



**KubeCon**



**CloudNativeCon**

**Europe 2023**



# From Automation to Community: A Deep Dive Into SIG Contributor Experience

Priyanka Saggu, SUSE  
Madhav Jivrajani, VMware  
Kaslin Fields, Google



**KubeCon**



**CloudNativeCon**

Europe 2023




# Code of Conduct

Remember the **Golden Rule**: Treat others as you would want to be treated - with kindness and respect

Scan the QR code to access and review the **CNCF Code of Conduct**:



# Virtual Audience Closed Captioning

Closed captioning for the virtual audience is available during each session through **Wordly**.  The **Wordly** functionality can be found under the “Translations” tab on the session page.

**Wordly** will default to English. If another language is needed, simply click the dropdown at the bottom of the “Translations” tab and choose from one of 26+ languages available so you don’t miss a beat from our presenters.

\*Note: Closed captioning is ONLY available during the scheduled live sessions and will not be available for the recordings on-demand within the virtual conference platform.

# Who Are We?



Priyanka Saggu  
@\_psaggu



Madhav Jivrajani  
@MadhavJivrajani



Kaslin Fields  
@kaslinfields

# Before We Start...

**registry.k8s.io is GA!** 🎉

🚧 ❄️ k8s.gcr.io is frozen ❄️ 🚧

More info on  
<https://k8s.io/image-registry-redirect>

# How Does ContribEx... ContribEx?



# How Does ContribEx... ContribEx?

Subprojects!

# Subprojects Overview

- **Community** - Owns and manages overall community repo, including community group documentation and operations.

# Subprojects Overview

- **Community** - Owns and manages overall community repo, including community group documentation and operations.
- **Community Management** - Manages operations and policy for upstream community group communication platforms.

# Subprojects Overview

- **Community** - Owns and manages overall community repo, including community group documentation and operations.
- **Community Management** - Manages operations and policy for upstream community group communication platforms.
- **Contributor Documentation** - Writes and maintains documentation around contributing to Kubernetes, including the Contributor's Guide, Developer's Guide, and contributor website.

# Subprojects Overview

- **Community** - Owns and manages overall community repo, including community group documentation and operations.
- **Community Management** - Manages operations and policy for upstream community group communication platforms.
- **Contributor Documentation** - Writes and maintains documentation around contributing to Kubernetes, including the Contributor's Guide, Developer's Guide, and contributor website.
- **Devstats** - Maintains and updates <https://k8s.devstats.cncf.io>, including taking requests for new charts.

# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.

# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.
- **Events** - Creates and runs contributor-focused events, such as the Contributor Summit. Event Teams are part of this subproject.

# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.
- **Events** - Creates and runs contributor-focused events, such as the Contributor Summit. Event Teams are part of this subproject.
- **GitHub Management** - Manages and controls Github permissions, repos, and groups, including Org Membership.



# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.
- **Events** - Creates and runs contributor-focused events, such as the Contributor Summit. Event Teams are part of this subproject.
- **GitHub Management** - Manages and controls Github permissions, repos, and groups, including Org Membership.
- **Mentoring** - Oversees and develops programs for helping contributors ascend the contributor ladder, including our group mentoring cohorts.

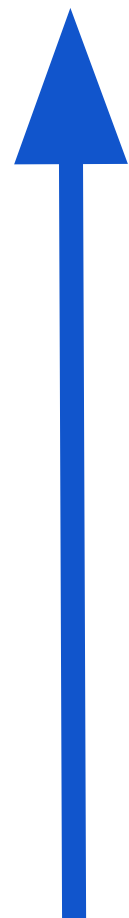
# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.
- **Events** - Creates and runs contributor-focused events, such as the Contributor Summit. Event Teams are part of this subproject.
- **GitHub Management** - Manages and controls Github permissions, repos, and groups, including Org Membership.
- **Mentoring** - Oversees and develops programs for helping contributors ascend the contributor ladder, including our group mentoring cohorts.
- **Slack Infra** - Creates and maintains tools and automation for Kubernetes Slack.

# Subprojects Overview

- **Elections** - New subproject! Oversees running elections in the community. Maintains documentation and software for elections.
- **Events** - Creates and runs contributor-focused events, such as the Contributor Summit. Event Teams are part of this subproject.
- **GitHub Management** - Manages and controls Github permissions, repos, and groups, including Org Membership.
- **Mentoring** - Oversees and develops programs for helping contributors ascend the contributor ladder, including our group mentoring cohorts.
- **Slack Infra** - Creates and maintains tools and automation for Kubernetes Slack.
- **Contributor Comms** - Amplifying the success of and distributing information to Kubernetes contributors.

# Mentoring Prereq - The Ladder



Subproject Owner

- Set priorities and approve proposals for subproject
- Responsibility and leadership for entire repository/directory

Approver

- Approve contributions for acceptance
- Highly experienced reviewer and contributor in subproject

Reviewer

- History of reviewing; reviews frequently
- Authorship in subproject

Member

- Active contributor to the project
- Sponsored by two Reviewers

Non-member Contributors

- Programs / Initiatives:
  - [Group Mentoring Cohorts](#) - Semi-structured group mentoring initiative with a small group of people
    - We need more SIGs/WGs to do this!
      - We are happy to help more groups kick this off, reach out to us on the #sig-contribex slack channel.
  - [Shadow Programs \(Building Teams\)](#) - Scalable apprenticeship program
    - The folks presenting here as shadows for SIG Contribex leadership positions!

- Few projects submitted for Google Summer of Code (GSoC)
- Others programs put on hold based on interest in the community:
  - Outreachy
  - Meet Our Contributors (merged with Office Hours)
- ***Need more approvers/subproject owners with bandwidth to be able to assist in mentoring efforts***

# GitHub Management: Update

## Changes In Lifecycle and Triage Flow of Issues

Auto-closing issues is harmful and causes friction #103151

New issue

✓ Closed

dims opened this issue on Jun 24, 2021 · 30 comments · Fixed by [kubernetes/test-infra#28266](#)



dims commented on Jun 24, 2021 · edited

Member

Previously open issue : [kubernetes/community#5473](#)

Twitter Discussion : <https://twitter.com/BenTheElder/status/1407774856033181696>

There's pros and cons to having a bot do this, let me try to enumerate current situation here:

- maintainers don't have to play bad cop
- keep set of open issues at a maintainable level
- bot commands are available to org members to `/reopen` reduces friction
- lifecycle labels help folks specify which issues not to touch
- we currently do not have dedicated team(s) that do issue triage well (except for some SIG like api-machinery that do this really well with twice a week triage session)
- initial issue labeling is still a problem, routing issues to the correct sig
- we have a new triage label which is not widely used yet across sigs

Assignees

No one assigned

Labels

**kind/bug** lifecycle/frozen needs-triage  
sig/contributor-experience

Projects

None yet

Milestone

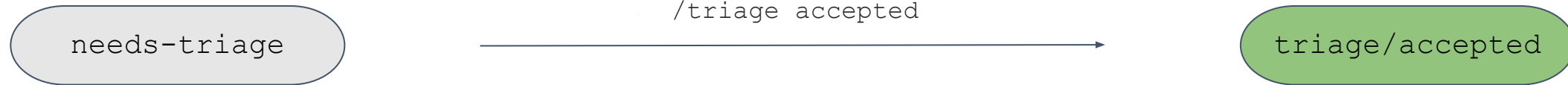
No milestone

Issue - <https://github.com/kubernetes/kubernetes/issues/103151>

Fix - <https://github.com/kubernetes/test-infra/pull/28266>

# GitHub Management: Update

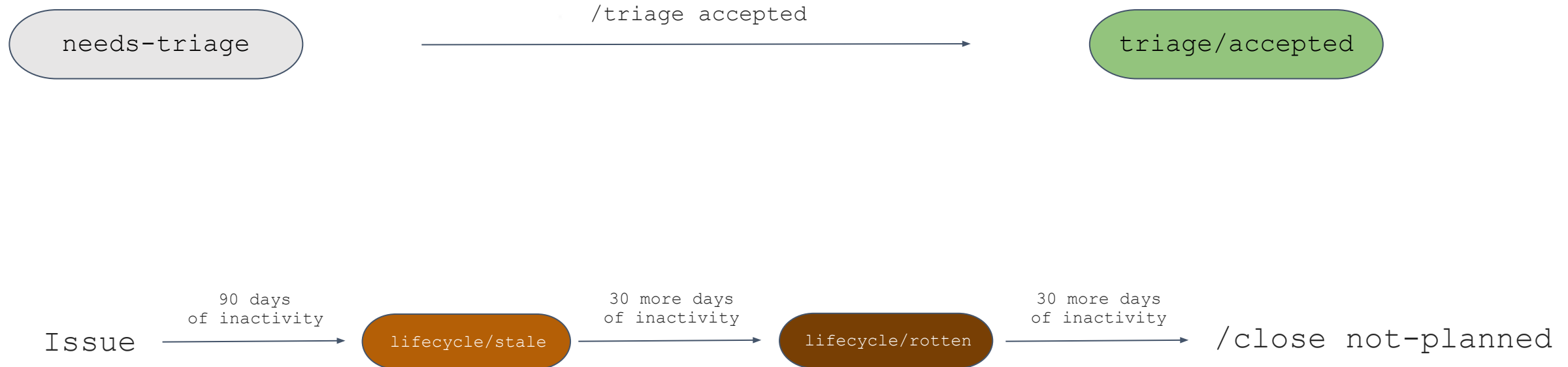
## Previous Issue Triage and Lifecycle Flow





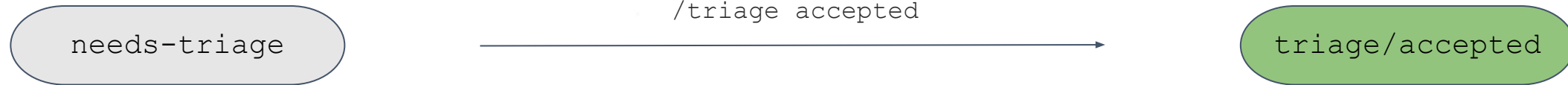
# GitHub Management: Update

## Previous Issue Triage and Lifecycle Flow



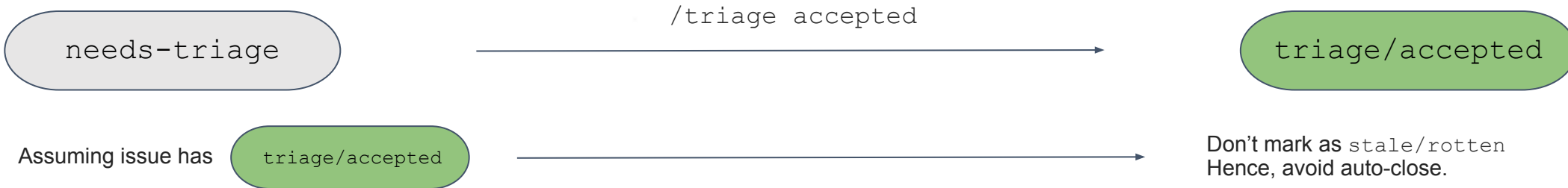
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



