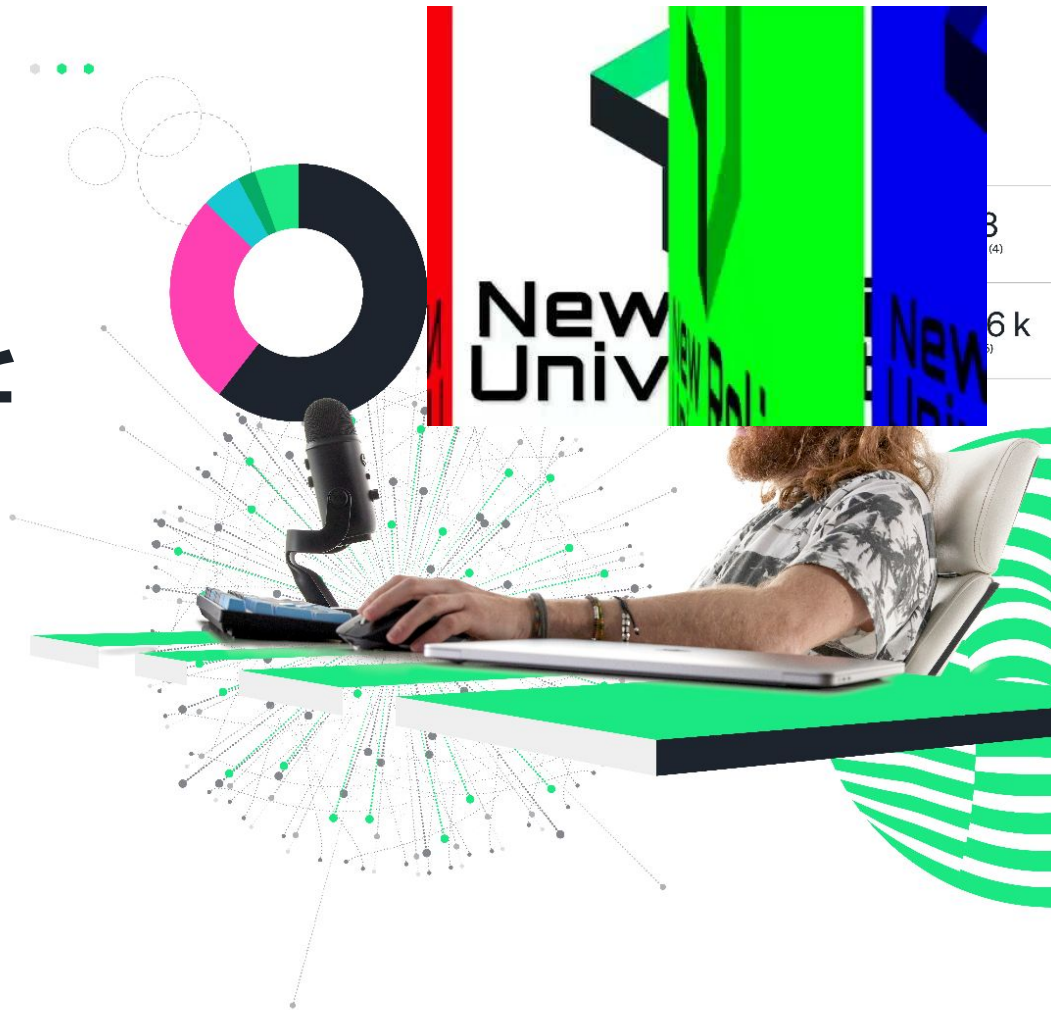




ダッシュボード開発と NRQLの基本編

NRU 302 - Dashboard / NRQL



ウェビナー 各種ご連絡

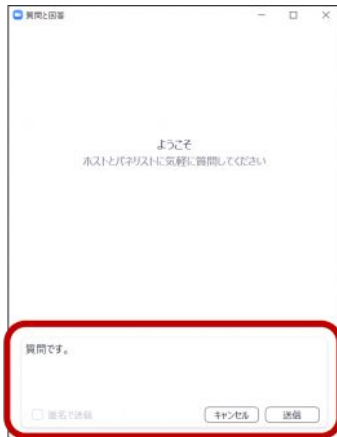
1. ご質問がある場合は、“Q&A”からご入力ください。



① 画面下
「Q&A」をクリック！

こちらにご質問をご記入し、
「送信」をクリックしてください！

②



2. 本日の資料はこの後“チャット”でURLを共有します。アクセスできない場合は、“Q&A”よりお名前とメールアドレスをご連絡ください。

Safe Harbor

This presentation and the information herein (including any information that may be incorporated by reference) is provided for informational purposes only and should not be construed as an offer, commitment, promise or obligation on behalf of New Relic, Inc. ("New Relic") to sell securities or deliver any product, material, code, functionality, or other feature. Any information provided hereby is proprietary to New Relic and may not be replicated or disclosed without New Relic's express written permission.

Such information may contain forward-looking statements within the meaning of federal securities laws. Any statement that is not a historical fact or refers to expectations, projections, future plans, objectives, estimates, goals, or other characterizations of future events is a forward-looking statement. These forward-looking statements can often be identified as such because the context of the statement will include words such as "believes," "anticipates," "expects" or words of similar import.

Actual results may differ materially from those expressed in these forward-looking statements, which speak only as of the date hereof, and are subject to change at any time without notice. Existing and prospective investors, customers and other third parties transacting business with New Relic are cautioned not to place undue reliance on this forward-looking information. The achievement or success of the matters covered by such forward-looking statements are based on New Relic's current assumptions, expectations, and beliefs and are subject to substantial risks, uncertainties, assumptions, and changes in circumstances that may cause the actual results, performance, or achievements to differ materially from those expressed or implied in any forward-looking statement. Further information on factors that could affect such forward-looking statements is included in the filings New Relic makes with the SEC from time to time. Copies of these documents may be obtained by visiting New Relic's Investor Relations website at ir.newrelic.com or the SEC's website at www.sec.gov.

New Relic assumes no obligation and does not intend to update these forward-looking statements, except as required by law. New Relic makes no warranties, expressed or implied, in this presentation or otherwise, with respect to the information provided.



Google Cloud Partner Top Engineer '2021



Yoshikazu Okawa

Cloud Architect for B2C Industry Infrastructure, Google Cloud

New Relic

Technical Account Manager



本ウェビナーの受講想定者

- New Relic を使用している
- すでに New Relic をハンズオンで触ったが、具体的なチャート作成や可視化方法を知りたい
- New Relicのアラート機能を使っている、またはこれから使いたいと思っている

New Relicの知識に不安のある方はこちらも受講ください! (オンデマンド視聴可)

参考: <https://newrelic.com/jp/webinar/nrb-newrelic-essentials>

本日のゴール

- New Relicのデータ分析機能について理解する
- New Relicが取得するデータ構造について理解する
- データ分析機能の中核となるNRQLについて理解する
- 自在にダッシュボードを作れるようになる
- アラート作成の概要を理解する

本日のタイムテーブル

時間	タイプ	内容
15:00 - 15:20	座学	New Relicとデータ分析 New Relicが取得するデータ
15:20 - 15:40	ハンズオン #1	データの理解
15:40 - 15:50	座学	New Relic ダッシュボード
15:50 - 16:10	ハンズオン #2	ダッシュボードの作成
16:10 - 16:15	座学	NRQL (New Relic Query Language)
16:15 - 16:35	ハンズオン #3	分析手法の習得
16:35 - 16:40	座学	高度なアラート設定
16:40 - 16:50	ハンズオン #4	アラートの作成
16:50 - 17:00	座学	さいごに

ハンズオンで使用するブラウザについて

⚠ ブラウザは下記のいずれかをご利用ください。

- Chrome
- Firefox
- Edge

⚠ 普段 New Relic をお使いの方はセッションが残っている場合があります。シークレットウィンドウなどをお使いください。

- Chrome: シークレットウィンドウ
- Firefox: プライベートウィンドウ
- Edge: InPrivateウィンドウ

参考: New Relic UIの対応ブラウザ

© 2024 New Relic, Inc. All rights reserved. <https://docs.newrelic.com/jp/docs/new-relic-solutions/get-started/supported-browsers-new-relics-ui/>

New Relicと データ分析

NRU302 - 座学

new relic オブザーバビリティプラットフォーム 全体像



Frontend

顧客体験の改善

Browser Monitoring

ブラウザ体験モニタリング
ユーザー目線で描画速度やエラーを把握

Mobile Monitoring

スマートフォンアプリをモニタリング
iOS, tvOS, Androidのネイティブアプリや
Flutter, React Native, Unityアプリまで対応

Synthetic Monitoring

外形監視ソリューション世界複数拠点に加え、
社内ネットワークからも外形監視



Backend

より良いソフトウェアの開発と稼働

APM 360

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

CodeStream

IDE上でNew Relicで特定したErrorから
該当ソースコードに直接ジャンプ可能

Log Management

トランザクションに結びついたアプリケーション
ログからサーバーログ、ネットワークログ、
その他どんなログも収集し高速検索

Infrastructure Monitoring

AWS, Azure, GCPに加えオンプレミスサーバー
ネットワーク状況 データベースまでモニタリング



Analysis

複雑かつ大規模システムの管理

New Relic AI

生成AIによりシステムの状況分析をアシスト
多様なデータから深いインサイトを引き出す

Alerts

Anomalyを利用した検知と根本原因示唆によるインシ
デント対応の高速化

Service Levels

ビジネスのKPIに合わせた定量的なサービス
レベルの設定と可視化分析

Dashboards

データ分析・共有・判断を超高速度
あらゆるテレメトリデータの可視化と共有

Vulnerability Management

アプリケーションの脆弱なライブラリ自動特定
システム全体の脆弱性を統合的に可視化



Perfect
Software

顧客視点データの取得

システムデータ

の取得

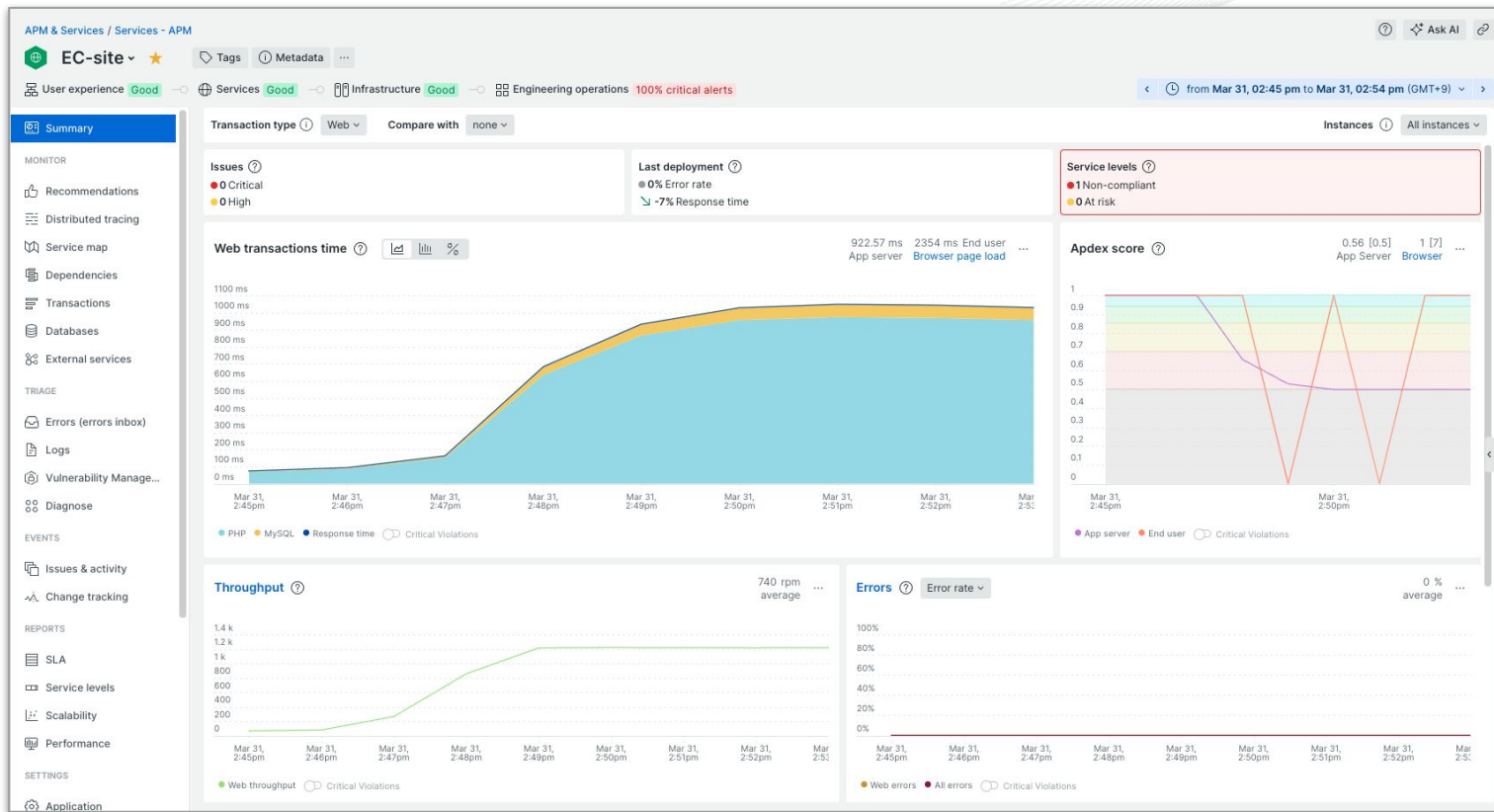


NRDB

分間、数十億件のデータを取り込み、処理
超高速検索を可能にするデータベース

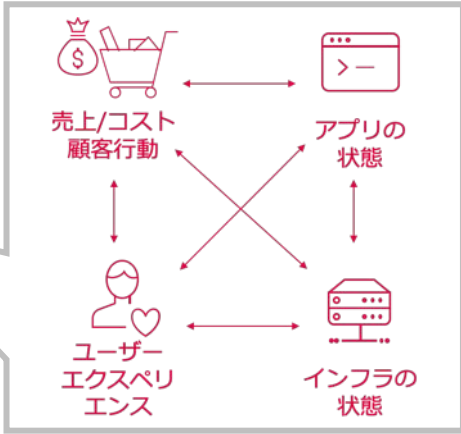
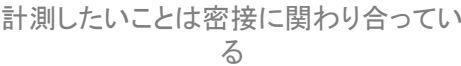
分析用データ集計

ビルトインのUIでもデータ分析はできます

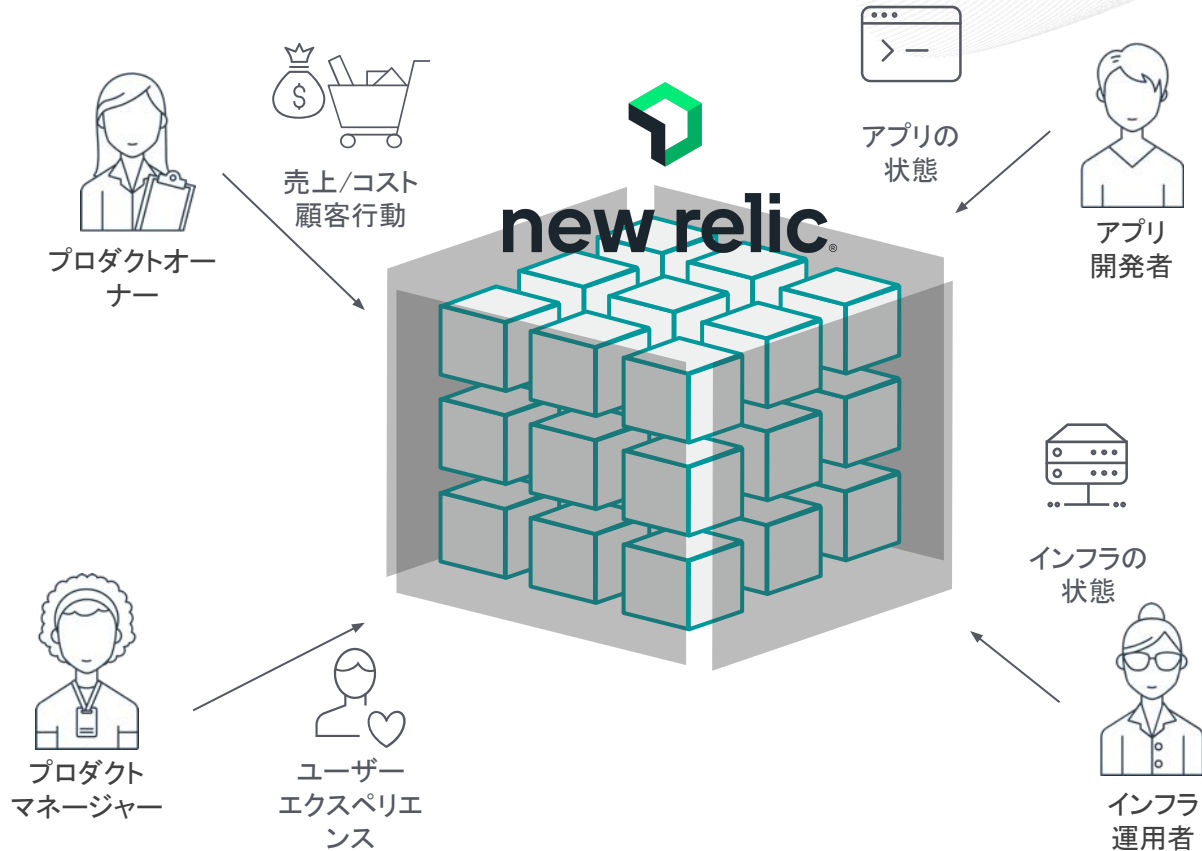


大きく異なる

システム



New Relic が提供するデータ分析機能



New Relicによるデータ分析のメリット



共通言語

それぞれの立場に沿った
共通言語をつくる



目標定量化

目標を計測可能にし、
中長期の目標値を合意する



達成確認

リアルタイムに
達成度合いを確認する

New Relic オブザーバビリティ 成熟モデル



オブザーバビリティ 成熟度モデル

特徴

KPIの例

0

Instrumentation
計測を開始する

サービスを理解するためのデータを集める
(ログ、トレース、API、ユーザ体験..)

- データ一元化率
- 対象システム割合
- ログ監視脱却→APM中心

1

Reactive
受動的対応

障害対応などの対応時間短縮
(関係者全員がリアルタイムに観測可能)

- サービス停止率
- 障害発生率
- MTTR削減

2

Proactive
積極的対応

大きな問題が起こる前に行動を起こす
(ユーザ起点からパフォーマンス改善)

- エラー発生率
- レスポンスタイム
- SLI/SLOの策定割合

3

Predictive
予測的対応

サービス改善のためのマインドチェンジ
(AIを活用した予防、開発スピード向上)

- 適正スケーリングによるコスト削減
- デプロイ高速化

4

Data Driven
データドリブン

より良いサービスのための投資と行動
(データに基づき正しい意思決定を加速)

- 顧客満足度の改善
- 市場投入までの時間(新製品・新機能の数)

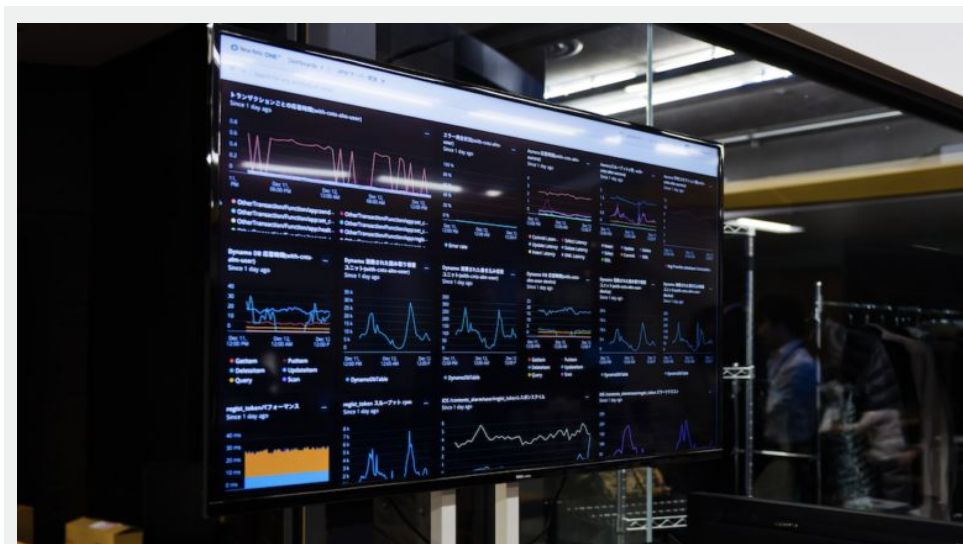
守り

攻め

データ分析を活用いただいているお客様事例

株式会社ウェザーニューズ様

<https://newrelic.com/jp/customers/weathernews>



ウェザーニューズ社のオフィスに表示されている New Relic ダッシュボード

New Relicが 取得するデータ

NRU302 - 座学

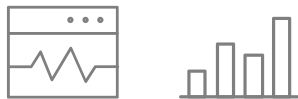
STEP1

データの
理解

New Relicが取得するデータ: MELT

M:メトリック

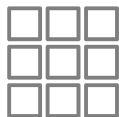
集計可能なデータの粒



例. キュー深度、
HTTPリクエスト数等

E:イベント

ある瞬間に発生する個別の
アクション



本日のトピック

L:ログ

個々の出来事の記録



例. デバッグログ、
エラーログ

T:トレース

単一ランザクションを構成する
あらゆる要素



例. 外部APIへのリクエスト、
DBへのクエリ等

参考: <https://newrelic.com/jp/blog/how-to-relic/metrics-events-logs-and-traces>

Eventとは

- 利用目的に応じた **Event名** と、複数の **属性(Attribute)** から成る構造化データ
- New Relicが扱うデータの中核であり、以下の2つの手段で収集される
 - 各エージェント(APM,Browser,Mobile,Synthetic,Infrastructure)から送信
 - Event APIでカスタムデータを送信
 - <https://docs.newrelic.co.jp/jp/docs/data-apis/ingest-apis/event-api/introduction-event-api/>
- Eventデータについて
 - <https://docs.newrelic.co.jp/jp/docs/data-apis/understand-data/new-relic-data-types/>

Event名について

- データ(Event)は種類に応じたEvent名が割り振られています
- 一例(他にも多数あります)

