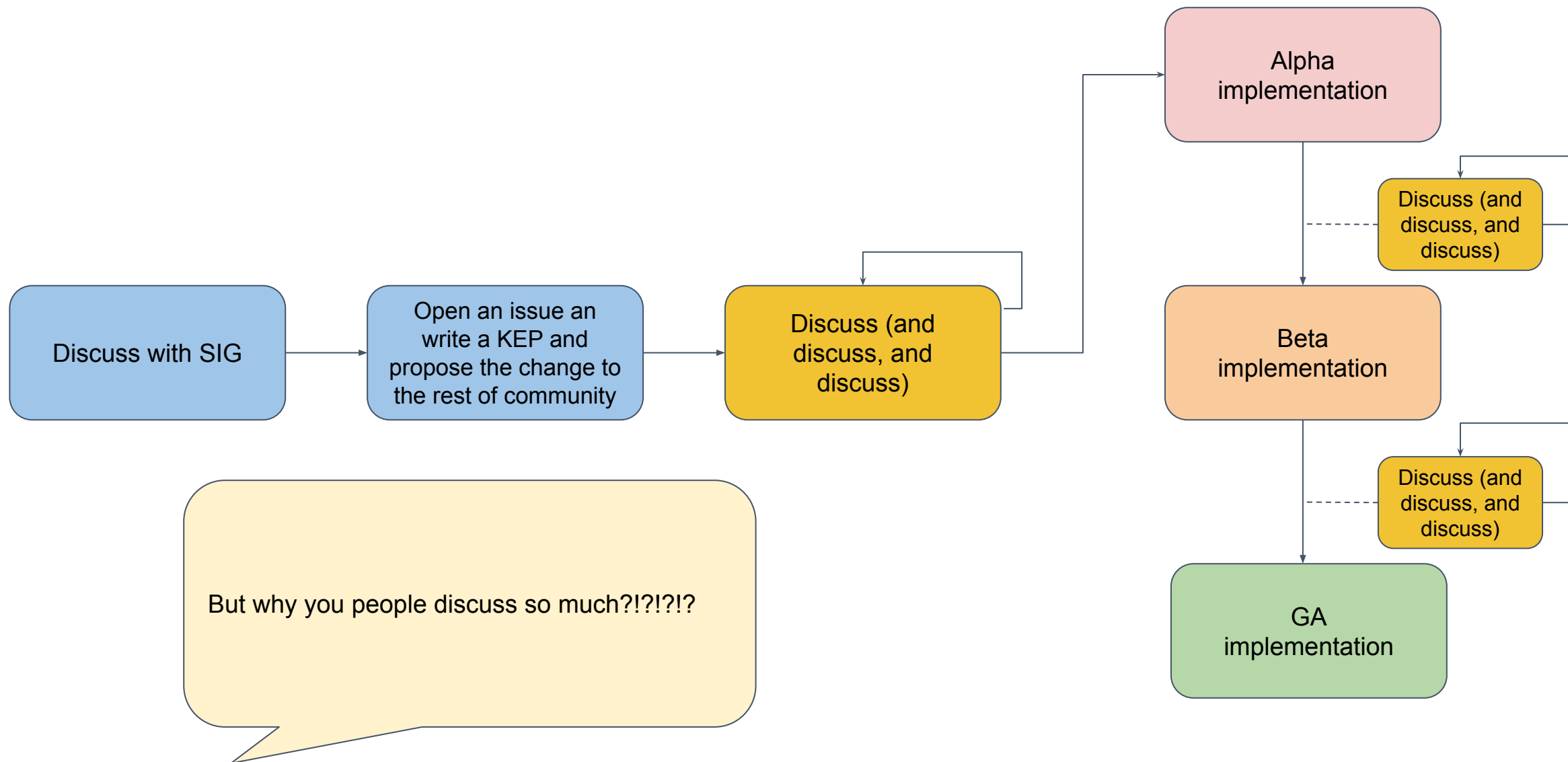


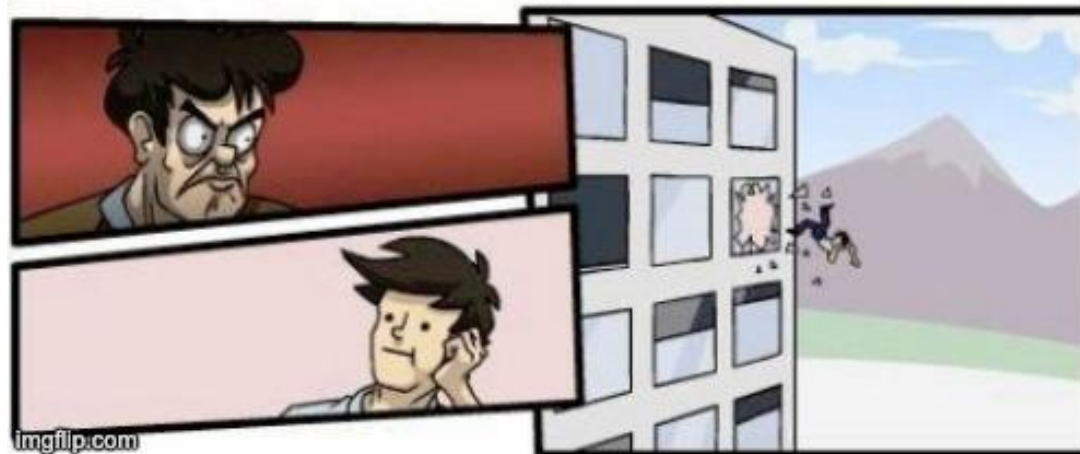
Before reaching GA, a feature can always be rejected and