Renovate Your Life

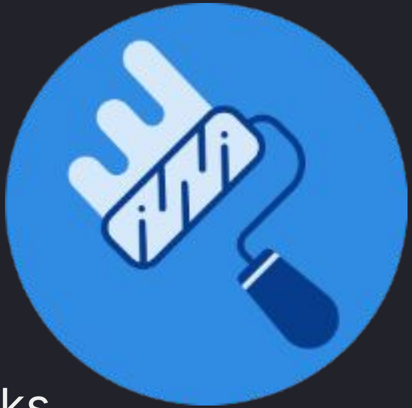
How we automated dependency updates for 1,300 Repos (and lived to tell the tale)

Dimitrios Sotirakis

Philip Hope

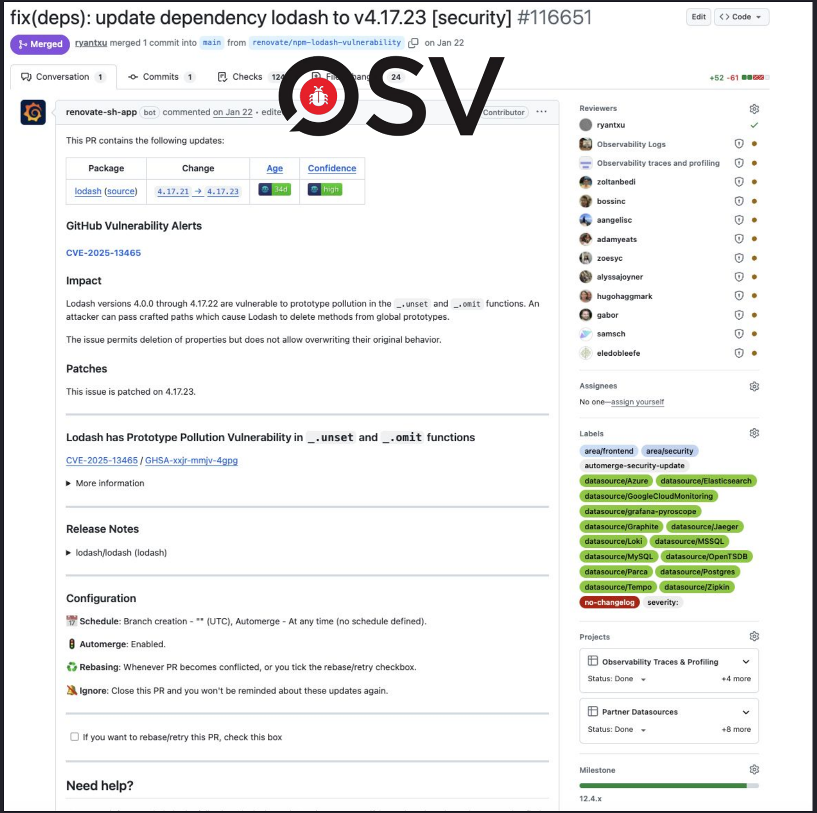
What is Renovate?

- Automated dependency update tool by Mend.io
- Scans repositories and creates update pull requests
- Multi-platform and multi-ecosystem support (30+ package managers)
- Eliminates manual tracking and reduces security risks
- Prevents technical debt accumulation



OSV Vulnerability Scanning

- Integration with OSV.dev database
- Real-time vulnerability detection
- Flags vulnerable dependencies in PRs
- Prioritizes security-critical updates



fix(deps): update dependency lodash to v4.17.23 [security] #116651

ryantxu merged 1 commit into `main` from `renovate/npm-lodash-vulnerability` on Jan 22

renovate-sh-app bot commented on Jan 22 • edite

This PR contains the following updates:

Package	Change	Age	Confidence
lodash (source)	`4.17.21` → `4.17.23`	34d	high

GitHub Vulnerability Alerts

CVE-2025-13465

Impact

Lodash versions 4.0.0 through 4.17.22 are vulnerable to prototype pollution in the `_.unset` and `_.omit` functions. An attacker can pass crafted paths which cause Lodash to delete methods from global prototypes.

The issue permits deletion of properties but does not allow overwriting their original behavior.

Patches

This issue is patched on 4.17.23.

Lodash has Prototype Pollution Vulnerability in `_.unset` and `_.omit` functions

CVE-2025-13465 / GHSA-xxjr-mmjv-4gpg

▶ More information

Release Notes

▶ lodash/lodash (lodash)

Configuration

📅 **Schedule**: Branch creation - "" (UTC), Automerge - At any time (no schedule defined).

🚦 **Automerge**: Enabled.

♻ **Rebasing**: Whenever PR becomes conflicted, or you tick the rebase/retry checkbox.