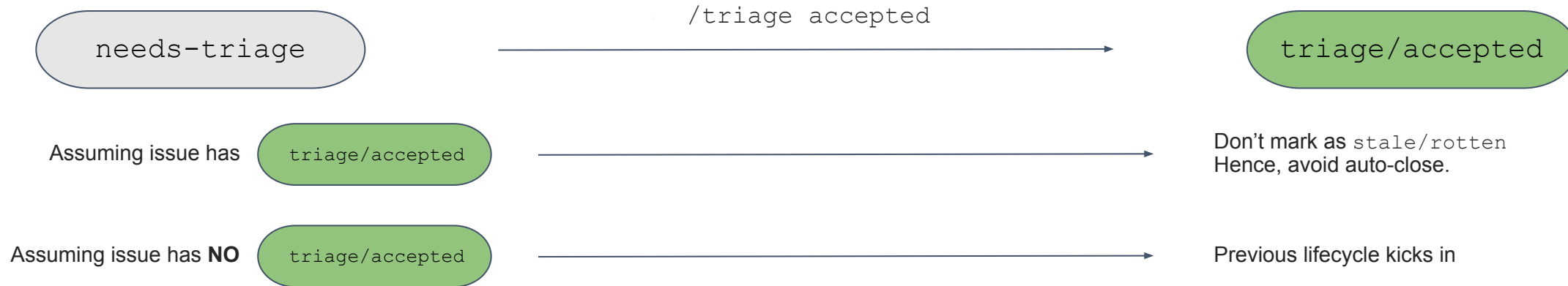
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



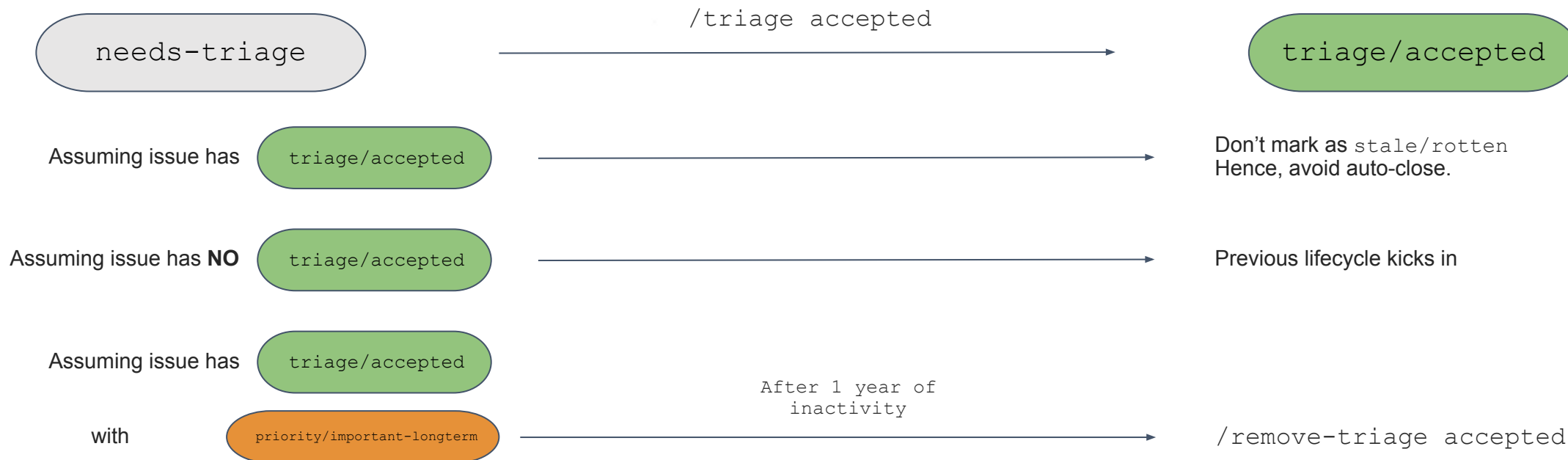
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



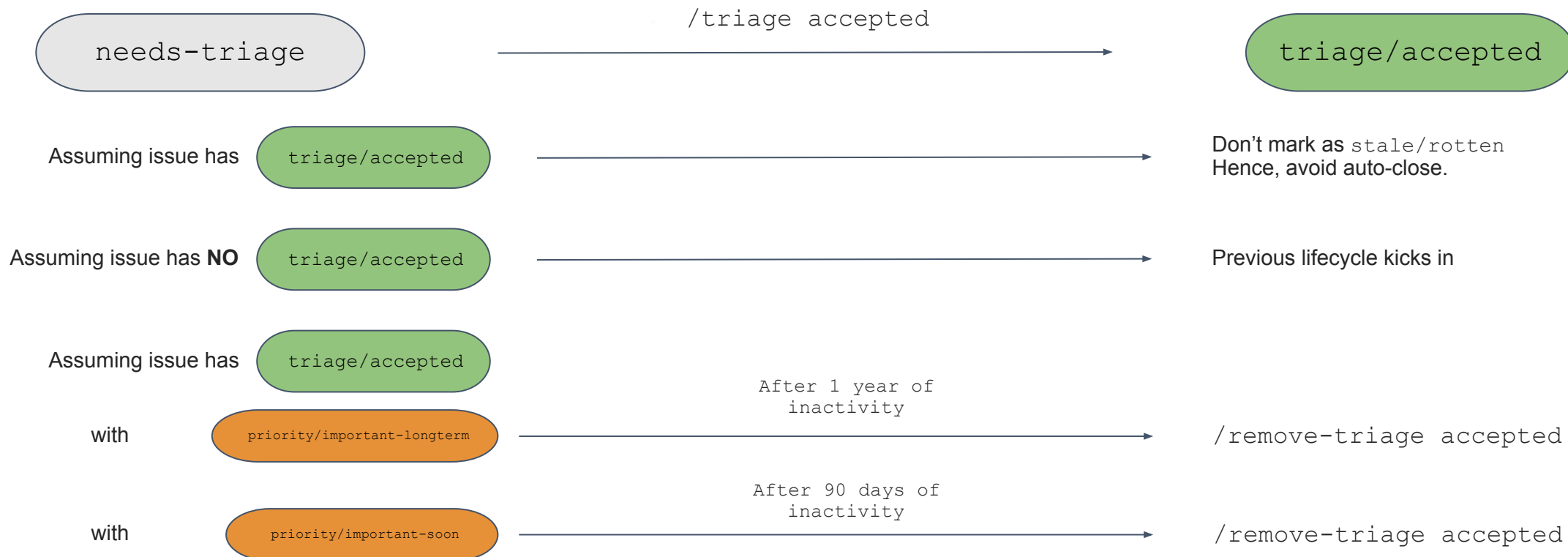
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



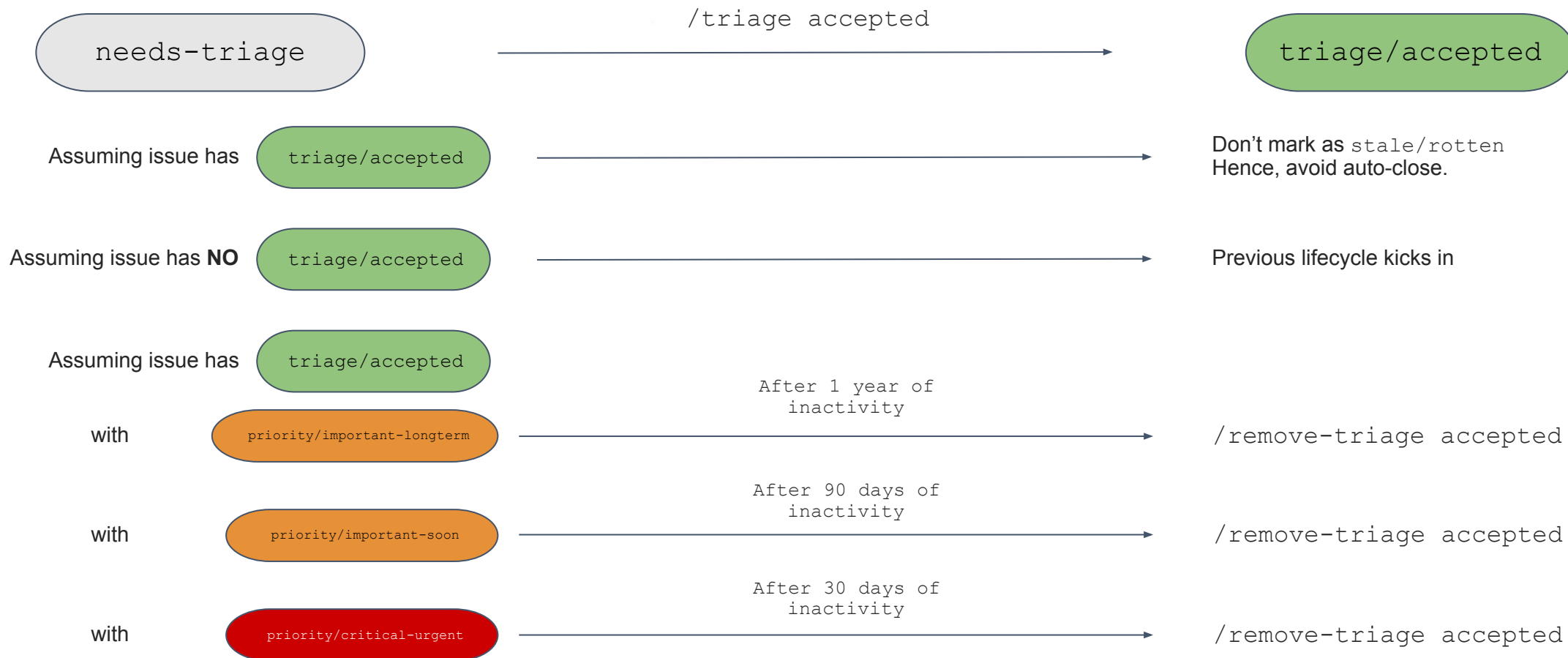
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



