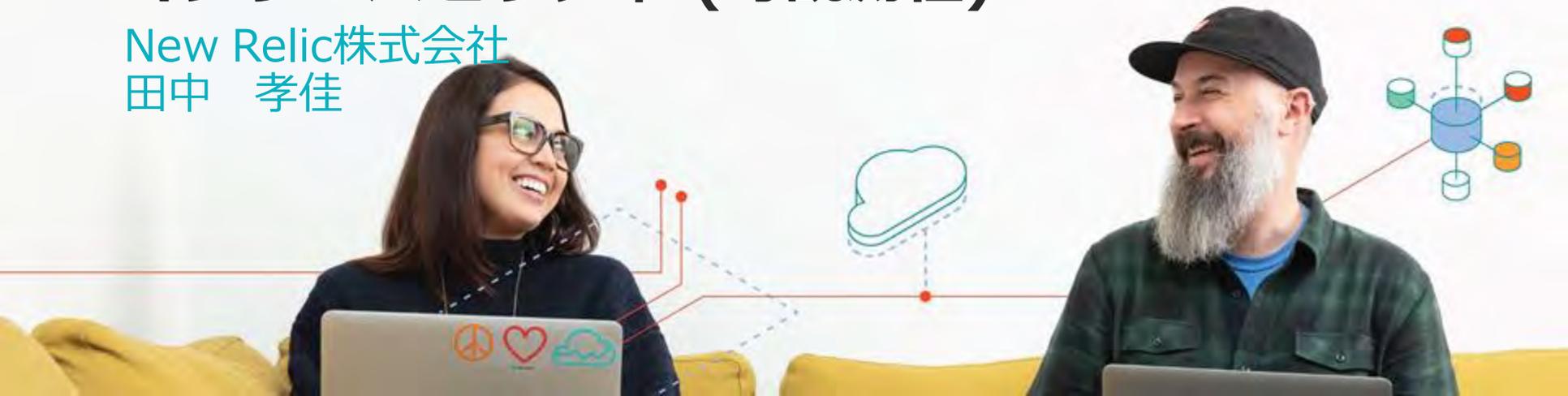
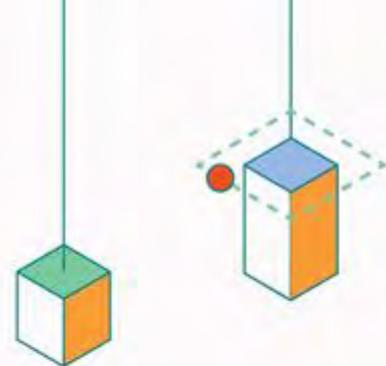




A-3 #devsumiA

# New RelicのOSSツールと Kubernetesクラスターの オブザーバビリティ（可観測性）

New Relic株式会社  
田中 孝佳





# 田中 孝佳

Senior Technical Support Engineer

Java, C#, Rubyなどでのソフトウェアエンジニア、  
New Relicユーザーとしてインフラエンジニア経験を経て  
テクニカルサポート職

Microsoft MVP (Azure, Developer Technologies)

好きな言語: C# (.NET Core)

マイブーム: ピアノ (子供に連弾せがまれて練習中...)



# 本日のゴール

- アプリケーションが動いてこそそのkubernetes運用
- kubernetesの運用はどこに注目すればよいのか
- OSSをどのように活用できるか

# 本日のAgenda

1. New Relicの簡単なお紹介
2. kubernetesのオブザーバビリティ（可観測性）
3. アプリケーション中心のオブザーバビリティへ
4. まとめ

# New Relic の 簡単なお紹介



私たちの存在意義

# *More Perfect Software*

“ソフトウェアをより完璧なものに”



# オブザーバビリティ（可観測性）とは？

## モニタリング

何かが間違っているときに通知  
When?What?

## オブザーバビリティ

根本原因まで特定可能  
Why?How?



# オブザーバビリティプラットフォーム

より良いソフトウェアの開発と実行

## New Relic BROWSER

ブラウザ体験モニタリング  
ユーザー目線でページロードやエラーを把握



Front-End

## New Relic MOBILE

モバイル環境をモニタリング  
iOSとAndroidアプリに対応

## New Relic SYNTHETICS

外形モニタリング  
世界複数拠点からの外形監視

顧客体験の改善



Back-End

## New Relic APM

アプリケーション性能モニタリング  
8言語と70を超えるフレームワークに対応

## New Relic INFRASTRUCTURE

あらゆるインフラ環境をモニタリング  
パブリッククラウドとオンプレミス

## New Relic LOGS

ログ収集と高速検索  
MELTを高速収集し検索可能に



Analysis

## New Relic ONE™

ダッシュボード開発/チャートビルダーで  
分析を超高速化し、あらゆるテレメトリ  
データの可視化を実現



Perfect  
Software

複雑かつ大規模システムの管理

## NRDB

世界最速のデータ収集と検索



# kubernetesの オブザーバビリティ

# kubernetesクラスター運用の課題



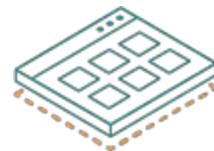
運用ツールの  
設計・運用



計測する指標の  
選定



動的に変化する  
クラスターへの  
追隨



アプリも対象  
とした運用

# kubernetesクラスター運用の課題の解決



運用ツールの  
設計・運用

オブザーバビリティ  
プラットフォーム  
の活用



計測する指標の  
選定

MELTに従った  
指標



動的に変化する  
クラスターへの  
追従

kubernetes  
オブジェクトに  
関連づけた計測



アプリも対象  
とした運用

アプリ中心の  
計測も採用

# オブザーバビリティプラットフォーム



データの保持、可視化、アラートなどを兼ね備えたプラットフォーム



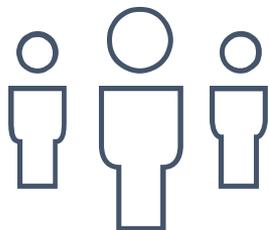
テレメトリデータの取得・送信

kubernetes



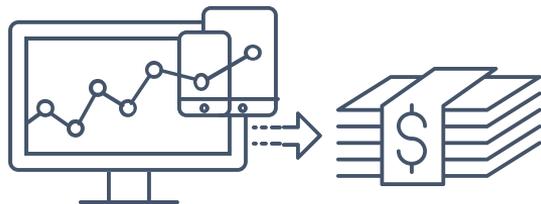
# オブザーバビリティプラットフォームに求められる特性

## 多くの人が使える



- トレーニングコストが低い
- アクセスしやすい

## 運用コストが低い

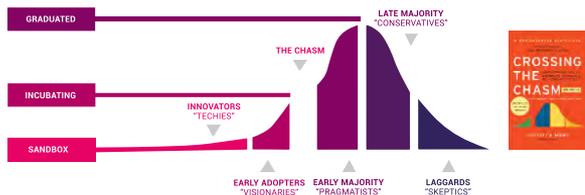


- 運用に特別なスキルが必要ない
- コストの予測が立てやすい

## オブザーバビリティの民主化

# どのように解決するか

## 広く使われているツール



<https://www.cncf.io/projects/>

例えば、CNCF Graduated Projects  
(kubernetesもこれ)

