kubernetes

Understanding Kuberentes Release Cycle and Team Dynamics

by
Satyam Soni

Kubernetes: *Back to Basics*

Who I am?

- Satyam Soni
- Kuberenetes Release Team Member V1.30
- Author of Last Week in Kubernetes Development
- Cloud Native New Delhi Organizer
- DoKC Ambassador

Let’s Begin

Agenda

- Kubernetes Release
- Kubernetes Release Cycle
- Kuberentes Release team
- How to apply for Kubernetes release team

Kubernetes Release

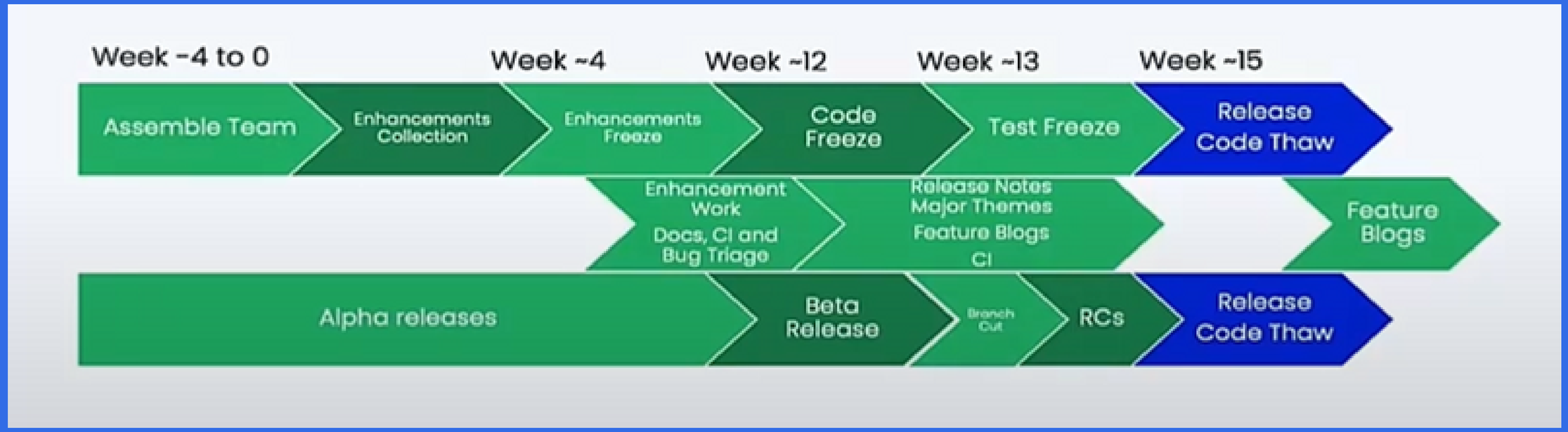
- A new version or update of the Kubernetes software that is made available to users.

- Each release undergoes thorough improvements , deprecations, testing and validation before being made publicly available to ensure reliability and compatibility with existing Kubernetes deployments.

- And there requires a dedicated team for Kubernetes Release and involved in the release cycle.

Release Cycle

- Involves assembling the team,making improvements, bug fixes, adding new features, tests and final release
- There are 3 release cycles in a year of 15 weeks each.



Enhancements Collection

Week 0-7

- Enhancements are ideas for feature requests.
- This is also Known as **KEPs(Kubernetes Enhancements Proposal)**
- Each SIG comes up with the ideas and features they want to be published or deprecated with the next release.
- All the ideas are placed in the enhancements tracking board with the alpha beta and graduated criteria.
- Collection of Major themes also gets started by Release Notes Team
- Code work also goes in parallel.

