



New Relic

# ダッシュボード開発と NRQLの基本編

NRU 302 - Dashboard / NRQL

# 本ウェビナーについて



本ウェビナーでは質問が可能です。

音声不具合などについても  
こちらからご連絡ください。

質問は最後のQ&Aの時間にて  
可能な範囲で回答いたします。

最後にアンケートが表示されます。セミナー品  
質向上のためにぜひご協力をお願いいたしま  
す。



# 松川 晋士

New Relic K.K.

Senior Technical Support Engineer

某国内メーカーにてフィーチャーフォンのソフトウェア開発に始まり、スマートフォンのソフトウェア開発に従事。その後、数名のベンチャー企業に参画し、開発者向けサービスの立ち上げを経験。AWS上でのサービス構築から、モバイル SDKの開発、運用、サポートなど各種業務を経験し、New Relicへ参加。



# 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を受講ください! (オンデマンド視聴可)

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

# 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	データの理解
15:40 - 15:50	座学	NRQL (New Relic Query Language)
15:50 - 16:10	ハンズオン	分析手法の習得
16:10 - 16:15	座学	New Relic Oneダッシュボード
16:15 - 16:35	ハンズオン	ダッシュボードの作成
16:35 - 16:40	座学	高度なアラート設定
16:40 - 16:50	ハンズオン	アラートの作成
16:50 - 17:00	--	まとめ、アンケートご記入

# New Relicとデータ分析

# New Relicのアーキテクチャ

## Front-end

## Back-end

## Management

New Relic.  
**BROWSER**™



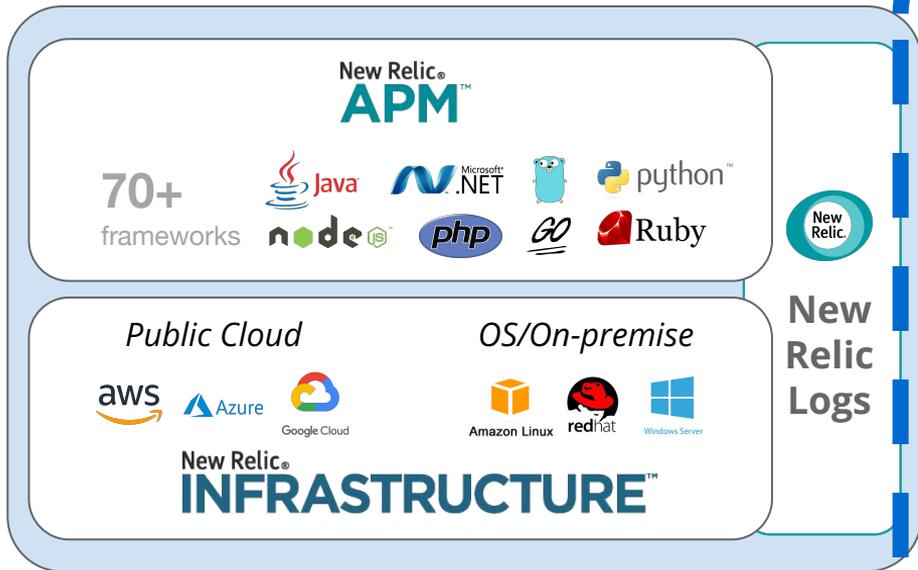
New Relic.  
**MOBILE**™



New Relic.  
**SYNTHETICS**™



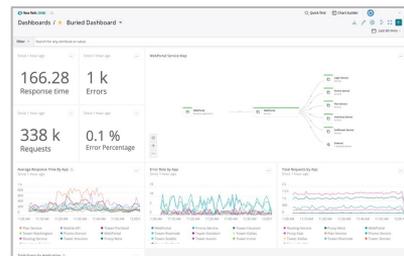
顧客視点データの取得



システムデータの取得



New Relic.  
**ONE**™ ダッシュボード

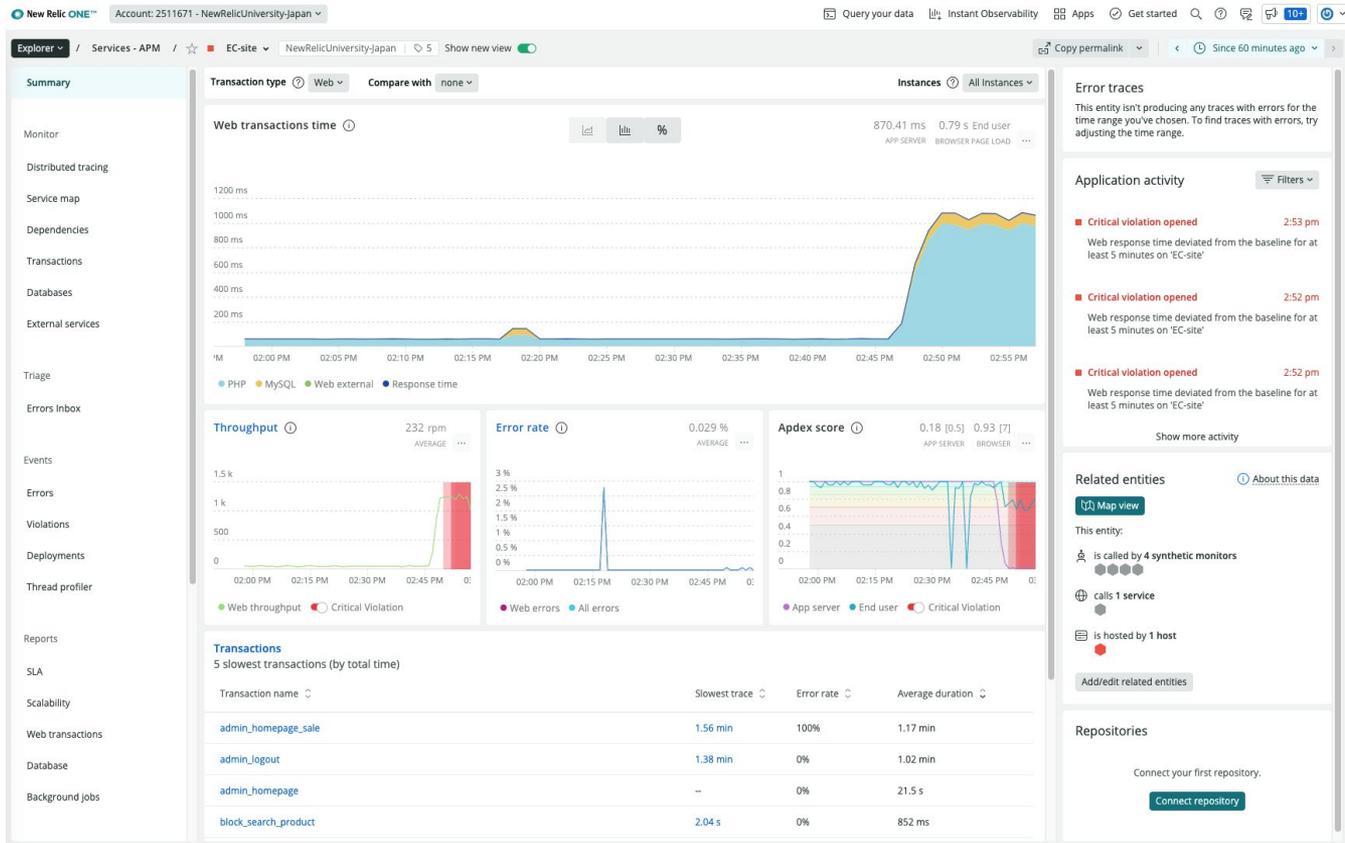


New Relic.  
**ONE**™ チャートビルダー (NRQL)

```
SELECT apdex(duration) FROM PageView facet
```