🔕 **Ignore**: Close this PR and you won't be reminded about these updates again.

☐ If you want to rebase/retry this PR, check this box

Need help?

Reviewers

- ryantxu ✓
- Observability Logs ●
- Observability traces and profiling ●
- zoltanbedi ●
- bossinc ●
- aangelisc ●
- adamyeats ●
- zoesyc ●
- alyssajoyner ●
- hugohaggmark ●
- gabor ●
- samsch ●
- eledobleefe ●

Assignees

No one—assign yourself

Labels

area/frontend area/security automerge-security-update datasource/Azure datasource/Elasticsearch datasource/GoogleCloudMonitoring datasource/grafana-pyroscope datasource/Graphite datasource/Jaeger datasource/Loki datasource/MSSQL datasource/MySQL datasource/OpenTSDB datasource/Parca datasource/Postgres datasource/Tempo datasource/Zipkin no-changelog severity:

Projects

Observability Traces & Profiling
Status: Done +4 more

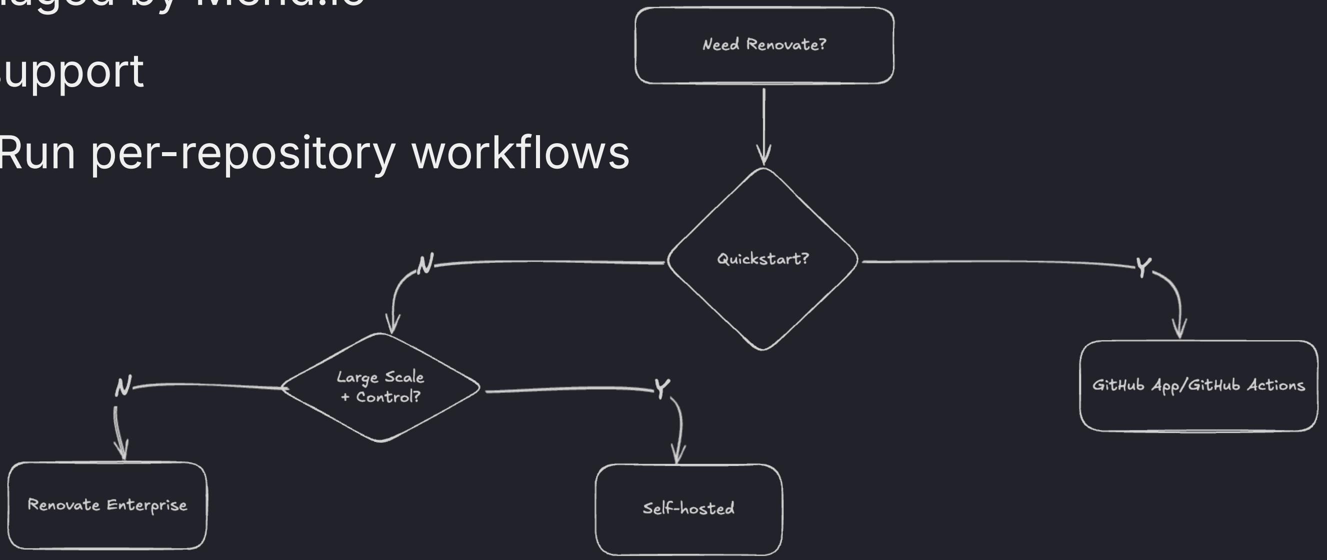
Partner Datasources
Status: Done +8 more

Milestone

12.4.x

How can it run?

- GitHub App - Hosted by Mend.io
- Self-hosted - Run in your infrastructure
- Enterprise - Managed by Mend.io with dedicated support
- GitHub Action - Run per-repository workflows



But why not Dependabot?

Renovate

VS

✅ Highly flexible configuration and advanced features (even works with Dependabot vuln alerts)
✅ Scales to hundreds of repos with centralized control
❌ Complex initial setup and ongoing maintenance

Dependabot

✅ Zero-config setup, works out of the box

❌ Limited customization and no cross-repo management

How do we run it?