Prevent unauthorised volume mode conversion during volume restore

Open | 10 tasks done | **RaunakShah** opened this issue on Jan 14, 2022 · 53 comments

RaunakShah commented on Jan 14, 2022 • edited | Member

Enhancement Description

- One-line enhancement description (can be used as a release note): Prevent unauthorised volume mode conversion during volume restore.
- Kubernetes Enhancement Proposal: https://github.com/kubernetes/enhancements/tree/master/keps/sig-storage/3141-prevent-volume-mode-conversion
- Primary contact (assignee): **@RaunakShah** **@xing-yang**
- Responsible SIGs: SIG-Storage
- Enhancement target (which target equals to which milestone):
  - Alpha release target (x.y): 1.24
  - Beta release target (x.y): 1.27
  - Stable release target (x.y): 1.30

- [x] Alpha
  - [x] KEP (`k/enhancements`) update PR(s):
    - https://github.com/kubernetes/enhancements/tree/master/keps/sig-storage/3141-prevent-volume-mode-conversion
  - [x] Code (`k/k`) update PR(s):
    - **Update generated code for SourceVolumeMode field in VolumeSnapshotContents** kubernetes-csi/external-snapshotter#665
    - **Update snapshotter and client modules** kubernetes-csi/external-snapshotter#670
  - [x] Docs (`k/website`) update PR(s):
    - **Add docs for preventing volume mode conversion** website#32673
- [x] Beta
  - [x] KEP (`k/enhancements`) update PR(s):
    - **KEP-3141: Graduate "Prevent unauthorised volume mode conversion" to Beta** #3556
  - [x] Code (`k/k`) update PR(s):
    - **Add tests for volume mode conversion feature** kubernetes-csi/external-provisioner#867
    - **Add and enable sidecar e2e test to enable volume mode conversion** kubernetes-csi/external-provisioner#832
    - **Fix string pointer comparison for source volume mode conversion** kubernetes-csi/external-provisioner#793
    - **Register metrics exposed by sig-storage-lib** kubernetes-csi/external-provisioner#792
    - **Change annotation name to allow volume mode change** kubernetes-csi/external-provisioner#791
    - **Add dataSource label to pvcs created** kubernetes-sigs/sig-storage-lib-external-provisioner#128
    - **Add optional canary job to all CSI sidecar repos** test-infra#28307
  - [x] Docs (`k/website`) update(s):
    - **Update volume mode change annotation in documentation** website#39367
- [x] Stable
  - [x] KEP (`k/enhancements`) update PR(s): **KEP-3141: Graduate "Prevent unauthorised volume mode conversion" to Stable** #4459, **KEP-3141: Add no known failure modes to README** #4497

