
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

BUILDING FOR THE ROAD AHEAD

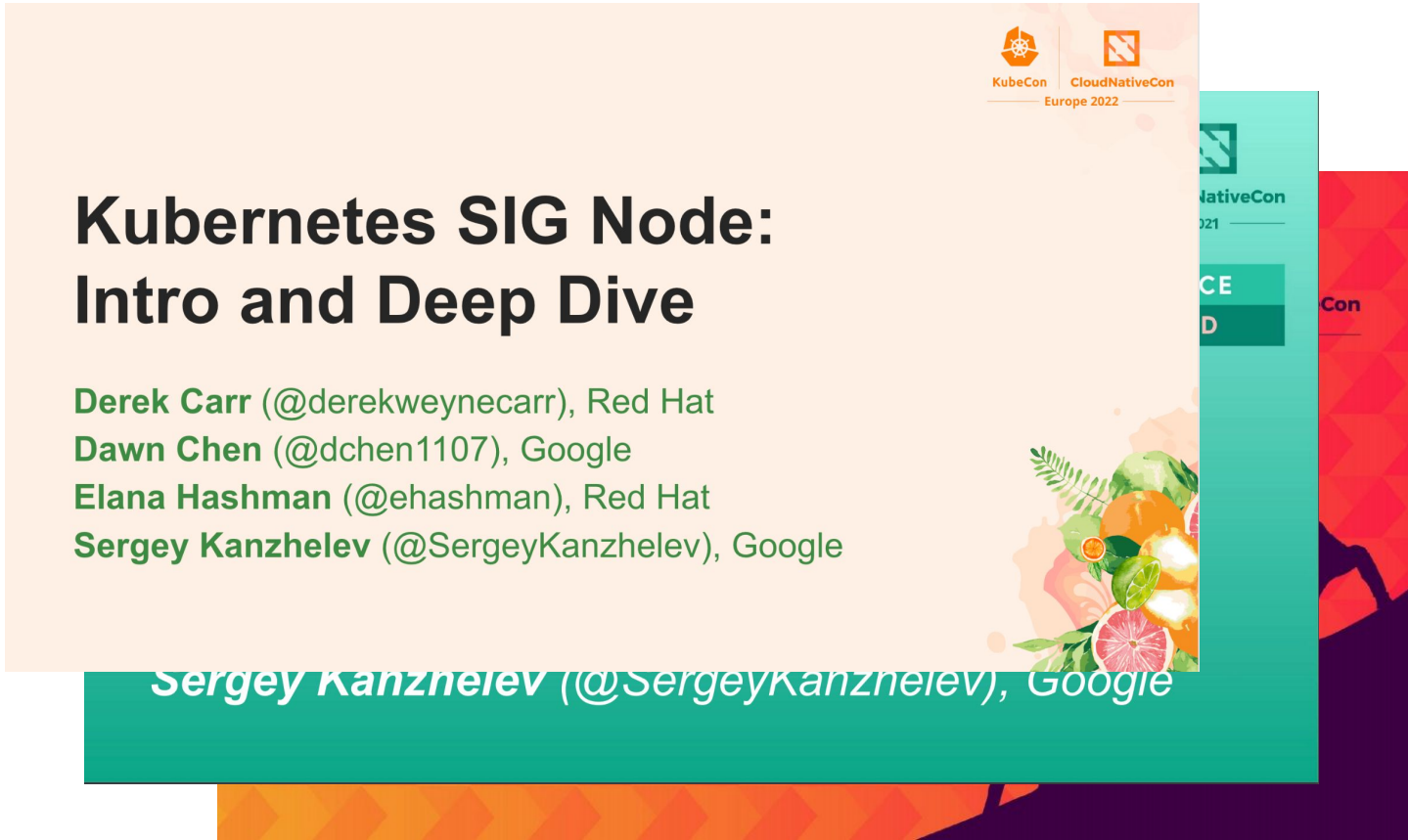
DETROIT 2022

Kubernetes SIG Node Intro And Deep Dive

Dawn Chen, Google
Derek Carr, Red Hat
Sergey Kanzhelev, Google

Previous updates from SIG Node

- Covered up to 1.25 roadmap at KubeCon EU 2022
  - **Recording** (https://youtu.be/FGRenKv4RgY)
  - **Sched** (https://sched.co/ytue)
  - **Slides**