What Renovate does already

Global Config

Local Config

Autodiscover

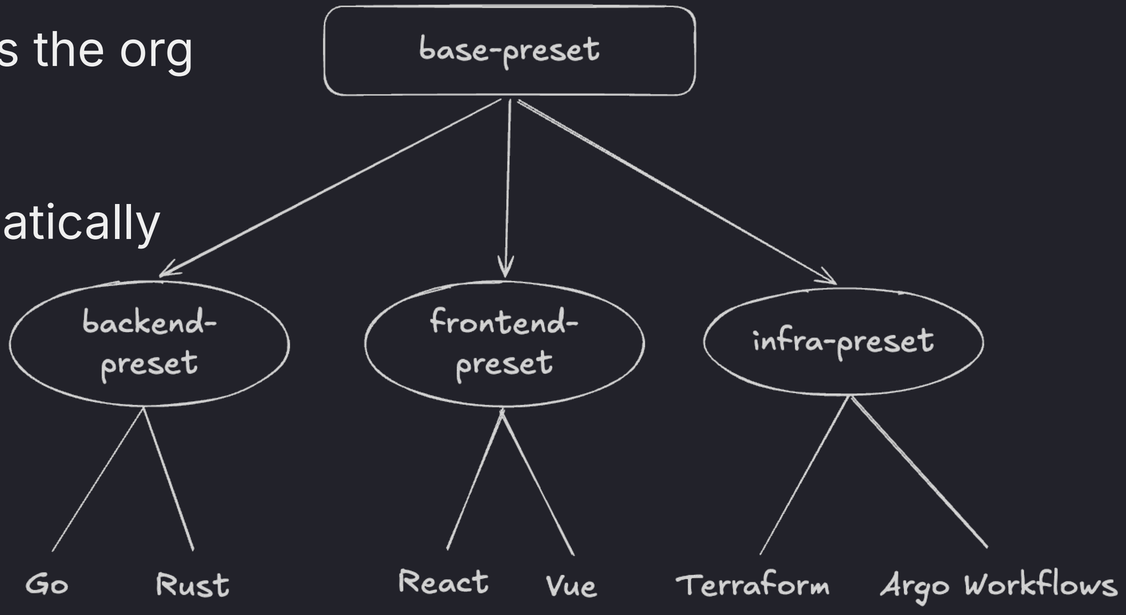
Shared Presets

- Central repository containing presets for different squads/teams
- Teams extend from mandatory global config + squad-specific presets
- Enables knowledge sharing

  and standardization across the org
- Version controlled —

  changes propagate automatically

  to all repos

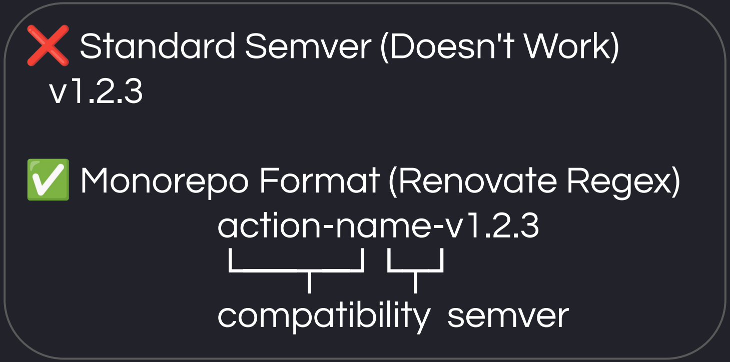


Authorising against private registries

- `hostRules` define per-registry authentication (Docker, npm, Maven, and more)
- Configured via env vars — never in the config file
- Credentials stay out of version-controlled config files
- Secrets sourced from Vault or Kubernetes secrets for secure rotation

Custom Versioning for Monorepos

- Monorepo actions released independently with tags like *action-name-v1.2.3*
- Standard semver tools can't parse this — Renovate's regex versioning can
- Regex extracts the action name and version components separately
- Customizes branch prefixes, PR titles and commit messages per action



❌ Standard Semver (Doesn't Work)

v1.2.3

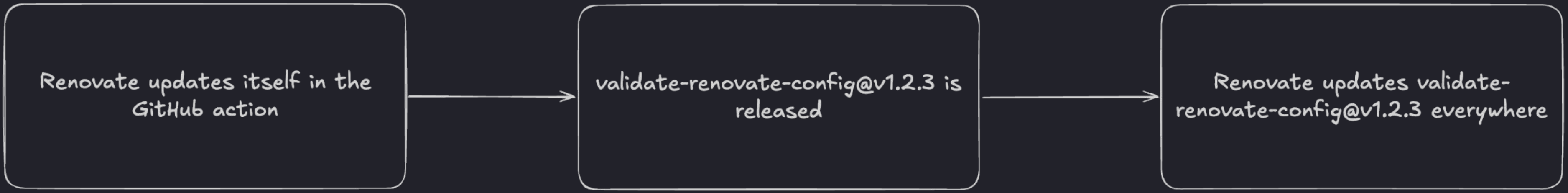
✅ Monorepo Format (Renovate Regex)

action-name-v1.2.3

compatibility semver

Validating Renovate

- Reusable GitHub Action wrapping renovate-config-validator
- Runs validation against a pinned Renovate version (renovate@XX.YY)
- Catches config errors before they reach production
- Defaults to renovate.json but accepts any config file path as input