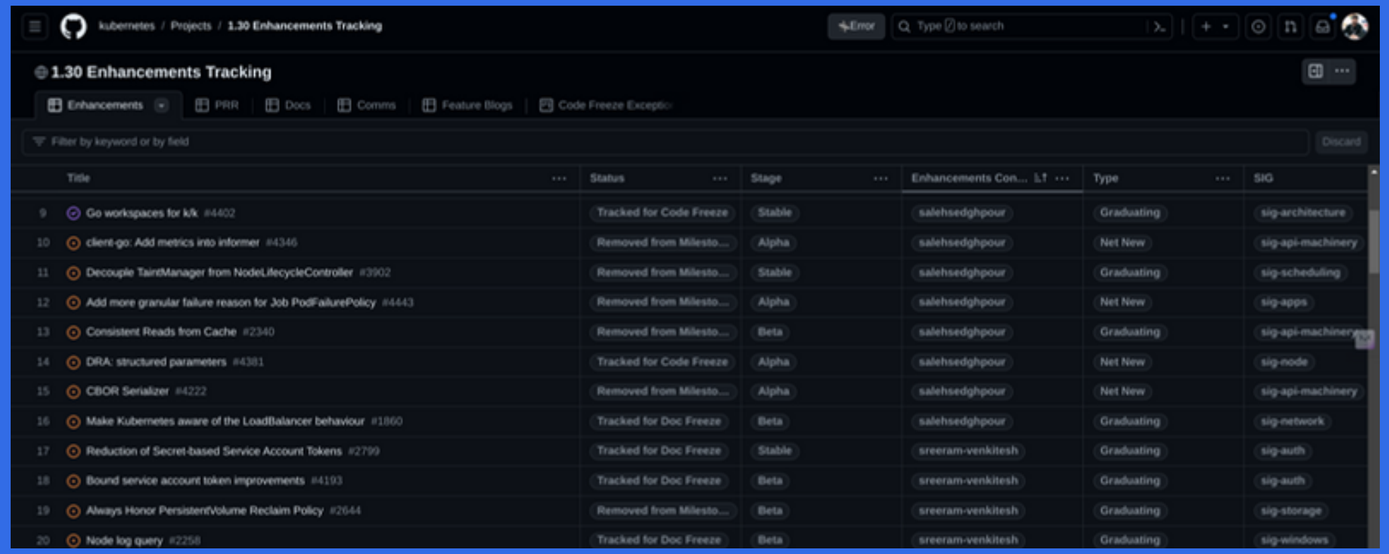
Assignees: RaunakS

Labels: lead-opted-, tracked/out-

Projects: (Test b, Status: No; 1.26 Cc, Status: No; 1.26 Er, Status: (; 1.27, Status: '; 1.30, Status: '

Mileston, v1.30

Enhancements Tracking Board



https://github.com/orgs/kubernetes/projects/175

Enhancements Freeze

Week 8

- This is the deadline by which the KEP has to be completed in order for enhancements to be the part of the current release.
- The enhancements should be in implementable states, have proper testing plans and have graduation criteria defined.
- There is also one more thing an Exception request which is the process of requesting an extension on the deadline for a particular Enhancement.

Code Freeze

Week 12

- This ensures that all the enhancements going into the release must be code complete and tests done.
- The PR should be in a merge-ready state with approved and LGTM labels applied.
- During this period only critical bug fixes are released into the release.
- Process of removing an Enhancement from a release milestone if it is not fully implemented or is otherwise considered unstable is pruning.
- We also have the Beta release available by this week.

Docs Deadline

week 9-12

- Most of the times for the deprecations and changes, it should be documented to help users learn about it.
- Publish the deprecations and removals blog by communications team.
- Most of the features are so big and major that they needs a feature blog seperately.

Test Freeze and Docs Freeze

week 13

- The day after no more test are changed.
- If any test finds to be failing either they are removed or reverted.
- Only in the case of release-blocking test it is fixed or updated.
- Enhancements that are determined to require documentation must have their docs PRs in a ready to merge.