Agenda

- SIG Node Overview
- Roadmap
  - Progress on no permanent betas
  - From 1.25 to 1.26 and onward
  - Highlight: Cgroup v2 GA’d
  - Highlight: inplace pod resizing
  - Highlight: Evented PLEG
- Subprojects
  - Kernel modules
  - Dynamic resource management
  - Batch group participation
  - CI Subproject
- Get Involved
  - How to contribute and where to find us

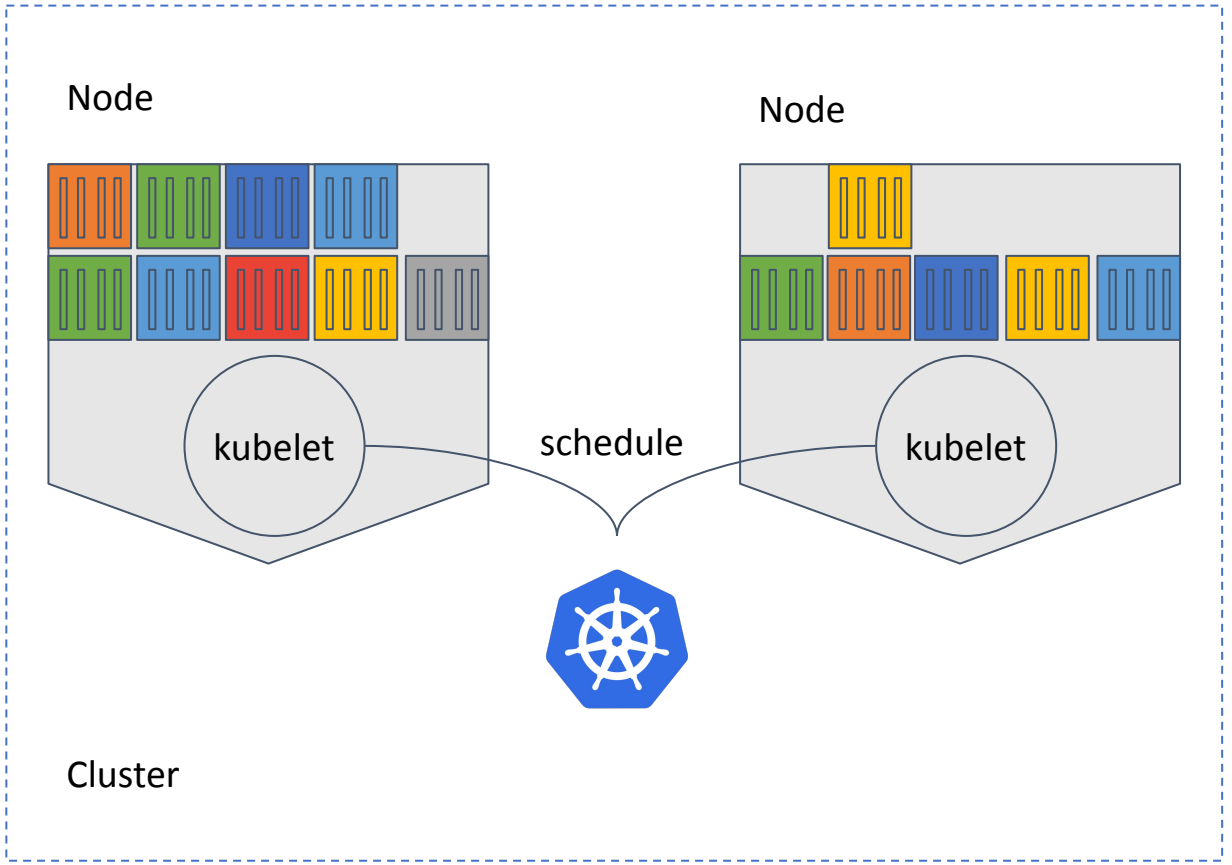
KubeCon | CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