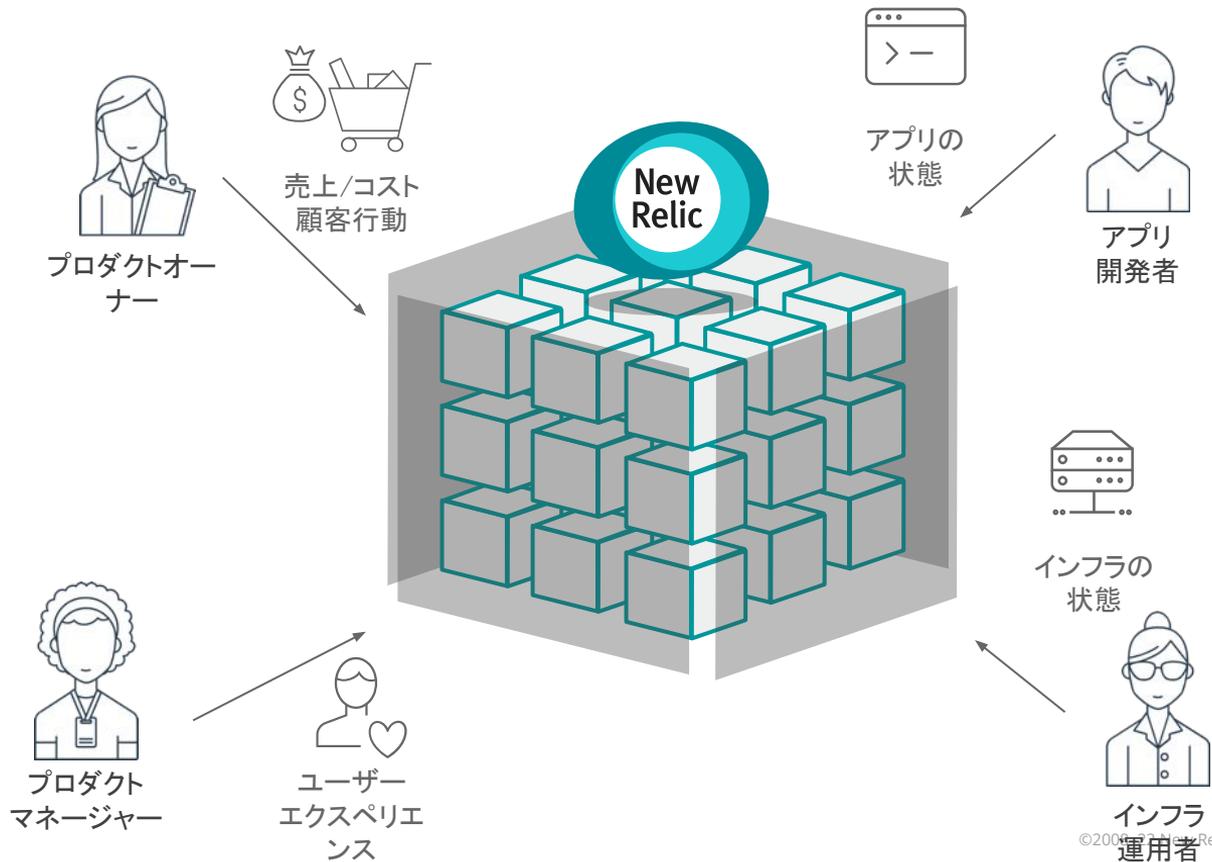
分析用データ集計

# デフォルトのUIでもデータ分析はできます





# 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

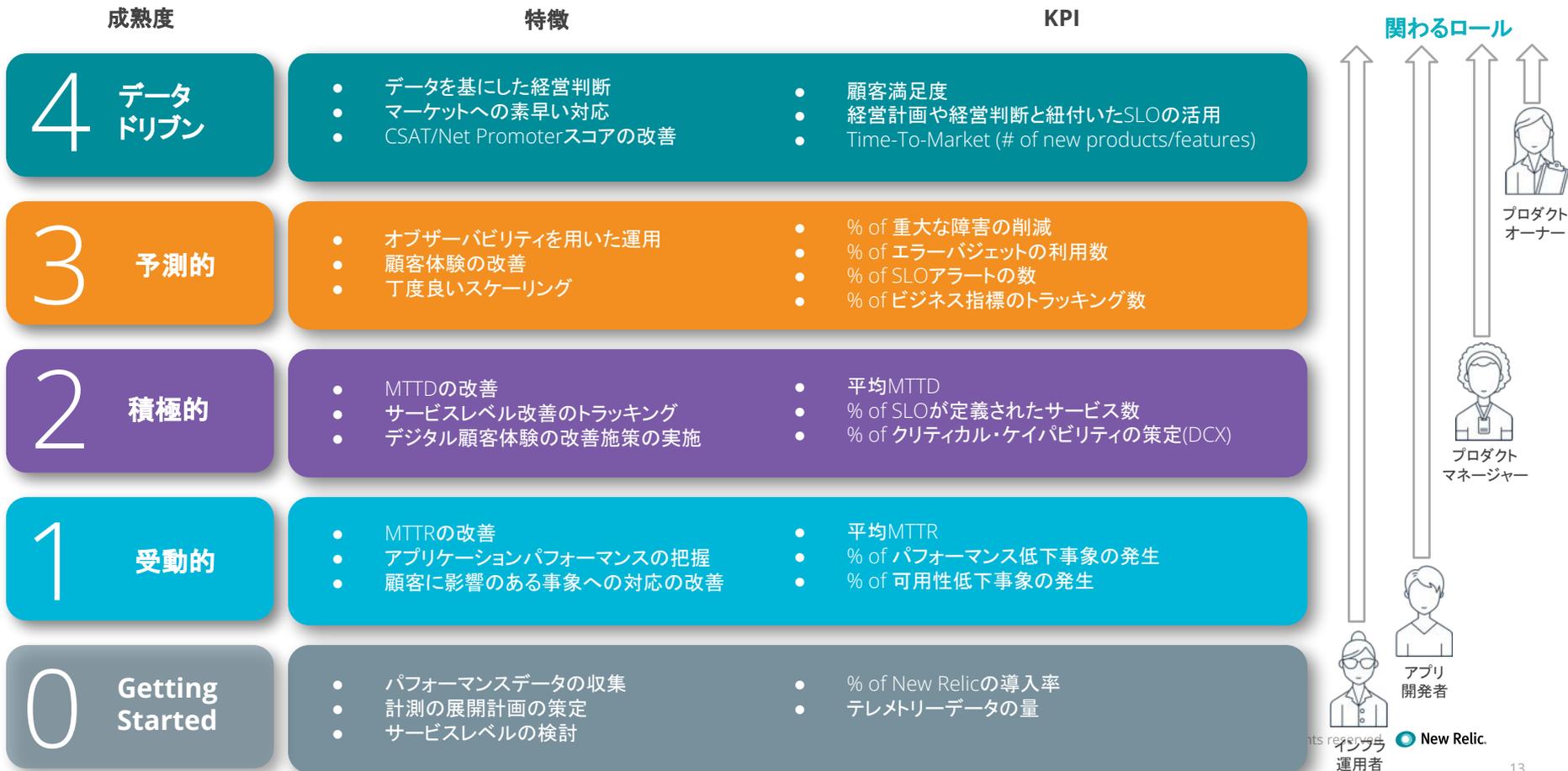


## 達成確認

---

リアルタイムに  
達成度合いを確認する

# New Relic のObservability 成熟モデル



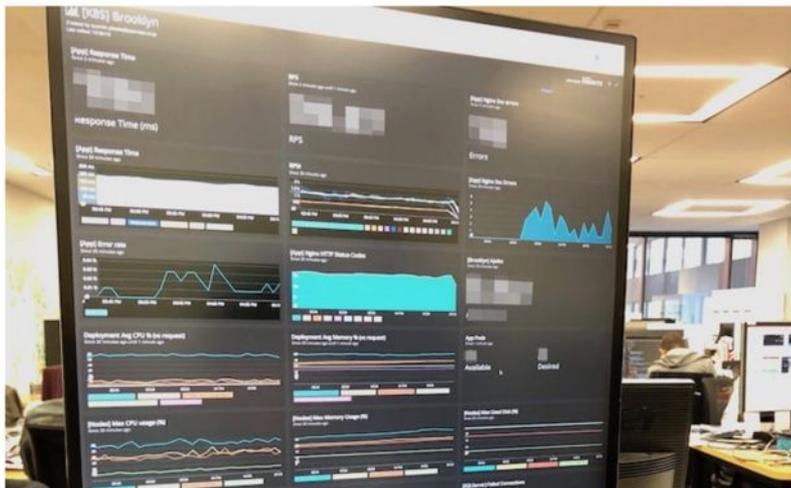
# データ分析を活用いただいているお客様事例

株式会社Mobility Technologies様

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

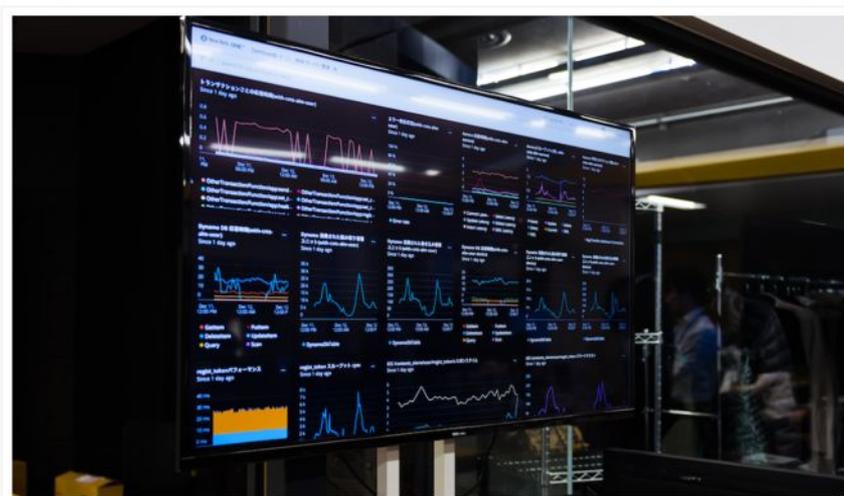
ダッシュボードを起点に天気によって数字が刻々と如実に変わります。天気予報をダッシュボード上に載せたいくらいです。

江尻氏 JapanTaxiソフトウェアエンジニア



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 One ダッシュボード

STEP1

データの  
理解

# New Relicが取得するデータ

# New Relicが取得するデータ: MELT

## M:メトリック

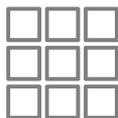
集計可能なデータの粒



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

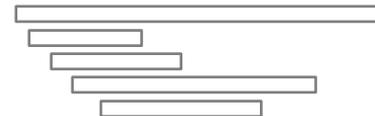
個々の出来事の記録



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

- 各種エージェント(APM,Browser,Mobile,Synthetics,Infrastructure)から送信
- Event APIでカスタムデータを送信  
(<https://docs.newrelic.co.jp/docs/data-apis/ingest-apis/event-api/introduction-event-api/>)

参考: Eventデータについて

<https://docs.newrelic.co.jp/docs/data-apis/understand-data/new-relic-data-types/#event-data>

# 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)	ComputeSample	計算リソースを提供するサービスのメトリックを記録(AWS EC2等)
	DatastoreSample	ストレージキャパシティを提供するサービスのメトリックを記録(AWS S3等)

参考:

<https://docs.newrelic.co.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 (Custom Attribute: <https://docs.newrelic.co.jp/docs/data-apis/custom-data/custom-events/collect-custom-attributes/>)。

## Transaction Eventの例

Transaction

```
"events": {
  {
    "appId": 445000097,
    "appName": "EC-site",
    "databaseCallCount": 2,
    "databaseDuration": 0.00041,
    "duration": 0.11225,
    "entityGuid": "MjUxMTY3MXxBUE18QVBQTE1DQVRJT058NDQ1MDAwMDk3",
    "error": false,
    "guid": "7ee4dbb4c9e14e31",
    "host": "ip-172-31-26-144.ap-northeast-1.compute.internal",
    "http.statusCode": 200,
    "http.responseCode": "200",
    "name": "WebTransaction/Action/block_search_product",
    "priority": 1.42012,
    "realAgentId": 445000099,
    "request.headers.accept": "text/html,application/xhtml+xml,application/json;q=0.9,application/javascript;q=0.9,text/javascript;q=0.9,appli",
    "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,
    "sampled": true,
    "timestamp": 1598476362069,
    "totalTime": 0.11225,
    "traceId": "7ee4dbb4c9e14e31",
    "transactionSubType": "Action",
    "transactionType": "Web"
  },
}
```



# 属性(Eventの構成要素)について

## Transaction

```
"events": [
  {
    "appId": 445000097,
    "appName": "EC-site", アプリケーション名は"appName"という属性
    "databaseCallCount": 2,
    "databaseDuration": 0.00041,
    "duration": 0.11225, Transactionの応答時間は"duration"という属性
    "entityGuid": "MjUxMTY3MXxBUE18QVBQTE1DQVRJT058NDQ1MDAwMDk3",
    "error": false,
    "guid": "7ee4dbb4c9e14e31",
    "host": "ip-172-31-26-144.ap-northeast-1.compute.internal",
    "http.statusCode": 200,
    "httpResponseCode": "200",
    "name": "WebTransaction/Action/block_search_product", Transactionの名前は"Name"という属性
    "priority": 1.42012,
    "realAgentId": 445000099,
```

# ハンズオン - 環境について

New Relicにログインしてください。

New Relic One: <https://one.newrelic.com>

- ユーザー: [japan-handson+2021@newrelic.com](mailto:japan-handson+2021@newrelic.com)
- パスワード: [oSz6nrupas](#)  
(オー、エス、ゼット、ロク、エヌ、アール、ユー、ピー、エー、エス)

[ご注意ください]

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のでプライベートブラウジングをお使いください。また、普段お使いのNew Relicで併せて操作いただいても問題ありません。

- 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
- IE: New Relicの一部機能はIEをサポートしていません。  
上記のいずれかのブラウザをご利用ください。

New Relic.

Log in to your account

EMAIL  
|

Next

[Forgot your password?](#) [Trouble logging in?](#) [Create a free account](#)

# ハンズオン(1) データの理解

このハンズオンのセクションでは、以下の点を学習します。

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



**15:20 - 15:40 (20min)**

Option!!

# ハンズオン(1)

## データの理解

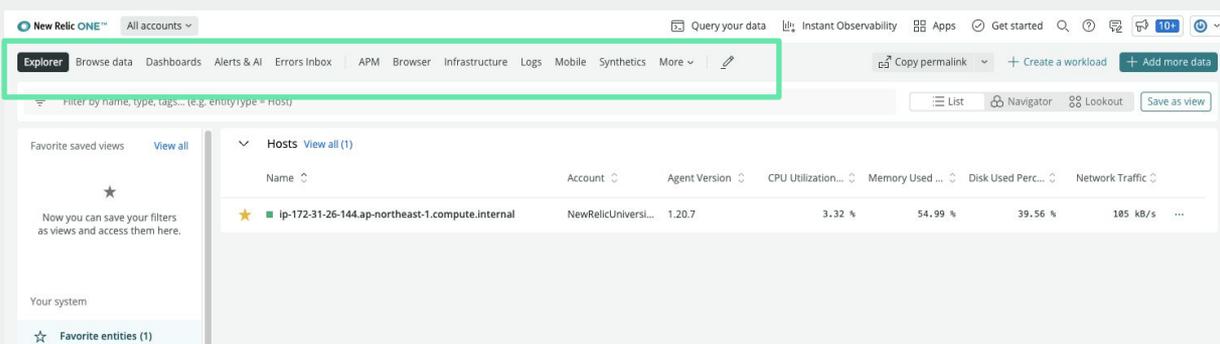
目的:

- イベントデータを確認する

New Relic OneポータルのUIに触れてみる

New Relic Oneにログインし、メニューを触ってみる

- メニューバーの中から数個メニューをクリックする
  - Dashboard
  - APM
  - Infrastructure
  - Logs



# ハンズオン(1) データの理解

目的:

- イベントデータを確認する

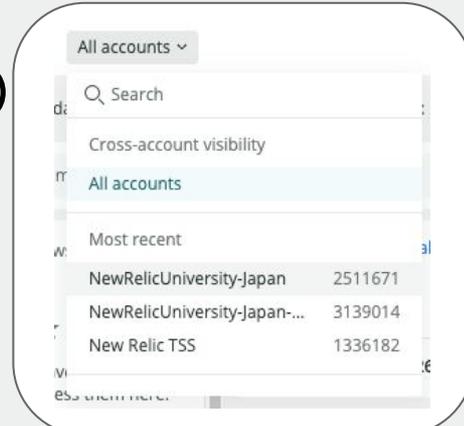
イベントデータを確認するための手順  
を実施する

New Relic Oneにログインし、  
Explorerを参照する

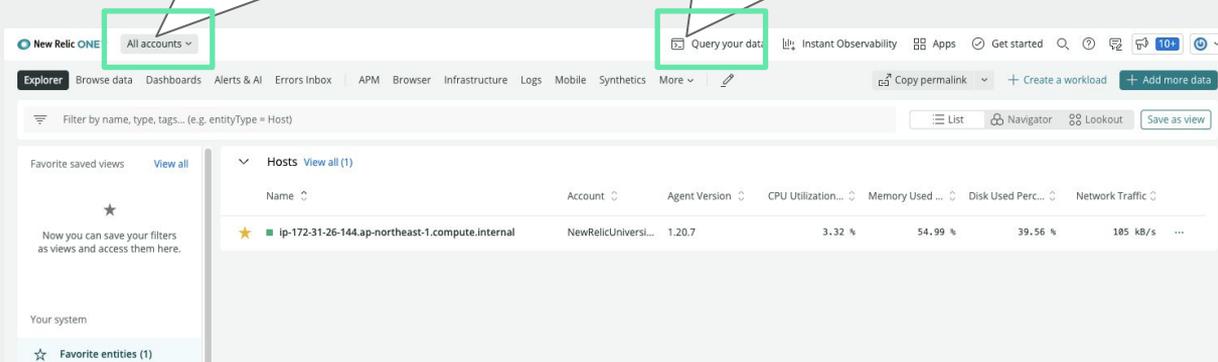
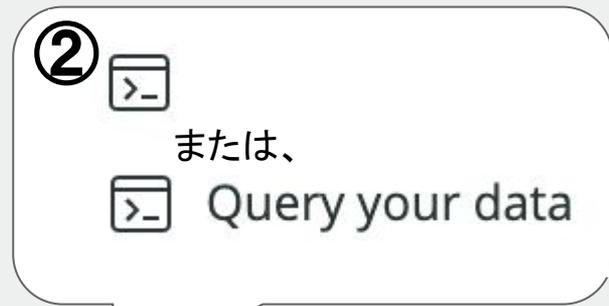
- All accountsから、  
NewRelicUniversity-Japanを  
選択する
- Query your dataをクリックする

ハンズオン

①



②



# ハンズオン(1) データの理解

目的:

- イベントデータを確認する

イベントデータを確認するための Data explorerを利用する

- Data explorerをクリックする
- Event typeからTransactionを選択します
- JSONをクリックします

どのようなデータを参照できるかを確認します

次のページに続きます

- Plotをクリックします

The screenshot shows the New Relic ONE Data Explorer interface. The 'Data explorer' tab is active, and the 'Transaction' event type is selected in the left sidebar. The main panel displays a query: `SELECT count(*) FROM Transaction SINCE 30 MINUTES AGO TIMESERIES`. The results are shown in a JSON format, with the 'JSON' button highlighted. The JSON data includes fields such as `appId`, `appName`, `databaseCallCount`, `duration`, `entity.guid`, `entityGuid`, `error`, `guid`, `host`, `http.statusCode`, `http.responseCode`, `name`, `priority`, `realAgentId`, `request.headers.accept`, `request.headers.host`, `request.method`, `request.uri`, `response.headers.contentType`, `response.statusCode`, `sampled`, `timestamp`, `totalTime`, `traceId`, `transactionSubType`, and `transactionType`.

① Data explorer

② Transaction

③ JSON

# ハンズオン(1) データの理解

目的:

- イベントデータを確認する

イベントデータを確認するための Data explorerを利用する

- Raw dataをクリックし、個々のイベントの内容を確認する
- Plot領域から、どのような数値属性があるかを確認する
- Dimensions領域から、どのような文字列属性があるかを確認する

右上の”Close X”をクリックすると元の Explorer画面に戻ります

The screenshot shows the New Relic ONE Data Explorer interface. The left sidebar is annotated with a green box and the number 2, containing the 'Plot' and 'Dimensions' sections. The 'Plot' section has a search bar and a list of metrics with 'Average' filters. The 'Dimensions' section is partially visible. The main area shows a query editor with the following SQL: `NRQL SELECT count(*) FROM Transaction SINCE 30 MINUTES AGO TIMESERIES`. Below the query is a table of results for 'Transaction' events. The table has columns: TIMESTAMP, APP ID, APP NAME, DATABASE CALL COUNT, DATABASE DURATION, and DURATION. The 'Raw data' button in the top right of the table is annotated with a red circle and the number 1. The 'Close X' button in the top right corner of the interface is annotated with a red circle and the number 3.

TIMESTAMP	APP ID	APP NAME	DATABASE CALL COUNT	DATABASE DURATION	DURATION
November 07, 2021 15:28:29	4450000...	EC-site	25	0.00706	0.0667
November 07, 2021 15:28:28	4450000...	EC-site	2	0.0002	0.0488
November 07, 2021 15:28:27	4450000...	EC-site	19	0.00492	0.0387
November 07, 2021 15:28:27	4450000...	EC-site	50	0.012	0.0644
November 07, 2021 15:28:24	4450000...	EC-site	18	0.007	0.0761
November 07, 2021 15:28:24	4450000...	EC-site	11	0.00476	0.0707
November 07, 2021 15:28:22	4450000...	EC-site	10	0.00296	0.0509
November 07, 2021 15:28:21	4450000...	EC-site	12	0.00376	0.0578
November 07, 2021 15:28:20	4450000...	EC-site	10	0.00305	0.0338
November 07, 2021 15:28:20	4450000...	EC-site	12	0.00379	0.0447
November 07, 2021 15:28:18	4450000...	EC-site	11	0.00527	0.0746
November 07, 2021 15:28:18	4450000...	EC-site	2	0.0002	0.0525
November 07, 2021 15:28:14	4450000...	EC-site	2	0.0002	0.0535
November 07, 2021 15:28:10	4450000...	EC-site	2	0.00035	0.0516
November 07, 2021 15:28:08	4450000...	EC-site	2	0.00021	0.0487
November 07, 2021 15:28:06	4450000...	EC-site	2	0.0002	0.0513

# ハンズオン(1) データの理解

目的:

- カスタムイベントを送信する

カスタムイベントを送信するための事前準備を実施します。

APIキーを作成します。

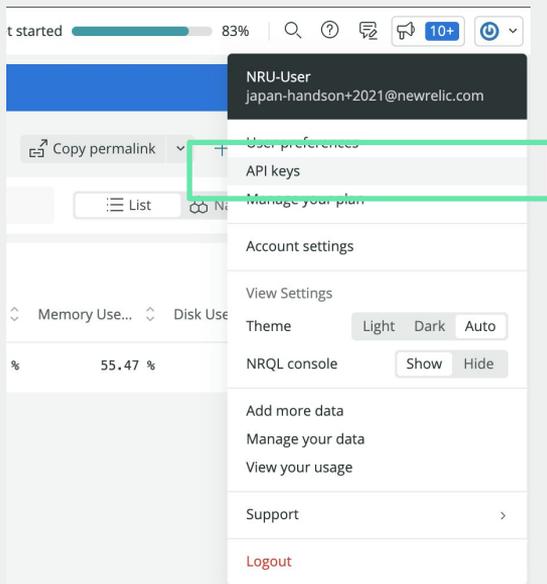
- NewRelicUniversity-Japanアカウントを指定し、Ingest - Licenseキーを作成する  
※Nameには、「NRU302-<企業名>-<イニシャル>」のように重複を避ける名前を設定してください。

参考:

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

ハンズオン

オプション演習



Create a key



## Create an API key

Ingest keys are for getting data into New Relic:

- License keys for agent configuration and metric, event, log and trace APIs
- Browser keys for browser applications
- Mobile keys for mobile applications

User keys are for querying data and managing configurations (Alerts, Synthetics, dashboards, etc.)

To learn more about API keys, [see our docs](#)

### Account

Account: 2511671 - NewRelicUniversity-Japan

### Key type

Ingest - License

Name 例: NRU302-<企業名>-<イニシャル>

my\_key\_name

### Notes

What do you want people to know about this key?

Cancel

Create a key

# ハンズオン(1) データの理解

## 目的:

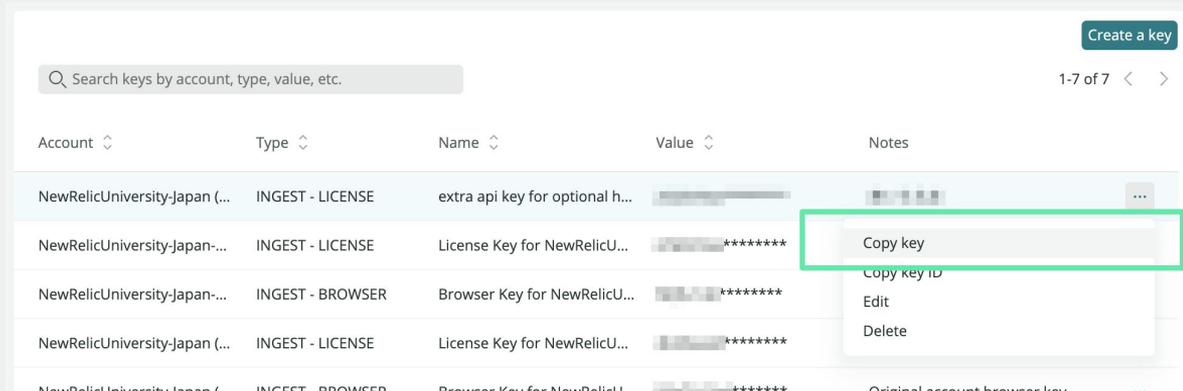
- カスタムイベントを送信する

## 作成したKeyの取得

- 先ほど作成したライセンスを選択して、[...]のプルダウンメニューからCopy keyにてKeyをコピーします。

## 参考:

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



## 補足:

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致します。

# ハンズオン(1) データの理解

## 目的:

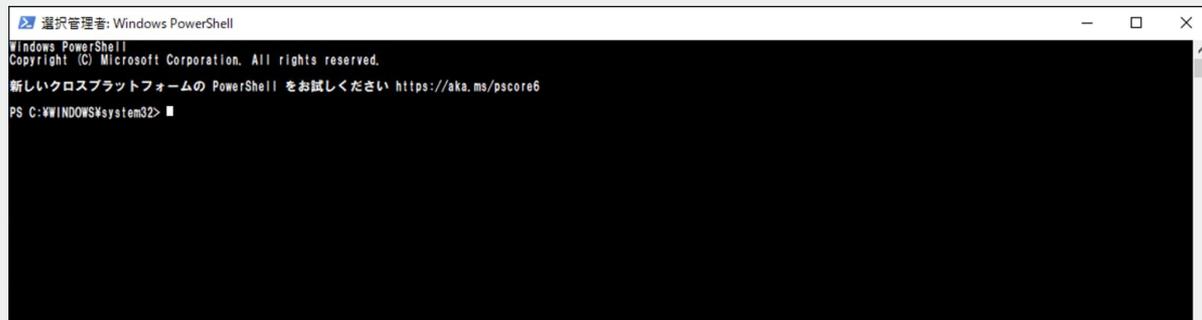
- カスタムイベントを送信する

ターミナルにて作業を行います。

- Windows: Powershellを開く
- Mac: ターミナルを開く

## 参考:

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



```
選択管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6
PS C:\WINDOWS\system32>
```



```
Macintosh HD — zsh — 129x26
NewRelic$
```

# ハンズオン(1) データの理解

## 目的:

- カスタムイベントを送信する

## Windows上での作業

- 右のスクリプトを参考にし、.ps1ファイルとして保存します。  
参考情報: ([Github](#))
- 保存後、Powershell上で当該スクリプトを実行します。

Windowsの方はこちらです (Macは次ページ)

```
$accountId = "2511671"
$insertkey = "(控えておいたキー)"
# Replace with your custom event for the body
$body = "[{"eventType": "NRULab", "labnumber": 1, "initial": "ご自身のイニシャルに書き換えてください", "comment": "任意の文字列を入れてください(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"
```

