

komodor

 codefresh

Removing CI/CD Blockers

Navigating Kubernetes with Codefresh & Komodor



Your Hosts today



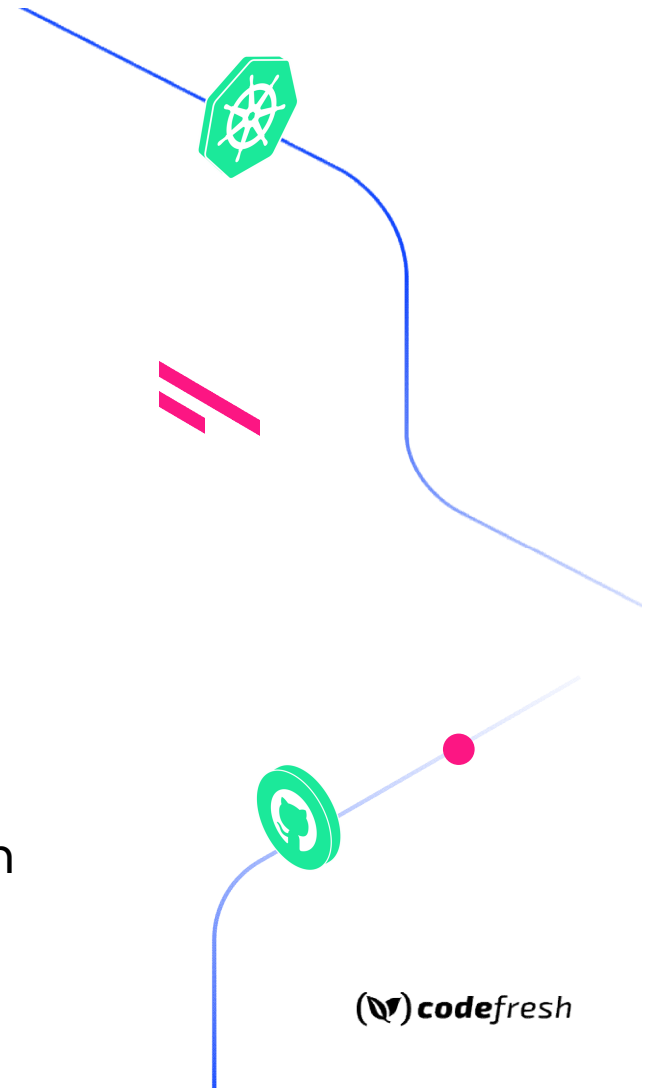
Itiel Schwartz

Co-founding CTO @Komodor



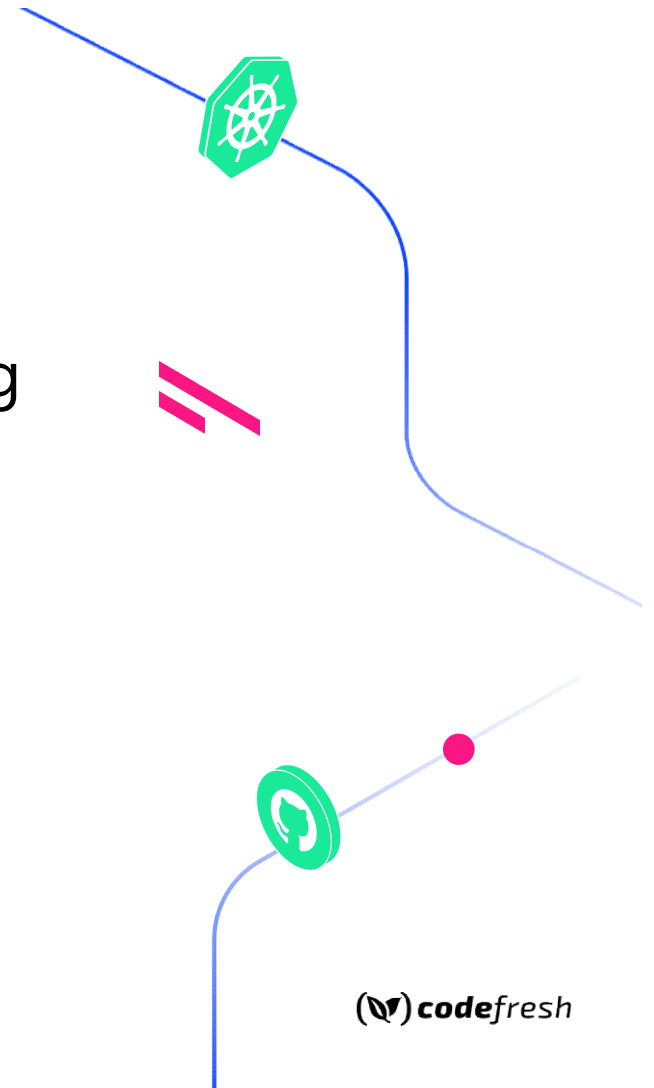
Kostis Kapelonis

Developer Advocate @Codefresh

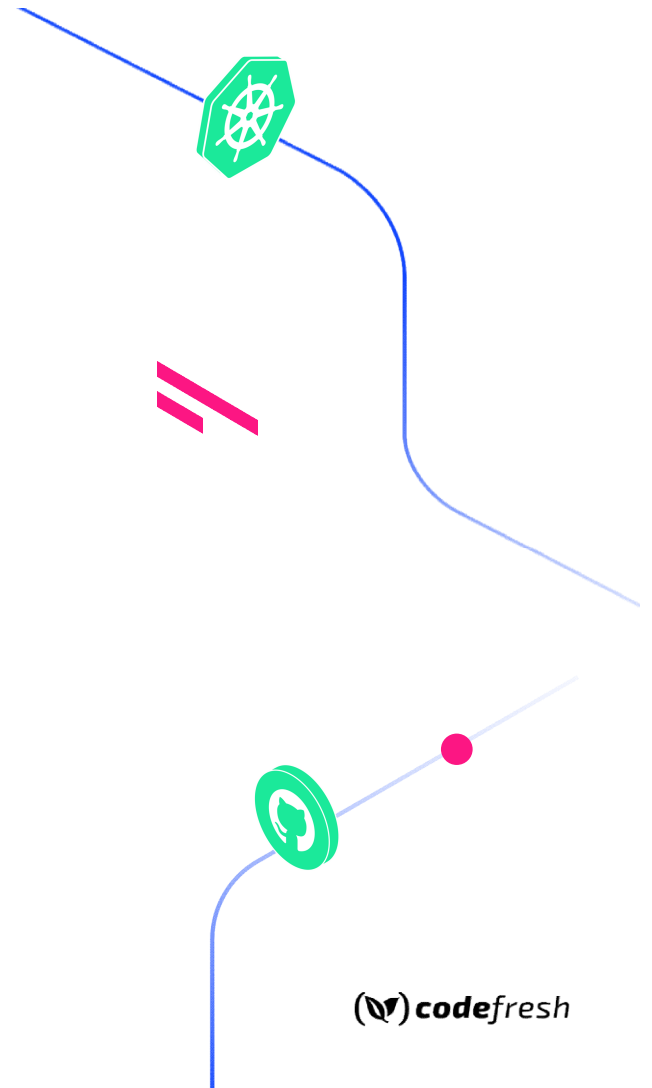


Agenda

1. The challenges of K8s troubleshooting
2. Why you should not abuse kubectl
3. How to supercharge your metrics
4. Demo
5. Discussion and Q/A