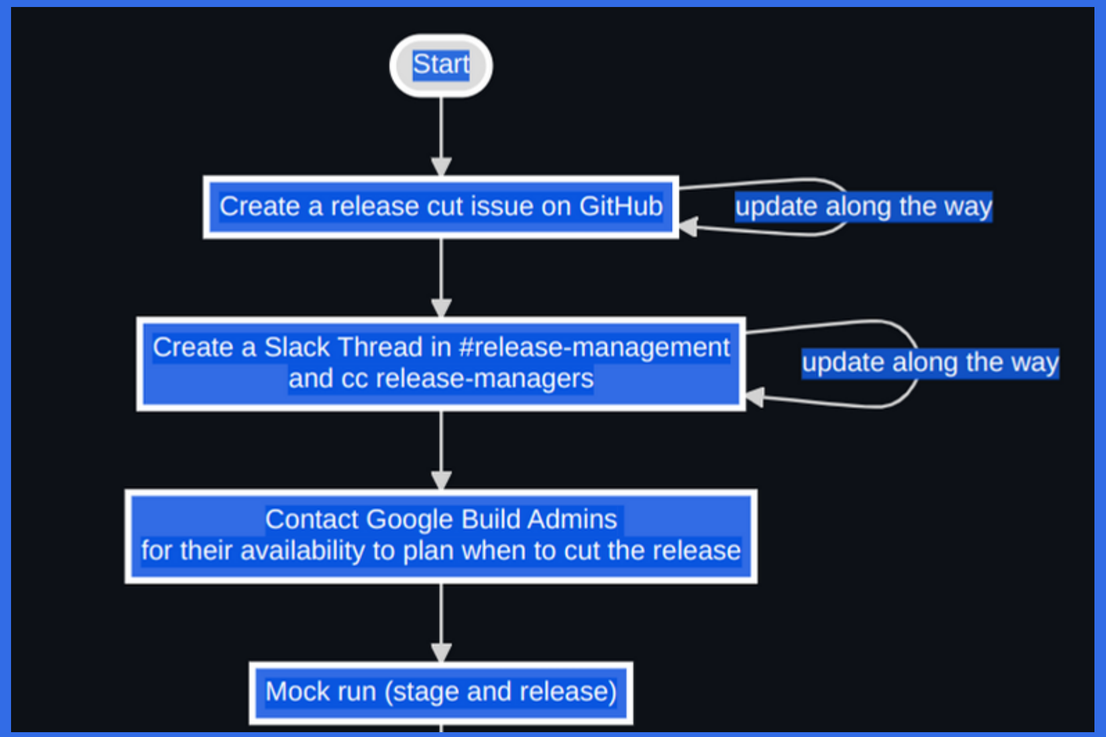
Thaw

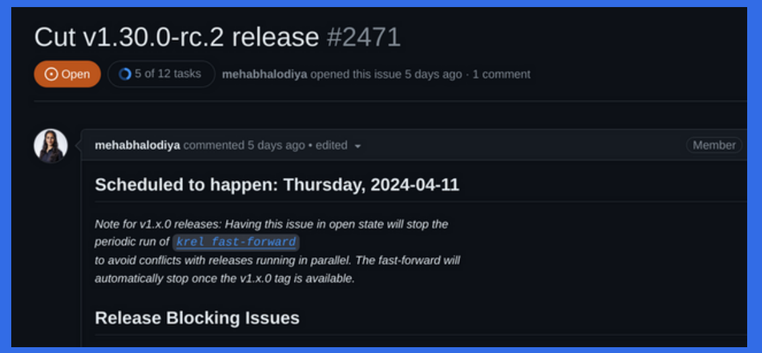
week 14

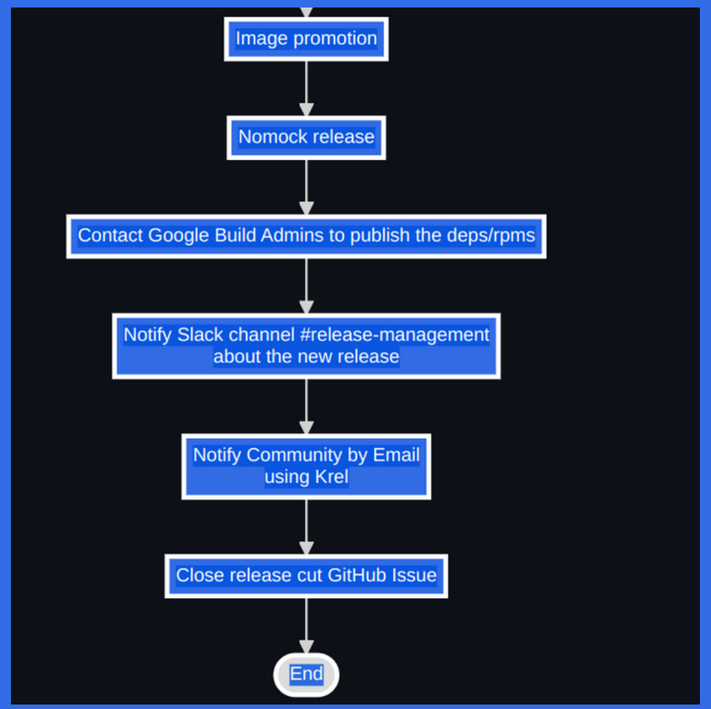
Thaw specifies the end of both Code Freeze and Test Freeze.

At this point it is expected that all outstanding PRs for the release of Kubernetes have been merged into the release branch.