データソース	Event名	データの種類
APM	Transaction	トランザクションの所要時間を記録
	TransactionError	アプリで発生したエラーを記録
Browser	PageView	ページがロードされた際の所要時間を記録
	JavaScriptError	フロントエンドのエラーを記録
Infrastructure (クラウド連携)	FinanceSample	AWS 請求データを記録
	TrustedAdvisorSample	AWS Trusted Advisor によるリソースのガイダンスデータを記録

参考: New Relic の機能によって報告されるデフォルトのイベント

<https://docs.newrelic.com/jp/docs/data-apis/understand-data/event-data/default-events-reported-new-relic-products/>

属性(Eventの構成要素)について

- Eventデータは複数の属性を持つJSON形式のデータになっています。
- 属性名はエージェントによって事前定義されているものもありますが、任意の属性を追加で送ることもできます。



```
Transaction
{
  "results": [
    {
      "events": [
        {
          "appId": 1084400303,
          "appName": "EC-site",
          "databaseCallCount": 2,
          "databaseDuration": 0.00019,
          "duration": 0.06434,
          "entity.guid": "Mzk0MDcxNnxBUE18QVB0TE1DQVRJT058MTA4NDQwMDMhMw",
          "entityGuid": "Mzk0MDcxNnxBUE18QVB0TE1DQVRJT058MTA4NDQwMDMhMw",
          "error": false,
          "guid": "60b9b772435240a5",
          "host": "ip-172-31-26-144.ap-northeast-1.compute.internal",
          "http.statusCode": 200,
          "httpResponseCode": "200",
          "name": "WebTransaction/Action/block_search_product",
          "priority": 1.82538,
          "realAgentId": 1084405260,
          "request.headers.accept": "text/html,application/xhtml+xml,application/json;q=0.9,application/javascript;q=0.9,text/javasc",
          "request.headers.host": "ec2-3-113-215-132.ap-northeast-1.compute.amazonaws.com",
          "request.method": "GET",
          "request.uri": "/ec-cube/index.php/",
          "response.headers.contentType": "text/html",
          "response.statusCode": 200,

```

参考: カスタム属性の収集

属性(Eventの構成要素)について

Transaction

```
{
  "results": [
    {
      "events": [
        {
          "appId": 1084400202,
          "appName": "EC-site",
          "databaseCallCount": 2,
          "databaseDuration": 0.00019,
          "duration": 0.06434,
          "entity.guid": "Mzk0MDcxNnxBUE18QVBQTElDQVRJT058MTA4NDQwMDMwMw",
          "entityGuid": "Mzk0MDcxNnxBUE18QVBQTElDQVRJT058MTA4NDQwMDMwMw",
          "error": false,
          "guid": "60b9b772435240a5",
          "host": "ip-172-31-26-144.ap-northeast-1.compute.internal",
          "http.statusCode": 200,
          "httpResponseCode": "200",
          "name": "WebTransaction/Action/block_search_product",
          "priority": 1.02538,
          "realAgentId": 1084405260,
          "request.headers.accept": "text/html,application/xhtml+xml,application/json;q=0.9,application/javascript;q=0.9,text/javasc",
          "request.headers.host": "ec2-3-113-215-132.ap-northeast-1.compute.amazonaws.com",
          "request.method": "GET",
          "request.uri": "/ec-cube/index.php/",
          "response.headers.contentType": "text/html"
```

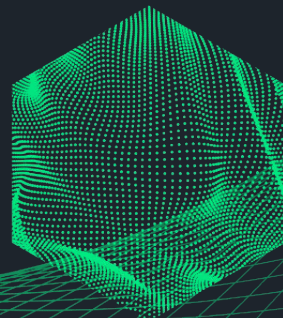
アプリケーション名は"appName"という属性

Transactionの応答時間は"duration"という属性

Transactionの名前は"name"という属性

データの理解

NRU302 - ハンズオン #1



ハンズオン - データの理解

このハンズオンでは、以下の点を学習します。

- New Relic プラットフォームにログインする
- New Relic の様々なUIに触れてみる
- Metrics & Events を活用して、収集したEventデータを参照する
- Option!! ● カスタムイベントを送信し、参照する



new relic®

15:20 - 15:40 (20min)

p. 25 - p. 37

New Relic ヘログイン

New Relic ポータル にログインします

new relic

Log in to see your data

② Email

Next

Log in to see your data

Email

③ Password

☐ Remember my email ?

Log in

new relic.

All Entities

Quick Find

+ Add Data

All Capabilities

All Entities

Alerts & AI

APM & Services

Browser

Dashboards

Errors Inbox

Infrastructure

Logs

Metrics & Events

Query Your Data

Service Levels

Synthetic Monitoring

Search by entity

See which parts of your system are unmonitored.
[View 1 gaps](#)

All entities (173)

Last viewed (7)

Coverage gaps (1)

Your system

Services - APM (1)

Hosts (1)

Containers (0)

Mobile applications (0)

Browser applications (0)

New Relic UI 対応ブラウザは「ハンズオンで使用するブラウザについて」を参照してください。



手順

- ① ブラウザで「<https://one.newrelic.com>」にアクセスします
- ② [Email] に「japan-handson+nru@newrelic.com」を入力し、[Next] をクリックします
- ③ [Password] に「[oSz6nrupas](#)」を入力し、[Login] をクリックします
(オー、エス、ゼット、ログ、エヌ、アール、ユー、ビー、エー、エヌ)

ログインユーザーの確認

ログインしたユーザーが正しいことを確認します

The screenshot shows the New Relic 'All Entities' page. The left sidebar contains navigation links: Quick Find, Add Data, All Capabilities, All Entities (selected), Alerts & AI, APM & Services, Browser, Dashboards, Errors Inbox, Infrastructure, Logs, Metrics & Events, Query Your Data, Service Levels, Synthetic Monitoring, and a Help link. The main content area is titled 'All Entities' and includes a search bar and filters. Below the filters, there are sections for 'Services - APM' and 'Hosts'. A callout box labeled '4' points to the 'Add User' and 'New Relic University Japan' options in the bottom left sidebar.

Name ↑	Response time (ms)	Throughput	Error rate
★ EC-site	118 ms	1.09 krpm	0%

Agent ve...	CPU usa...	Memory ...	Storage ...	Network ...	Network ...
-26-144.ap-northeast-1.comp...	1.51.0	6.41%	58.19%	44.35%	1.12 MB/s

Through...	Largest c...	Interacti...	Errors	Payload...	Ajax thro...
9.4 page...	463 ms	-	-	866 ms	1.77 rpm

④ 画面左下のユーザー名が「New Relic University Japan」であることを確認します



手順

メニューの操作

New Relic ポータルの UI を確認します

⑤

The screenshot displays the New Relic portal interface. On the left, a sidebar menu is visible with a green box highlighting the following items: Alerts & AI, APM & Services, Browser, Dashboards, Errors Inbox, Infrastructure, Logs, Metrics & Events, Query Your Data, Service Levels, and Synthetic Monitoring. The main content area is titled 'All Entities' and includes a search bar, filters, and a list of entity types. The 'Services - APM' section is expanded, showing a table with columns: Name, Response time (ms), Throughput, and Error rate. The 'Hosts' section is also expanded, showing a table with columns: Name, Agent ve..., CPU usa..., Memory ..., Storage ..., Network ..., and Network ... The 'Browser applications' section is expanded, showing a table with columns: Name, Through..., Largest c..., Interacti..., Errors, Pageload..., and Ajax thro... The 'Synthetic' section is partially visible at the bottom.

⑤ メニューバーの中からいくつかのメニューをクリックし、どのような画面が表示されるか確認します

- APM & Services
- Infrastructure
- Dashboards
- Logs



手順

メニューのピン留め

よく使うメニューのピン留めができます

⑥

new relic.

Quick Find

+ Integrations & Agents

All Capabilities

All Entities

AI Monitoring

Dashboards

Alerts

Metrics & Events

APM & Services

Infrastructure

Logs

Apps

Browser

Synthetic Monitoring

Service Levels

...

Help 21

Add User

New Relic University Japan

All Capabilities

Search capabilities...

Browse all our tools, views, and features and pin your favorites to the side menu.

AI Monitoring New APM for AI: Adopt your AI ecosystem with comprehensive...	Alerts & AI Set up robust alerts with intelligent correlation & analysis.	APM & Services Monitor every aspect of your applications and services.	Apps Add or build apps to extend New Relic.
AWS Lambda Setup Send your Amazon CloudWatch logs to New Relic.	Browser Understand what's affecting your end-user experience.	Dashboards Build and share tailored, user-friendly charts & visualizations.	Errors Inbox Proactively detect, triage, and fix all your errors.
IAST New Find exploitable vulnerabilities with interactive application security...	Infinite Tracing Settings Analyze all your application traces with tail-based sampling.	Infrastructure Monitor all your cloud, host, and container-based services.	Key Transactions Watch the application transactions that matter to you.
Kubernetes Get a full view of your containers, clusters, and pods.	Logs Connect your logs with the rest of your data in one place.	Lookout Discover potential issues with a high-level view of your systems.	Metrics & Events Navigate all your data visually, without any NRQL knowledge.
Mobile Get a comprehensive view of your mobile app performance.	Model Performance Monitor machine learning model issues in production.	Network See how your network is affecting performance.	Query Your Data Run queries of your data to create custom charts and other...
Recommendations	Saved Views New	Serverless Functions	Service Levels

本ハンズオンでは次の機能を利用します

- ・Alerts & AI
- ・Dashboards
- ・Query Your Data
- ・Metrics & Events



手順

- ⑥ メニューから **[All Capabilities]** をクリックします
- ⑦ メニューにピン留めしたい機能の **[Pin]** をクリックします

イベントデータの確認 (JSON)

Data Explorer を使用して Transaction イベントにどのようなデータがあるか確認します

The screenshot shows the New Relic Data Explorer interface. On the left sidebar, the 'Metrics & Events' menu item is highlighted with a green box and a circled '8'. A blue arrow points from this menu item to the 'Events' tab in the 'Metrics & Events' section, which is also highlighted with a green box and a circled '9'. Below the 'Events' tab, the 'Event type' list shows 'Transaction' selected with a green box and a circled '10'. A dashed green arrow points from 'Transaction' to the 'Transaction' query results, which are displayed in JSON format. A green box and a circled '11' highlight the JSON output in the 'Transaction' section.

```
NRQL SELECT * FROM Transaction SINCE 30 MINUTES AGO LIMIT 100
```

```
{
  "results": [
    {
      "events": [
        {
          "appId": "1084400303",
          "appName": "EC-site",
          "databaseCallCount": 2,
          "databaseDuration": 0.00023,
          "duration": 0.05816,
          "entityGuid": "Mzk0MDcxNnxBUE18QVBQTElQVJRJT058MTA4NDQwMDMwMw",
          "error": false,
          "guid": "db2531c4de823106",
          "type": "Transaction"
        }
      ]
    }
  ]
}
```



手順

⑧ メニューから [Metrics & Events] をクリックします

⑨ スコーピングセクションから [Events] をクリックします

⑩ Event type から [Transaction] を選択します

⑪ チャートピッカーから [JSON] を選択し、どのようなデータが表示されるか確認します

イベントデータの確認 (RAW)

Data Explorer を使用して Transaction イベントにどのようなデータがあるか確認します

Metrics & Events

Metrics **Events** Filter to All entities

30 minutes

Event type

Plot

Search...

count(*)

databaseCallCount Average

databaseDuration Average

duration Average

error Average

http.statusCode Average

parent.transportDuration Average

priority Average

request.headers.contentLength Average

response.statusCode Average

sampled Average

Dimensions

Select From Group by Limit Where

count(*) Transaction 10

NRQL SELECT * FROM Transaction SINCE 30 MINUTES AGO LIMIT 100

Transaction

Timestamp	App Id	App Name	Database Call Count	Database Duration	Duration	Entity.Guid
April 02, 2024 13:16:26	1084400303	EC-site	2	0.00079	0.111	Mzk0MDcxNn:
April 02, 2024 13:16:26	1084400303	EC-site	2	0.00023	0.0993	
April 02, 2024 13:16:26	1084400303	EC-site	2	0.00063	0.109	
April 02, 2024 13:16:26	1084400303	EC-site	7	0.00407	0.168	
April 02, 2024 13:16:26	1084400303	EC-site	7	0.00265	0.0789	
April 02, 2024 13:16:26	1084400303	EC-site	7	0.00303	0.0943	
April 02, 2024 13:16:26	1084400303	EC-site	5	0.00131	0.0865	
April 02, 2024 13:16:26	1084400303	EC-site	5	0.00356	0.119	
April 02, 2024 13:16:26	1084400303	EC-site	5	0.0014	0.118	



手順

⑫ チャートピッカーから [RAW] を選択します

⑬ データブラウジングエリアの [Plot] にどのような数字属性があるか確認します

⑭ [Dimensions] にどのような文字属性があるか確認します

Option!!

カスタムイベントを送信する

NRU302 - ハンズオン

Option!! APIキーの作成

カスタムイベントを送信する際の APIキーを作成します

The screenshot illustrates the process of creating an API key in the New Relic console. It shows the navigation from the user menu to the API Keys section, then to the 'Create a key' form. The form includes fields for 'Key type' (set to 'Ingest - License') and 'Name' (set to 'my_key_name'). A red callout box explains that the name should be unique to avoid conflicts with other participants, providing the example 'NRU302-企業名-イニシャル' (e.g., NRU302-NEWRELIC-NR). The 'Create a key' button is highlighted at the bottom of the form.

① ユーザーメニューから [New Relic University Japan] - [API Keys] をクリックします

② [Create a key] をクリックします

③ [Key Type] から「Ingest - License」を選択し、[Name] に一意となる名前を入力します

④ [Create a key] をクリックし、APIキーを作成します

[Name] は他の参加者との重複を避けるため次の形式で入力してください

NRU302-企業名-イニシャル
e.g. NRU302-NEWRELIC-NR

参考: Event APIを使用したカスタムイベントの送信

<https://docs.newrelic.com/jp/docs/data-apis/ingest-apis/event-api/introduction-event-api/>

Option!! APIキーの取得

前のステップで作成したAPIキーを取得(クリップボードにコピー)します

☐	Acco... ▾	Acco... ▾	Type ▾	Name ▾	Value ▾	Notes	User ▾
☐	New Re...	3940716	INGEST - LICENSE	[DONT ...	13c29ed1*****	Automa...	...
☐	New Re...	3940716	INGEST - LICENSE	[DONT ...	5c65868c*****	Hands...	...
☐	New Re...	3940716	INGEST - LICENSE	License...	7ddce822*****		
☐	New Re...	3940716	INGEST - LICENSE	[DONT ...	7f483d23*****		
☐	New Re...	3940716	INGEST - LICENSE	[DONT ...	c500fc3b*****		

⑤

⑥

Copy key
Copy key ID
Edit
Delete

参考:

APIキーにはOriginalのKeyが作成されていますが、セキュリティの観点から、新たに作成したKeyを活用することをお勧めいたします。



手順

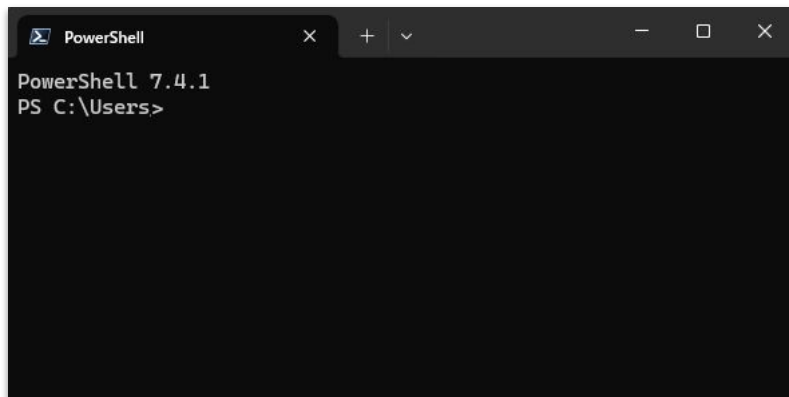
⑤ 作成したAPIキー右側の [●●●] をクリックします

⑥ [Copy key] をクリックし、APIキーをクリップボードにコピーします

Option!! ターミナルの起動

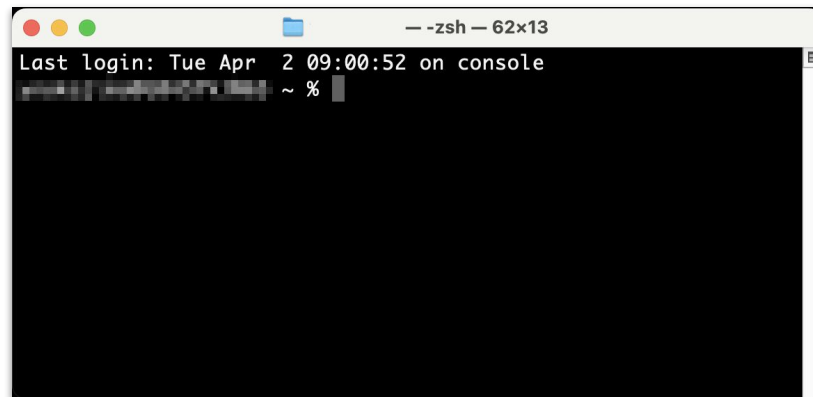
カスタムイベントを送信するため、OSのターミナルアプリを起動します

Windows - PowerShell



A screenshot of a Windows PowerShell terminal window. The title bar shows 'PowerShell' with standard window controls. The terminal content displays 'PowerShell 7.4.1' and 'PS C:\Users>' on a black background with white text.

Mac - ターミナル



A screenshot of a Mac Terminal window. The title bar shows '— zsh — 62x13' with standard window controls. The terminal content displays 'Last login: Tue Apr 2 09:00:52 on console' and a prompt '~ %' on a black background with white text.



手順

⑦ ご利用のOSのターミナルアプリケーションを起動します

Windows: PowerShell

Mac: ターミナル

Option!! カスタムイベントの送信 (Windows)

WindowsのPowerShellからカスタムイベントを送信します

[GitHub - script.ps1](#)

```
$accountId = "3940716"
$insertkey = "{APIキー}"
# Replace with your custom event for the body
$body = '["eventType":"NRULab", "labnumber":1, "initial":"{ご自身のイニシャル}", "comment":"{任意の文字列 (日本語可)}"]'

$headers = @{}
$headers.Add("X-Insert-Key", "$insertkey")
$headers.Add("Content-Encoding", "gzip")

$encoding = [System.Text.Encoding]::UTF8
$enc_data = $encoding.GetBytes($body)
$output = [System.IO.MemoryStream]::new()

$gzipStream = New-Object System.IO.Compression.GzipStream $output, ([IO.Compression.CompressionMode]::Compress)
$gzipStream.Write($enc_data, 0, $enc_data.Length)
$gzipStream.Close()
$gzipBody = $output.ToArray()

Invoke-WebRequest -Headers $headers -Method Post -Body $gzipBody "https://insights-collector.newrelic.com/v1/accounts/$accountId/events"
```



手順

- ⑧ 上のスクリプト内の次のパラメータを変更します
- ・{APIキー}: [手順⑥](#) でコピーしたAPIキー
 - ・{ご自身のイニシャル} および {任意の文字列}: 任意
- ⑨ 「**script.ps1**」として保存し、PowerShellから実行します

Option!! カスタムイベントの送信 (Mac)

Macのターミナルからカスタムイベントを送信します

[GitHub - nrulab.json](#)

```
[
{
  "eventType": "NRULab",
  "labnumber": 1,
  "initial": "{ご自身のイニシャル}",
  "comment": "{任意の文字列 (日本語可)}"
}
```

[GitHub - script.sh](#)

```
gzip -c nrulab.json | curl --data-binary @- -X POST -H "Content-Type: application/json" -H "X-Insert-Key: {APIキー}" -H "Content-Encoding: gzip"
https://insights-collector.newrelic.com/v1/accounts/3940716/events
```



手順

⑧ 上のJSON内の次のパラメータを変更し、「**nrulab.json**」として保存します

- ・{ご自身のイニシャル} および {任意の文字列}: 任意

⑨ コマンド内の次のパラメータを変更し、ターミナルから実行します (nrulab.json が保存されたディレクトリで実行)

- ・{APIキー}: [手順⑥](#) でコピーしたAPIキー

Option!! カスタムイベントの確認

カスタムイベントが正しく送信されていることを確認します

The screenshot shows the New Relic web interface. On the left, the 'Metrics & Events' menu item is highlighted with a green box and labeled ⑩. In the main panel, the 'Events' tab is selected with a green box and labeled ⑪. Under the 'Event type' section, 'NRULab' is selected with a green box and labeled ⑫. On the right, the 'NRQL' query editor shows a query: 'NRQL SELECT * FROM NRULab SINCE 30 MINUTES AGO LIMIT 100'. Below the query, a table of results is displayed. The first row is highlighted with a green box and labeled ⑬. The table has columns: Timestamp, Comment, Initial, and Labnumber. The first row contains the values: April 02, 2024 18:47:14, NRU302ハンズオン, NR, and 1.