How it all started



 **codefresh**



komodor



DEVOPS

Troubleshooting Kubernetes Clusters as a Developer with Komodor

9 min read

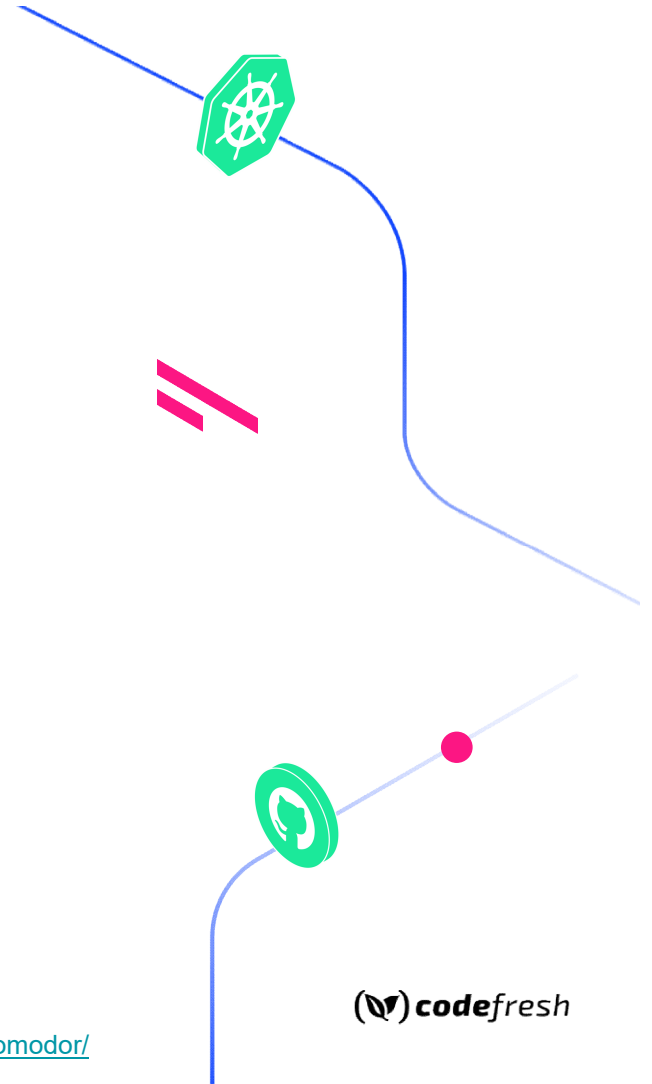


Kostis Kapelonis · May 13, 2021

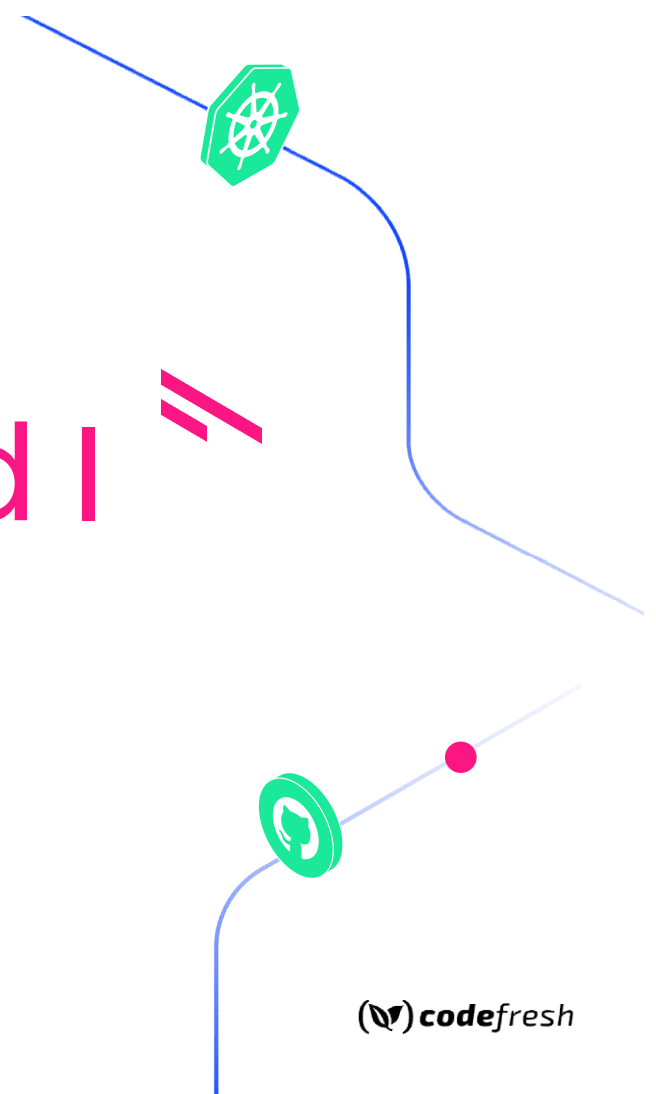
The container ecosystem is moving very fast and new tools designed specifically for Kubernetes clusters are introduced at a very fast pace. Even though several times a new tool is simply implementing a well-known mechanism (already present in the VM world) with a focus on containers, every once in a while we see tools that are designed from scratch rather than adapting a preexisting idea. One such tool is [Komodor](#).



<https://codefresh.io/devops/troubleshooting-kubernetes-with-komodor/>



Why should I
care?

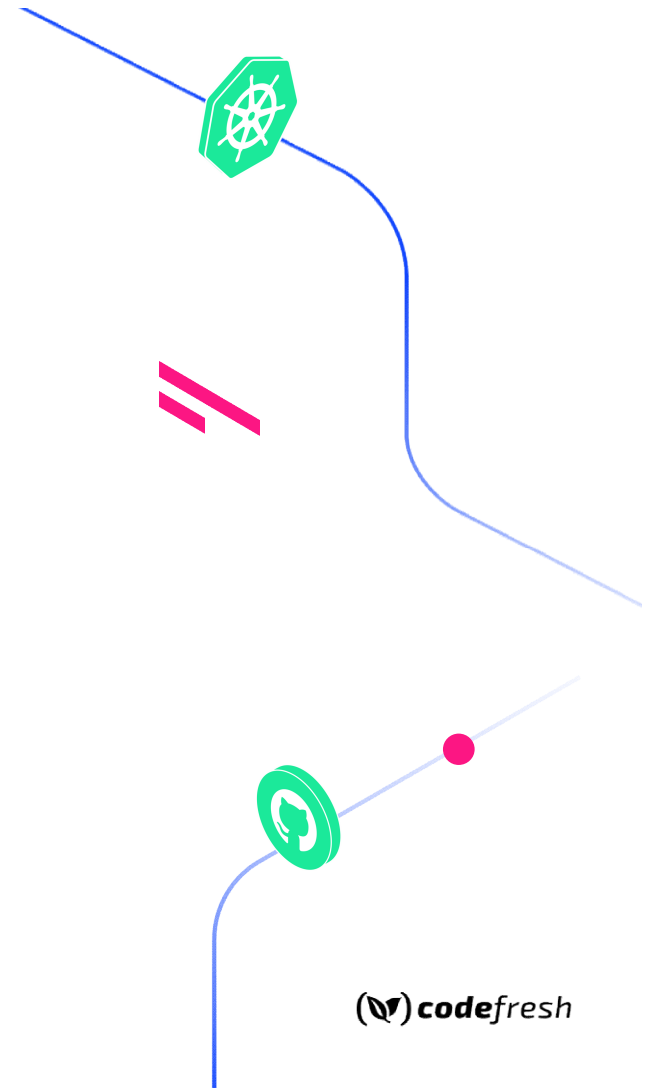


Kubernetes Everywhere

Lots of tools for:

- CI/CD
- Security scans
- Unit/Integration tests
- Monitoring

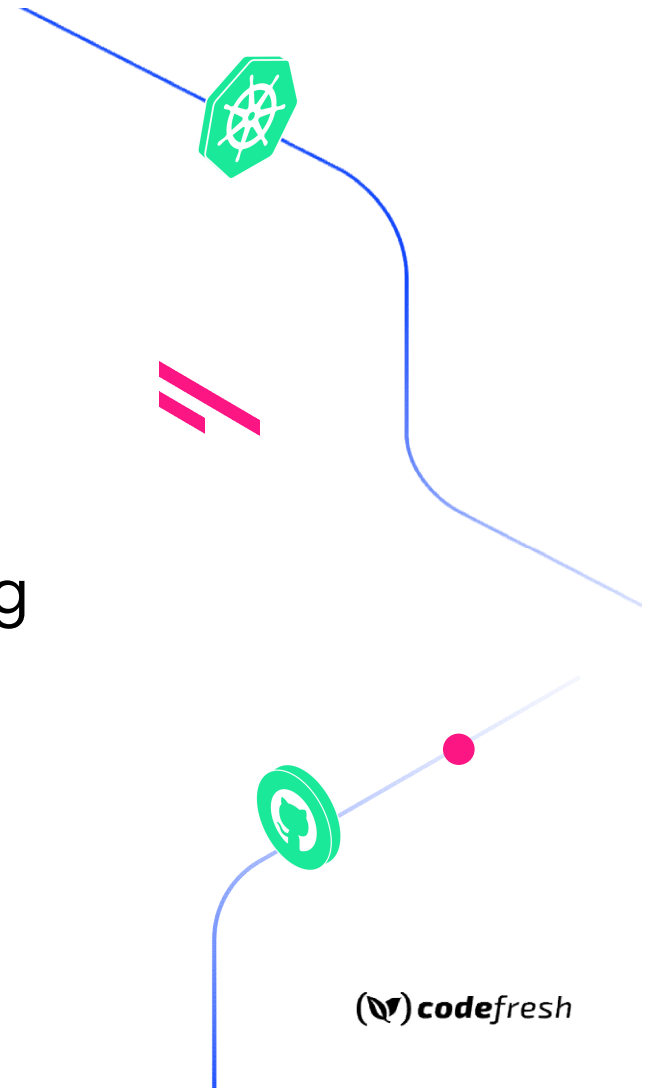
Zero tools for troubleshooting



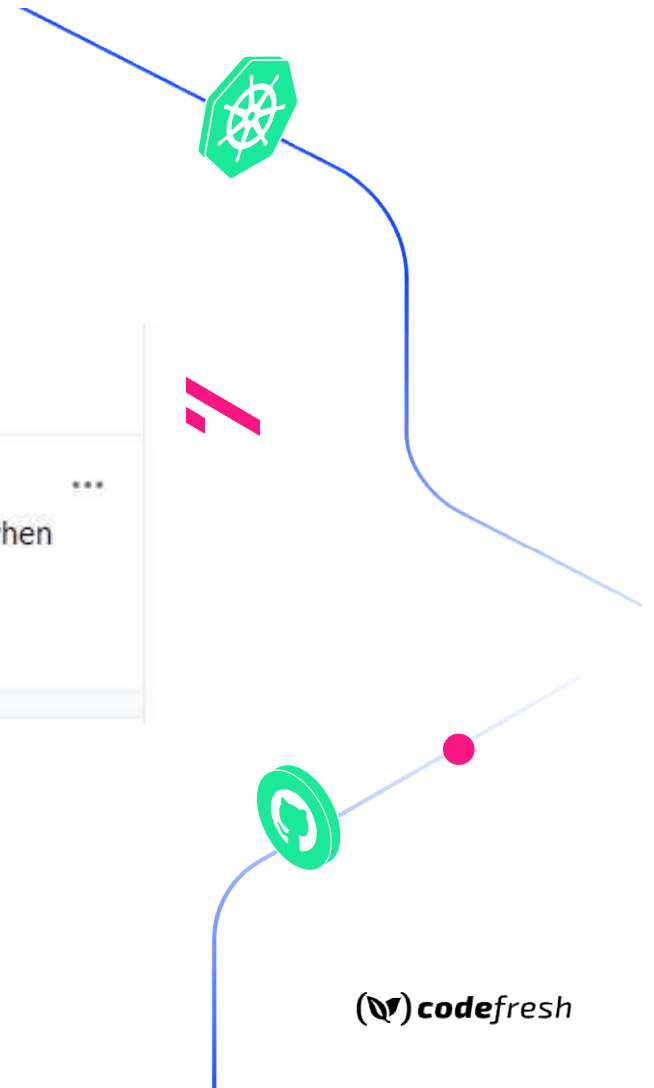
Repeat After Me

kubectl is **NOT** a troubleshooting tool

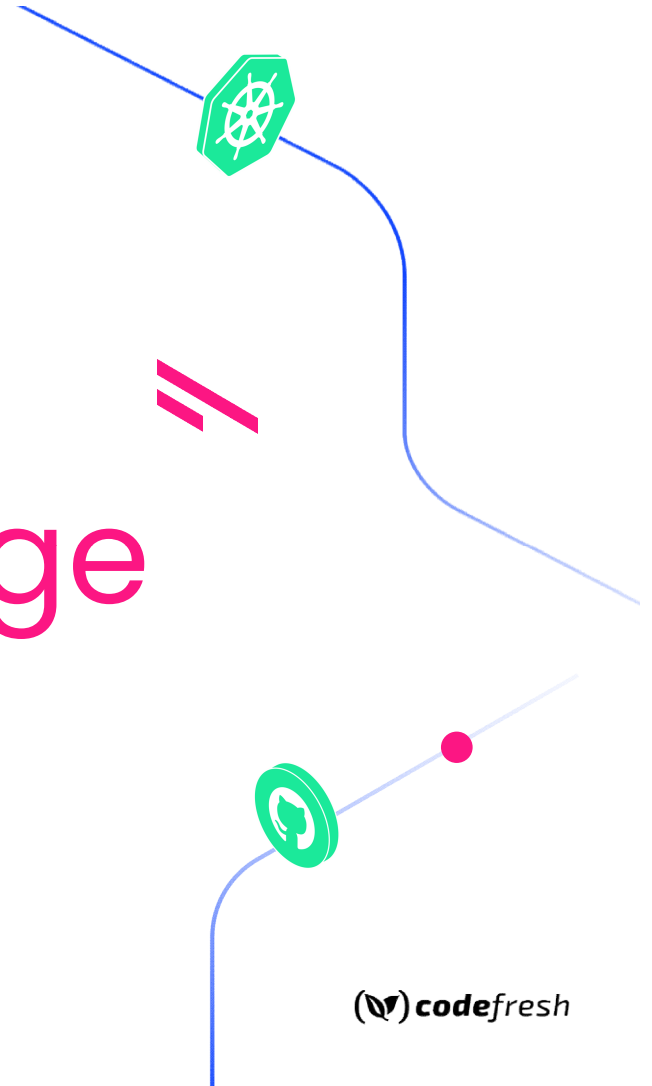
(especially when it is 3AM in the morning
and the money app is down)



Don't believe me?