Cutting the release (creating the final pkg) Week 15

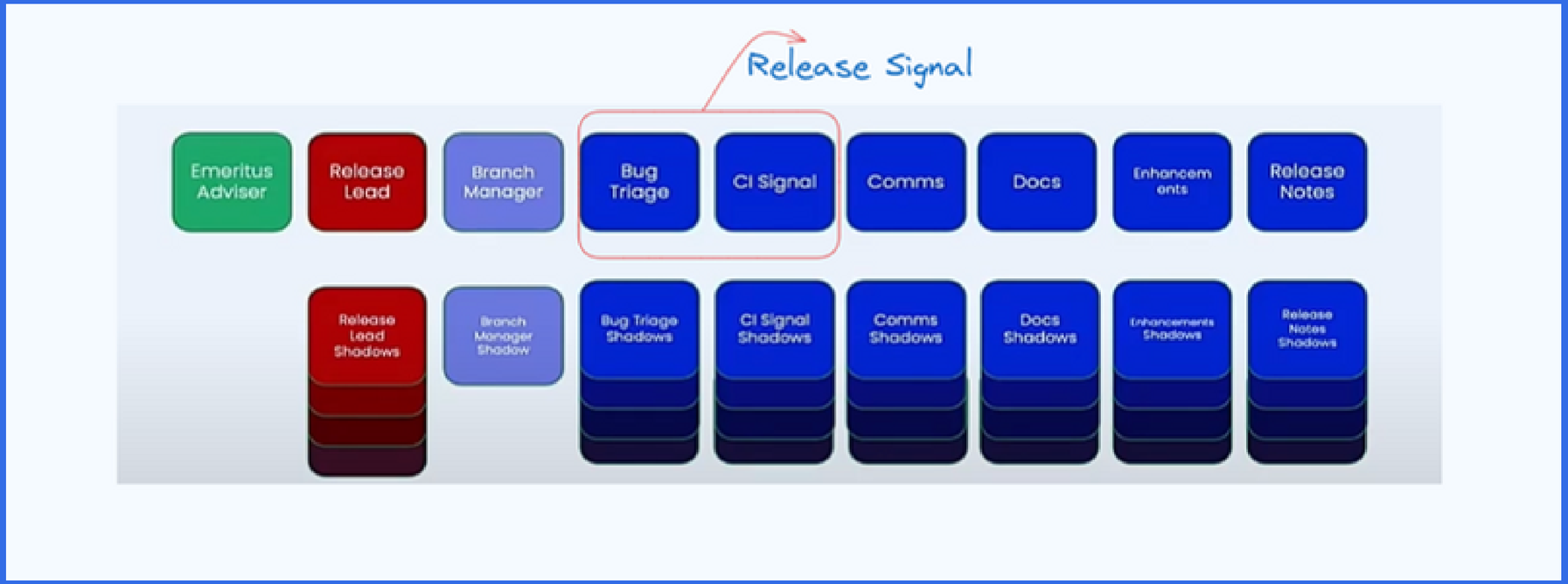






Release Team

- Release Team is responsible for succesful release of Kubernetes versions.
- There are around 30-35 people in the team across the globe.
- There are 8 roles.