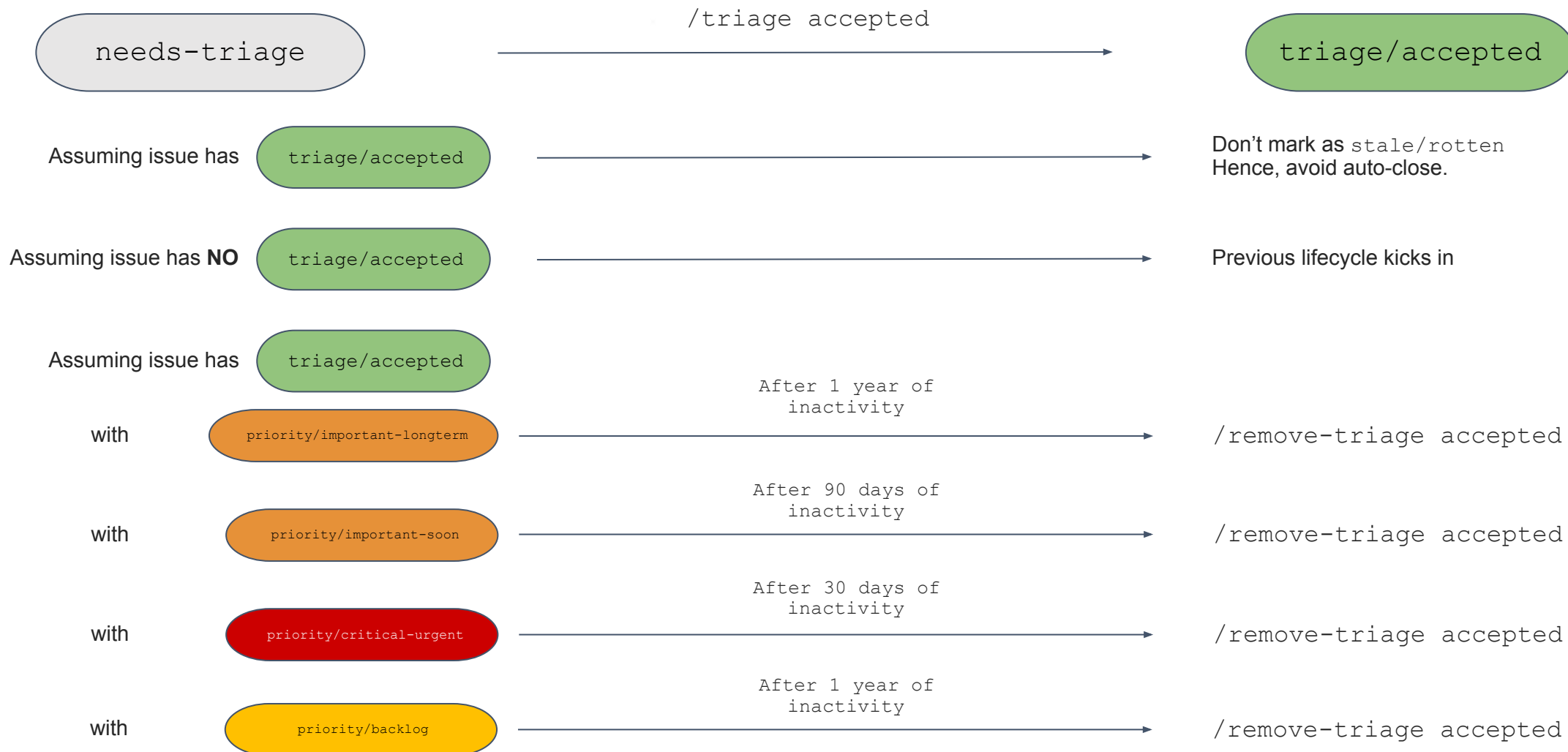
# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



# GitHub Management: Update

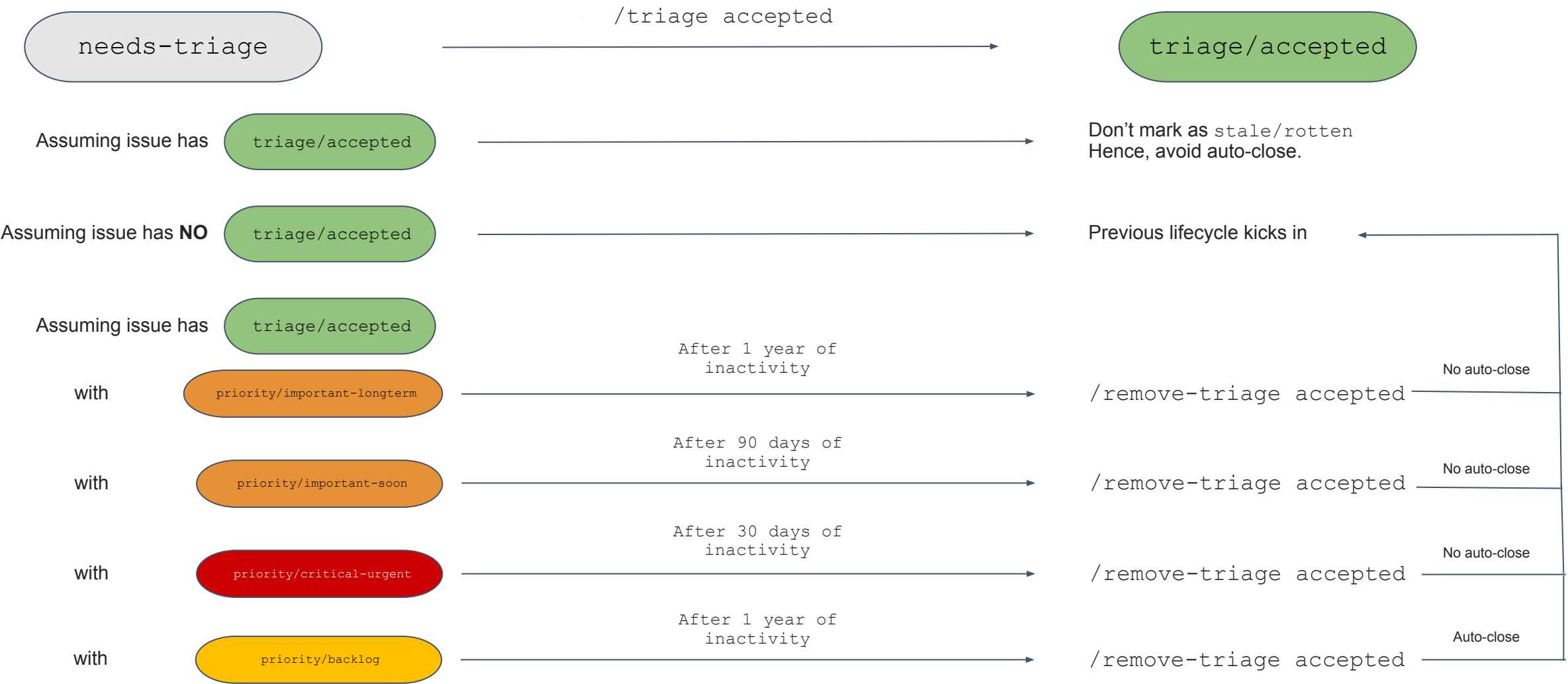
## New Issue Triage and Lifecycle Flow





# GitHub Management: Update

## New Issue Triage and Lifecycle Flow



## New Issue Triage and Lifecycle Flow

Note: Additionally, if issues have `good first issue` and/or `help wanted` labels, we exclude them from the entire lifecycle process.

# GitHub Management: Update

Thank you @tallclair and @BenTheElder for these changes! ✨

# Deep Dive: Annual Reports

## What Are Annual Reports?

- The Kubernetes Annual Report is a document worked on and produced by the Kubernetes community to succinctly describe what the community has been upto in the past year.
- This document covers details from what each SIG/WG has been upto, to broader topics of community health and calls for action.

## Why Are Annual Reports Important?

- Among other important things, it breaks down these updates by SIGs/WGs.
- And each of these have a **Help Wanted** section.
  - Use this to see what areas need most help in a SIG
  - You can also use this to aid conversations with your employer!

# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

```
› git clone https://github.com/kubernetes/community.git
```

```
› cd community/
```

```
› make ANNUAL_REPORT=true
```

# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

```
> make ANNUAL_REPORT=true
```

```
go run ./generator/app.go
```

```
...
```

```
Generating annual reports
```

```
Generating generator/generated/2022_sig-docs.md
```

```
...
```

```
Generating sig-api-machinery/annual-report-2022.md
```

```
...
```

```
Generating
```

```
generator/generated/2022_sig-instrumentation.md
```

```
...
```

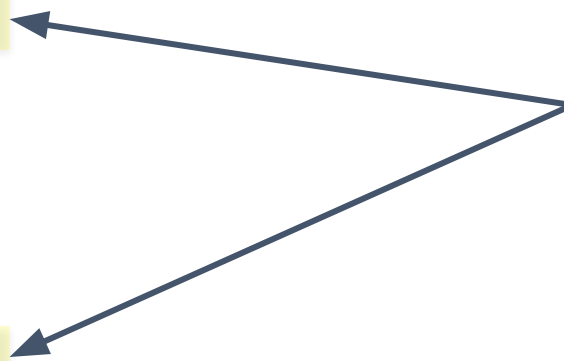
```
Generating wg-api-expression/annual-report-2022.md
```

```
...
```

```
Finished generation!
```

```
https://git.k8s.io/community/generator#how-to-use
```

**Generated Annual  
Report templates for  
SIG(s) and WG(s)**



# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

### 2022 Annual Report: SIG API Machinery

#### Current initiatives

1. What work did the SIG do this year that should be highlighted?
  - 
  - 
  -
2. What initiatives are you working on that aren't being tracked in KEPs?
  - 
  - 
  -
3. KEP work in 2022 (v1.24, v1.25, v1.26):
  - alpha:
    - [3156 - HTTP3](#) - v1.24
  - beta:
    - [1965 - kube-apiserver identity](#) - v1.26
    - [2876 - CRD Validation Expression Language](#) - v1.25
  - stable:
    - [1164 - Deprecate and remove SelfLink](#) - v1.24
    - [1904 - Efficient watch resumption after kube-apiserver reboot](#) - v1.24