The challenge

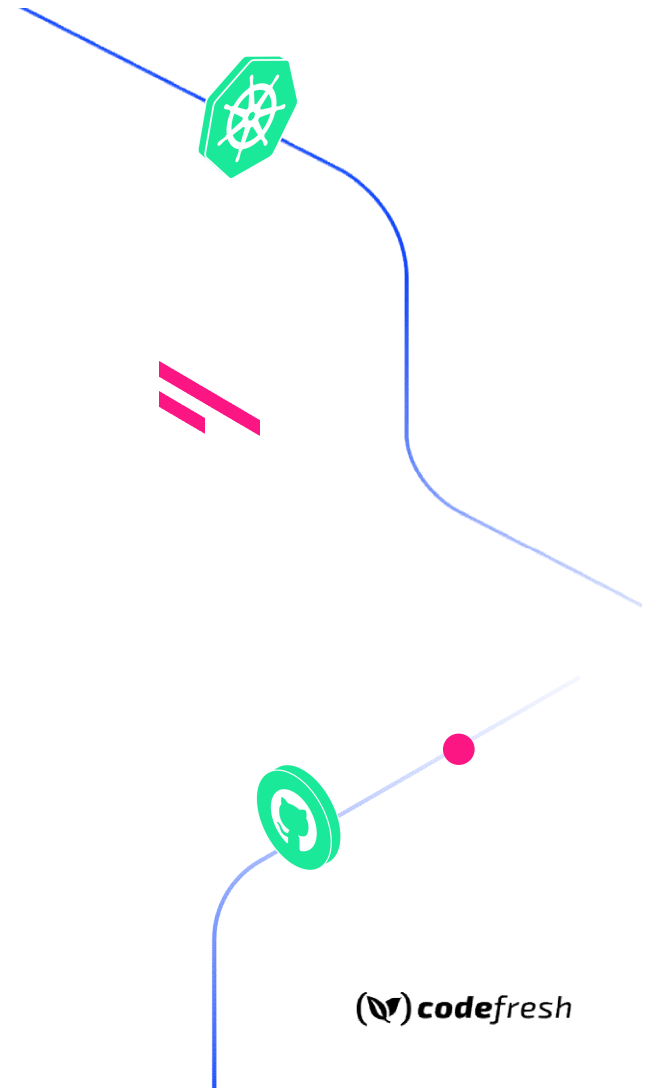


 **codefresh**

Common Scenarios

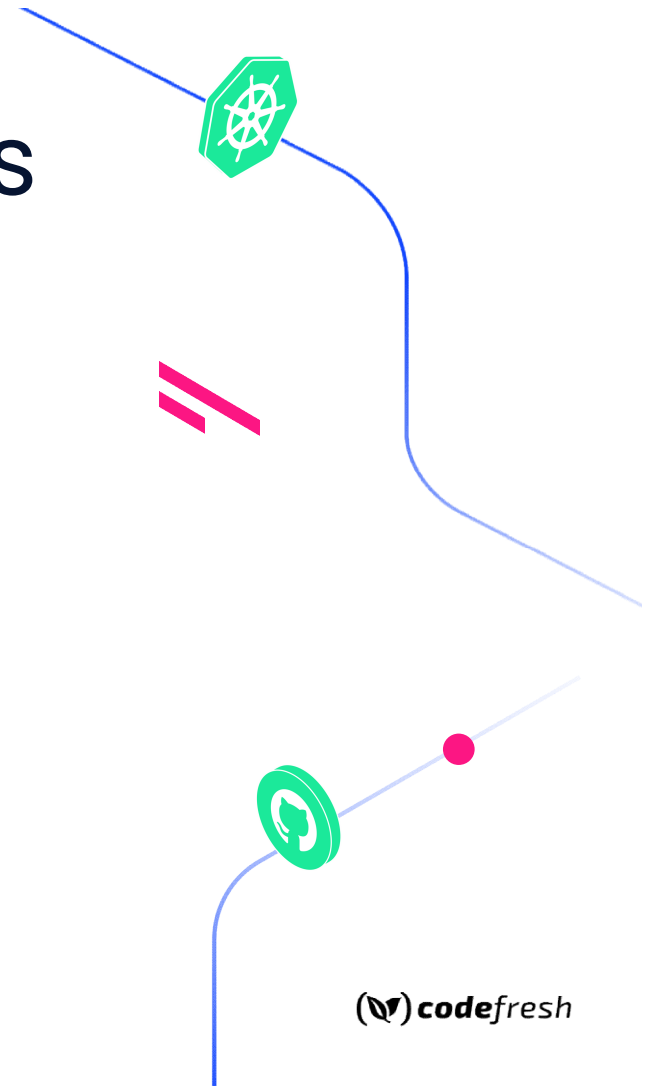
You are paged at 3am. The application is down:

1. On a cluster you are familiar with
2. On an unfamiliar cluster



The Important Questions

- What did we change in the cluster?
- Who made the last change?
- Where are the CI/CD pipelines?
- What about dependencies?



How to Spend Your Night

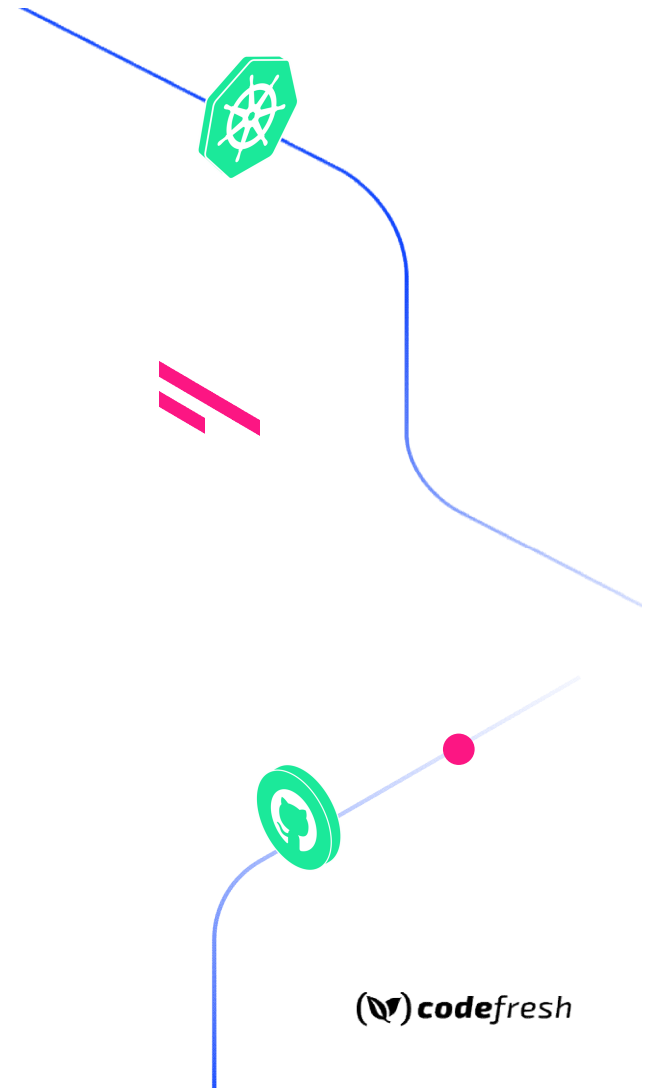


Host	Item	Severity	Resolution
Application	artprod01	SEV1	Check logs, restart the service
	artprod02	SEV1	Check logs, restart the service
	arttest01	SEV1	Delete rotated logs
	arttest02	SEV2	Check logs, restart the service
Database	mysql	SEV1	Check logs, restart the service
	disk_root	SEV1	Delete rotated logs
	disk_data	SEV1	See Maintenance and Cleanup section
			start the service