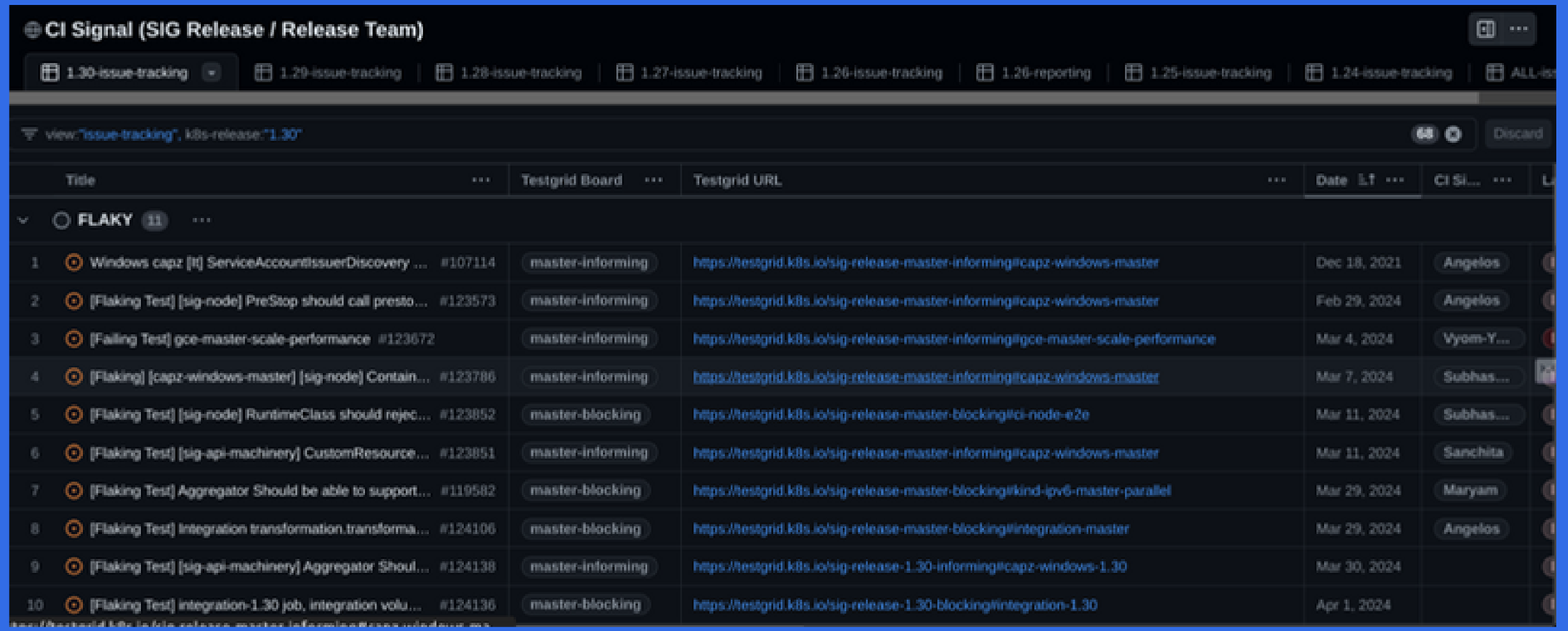


Enhancements

- Maintains the status of Kubernetes Enhancements Proposals or KEP(s)
- Reach out to different SIGs and ask them to finish their issue templates for the proposal.
- Responsible for tracking issues in the Kubernetes/Enhancements repository
- Track the progress of issue alpha to beta to graduated to stable.
- First 8 weeks are very hectic for the Enhancements team.

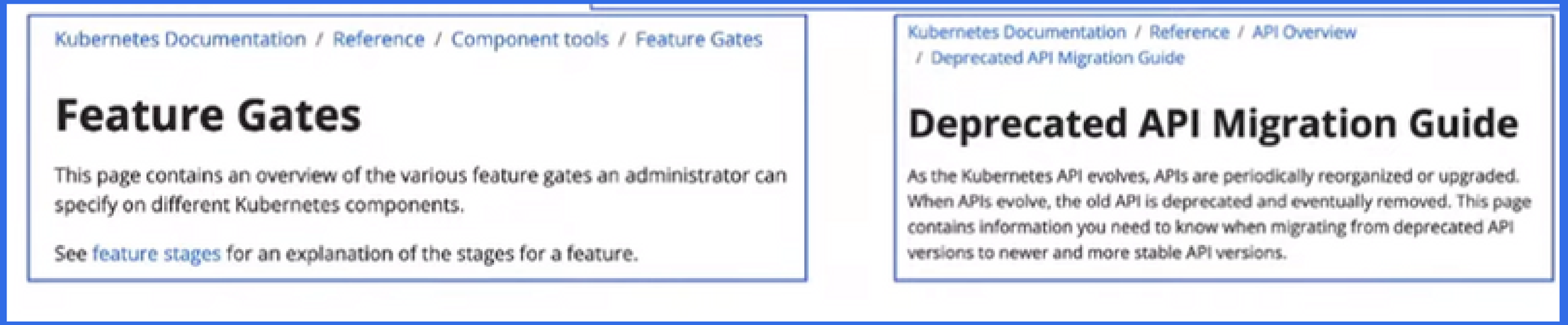
Release signal

- Manage Issues/PRs
- Monitor e2e tests
- Automate tracking
- Triage issues
- Track blockers