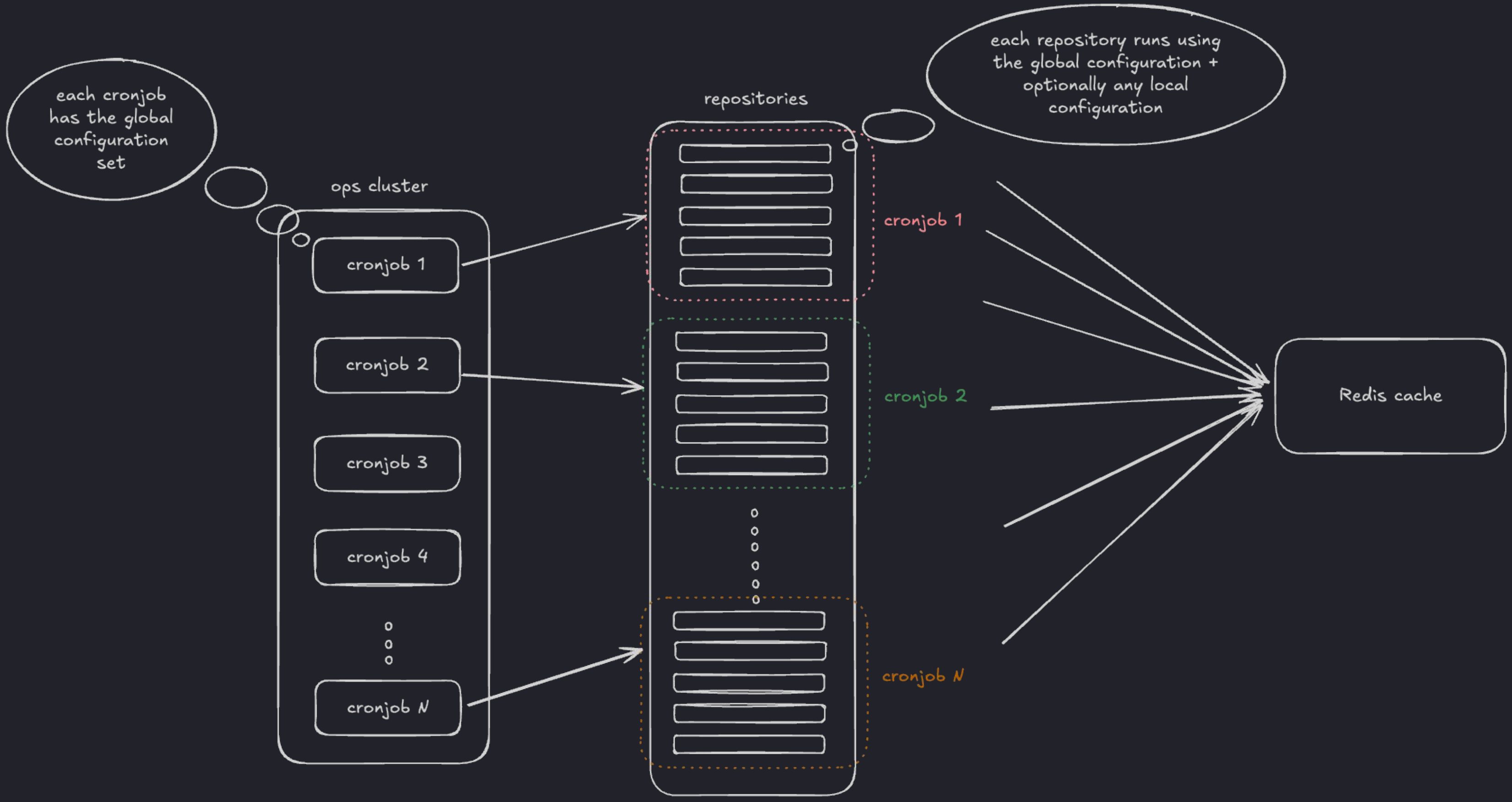


Renovate updates itself in the GitHub action

validate-renovate-config@v1.2.3 is released

Renovate updates validate-renovate-config@v1.2.3 everywhere

How self-hosted Renovate runs


each cronjob has the global configuration set
ops cluster
cronjob 1
cronjob 2
cronjob 3
cronjob 4
cronjob N
repositories
each repository runs using the global configuration + optionally any local configuration
cronjob 1
cronjob 2
cronjob N
Redis cache