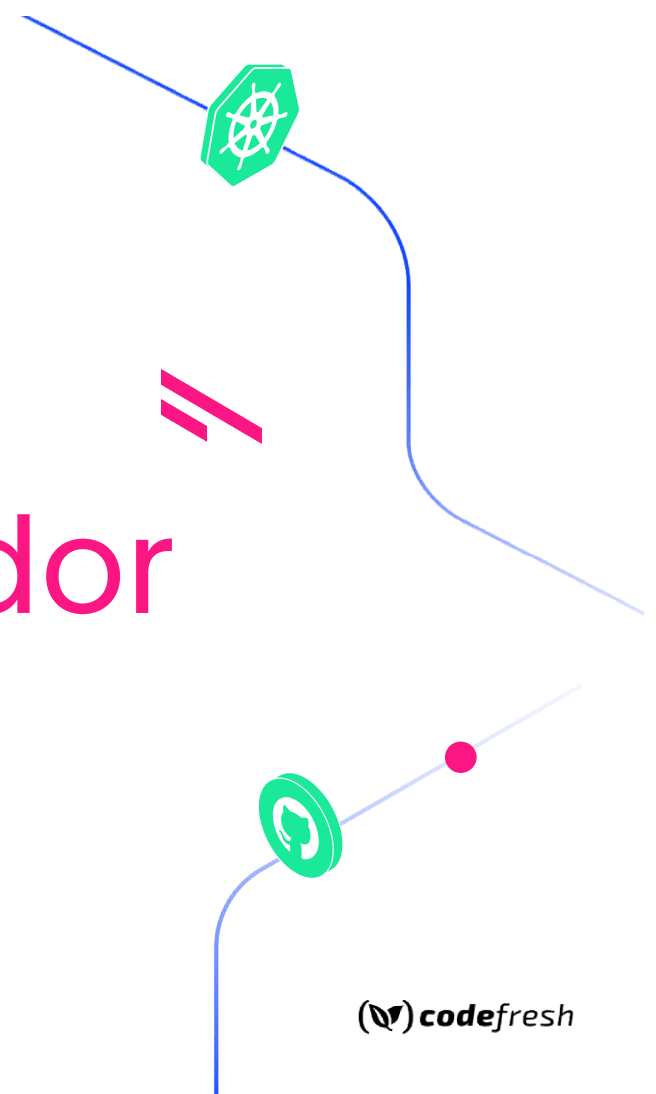


The Goal

1. Do we really need to visit 5+ different systems to understand the issue?
1. Can we avoid escalating & waking up other colleagues?



Enter Komodor



What is Komodor?



 **codefresh**

moneymaker-app

HEALTHY cluster: myawscluster namespace: komodor-demo image: loan:8af5919 replicas: 2/2

About this service

Pods status and Logs **LIVE**

Activity
Last deploy 2 days ago

K8s Metadata
Replicas 2/2
Service type Deployment

Annotations
deployment.kubernetes.io/revision: 8

Additional Links
grafana-overall-system-health

Events (31)

Filter events by: Event type, status, and details

Show events from: Aug 30, 2021, 12:22 PM → Aug 30, 2021, 2:04 PM (UTC...)

Event type	Summary	Start time (UTC+3)	Last updated time	Details	Service	Status
Health change	moneymaker-app service was unhealthy for 3 minutes	Aug 30, 2021, 1:30:55 PM	Aug 30, 2021, 1:34:03 PM	NonZeroExitCode (Exit code 2)	myawscluster/moneymaker-app	Healthy
Deploy	image updated to tag b62ebad	Aug 30, 2021, 1:33:07 PM	Aug 30, 2021, 1:33:27 PM	Changes: image, annotation	myawscluster/hidden-dependency	Completed
Health change	hidden-dependency service was unhealthy for 3 minutes	Aug 30, 2021, 1:30:18 PM	Aug 30, 2021, 1:33:27 PM	NonZeroExitCode (Exit code 2)	myawscluster/hidden-dependency	Healthy

Related services (2)

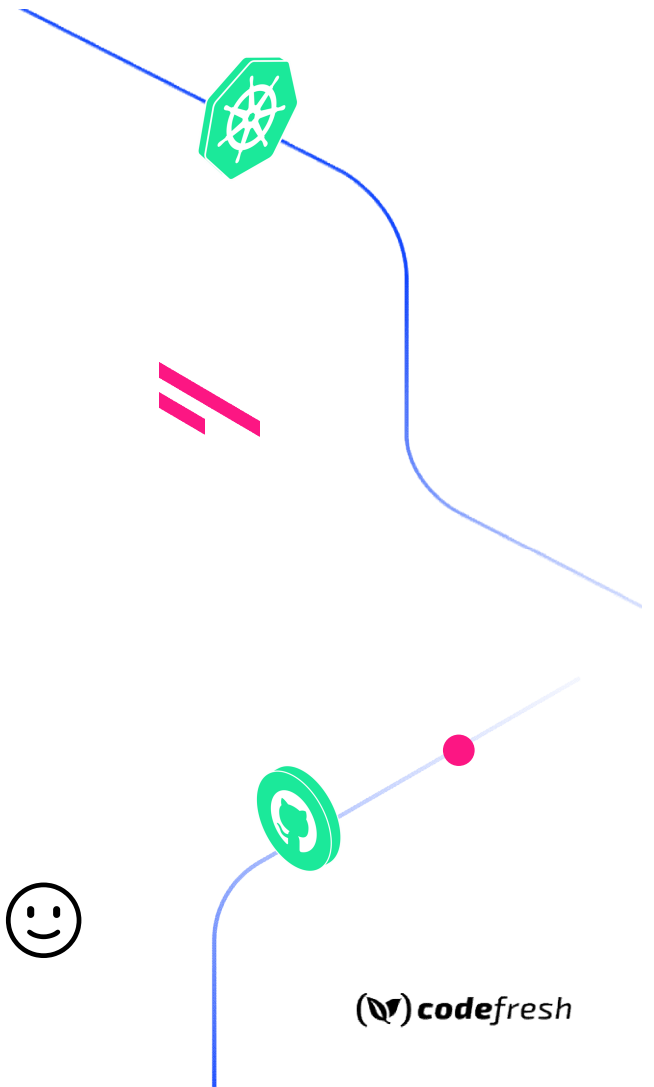
Show by: namespace: komodor-demo

- hidden-dependency**
HEALTHY
Modified 2 days ago
cluster: myawscluster
namespace: komodor-demo
- simple-deployment**
HEALTHY
Modified 2 days ago
cluster: myawscluster
namespace: komodor-demo

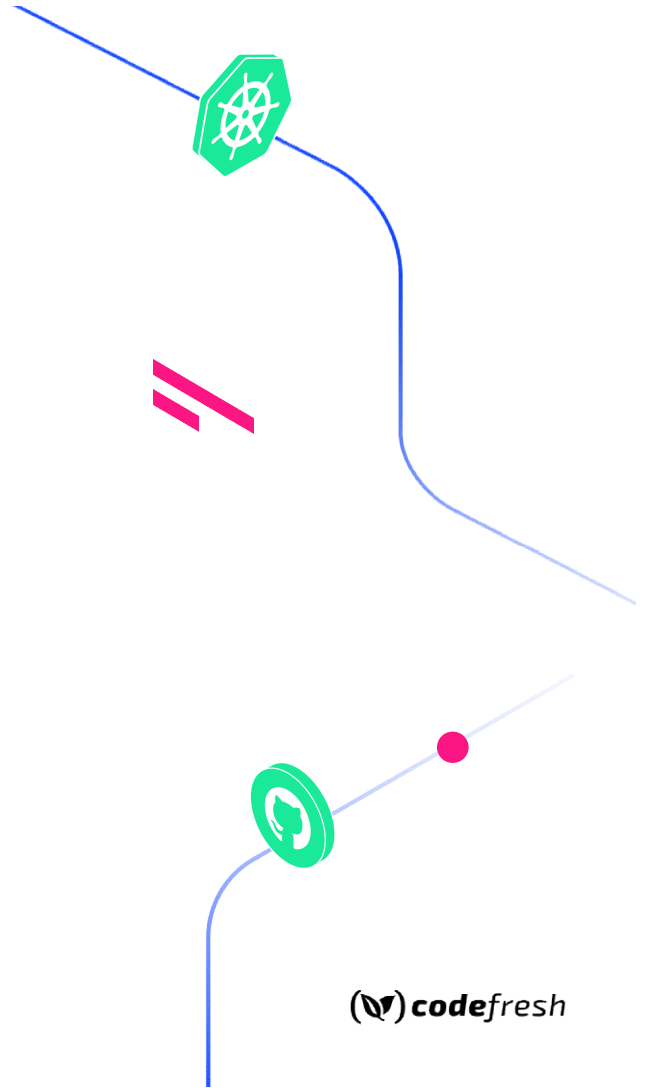


What is Komodor?