## 参考:

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

# ハンズオン(1) データの理解

## 目的:

- カスタムイベントを送信する

## Mac上での作業

- 右のJSONファイルを参考にし、ローカルにnrulab.jsonとして保存します。  
参考情報: ([Github](#))
- ターミナルを起動し、nrulab.jsonを保存したディレクトリに移動します。移動後、右のコマンドを実行します。  
参考情報: ([Github](#))

## 参考:

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

Macの方はこちらです (Windowsは前ページ)

```
[
  {
    "eventType": "NRULab",
    "labnumber": 1,
    "initial": "ご自身のイニシャルに書き換えてください",
    "comment": "任意の文字列を入れてください(日本語可)"
  }
]
```

“nrulab.json”としてローカルに保存する



```
gzip -c nrulab.json | curl --data-binary @- -X POST -H "Content-Type: application/json" -H "X-Insert-Key: (控えておいたキー)" -H "Content-Encoding: gzip" https://insights-collector.newrelic.com/v1/accounts/2511671/events
```

“nrulab.json”のあるディレクトリで実行する

# ハンズオン(1) データの理解

目的:

- カスタムイベントを送信する

送信したカスタムイベントを参照します。

- Explorer UIに戻ります。  
補足: UI上で迷ったら、左上の New Relic Oneのロゴをクリックして下さい。
- 右上中程にある Query your dataのアイコンをクリックします。

参考:

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

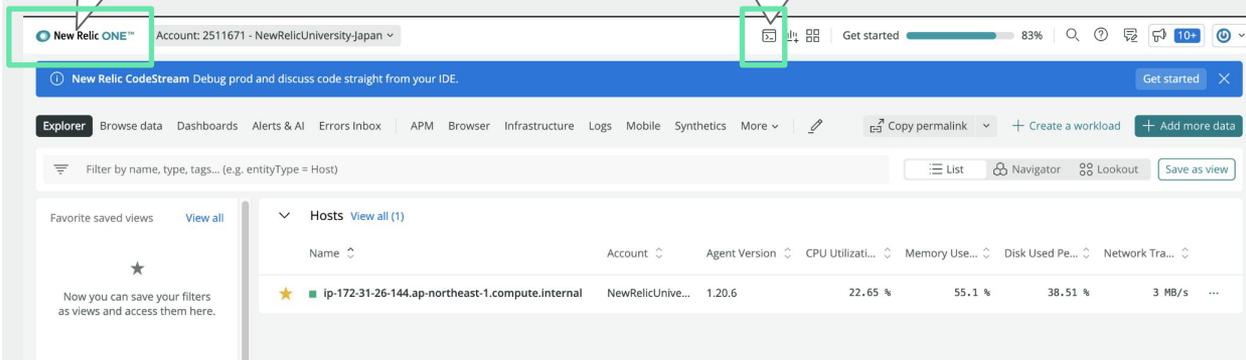
補足: トップ画面に戻ります。



または、



Query your data



# ハンズオン(1) データの理解

目的:

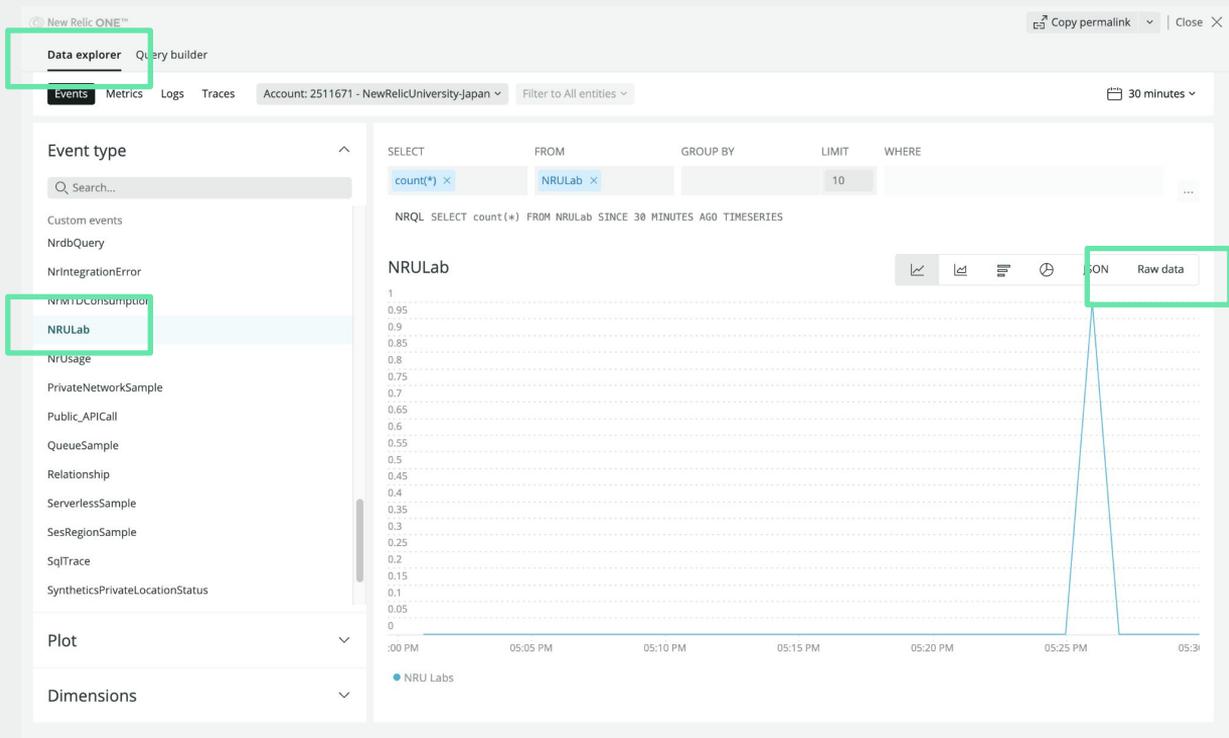
- カスタムイベントを送信する

送信したカスタムイベントを参照します。

- Data explorerを選択し、Event typeがNRULabを選択します。
- 受信したデータ内容を参照する場合、Raw dataをクリックします。

参考:

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



STEP2

分析手法  
の習得

# NRQL (New Relic Query Language)

# NRQLとは

- New Relic Query Language の略
- Eventデータを含め、New Relic DBに格納されたデータを分析するためのクエリ言語
  - クエリを直接書くこともできるが、Data explorerやクエリービルダーのBasic機能を補助的に利用することも可能



# Data Explorer

- GUIで実データを見ながらデータを選択し、分析することができる機能

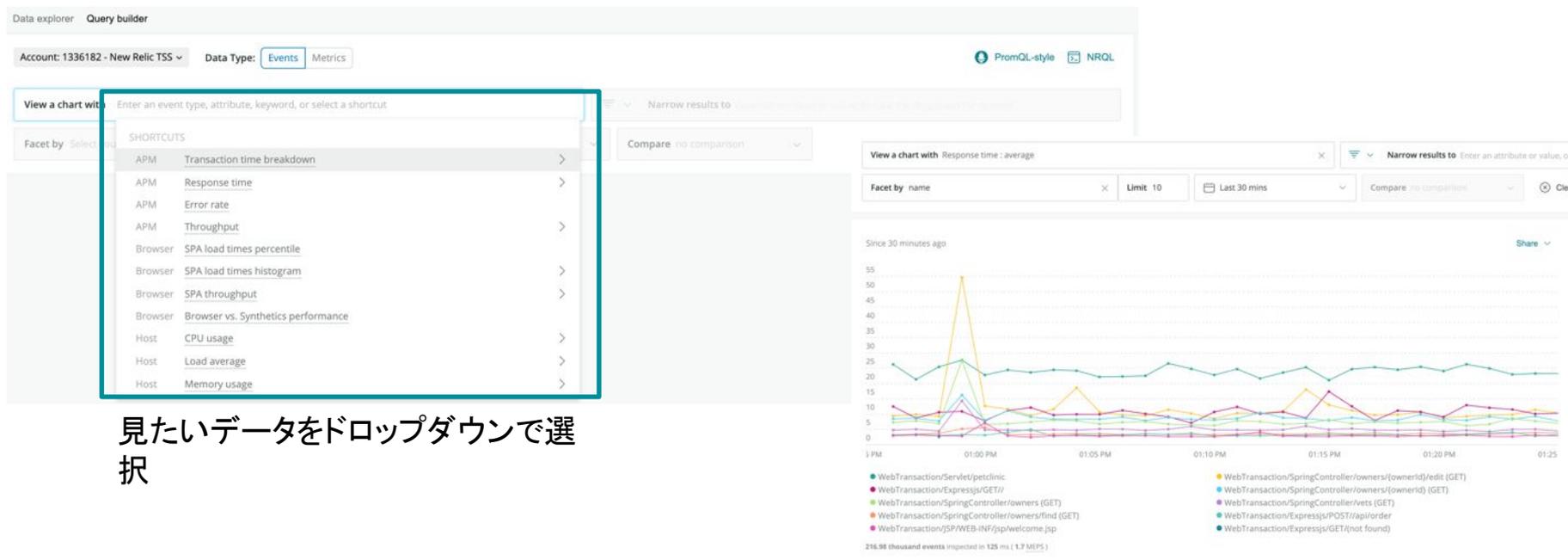


結果を右メニューで表示

見たいデータを左メニューで選択

# クエリビルダー(Basic)

- ドロップダウンでデータを選択し、分析することができる機能



見たいデータをドロップダウンで選択

# クエリビルダー(Basic)の使い方

チャートに表示したいデータを選択  
この例では、TransactionというEventから  
durationという属性を選択し、平均値を出す

データの絞り込み(オプション)  
例.特定のアプリのデータのみを表示する

The screenshot shows the New Relic Query Builder interface for account 'Demotron V2'. The 'Data Type' is set to 'Events'. The main query is 'View a chart with Transaction : duration : average'. Below the query, there are several options: 'Facet by name', 'Limit 10', 'Last 30 mins' (selected), and 'Compare no comparison'. A filter is applied: 'Narrow results to appName = 'Plan ...rvice''. The interface includes a 'Clear' button and a '1' indicator next to the filter.

データのグループ分け(オプション)  
例.トランザクション名ごとにデータを分けて表示

表示させるデータ期間

比較のための過去の  
データを重ねて表示  
(オプション)

# 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
- Data ExplorerやBasicよりも柔軟なデータ分析が可能

```
Account Demotron V2 Switching to basic mode will erase any changes you make here. Basic  
SELECT average(duration) FROM Transaction WHERE appName = 'Plan Service' TIMESERIES FACET `name` LIMIT 10 SINCE 1800 seconds ago EXTRAPOLATE  
My recent queries Clear Run
```

## 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: <https://docs.newrelic.com/docs/query-data/nrql-new-relic-query-language/getting-started/introduction-nrql>

# NRQL使いこなしTips①

- この句だけは覚えましょう！
  - **FROM イベント名** : どの名前のイベント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Tips②

- 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Tips③

- 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

# ハンズオン(2) 分析手法の習得

このハンズオンのセクションでは、以下の点を学習します。

- ダッシュボードの大枠を作る
- Data Explorerを用いて、チャートをダッシュボードに追加する
- NRQLを用いてチャートを編集する
- クエリビルダーでチャートを作成する



**15:50 - 16:10 (20min)**

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

ダッシュボード作成を通してNRQLの  
利用方法を身につける

New Relic Oneにログインし、メ  
ニューからDashboardsをクリックする

- Create a dashboardボタンをクリックする
- 右側より画面が出てくるので、  
Create a new dashboardを選  
択します

ハンズオン

準備

The screenshot shows the New Relic ONE interface. At the top, the account name is 'New Relic ONE™ Account: 2511671 - NewRelicUniversity-Japan'. The navigation bar includes 'Explorer', 'Browse data', 'Dashboards', 'Alerts & AI', 'Errors Inbox', 'APM', 'Browser', 'Infrastructure', 'Logs', 'Mobile', 'Synthetics', and 'More'. The 'Dashboards' menu item is highlighted with a green box and a circled '1'. Below the navigation bar, there is a filter input field: 'Filter by name, type, tags... (e.g. entityType = Host)'. The main content area shows 'Favorite saved views' with a 'View all' link and a star icon. Below this, a message states: 'Now you can save your filters as views and access them here.' To the right, there are sections for 'Services - APM View all (1)' and 'Hosts View all (1)'. The 'Hosts' section shows a table with columns for 'Name', 'Account', 'Agent Vers...', and 'CPU U...'. One host is listed: 'ip-172-31-26-144.ap-northeast-1.compute.int...' with account 'NewRelicUniv...' and version '1.20.7'. At the bottom of the page, there are buttons for 'Copy permalink', 'Import dashboard', and 'Create a dashboard'. The 'Create a dashboard' button is highlighted with a green box and a circled '2'. A green arrow points from the 'Dashboards' menu item to this button.