Operating at scale

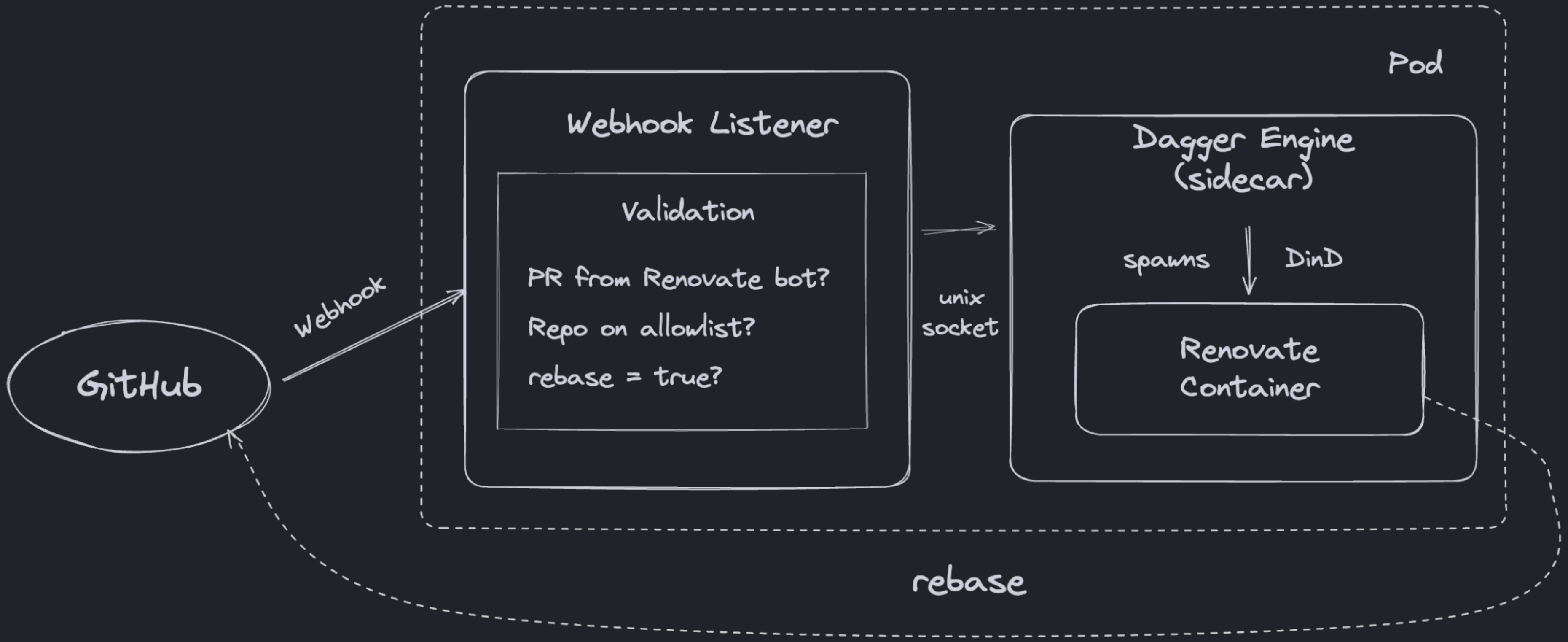
Rate Limits

App Token TTL

Autodiscover

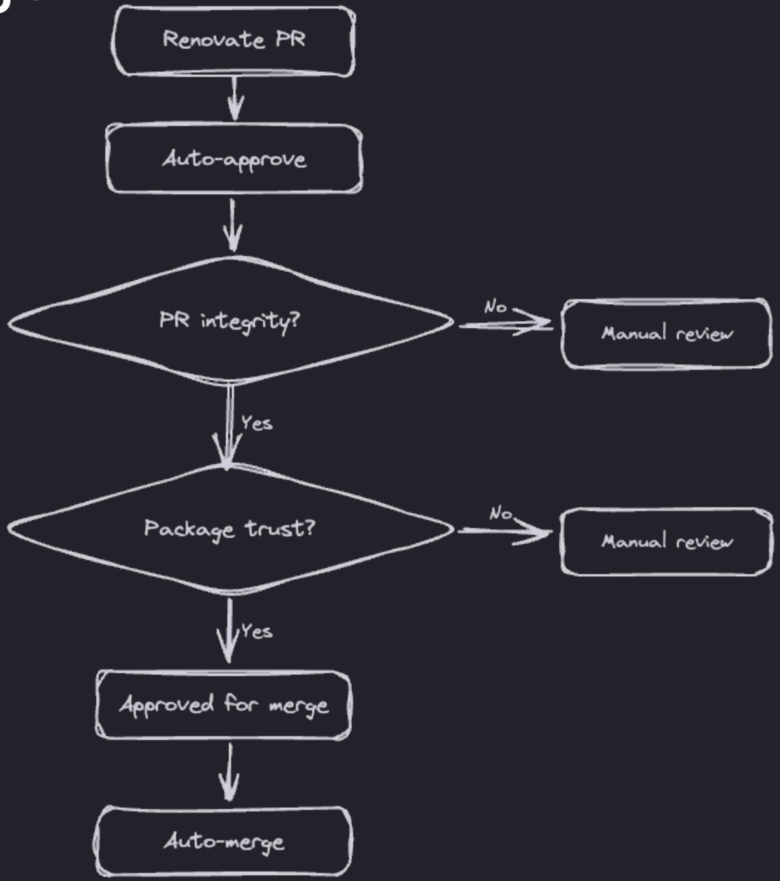
- Repositories discovered via regex filter patterns, not a manual list
- Patterns split alphabetically across batches
- Ignored repos excluded via negative patterns
- New repos picked up automatically when they match a pattern