- Komodor is a troubleshooting tool
- It introduces a new family of tools
- It is not a metric solution
- It is not a deployment solution
- It is not a networking solution
- But it integrates with all the above 😊

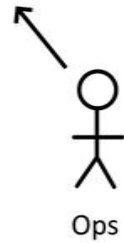
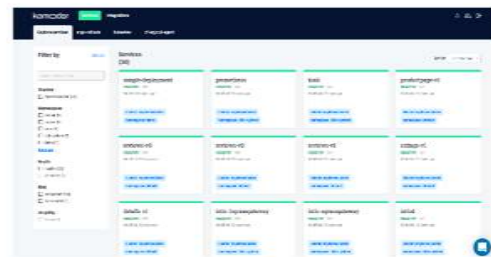


Before Komodor



(👉) **code**fresh

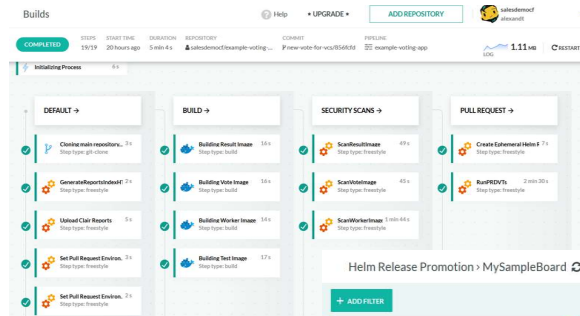
After Komodor



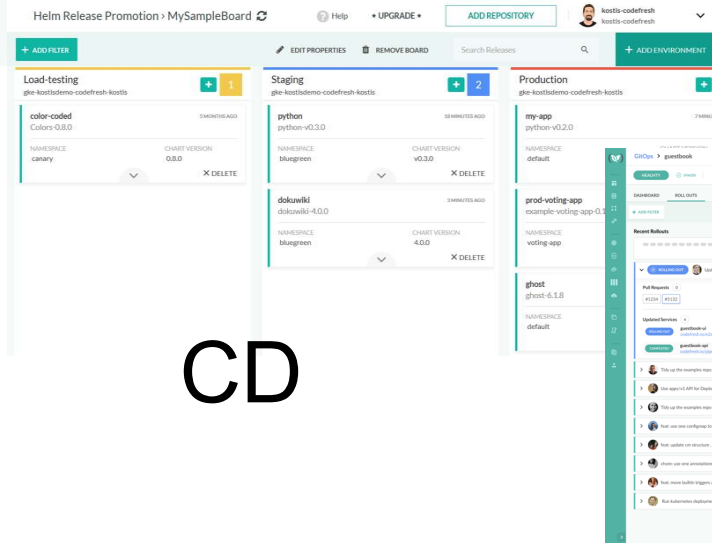
 **codefresh**



About Codefresh



CI



CD

GitOps



 codefresh

Demo

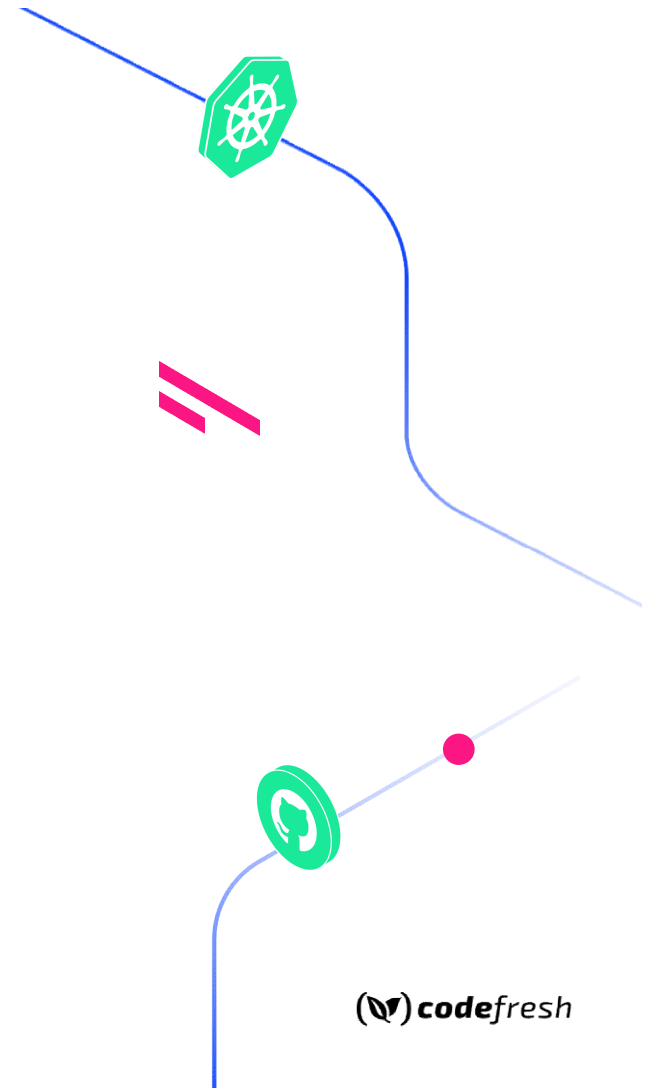


 **codefresh**



Komodor Deep Dive

- Service explorer
- Related services
- Events view
- Pod status & logs
- Setting up Komodor + Integrations



Service Explorer

We collect data from Kubernetes and enrich it with observability, code repository, CI/CD and alerting tools. The data is organized in a comprehensive way, ready for a drill down from the big picture to its details.