Timestamp	Comment	Initial	Labnumber
April 02, 2024 18:47:14	NRU302ハンズオン	NR	1



手順

- ⑩ メニューから [**Metrics & Events**] をクリックします
- ⑪ スコーピングセクションから [**Events**] をクリックします
- ⑫ Event type から [**NRULab**] を選択します
- ⑬ チャートピッカーの [**Raw**] をクリックしデータを確認します

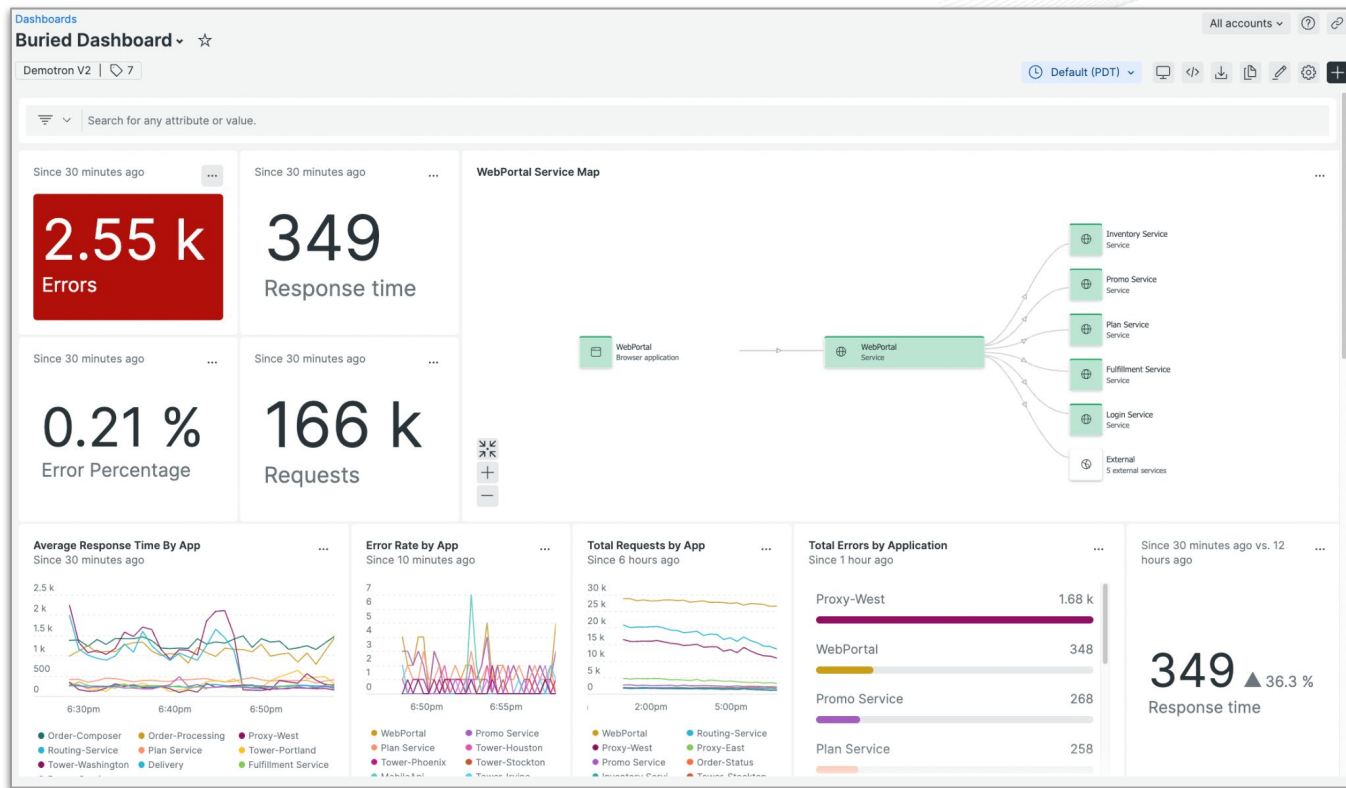
New Relic ダッシュボード

NRU302 - 座学

STEP2

ダッシュ
ボードの
作成

New Relic ダッシュボード



ダッシュボード活用シーン

- みんなが見えるようにずっとモニタに映しておく
- (ご紹介したウェザーニューズ社様の事例)
- キャンペーン中のリアルタイムな状況把握のために使う
- カスタマーサポートが問い合わせに応じて状況確認のために使う
- サービスに関わるチーム間ミーティングで共通認識を得るために使う
- 定期報告書代わりに使う

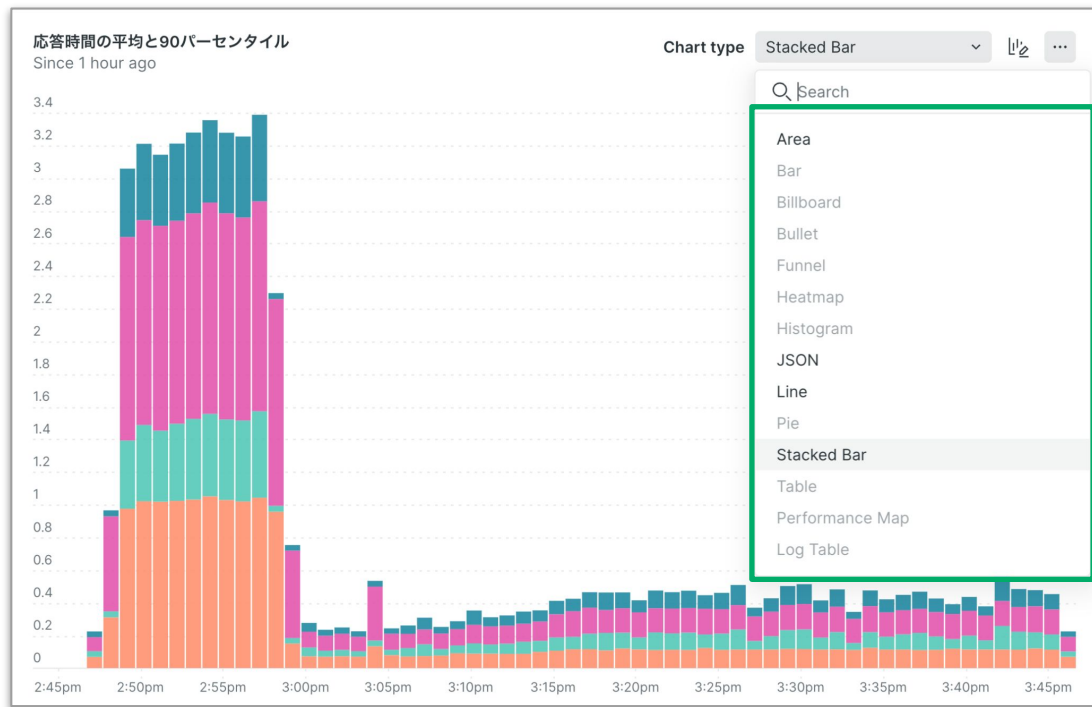
など

ダッシュボードが得意とすること

- データの選択と集約
 - 複数アプリケーションやアプリケーションとインフラのメトリックの相関関係など、様々なソースからのデータを一つの画面で把握したい場合
- データの加工
 - チームで定めたKPIに対する実測値を把握したい場合
- データのビジュアライズ
 - 集めたデータを目で見てわかりやすい形式で表示したい場合

データのビジュアライズ

- 加工したデータを様々なチャートタイプで表示

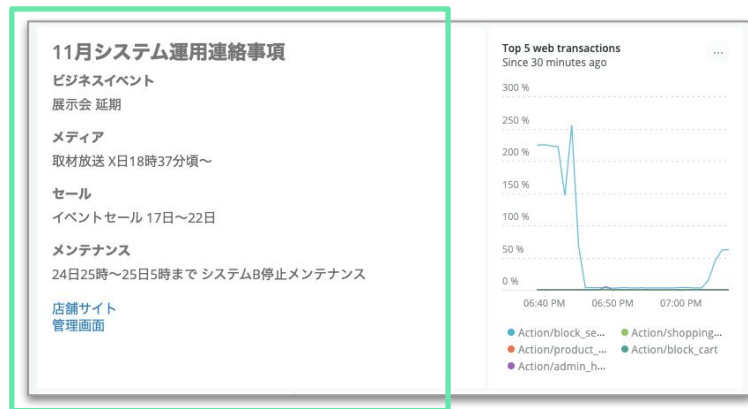


ダッシュボードの便利機能 ①

- 表示系

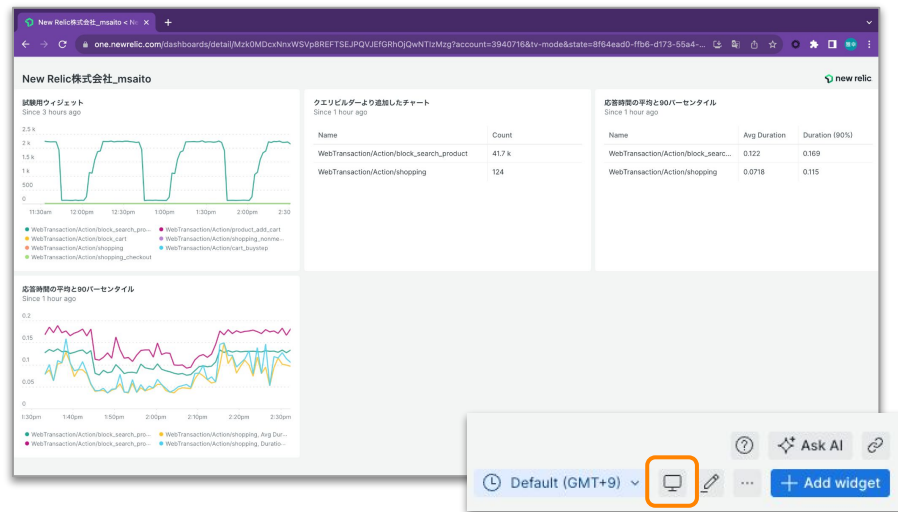
Note機能

任意の文字や画像をダッシュボードに埋め込む



TVモード

全画面表示にする
(オフィスのディスプレイに表示するなど)



ダッシュボードの便利機能 ②

- 分析系

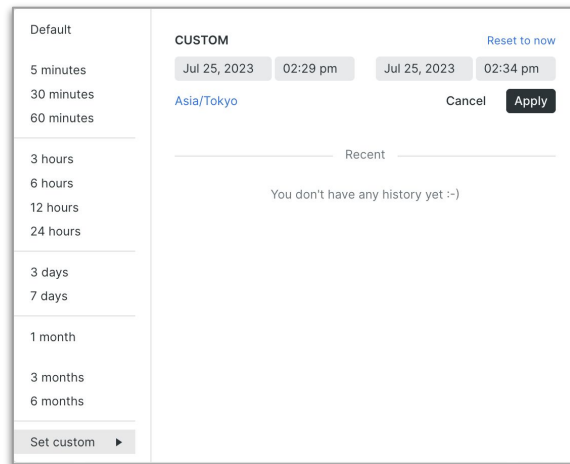
フィルタ機能

あるチャートで選んだ要素
に基づいて他のチャートの
情報が絞り込まれる

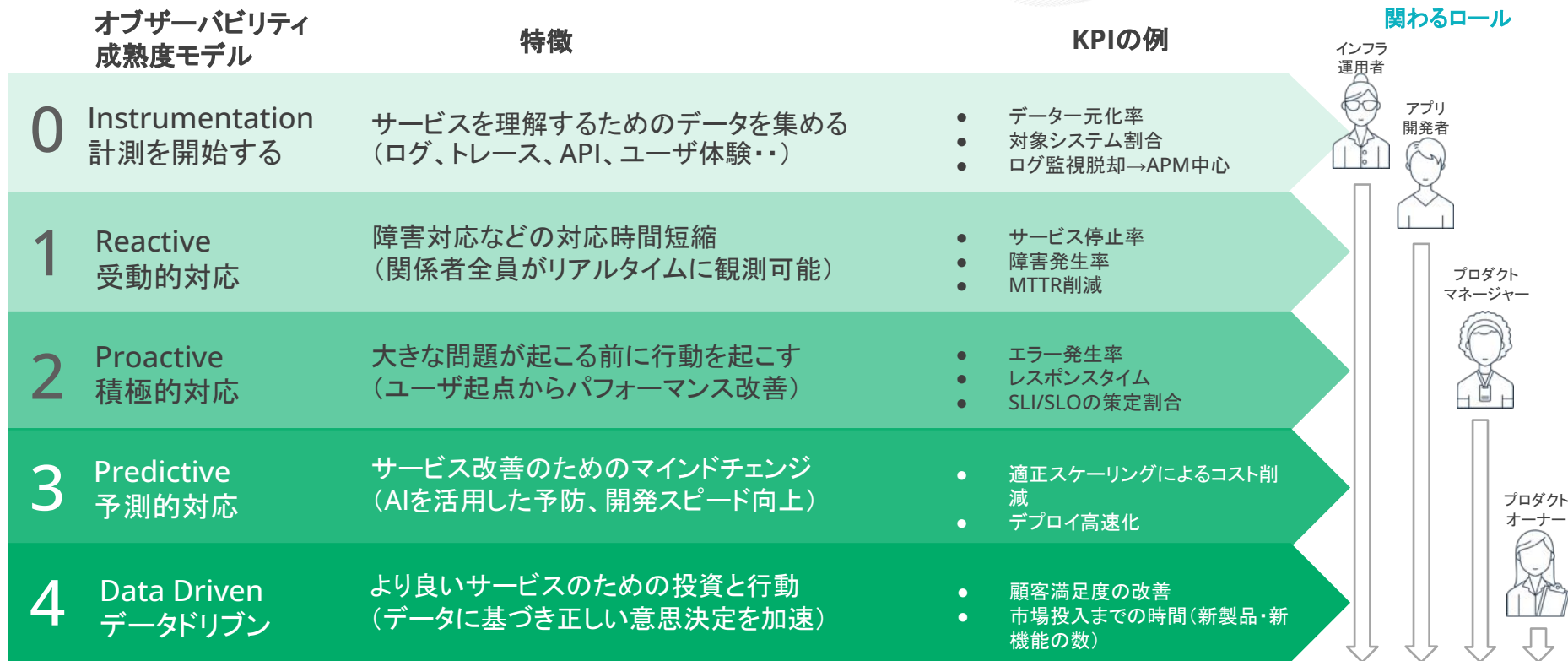


時間指定

任意の時間を指定する
(チャートからドラッグで指定すること
も可能)



New Relic オブザーバビリティ 成熟モデル



目的に応じたダッシュボード

1 受動的

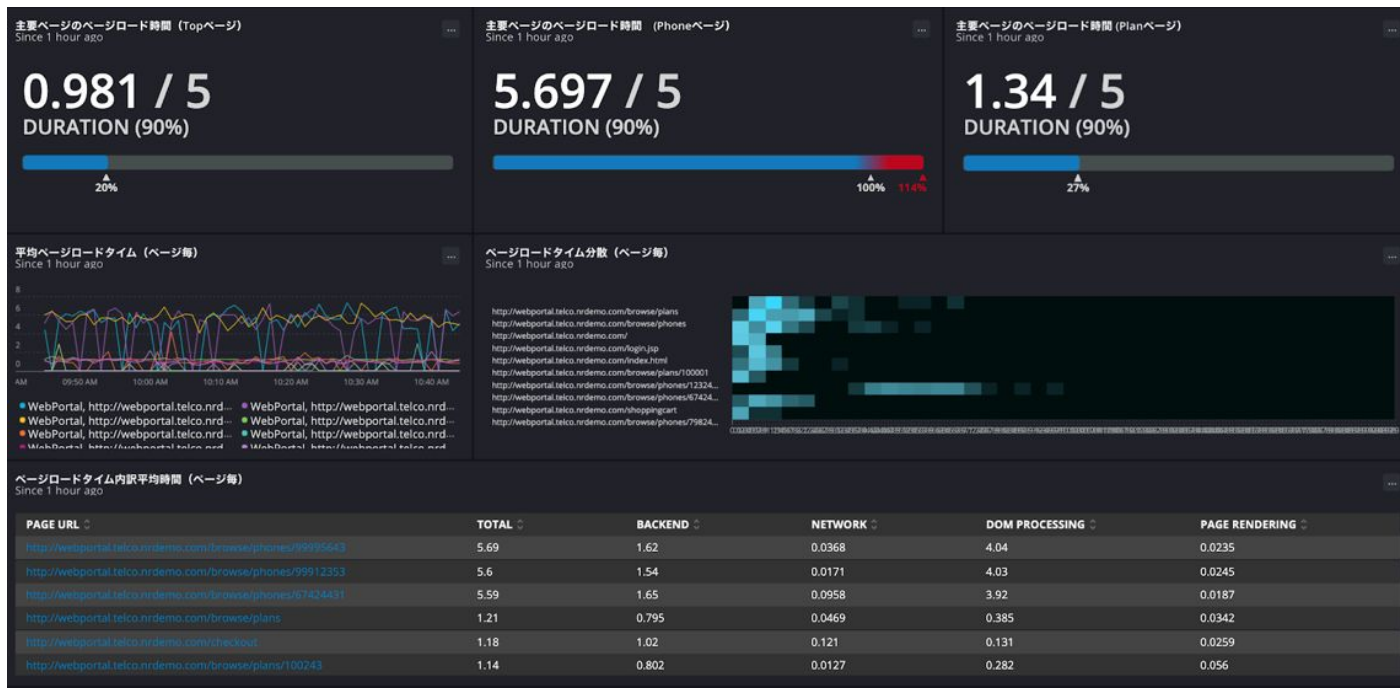
例1: サーバー稼働状況



目的に応じたダッシュボード

2 積極的

例2: ユーザー体験のリアルタイムモニタリング



目的に応じたダッシュボード

3 予測的

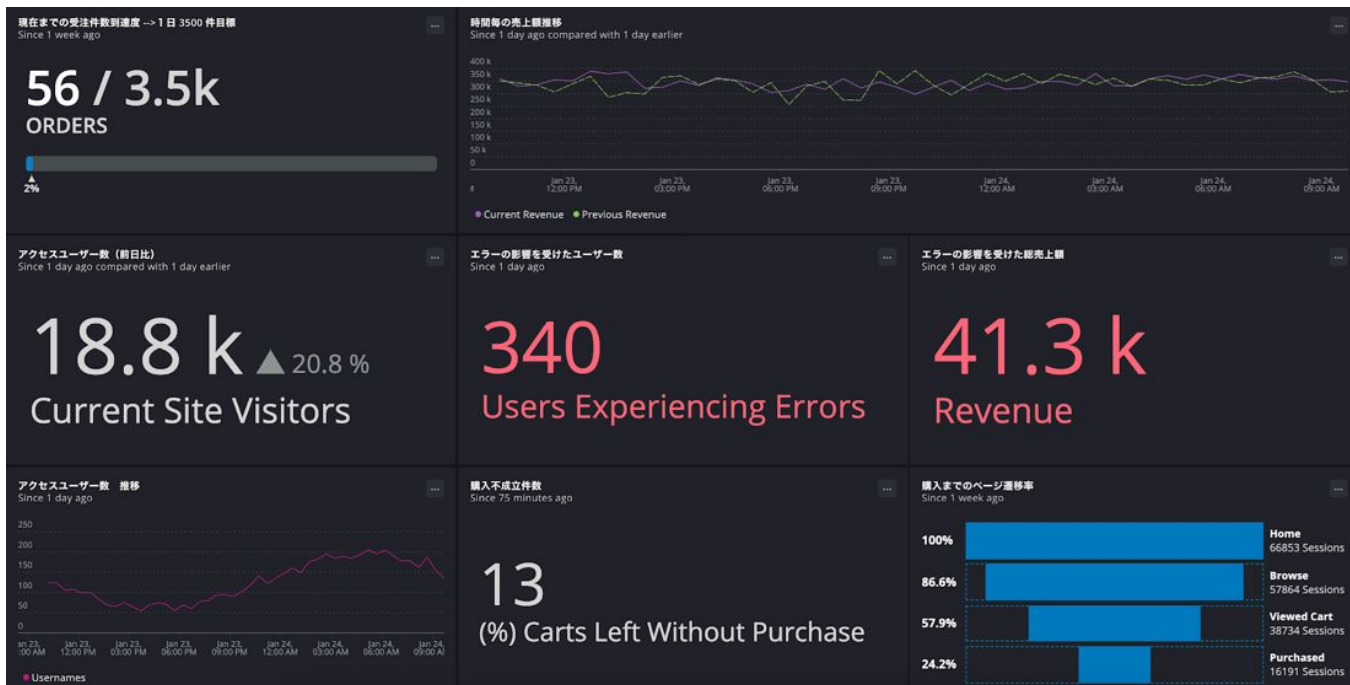
ECサイト購買分析



目的に応じたダッシュボード

4 データドリブン

例4: 売上データと連動したビジネスダッシュボード



さらに高度なビジュアライズも

- Reactを使ってリッチなダッシュボードを自分でカスタマイズできます
- 一部のダッシュボードはオープンソースとして公開しており自由に利用できます
 - <https://opensource.newrelic.com/nerdpacks/>

環境をセッティングして開発



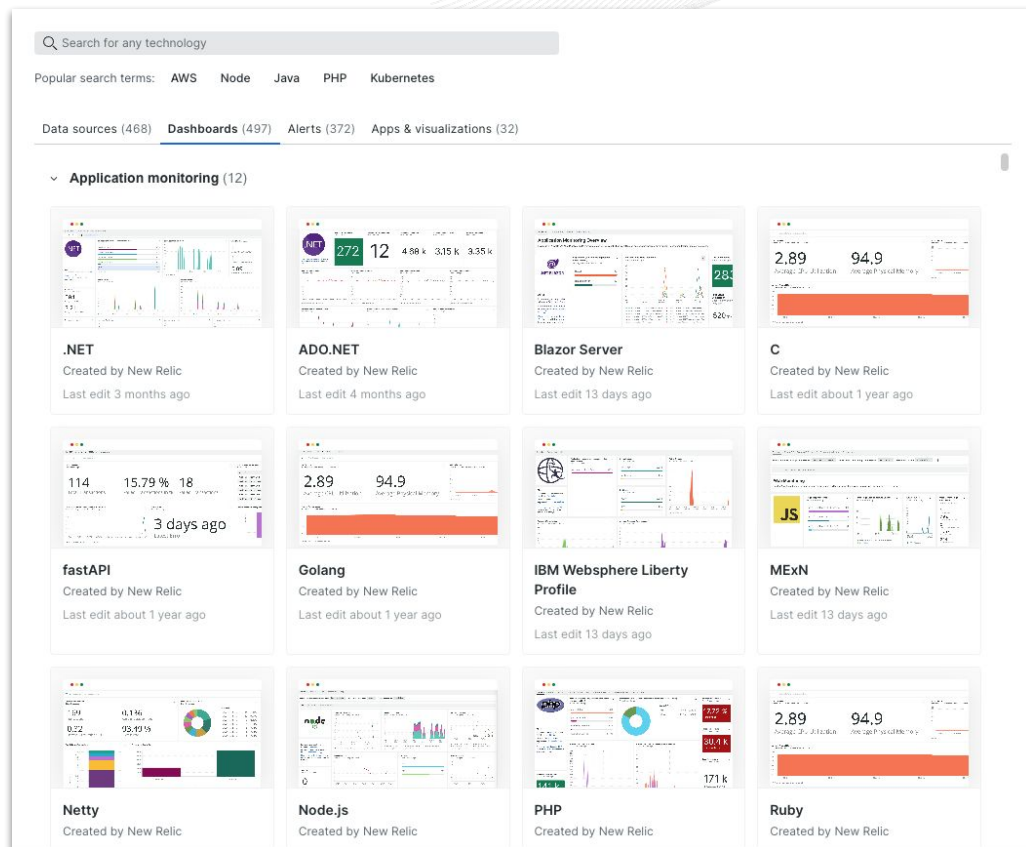
オリジナルのダッシュボードを作成可能



Pre-Build Dashboards

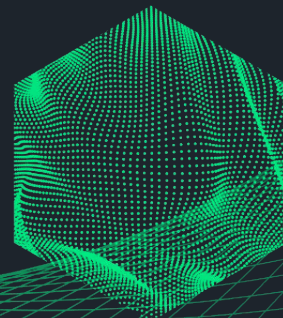
New Relicがあらかじめ準備している
ダッシュボードがあります。

- 言語
- ミドルウェア
- クラウド



ダッシュボードの 作成

NRU302 - ハンズオン #2



ハンズオン - ダッシュボードの作成

このハンズオンでは、以下の点を学習します。

- ダッシュボードの大枠を作る
- S3ダッシュボードのチャートを利用する
- Metrics & Events を用いて、チャートをダッシュボードに追加する
- チャートタイプを変更する
- フィルタ連携機能を利用する
- Option!! ● NOTE機能を利用する
- Option!! ● チャートのサイズや配置を変更する