Webhook Listener



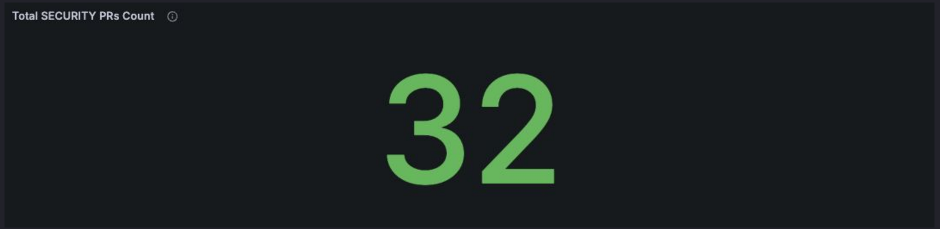
Automating PR approval & merge

- Auto-approve bot runs explicit checks
- PR integrity: is it a genuine Renovate PR?
- Package trust: is it allow-listed, the expected level, sufficiently aged?
- Auto-merge bot handles the rest once verified



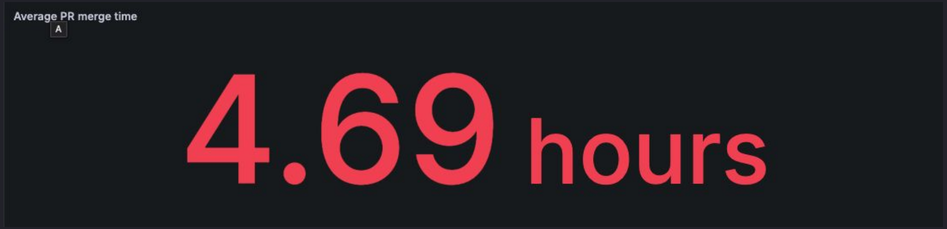
Infinity data source

- Queries GitHub API directly
- Parses and transforms JSON responses into dashboard metrics
- Total Security PRs count
- Average PR merge times