# ハンズオン(2)分析手法の習得

目的:

- NRQLの利用方法を取得する

ダッシュボード作成を通してNRQLの利用方法を身につける

以下の点に注意して、ダッシュボードの作成を進めます

- Dashboard name
  - 受講される方々の間で重複しない様にする
    - 設定例: “ご自身の会社名\_お名前”
- Select your account
  - Account: 2511671 - NewRelicUniversity-japanが選択されていることを確認する

Createボタンをクリックします

Create a dashboard

Dashboard name

e.g. Awesome dashboard

Select your account

Account: 2511671 - NewRelicUniversity-japan ▾

Back

Create



Dashboard name

テスト株式会社\_khara

Select your account

Account: 2511671 - NewRelicUniversity-japan ▾

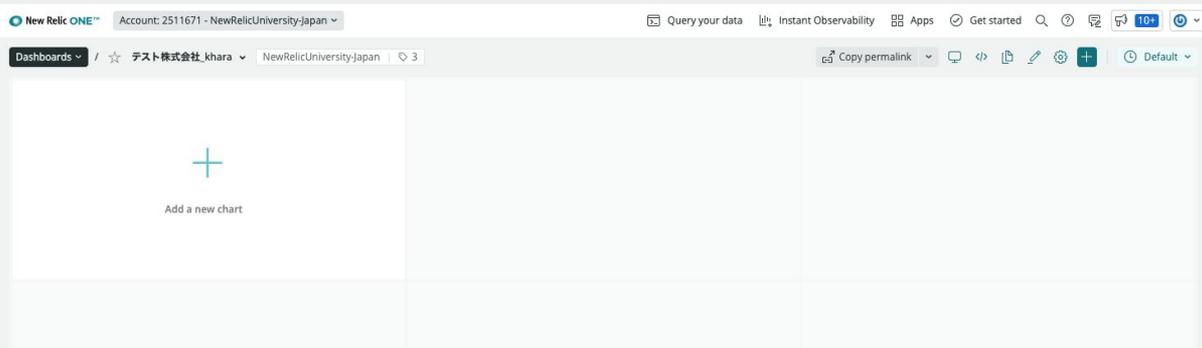
# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

ダッシュボード作成を通してNRQLの  
利用方法を身につける

右の様な画面が表示されれば、準備  
が完了です



この画面が出てきたら、準備完了です。

もし、他の画面に遷移してしまった場合、メニューより Dashboards をクリックし、一覧から作成したダッシュボード名を選択してください。

# ハンズオン(2) 分析手法の習得

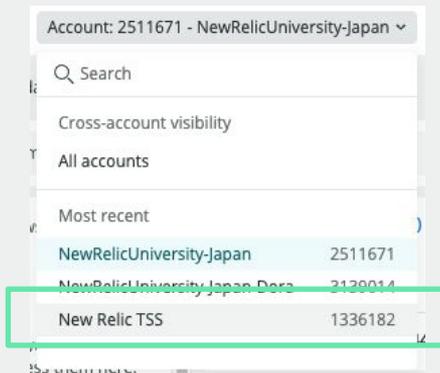
目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

ハンズオン上のポイント

- New Relic TSSというアカウントのデータを利用します
- Transaction Eventを解析します
- 作成したダッシュボードに解析した結果をチャートとして追加します



具体的な切り替え方法や指定方法はハンズオン内で説明しておりますので、ご安心下さい

以下のデータを抽出します

- トランザクション総数を用います ( count(\*) )
- 特定アプリからのデータだけを抽出します (属性名: appName、WHERE appName = 'New Relic Pet Clinic')
- トランザクション名ごとにわけて分析します (属性名: name、FACET name)
- 過去3時間のデータで分析します (SINCE 3 hours ago)

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerを起動し、解析するデータを取得する

- Data explorer上で、New Relic TSSアカウントに変更します
- Event typeからTransactionを選択します
- Plotからcount(\*)を選択します

The screenshot illustrates the steps for configuring a query in the New Relic Data Explorer. It is divided into three numbered sections:

- ① Account Selection:** A search box at the top shows a list of accounts. The account "New Relic TSS" with ID "1336182" is highlighted with a green box. A callout box points to this selection.
- ② Event Type Selection:** The "Event type" dropdown menu is open, showing a list of event categories. "Transaction" is highlighted with a green box.
- ③ Plot Configuration:** The "Plot" configuration panel is shown. The "count(\*)" metric is selected and highlighted with a green box. Other metrics like "Build", "customer", "databaseCallCount", "databaseDuration", "duration", and "error" are listed with "Average" aggregation.

Green arrows indicate the flow from the account selection to the event type selection, and then to the plot configuration.

# ハンズオン(2) 分析手法の習得

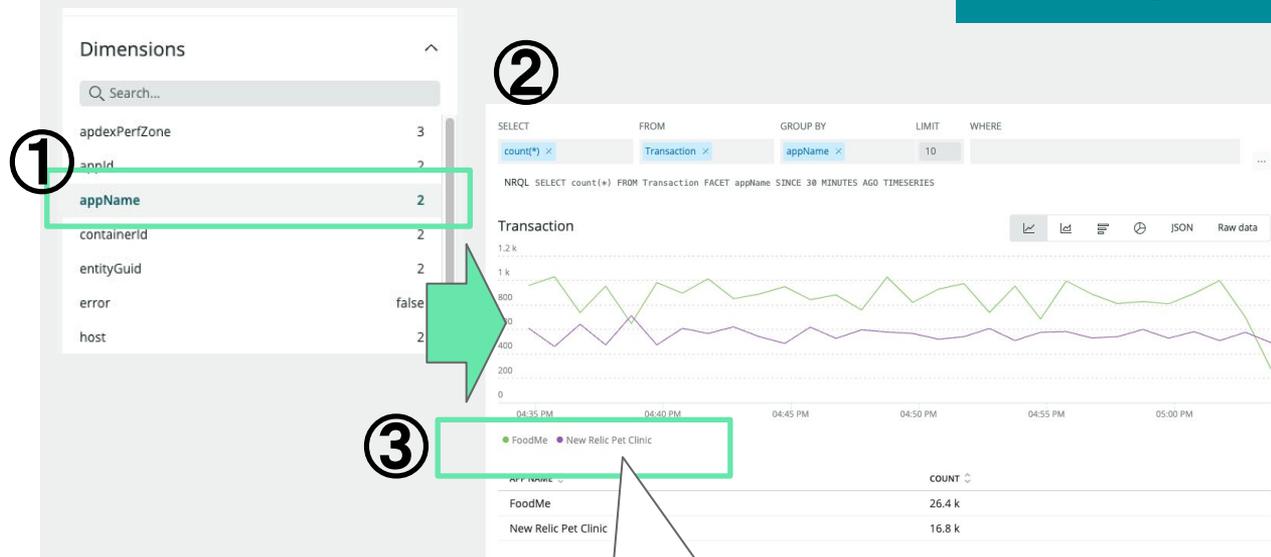
目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerを起動し、解析するデータを取得する

- Dimensionsを開き、appNameをクリックします
- 右側のエリアのグラフが変わり、appNameの値が表示されます
- New Relic Pet Clinicをクリックし、グラフの表示が変わるのを確認します



● FoodMe ● New Relic Pet Clinic

● FoodMe ● New Relic Pet Clinic

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerを起動し、解析するデータを取得する

- Dimensionsを開き、nameをクリックします
- 右側のエリアのグラフが変割るのを確認します
- 【重要】NRQLが表示されていることを確認します

①

Dimensions

Q Search...

apdexPerfZone	3
appid	2
appName	2
containerid	2
entityGuid	2
error	false
host	2
httpResponseCode	5
httpResponseMessage	4
<b>name</b>	<b>11</b>
realAgentId	7
request.headers.accept	5
request.headers.contentLe...	12

②

SELECT count(\*) FROM Transaction FACET name SINCE 30 MINUTES AGO TIMESERIES

Transaction

04:50 PM 04:55 PM 05:00 PM 05:05 PM 05:10 PM 05:15 PM

- WebTransaction/Expressjs/GET//
- WebTransaction/Servlet/petclinic
- WebTransaction/Expressjs/GET//api/restaurant
- WebTransaction/SpringController/owners/{ownerid} (GET)
- WebTransaction/SP/WEB-INF/jsp/welcome.jsp
- WebTransaction/SpringController/owners/{ownerid}/edit (GET)
- WebTransaction/SpringController/vets (GET)

NAME	COUNT
WebTransaction/Expressjs/GET//	23.9 k
WebTransaction/Servlet/petclinic	12.6 k
WebTransaction/SpringController/owners/{ownerid} (GET)	1.08 k
WebTransaction/Expressjs/GET//api/restaurant	655
WebTransaction/SP/WEB-INF/jsp/welcome.jsp	599
WebTransaction/SpringController/owners (GET)	597
WebTransaction/SpringController/vets (GET)	597

③

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerで解析したグラフをダッシュボードに追加する

- プルダウンより3 hoursに変更する
- [...]からAdd to dashboardを選択します。

[次のスライドに進みます]

Account: 1336182 - New Relic TSS | Filter to All entities

SELECT count(\*) FROM Transaction GROUP BY name LIMIT 10

NRQL SELECT count(\*) FROM Transaction FACET name SINCE 30 MINUTES AGO TIMESERIES

Transaction

1.2 k  
1 k  
800  
600  
400  
200  
0

08:40 PM 08:45 PM 08:50 PM 09:00 PM 09:05 PM

Get as image  
Get chart link  
Create alert condition  
Add to dashboard

COUNT  
24.5 k

① 30 minutes

②

©2008–22 New Relic, Inc. All rights reserved | New Relic

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerで解析したグラフをダッシュボードに追加する

- 設定画面が右側からスライドして表示されるので、任意のタイトルを設定します。
- リストより、準備の段階で作成したダッシュボードを選択します。
- Copy to dashboardをクリックします

① Copy to a dashboard  
Select a dashboard where you would like to add the widget.

Widget title  
試験用ウィジェット

Select an existing dashboard  
We excluded dashboards you don't have permission to edit.

Search by name, account or creator

Dashboard Name	Account
2W-SummerGames / 2W-SummerGa...	New Reli...
SP-SummerGames / SP-SummerGa...	New Reli...
SV-SummerGames / SV-SummerGa...	New Reli...
tanaka-20211104-test / tanaka-2021...	NewReli...
TRA-SummerGames / TRA-Summer...	New Reli...
VV-SummerGames / VV-SummerGa...	New Reli...
テスト株式会社_khara / テスト株式会...	NewReli...
蜆-dashboard / 蜆-dashboard	NewReli...

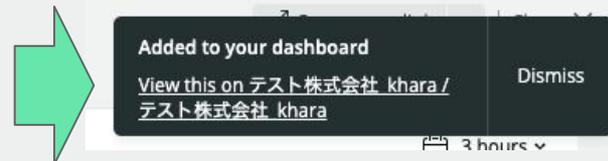
②

③

Cancel Copy to dashboard

Or create a new dashboard

ハンズオン



この表示が出れば、ダッシュボードへの追加が行われました。追加されたダッシュボードを参照してください。

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

ダッシュボードからNRQLを確認し、  
編集する

チャートに設定されたNRQLを確認し  
ます

- 追加したチャートの[...]から  
View queryを選択します
- 画面下からNRQLがスライドし  
て表示されることを確認します

New Relic ONE™ Account: 2511671 - NewRelicUniversity-Japan

Dashboards / ☆ テスト株式会社\_khara / NewRelicUniversity-Japan / 8

Search for any attribute or value.