## SaaSの利用



運用ツール自体の運用はSaaSまかせ  
コストの予測可能性はSaaS次第

## この2つを組み合わせよう

# New Relicによるkubernetesのオブザーバビリティ獲得

 New Relic®



kube-state-metrics +  
forwarder

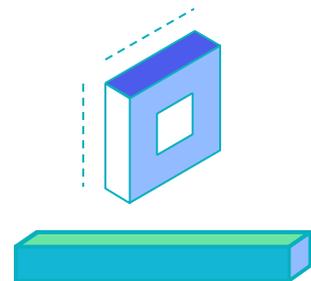
event  
forwarder

fluentbit +  
newrelic plugin

kubernetes

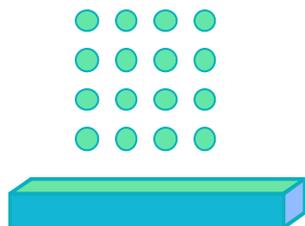


# オブザーバビリティの四本柱MELT



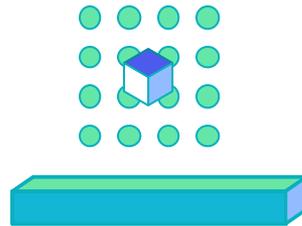
Metrics

定期的に集約した測定値の集合



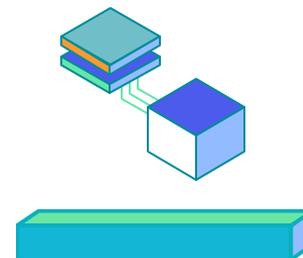
Events

タイムスタンプ付きのアクションの記録



Logs

タイムスタンプ付きのテキストメッセージ



Traces

アプリケーションの処理の連鎖



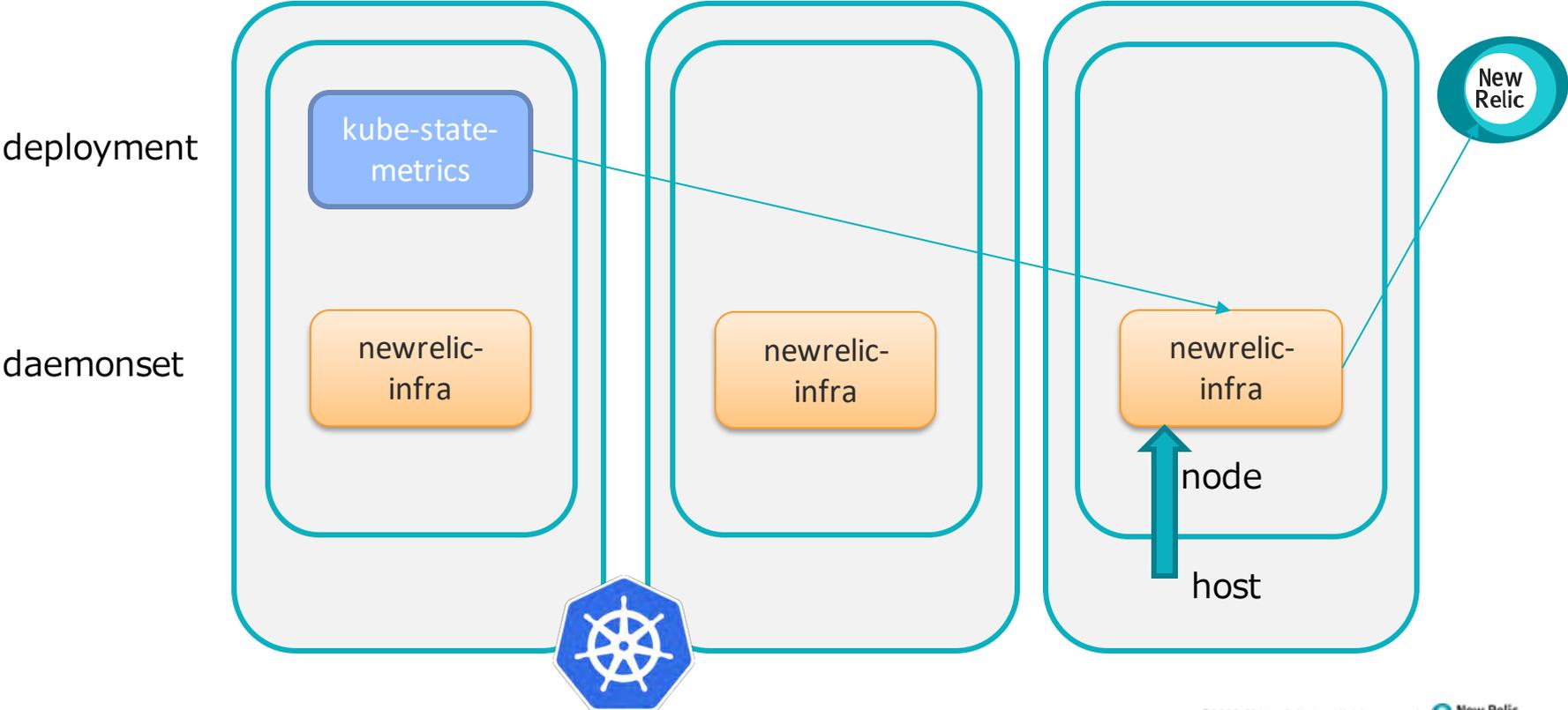
多すぎるイベントを属性ごとに集約して数値化



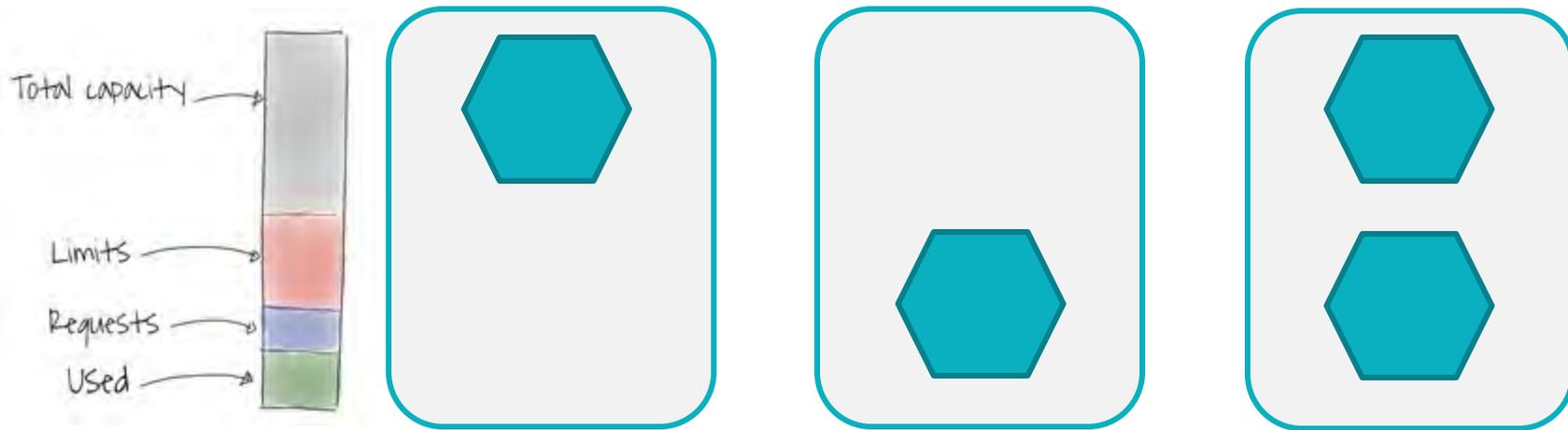
より汎用的に使えるように簡素化

<https://blog.newrelic.co.jp/story/metrics-events-logs-and-traces/>

# New RelicによるMetricsのexport

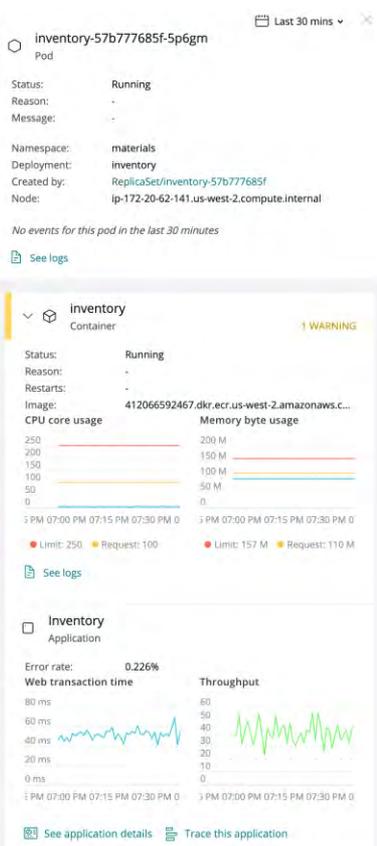
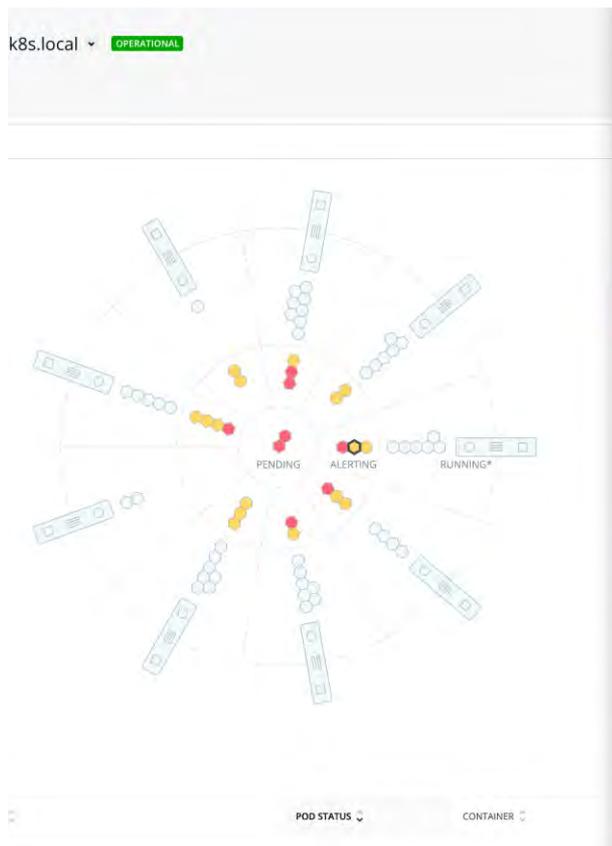