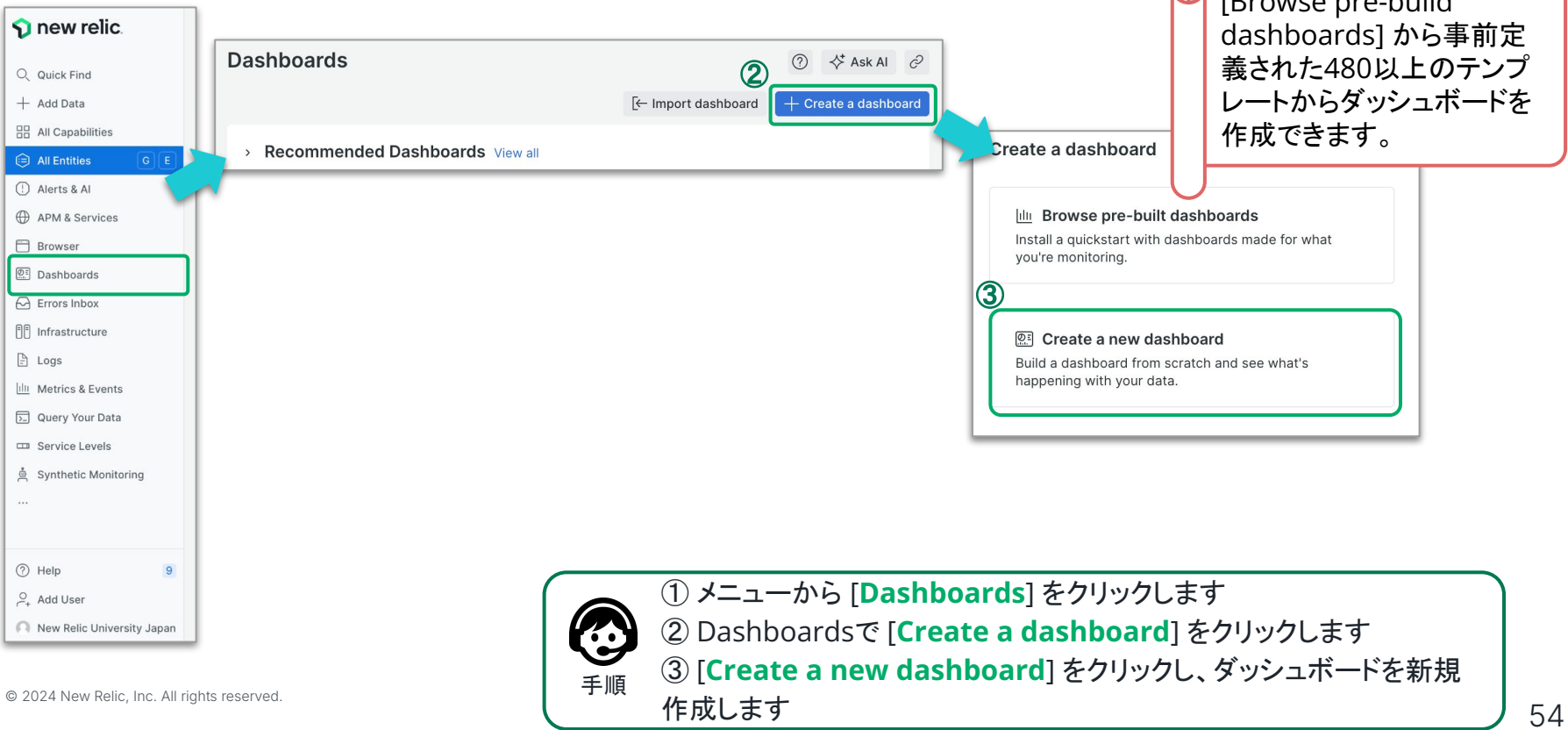
new relic®

15:50 - 16:10 (20min)

p. 54 - p. 66

ダッシュボードの作成

ダッシュボードの作成を通してデータの表示方法を学びます



①

new relic

Quick Find

Add Data

All Capabilities

All Entities

Alerts & AI

APM & Services

Browser

Dashboards

Errors Inbox

Infrastructure

Logs

Metrics & Events

Query Your Data

Service Levels

Synthetic Monitoring

Help

Add User

New Relic University Japan

Dashboards

Import dashboard

Create a dashboard

Recommended Dashboards

Create a dashboard

Browse pre-built dashboards

Install a quickstart with dashboards made for what you're monitoring.

Create a new dashboard

Build a dashboard from scratch and see what's happening with your data.

[Browse pre-build dashboards] から事前定義された480以上のテンプレートからダッシュボードを作成できます。

① メニューから **[Dashboards]** をクリックします

② Dashboardsで **[Create a dashboard]** をクリックします

③ **[Create a new dashboard]** をクリックし、ダッシュボードを新規作成します

手順

ダッシュボードの作成

ダッシュボードの名前と権限を設定します

④ Create a dashboard

Dashboard name

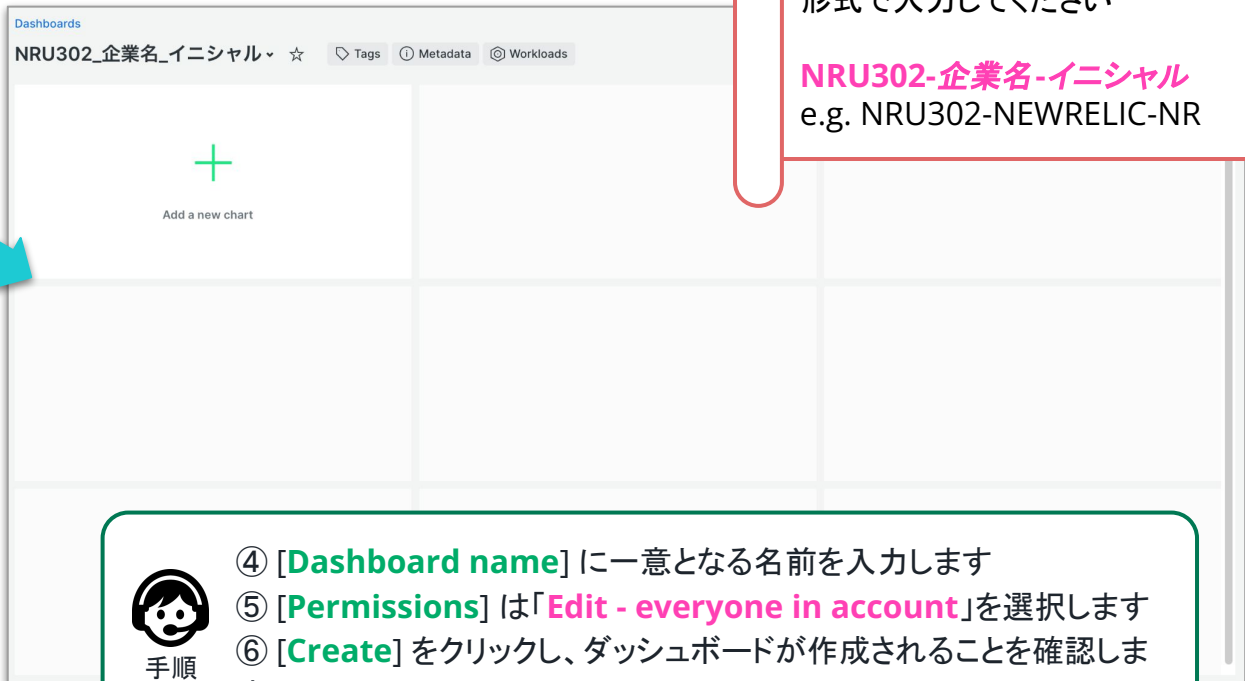
NRU302_企業名_イニシャル

⑤ Permissions ⓘ

Edit - everyone in account ▾

⑥

Back Create



[Dashboard name] は他の参加者との重複を避けるため次の形式で入力してください

NRU302-企業名-イニシャル
e.g. NRU302-NEWRELIC-NR



手順

- ④ **[Dashboard name]** に一意となる名前を入力します
- ⑤ **[Permissions]** は「**Edit - everyone in account**」を選択します
- ⑥ **[Create]** をクリックし、ダッシュボードが作成されることを確認します

Amazon S3 Dashboardの確認

既存チャートを Amazon S3 Dashboard から選択します

The screenshot shows the New Relic console interface. On the left, the 'Infrastructure' menu item is highlighted with a green box and a green arrow pointing to it from a green circle labeled '7'. In the center, the 'AWS' integration is selected, highlighted with a green box and a green arrow pointing to it from a green circle labeled '8'. On the right, the 'S3 dashboard' is highlighted with a green box and a green arrow pointing to it from a green circle labeled '9'. A large blue arrow points from the 'AWS' section towards the 'S3 dashboard'.

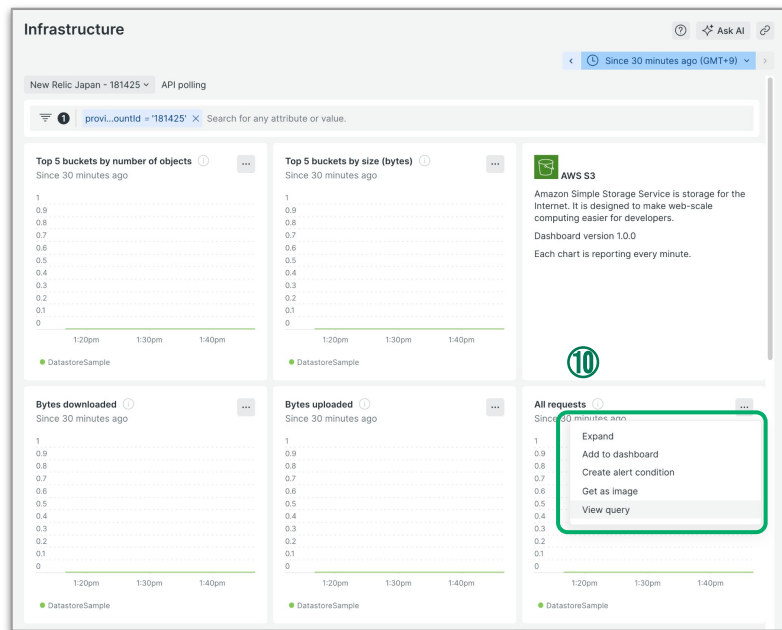
Integration name ↑	Alert	Dashboards	Documentation	Data	Configure
Billing	Set up alert	Billing (Costs) dashboard Billing (Budgets) dashboard	See our docs	Explore data	Configure
CloudTrail	Set up alert	CloudTrail dashboard	See our docs	Explore data	Configure
Health	Set up alert	Health dashboard	See our docs	Explore data	Configure
S3	Set up alert	S3 dashboard	See our docs	Explore data	Configure
Trusted Advisor	Set up alert	Service Limits dashboard	See our docs	Explore data	Configure
X-Ray	Set up alert		See our docs	Explore data	Configure

⑦ メニューから [Infrastructure] - [AWS] をクリックします
⑧ [Select a provider account] から「New Relic Japan - 181425」を選択します
⑨ [S3 Dashboard] をクリックします

手順

既存グラフのチャートを追加

Amazon S3 Dashboard 内のチャートをダッシュボードに追加します



Copy to a dashboard

Select a dashboard where you would like to add the widget.

Widget title

All requests

Select an existing dashboard

We excluded dashboards you don't have permission to edit.

Search by name, account or creator

Dashboard name Account

New Relic 株式会社 NRU302 Sam... New Relic...

New Relic 株式会社_NRU303 / Ne... New Relic...

New Relic 株式会社_NRU303 sam... New Relic...

New Relic株式会社_msaito / ハン... New Relic...

New Relic株式会社_msaito / ハン... New Relic...

NRU#ホワイトボードサンプル / NR... New Relic...

NRU#レポートサンプル / NRU#レ... New Relic...

test / test New Relic...

Cancel

Create a new dashboard

Copy



手順

⑩ All requests チャート右上の [...] - [Add to dashboard] をクリックします

⑪ 表示されたモーダルで先ほど作成したダッシュボードを選択し、[Copy]で追加します

カスタムチャートの作成

ダッシュボードに追加するチャートを **Metrics & Events** から作成します

⑫ new relic

Quick Find

+ Add Data

All Capabilities

All Entities

Alerts & AI

APM & Services

Browser

Dashboards

Errors Inbox

Infrastructure

Logs

Metrics & Events

Query Your Data

Service Levels

Synthetic Monitoring

Help

Add User

New Relic University Japan

Metrics & Events

Metrics Events Filter to All entities

Event type

Search

New Relic Events storage sample

SyntheticCheck

SyntheticRequest

SystemSample

Transaction

WorkloadStatus

Custom events

ApacheSample

ApplicationAgentContext

ComputeSample

Plot

Dimensions

Event type

Plot

Search

count(*)

databaseCallCount

databaseDuration

duration

error

http.statusCode

parent.transportDuration

priority

request.headers.contentLength

Average

Average

Average

Average

Average

Average

Average

Dimensions

Search

request.method

request.uri

response.headers.content...

response.statusCode

sampld

tags.account

tags.accountid

3

60

2

3

2

New Relic University Japan

3940716

Select

From

count(*)

Transaction

NRQL SELECT count(*) FROM Transaction FACET 'r

Transaction

800

600

400

200

0

11:25am

11:30am

/ec-cube/index.php/products/list

/ec-cube/index.php

/ec-cube/index.php/block/cart

/ec-cube/index.php

Request.Uri

/ec-cube/index.php/products/list

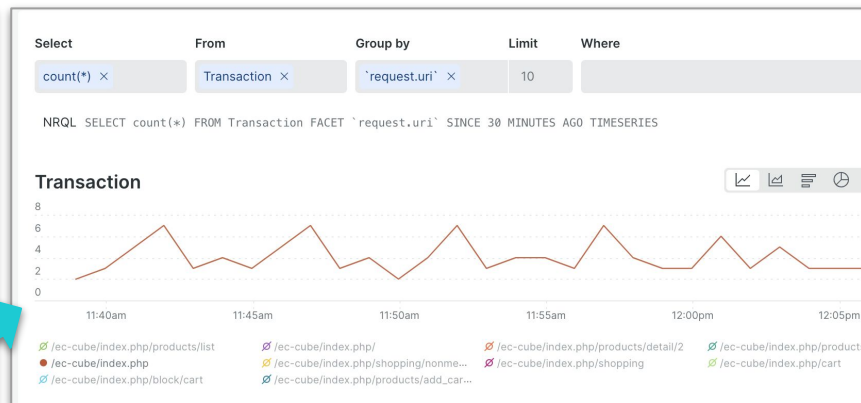
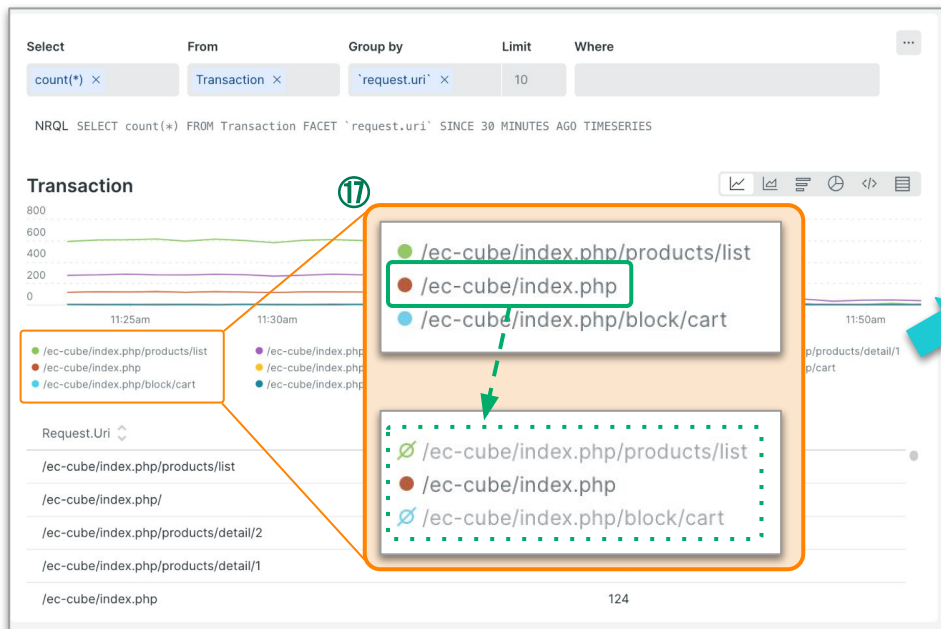


手順

- ⑫ メニューから **[Metrics & Events]** をクリックします
- ⑬ スコーピングセクションから **[Events]** をクリックします
- ⑭ Event type から **[Transaction]** を選択します
- ⑮ Plotから **[count(*)]** を選択します
- ⑯ Dimensionsから **[request.uri]** を選択します

カスタムチャートの確認 (1/2)

チャートにプロットされるエンティティを選択し表示を確認します

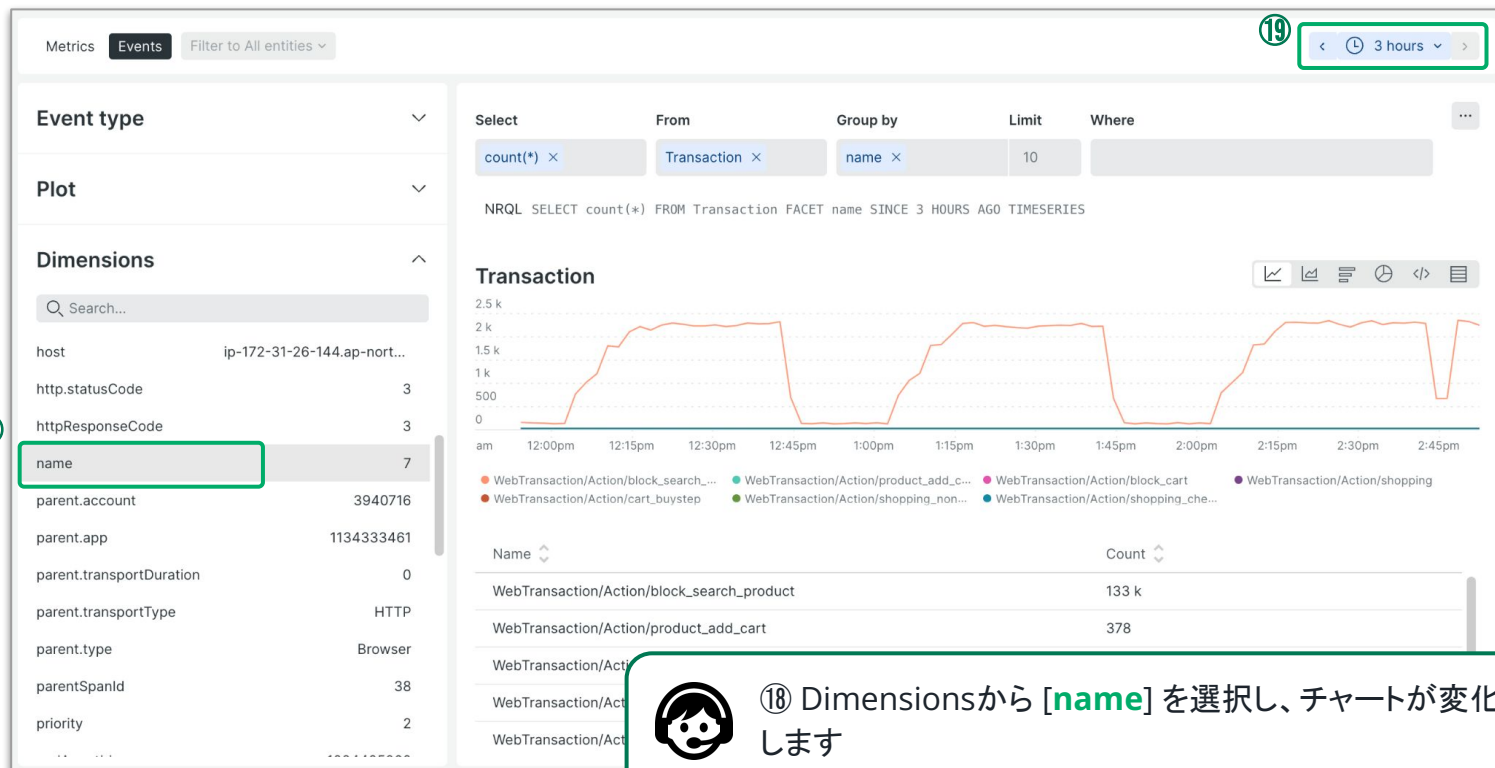


手順

① チャートの下に表示されている「/ec-cube/index.php」をクリックし、チャートの表示が変わることを確認します

カスタムチャートの確認 (2/2)

チャートの対象となる **Dimension** を選択し表示を確認します



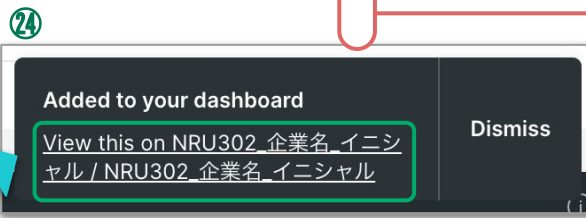
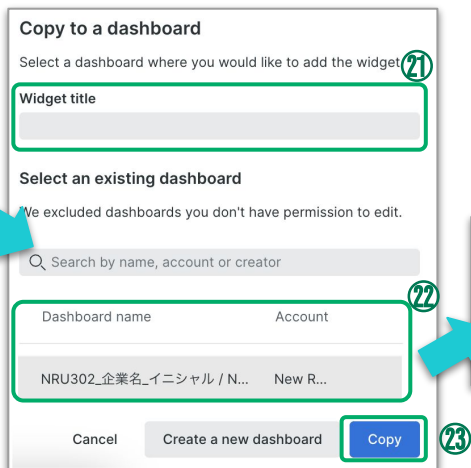
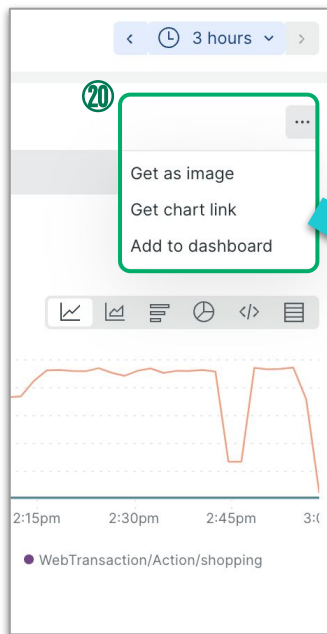
手順

18 Dimensionsから **[name]** を選択し、チャートが変化することを確認します

19 タイムピッカーから **[3 hours]** を選択します

チャートの追加

ご自身のダッシュボードにチャートを追加します



手順②④でリンクをクリックできなかった場合は、メニューの [Dashboards] から選択してください



手順

- ②① チャート右上の [⋮] - **[Add to dashboard]** をクリックします
- ②② **[Widget title]** に任意のチャート名を入力します
- ②③ **[Select an existing dashboard]** から**手順⑥** で作成したダッシュボードを選択します
- ②④ **[Copy]** をクリックします
- ②⑤ ポップアップメッセージから **[View this on ダッシュボード名]** のリンクをクリックしダッシュボードを表示します

チャートの複製

作成したチャートを複製しカスタマイズします

The first screenshot shows a line chart titled "Hands on 2-2" with a menu open, highlighting the "Duplicate" option (labeled 25). A blue arrow points to the second screenshot, which shows the same chart with the menu open, highlighting the "Edit" option (labeled 26). Another blue arrow points to the third screenshot, which shows the "Edit widget" screen with the "See customization options" button highlighted (labeled 27). The chart type is set to "Line" and the chart name is "Enter a chart name".