試験用ウィジェット  
Since 3 hours ago

2 k  
1.5 k  
1 k  
500  
0

4 06:30 PM 07:00 PM 07:30 PM 08:00 PM 08:30 PM 09:00 PM 05

- WebTransaction/Expressjs/GET//
- WebTransaction/Servlet/petclinic
- WebTransaction/SpringControlle...
- WebTransaction/Expressjs/GET//...
- WebTransaction/SpringControlle...
- WebTransaction/SpringControlle...
- WebTransaction/SPD/MER\_INI/...
- WebTransaction/SpringControlle...

①

Expand  
Get as image  
Get chart link  
Create alert condition  
View query  
Copy  
Edit  
Duplicate  
Delete

```
Account: 1336182 - New Relic TSS  
SELECT count(*) FROM Transaction FACET name SINCE 3 HOURS AGO TIMESERIES|
```

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

ダッシュボードからNRQLを確認し、  
編集する

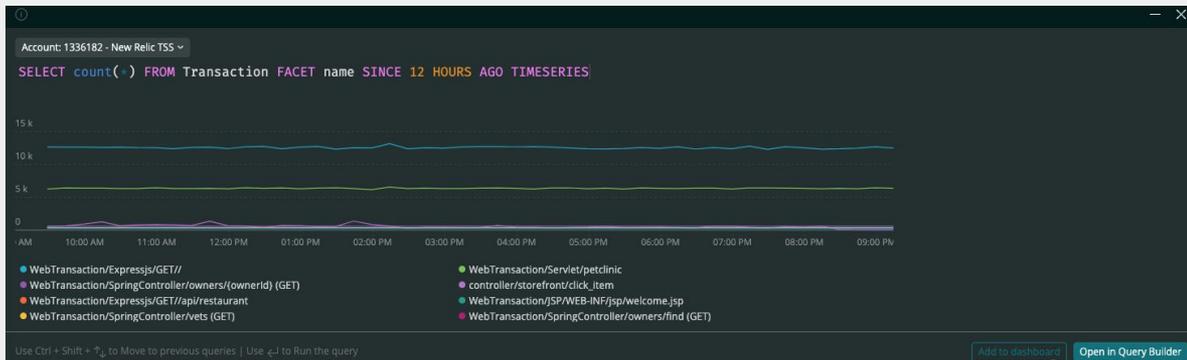
チャートに設定されたNRQLを変更し  
ます

- 表示されているNRQLの3 HOURSを12 HOURSに変更し、リターンを押します
- NRQLに併せて、チャートが表示されることを確認します
- Add to dashboardとOpen in Query Builderボタンが表示されていることを確認します

①

```
Account: 1336182 - New Relic TSS ▾
SELECT count(*) FROM Transaction FACET name SINCE 3 HOURS AGO TIMESERIES|
```

②



③

Add to dashboard

Open in Query Builder

# ハンズオン(2) 分析手法の習得

目的:

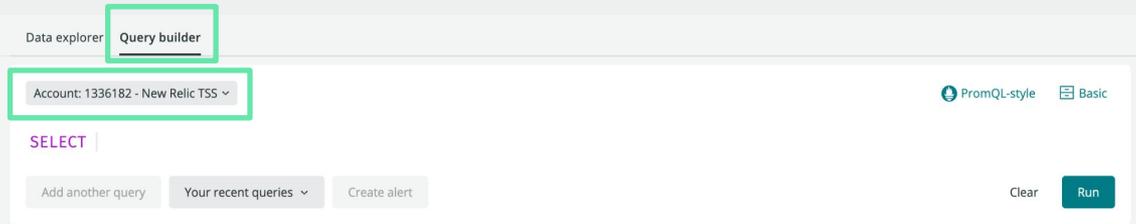
- NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集し、解析結果をダッシュボードに追加する

クエリビルダー上で、NRQLを編集します

- Query your dataからクエリクエリビルダーを開きます
- 右のNRQLをクエリビルダーに入力します
- 右下にあるRunボタンをクリックし、チャートが描画されることを確認します

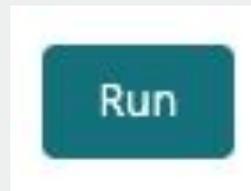
①



②

```
SELECT count(*) FROM Transaction FACET name SINCE 3 HOURS AGO TIMESERIES
```

③



# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集し、解析結果をダッシュボードに追加する

クエリビルダー上で、NRQLを編集します

- WHERE句を追加します
- FACET句を変更します
- 表示時間を変更します
- TIMESERIESを削除します

ハンズオン

① 追加 WHERE appName IN('FoodMe','New Relic Pet Clinic')

② 変更 FACET appName

③ 変更 SINCE 1 HOUR AGO

④ 削除 TIMESERIES

Run

Run

Run

Run

```
SELECT count(*) FROM Transaction FACET appName SINCE 1 HOUR AGO WHERE appName IN ('FoodMe','New Relic Pet Clinic')
```

Account: 1336182 - New Relic TSS

PromQL-style Basic

```
SELECT count(*) FROM Transaction FACET appName SINCE 1 HOUR AGO WHERE appName IN ('FoodMe','New Relic Pet Clinic')
```

Add another query Your recent queries Create alert Clear Run

Since 1 hour ago

APP NAME	COUNT
FoodMe	52.9 k
New Relic Pet Clinic	33.3 k

Chart name Enter a chart name

Chart type Table More visualizations in I/O

Other groups

# ハンズオン(2) 分析手法の習得

目的:

- NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集し、解析結果をダッシュボードに追加する

作成したチャートをダッシュボードに追加する

- チャート画面の右下にある Add to Dashboard から、任意の名前をつけてチャートを登録します
- ダッシュボードを確認します

①

②

③

# ハンズオン(2)

## 分析手法の習得

目的:

- NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集し、解析結果をダッシュボードに追加する

作成したチャートをダッシュボードに追加する

- 追加したチャートから View queryを選択し、SELECT句を変更する
  - 応答時間の平均と90パーセンタイルを求めるように変更する
- 時系列データとして表示する



- 応答時間は、duration属性を用いる
- 平均は、average()関数を用いる
- 90パーセンタイルは、percentile(属性名,パーセンタイル数)関数を用いる

SELECT average(duration), percentile(duration,90) FROM Transaction FACET appName SINCE 1 HOUR AGO WHERE appName IN ('FoodMe','New Relic Pet Clinic')

Run

SELECT average(duration), percentile(duration,90) FROM Transaction FACET appName SINCE 1 HOUR AGO WHERE appName IN ('FoodMe','New Relic Pet Clinic') **TIMESERIES**

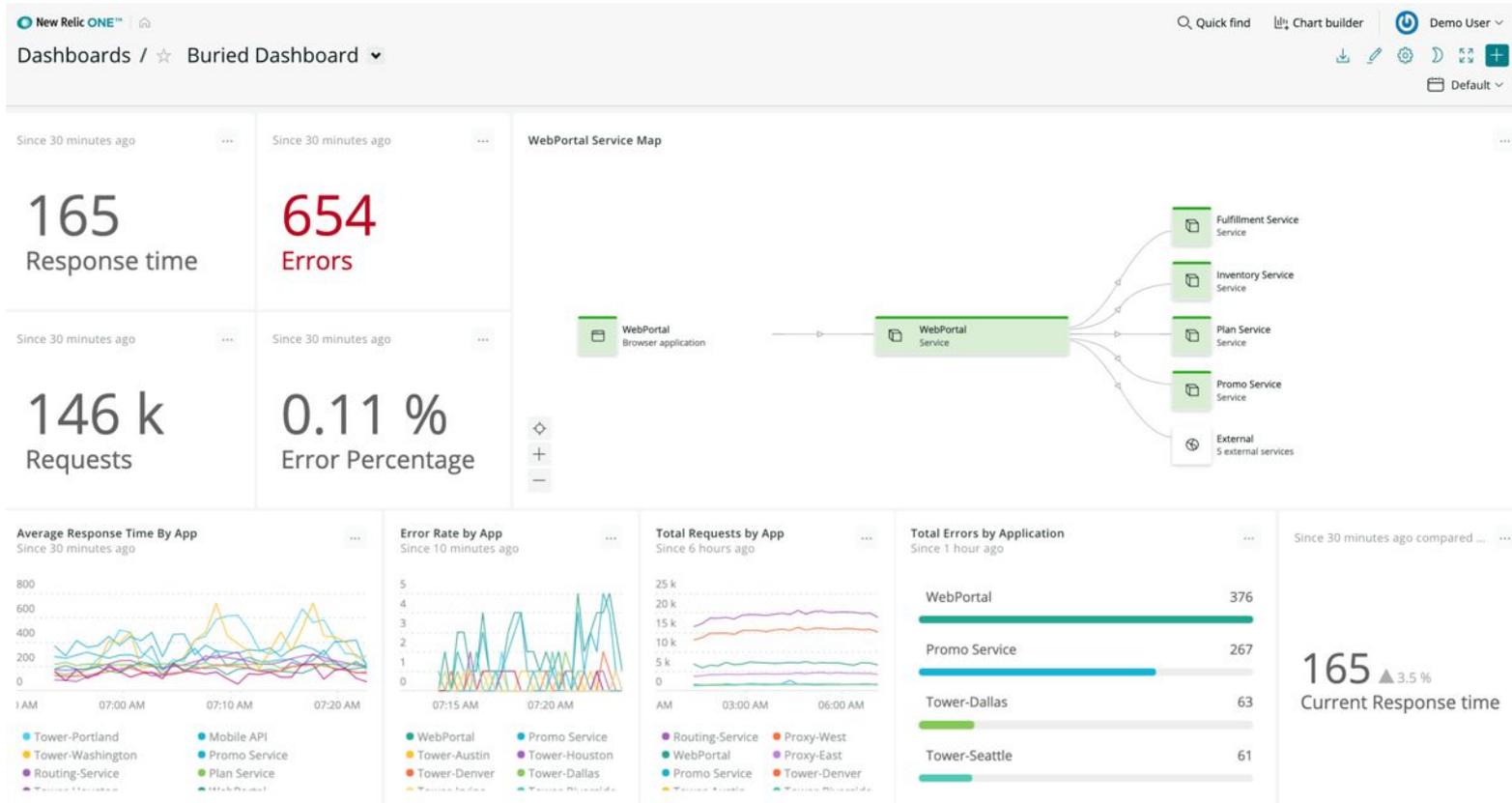
Run

STEP3-1

ダッシュ  
ボードの  
作成

# New Relic One ダッシュボード

# New Relic One 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2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

Since 30 minutes ago

Share ▾

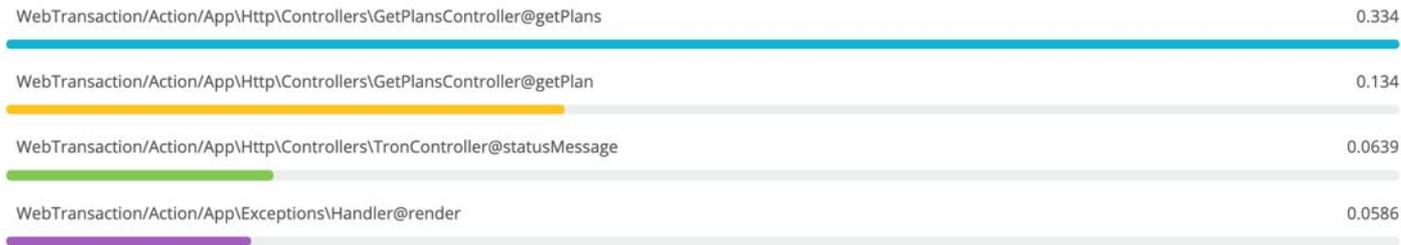


CHART NAME

Enter a chart name

Chart type Bar

JSON

Line

Area

**Bar**

Bullet

Billboard

Table

Pie

Histogram

Heatmap

Funnel

# ダッシュボードの便利機能(1/2)

- 表示系

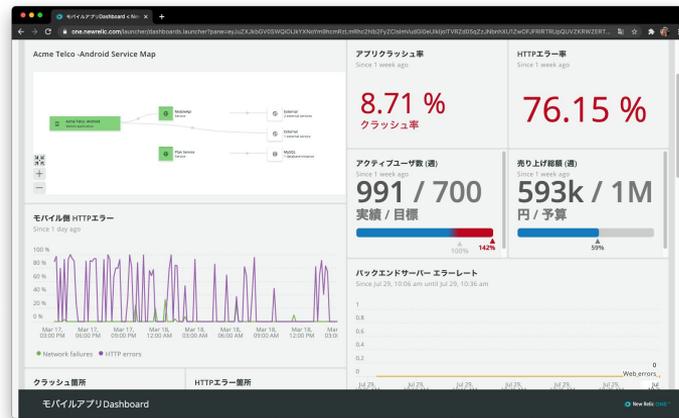
## Note機能

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(2/2)