rolled back if some problem is identified!

Discussions everywhere!



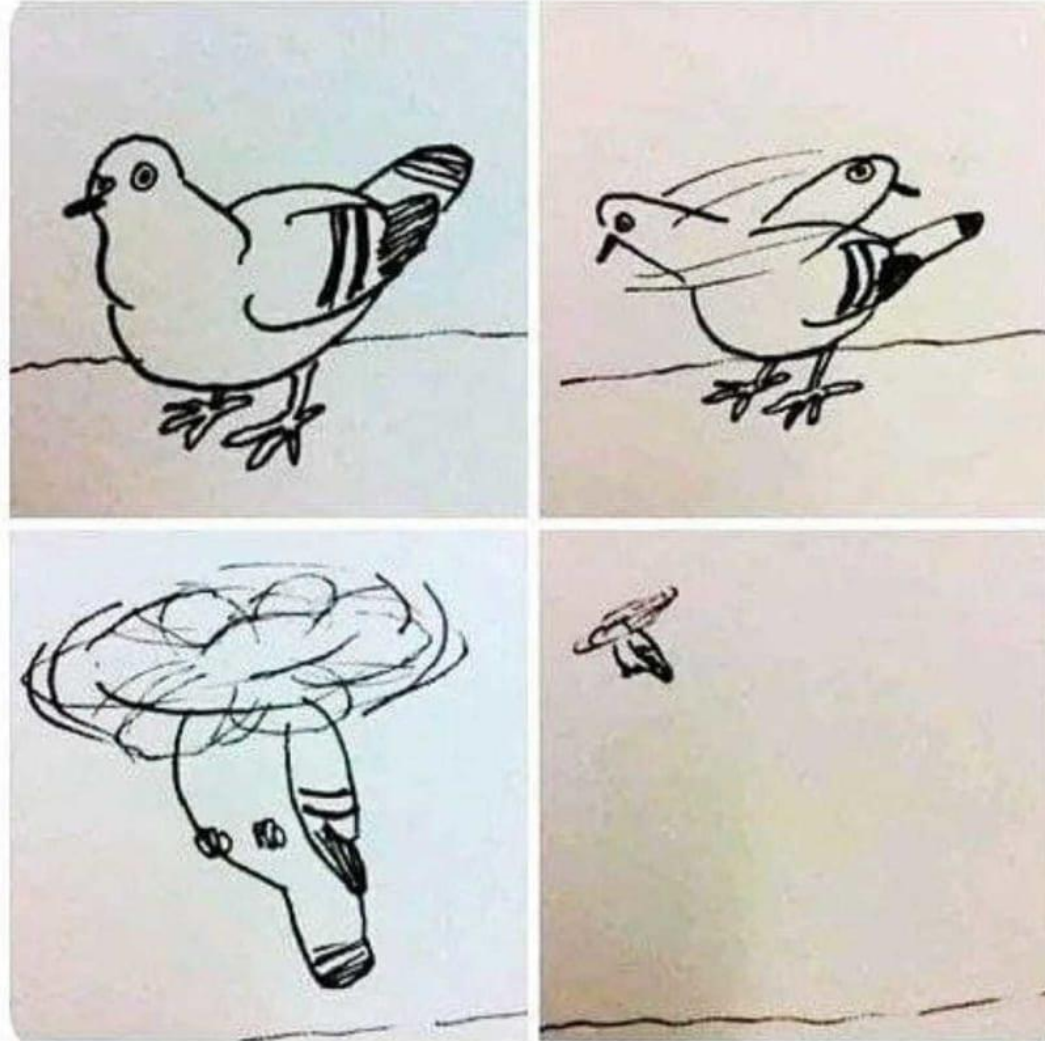
Discussions everywhere!