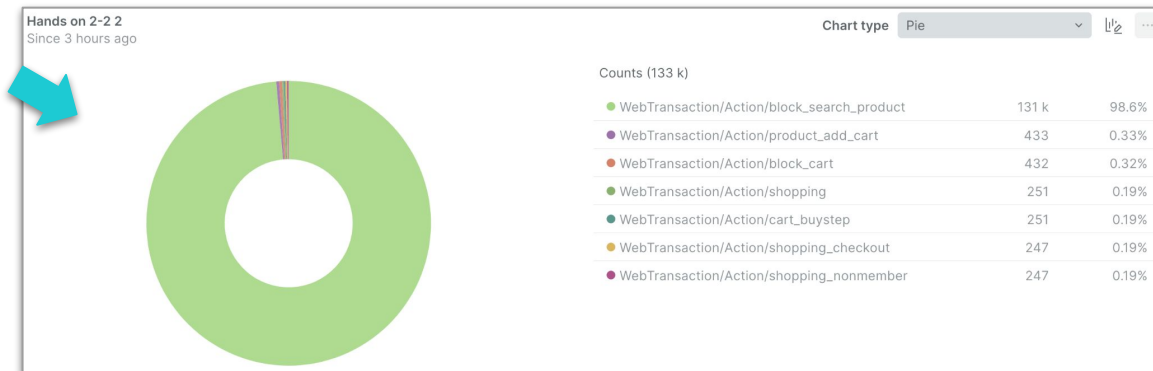
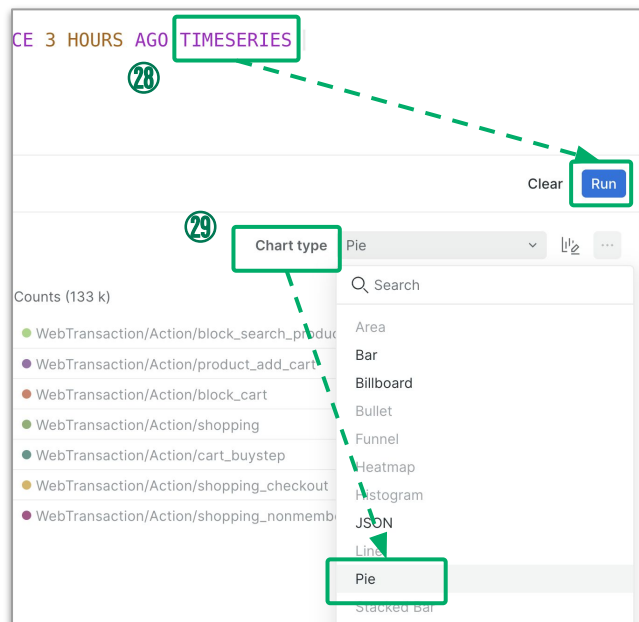
手順

- ②⑤ 追加したチャート右上の [...] - **[Duplicate]** をクリックします②⑥ 画面左にチャートが複製されますので、[...] - **[Edit]** をクリックします
- ②⑦ Edit widget画面の[See customization options] を表示させ **[Chart name]** にチャート名を入力します

複製後のチャートは元のチャートと同じ名前になるので変更しておきます

チャートタイプの変更

チャートのタイプを Line から Pie に変更します



手順

②⑧ 表示されているクエリ文字列から[TIMESERIES]を削除し[Run]を実行します

②⑨ [Chart Type] から「Pie」を選択し、パイチャート (円グラフ) に変更されることを確認します

ファセットフィルタリングの設定

FACET 句で指定した属性によるフィルタを行えるよう設定します

Basic information

Chart name

Chart type Pie

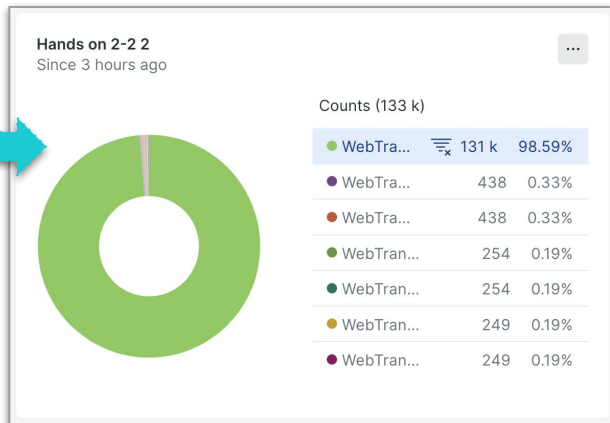
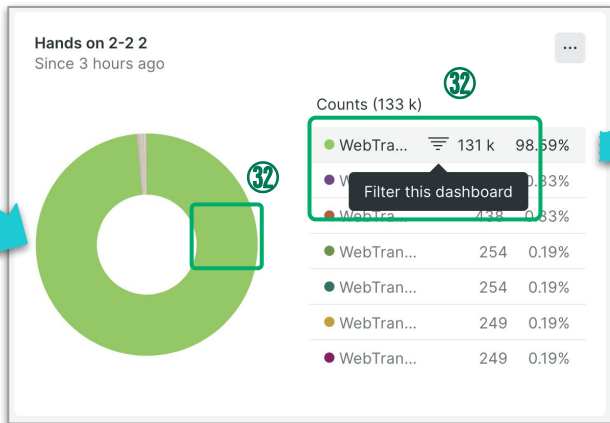
More visualizations in I/O [↗](#)

▼ **Facet Linking**

Filter the current dashboard ☒ ③①

Filter a related dashboard ☐ ①

Cancel Save as... Save ③①



手順

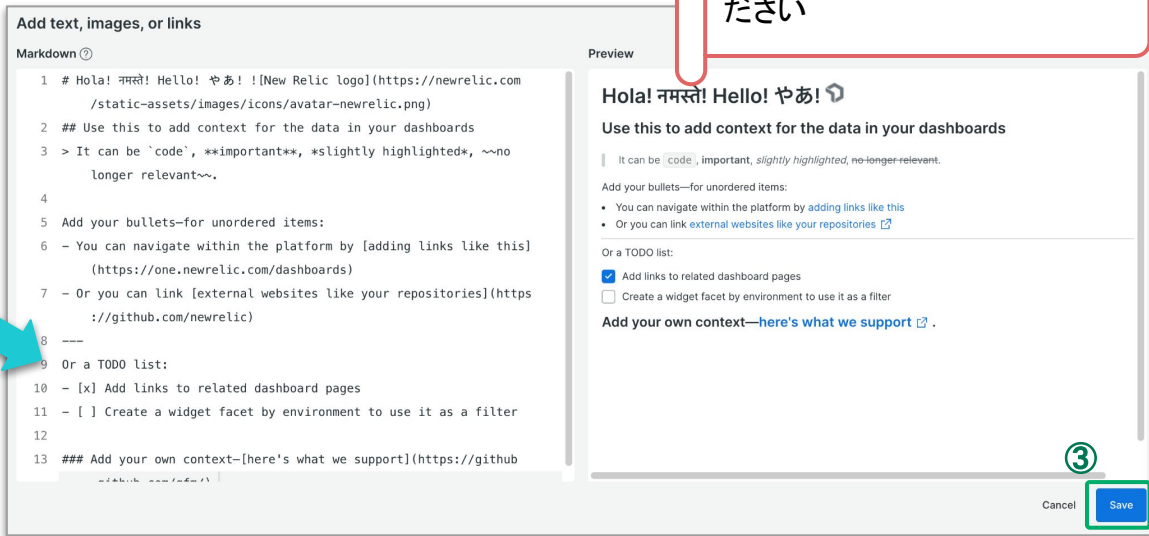
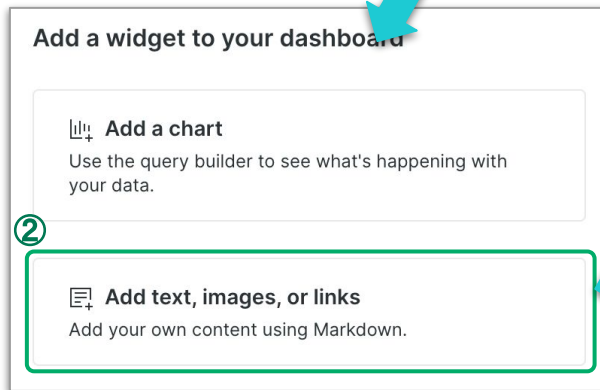
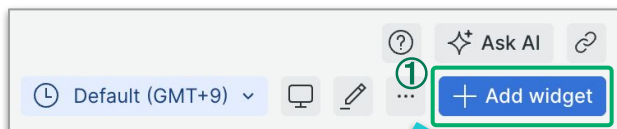
③① [Filter the current dashboard] を有効化 (☒)します

③① チャートを保存します

③② チャート、あるいは属性の **[Filter this dashboard]** をクリックすることにより、他のチャートが変化することを確認します

Option!! NOTE機能の利用

Markdown を利用してダッシュボードにテキスト・画像・リンクなどを追加します



参考: GtiHub マークダウンガイド

<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>

© 2024 New Relic, Inc. All rights reserved.



手順

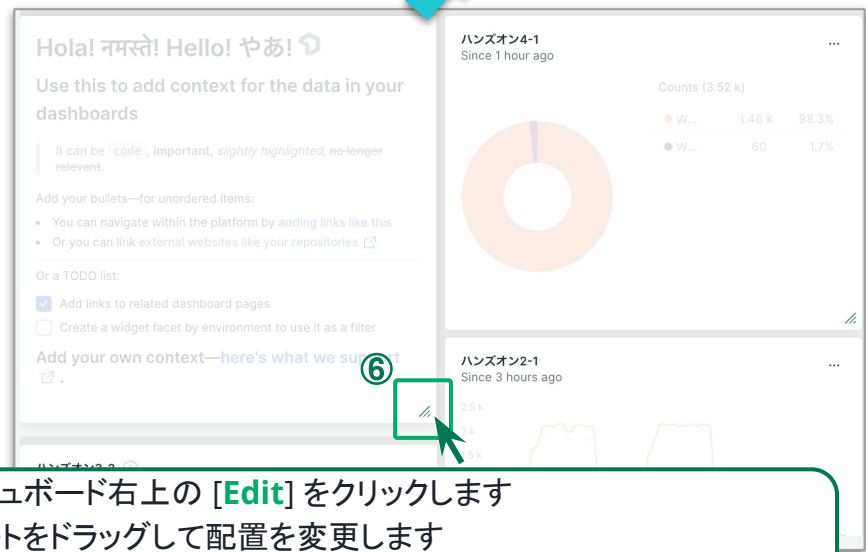
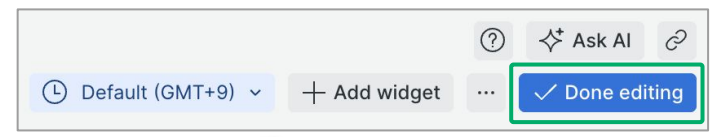
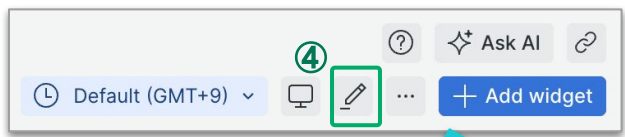
① ダッシュボード右上の [+ Add widget] をクリックします

② [Add text, images, or links] をクリックします

③ Markdownの編集画面が表示されることを確認し、[Save] をクリックします

Option!! チャートの配置やサイズの変更

ダッシュボード内のチャートを並び替えたりサイズを変更したりします



- ④ ダッシュボード右上の **[Edit]** をクリックします

⑤ チャートをドラッグして配置を変更します

⑥ チャートの右下をドラッグしてサイズを変更します

⑦ **[Done editing]** をクリックし、編集を完了します

STEP3

分析手法
の習得

NRQL

(New Relic Query Language)

NRU302 - 座学

NRQLとは

- New Relic Query Language の略
- Eventデータを含め、New Relicデータベース (NRDB) に格納されたデータを分析するためのクエリ言語

NRQLを直接書く

- 自分でクエリを書いて、見たい情報をチャートに表現
- 柔軟なデータ分析が可能



NRQL構文

```
SELECT function(attribute) [AS 'label'][, ...]  
FROM event  
[WHERE attribute [comparison] [AND|OR ...]][AS 'label'][, ...]  
[FACET attribute | function(attribute)] [LIMIT number]  
[SINCE time] [UNTIL time]  
[WITH TIMEZONE timezone]  
[COMPARE WITH time]  
[TIMESERIES time]
```

参考: New Relic の機能によって報告されるデフォルトのイベント

<https://docs.newrelic.com/jp/docs/nrql/get-started/introduction-nrql-new-relics-query-language/>

NRQL使いこなしTips ①

この句だけは覚えましょう！

- **FROM データタイプ名**：どの名前のデータタイプから情報を収集するか
- **SELECT 属性**：どの属性の情報を収集するか
 - 数値データは集計関数が見える(次のページ参照)
 - 単純にイベント数をカウントしたい場合はcount(*)と指定
- **WHERE 条件**：条件に合致したデータだけを抽出
 - 属性 [=, LIKE, RLIKE, IN] 値 のような書式になる
- **FACET 属性**：指定した属性に沿ってデータをグルーピング
 - デフォルトでは10グループまで表示される、変更したい場合はLIMIT [数値]で指定
- **SINCE 時間 AGO (TIMESERIES [時間])**：検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]

NRQL使いこなしTips ②

数値データの集計関数を覚えましょう！

- 平均値: **average(属性)**
- パーセンタイル: **percentile(属性 [, 何パーセンタイルにするかの数値])**
- 最大値、最小値: **max(属性) , min(属性)**
- 合計: **sum(属性)**
- 最新値: **latest(属性)**
- ヒストグラム: **histogram(属性, データ最大値, スロット数)**
- ある条件に合致するものの割合: **percentage(関数(属性), WHERE 条件)**
- 属性のバリエーション数のカウント: **uniqueCount(属性)**

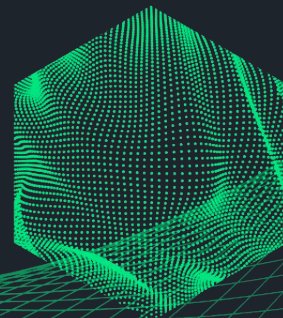
NRQL使いこなしTips ③

WHERE句について

- 部分一致や正規表現が使えます
 - **WHERE 属性 LIKE '%nru%'** (部分一致)
 - **WHERE 属性 RLIKE 'nru..'** (正規表現)
- AND/OR条件が使えます
 - ただし、ORを羅列する場合は **WHERE 属性 IN (値1, 値2, ...)** のほうが推奨の書き方です
- 数値データの場合は不等号が使えます
 - **WHERE duration > 1** など

分析手法の習得

NRU302 - ハンズオン #3



ハンズオン - 分析手法の習得

このハンズオンでは、以下の点を学習します。

- NRQLの確認
- Query Builder でチャートを作成する
- NRQL を用いてチャートを編集する
- ダッシュボードを確認する
- NRQLを用いて様々なチャートを作成する

Option!!



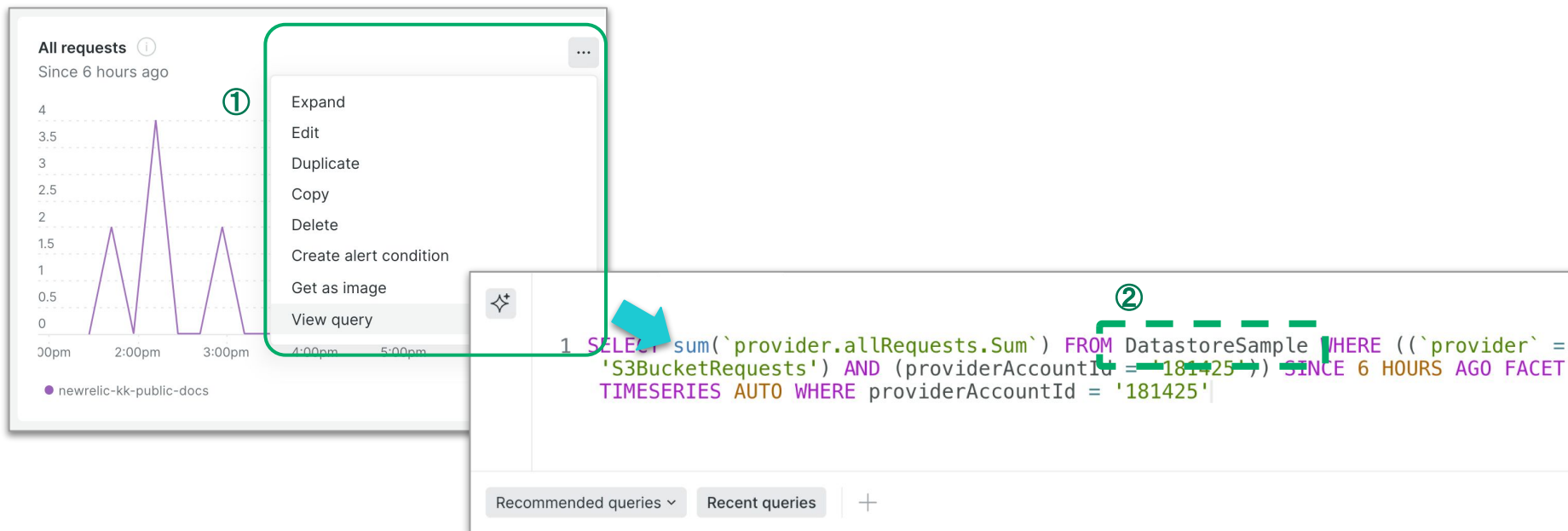
new relic®

16:15 - 16:35 (20min)

p. 75 - p. 93

NRQLの確認

ハンズオン#2でダッシュボードに追加した S3チャートの NRQL を確認します



手順

① ダッシュボードを開き、追加したチャート右上の [⋮] - [View query] をクリックします

② NRQLのFROM句で指定されているデータタイプを確認します

NRQLの変更

NRQL の SELECT 句および SINCE 句を変更しダッシュボードに追加します

変更前

```
SELECT sum(provider.allRequests.Sum) FROM DatastoreSample
WHERE ((provider = 'S3BucketRequests') AND (providerAccountId = '181425'))
SINCE 6 HOURS AGO FACET entityName TIMESERIES AUTO WHERE providerAccountId = '181425'
```



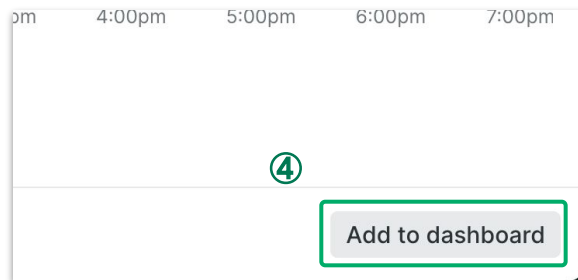
変更後

```
SELECT sum(aws.s3.GetRequests) FROM Metric
WHERE newrelic.cloudIntegrations.providerAccountId = '181425'
SINCE 1 day AGO FACET entity.name TIMESERIES AUTO
```



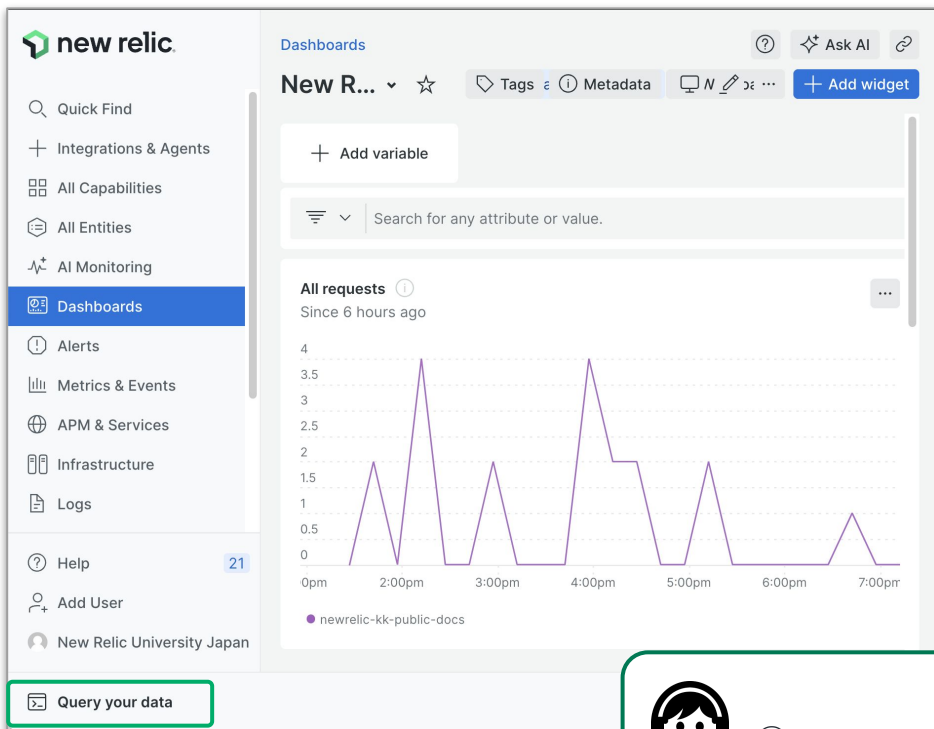
③ 変更後のNRQLを実行します

④ Query builder 右下の [Add to dashboard] をクリックし、チャート
手順 トをダッシュボードに追加します



Query builderの表示

NRQL を操作するため Query builder を開きます



本ページ以降でいくつかのNRQLを入力し結果の確認を行います

- WHERE
- FACET
- SINCE
- TIMESERIES



手順

⑤ メニュー下にあるから [Query Your Data] をクリックします

NRQLの作成

ベースとなる NRQL を実行します

新規

```
SELECT count(*) FROM Transaction FACET request.uri  
SINCE 3 HOURS AGO TIMESERIES
```



