Deploy Fast, Without Breaking Things: Level Up APIOps With OpenTelemetry

Adnan Rahić, Sonja Chevre
Speakers

Adnan Rahić
Senior Developer
Advocate
Tracetest

Sonja Chevre
Group Product Manager
Tyk Technologies
WHAT'S TO COME

Key points

01 What is APIOps?
02 APIOps in Kubernetes with ArgoCD and Tyk for fast flow
03 Getting Feedback from Production with OpenTelemetry and Jaeger
04 Getting Feedback earlier with OpenTelemetry and Tracetest
What is APIOps?
Traditional API Management makes us slow.

1. Manual tasks causes frequent errors
2. New API versions disrupt existing integrations
3. Loose controls expose us to security vulnerabilities
4. Scaling problems emerge with increased API usage
5. It takes weeks (or months) for developers to get feedback after committing their changes
APIOps is the offspring of DevOps and API management.

APIOps is about achieving speed and quality throughout the API management lifecycle - delivering value fast without disrupting your users.
Applying the DevOps principles to API Management:

- Fast flow
- Feedback loops
- Culture of learning

Observability as a key enabler for APIOps.

Three Ways of DevOps by Gene Kim
APIOps in Kubernetes with ArgoCD and Tyk for fast flow
Deploying an API with APIOps Practices

New feature or bug fix committed to Git repo

Developer

API source code repository

Trigger CI pipeline

Test

Build

Push to docker repo

CI pipeline
Argo CD

GitOps continuous delivery tool for Kubernetes
Deploying an API with APIOps Practices

New feature or bug fix committed to Git repo

DevOps

Deployment configuration repository

Track changes and sync environment

ArgoCD

Test
Build
Push to docker repo
Update K8s manifest file

CI pipeline

API source code repository

Trigger CI pipeline

Developer
Tyk API Gateway

Cloud-native and open source API Gateway for REST, GraphQL, TCP and gRPC protocols

The internet

API.EXAMPLE.COM

API.EXAMPLE.COM/WIDGETS

PARTNERS.EXAMPLE.COM

Your clouds and infrastructure

Tyk API Gateway

Authentication/Authorization
Protocol Mediation & Transforms
Custom Plugins & Functions
Analytics & Tracing
Versioning & Lifecycle Management
Caching
Endpoint Protection
Logging

HTTP

GraphQL

TCP

gRPC
apiVersion: tyk.tyk.io/v1alpha1
kind: ApiDefinition
metadata:
  name: httpbin
spec:
  name: httpbin
  use_keyless: true
  protocol: http
  active: true
  detailed_tracing: true
  proxy:
    target_url:
      http://go-httpbin-service.go-httpbin.svc.cluster.local
    listen_path: /httpbin
    strip_listen_path: true
Deploying an API with APIOps Practices

New feature or bug fix committed to Git repo

Trigger CI pipeline

API source code repository

Test
Build
Push to docker repo
Update K8s manifest file

CI pipeline

Deployment configuration repository

Track changes and sync environment

ArgoCD

DevOps
ArgoCD

Deployment configuration repository

Tyk Operator will update API definitions within the API Gateway.
WHAT WE HAVE ACHIEVED

Automation for Fast flow

- Preventing Configuration Drift
- Enhanced Security
- Improved Reliability
- Efficiency in Operations
- Better Audit Trails
- Collaboration and Visibility

But that’s not enough...
Getting Feedback from Production with OpenTelemetry and Jaeger
Without Observability we don’t know if our users are happy
Very unhappy users
See it all with **OpenTelemetry** and **Jaeger**

**OpenTelemetry**: Open standard to generate, collect, and export telemetry from API requests.

**Jaeger**: Back-end to store telemetry for monitoring and troubleshooting.
Configure OpenTelemetry in Tyk

Enable OpenTelemetry and configure OpenTelemetry Collector as endpoint

- tyk-gateway.gateway.opentelemetry.enabled: true
- tyk-gateway.gateway.opentelemetry.endpoint: opentelemetry-collector.observability.svc:4317
- tyk-gateway.gateway.opentelemetry.exporter: grpc
Distributed trace for every API request.

Authentication error
Distributed trace for every API requests

Most of the time is spent in the upstream service

API request is taking 3 s
WHAT WE HAVE ACHIEVED

Feedback from

API Platform
WHAT WE HAVE ACHIEVED

Feedback from Production

- No longer relying on users reporting failures
- Understand API performance
- Pinpoint the root cause of issues and outages
- Solving issues faster

But that’s not enough...
Stopping bugs before they impact our users
04 Getting Feedback earlier with OpenTelemetry and Tracetest
Without Trace-based testing we don’t know if releases are reliable
Validate Auth, API Gateways, Ingresses, Databases, Queues, ...

Create test specs against trace data

- Validate the Auth.
  - 1 span

- Validate all HTTP requests return status code 200.
  - 1 span

- Validate all gRPC requests return status OK.
  - 0 spans

- Validate all GraphQL requests return status OK.
  - 0 spans
Validate GraphQL APIs
Performance testing API responses, req/res size, rate limiting...
Shift Left with Tracetest!

Tracetest: Leverage existing OpenTelemetry data for deep API and integration tests.
Tracetest integration tests run every time there is a sync.
Overview of trace-based tests
WHAT WE HAVE ACHIEVED

Validating releases in Pre-Production

Diagram:
- Tracetest
- Tyk API Gateway
- OpenTelemetry Collector
- Observability tools
- Deploy APIs to staging
- API developer

Diagram shows the integration of Tracetest with the API platform, including Tyk API Gateway, OpenTelemetry Collector, and observability tools. The diagram also indicates the deployment of APIs to the staging environment by API developers.
Validating releases in Pre-Production

- Functional testing:
  - Validate upstream/downstream APIs at every point of a request transaction
- Performance testing:
  - HTTP/gRPC/GraphQL response times
  - Cache and database response times

See you space cowboy...
LEAP 2024
THE API OBSERVABILITY CONFERENCE
A VIRTUAL EVENT HOSTED BY Tyk | 29 FEB 2024
Register today
Thank you!

Come talk to us to continue the discussion or reach out:

- Adnan Rahić
  - [linkedin.com/in/adnanrahic/](https://linkedin.com/in/adnanrahic/)

- Sonja Chevre
  - [linkedin.com/in/sonjachevre/](https://linkedin.com/in/sonjachevre/)