Some real life examples

When your program
is a complete mess,
but it does its job



When things can go wrong

Network Policy Port Range - "just a new field"

Problem: Can we add a new field in Network Policy that represents a range of ports instead of a single port?

When things can go wrong

Network Policy Port Range - "just a new field"

Problem: Can we add a new field in Network Policy that represents a range of ports instead of a single port?

```
spec:
  podSelector:
    matchLabels:
      role: someapp
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
      ports:
      - protocol: TCP
        portRange:
          from: 30000
          to: 32000
          except:
            - 31000
            - 31001
```

What happens if my CNI
doesn't know the new field?
Fail Open (all traffic allowed)!



```
spec:
  podSelector:
    matchLabels:
      role: someapp
  egress:
    - to:
      - ipBlock:
          cidr: 10.0.0.0/24
      ports:
        - protocol: TCP
```

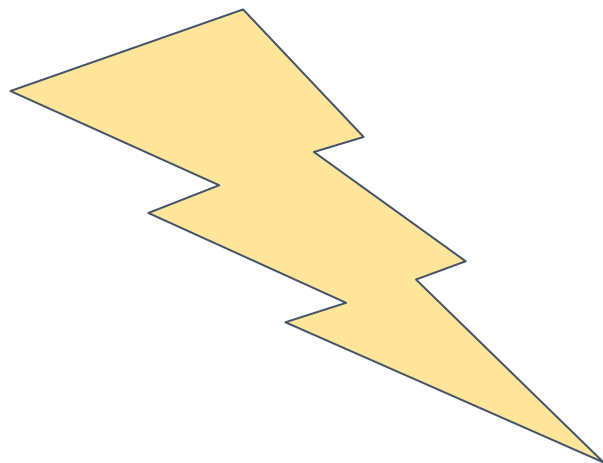
When things can go wrong

Network Policy Port Range - "just a new field"

Problem: Can we add a new field in Network Policy that represents a range of ports instead of a single port?

```
spec:
  podSelector:
    matchLabels:
      role: someapp
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
      ports:
        - protocol: TCP
          port: 30000
          endPoint: 32000
```

Dropped "exceptions" in ports. Too hard to implement!



What happens if my CNI doesn't know the new field now? Fail close (only a single port will be opened)

```
spec:
  podSelector:
    matchLabels:
      role: someapp
  egress:
    - to:
      - ipBlock:
          cidr: 10.0.0.0/24
      ports:
        - protocol: TCP
          port: 30000
```


Had a great idea in 2019



Still discussing the impacts in 2022



When things are stuck in discussions

Adding colors to kubectl (it says kube c-t-l!!) ;x
Since: 2018

Problem: Can we add colors to kubectl output? (describe, get pods, etc)