SIG Node Overview

SIG Node Overview

SIG Node Areas

- Kubelet
- Node and Pods Lifecycle
- Resource management
- Container runtimes



SIG Charter & Subprojects

- **Charter**: We are responsible for the components that support the controlled interactions between pods and host resources.
- SIG Node is a *vertical SIG*
- **Subprojects**:
  - Kubelet
  - Container Runtime Interface (CRI)
  - Node Problem Detector
  - Kernel modules
  - … and more!

SIG Node
no permanent betas

Avoiding permanent betas

Starting 1.20 the focus was to avoid permanent betas.

https://kubernetes.io/blog/2020/08/21/moving-forward-from-beta/#avoiding-permanent-beta

Many features were graduated or deprecated:

- DynamicKubeletConfig
- Ephemeral containers
- RuntimeClass & PodOverhead
- CRI Logs rotation
- Sysctl support
- …

Old feature gates

Goal: Graduate or deprecate

- **AppArmor** beta since 1.4 *(plan to graduate in 1.26)*
- **DevicePlugins** beta since 1.10 *(plan to graduate in 1.26)*
- **QOSReserved** alpha since 1.11
- **RotateKubeletServerCertificate** beta since 1.12
- **CustomCPUCFSQuotaPeriod** alpha since 1.12
- **KubeletPodResources** beta since 1.15
- **TopologyManager** beta since 1.18 *(discussions are ongoing)*
- **DownwardAPIHugePages** beta since 1.21

Old feature WIP

Goal: Some features are WIP

- **CPUManager** beta since 1.10
- **CPUManagerPolicyAlphaOptions** alpha since 1.23
- **CPUManagerPolicyBetaOptions** beta since 1.23

The work is ongoing

Feature gates in 1.22+

Goal: avoid making them permanent betas.

~21 feature gates with last update in 1.22+

We actively develop new features and keep focus on features adoption and graduation.

KubeCon | CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

SIG Node Roadmap
(1.25-1.26+)

Graduations and Deprecations

Goal: clean up tech debt, reduce maintenance surface

Graduation: when a feature flag is removed

Deprecation: when a feature is disabled and eventually removed

- #277 Ephemeral containers graduate to stable (1.25)
- #361 Local ephemeral storage capacity isolation (1.25)
- #2254 Cgroup v2 graduate to stable (1.25)
- #1867 In-tree accelerator usage metrics removed (1.25)
- #2803 Identify Pod OS to stable (1.25)

Beta graduations

Beta: when feature gated code defaults to enabled

- #2413 Enable seccomp by default (1.25)

New (alpha) features

Alpha: net new features, behind a feature gate and disabled by default

- #2371 cAdvisor-less, CRI-full Container and Pod stats (1.25)
- #2008 Forensic Container Checkpointing (1.25)
- #2831 Kubelet OpenTelemetry Tracing (1.25)
- #127 User namespaces for stateless pods (1.25)

1.26 and onward

- **In place update of pod resources:** enable vertical resizing of pods
- **Evented PLEG:** improve node overhead and responsiveness
- **Ensure secret pulled images:** moving more image management to runtimes
- **Pod lifecycle flexibility:** exploring possible evolutions of pod lifecycle approaches
- **Resource plugins:** exploring possible evolutions to delegate resource management to others

…and more!

cgroups v2

Capability

- **Goal**
  - Cgroup v1 and v2 have feature parity since 1.24
  - We are NOT deprecating Cgroup v1 support
  - New resource controllers to be added for v2 only
- Enabling cgroup v2
  - Defaulted to on in newer versions of Fedora, Ubuntu, COS, and other Linux distros
- Exploring new features
  - Memory QoS
  - oomd support for userspace out of memory killer