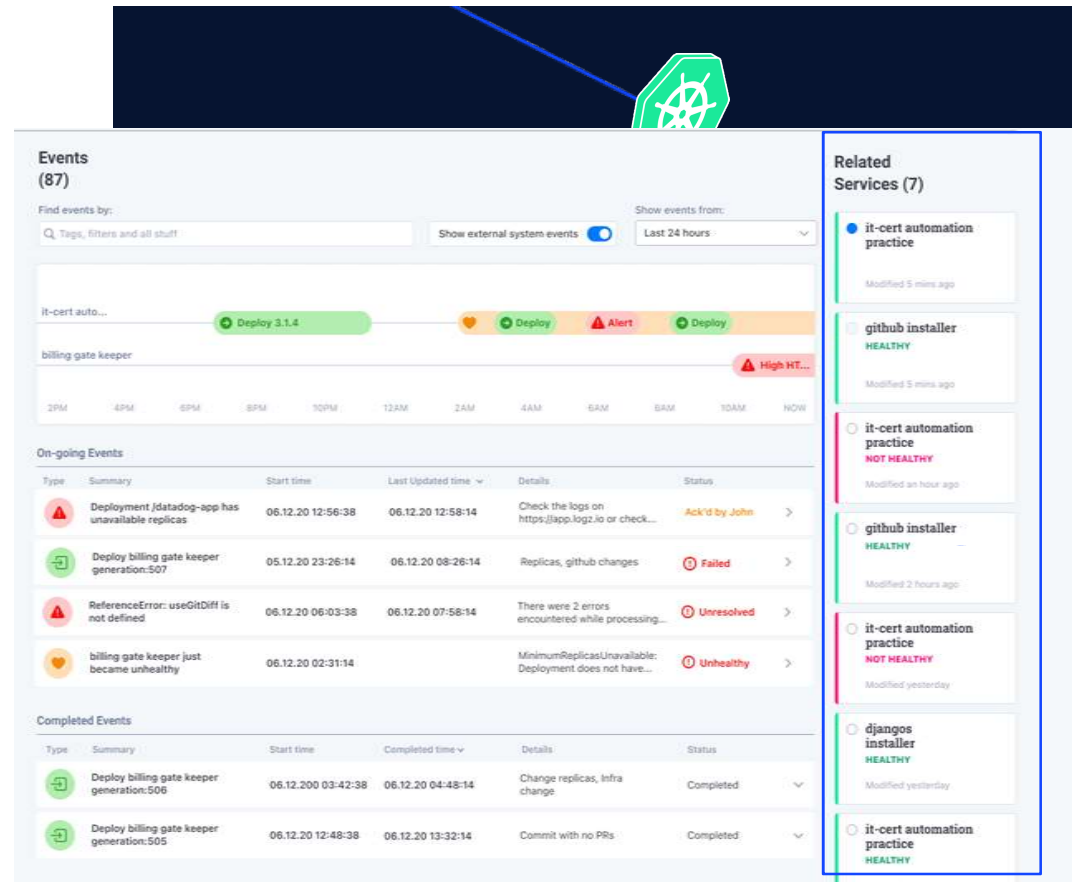
The screenshot displays the Komodor Service Explorer interface. At the top, there's a navigation bar with 'Services' highlighted. Below it, the current view is 'Billing Gate Keeper'. A 'Filter by' sidebar on the left allows filtering by Clusters, Namespaces, Health, and Ongoing. The main area shows a grid of 16 service cards, each with a health status (Healthy or Unhealthy), a version (e.g., 4/4), and a modification time. Each card also displays specific labels like 'TENANT:01', 'ENV:PROD', and 'ESE_CREATION_DATE: 1601457418'. The interface is clean and modern, with a dark blue header and a light gray main content area.



Related Services

Troubleshooting microservices requires a deep understanding of connections and dependencies.

In one click, you can add more services to the service view, so it's correlated on one timeline.



The screenshot displays a monitoring dashboard interface. At the top, there's a search bar for events and a filter for 'Show external system events'. Below this is a timeline view showing events for 'it-cert auto...' and 'billing gate keeper'. The 'it-cert auto...' timeline shows a 'Deploy 3.1.4' event, followed by a 'Deploy' event, an 'Alert' event, and another 'Deploy' event. The 'billing gate keeper' timeline shows a 'High HT...' event.

The 'On-going Events' section contains a table with the following data:

Type	Summary	Start time	Last updated time	Details	Status
Alert	Deployment /datadog-app has unavailable replicas	06.12.20 12:56:38	06.12.20 12:58:14	Check the logs on https://app.logz.io or check...	Ack'd by John
Deployment	Deploy billing gate keeper generation:507	05.12.20 23:26:14	06.12.20 08:26:14	Replicas, github changes	Failed
Alert	ReferenceError: useGitDiff is not defined	06.12.20 06:03:38	06.12.20 07:58:14	There were 2 errors encountered while processing...	Unresolved
Alert	billing gate keeper just became unhealthy	06.12.20 02:31:14		MinimumReplicasUnavailable: Deployment does not have...	Unhealthy

The 'Completed Events' section contains a table with the following data:

Type	Summary	Start time	Completed time	Details	Status
Deployment	Deploy billing gate keeper generation:506	06.12.20 03:42:38	06.12.20 04:48:14	Change replicas, Infra change	Completed
Deployment	Deploy billing gate keeper generation:505	06.12.20 12:48:38	06.12.20 13:32:14	Commit with no PRs	Completed

The 'Related Services (7)' section on the right lists several services:

- it-cert automation practice (Modified 5 mins ago)
- github installer (HEALTHY, Modified 5 mins ago)
- it-cert automation practice (NOT HEALTHY, Modified an hour ago)
- github installer (HEALTHY, Modified 2 hours ago)
- it-cert automation practice (NOT HEALTHY, Modified yesterday)
- django installer (HEALTHY, Modified yesterday)
- it-cert automation practice (HEALTHY)



Events View

The 'Events' feature offers a panoramic view of all occurrences across your entire K8s environment.

With this system-wide visibility, Komodor Events makes it easier to troubleshoot elusive issues, particularly those that aren't traced to any one specific service or cluster.



The screenshot shows the Komodor Events view interface. The navigation bar includes 'Services', 'Events (122)', 'Notifications', and 'Integrations'. The 'Events' section is active, showing a search bar and a 'Show events from' range set to '05/25/2021, 6:19 PM' to '05/25/2021, 6:19 PM'. The interface is divided into a left sidebar and a main content area.

Saved Filters:

- Brain file changes
- Outdated production issues
- Default code changes
- File changes on default cluster
- Brain
- Brain production

Filter events: [Reset filter](#) [Clear all](#)

Clusters: main (112)

Namespaces: default (112)

Services:

- admin (2)
- brain-consumer-notifications (4)
- brain-consumer-pg (4)
- brain-indicator-pg (2)
- consumer-dq (2)

See more

Types:

- Outdated Monitor Alert (2)
- Deploy (101)
- External Status (16)
- Health change (6)

Deploy details: Git changes (94)

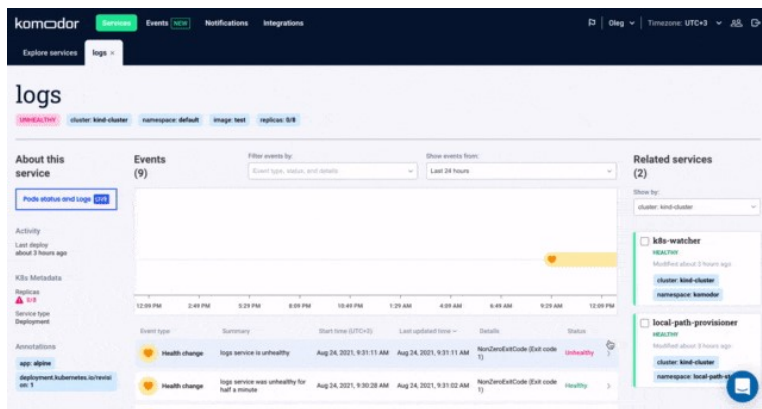
Found 122 events

The chart shows event frequency over time, with peaks around 24:00 and 25:00. The table below lists the events:

Event type	Summary	Start time	Last updated time	Details	Service	Status
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:22 PM	5/25/2021, 6:19:29 PM	Changes env, image	main/brain-events-collector	Completed
Deploy	no image change	5/25/2021, 6:18:22 PM	5/25/2021, 6:19:27 PM	Changes image	main/brain-server	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:23 PM	5/25/2021, 6:18:44 PM	Changes env, image	main/test-api	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:22 PM	5/25/2021, 6:18:37 PM	Changes env, image	main/brain-events-core	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:19 PM	5/25/2021, 6:18:37 PM	Changes env, image	main/brain-pg	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:23 PM	5/25/2021, 6:18:36 PM	Changes image	main/test	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:24 PM	5/25/2021, 6:18:35 PM	Changes image	main/brainbook-frontend	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:23 PM	5/25/2021, 6:18:35 PM	Changes env, image	main/brain-installer	Completed
Deploy	Image updated to tag 1.1234.962	5/25/2021, 6:18:23 PM	5/25/2021, 6:18:29 PM	Changes image	main/brain-message-sender	Completed