When things are stuck in discussions

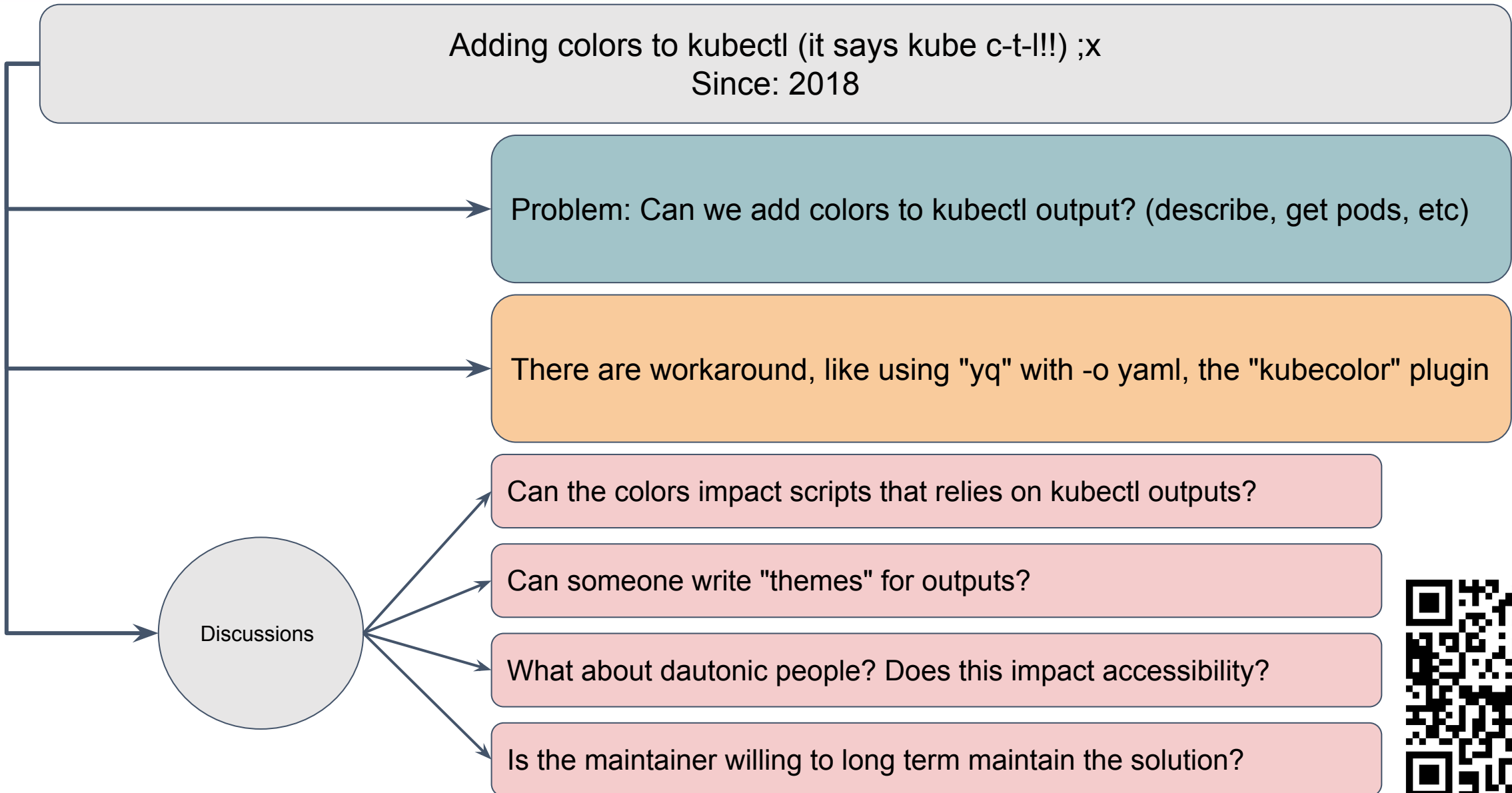
Adding colors to kubectl (it says kube c-t-l!!) ;x
Since: 2018

Problem: Can we add colors to kubectl output? (describe, get pods, etc)

There are workarounds, like using "yq" with -o yaml, the "kubecolor" plugin





When things are stuck in discussions



A new kuberc file (to configure kubectl behavior) is ongoing!