Total SECURITY PRs Count

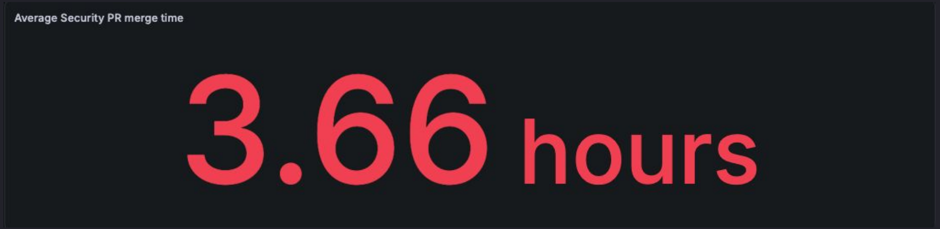
32



Average PR merge time

A

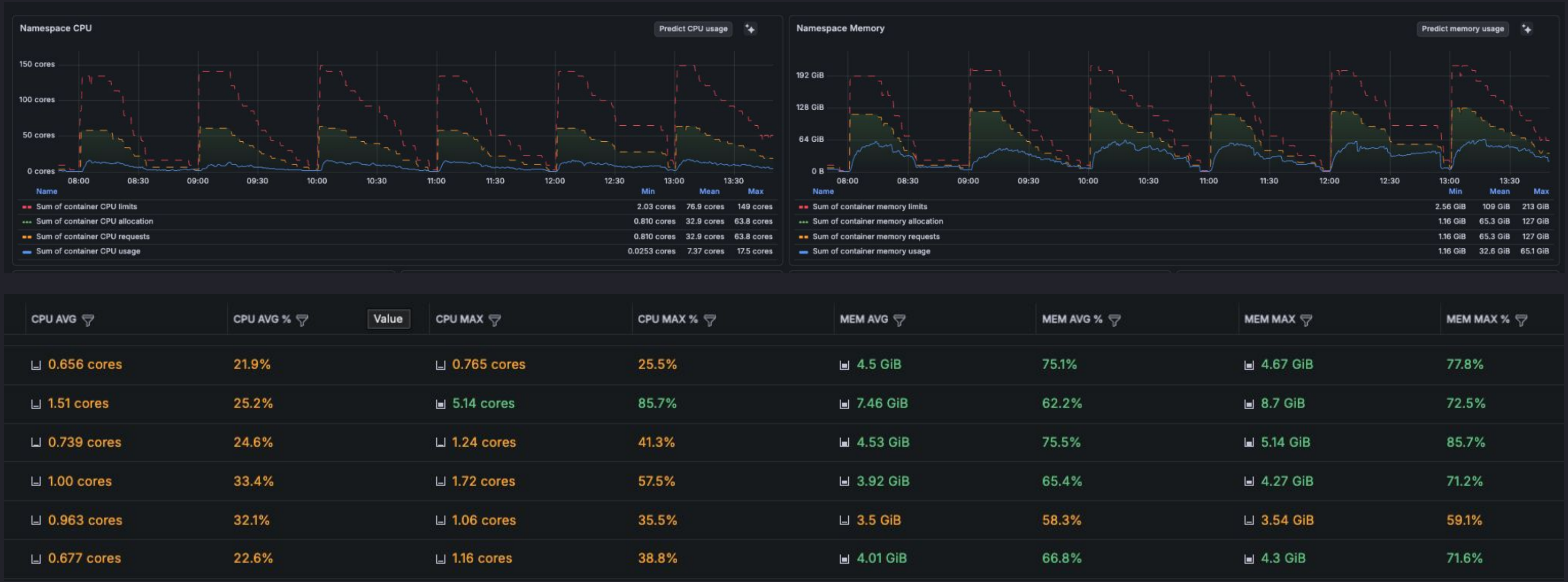
4.69 hours



Average Security PR merge time

3.66 hours

K8s-monitoring



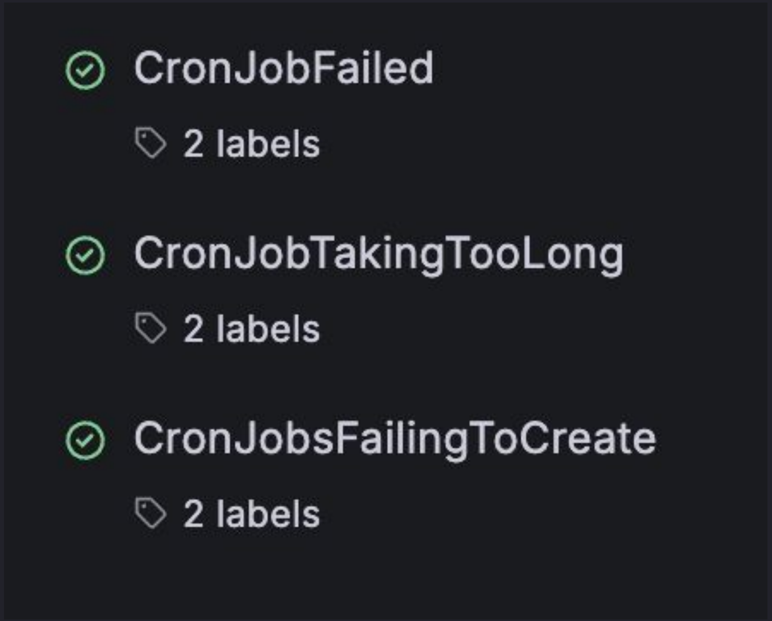
Name	Min	Mean	Max
Sum of container CPU limits	2.03 cores	76.9 cores	149 cores
Sum of container CPU allocation	0.810 cores	32.9 cores	63.8 cores
Sum of container CPU requests	0.810 cores	32.9 cores	63.8 cores
Sum of container CPU usage	0.0253 cores	7.37 cores	17.5 cores

Name	Min	Mean	Max
Sum of container memory limits	2.56 GiB	109 GiB	213 GiB
Sum of container memory allocation	1.16 GiB	65.3 GiB	127 GiB
Sum of container memory requests	1.16 GiB	65.3 GiB	127 GiB
Sum of container memory usage	1.16 GiB	32.6 GiB	65.1 GiB

CPU AVG	CPU AVG %	CPU MAX	CPU MAX %	MEM AVG	MEM AVG %	MEM MAX	MEM MAX %
0.656 cores	21.9%	0.765 cores	25.5%	4.5 GiB	75.1%	4.67 GiB	77.8%
1.51 cores	25.2%	5.14 cores	85.7%	7.46 GiB	62.2%	8.7 GiB	72.5%
0.739 cores	24.6%	1.24 cores	41.3%	4.53 GiB	75.5%	5.14 GiB	85.7%
1.00 cores	33.4%	1.72 cores	57.5%	3.92 GiB	65.4%	4.27 GiB	71.2%
0.963 cores	32.1%	1.06 cores	35.5%	3.5 GiB	58.3%	3.54 GiB	59.1%
0.677 cores	22.6%	1.16 cores	38.8%	4.01 GiB	66.8%	4.3 GiB	71.6%