# kube-state-metricsを使ったMetricsの収集



- deploymentやdaemonsetで配置されるPodは必要に応じて生成・消滅する
  - Pod単体としての値以上に、deploymentとしての値が必要
- requests, limitsが制限されている場合、それとの比も重要
- kube-state-metricsを利用
  - <https://github.com/kubernetes/kube-state-metrics>

# New RelicによるMetricsの可視化



deployment単位のCPU使用率

Since 1 hour ago until 1 minute ago

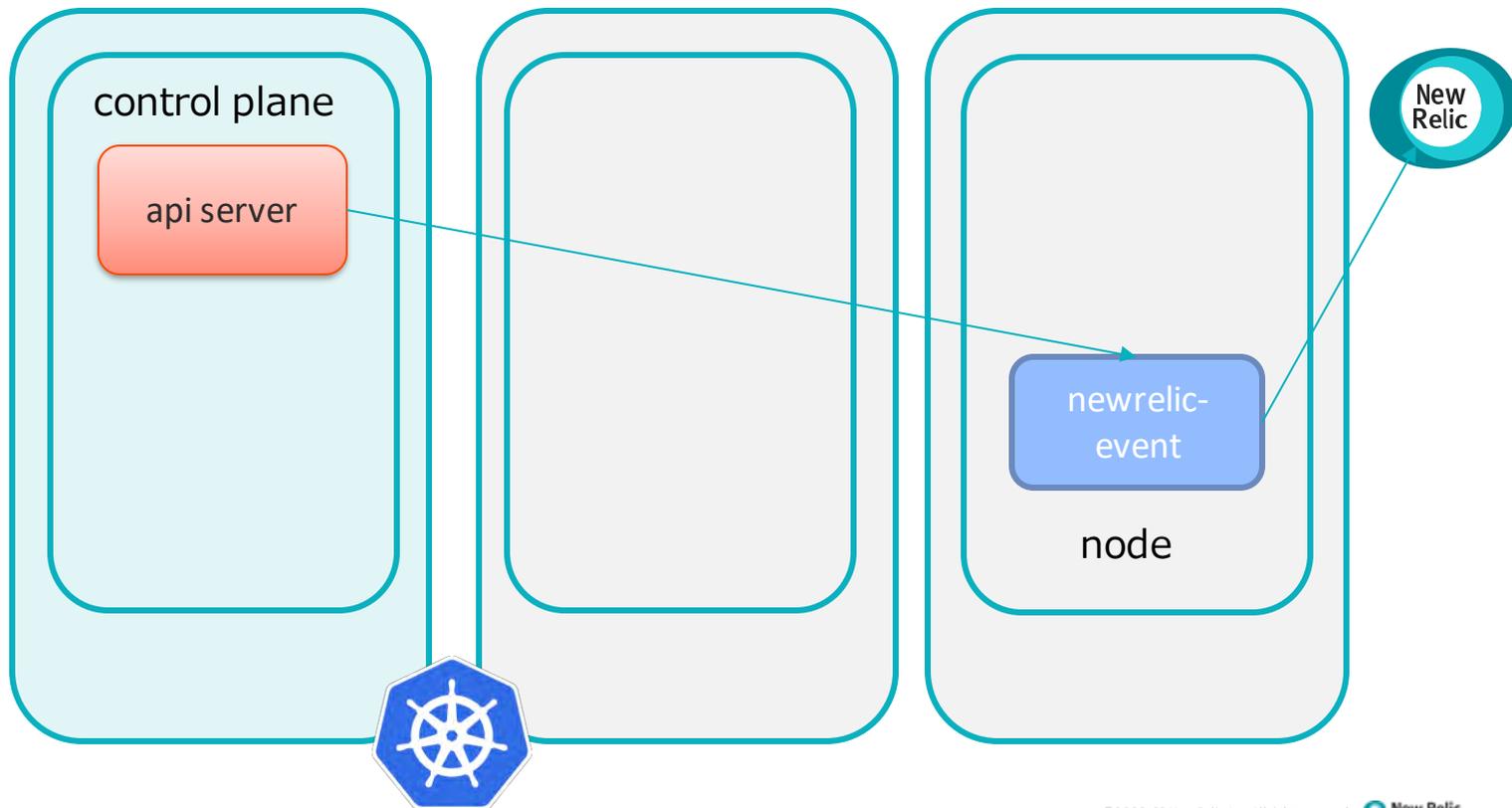


# Events

さまざまな  
イベント

Nodeが  
利用不可 

Imageが  
見つからない 





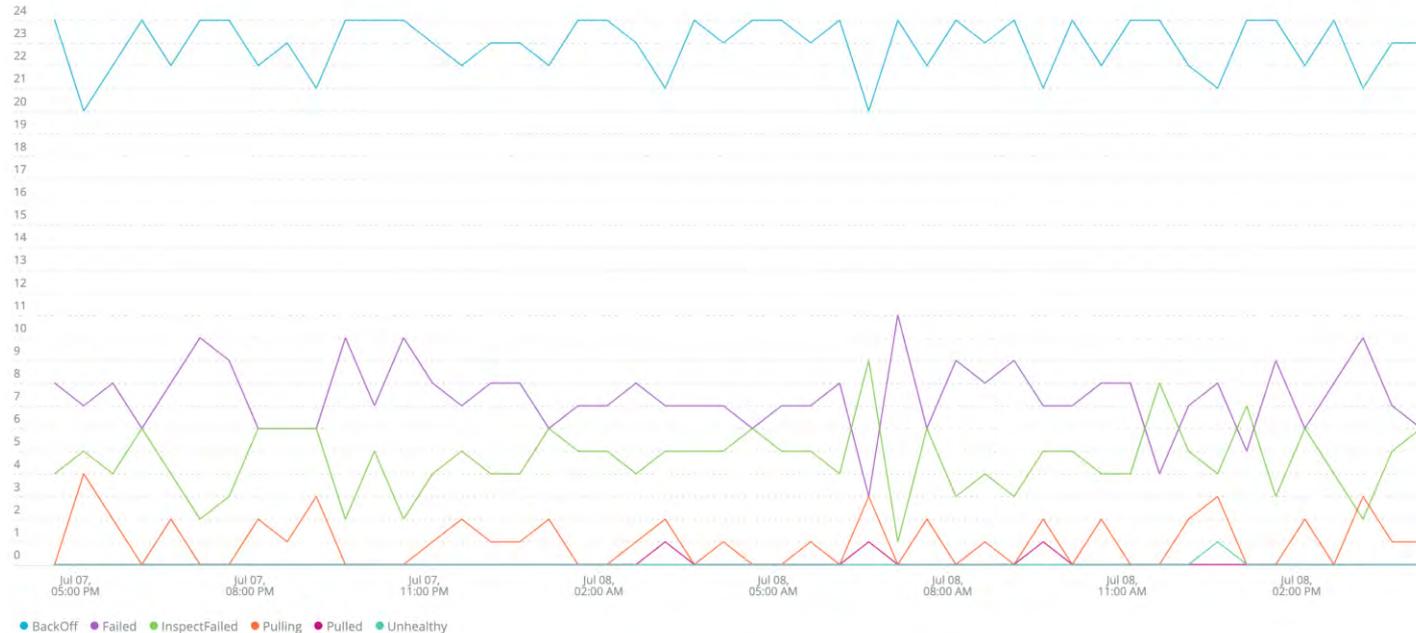
# Eventsの集計

```
SELECT count(*) FROM InfrastructureEvent FACET event.reason WHERE (category = 'kubernetes') SINCE 24 HOURS AGO TIMESERIES
```