改行はShift + ENTER/RETURN



手順

⑥ テキストで指示されたNRQLを入力します

⑦ [Run] をクリックし、チャートが表示されることを確認します

NRQLの変更 #1

NRQL に WHERE 句を追加します

変更前

```
SELECT count(*) FROM Transaction FACET request.uri  
SINCE 3 HOURS AGO TIMESERIES
```



変更後

```
SELECT count(*) FROM Transaction FACET request.uri  
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')  
SINCE 3 HOURS AGO TIMESERIES
```

- **WHERE 条件**: 条件に合致したデータだけを抽出
 - 属性 [=, LIKE, RLIKE, IN] 値 のような書式になる



⑧ 変更後のNRQLを実行し結果を確認します

手順

NRQLの変更 #2

NRQL の FACET 句を変更します

変更前

```
SELECT count(*) FROM Transaction FACET request.uri  
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')  
SINCE 3 HOURS AGO TIMESERIES
```



変更後

```
SELECT count(*) FROM Transaction FACET name  
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')  
SINCE 3 HOURS AGO TIMESERIES
```

- **FACET 属性**: 指定した属性に沿ってデータをグルーピング
 - デフォルトでは10グループまで表示される、変更したい場合はLIMIT [数値]で指定



⑨ 変更後のNRQLを実行し結果を確認します

手順

NRQLの変更 #3

NRQL の SINCE 句を変更します

変更前

```
SELECT count(*) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 3 HOURS AGO TIMESERIES
```



変更後

```
SELECT count(*) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO TIMESERIES
```

- **SINCE 時間 AGO** (TIMESERIES [時間]): 検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]



手順

⑩ 変更後のNRQLを実行し結果を確認します

NRQLの変更 #4

NRQL から **TIMESERIES** 句を削除します

変更前

```
SELECT count(*) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO TIMESERIES
```



変更後

```
SELECT count(*) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO
```

↑ “TIMESERIES” を削除

- SINCE *時間* AGO (**TIMESERIES** [*時間*]) : 検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]

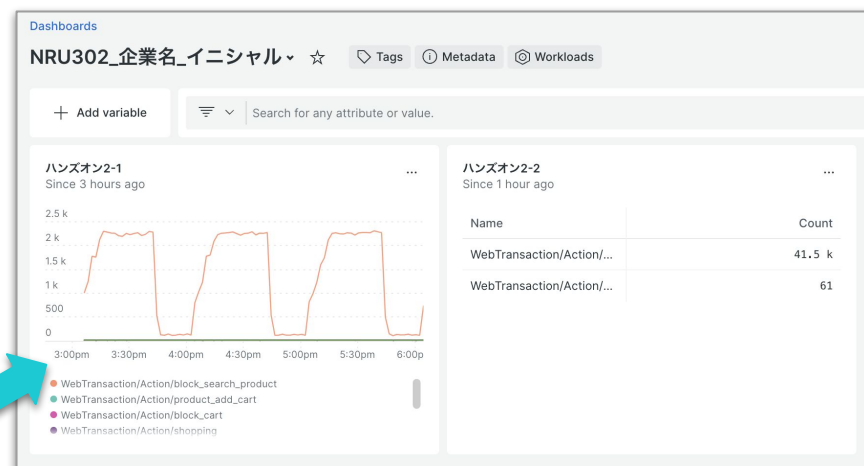
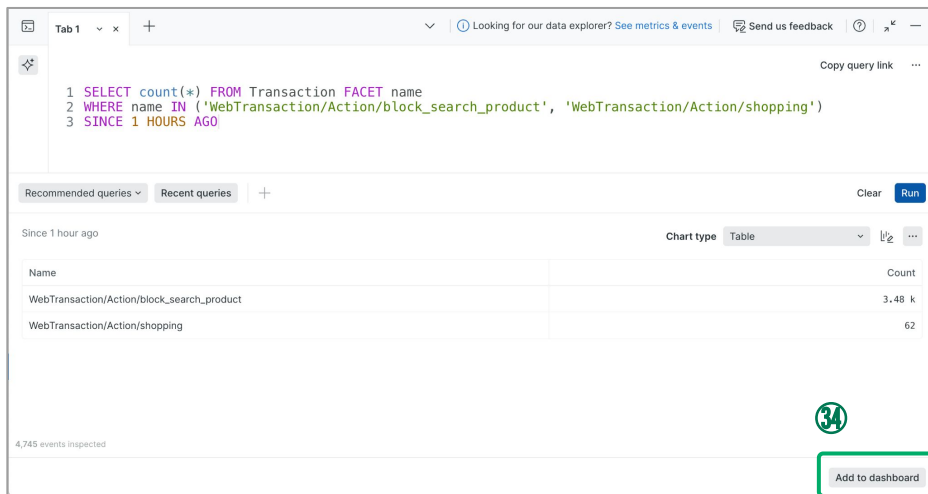


⑪ 変更後のNRQLを実行し結果を確認します

手順

チャートの追加

Query builder で作成したチャートをダッシュボードに追加します

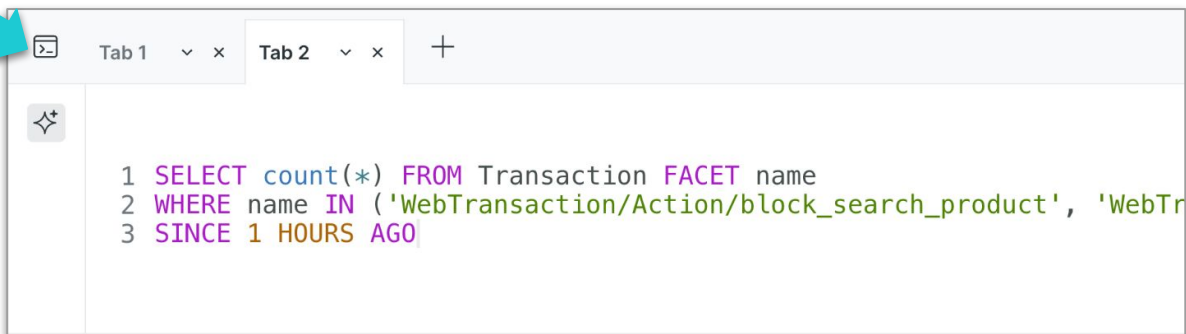
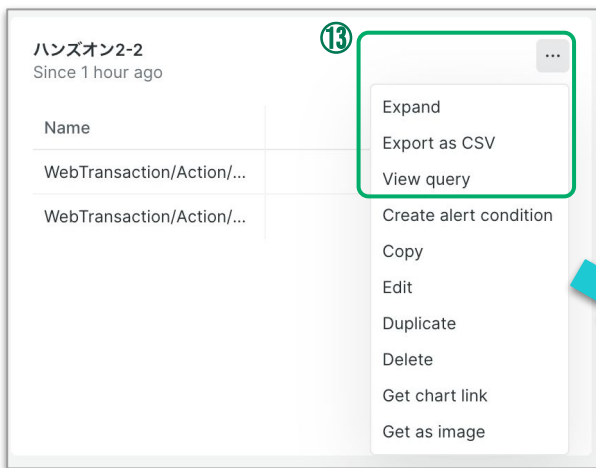


手順

⑫ [Add to dashboard] をクリックし、作成したチャートをダッシュボードに追加・確認します

さまざまなチャートの追加

追加したチャートをベースにして、他のチャートもダッシュボードに追加します



手順

⑬ ダッシュボードを開き、[手順⑫](#) で追加したチャート右上の [⋮] -
[View query] をクリックします

さまざまなチャートの追加 #1

応答時間 (**duration**) を求めるようNRQLを変更しダッシュボードに追加します

変更前

```
SELECT count(*) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO
```



変更後

```
SELECT average(duration), percentile(duration,90) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO
```

- **average(属性)**: 属性の平均値を算出
- **percentile(属性 [, 何パーセンタイルにするかの数値])**: 指定パーセンタイルでの属性の概算値を算出



手順

⑭ 変更後のNRQLを実行し、ダッシュボードに追加します

さまざまなチャートの追加 #2

TIMESERIES 句を追加して時系列データを求めるよう **NRQL** を変更しダッシュボードに追加します

変更前

```
SELECT average(duration), percentile(duration,90) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO
```



変更後

```
SELECT average(duration), percentile(duration,90) FROM Transaction FACET name
WHERE name IN ('WebTransaction/Action/block_search_product', 'WebTransaction/Action/shopping')
SINCE 1 HOURS AGO TIMESERIES
```

- **SINCE 時間 AGO (TIMESERIES [時間])** : 検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]

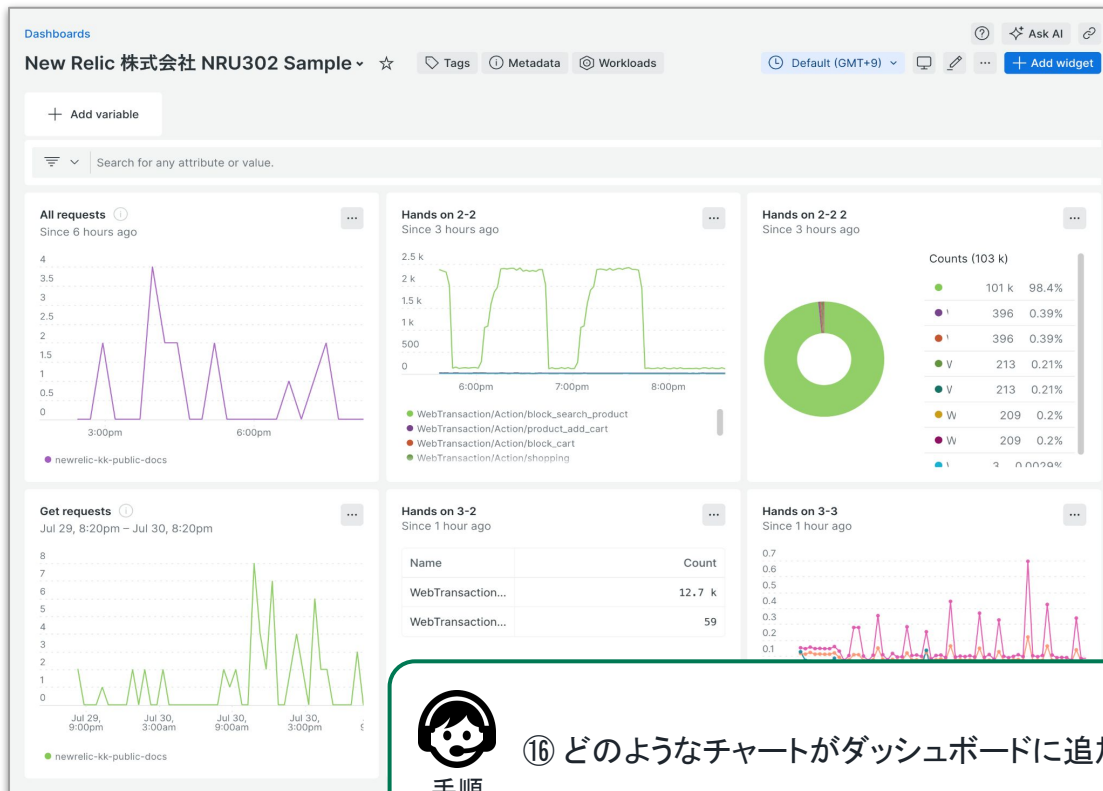


手順

⑮ **手順⑭** で追加したチャートを元にNRQLを編集し、チャートをダッシュボードに追加します

ダッシュボードの確認

これまで追加したチャートをダッシュボードで確認します



手順

⑩ どのようなチャートがダッシュボードに追加されたか確認します

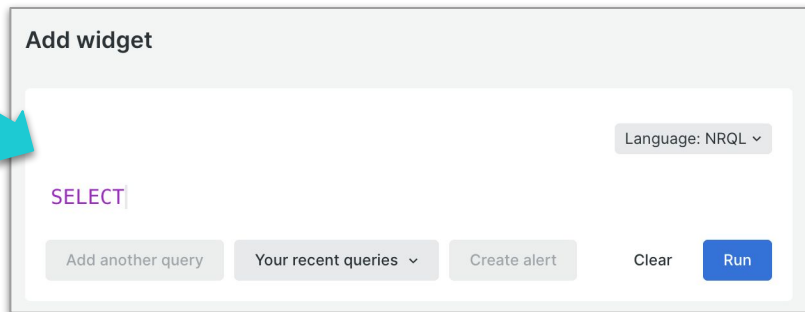
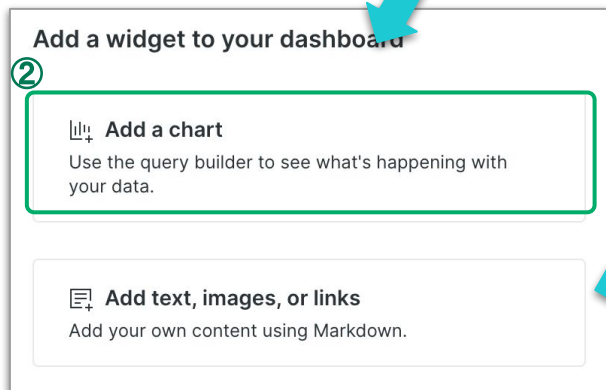
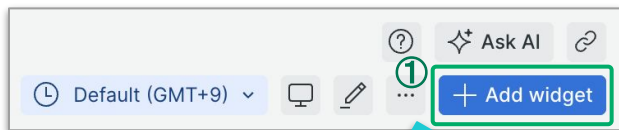
Option!!

NRQLを用いて様々な チャートを作成する

NRU302 - ハンズオン

Option!! NRQLを用いてチャートを作成する

オブザーバビリティ成熟度モデルに例にした 4つのチャートを作成します



オブザーバビリティ成熟度モデル	特徴	KPIの例
0 Instrumentation 計測を開始する	サービスを理解するためのデータを集める (ログ、トレース、API、ユーザー体験...)	- データ一元化率 - 対象システム割合 - ログ監視範囲 - APM中心
1 Reactive 受動的対応	障害対応などの対応時間短縮 (関係者全員がリアルタイムに観測可能)	- サービス停止率 - 障害発生率 - MTTR削減
2 Proactive 積極的対応	大きな問題が起こる前に行動を起こす (ユーズ起点からパフォーマンス改善)	- エラー発生率 - レスポンスタイム - SLI/SLOの測定割合
3 Predictive 予測的対応	サービス改善のためのマインドチェンジ (AIを活用した予防、開発スピード向上)	- 適正スケーリングによるコスト削減 - デプロイ高速化
4	より良いサービスのための投資と行動 (データに基づき正しい意思決定を加速)	- 顧客満足度の改善 - 市場投入までの時間 (新製品、新機能の数)



手順

- ① ダッシュボード右上の [+ Add widget] をクリックします
- ② [Add a chart] をクリックします
- ③ 次ページ以降で指示された NRQL および Chart Type のチャートを作成してみてください

Option!! NRQLを用いてチャートを作成する

過去1週間の稼働率を前週と比較するビルボードを作成します

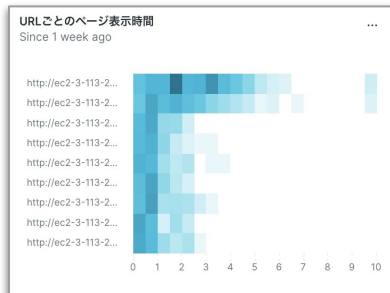


```
SELECT percentage(count(result), where result = 'SUCCESS') AS '稼働率(週)'
FROM SyntheticCheck
SINCE 1 week AGO
COMPARE WITH 1 week AGO
```

- Synthetic (外形監視) のチェック結果である **SyntheticsCheck** イベントから、SUCCESS という結果が入ってるイベントの割合を抽出します
- **AS** 句を使うとクエリ結果をダッシュボードで表示させる際に、わかりやすいラベルに置き換えることができます
- **COMPARE WITH** 句を使ってその前の週と比較しています
- Chart Typeは **Billboard** を選択します

Option!! NRQLを用いてチャートを作成する

URL ごとのページ表示時間のヒートマップを作成します

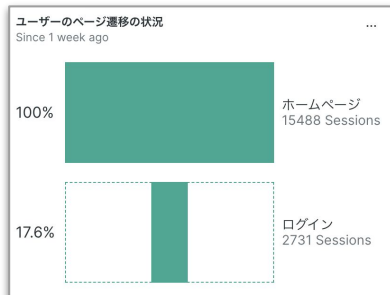


```
SELECT histogram(duration,10,20)
FROM PageView
FACET pageUrl
SINCE 30 MINUTES AGO
```

- Browser エージェントによって計測された**PageView** イベントから、応答時間を抽出します
- **histogram()** 関数を用いて応答時間10 秒を 20 個のウィンドウに分割し、それぞれのウィンドウに当てはまる応答時間をカウントします
- **FACET** 句を使い、pageUrl 毎にヒストグラムを作成します
- グラフは pageUrl を縦軸としたヒートマップで表示されます
- Chart Typeは **Heatmap** を選択します

Option!! NRQLを用いてチャートを作成する

ユーザーのページ遷移の状況を解析するファネルを作成します



```
SELECT funnel(session,  
WHERE pageUri LIKE  
'http://ec2-3-113-215-132.ap-northeast-1.compute.amazonaws.com/ec-cube/index.php%' AS 'ホームペー  
ジ',  
WHERE pageUri LIKE  
'http://ec2-3-113-215-132.ap-northeast-1.compute.amazonaws.com/ec-cube/index.php/mypage/login'  
AS 'ログイン')  
FROM PageView SINCE 1 day AGO
```

- Browser エージェントによって計測された**PageView** イベントから、セッションID を格納している session を抽出します
- **funnel()** 関数を用いて、pageUri の値を条件にして、条件に当てはまる session 数をカウントします
- Chart Typeは **Funnel** を選択します

Option!! NRQLを用いてチャートを作成する

ハンズオン#1 で実施したカスタムイベントの一覧をテーブルで作成します

カスタムイベントの一覧

Since 1 week ago

Timestamp	Initial	Comment
April 02, 2024 18:47:14	NR	NRU302ハンズオン
April 02, 2024 18:06:18	ご自身のイニシ...	任意の文字列を入れ

```
SELECT initial, comment  
FROM NRULab  
SINCE 1 day AGO
```

- ハンズオン #1 で送信したカスタムイベントの**NRULab** イベントから必要な情報を抽出します
- Chart Typeは **Table** を選択します

STEP4

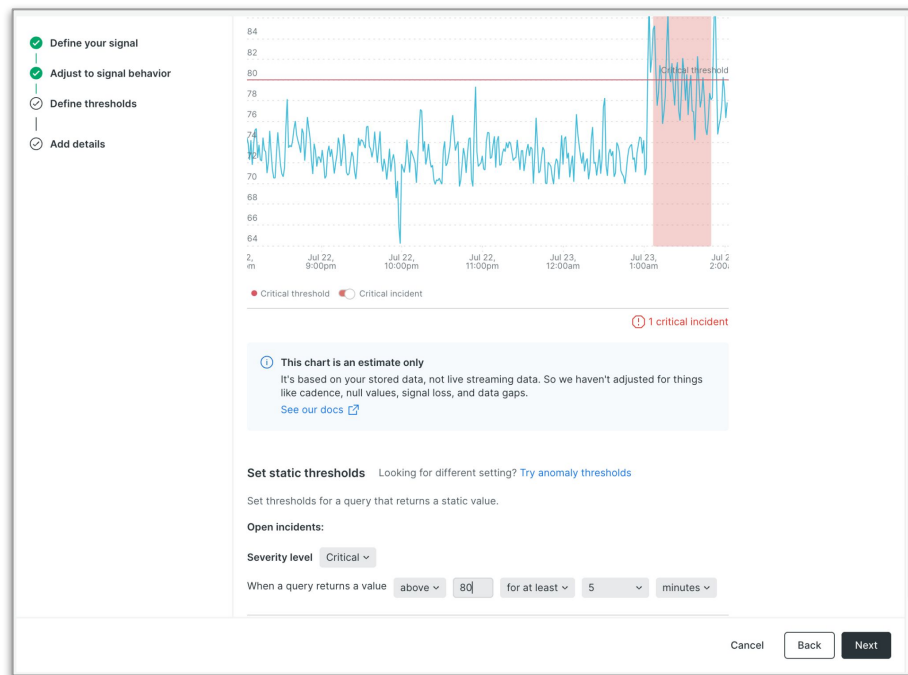
アラートの
作成

高度なアラート設定

NRU302 - 座学

アラート機能

- New Relicが取得しているデータを使ってしきい値(動的/静的)を設定し、アラートを発報することが可能



アラートの通知と確認

メールやSlack, モバイルアプリ等でアラートを受信 (下はSlackの例)

アラートの詳細を New Relic 上で確認

Critical priority issue is active

NewRelic Signup query result is > 0.5 for 5 minutes on 'Signup'

Acknowledge

✓ Close

...