#### Project health

1. What areas and/or subprojects does your group need the most help with? Any areas with 2 or fewer OWNERS? (link to more details)
  - 
  -

### 2022 Annual Report: WG API Expression

#### Current initiatives

1. What work did the WG do this year that should be highlighted? For example, artifacts, reports, white papers produced this year.
  - 
  - 
  -
2. What initiatives are you working on that aren't being tracked in KEPs?
  - 
  - 
  -

#### Project health

1. What's the current roadmap until completion of the working group?
  - 
  - 
  -
2. Does the group have contributors from multiple companies/affiliations?
  -
3. Are there ways end users/companies can contribute that they currently are not? If one of those ways is more full time support, what would they work on and why?
  -



# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

```
> make ANNUAL_REPORT=true
```

```
go run ./generator/app.go
```

```
...
```

```
Generating annual reports
```

```
Generating generator/generated/2022_sig-docs.md
```

```
...
```

```
Generating sig-api-machinery/annual-report-2022.md
```

```
...
```

```
Generating generator/generated/2022_sig-instrumentation.md
```

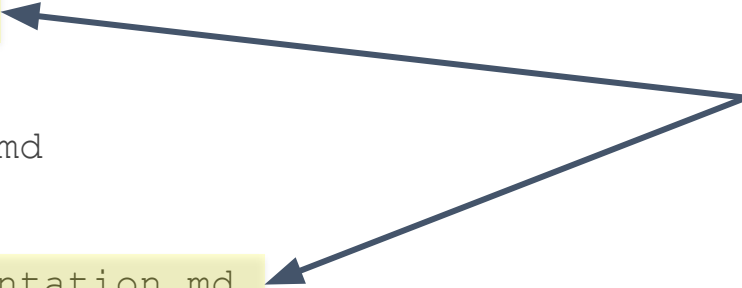
```
...
```

```
Generating wg-api-expression/annual-report-2022.md
```

```
...
```

```
Finished generation!
```

Generated GitHub  
Issue Templates  
for SIG(s) and  
WG(s)



# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

```
› for i in $(ls -1 generator/generated/*.md); do hub issue create -F $i && rm $i; done
```

# Deep Dive: Annual Reports

## Current Process for Generating Annual Reports

```
> for i in $(ls -1 generator/generated/*.md); do hub issue create -F $i && rm $i; done
```

### 2022 Annual Report: SIG Instrumentation #7107

Closed 6 tasks cblecker opened this issue on Feb 9 · 0 comments · Fixed by #7177

 **cblecker** commented on Feb 9 Member ...

Chairs: @ehashman @logicalhan  
Liaison: @cblecker

Actions for the chair/organizer of the community group:

- ☐ Consult your community group to complete draft of report
- ☐ If needed, consult with Steering Committee liaison on any private issues or concerns that arise while completing report
- ☐ Submit PR to [git.k8s.io/community](https://git.k8s.io/community) repo with copy of report, assigning your Steering Committee liaison for review
- ☐ Steering Committee reviews report, and may make comments/ask questions
- ☐ Once all comments are addressed, Steering Committee will approve report to merge
- ☐ If needed, any follow up items may be brought to the "Chairs, Tech Leads, and Organizers" meeting in April

Once all the above items are complete, this issue may be `/close 'd`

A template has already been generated for your group, and is available [here](#).

Key dates:

- Initial PR to community repo should be opened by March 24th, 2023
- PR should be reviewed and merged by April 7th, 2023


More detailed information on the annual reports process is available [here](#).

Your group's report template is available here: <https://git.k8s.io/community/sig-instrumentation/annual-report-2022.md>

If you have any questions or concerns about this process, you may reach the Steering Committee via the following methods:

### 2022 Annual Report: SIG Docs #7106

Closed 6 tasks cblecker opened this issue on Feb 9 · 0 comments · Fixed by #7216

 **cblecker** commented on Feb 9 Member ...

Chairs: @divya-mohan0209 @jimangel @natalisucks @reylejano  
Liaison: @cpanato

Actions for the chair/organizer of the community group:

- ☐ Consult your community group to complete draft of report
- ☐ If needed, consult with Steering Committee liaison on any private issues or concerns that arise while completing report
- ☐ Submit PR to [git.k8s.io/community](https://git.k8s.io/community) repo with copy of report, assigning your Steering Committee liaison for review
- ☐ Steering Committee reviews report, and may make comments/ask questions
- ☐ Once all comments are addressed, Steering Committee will approve report to merge
- ☐ If needed, any follow up items may be brought to the "Chairs, Tech Leads, and Organizers" meeting in April

Once all the above items are complete, this issue may be `/close 'd`

A template has already been generated for your group, and is available [here](#).

Key dates:

- Initial PR to community repo should be opened by March 24th, 2023
- PR should be reviewed and merged by April 7th, 2023

More detailed information on the annual reports process is available [here](#).

Your group's report template is available here: <https://git.k8s.io/community/sig-docs/annual-report-2022.md>

If you have any questions or concerns about this process, you may reach the Steering Committee via the following methods:

# Deep Dive: Annual Reports

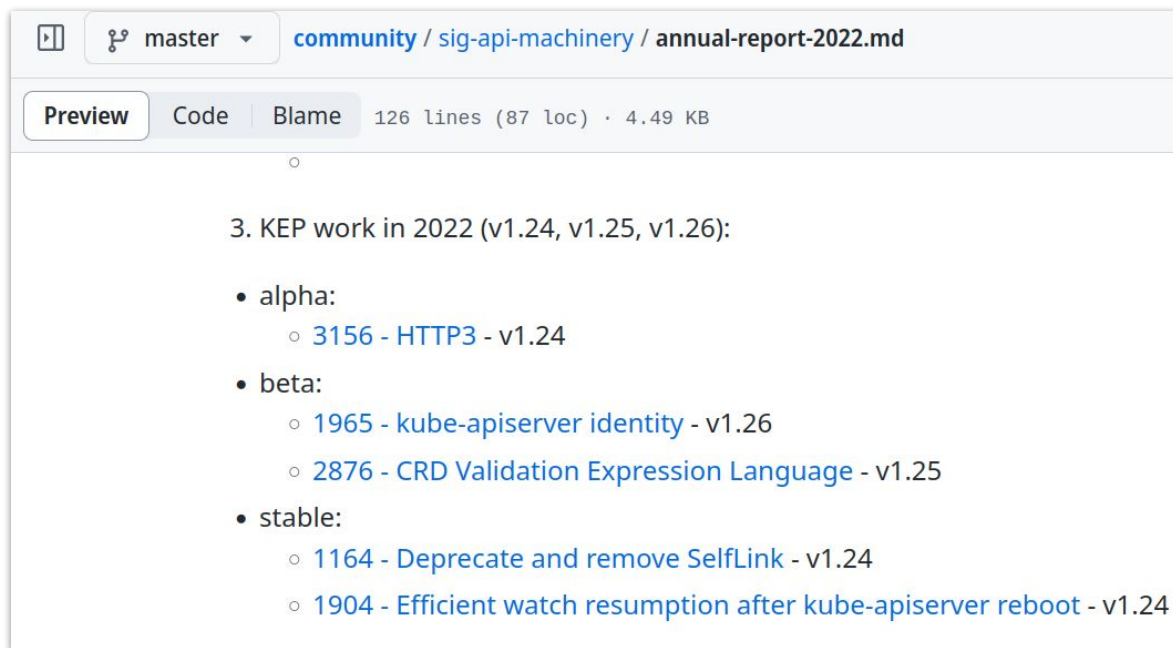
## Sprinkling Some Automation On Annual Reports

Auto generating list of KEPs worked on by SIG(s) or WG(s) during the Annual Report year

# Deep Dive: Annual Reports

## Sprinkling Some Automation On Annual Reports

Auto generating list of KEPs worked on by SIG(s) or WG(s) during the Annual Report year



community / sig-api-machinery / annual-report-2022.md

Preview Code Blame 126 lines (87 loc) · 4.49 KB

3. KEP work in 2022 (v1.24, v1.25, v1.26):

- alpha:
  - [3156 - HTTP3](#) - v1.24
- beta:
  - [1965 - kube-apiserver identity](#) - v1.26
  - [2876 - CRD Validation Expression Language](#) - v1.25
- stable:
  - [1164 - Deprecate and remove SelfLink](#) - v1.24
  - [1904 - Efficient watch resumption after kube-apiserver reboot](#) - v1.24



community / sig-network / annual-report-2022.md

Preview Code Blame 130 lines (91 loc) · 5.33 KB

3. KEP work in 2022 (v1.24, v1.25, v1.26):

- alpha:
  - [2091 - Add support for AdminNetworkPolicy resources](#) - v1.24
  - [2438 - Dual Stack API Server](#) - v1.24
- beta:
  - [2595 - Expanded DNS Configuration](#) - v1.26
- stable:
  - [1435 - Different protocols in the same service definition with type=loadbalancer](#) - v1.26
  - [1672 - Tracking Terminating Endpoints in EndpointSlice](#) - v1.26
  - [1864 - Optionally Disable Node Ports for Service Type=LoadBalancer](#) - v1.24
  - [2079 - Allow a Network Policy to contemplate a set of ports in a single rule](#) - v1.25
  - [2086 - Service Internal Traffic Policy](#) - v1.26
  - [2433 - Topology Aware Hints](#) - v1.26
  - [3070 - Reserve Service IP Range For Dynamic and Static IP Allocation](#) - v1.26

# Deep Dive: Annual Reports

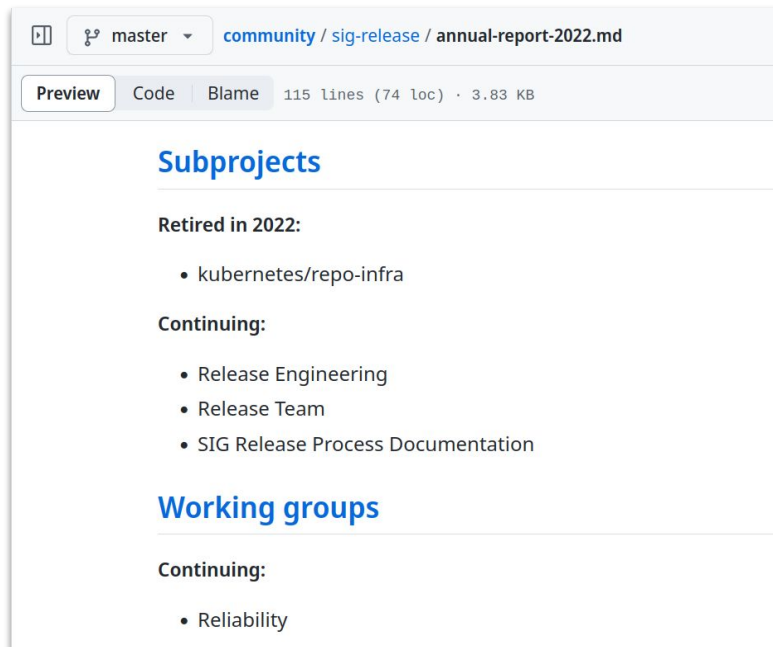
## Sprinkling Some Automation On Annual Reports