Your recent queries ▾

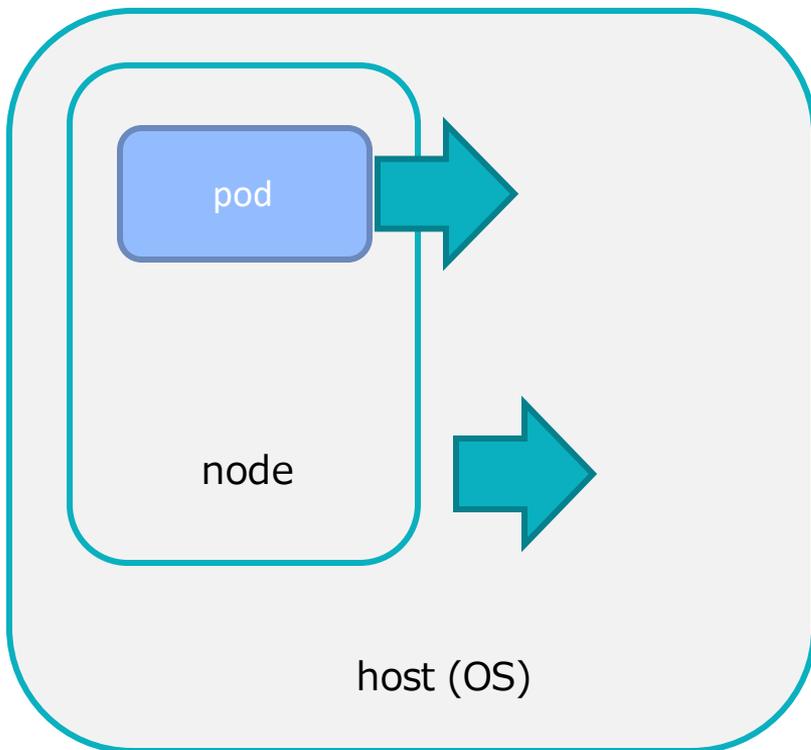
Since 1 day ago

Share ▾



3.77 thousand events inspected in 64 ms ( 0.1 MEPS)

# kubernetesのLog



コンテナから出力されるログ

→次ページで説明

fluntbit+New Relic output pluginで転送

コンテナ以外のプロセスから  
出力されるログ

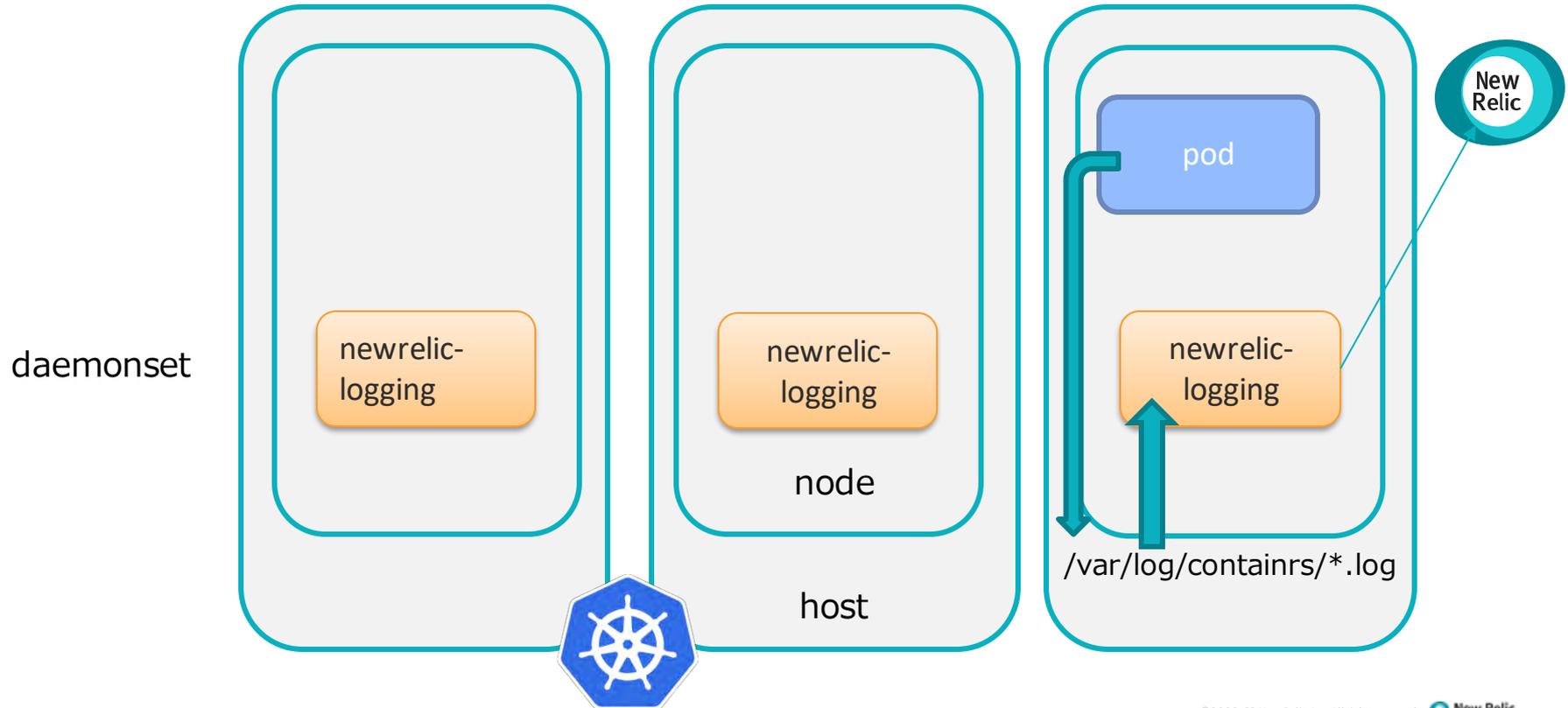
→ hostのinfra-agentを経由したり、  
クラウドサービスを経由したりする

参考)

<https://github.com/newrelic/newrelic-fluent-bit-output>

<https://github.com/newrelic/newrelic-fluentd-output>

# New RelicへのPodのLog転送



# Logsの可視化

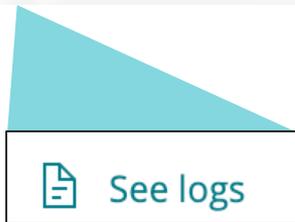
order-status-7b45f498c7-85zxr  
Pod

Status: Running  
Reason: -  
Message: -

Namespace: interactions  
Deployment: order-status  
Created by: ReplicaSet/order-status-7b45f498c7  
Node: ip-172-20-59-103.us-west-1.compute.internal

No events for this pod in the last 30 minutes

[See logs](#)



Account Demotron V2

Find logs where pod\_name:"order-status-7b45f498c7-85zxr"

Since 30 minutes ago

timestamp message

08:52:45.941	PO 2290769: order-composer called order-status/warehouse
08:52:45.941	PO 9104388: processing on order-status/prepare_order
08:52:45.941	PO 7247735: processing on order-status/clear_for_warehouse
08:52:45.941	PO 7247735: order-composer called order-status/warehouse
08:52:45.941	PO 7252751: order-composer called order-status/ready
08:52:45.941	PO 7914967: order-composer called order-status/ready
08:52:45.941	PO 1197900: order-processing called order-status/orders
08:52:45.941	PO 7474707: processing on order-status/prepare_order
08:52:45.941	PO 1455142: order-processing called order-status/orders
08:52:45.941	PO 8518093: processing on order-status/prepare_order
08:52:45.941	PO 6491161: order-composer called order-status/start_order
08:52:45.941	PO 4825339: processing on order-status/prepare_order
08:52:45.941	PO 6588564: processing on order-status/return_order
08:52:45.941	PO 6588564: order-processing called order-status/returns

# helm chartとして提供

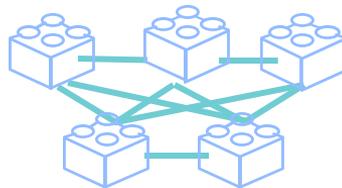
- ここまで紹介した機能はすべてkubernetes自身の上で展開できる
- コンテナイメージ+展開用のオブジェクト定義(YAML)を提供
- より自動化・管理しやすいようhelm chartとして提供
  - <https://github.com/newrelic/helm-charts>