KEP-3104: Introduce kuberc #3392

 Merged k8s-ci-robot merged 1 commit into [kubernetes:master](#) from [eddiezane:ez/add-kuberc](#)  on Jun 23

 Conversation 66

 Commits 1

 Checks 0

 Files changed 3




eddiezane commented on Jun 13

Member



 Tip ...

- One-line PR description: Add initial KEP.
- Issue link:  [Separate kubectl user preferences from cluster configs #3104](#)
- Other comments:

/cc @liggitt @soltys



5



COULD NEVER IMAGINE THAT



THIS CRAZY IDEA WOULD WORK

When things takes time but goes right

Ephemeral containers
Since: 2017

Problem: I need a way to debug my Pod that runs "distroless" images

When things takes time but goes right

Ephemeral containers
Since: 2017

Problem: I need a way to debug my Pod that runs "distroless" images

This is a really hard feature! Needs a lot of machinery under the hood
(CRI, Process namespaces, network namespaces, etc)

MERGED AS GA!

When things takes time but goes right

Ephemeral Containers #277



20 of 23 tasks

verb opened this issue on Apr 25, 2017 · 169 comments



verb commented on Apr 25, 2017 · edited ▾

Member



Tip ...

Feature Description

- One-line feature description (can be used as a release note): Support advanced troubleshooting of running pods by running a new container image in shared pod namespaces.
- Kubernetes Enhancement Proposal: <https://git.k8s.io/enhancements/keps/sig-node/277-ephemeral-containers>
- Primary contact (assignee): @verb
- Responsible SIGs: sig-node
- Feature target (which target equals to which milestone):
 - Alpha release target (1.16)
 - Beta release target (1.23)
 - Stable release target (1.25)

Documentation

- [Ephemeral Containers Overview](#)
- [Creating Ephemeral Containers using kubectl](#)





setenforce 0

systemctl disable firewalld

When things goes wrong

Pod Security Policies - Because KUBERNETES IS NOT SECURE BY DEFAULT!

Problem: I need to establish policies that blocks my Pods to run with insecure configurations (privileged, hostNetwork, etc!)

When things goes wrong

Pod Security Policies - Because KUBERNETES IS NOT SECURE BY DEFAULT!

Problem: I need to establish policies that blocks my Pods to run with insecure configurations (privileged, hostNetwork, etc!)

PSP is too hard to configure in a workload

It depends of a bunch of knowledge to work

The namespace owner can still configure a Pod that binds to a Service Account that uses privileged PSP

It ends up being easier to "bypass" it, a la "SELinux 'setenforce 0'"

When things goes wrong

Pod Security Policies - Because KUBERNETES IS NOT SECURE BY DEFAULT!

Problem: I need to establish policies that blocks my Pods to run with insecure configurations (privileged, hostNetwork, etc!)

PodSecurityPolicy is deprecated now! Use PodSecurityAdmission instead!

PSA is configured on a namespace basis (a namespace label) so it applies to all the workloads on that Namespace

"Hardcoded" baselines (they are not configurable!)



Conclusions

Features are hard. We discuss (and need a lot of discussion) in Kubernetes code

If something breaks, people are gonna be mad at us! So we need to be extra careful when changing something.

Sometimes we just don't have people willing to implement and follow the whole process (yeah, it is tiring, but it pays its price)!

Takeaways

Don't give up! Kubernetes needs ideas!!! The execution sometimes is hard, but it pays its price!!

Take a look into past enhancements! Explore the features! Track new ones that are in Alpha! We need feedbacks!

Don't be shy bringing new ideas! There is no right or wrong! If you think it is worth to implement it, raise the issue, bring to a SIG discussion, understand the reasoning! We are all nice people!

Not only of code Kubernetes is made! You can propose improvements in Docs, feedback in your experience as a user and/or contributor and so on! Reach us in Slack!



Please scan the QR Code above to
leave feedback on this session