Auto-generating new, retired & continuing subprojects & working groups during the Annual Report year

# Deep Dive: Annual Reports

## Sprinkling Some Automation On Annual Reports

Auto-generating new, retired & continuing subprojects & working groups during the Annual Report year



community / sig-release / annual-report-2022.md

Preview Code Blame 115 lines (74 loc) · 3.83 KB

### Subprojects

Retired in 2022:

- kubernetes/repo-infra

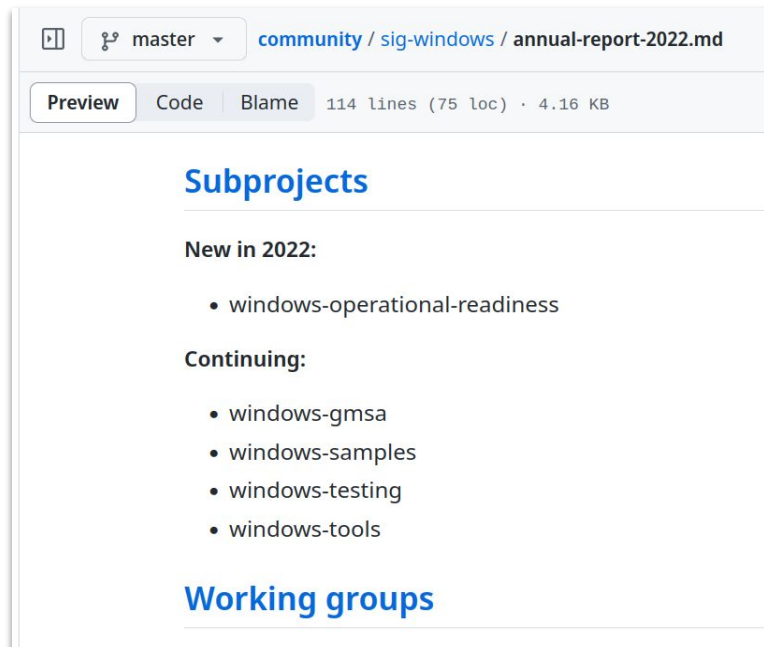
Continuing:

- Release Engineering
- Release Team
- SIG Release Process Documentation

### Working groups

Continuing:

- Reliability



community / sig-windows / annual-report-2022.md

Preview Code Blame 114 lines (75 loc) · 4.16 KB

### Subprojects

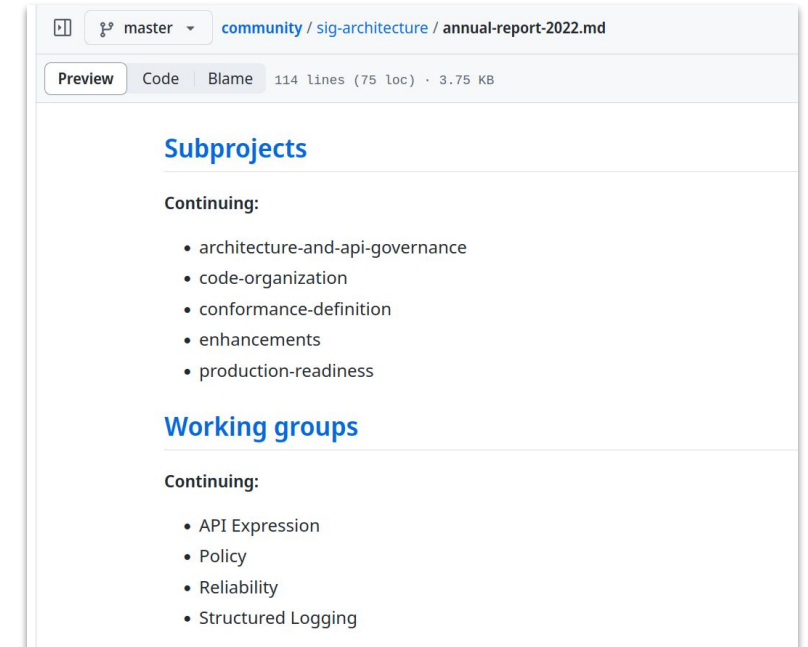
New in 2022:

- windows-operational-readiness

Continuing:

- windows-gmsa
- windows-samples
- windows-testing
- windows-tools

### Working groups



community / sig-architecture / annual-report-2022.md

Preview Code Blame 114 lines (75 loc) · 3.75 KB

### Subprojects

Continuing:

- architecture-and-api-governance
- code-organization
- conformance-definition
- enhancements
- production-readiness

### Working groups

Continuing:

- API Expression
- Policy
- Reliability
- Structured Logging

# Deep Dive: Granular Approval

What is “Approval” and how does it work currently?

- Reviewers
- Approvers
- OWNERS file

An OWNERS file:

```
reviewers:
```

```
- x
```

```
- y
```

```
approvers:
```

```
- z
```



# Deep Dive: Granular Approval

## Current Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    ├── OWNERS  
    └── apps/
```

`pkg/api/OWNERS = x, y, z (approvers)`

`pkg/registry/OWNERS = a, b, c (approvers)`

# Deep Dive: Granular Approval

## Current Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
      └─ apps/
```

`pkg/api/OWNERS = x, y, z (approvers)`

`pkg/registry/OWNERS = a, b, c (approvers)`

```
pkg  
├─ api  
│   └─ first.go  
│       └─ first_test.go  
└─ registry  
    └─ apps  
        └─ one.go  
            └─ one_test.go  
                └─ first.go
```

# Deep Dive: Granular Approval

## Current Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
      └─ apps/
```

pkg/api/OWNERS = x, y, z (approvers)

pkg/registry/OWNERS = a, b, c (approvers)

**x** comments: /approve

```
pkg  
├─ api  
│   └─ first.go  
│       └─ first_test.go  
└─ registry  
    └─ apps  
        └─ one.go  
            └─ one_test.go  
                └─ first.go
```

# Deep Dive: Granular Approval

## Current Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
      └─ apps/
```

pkg/api/OWNERS = x, y, z (approvers)

pkg/registry/OWNERS = a, b, c (approvers)

**x** comments: /approve

```
pkg  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry  
    └─ apps  
        └─ one.go  
            └─ one_test.go  
                └─ first.go
```

# Deep Dive: Granular Approval

## Current Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
        └─ apps/
```

pkg/api/OWNERS = x, y, z (approvers)

pkg/registry/OWNERS = a, b, c (approvers)

**b** comments: /approve

```
pkg ✓  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry ✓  
    └─ apps  
        └─ one.go ✓  
            └─ one_test.go ✓  
                └─ first.go ✓
```

# Deep Dive: Granular Approval

## Current Approval Process: The Problem

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
      └─ apps/
```

**b** comments: /approve

**b** only has expertise on testing,  
remaining expertise lies with **c**, **a**.

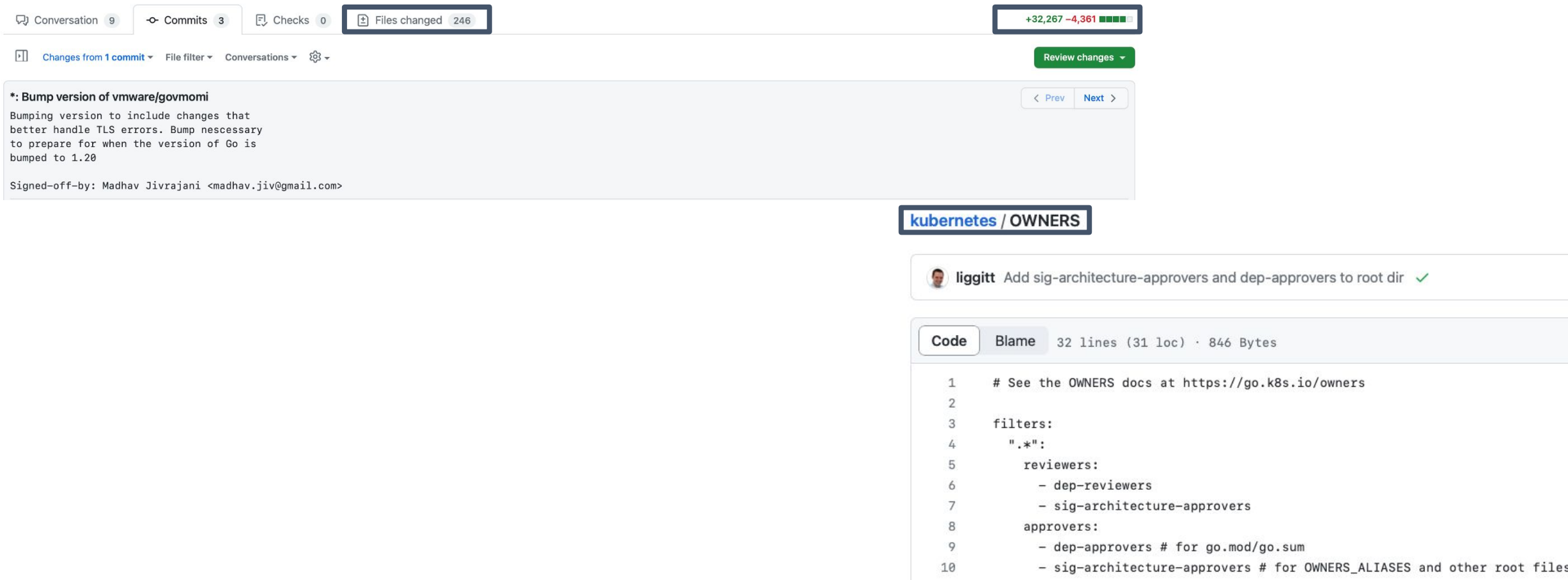
pkg/api/OWNERS = x, y, z (approvers)  
pkg/registry/OWNERS = a, b, c (approvers)

```
pkg ✓  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry ✓  
    └─ apps  
        └─ one.go ✓  
            └─ one_test.go ✓  
                └─ first.go ✓
```

# Deep Dive: Granular Approval

## Current Approval Process: The Problem

The problem becomes more elaborate when changes like dependency bumps are made to Kubernetes



The screenshot displays a GitHub pull request interface. At the top, navigation tabs include 'Conversation' (9), 'Commits' (3), 'Checks' (0), and 'Files changed' (246). A summary bar shows '+32,267 -4,361' with a green progress indicator. A 'Review changes' button is visible. The commit message is '\*: Bump version of vmware/govmmomi', with a description: 'Bumping version to include changes that better handle TLS errors. Bump necessary to prepare for when the version of Go is bumped to 1.20'. The commit is signed-off by Madhav Jivrajani. Below the commit, the 'kubernetes / OWNERS' file is highlighted. A review by 'liggitt' is shown, with the comment: 'Add sig-architecture-approvers and dep-approvers to root dir'. The 'Code' tab is selected, showing 32 lines of YAML configuration for the OWNERS file.