1 incident · NewRelic Signup query result is > 0.5 for 5 minutes on 'Signup'

1 impacted entity · NewRelic Signup

1 condition · Signup

1 policy · Synthetics Check policy

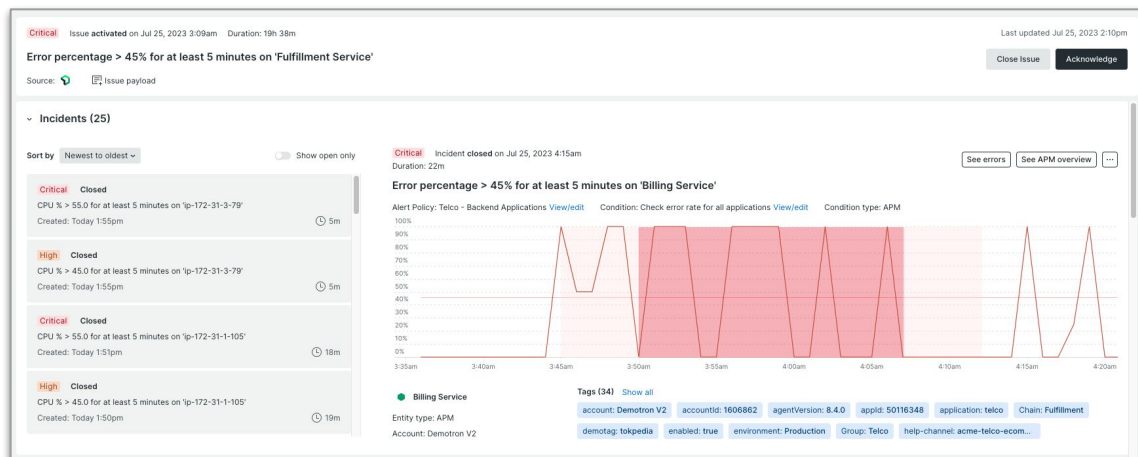
Violation description:

condition-1-a desc

This notification was sent via the "testWorkflow" workflow.

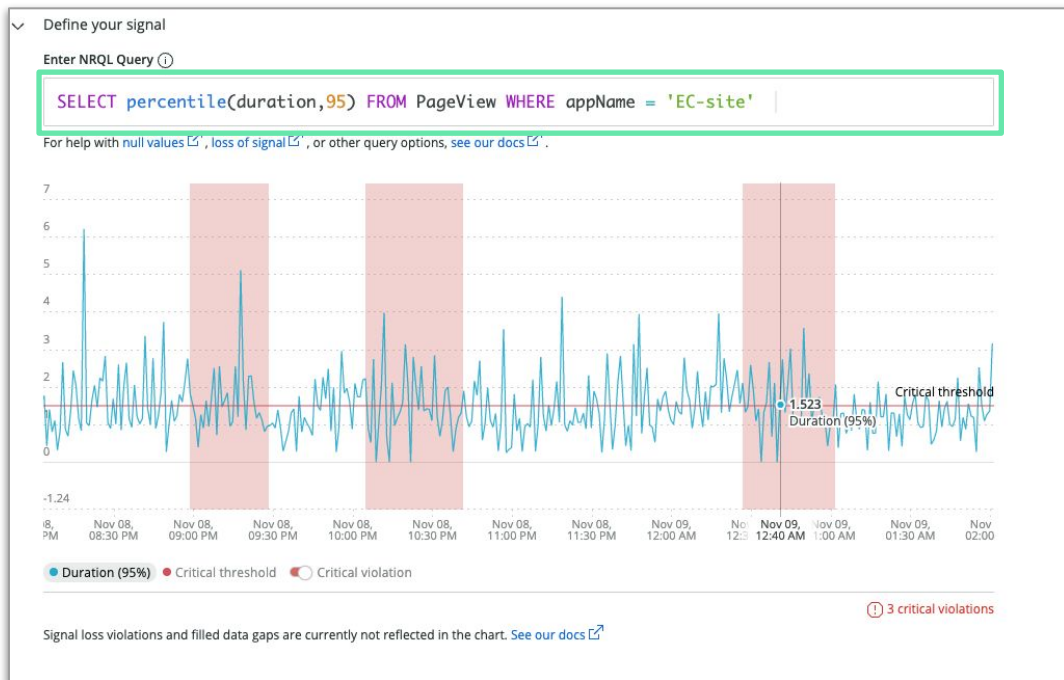
Edit workflow

アラートへのリンク



アラートのカスタマイズ

- NRQLの結果をしきい値にアラートを設定することも可能



- アラートの設定画面で直接 NRQL を記述します
- SELECT句, FROM句および WHERE句(オプション)のみで記述します
- UIを確認しながら、閾値の設定を行います

NRUハンズオンのお知らせ

2024.11.20 (Wed) 15:00 - 17:00

NRU304 - アラート設計の基本と活用

<https://newrelic.com/jp/events/2024-11-20/nru304>

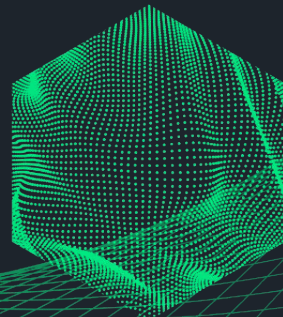


このハンズオントレーニングでカバーするトピック

- ユーザー視点のアラート
- New Relic のアラート機能
- アラートの作成手順
- アラート条件の作成手順と詳細
- アラート運用を補完する機能

アラートの作成

NRU302 - ハンズオン #4



ハンズオン - アラートの作成

このハンズオンでは、以下の点を学習します。

- アラートのためのコンディション作成



new relic®

16:40 - 16:50 (10min)

p. 101 - p. 105

アラートポリシーの選択

アラートコンディションを追加するポリシーを選択します

The screenshot shows the New Relic interface. On the left, the 'Alerts & AI' menu item is highlighted with a green box and a red circle labeled '1'. A red arrow points from this menu item to the 'Alert Policies' section in the main content area. In the 'Alert Policies' section, a table lists four policies. The policy 'ダッシュボードハンズオン用アラートポリシー' is highlighted with a green box and a red circle labeled '2'.

Alert Policies

Search by policy name or id Policy Name = All

Showing 4 policies

Name	Open issues	# of conditions	
NRU-Sample-Policy	1	6	...
NRU環境整備	0	2	...
Service Levels default policy for account 3940716	1	1	...
ダッシュボードハンズオン用アラートポリシー	0	0	...

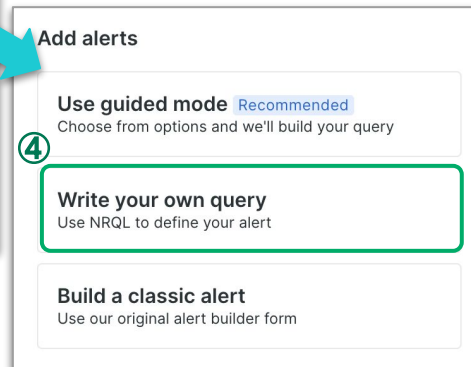
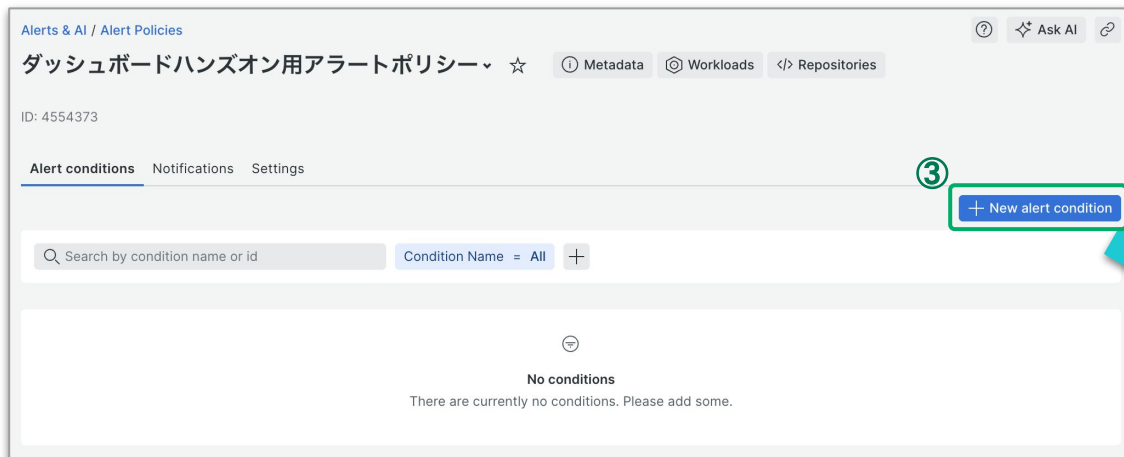


手順

- ① メニューから [Alerts & AI] - [Alert Policies] をクリックします
- ② ポリシーの一覧から「ダッシュボードハンズオン用アラートポリシー」を選択します

コンディションの作成

NRQL によるアラートコンディションを新規作成します



手順

③ Alert conditions画面から **[+ New alert condition]** をクリックします

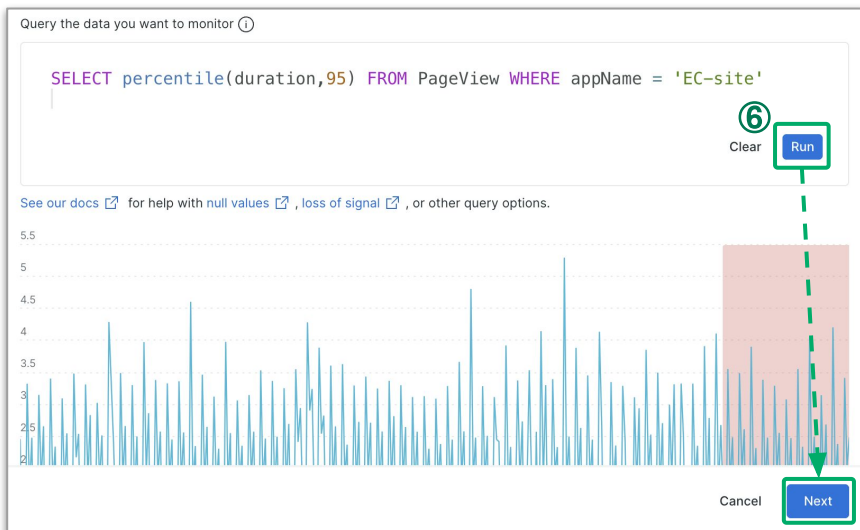
④ Add alerts画面から **[Write your own query]** をクリックします

NRQLの入力

PageView イベントから 95 パーセンタイルの時間を抽出する NRQL を入力します

NRQL

```
SELECT percentile(duration,95) FROM PageView WHERE appName = 'EC-site'
```

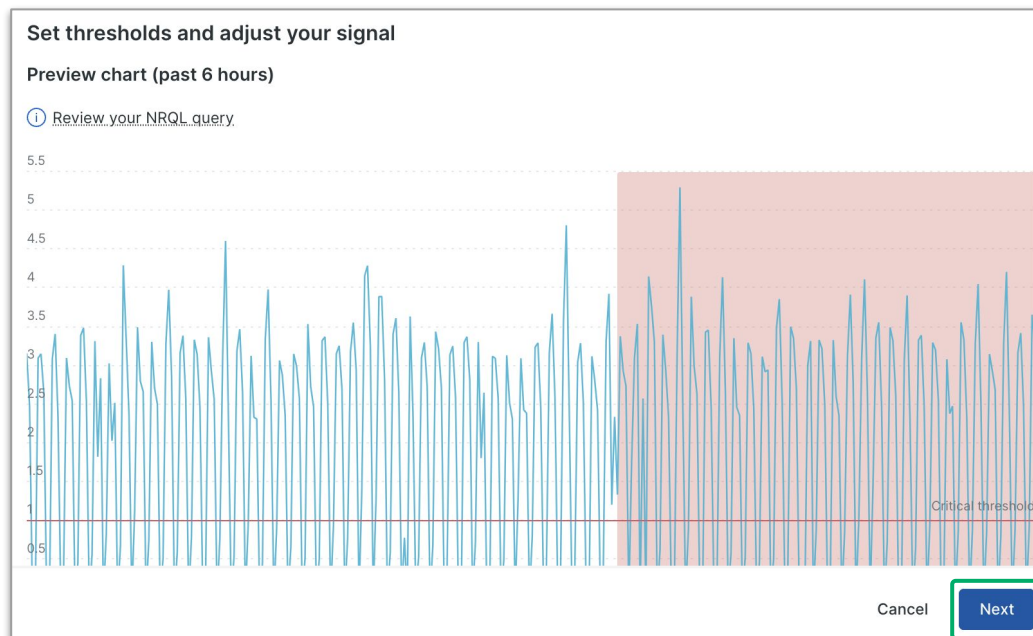


手順

- ⑤ テキストで指示された NRQL を入力します
- ⑥ [Run] をクリックし、チャートが表示されることを確認します
- ⑦ [Next] をクリックします

スレッシュホールドの設定

アラートコンディションの閾値などを設定します



⑧



手順

⑧ 本ハンズオンではスレッシュホールドの設定はデフォルトのままとしますので、何も変更せず **[Next]** をクリックします

アラートコンディションの保存

作成したアラートコンディションを保存します

Add details

⑨ **Name your alert condition ***

Use a clear name that indicates what's wrong

We recommend: EC-site Page Load Time 95th Percentile Exceeds Threshold

Close open incidents after ① 3 days

Send a custom incident description (optional) ①

4,000 character limit

Runbook URL (optional)

Cancel <> View as code **⑩ Save condition**

[Name your alert condition]
は他の参加者との重複を避ける
ため次の形式で入力してください

NRU302-企業名-イニシャル
e.g. NRU302-NEWRELIC-NR



手順

- ⑨ **[Name your alert condition]** に一意となる名前を入力します
- ⑩ **[Save condition]** をクリックし、コンディションが保存されることを確認します

さいごに

NRU302

Dashboard と NRQL の 便利な機能

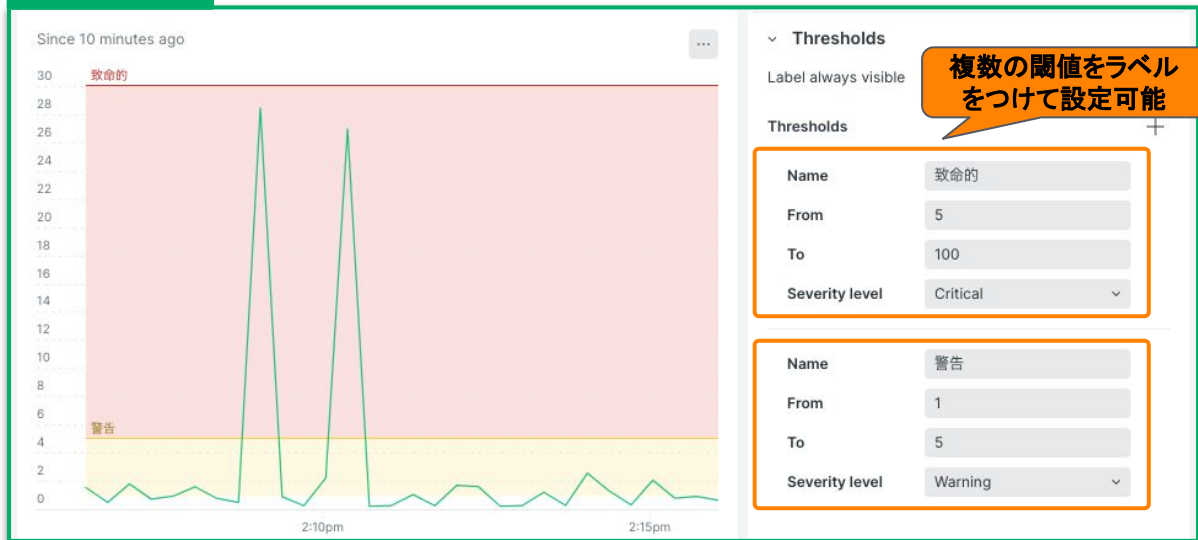
NRU302

LineとTableチャートで直感的に問題を発見

- LineとTableチャートで **閾値を設定** できるようになりました
- Lineチャート上に **重要度に合わせた静的閾値を表示**、SLOやアラートの静的閾値などを視覚化可能
- Tableチャート上の **セルを重要度に** **合わせて配色**、直感的に問題発生を把握可能

参照: [公式ドキュメント](#)

Lineチャート



Tableチャート

New!

Since 1 hour ago

App Name	Avg Duration	Avg Database Duration	Avg External Duration
front-service	0.0721	0.00806	0.264
catalogue-web	0.0621	0.0188	

対象のカラムに対して閾値と重要度を指定

Column Avg External Duration

From 0.1

To 1000

Severity level Critical

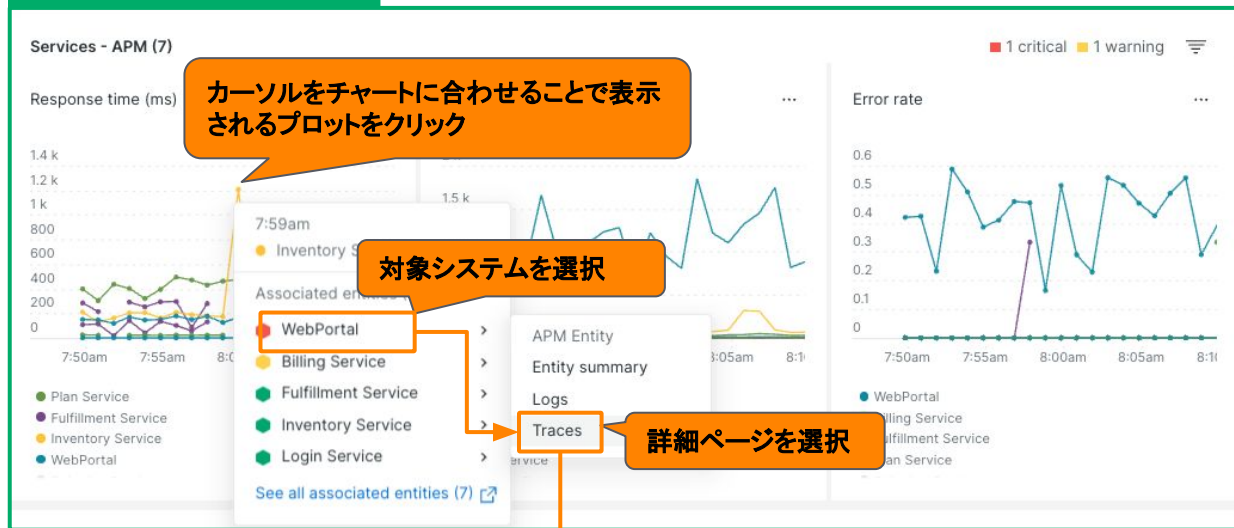
チャートからシームレスにドリルダウンして詳細分析

- Dashboard、Workloads、その他のチャートから**シームレスに情報の詳細を表示**できるようになった
- NRQLの**WHERE句で対象のアプリを絞る**[†]ことで機能が有効化

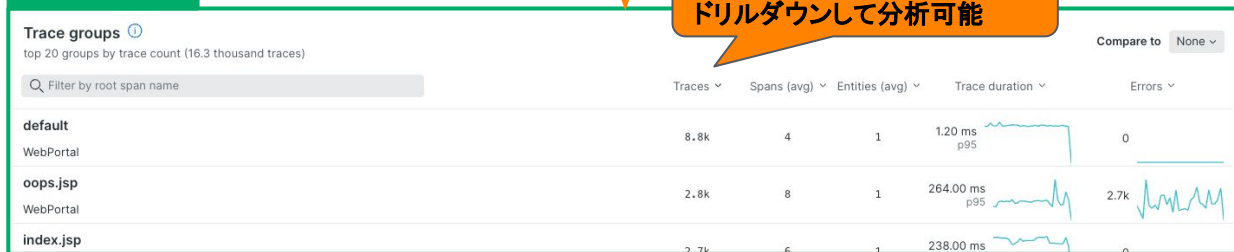
[†]WHERE句でentityGuid、entity.guid、appName、entity.name、または、entityNameを指定

参照: [What's new](#)

例) Workloads > Activity



例) Traces

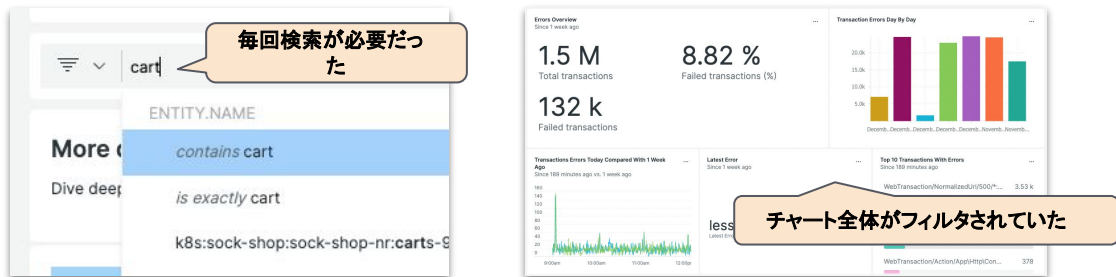


ダッシュボード テンプレート変数

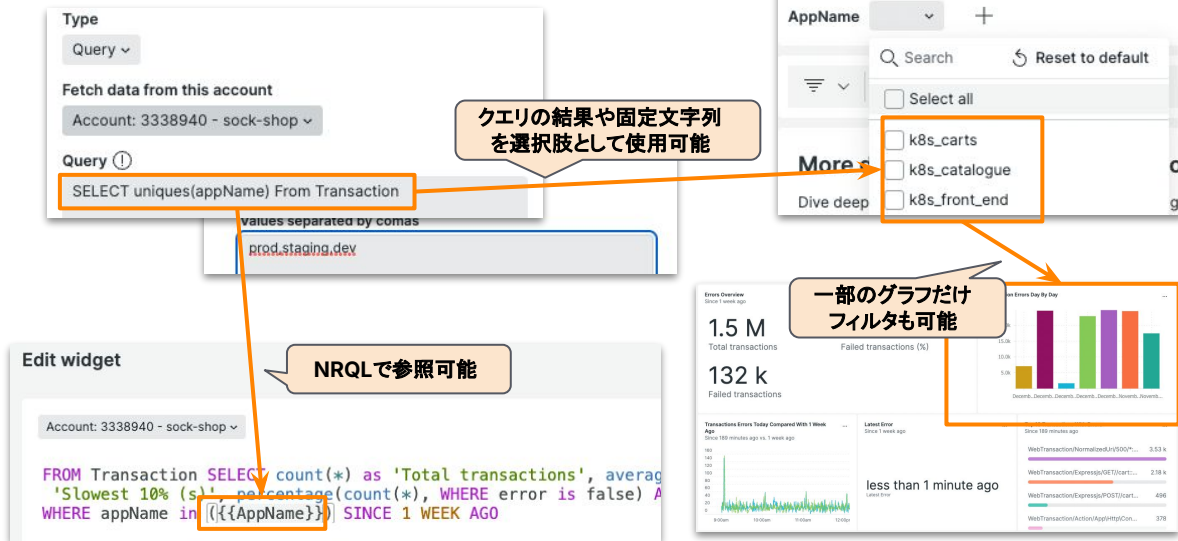
- あらかじめ変数を用意することで、フィルタ項目の検索が不要になります
- 個々のチャートをそれぞれ別々の変数でフィルタリングできます
- 固定リストとNRQLのクエリ結果を選択肢にできます