Get ready for cgroup v2

- Most workloads have no dependency on cgroup version.
- Recommend usage of systemd cgroup manager in runtime.
- It is hard to automatically find dependency if any, please test.
- Check for third-party vendors
  - It is typical for security and monitoring agents to have dependency on cgroup version
  - Most of latest versions of agents has support for cgroup v2
  - ex: use latest versions of cAdvisor to have cgroup v2 support

Domain specific challenges

- **Things to be aware of**
  - Telco: Disabling CPU load balancing not in v2
    - Working with upstream kernel community to get it back
  - Golang: Manually set GOMAXPROCS or use tooling
  - Java: Needs JDK15 or later to leverage
  - Autoscalers: More testing needed with resource metric providers
    - Help needed with in-place updates
- **Call to action**
  - If your domain/runtime/vendor has known gaps, please share!
- **Future areas of investigation (reach out if interested)**
  - Using pressure stall information (PSI) metrics for eviction
  - User space oom killer (oomd)

In Place Pod Resizing

Autoscaling in Kubernetes

- Horizontal pod autoscaler to handle more requests
- Cluster autoscaling to scale the overall cluster size
- Vertical pod autoscaler automatically sets resource limits and requests

But VPA requires a restart?!

In Place Pod Resizing

- **Goal**
  - Right size the application without restarts
- **Long Desired Feature**
  - Initial discussion since 2015
  - Actively designed and implemented since 2018
  - Iterated and evolved along with hugepage, swap, cgroup v2, …
  - Planed to alpha in 1.26
- Call to action
  - Review, try and share the feedback!