Conversation 9 Commits 3 Checks 0 Files changed 246

+32,267 -4,361

Review changes

< Prev Next >

\*: Bump version of vmware/govmmomi

Bumping version to include changes that better handle TLS errors. Bump necessary to prepare for when the version of Go is bumped to 1.20

Signed-off-by: Madhav Jivrajani <madhav.jiv@gmail.com>

kubernetes / OWNERS

liggitt Add sig-architecture-approvers and dep-approvers to root dir ✓

Code Blame 32 lines (31 loc) · 846 Bytes

```
1 # See the OWNERS docs at https://go.k8s.io/owners
2
3 filters:
4   ".*":
5     reviewers:
6       - dep-reviewers
7       - sig-architecture-approvers
8     approvers:
9       - dep-approvers # for go.mod/go.sum
10      - sig-architecture-approvers # for OWNERS_ALIASES and other root files
```

## Granular Approval

This work allows approvers to approve files granularly:

- `/approve files <path-to-a-single-file>`
- `/approve files <path-with-wild-card>`
- `/approve files <regex matching files>`

*This helps distribute approval privileges among approvers and approvers no longer have to (unavoidably) approve files they do not have confidence over.*



# Deep Dive: Granular Approval

## Granular Approval Process

`b comments: /approve *_test.go`

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
        └─ apps/
```

`pkg/api/OWNERS = x, y, z (approvers)`

`pkg/registry/OWNERS = a, b, c (approvers)`

```
pkg  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry  
    └─ apps  
        └─ one.go  
            └─ one_test.go  
                └─ first.go
```

# Deep Dive: Granular Approval

## Granular Approval Process

`b comments: /approve *_test.go`

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
        └─ apps/
```

`pkg/api/OWNERS = x, y, z (approvers)`

`pkg/registry/OWNERS = a, b, c (approvers)`

```
pkg  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry  
    └─ apps  
        └─ one.go  
            └─ one_test.go ✓  
└─ first.go
```

# Deep Dive: Granular Approval

## Granular Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
        └─ apps/
```

pkg/api/OWNERS = x, y, z (approvers)

pkg/registry/OWNERS = a, b, c (approvers)

```
b comments: /approve *_test.go  
c comments: /approve pkg/registry/apps/*
```

```
pkg  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry  
    └─ apps ✓  
        └─ one.go ✓  
            └─ one_test.go ✓  
└─ first.go
```

# Deep Dive: Granular Approval

## Granular Approval Process

```
pkg/  
├─ api/  
│   └─ OWNERS  
└─ registry/  
    └─ OWNERS  
        └─ apps/
```

pkg/api/OWNERS = x, y, z (approvers)

pkg/registry/OWNERS = a, b, c (approvers)

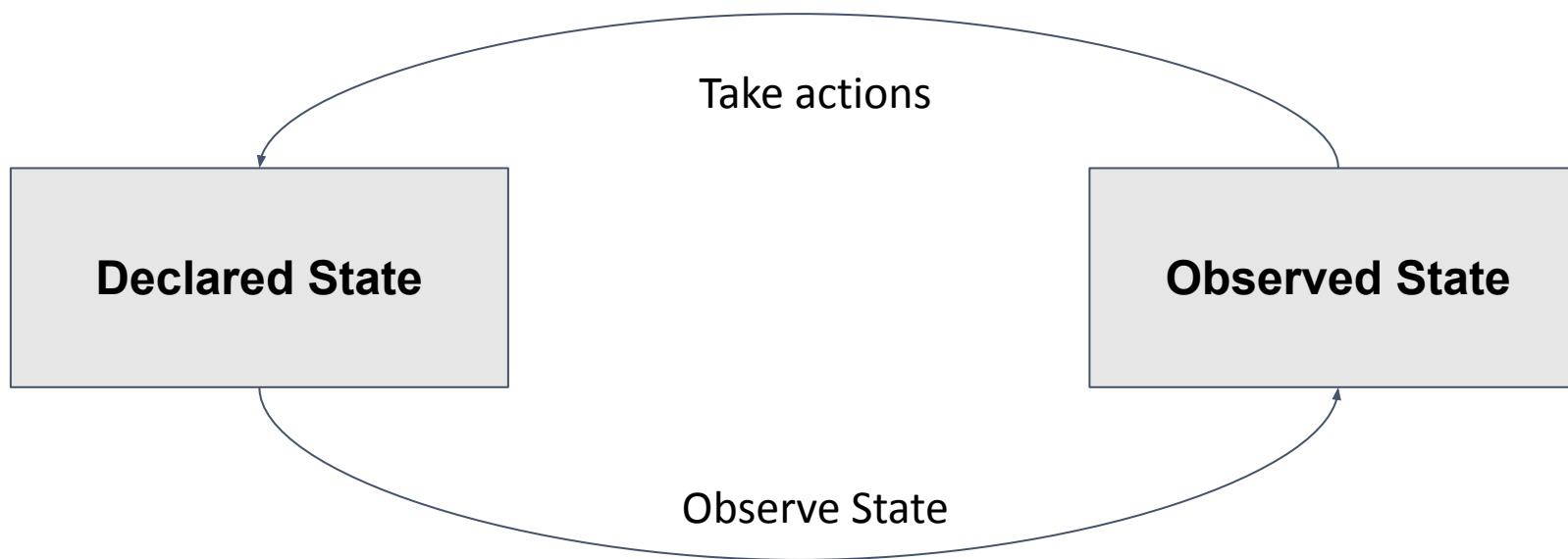
```
b comments: /approve *_test.go  
c comments: /approve pkg/registry/apps/*  
a comments: /approve
```