# アプリケーション中心の オブザーバビリティへ



# アプリケーションへ



**kubernetes**

- kubernetesはインフラ基盤
- その上で動くアプリが健全に動いてこそサービスを提供できる



日々の作業を改善することは、  
日々の作業をこなすことより  
はるかに重要です



**Gene Kim**

*The Phoenix Project: A Novel about IT,  
DevOps, and Helping Your Business Win*

## ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have...



**208**  
TIMES MORE  
frequent code deployments

**106**  
TIMES FASTER  
lead time from  
commit to deploy



**2,604**  
TIMES FASTER  
time to recover from incidents

**7**  
TIMES LOWER  
change failure rate  
(changes are 1/7 as likely to fail)



Throughput Stability

サービス復旧までの時間

SOURCE:

[2019 Accelerate State of DevOps Report](#)

# アプリケーションのオブザーバビリティ



## 計測すべき指標の選定

APMを入れて  
MELTを取得する



## 動的なインフラ環境との 関連づけ

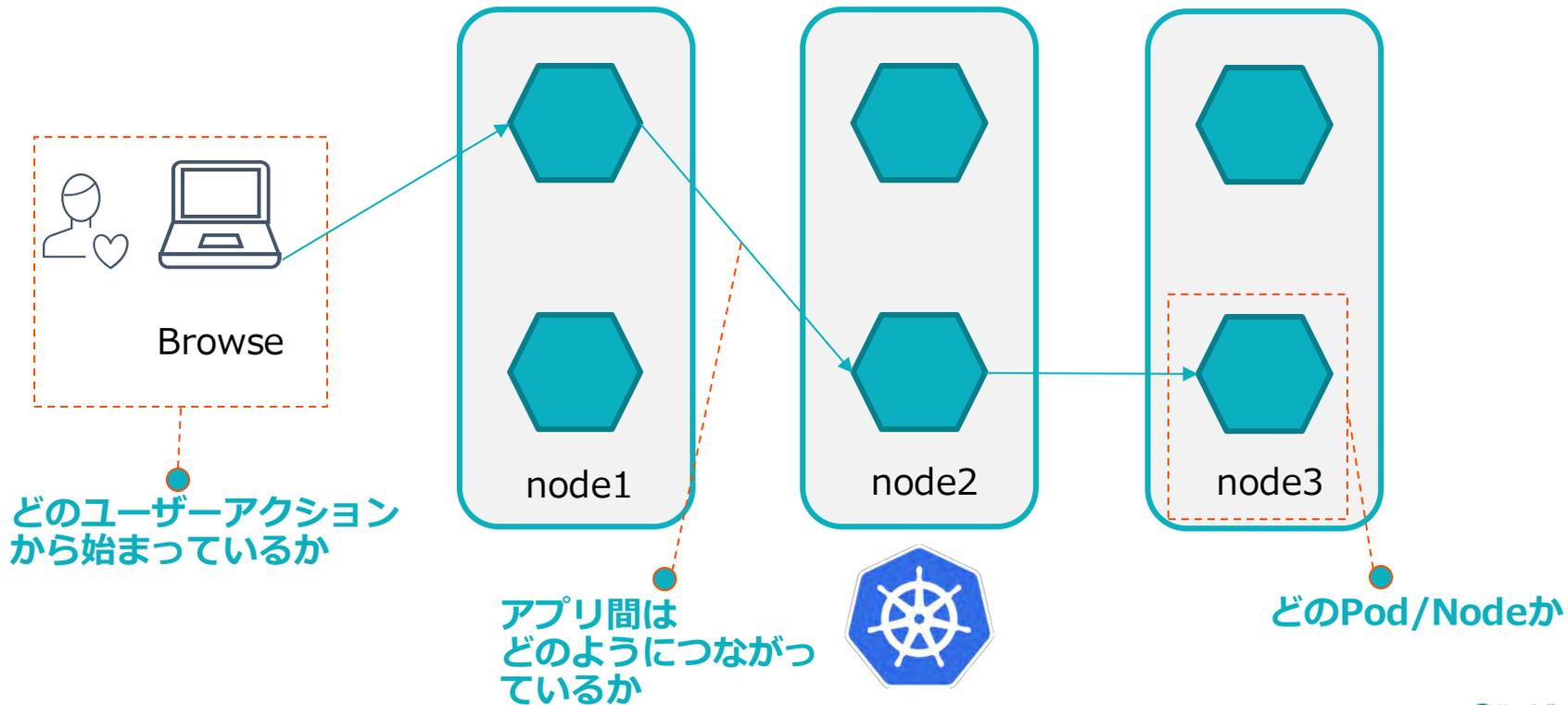
kubernetesオブジェクト  
とのContextを取得する



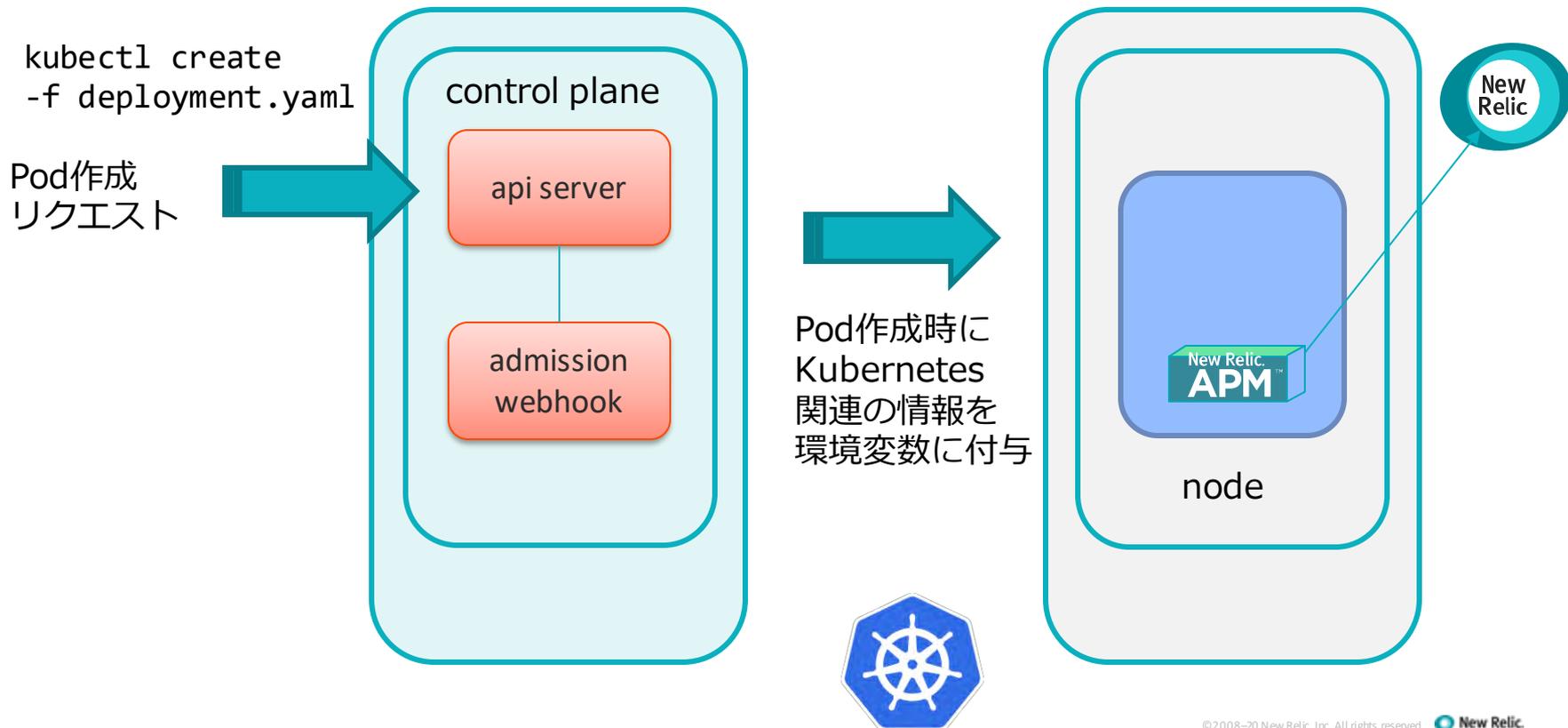
## 協調して動く アプリ同士の関連付け

アプリケーション同士の  
Contextを取得する  
特に分散トレーシング

# Contextの重要性

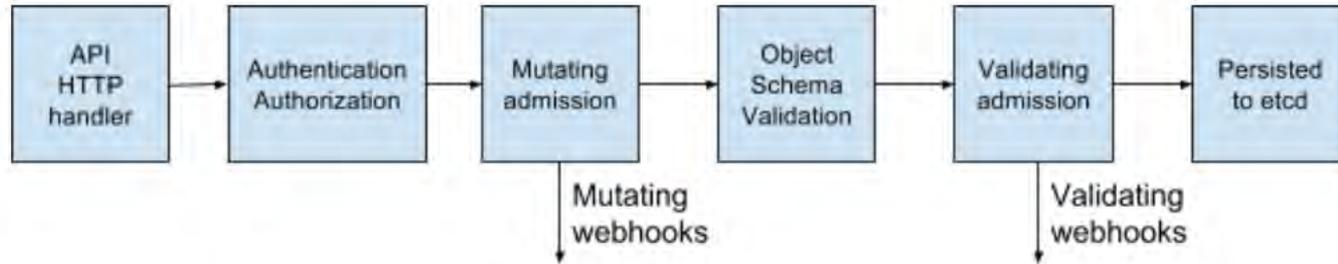


# APM metadata injectionによるContextの取得



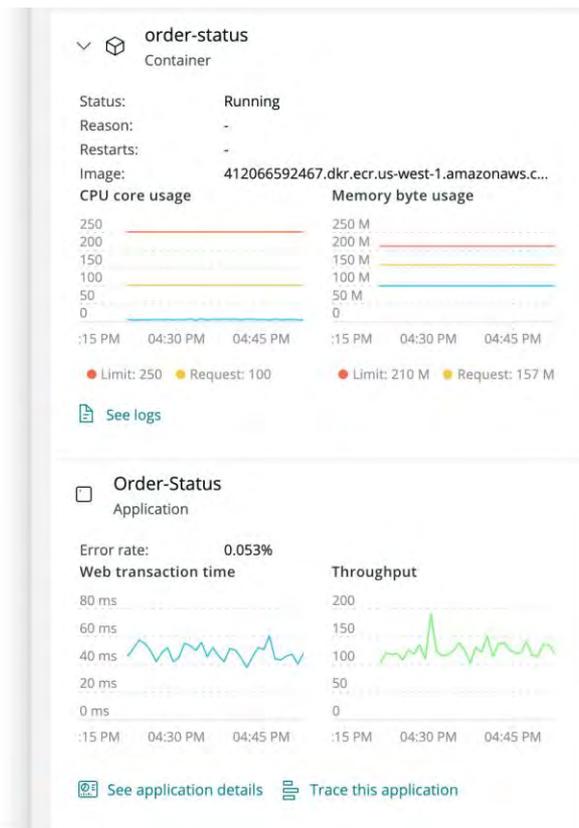
# admission webhookとは

kubernetesへのAPIリクエストに介入できる拡張ポイントみたいなもの

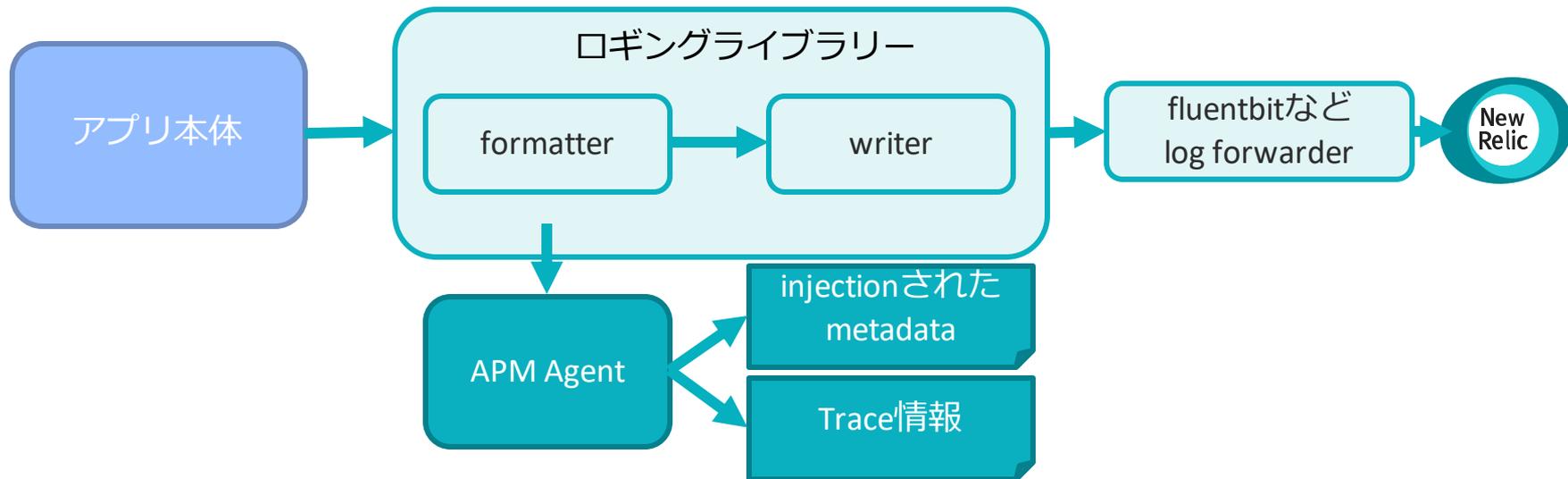


- 説明  
<https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers/#mutatingadmissionwebhook>
- New Relicのコード  
<https://github.com/newrelic/k8s-metadata-injection/>