- 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)

Default

Last 30 mins

Last 60 mins

Last 3 hrs

Last 6 hrs

Last 12 hrs

Last 24 hrs

Last 3 days

Last 7 days

Custom >

Custom Reset to now

Start date Start time - Jun 18, 2020 02:34 pm

Asia/Tokyo Cancel Apply

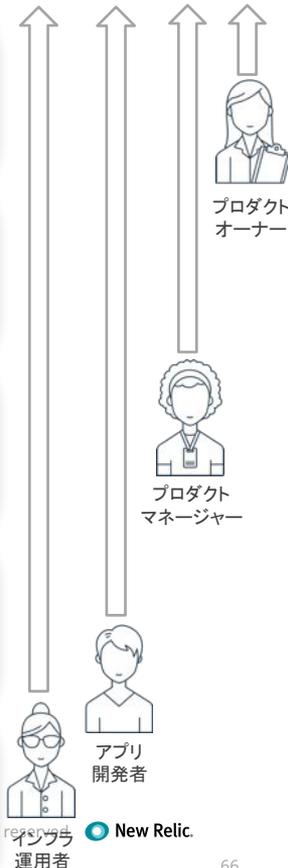
Recent

You don't have any history yet :)

# New Relic のObservability 成熟モデル

成熟度	特徴	KPI
4 データドリブン	<ul style="list-style-type: none"> <li>データを基にした経営判断</li> <li>マーケットへの素早い対応</li> <li>CSAT/Net Promoterスコアの改善</li> </ul>	<ul style="list-style-type: none"> <li>顧客満足度</li> <li>経営計画や経営判断と紐付いたSLOの活用</li> <li>Time-To-Market (# of new products/features)</li> </ul>
3 予測的	<ul style="list-style-type: none"> <li>オブザーバビリティを用いた運用</li> <li>顧客体験の改善</li> <li>丁度良いスケーリング</li> </ul>	<ul style="list-style-type: none"> <li>% of 重大な障害の削減</li> <li>% of エラーバジェットの利用数</li> <li>% of SLOアラートの数</li> <li>% of ビジネス指標のトラッキング数</li> </ul>
2 積極的	<ul style="list-style-type: none"> <li>MTTDの改善</li> <li>サービスレベル改善のトラッキング</li> <li>デジタル顧客体験の改善施策の実施</li> </ul>	<ul style="list-style-type: none"> <li>平均MTTD</li> <li>% of SLOが定義されたサービス数</li> <li>% of クリティカル・ケイパビリティの策定(DCX)</li> </ul>
1 受動的	<ul style="list-style-type: none"> <li>MTTRの改善</li> <li>アプリケーションパフォーマンスの把握</li> <li>顧客に影響のある事象への対応の改善</li> </ul>	<ul style="list-style-type: none"> <li>平均MTTR</li> <li>% of パフォーマンス低下事象の発生</li> <li>% of 可用性低下事象の発生</li> </ul>
0 Getting Started	<ul style="list-style-type: none"> <li>パフォーマンスデータの収集</li> <li>計測の展開計画の策定</li> <li>サービスレベルの検討</li> </ul>	<ul style="list-style-type: none"> <li>% of New Relicの導入率</li> <li>テレメトリーデータの量</li> </ul>

## 関わるロール



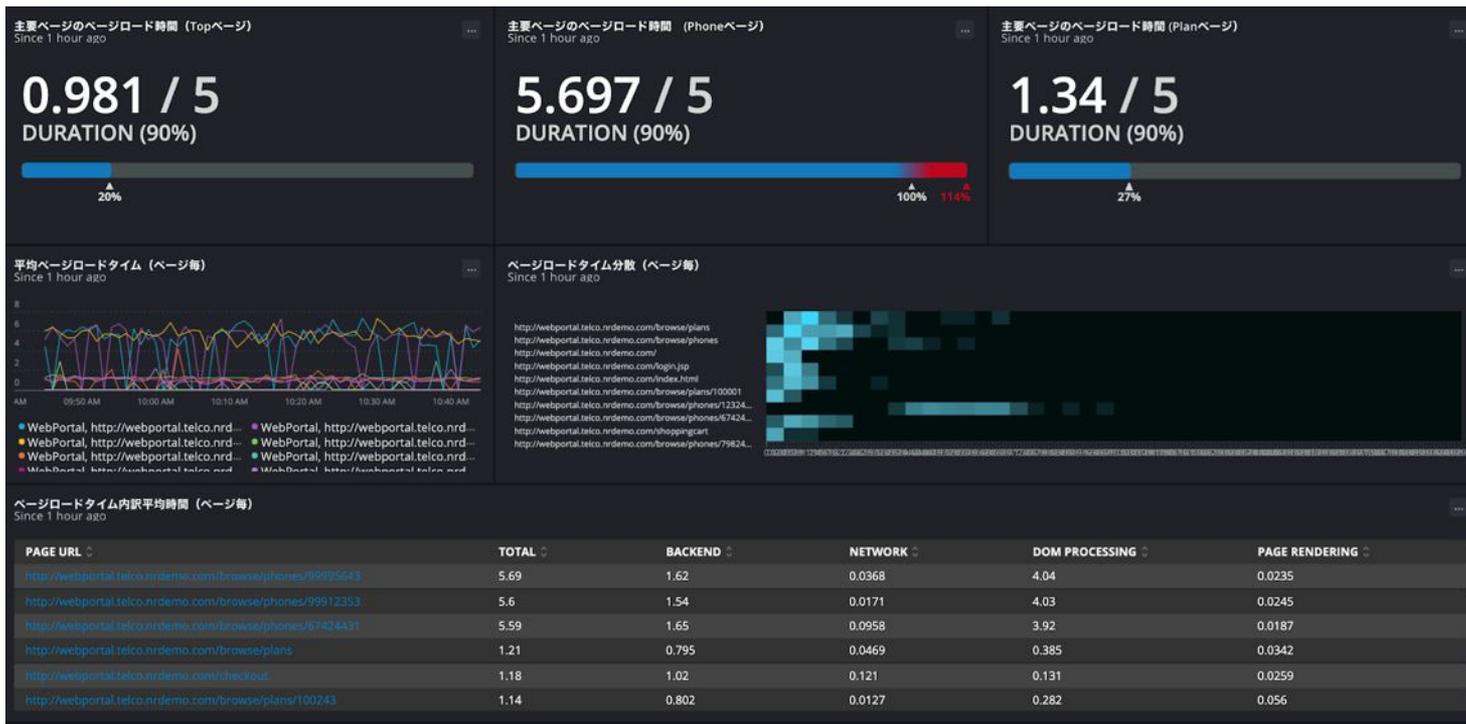
# 目的に応じたダッシュボード

## 例1: サーバー稼働状況



# 目的に応じたダッシュボード

## 例2: ユーザー体験のリアルタイムモニタリング



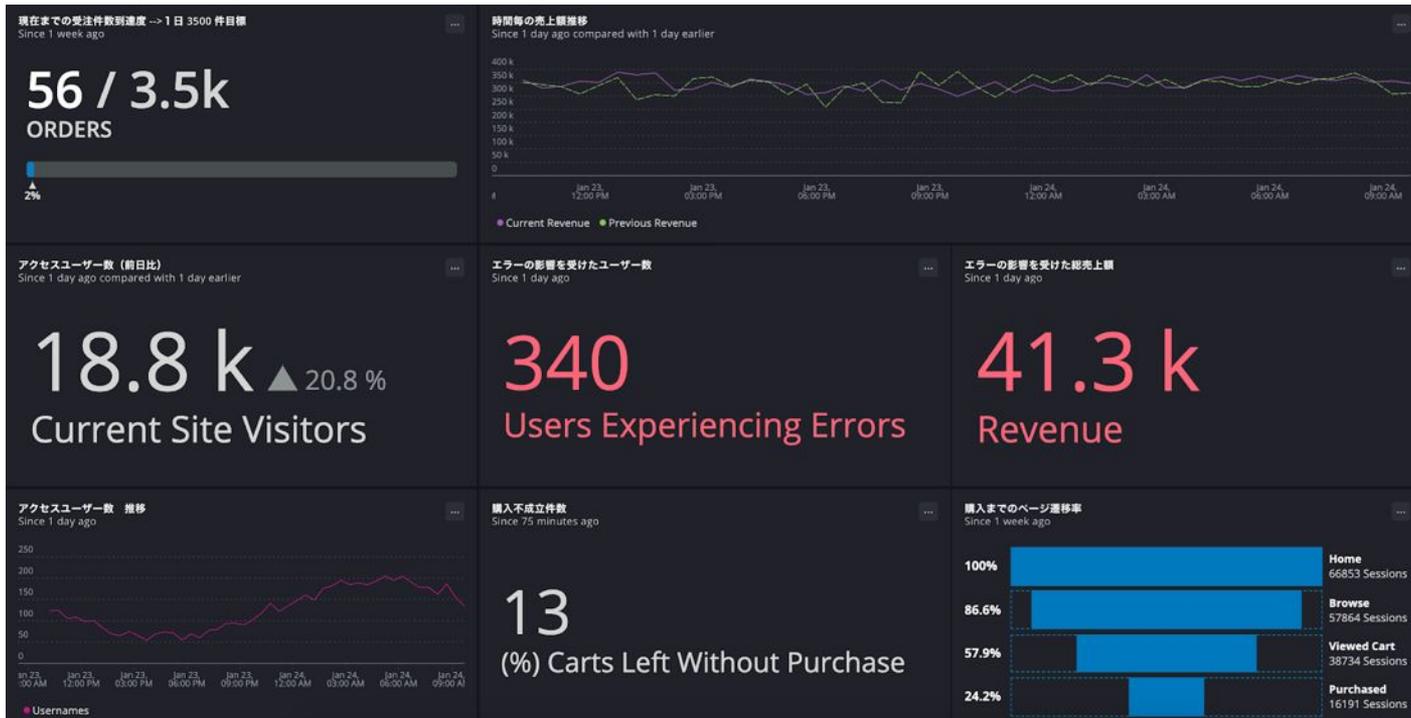
# 目的に応じたダッシュボード

## 例3: ECサイト購買分析



# 目的に応じたダッシュボード

## 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することもできます
- 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

```
1. Get your API key
Select or create an API key

2. Install the NR1 CLI
Mac Linux Windows
Download installer

3. Run this command to ensure everything is up and running
nr1 --version

4. Save your credentials
nr1 profiles:add --name [account-slug] --api-key [api-key] --region [us|eu]

5. Create your package
nr1 create --type nerdpack --name my-awesome-nerdpack

6. Start developing
cd my-awesome-nerdpack && nr1 nerdpack:serve
```

オリジナルのダッシュボードを作成可能



# ハンズオン(3) ダッシュボードの作成

このハンズオンのセクションでは、以下の点を学習します。

- S3ダッシュボードのチャートを利用する
- S3のチャートをカスタマイズする
- NOTE機能を利用する
- チャートのサイズや配置を変更する
- **Option!!** NRQLを用いてチャートを作成する



16:15 - 16:35 (20min)