```
pkg ✓  
├─ api ✓  
│   └─ first.go ✓  
│       └─ first_test.go ✓  
└─ registry ✓  
    └─ apps ✓  
        └─ one.go ✓  
            └─ one_test.go ✓  
                └─ first.go ✓
```

# Deep Dive: Peribolos Team Management

What is “Peribolos”?

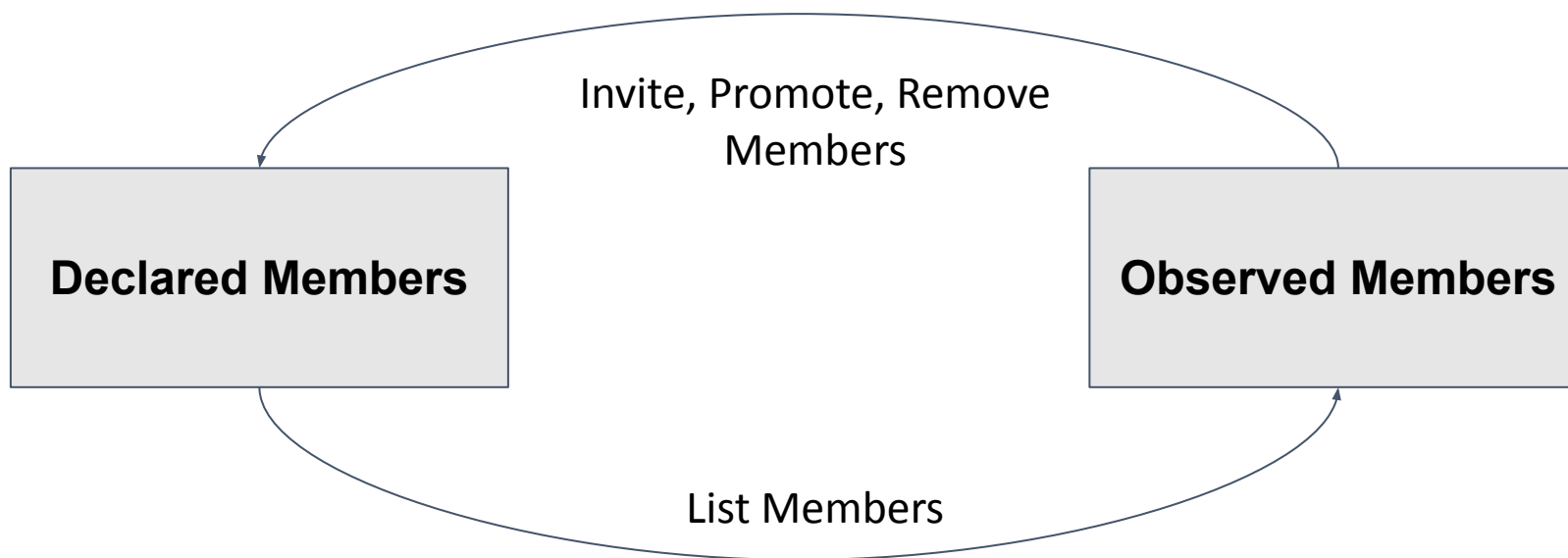
## Kubernetes Reconciliation Loop



# Deep Dive: Peribolos Team Management

What is “Peribolos”?

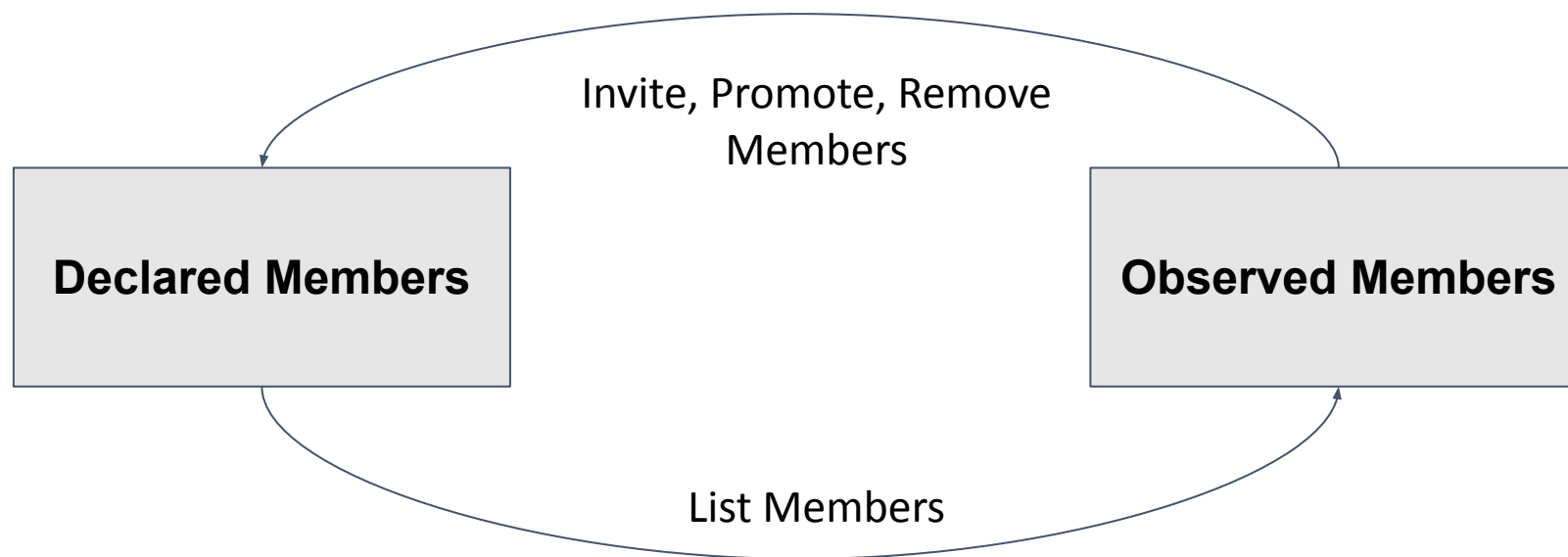
## Org Reconciliation Loop



# Deep Dive: Peribolos Team Management

What is “Peribolos”?

## Org Reconciliation Loop



Repeat for metadata, teams, et. al.

# Deep Dive: Peribolos Team Management

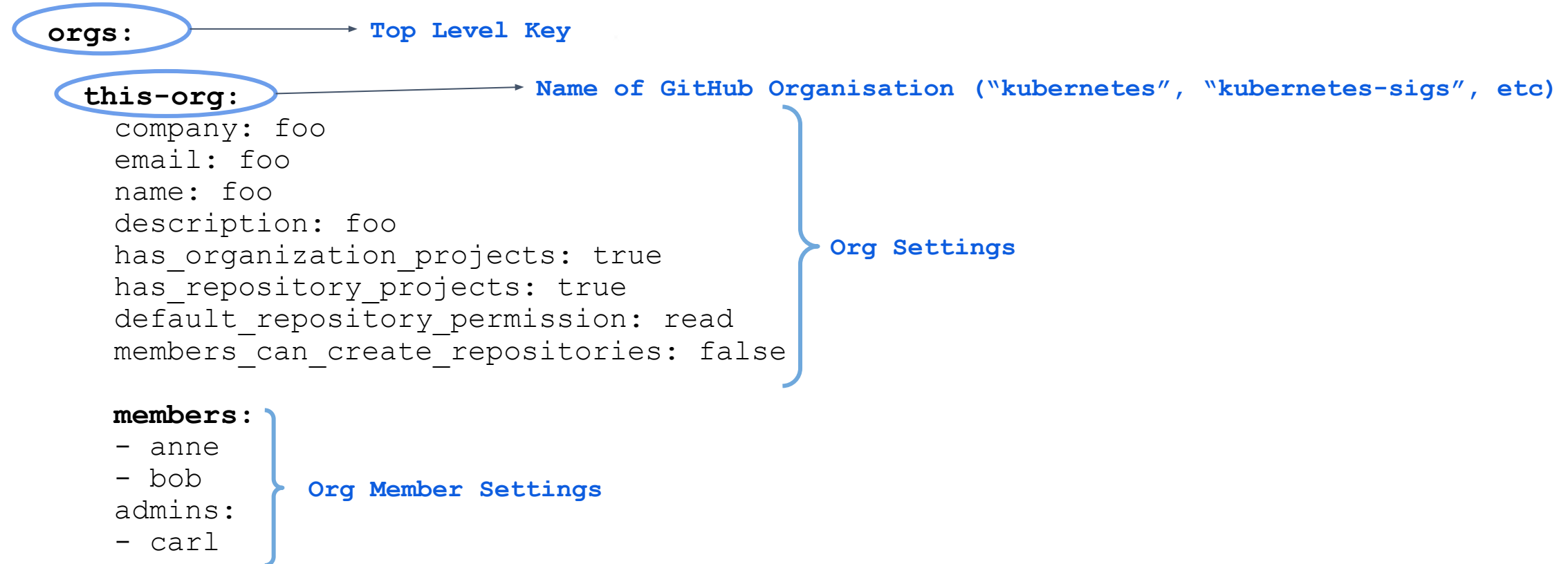
## How Does Peribolos Work?

- Peribolos enables the declaration of GitHub org settings, teams, and memberships in yaml format.
- GitHub is then updated to match the declared Org configuration.



# Deep Dive: Peribolos Team Management

## How Does Peribolos Work?



# Deep Dive: Peribolos Team Management

## How Does Peribolos Work?

orgs:

  this-org:

    ...

**teams:** → Teams Settings

      team-one:

        description: people working on team-one

        privacy: closed

        previously:

        - old-team-one

        members:

        - anne

        maintainers:

        - jane

        repos:

        some-repo: admin

        other-repo: read

      team-two:

    ...

  that-org:

    ...

Team Config

Team Members

Ensure Team has Listed Permissions Levels on Repos in Org

# Deep Dive: Peribolos Team Management

## How Does Peribolos Work?

- Peribolos can dump the current configuration of a GitHub org into a yaml format.

```
› go run ./prow/cmd/peribolos --dump kubernetes-sigs \  
  --github-token-path ~/github-token > org-config.yaml
```

## How Does Peribolos Work?

- Peribolos can dump the current configuration of a GitHub org into a yaml format.

```
› go run ./prow/cmd/peribolos --dump kubernetes-sigs \  
  --github-token-path ~/github-token > org-config.yaml
```

- Edit the ``org-config.yaml`` file before applying the org config, to remove any unwanted metadata that Peribolos shouldn't handle.

*( use ``org-config.yaml`` for dry-run first! )*