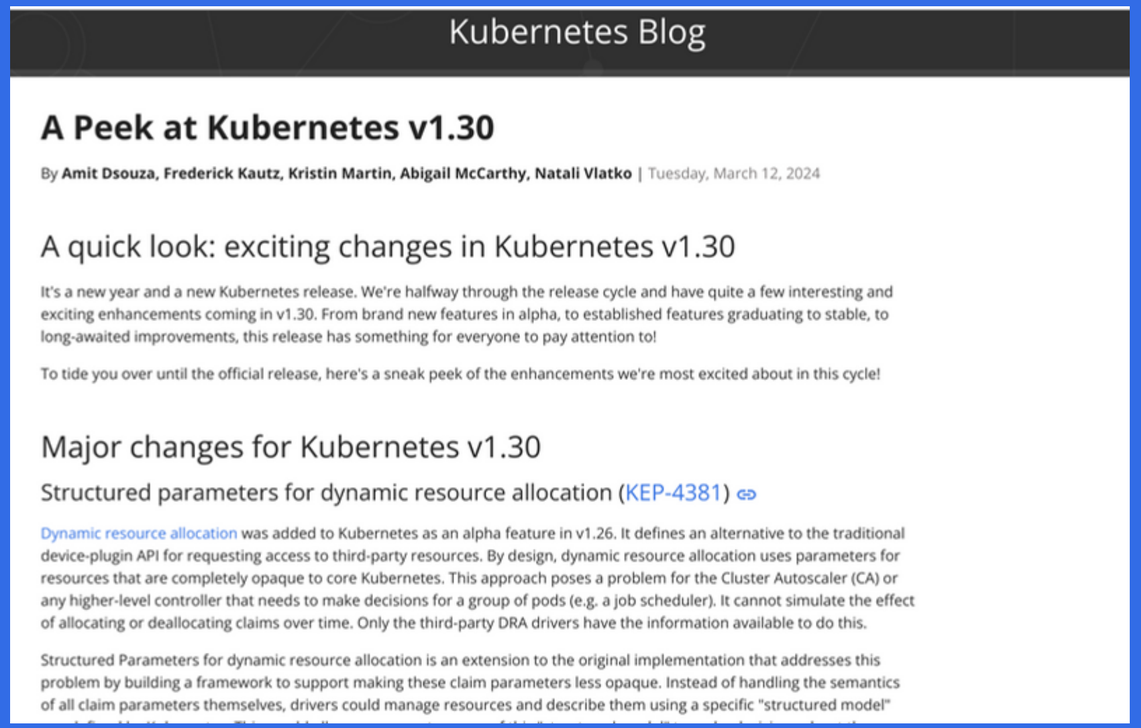
Documentation

- Responsible for documentation updates for the next Kubernetes release.
- Ensures new and graduating features have appropriate documentation.



Communications

- Responisble for communication in and outside the Kuberenetes.
- Authors the Kubernetes release blog.
- Coordinate and support of the feature blog series and optional deprecations blog.
- Schedules press activities and the post-release webinar



Kubernetes Blog

A Peek at Kubernetes v1.30

By **Amit Dsouza, Frederick Kautz, Kristin Martin, Abigail McCarthy, Natali Vlatko** | Tuesday, March 12, 2024

A quick look: exciting changes in Kubernetes v1.30

It's a new year and a new Kubernetes release. We're halfway through the release cycle and have quite a few interesting and exciting enhancements coming in v1.30. From brand new features in alpha, to established features graduating to stable, to long-awaited improvements, this release has something for everyone to pay attention to!