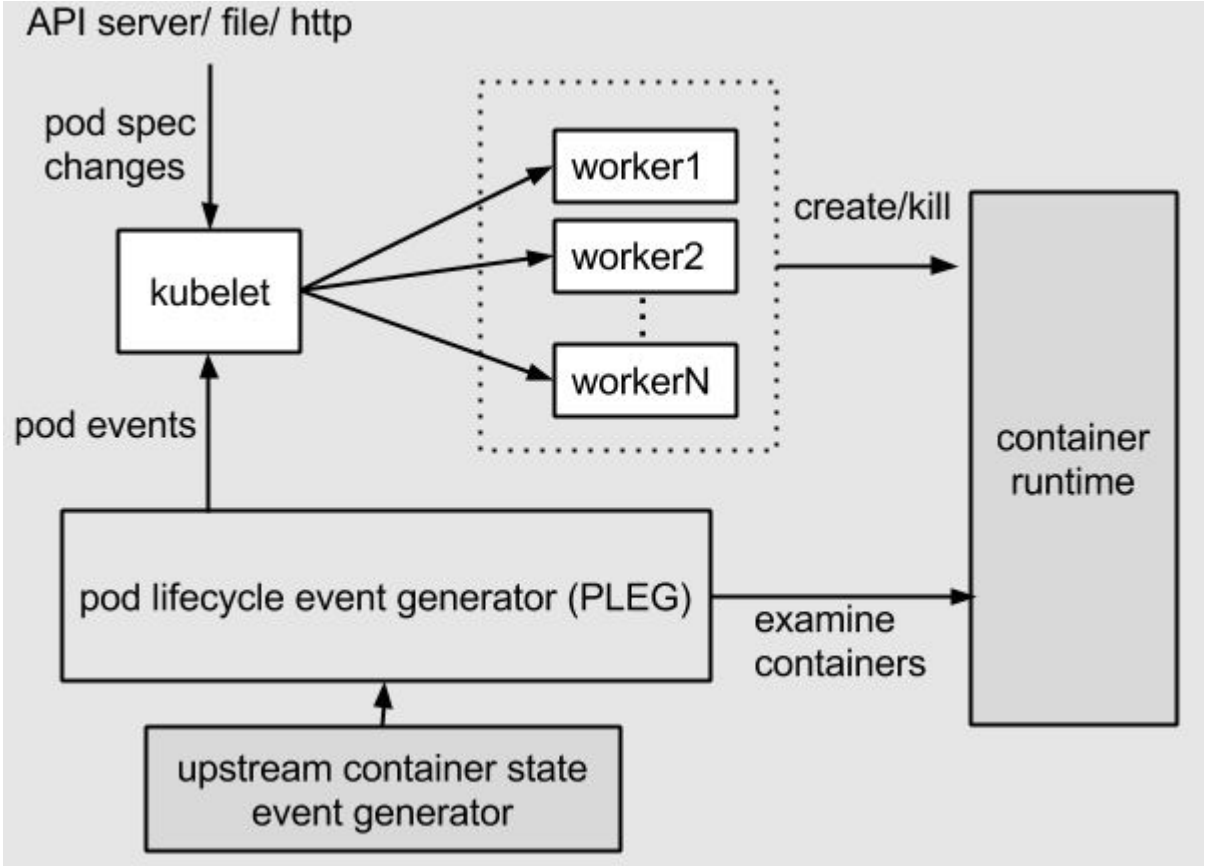
Evented PLEG

PLEG

The PLEG (Pod Lifecycle Event Generator) is responsible for generating events for when containers start and stop on the container runtime.

Current implementation:

The PLEG periodically polls the container runtime for all containers and compares previous state to current state and generates events based on this information that feed into the SyncLoop.



PLEG v2

New requirements:

- More pods and containers
- High availability workloads and failure detection
- “Low overhead” environments support

Current implementation

- Simple
- Guarantees consistency
- Performance is acceptable
- Works with any runtime

Evented PLEG

- Streaming and eventing
- Faster detection
- Improved resource utilization
- Modern runtimes are needed

Subprojects

Kernel Module Management

kubernetes-sigs / kernel-module-management

Public

generated from kubernetes/kubernetes-template-project

Code | Issues 9 | Pull requests 8 | Actions | Projects | Security | Insights

main | 5 branches | 0 tags | Go to file | Add file | Code

bthurber Update contacts (#48) ... 56b8bb3 5 days ago **211** commits

.github	Streamline GitHub Actions. (#39)	19 days ago
api/v1beta1	Adding support for image push during preflight build verification (#76)	7 days ago
ci	Add the -operator suffix to the image name. (#79)	7 days ago
config	Update of Build configuration in Module should trigger new Job (#81)	5 days ago
controllers	Update of Build configuration in Module should trigger new Job (#81)	5 days ago
docs	Add first document for lab (#23)	7 days ago
hack	Run operator-sdk init.	7 months ago
internal	Fixing build job deletion, in case its pods are still running (#85)	5 days ago
.dockerignore	Run operator-sdk init.	7 months ago

Dynamic Resource Allocation

- Goals
  - Accelerate the new hardware / device adoptions
- Timelines
  - The initial proposal made in 2018
  - New design approved and merged in 1.25
  - Plan to have alpha in 1.26
- Please join **Slack channel:** #sig-node for discussion

Batch Working group participation

Batch Working Group

Batch workload is not typical for Kubernetes. Batch processing is a method of running **jobs** in batches automatically. While users are required to submit the jobs, no other interaction by the user is required to process the batch. Batches may automatically be run at scheduled times as well as being run contingent on the availability of computer resources.

From WG proposal:

Support for Batch lagged in Kubernetes core, leading to a challenging migration journey of batch workloads to Kubernetes. Multiple past efforts tried to improve this status, but those efforts lacked continuity, in some cases leading to forked projects outside k8s, including a forked scheduler.

Batch Working Group

SIG Node involvement:

- Retriable and non-retriable Pod failures for Jobs
- Keystone containers
- Better resource utilization
  - NUMA-aware scheduling
  - Device Plugins, etc.

CI Subproject

CI workgroup

CI subproject tracks tests health and proactively control k8s reliability.