```
› go run ./prow/cmd/peribolos --config-path org.yaml \  
  --github-token-path ~/github-token # --confirm
```

# Deep Dive: Peribolos Team Management

## The Present State!

### Use peribolos to manage team repo permissions #2465



spiffxp opened this issue on Jan 28, 2021 · 12 comments · May be fixed by #2614



spiffxp commented on Jan 28, 2021

Member ...

Peribolos supports defining which repos a given team should have which level of access to (ref: <https://github.com/kubernetes/test-infra/tree/master/prow/cmd/peribolos#org-configuration>)

Currently this is all done manually. I think it would reduce our toil and make team permission changes more auditable if we drove this through our config files.

Some things to consider:

- the config looks like it only supports in-org assignments, do we need to support out-of-org assignments?
- we might want to consider adding a restrictions model to peribolos (similar to slack-infra), so that e.g. a team in a sub-folder can't arbitrarily give themselves kubernetes/kubernetes admin access



# Deep Dive: Peribolos Team Management

## The Present State!

### Use peribolos to manage team repo permissions #2465



spiffxp opened this issue on Jan 28, 2021 · 12 comments · May be fixed by #2614



spiffxp commented on Jan 28, 2021

Member ...

Peribolos supports defining which repos a given team should have which level of access to (ref: <https://github.com/kubernetes/test-infra/tree/master/prow/cmd/peribolos#org-configuration>)

Currently this is all done manually. I think it would reduce our toil and make team permission changes more auditable if we drove this through our config files.

Some things to consider:

- the config looks like it only supports in-org assignments, do we need to support out-of-org assignments?
- we might want to consider adding a restrictions model to peribolos (similar to slack-infra), so that e.g. a team in a sub-folder can't arbitrarily give themselves kubernetes/kubernetes admin access



# Deep Dive: Peribolos Team Management

## The Present State!

### Use peribolos to manage team repo permissions #2465



spiffxp opened this issue on Jan 28, 2021 · 12 comments · May be fixed by #2614



spiffxp commented on Jan 28, 2021

Member ...

Peribolos supports defining which repos a given team should have which level of access to (ref: <https://github.com/kubernetes/test-infra/tree/master/prow/cmd/peribolos#org-configuration>)

Currently this is all done manually. I think it would reduce our toil and make team permission changes more auditable if we drove this through our config files.

Some things to consider:

- the config looks like it only supports in-org assignments, do we need to support out-of-org assignments?
- we might want to consider adding a restrictions model to peribolos (similar to slack-infra), so that e.g. a team in a sub-folder can't arbitrarily give themselves kubernetes/kubernetes admin access





# Deep Dive: Peribolos Team Management

(WIP) Solution!



nikhita commented on Apr 4, 2021 • edited ▾

Member ...

Fixes [#2465](#)

→ This PR starts using peribolos to manage team repo permissions. Note - peribolos will fail if the repo doesn't exist, so we need to ensure that the repo exists before specifying it in the config. We could probably write a test to query the GitHub API to validate this, but I'd like to cover this in a follow-up.

Specifically, it includes the following changes:

- Add a new binary in `cmd/restrictions` to define restrictions to the team repo config so that a team in a sub-folder can't arbitrarily give themselves `kubernetes/kubernetes` admin access. The restrictions config is similar to [slack-infra](#), and can be extended in the future to define further restrictions.
- Add a verify script for restrictions. Note: I've not added this to bazel directly because I want to be able to run this script without bazel. This also meant that `verify-all.sh` now contains code to run all `hack/verify-*` scripts.
- Update all org configs to dump team repo permissions.
- There are a few teams that use the `triage` and `maintain` permissions. Peribolos was recently updated to support these ([🔗 Added support for full set of repository permissions in Peribolos](#) `test-infra#21526`, thanks [@MadhavJivrajani](#)!). To support this feature + some critical fixes related to team permissions ([🔗 peribolos: fix updating teams](#) `test-infra#19338`), this PR also bumps test-infra to the latest master (`kubernetes/test-infra@ 01a7a10`). I also copied over the replace statements from test-infra's [go.mod on master](#). Most of the diff is generated bazel changes due to this bump.
- Add a `config/restrictions.yaml` to restrict which repos teams can specify. The GitHub Management team will be able to approve changes to this file.



# Deep Dive: Peribolos Team Management

(WIP) Solution!



nikhita commented on Apr 4, 2021 • edited ▾

Member ...

Fixes [#2465](#)

This PR starts using peribolos to manage team repo permissions. Note - peribolos will fail if the repo doesn't exist, so we need to ensure that the repo exists before specifying it in the config. We could probably write a test to query the GitHub API to validate this, but I'd like to cover this in a follow-up.

Specifically, it includes the following changes:

- Add a new binary in `cmd/restrictions` to define restrictions to the team repo config so that a team in a sub-folder can't arbitrarily give themselves `kubernetes/kubernetes` admin access. The restrictions config is similar to [slack-infra](#), and can be extended in the future to define further restrictions.
- Add a verify script for restrictions. Note: I've not added this to bazel directly because I want to be able to run this script without bazel. This also meant that `verify-all.sh` now contains code to run all `hack/verify-*` scripts.
- Update all org configs to dump team repo permissions.
- There are a few teams that use the `triage` and `maintain` permissions. Peribolos was recently updated to support these ([🔗 Added support for full set of repository permissions in Peribolos](#) test-infra#21526, thanks @MadhavJivrajani!). To support this feature + some critical fixes related to team permissions ([🔗 peribolos: fix updating teams](#) test-infra#19338), this PR also bumps test-infra to the latest master ([kubernetes/test-infra@ 01a7a10](#)). I also copied over the replace statements from test-infra's [go.mod on master](#). Most of the diff is generated bazel changes due to this bump.
- Add a `config/restrictions.yaml` to restrict which repos teams can specify. The GitHub Management team will be able to approve changes to this file.

# Deep Dive: Peribolos Team Management

(WIP) Solution!



nikhita commented on Apr 4, 2021 • edited ▾

Member ...

Fixes [#2465](#)

This PR starts using peribolos to manage team repo permissions. Note - peribolos will fail if the repo doesn't exist, so we need to ensure that the repo exists before specifying it in the config. We could probably write a test to query the GitHub API to validate this, but I'd like to cover this in a follow-up.

Specifically, it includes the following changes:

- Add a new binary in `cmd/restrictions` to define restrictions to the team repo config so that a team in a sub-folder can't arbitrarily give themselves `kubernetes/kubernetes` admin access. The restrictions config is similar to [slack-infra](#), and can be extended in the future to define further restrictions.
- Add a verify script for restrictions. Note: I've not added this to bazel directly because I want to be able to run this script without bazel. This also meant that `verify-all.sh` now contains code to run all `hack/verify-*` scripts.
- Update all org configs to dump team repo permissions.
- There are a few teams that use the `triage` and `maintain` permissions. Peribolos was recently updated to support these ([🔗 Added support for full set of repository permissions in Peribolos](#) test-infra#21526, thanks @MadhavJivrajani!). To support this feature + some critical fixes related to team permissions ([🔗 peribolos: fix updating teams](#) test-infra#19338), this PR also bumps test-infra to the latest master ([kubernetes/test-infra@01a7a10](#)). I also copied over the replace statements from test-infra's [go.mod on master](#). Most of the diff is generated bazel changes due to this bump.
- Add a `config/restrictions.yaml` to restrict which repos teams can specify. The GitHub Management team will be able to approve changes to this file.

# Deep Dive: Peribolos Team Management

(WIP) Solution!

```
# restrictions.yaml
```

```
restrictions:
```

```
...
```

```
...
```

```
- path: "kubernetes/sigs-docs/teams.yaml"
  allowedRepos:
  - "^website"
```

```
...
```

```
...
```

```
- path: "kubernetes/sig-testing/teams.yaml"
  allowedRepos:
  - "^test-infra"
  - "^publishing-bot"
  ...
  ...
```

```
# kubernetes/sig-docs/teams.yaml
```

```
teams:
```

```
...
```

```
  website-admins:
```

```
    description: Admin access to the website
```

```
  repo
```

```
    previously:
```

```
    - kubernetes-website-admins
```

```
    members:
```

```
    - divya-mohan0209
```

```
    - jimangel
```

```
    - natalisucks
```

```
    - reylejano
```

```
  privacy: closed
```

```
  repos:
```

```
    website: admin
```

```
    test-infra: admin
```

# Deep Dive: Peribolos Team Management

(WIP) Solution!

```
# restrictions.yaml
```

```
restrictions:
```

```
  ...
  ...
  - path: "kubernetes/sigs-docs/teams.yaml"
    allowedRepos:
      - "^website"
  ...
  ...
  - path: "kubernetes/sig-testing/teams.yaml"
    allowedRepos:
      - "^test-infra"
      - "^publishing-bot"
  ...
  ...
```

```
# kubernetes/sig-docs/teams.yaml
```

```
teams:
```

```
  ...
  website-admins:
    description: Admin access to the website
    repo
    previously:
      - kubernetes-website-admins
    members:
      - divya-mohan0209
      - jimangel
      - natalisucks
      - reylejano
    privacy: closed
    repos:
      website: admin ✓
      test-infra: admin ✗
```



› make verify

INFO[0000] Validating restrictions for kubernetes org

**"kubernetes/sig-docs/teams.yaml": cannot define repo "test-infra" for team "website-admins"**

FATA[0000] restriction violation(s) detected.

exit status 1

# Deep Dive: Peribolos Team Management

(WIP) Solution!

```
# restrictions.yaml
```

```
restrictions:
```

```
...
```

```
...
```

```
- path: "kubernetes/sigs-docs/teams.yaml"
  allowedRepos:
    - "^website"
```

```
...
```

```
...
```

```
- path: "kubernetes/sig-testing/teams.yaml"
  allowedRepos:
    - "^test-infra"
    - "^publishing-bot"
  ...
  ...
```

```
# kubernetes/sig-docs/teams.yaml
```

```
teams:
```

```
...
```

```
  website-admins:
```

```
    description: Admin access to the website
    repo
```

```
    previously:
```

```
    - kubernetes-website-admins
```

```
    members:
```

```
    - divya-mohan0209
```

```
    - jimangel
```

```
    - natalisucks
```

```
    - reylejano
```

```
  privacy: closed
```

```
  repos:
```

```
    website: admin ✓
```

```
> make verify
```

```
INFO[0000] Validating restrictions for kubernetes org
```

```
SUCCESS verify-restrictions.sh 0s
```



# Get Involved!

- Every SIG/WG/UG has regularly scheduled meetings
- All meetings are uploaded to the [Kubernetes Community YouTube Channel](#)
- [SIG ContribEx Mailing List](#)

# Tips on your first SIG Meeting

- Find a buddy
- Volunteer to take notes
- Attend regularly
  - Be stubborn!
- Small, dependable contributions > volunteering for the world
- SIGs should have a list of “good first issues” for you to chew on, if they don’t then ... start with issue triaging



# There are non-code related paths available

**REQUEST: New Slack channel (k8s-infra-alerts) and integration (Google Cloud Monitoring) #4534**

**Closed** thockin opened this issue 15 days ago · 5 comments

thockin commented 15 days ago · edited ▾

Member + 😊 ...

Assignees

- jeefy —unassign me
- mrboobytables

Labels

- area/community-manag
- area/slack-management

GitHub Username: thockin

Slack Username: thockin

What Type of Request is it (Channel, User Group, Bot, Token, or Webhook)?. Channel & integration

Name of Requested Resource: k8s-infra-alerts and enable GCP Monitoring integration

**REQUEST: New moderator for New moderator for markyjackson-taulia of Zoom and YouTube #437**

**Closed** markyjackson-taulia opened this issue on Jan 7 · 4 comments · Fixed by #4375

markyjackson-taulia commented on Jan 7

Member + 😊 ...

GitHub Username

@markyjackson-taulia

Property you'd like to Moderate

Zoom, YouTube

Username on Property Requested

@markyjackson-taulia

Requirements

- ☒ I am a Kubernetes Org member already
- ☒ I have enabled 2FA on my account for the property I would like to moderate (if applicable).
- ☒ I have read and will abide by the policies mentioned in the [Moderation Guidelines](#)
- ☒ I have two sponsors that meet the sponsor requirements listed in the moderator guidelines
- ☒ I have spoken to my sponsors ahead of this application, and they have agreed to sponsor my application

**KEP for release notes improvements #928**

**Merged** k8s-ci-robot merged 1 commit into [kubernetes:master](#) from [jeefy:relnotes-kep](#) on Apr 23, 2019

Conversation 7 Commits 1 Checks 0 Files changed 1

jeefy commented on Mar 30, 2019

Member + 😊 ...

Rough draft for a KEP outlining several release notes improvements that have been discussed in various forums. I fully expect this to be iterated on quite a bit. 😊

/sig release

/cc @calebamiles @tpepper @spiffxp

3

If you're interested in having a non-code contributions meeting on a recurring meeting, please reach out on the #sig-contribex slack channel!



# What are the most important areas that you can help with?

- Mentoring Mentoring Mentoring!
  - Get involved with the reorganization of mentoring, in any area!
    - Mentoring co-lead
    - Group Mentoring Coordinator
    - Outreachy, GSoC and/or LFX coordinator
    - SIG Lead Mentoring Coordinator

# Where to find us

- Chairs
  - @mrbbobytables
  - @jberkus
  - @kaslin
  - @palnabarun
  - you?
- Technical Leads
  - @cblecker
  - @nikhita
  - @MadhavJivrajani
  - @Priyankasaggu11929
  - you?
- Home page: [README link](#)
- Slack channel: <https://kubernetes.slack.com/messages/sig-contribex>
- List: <https://groups.google.com/forum/#!forum/kubernetes-sig-contribex>

- Virtual attendees may submit questions to speakers through the CNCF Slack channel: **#2-Kubecon-sessions**
- Please create a thread and tag the speaker(s) with questions about their talk.
- Questions will be answered by the speaker and/or other community members after the session concludes.



**KubeCon**



**CloudNativeCon**

Europe 2023

# Thank you!