# Metricsとインフラをむすびつける



# LogsのContext



- ロギングライブラリーごとにformatterなどを提供する
- New RelicのformatterはAPM AgentのAPIから必要なContextをログに付与する
- New Relicが提供していないライブラリでも自作するための仕様を公開

# LogsとTrace, インフラを結びつける

The screenshot shows the New Relic WebPortal interface. On the left is a navigation menu with categories like MIDWINTER, Overview, Distributed tracing, Service maps, Transactions, External services, JVMs, ERROR, Error analytics, Errors, Logs, Violations, Deployments, Thread profiler, REPORTS, SLA, Scalability, Web transactions, Databases, and Background jobs. The main content area displays error details for an `org.apache.http.client.HttpResponseException` with the message `org.apache.http.client.HttpResponseException: Internal Server Error`. Below this is a stack trace showing the call path through various interceptors and service proxies. A 'See logs' button is visible in the top right of the error details section.

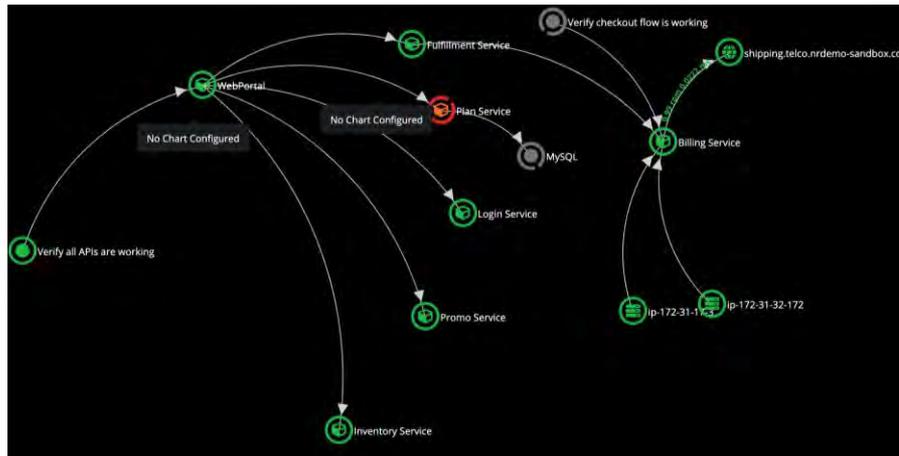


The screenshot shows the New Relic Logs viewer interface. At the top, it says 'Logs' and 'Find logs where trace id:bed414224525247c OR span id:026f59cb13cb64'. Below this is a search bar and a 'Query logs' button. A line graph shows log activity over time, with a single sharp peak at 07:25 PM. Below the graph is a table of log entries:

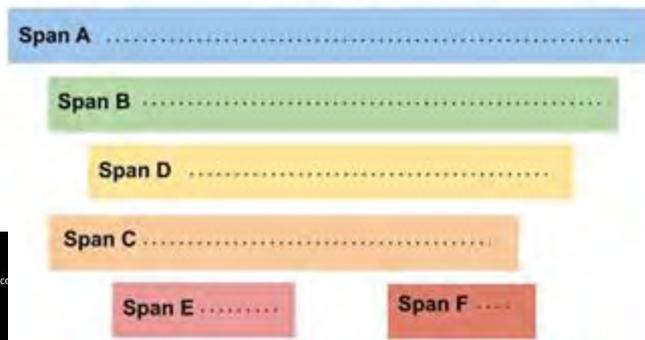
timestamp	message
11:23:14.526	Validating coupon...
11:23:14.785	Adding X-TELCO-USERID: (1659551269)
11:23:14.791	/api/v1/coupons/c6c60f9f-12cd-4a99-bba6-872781342396/invalid GET
11:23:15.311	Error validating coupon: Internal Server Error

# アプリケーションのつながりを見る

分散トレーシング を有効化し、  
アプリのつながりを見て  
ボトルネックを特定

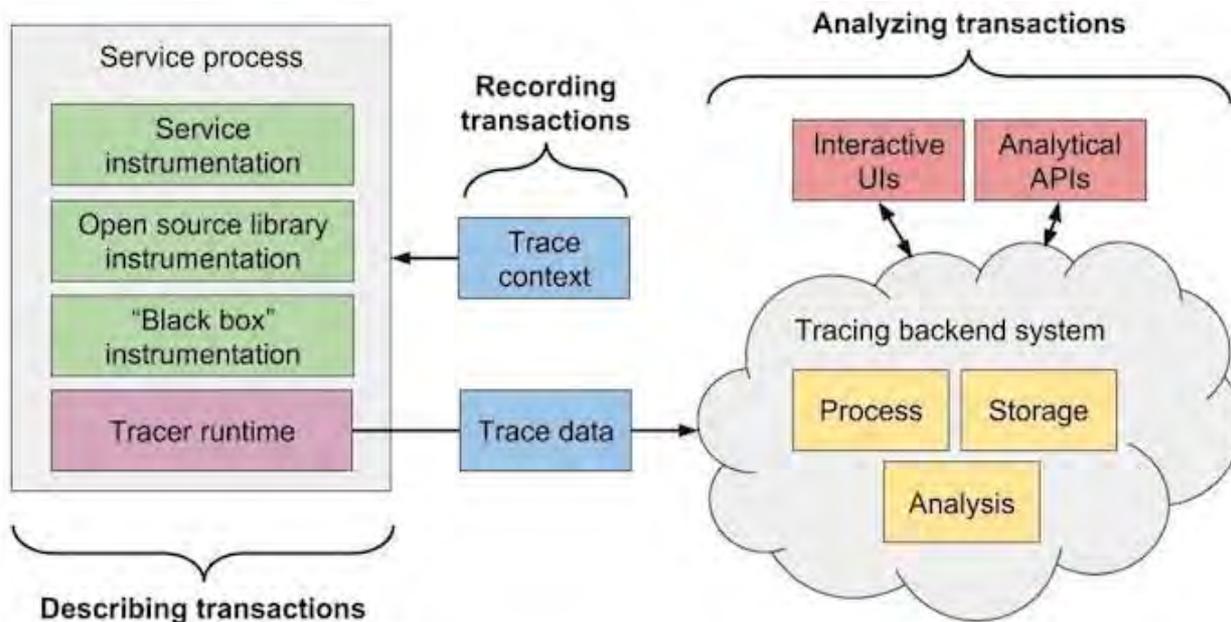


Trace



Span : 処理の単位

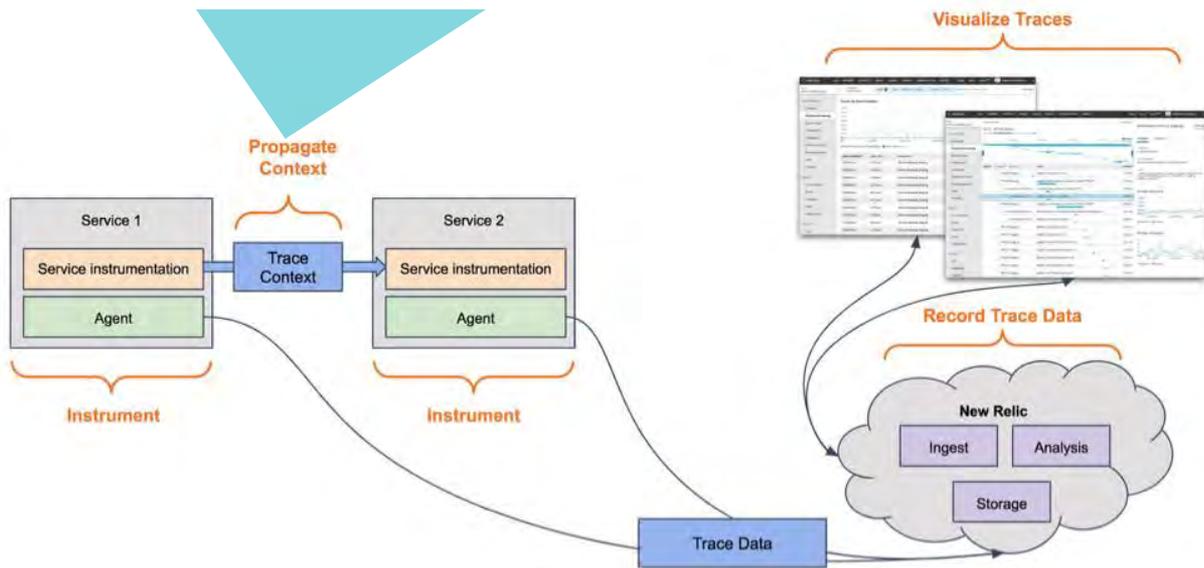
# 分散トレーシング計測の概要



<https://blog.newrelic.co.jp/engineering/understanding-distributed-tracing/>

# 分散トレーシングに必要な情報伝播

伝搬すべきTrace Context情報

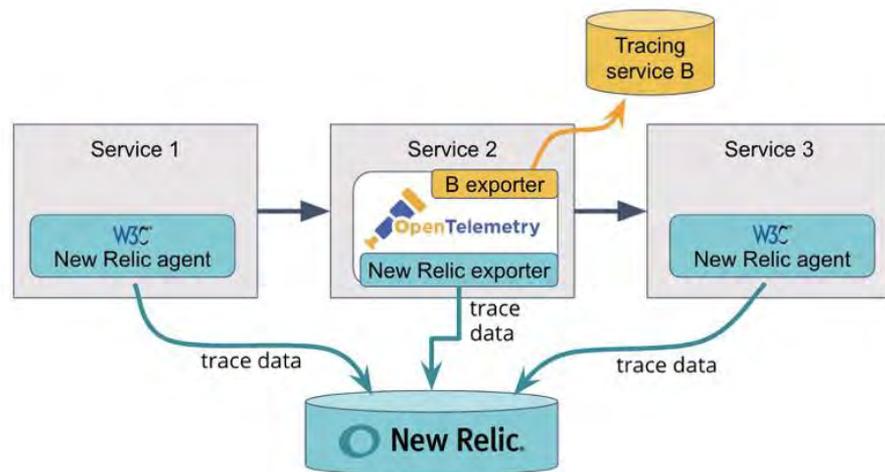


W3C Distributed Tracing Working Group

Defining common standards for propagating traces and context between tracing tools.