Pod Status and logs

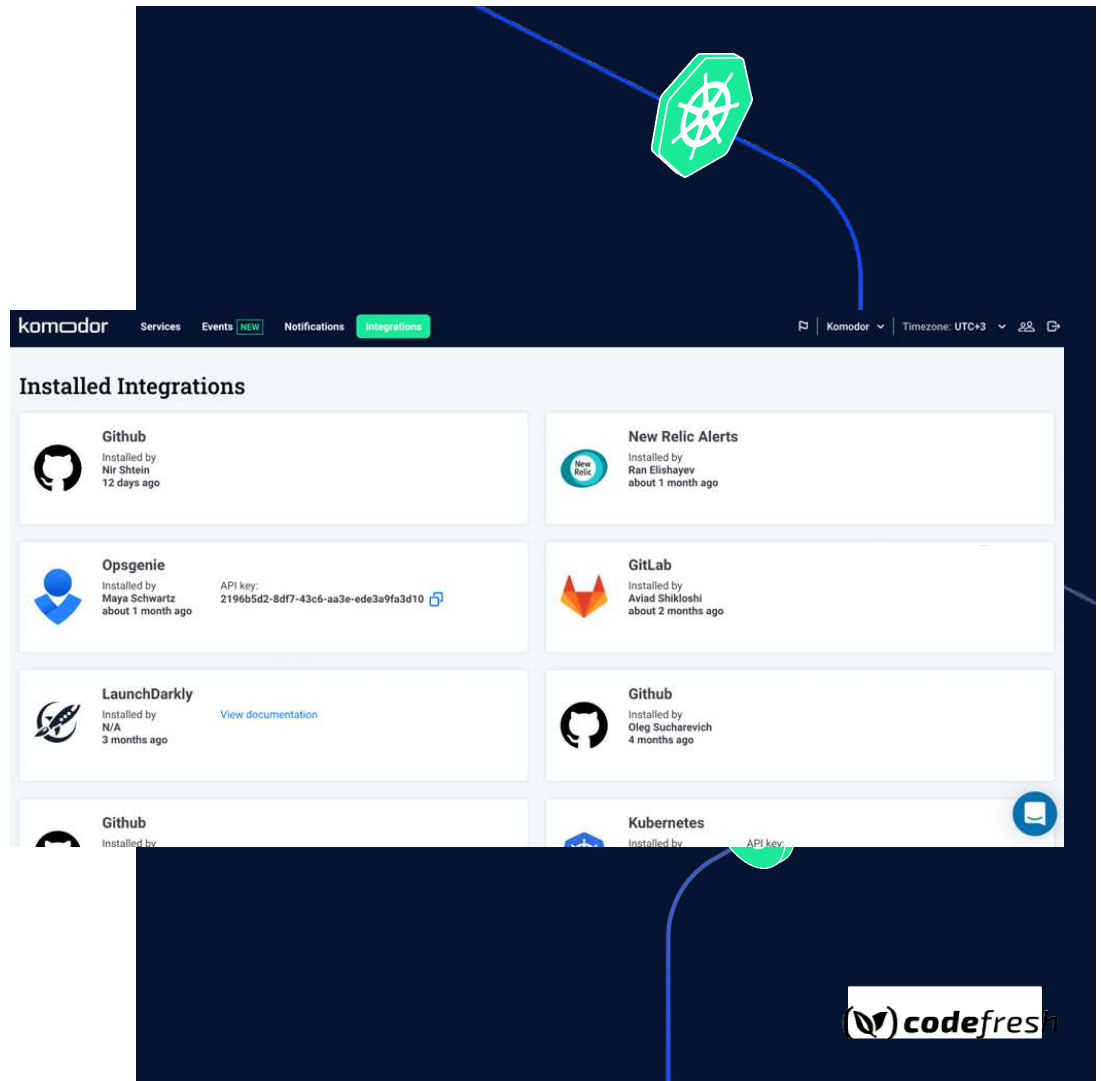


'Pods Status and Logs' enables you to quickly drill down in the pods of an unhealthy service. This offers quick access to all of the pod-level data you'll need for troubleshooting, including:

- Overview of all pods running the service
- Pod details, similar to what you would get with `kubectl describe`
- Live view of all events
- Pod containers' logs

Installation and integration

- Komodor takes about 5 minutes to install.
- K8s agent documentation can be found here:
<https://github.com/komodorio/helm-charts/tree/master/charts/k8s-watcher>
- Komodor integrates with all of your favorite DevOps tools



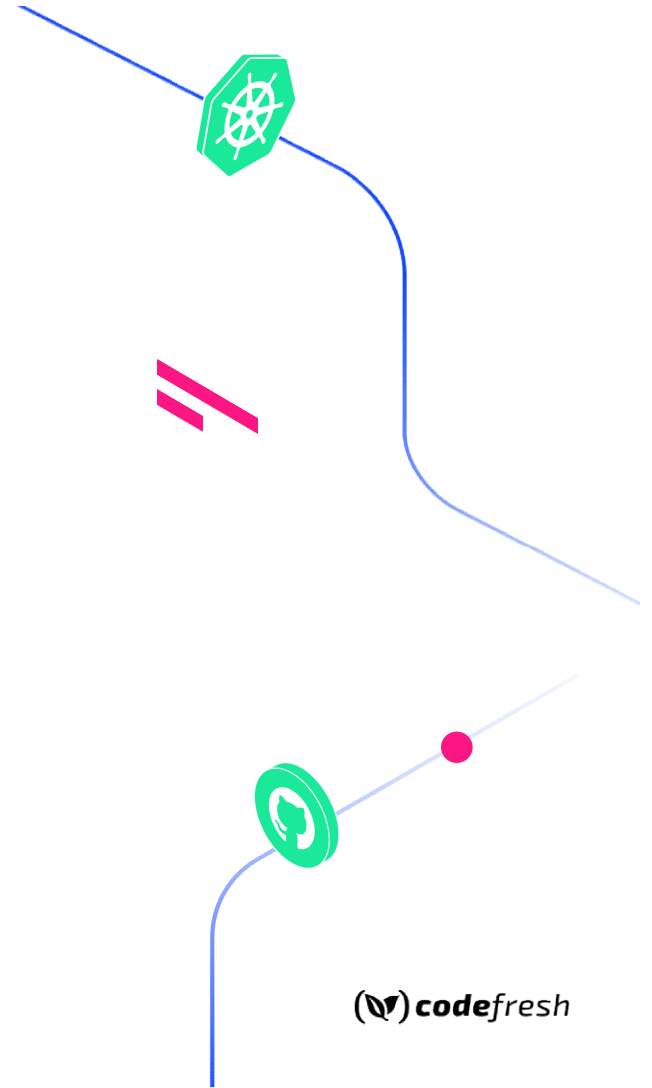
The screenshot displays the Komodor web interface. At the top, there is a navigation bar with the Komodor logo and menu items: Services, Events (with a 'NEW' badge), Notifications, and Integrations (highlighted in green). The main content area is titled 'Installed Integrations' and lists several connected services in a grid:

- Github**: Installed by Nir Shtein, 12 days ago.
- New Relic Alerts**: Installed by Ran Elshayev, about 1 month ago.
- Opsgenie**: Installed by Maya Schwartz, about 1 month ago. Includes an API key: 2196b5d2-8df7-43c6-aa3e-ede3a9fa3d10.
- GitLab**: Installed by Aviad Shikloshi, about 2 months ago.
- LaunchDarkly**: Installed by N/A, 3 months ago. Includes a 'View documentation' link.
- Github**: Installed by Oleg Sucharevich, 4 months ago.
- Github**: Partially visible at the bottom left.
- Kubernetes**: Installed by [redacted], includes an API key field.

The interface also shows a user profile 'Komodor' and a timezone setting of 'UTC+3'. A 'codefresh' logo is visible in the bottom right corner of the screenshot.



Q&A



Thank you!

Komodor sign up

<https://komodor.com/sign-up/>

Codefresh sign up

<https://codefresh.io/codefresh-signup/>