参照: [テンプレート変数](#)

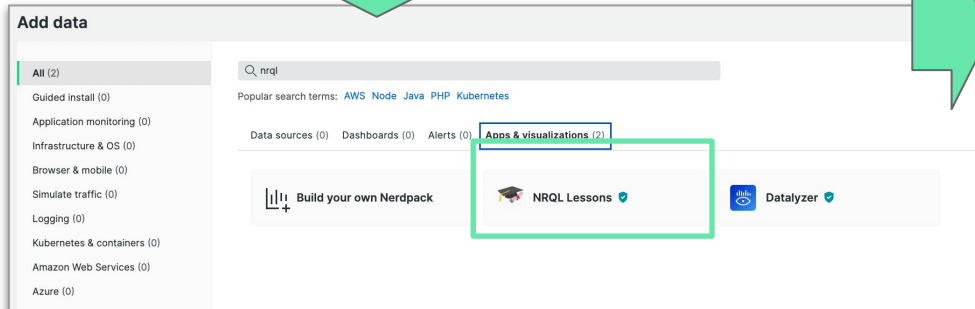
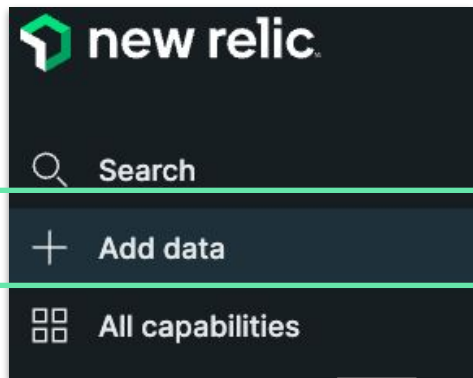
従来のフィルタ機能



追加のフィルタ機能



NRQL Lessons - NRQL学習ツール



NRQLに関するさまざまな利用方法を学習することができます

Select Language: 日本語

レベル1: コツを掴む

- ようこそ
- 初めてのクエリ
- データの集約
- 時間範囲
- 時系列クエリ
- WHERE句

初めてのクエリ

まず、New Relic APMで収集される、**Transaction**イベントのすべてのNRQLクエリは、**SELECT**と**FROM**を持たなければなりません。Transactionイベントの全ての情報を取得する簡単なクエリ

NRQL

```
SELECT * FROM Public_APICall
```

Result

Timestamp	Api
-----------	-----

Query your data画面 のUXが改善され 使いやすくなりました

- 全てのページの最下部から、**ワンクリックでアクセス可能**
- 最大5アカウントのデータを**同時にクエリ可能**
- 分析したいことを自然言語で **New Relic AI**に問い合わせてクエリを生成可能
- **おすすめのクエリのリスト** を提供

参照: [What's new](#)

The screenshot shows the New Relic Query Builder interface. The top navigation bar includes 'APM & Services / Services - APM', 'WebPortal', and various filters like 'critical alerts' and 'Good'. The main area is titled 'Query your data' and contains several sections:

- Callout 1:** Points to a button at the bottom of the page that says 'Query your data'.
- Callout 2:** Points to a section titled 'Query using AI - Experimental' which includes a text input for natural language queries and a 'Generate query' button.
- Callout 3:** Points to a section titled 'Recommended queries' which displays a list of suggested queries.
- Callout 4:** Points to a section where an NRQL query is entered: 'FROM Transaction SELECT count(*) FACET accountId() TIMESERIES'.
- Callout 5:** Points to a dropdown menu for 'Accounts' which is set to 'Demotron V2'.
- Callout 6:** Points to a section titled 'Customize this visualization' which includes a 'Basic information' section with fields for 'Name' and 'Chart type' (set to 'Line').

The bottom of the interface shows a line chart with the title 'Since 1 hour ago' and a legend indicating '1,165,811 events inspected in 0 ms (Infinity MEPS)'.

NRQLで使える関数が大幅に増えました

- **jparsed/mapKeys/mapValues関数**
JSON形式ログを解析
- **toTimestamp/toDatetime関数**
文字列をタイムスタンプや日付に
- **cidrAddress関数**
IP文字列をCIDRに変換
- **encode/decode関数**
Base64デコード / エンコード

参照: [What's new](#)

例) jparse, toTimestamp, toDatetime, encodeの使用例

```
SELECT
  jparse(builds)[0][stage] as Stage,
  toTimestamp(jparse(builds)[0][started_at]) as StartedAt,
  toDatetime(toTimestamp(jparse(builds)[0][finished_at]), 'yyyy/MM/dd') as FinishedAt,
  toTimestamp(jparse(builds)[0][finished_at]) - toTimestamp(jparse(builds)[0][started_at]) as Duration,
  encode(builds, 'base64'),
  builds
FROM Log SINCE 40 MINUTES AGO WHERE builds is not null
```

1	フィールドを取得	2	文字列から日時に変換	3	変換後計算可能	4	エンコード	5	元のデータ
timestamp	Stage	StartedAt	FinishedAt	Duration	encode(builds, ...)	builds			
13:33:29.611	build	1707117637233	2024/02/05	40189	W3siaWQioJEwODU...	{ "id": 10853, "stage": "build", "name":			

例) cidrAddressの使用例

```
WITH '172.27.10.123' as ip
SELECT cidrAddress(ip, 10) as '/10', cidrAddress(ip, 12) as '/12', cidrAddress(ip, 24) as '/24'
```

/10	/12	/24
172.0.0.0/10	172.16.0.0/12	172.27.10.0/24

CIDRを取得

Lookupテーブルを アップロードし、 NRQLで分析可能に

- UI上からCSV形式のデータを
Lookupテーブルとしてアップロード
- Lookupテーブルの内容はNRQL
のlookup関数で確認可能
- JOIN句と併用することで、NRQL
での分析結果をより
リッチに表示して可視性を向上

参照: [公式ドキュメント](#)

The screenshot shows the New Relic interface for managing Lookup tables. On the left, a sidebar lists navigation options: YOUR LOGS (All logs, Attributes, Patterns), MANAGE DATA (Parsing, Data partitions, Obfuscation, and Lookup tables, which is highlighted). The main panel is titled 'Lookup tables' and includes a table of existing tables. An orange callout bubble points to the 'Add a table' button, stating 'UI上からCSVをアップロード'. Below the table, an NRQL query is shown: `FROM lookup(statusCodeTable) SELECT *`, with an orange callout bubble stating 'NRQLのlookup関数で内容を確認可能'. A table of data from 'statusCodeTable' is displayed, showing columns for Timestamp, Status Code, Status Definition, and Status Summary. Another orange callout bubble points to this table, stating 'JOIN句と併用して、各種イベントの分析結果に Lookupテーブルのデータを付加して可視性を向上'. Below this, two NRQL queries are shown. The first query is `FROM Transaction SELECT count(*) facet http.statusCode`, and the second is `FROM Transaction | JOIN (FROM lookup(statusCodeTable) SELECT status_code, status_summary, status_definition) ON http.statusCode=status_code | SELECT count(*) FACET status_summary, status_definition, http.statusCode`. The results of the first query are shown in a table with columns 'Http.Status Code' and 'Count'. The results of the second query are shown in a table with columns 'Status Summary', 'Status Definition', 'Http.Status Code', and 'Count'. An orange callout bubble points to the second query's results, stating 'LookupテーブルのデータをJOINし、より分析しやすく'.

YOUR LOGS

- All logs
- Attributes
- Patterns

MANAGE DATA

- Parsing
- Data partitions
- Obfuscation
- Lookup tables

Logs

Lookup tables

Connect your business data with your performance data—add external lookup tables here and then query them alongside your other data.

Add a table

Name	Description	Rows	File si...	Modified by	Last updated ↓	
statusCodeTable		59	2 KB	ajones@example.com	6/22/2023 2:06:36 PM	...
Shady						
itemN						
bad_I						

FROM lookup(statusCodeTable) SELECT *

Add another query Your recent queries Create alert

NRQLのlookup関数で内容を確認可能

Since 3 weeks ago

Timestamp	Status Code	Status Definition	Status Summary
June 22, 2023 12:06:36	102	Processing	Informational
June 22, 2023 12:06:36	103	Early Hints	Informational
June 22, 2023 12:06:36	200	OK	Success

JOIN句と併用して、各種イベントの分析結果に
Lookupテーブルのデータを付加して可視性を向上

FROM Transaction SELECT count(*) facet http.statusCode

Add another query Your recent queries Create alert

Since 1 hour ago

Http.Status Code	Count
200	372 k
304	716 k
404	218
302	201
400	169
500	158
501	143

FROM Transaction | JOIN (FROM lookup(statusCodeTable) SELECT status_code, status_summary, status_definition) ON http.statusCode=status_code | SELECT count(*) FACET status_summary, status_definition, http.statusCode

Add another query Your recent queries Create alert

Since 1 hour ago

Status Summary ↓	Status Definition	Http.Status Code	Count
Success	OK	200	371 k
Success	Partial Content	206	18
ServerError	Internal Server Error	500	147
Redirection	Found	302	209

ステータスコード(数値)
別のトランザクション数

LookupテーブルのデータをJOINし、より分析しやすく

NRQLのサブクエリでJOIN句が利用可能に

- 異なる Event Type にまたがるデータを結合して表示が可能
- 2つの Event Type に共通する属性値があれば、どんな組み合わせ⁺でも使える！
- Query Builder で利用可能

参照:

- [公式ドキュメント](#)

⁺Keyとなるデータは最大100件まで対応

アプリケーション由来のデータ (Transaction Event)

FROM Transaction SELECT average(duration) FACET host

Add another query Your recent queries Create alert

ホスト名	アプリ応答時間
Host	Avg Duration
order-composer-5495f9b947-m9ts6	5.71
order-composer-5495f9b947-2gt6z	5.67
order-composer-5495f9b947-dr6pl	5.53
order-composer-5495f9b947-n8x52	5.45
order-processing-695bdd958f-bzzvn	4.73

インフラ由来のデータ (SystemSample Event)

FROM SystemSample SELECT average(cpuPercent),average(memoryUsedPercent) FACET hostname

Add another query

Your recent queries ▾

Create alert

ホスト名

Hostname
host-tower-portland
host-tower-washington
host-tower-houston
host-tower-indianapolis
host-tower-stockton
host-tower-austin

インフラリソース使用率

Avg Cpu Percent	Avg Memory Used Percent
99.8	39.8
99.8	32.8
96.6	66.1
95.1	79.2
72.2	59.9
71.8	61.2

FROM Event [INNER|LEFT] JOIN (Subquery) ON [key =] key SELECT ...

FROM Transaction JOIN (FROM SysSample SELECT average(cpuPercent) AS cpuPerc, average(memoryUsedPercent) AS memPerc facet hostname) ON host = hostname SELECT average(duration), latest(cpuPerc), latest(memPerc) FACET hostname since 10 minutes ago

Add another query

Your recent queries ▾

Create alert

Clear

Run

Since

ホスト名

アプリ応答時間

インフラリソース使用率

...

Hostname	Avg Duration	Cpu Perc	Mem Perc
host-tower-portland	0.264	99.857	63.123
host-tower-washington	0.247	99.817	44.583
host-routing-service-2	0.161	50.42	86.2
host-tower-houston	0.129	92.325	90.382

Basic information

Chart name

Enter a chart name

Chart type

Table

More visualizations in [UI](#)

> Other groups (1)

ホスト名をKeyにして、アプリケーション由来のアプリ応答時間とインフラ由来のインフラリソース使用率を単一テーブルとして表示

NRQLにコメント機能が追加されました

- **1行コメント**と**複数行コメント**の両方を追加することが可能
- NRQLの複雑なクエリにコメントを追記して**メンテナンス性を向上**
- クエリを一時的にコメントアウトしてデバッグができるようになり**クエリ作成を効率化**

参照: [公式ドキュメント](#)

```
FROM TransactionTrace // 1行コメント
-- 1行コメント
SELECT count(*)
```

1行コメントする場合は
// か --を使用する

```
FROM TransactionTrace /*このコメントは、
複数行に
またがっています
*/
SELECT count(*)
```

複数行コメントは
/* */を使用する

従来のNRQL

複雑なクエリだと処理内容がわからない

```
SELECT average(queueDuration) AS 'Request queueing', average(webAppDuration) AS
'Web application', average(networkDuration) AS 'Network duration', average
(domProcessingDuration) AS 'DOM processing', average(pageRenderingDuration) AS
'Page rendering' FROM PageView WHERE appName IN ({{AppName}}) SINCE 1800 seconds
AGO TIMESERIES
```

NRQLのコメント機能

```
SELECT average(queueDuration) AS 'Request queueing'
, average(webAppDuration) AS 'Web application'
, average(networkDuration) AS 'Network duration'
, average(domProcessingDuration) AS 'DOM processing'
, average(pageRenderingDuration) AS 'Page rendering'
FROM PageView
WHERE appName IN ({{AppName}}) -- リストで選択されたアプリ名でフィルタ
// AND city = 'Tokyo'
SINCE 1800 seconds AGO TIMESERIES
```

条件の説明を追記できる

コメントアウトでデバッグ可能

NRU GUIDE

NRU302

New Relic University

New Relicの基礎から応用までを学べ、認定資格も取得できるセルフラーニングコンテンツです

Install	NRU 100	NRU 200	NRU 300/400	Exam
New Relic を使い始める New Relic One へのサインアップやエージェントインストールの方法などのガイドを提供 APM / Browser / Infrastructure / Logs / Mobile (iOS/Android) / AWS統合 / Azure統合 / GCP統合 インストール手順	Observability/New Relic を知る New Relic One やオプザバビリティに関する基礎知識を座学にて学習 NRU Practitioner オプザバビリティ入門 NRU 101 New Relic One 入門	New Relic の主要機能を学ぶ New Relic One に含まれる3つの主要機能に含まれる54の機能群を動画で説明 NRU201 Telemetry Data Platform NRU202 Full Stack Observability NRU203 Applied Intelligence	New Relic の使い方を体感する New Relic One を実際に操作し、主要機能を利用できる状態にするためのトレーニング NRU 301 アプリケーションとインフラ性能観測の基本 NRU 302 ダッシュボード開発とNRQLの基本 NRU 303 SLI/SLO設計の基本 NRU 304 AIOps とアラート設計の基本 NRU 401 IDEと連携し、問題解決を加速する New Relic活用の実践	資格を得る New Relicの知識を有していることを証明するための試験、合格すると資格バッジを授与 フルスタックオプザバビリティ認定試験
▶サインアップ方法 https://newrelic.com/jp/blog/how-to-relic/create-new-account ▶インストールガイド https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide	▶オンデマンドセミナー https://newrelic.com/jp/resources/presentations/nru-practitioner2022 https://newrelic.com/jp/resources/presentations/nru101-2022	▶主要機能解説動画 https://newrelic.com/jp/learn	▶開催スケジュール https://newrelic.com/jp/events	▶受験サイト https://learn.newrelic.com/full-stack-observability-exam-jp

2024年10-12月期開催のハンズオンセミナー

2024年10月9日(水)15:00-17:00

NRU301 - アプリケーションとインフラ性能観測の基本

<https://newrelic.com/jp/events/2024-10-09/nru301>

2024年10月23日(水)15:00-17:00

NRU302 - ダッシュボード開発とNRQLの基本

<https://newrelic.com/jp/events/2024-10-23/nru302>

2024年11月6日(水)15:00-17:00

NRU303 - SLI/SLO 設計の基本

<https://newrelic.com/jp/events/2024-11-06/nru303>

2024年11月20日(水)15:00-17:00

NRU304 - アラート設計の基本と活用

<https://newrelic.com/jp/events/2024-11-20/nru304>

2024年12月11日(水)15:00-17:00

NRU401 - IDEと連携し、問題解決を加速する New Relic 活用の実践

<https://newrelic.com/jp/events/2024-12-11/nru401>

【開催形式】

全てZoomによるウェビナー

[こちらから](#)も一覧でご覧いただけます！



Observability Engineerに！ Full Stack Observability Practitioner認定試験

【この認定試験を通じて身につくスキル】

- Observabilityの実現のために New Relicが取得するデータの理解と目的に応じたデータ分析やアラート設定
- バックエンドおよびフロントエンドの問題発見とトラブルシューティング

【合格者特典】

- デジタル認定証とバッジ
- 合格者限定ノベルティの送付
★今だけ認定者限定デザイン Tシャツ！

【認定試験に向けた準備】

- New Relicの基本的な操作経験
- **ラーニングパス** に沿った学習
 - [座学](#)
 - [ハンズオン](#)
 - [認定試験対策講座](#) (2024.12.18)



認定試験受験
サイトは [こちら](#)

- 受験料：無料
- 試験の言語：日本語
- 受験形態：オンライン（いつでも受験可能）

New Relic を知る

その他下記にマッピングされない情報提供

New Relic Status
ページ

テクニカル
サポート
※詳細は後述

経験者向け

コミュニティサイト(Explorers Hub)

※既存投稿は英語ですが、日本語の投稿も可能

YouTube Channel

機能アップデート

個別ご活用
支援

※ご契約状況によって
ご提供できる内容が異
なります

New Relic
ブログ

NRUG
(ユーザー
グループ)

ユースケース
実践的な情報

機能理解
ラーニング

Docs
オンラインマ
ニュアル

NRU
New Relic
University

※詳細は後述

New Relic
実践入門
(書籍)

お客様事例講演
※作成中

New Relic
セミナー

お客様事例紹介

初めての方向け

日本語で提供

英語でのみ提供

お知らせ

NRU302

New Relic [ずっと] 無料サインアップ ([Link](#))

1名のフルユーザーアクセス、100GB/月のデータ保存容量

① 仮登録への入力



The form is titled "New Relic 無料サインアップ" (New Relic Free Sign Up). It includes a "サインアップ" (Sign Up) button. Below the button, there is a section for "サインアップ" (Sign Up) with fields for "山" (Mountain) and "太郎" (Taro). There is also a "会社名" (Company Name) field with "New Relic, Inc." entered. A "選択してください" (Please select) dropdown menu is at the bottom.

② 本登録へ遷移

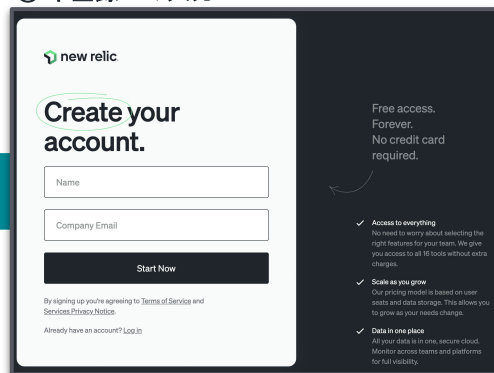
仮登録ありがとうございます。
続いて本登録をお願い致します。

以下のボタン、またはご登録いただいたメールに届いたリンクより、サインアップのための本登録をお願いします。

※ 仮登録と本登録は同じメールアドレスをご入力ください。

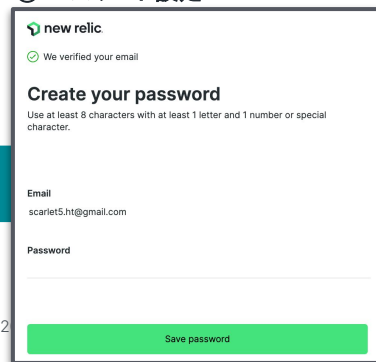
本登録はこちら

③ 本登録への入力



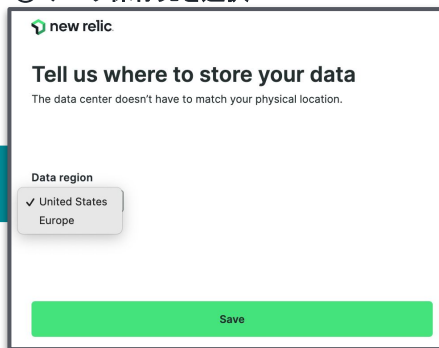
The form is titled "Create your account." It includes fields for "Name" and "Company Email". A "Start Now" button is at the bottom. To the right, there is a section titled "Free access. Forever. No credit card required." with three bullet points: "Access to everything", "Scale as you grow", and "Data in one place".

④ パスワード設定



The form is titled "Create your password". It includes a "Save password" button. Below the button, there is a "Password" field. The email address "scarlet5.ht@gmail.com" is displayed.

⑤ データ保存先を選択



The form is titled "Tell us where to store your data". It includes a "Data region" section with "United States" and "Europe" options. A "Save" button is at the bottom.



実践的なオブザーバビリティを 学べるガイドブック

■Part 1: New Relicを知る

- ・第1章: オブザーバビリティの重要性
- ・第2章: New Relicの全体像

■Part 2: New Relicを始める

- ・第3章: New Relic Synthetic Monitoring
- ・第4章: New Relic Mobile
- ・第5章: New Relic Browser
- ・第6章: New Relic APM
- ・第7章: New Relic Infrastructure
- ・第8章: New Relic NPM
- ・第9章: New Relic Log Management
- ・第10章: New Relic Alerts & AI ①: New Relic Alerts
- ・第11章: New Relic Alerts & AI ②: AI
- ・第12章: DevSecOps
- ・第13章: ビジュアライゼーション

■Part 3: New Relic活用レシピ



<https://www.amazon.co.jp/dp/4798184500/>

NRUG ぬるぐで学ぶ

New Relic User Group

New Relic ユーザーが集い、実践事例や最新機能紹介などを実施。初心者支部や SRE 支部などが形成されており、エンジニア同士でのネットワーキングや信頼性の高い情報交換が可能。



[NRUG 本部](#)



[NRUG SRE支部](#)



[NRUG 沖縄支部](#)

以上、お疲れさまでした
ご質問があればQ&Aにご記入ください
アンケートにご協力お願いいたします

Thank you.
NRU302

以上、お疲れさまでした
ご質問があればQ&Aにご記入ください
アンケートにご協力お願いいたします

Thank you.

NRU302