伝搬の方法の標準規格を作る

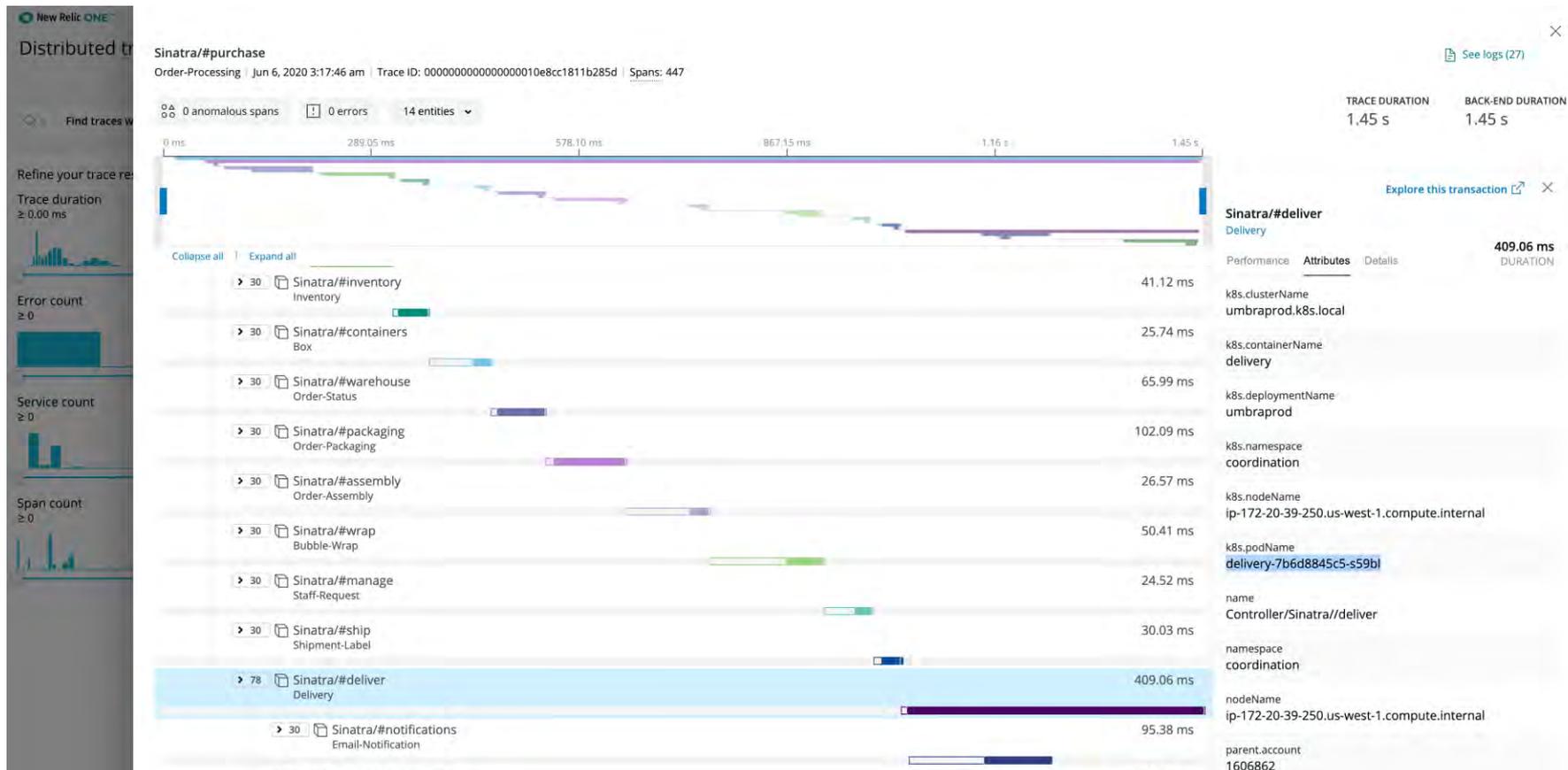
# すべてを分散トレーシングに参加させる



- New Relic APM Agentを入れられないアプリからもTraceをNew Relicに送信
  - New Relic以外のAPMを利用したい
  - IstioなどAgentを入れられないレイヤーが存在する
- OpenTelemetry以外にもZipkinフォーマットなどにも対応

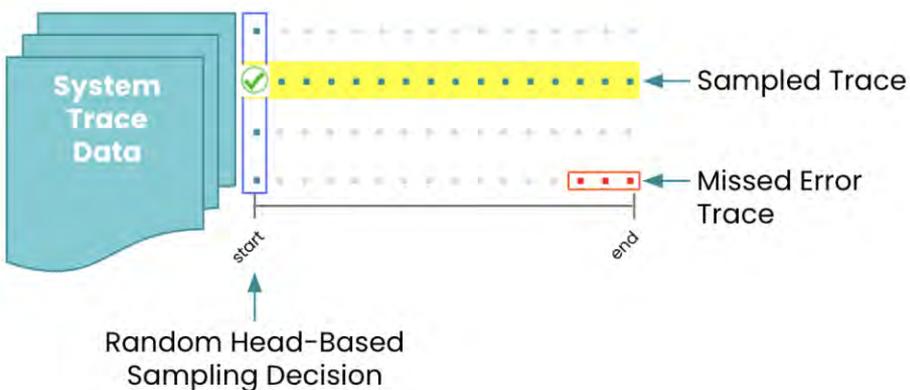
<https://blog.newrelic.co.jp/product-news/distributed-tracing-interoperability-new-relic-apm-w3c-trace-context/>

# kubernetesの情報と分散トレーシングの関連付け

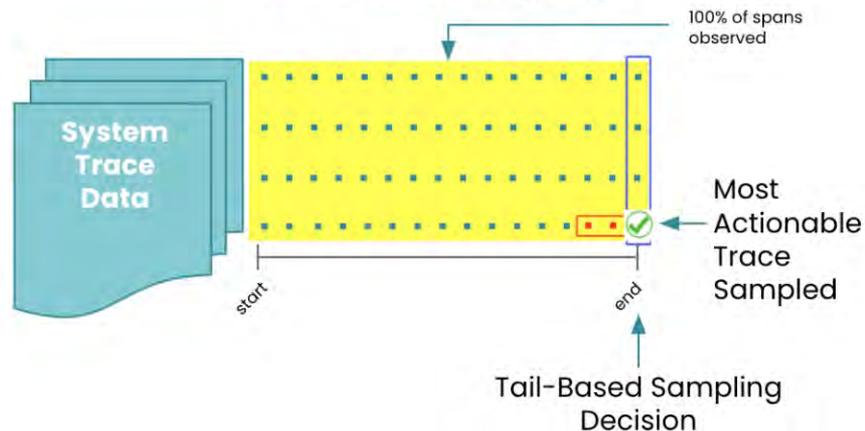


# (参考) headベースとtailベース

## Traditional Head-Based Sampling



## Tail-Based Sampling



エラーなど異常なトレースこそ見たいが、headベースだとサンプリングで除外されうるより重要なトレースを確実に計測するにはTailベースが必要  
New Relic EdgeはマネージドなTailベースサンプリングの分散トレーシングを提供します

# New RelicとOSS



External Projects   New Relic Projects

**Open standards.  
Open instrumentation.  
Open collaboration.**

We **built** this site to make it easy for *you* to explore the **open source projects** we maintain and the **open standards projects** we participate in. Learn [more](#).



## Projects that we support

New Relic contributes resources to the development of these projects.



### OpenTelemetry

OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software.



### W3C Distributed Tracing Working Group

Defining common standards for propagating traces and context between tracing tools.

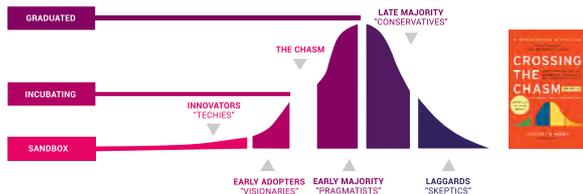


### Adopt OpenJDK

The Community and code behind the Build Farm which produces high quality, FREE OpenJDK (Java) binaries.

# どのように解決するか

## 広く使われているツール



例えば、CNCF Graduated Projects  
(kubernetesもこれ)

## SaaSの利用



運用ツール自体の運用はSaaSまかせ  
コストの予測可能性はSaaS次第

## この2つを組み合わせよう



**マネージドで自動化された  
ソリューションのおかげで、  
問題を特定しMTTRを削減するために、  
チームが必要とする  
テレメトリデータへのアクセスを  
民主化することができます。**



Jeremy Castile

Principal Product Marketing Manager at New Relic

<https://diginomica.com/business-impact-managed-distributed-tracing>

# まとめ

- アプリケーションが動いてこそそのkubernetes運用
  - アプリケーション中心のオブザーバビリティも獲得しよう
- kubernetesの運用はどこに注目すればよいのか
  - オブザーバビリティプラットフォームの利用
  - MELTに着目したメトリクスの取得
- OSSをどのように活用できるか
  - より広く使われ、より安心して自分のアプリの近い部分に組み込める

# ISUCON向け特別ライセンス提供します

New Relic 株式会社



ISUCON10  New Relic.

詳しくはISUCON公式サイトから弊社の特設記事へ  
<http://isucon.net/archives/54638481.html#NewRelic>

# Thank You

ttnaka@newrelic.com

@tanaka\_733