To tide you over until the official release, here's a sneak peek of the enhancements we're most excited about in this cycle!

Major changes for Kubernetes v1.30

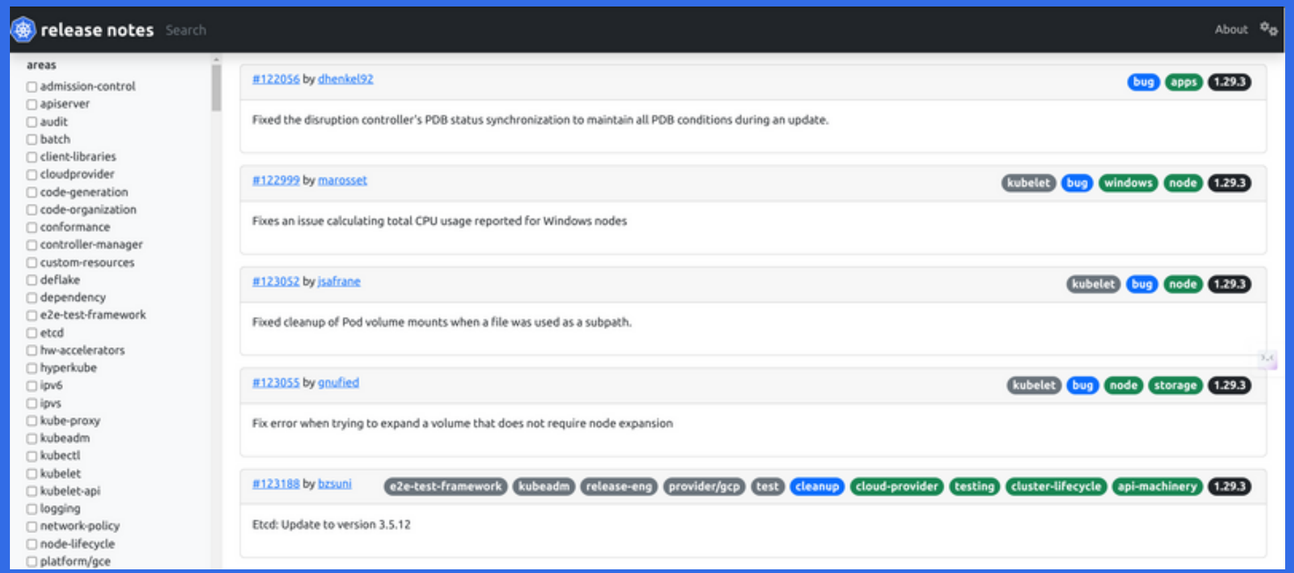
Structured parameters for dynamic resource allocation (KEP-4381)

Dynamic resource allocation was added to Kubernetes as an alpha feature in v1.26. It defines an alternative to the traditional device-plugin API for requesting access to third-party resources. By design, dynamic resource allocation uses parameters for resources that are completely opaque to core Kubernetes. This approach poses a problem for the Cluster Autoscaler (CA) or any higher-level controller that needs to make decisions for a group of pods (e.g. a job scheduler). It cannot simulate the effect of allocating or deallocating claims over time. Only the third-party DRA drivers have the information available to do this.

Structured Parameters for dynamic resource allocation is an extension to the original implementation that addresses this problem by building a framework to support making these claim parameters less opaque. Instead of handling the semantics of all claim parameters themselves, drivers could manage resources and describe them using a specific "structured model"

Release Notes

- Responisble for generating and editing release-notes.
- There is a tool Krel, which helps draft the PR for release Notes by the help of some subcommands.



Branch Management

- Responsible for cutting the release

Release Lead

- Coordinating release activities
- Assemble the release team
- Accountable for all release tasks to be completed on time
- Trains release lead shadows

Emeritus Advisor

- Lead retrospectives
- Track AIs
- Assist selection
- Shadow orientation
- Check leads
- Offer support
- Conduct surveys
- Facilitate retrospectives

How to apply?

Thank you

Scan the QR to connect with me