Alerting

- Job failures
- Long-running jobs
- Failure to create



✓ CronJobFailed

2 labels

✓ CronJobTakingTooLong

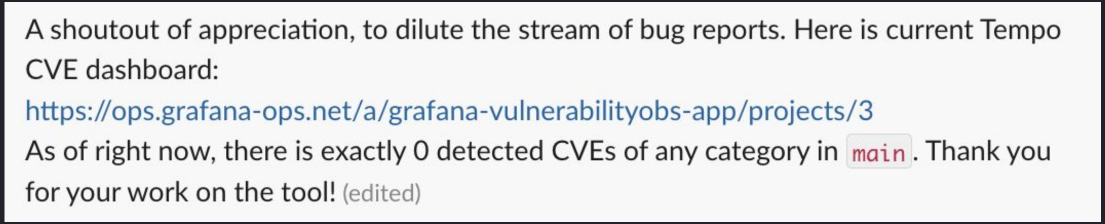
2 labels

✓ CronJobsFailingToCreate

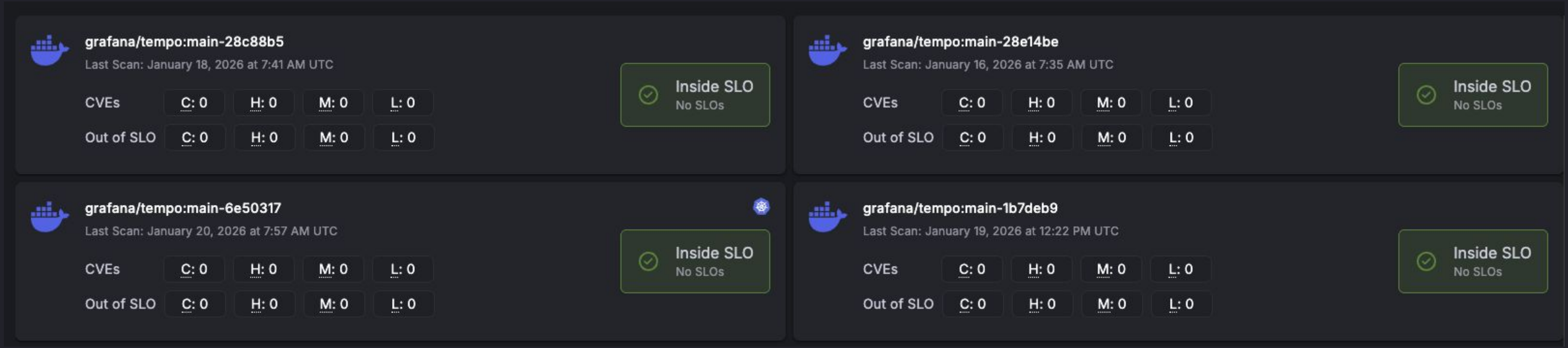
2 labels

Demo

Feedback



A shoutout of appreciation, to dilute the stream of bug reports. Here is current Tempo CVE dashboard:
https://ops.grafana-ops.net/a/grafana-vulnerabilityobs-app/projects/3
As of right now, there is exactly 0 detected CVEs of any category in `main`. Thank you for your work on the tool! (edited)



grafana/tempo:main-28c88b5
Last Scan: January 18, 2026 at 7:41 AM UTC

CVEs	C: 0	H: 0	M: 0	L: 0
Out of SLO	C: 0	H: 0	M: 0	L: 0

Inside SLO
No SLOs

grafana/tempo:main-28e14be
Last Scan: January 16, 2026 at 7:35 AM UTC

CVEs	C: 0	H: 0	M: 0	L: 0
Out of SLO	C: 0	H: 0	M: 0	L: 0

Inside SLO
No SLOs

grafana/tempo:main-6e50317
Last Scan: January 20, 2026 at 7:57 AM UTC

CVEs	C: 0	H: 0	M: 0	L: 0
Out of SLO	C: 0	H: 0	M: 0	L: 0

Inside SLO
No SLOs

grafana/tempo:main-1b7deb9
Last Scan: January 19, 2026 at 12:22 PM UTC

CVEs	C: 0	H: 0	M: 0	L: 0
Out of SLO	C: 0	H: 0	M: 0	L: 0

Inside SLO
No SLOs

Dimitrios Sotirakis

Philip Hope

Thank you!