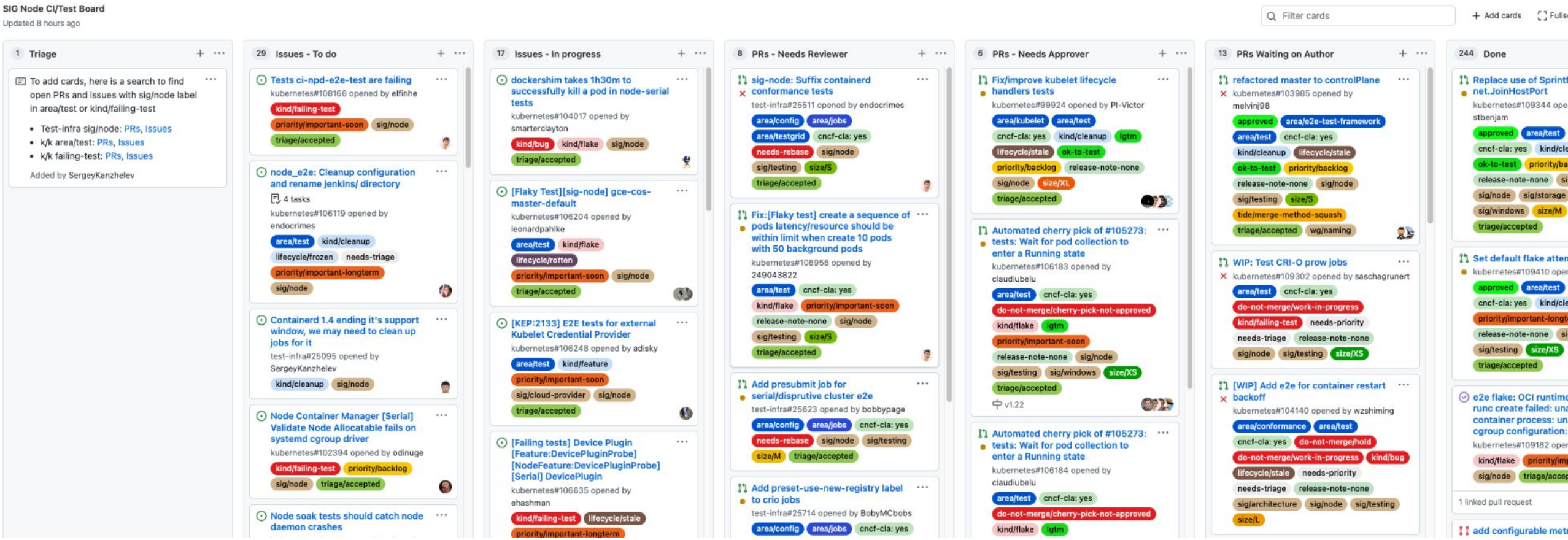
No perma failures:

- Some tests are failing for a long time now
- Old test grid tabs being removed or fixed
- CRI-O is being tested regularly
- New tests classification work is ongoing

CI/Test Issues and PRs review

KubeCon CloudNativeCon North America 2022

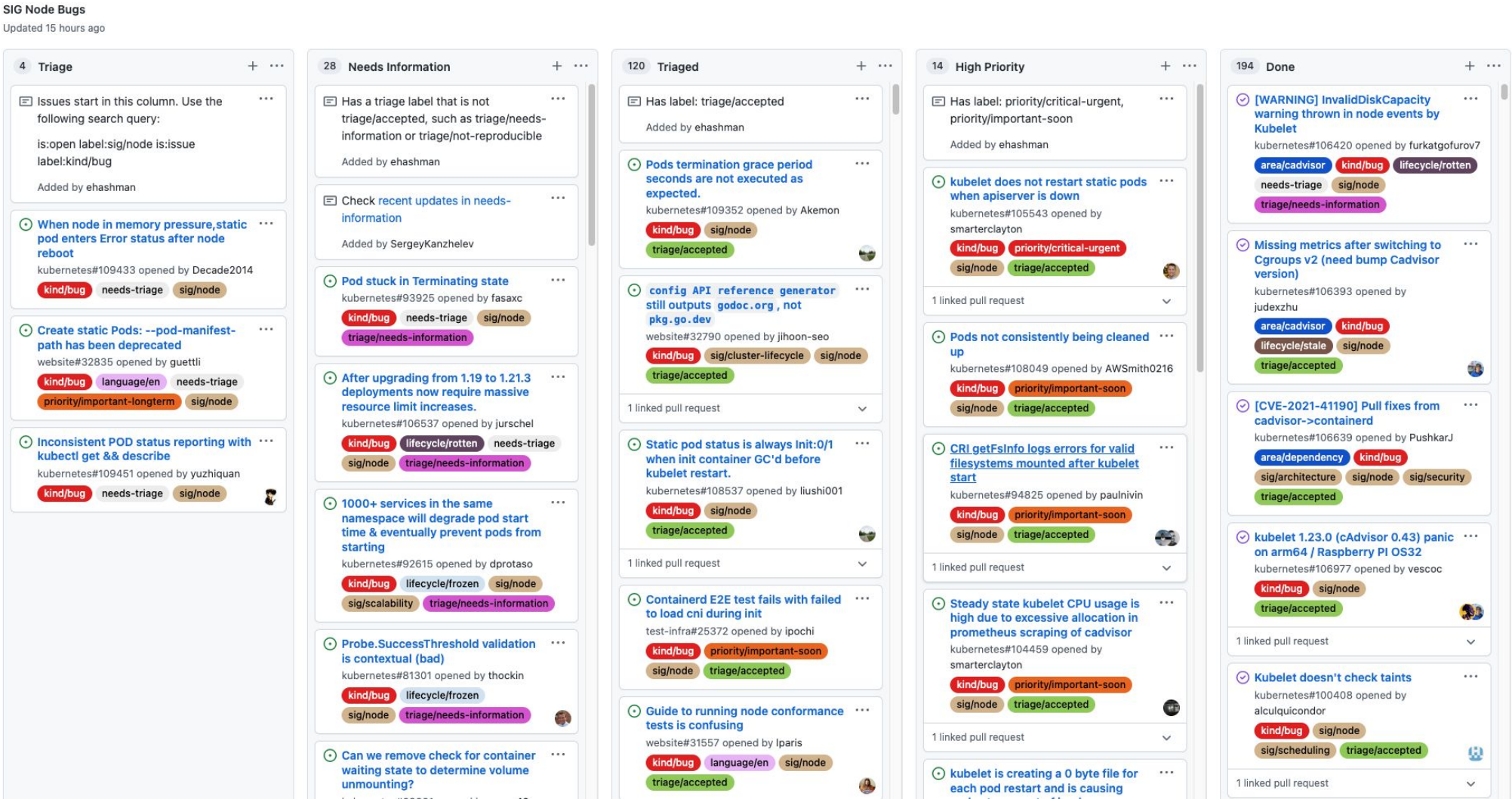
- Tracking work



Bugs triage

KubeCon CloudNativeCon North America 2022

- Receiving reliability signal early

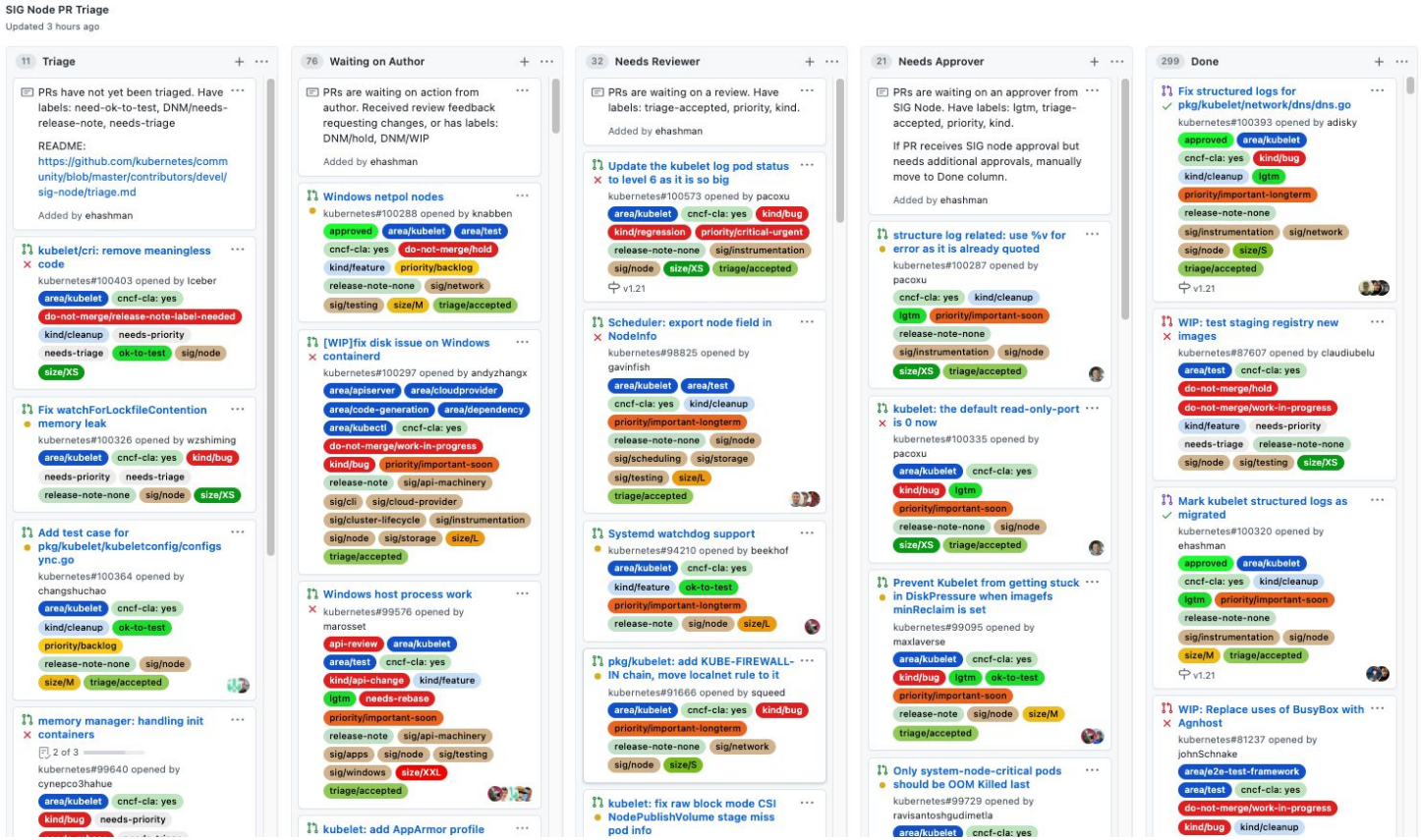


Get Involved!

GitHub workload

SIG Node is the **3rd largest** SIG by absolute workload.

- **200+** average **open** PRs
- **20-60** PRs **merged/ closed** weekly
- Devs from **39+ companies made** contributions over the past year
- Help is needed!



Contribution Priorities

- Stability first!
  - Tests > bug fixes/open issues > features
  - Test infrastructure monitoring and health
- Optimizations
- Features
- Improve the user and developer experience
  - Documentation
  - Enhance logging and metrics
  - Keeping on top of PRs/issues

How to Contribute

- **Attend our SIG meetings!** 🤝
  - *SIG meetings* cover features, KEPs, and more
  - *CI/Triage meetings* are a smaller, hands-on group
    - Good setting for learning how to triage, improve CI
- **Participate in reviews, issues, and docs!** 🙋
  - Triage Guide
  - Main Node PRs / CI and Test Enhancements Board / Bugs
- **Adopt new features and give feedback!** 🎁
  - Testing in real environments is critical for graduating features

Where to find us

- **SIG Meetings:**
  - Regular meeting, **weekly** on **Tuesdays at 10am Pacific Time**
  - CI/Triage meeting, **weekly** on **Wednesdays at 10am Pacific Time**
- **Slack channel:** #sig-node
- **Mailing list:** kubernetes-sig-node
- **Chairs:** Dawn Chen and Derek Carr

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Thank you!