# ハンズオン(3) ダッシュボード の作成

目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

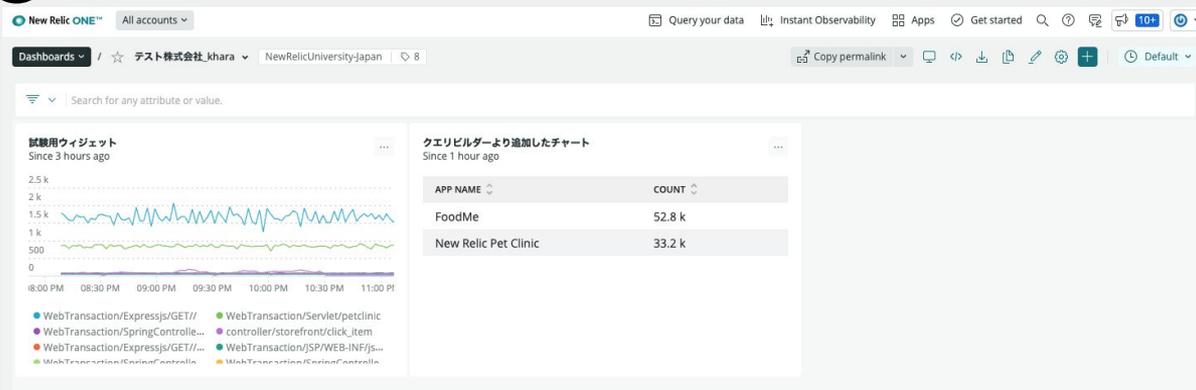
S3ダッシュボードを用いて、ダッシュボードをカスタマイズする

- ハンズオン(2)で作成したダッシュボードを開く
- 別のタブで、S3ダッシュボードを開く

ハンズオン

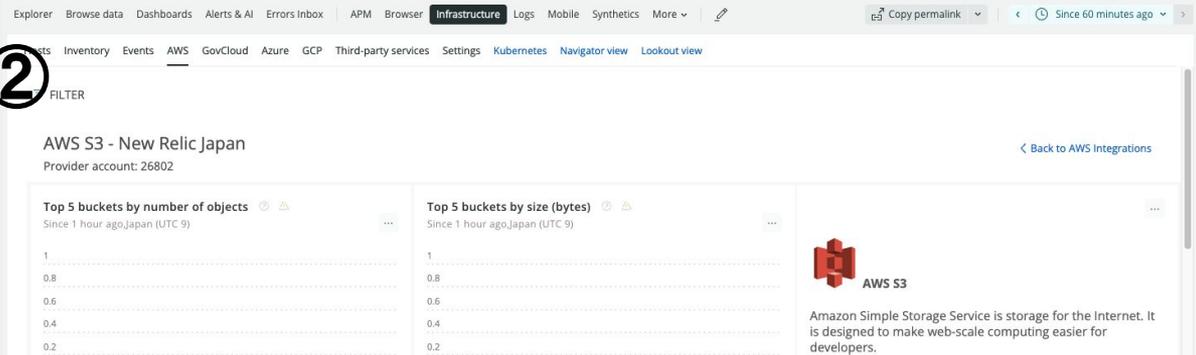
準備

①



## Infrastructure → AWS → S3 dashboard

②



# ハンズオン(3) ダッシュボード の作成

目的:

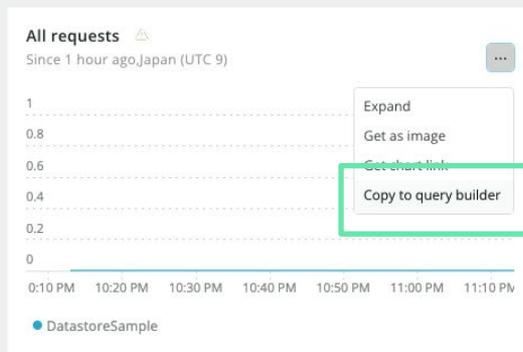
- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

S3ダッシュボードを用いて、ダッシュボードをカスタマイズする

- S3ダッシュボードからAll requestsというチャートをクエリビルダーにコピーする
- クエリビルダーが開いたら、右下のAdd to Dashboardボタンをクリックする
- 任意名で、ダッシュボードに保存する

①



②



クエリビルダー上での変更は何も必要ありません

③



# ハンズオン(3) ダッシュボード の作成

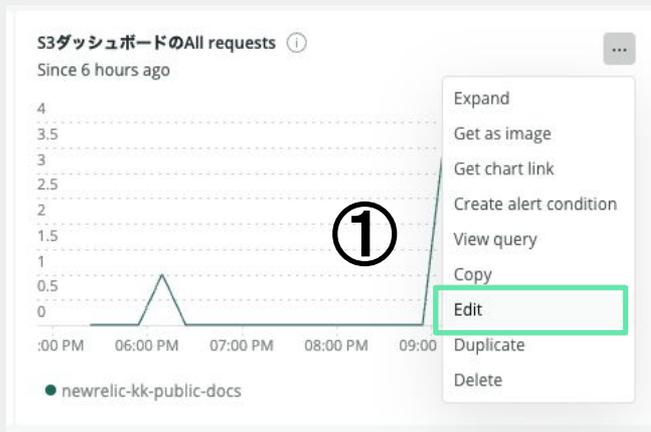
目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

S3ダッシュボードを用いて、ダッシュボードをカスタマイズする

- 追加したチャートでEditを選択し、直接クエリビルダーを起動する
- データ格納元のEvent名を確認する



②

Account: 2511671 - NewRelicUniversity-Japan

PromQL-style Basic

```
SELECT sum(`aws.s3.AllRequests`) FROM Metric WHERE newrelic.cloudIntegrations.providerAccountId = '26802' SINCE 6 hours ago TIMESERIES auto facet entity.name
```

Add another query Your recent queries Create alert Clear Run

# ハンズオン(3) ダッシュボード の作成

目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

S3ダッシュボードを用いて、ダッシュボードをカスタマイズする

- Requestの対象をallからgetに絞る  
(属性名:aws.s3.GetRequests)
- データ参照時間を1日に変更する
- Save as...ボタンを押し、ダッシュボードに保存します

Account: 2511671 - NewRelicUniversity-Japan

PromQL-style Basic

```
SELECT sum(`aws.s3.AllRequests`) FROM Metric WHERE newrelic.cloudIntegrations.providerAccountId = '26802' SINCE 6 hours ago TIMESERIES auto facet entity.name
```

Add another query Your recent queries Create alert

Clear Run



① SELECT sum(`aws.s3.GetRequests`)



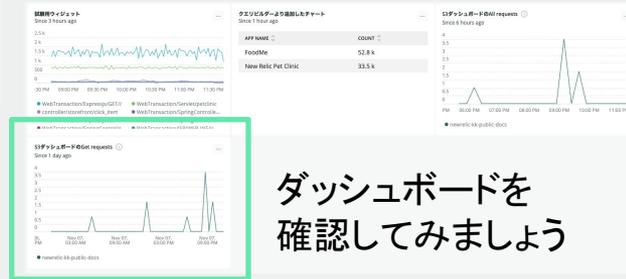
② since 1 day ago



③ Cancel Save as... Save

Run

Run



ダッシュボードを  
確認してみましょう

# ハンズオン(3) ダッシュボード の作成

目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

作成したチャートの表示スタイルを変更する

- [...]からEditを選択し、クエリビルダーを起動する
- Chart typeのプルダウンメニューからPieを選択する
- ダッシュボード上での表示が変わったことを確認する

[次のスライドに続きます]

クエリビルダーより追加したチャート  
Since 1 hour ago

APP NAME	COL
FoodMe	52.
New Relic Pet Clinic	33.

①

Chart name: クエリビルダーより追加したチャート

Chart type: Table

Use as filter: JSON, Line, Area, Stacked bar, Bar, Bullet, Billboard

Other group: Table, Pie, Histogram, Heatmap, Funnel

Data Form: [ ]

②

Useful for showing percentages and proportional data.

③

TRANSACTIONS (85.9 K)

	Count	Percentage
FoodMe	52.5 k	61.08 %
New Relic Pet Clinic	33.4 k	38.92 %

95.19 thousand events inspected in 124 ms ( 0.8 MEPS )

# ハンズオン(3) ダッシュボード の作成

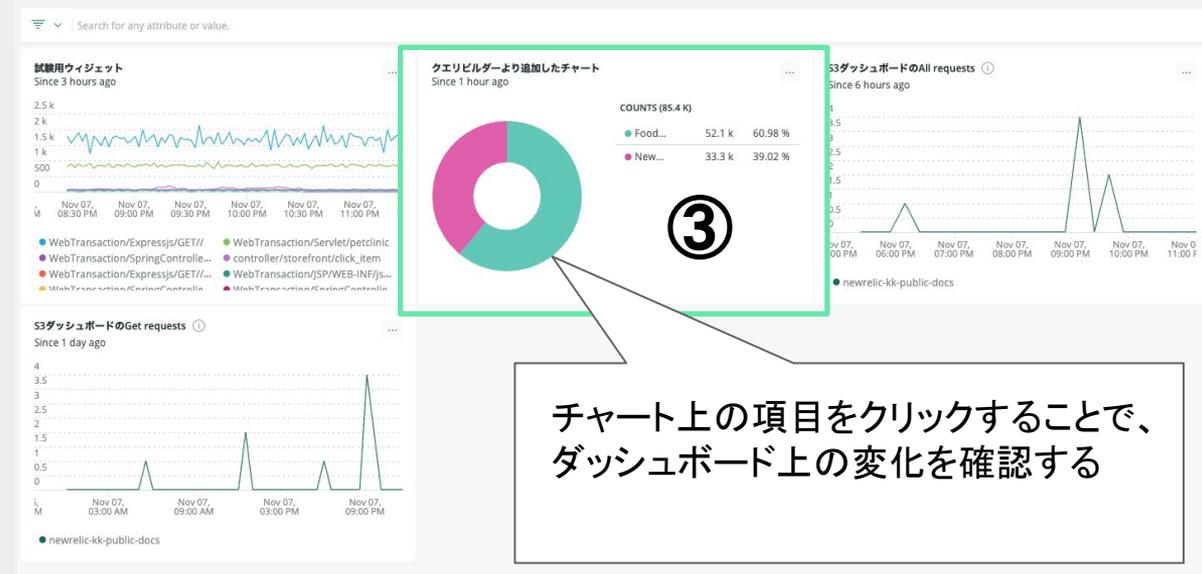
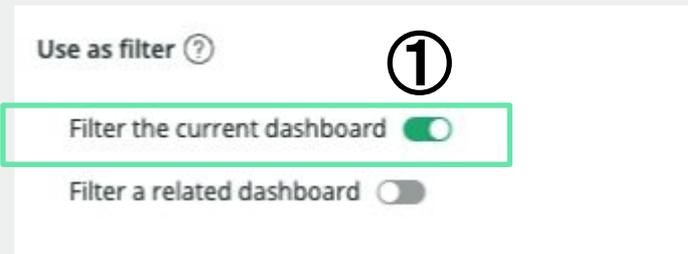
目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

作成したチャートの表示スタイルを変更する

- クエリビルダー上の Filter the current dashboard のトグルスイッチを右に入れる
- Save を押し、変更を保存する
- ダッシュボード上の変化を確認する



# ハンズオン(3) ダッシュボード の作成

目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

ダッシュボードにテキストや画像など  
チャート以外の情報を追加する

- ダッシュボードのEditボタンをクリックする
- Add widgetをクリックする
- Add text, Images, or linksをクリックする
- 編集画面が右からスライドして表示されることを確認する

[次のスライドに続きます]

①



②



Add to your dashboard

Add a chart

Use the query builder to see what's happening with your data.

Add text, images, or links

Add your own content using Markdown.

③

Add text, images, or links

Markdown

```

1 Title
2 ---
3 # Heading 1
4 ## Heading 2
5 ### Heading 3
6
7 • You can create bullets
8 • [Add links](https://www.newrelic.com)
9 • ![Add Images](https://newrelic.com/assets/newrelic/logo_nr-tw.png)
10
11 Text can be formatted as bold or italic
12
13 Code snippets can be defined as this

```

Preview

Title

Heading 1

Heading 2

HEADING 3

- You can create bullets
- Add links

New Relic

Text can be formatted as bold or italic

Code snippets can be defined as `this`

④

任意のテキストを入力し、保存します。

# ハンズオン(3) ダッシュボード の作成

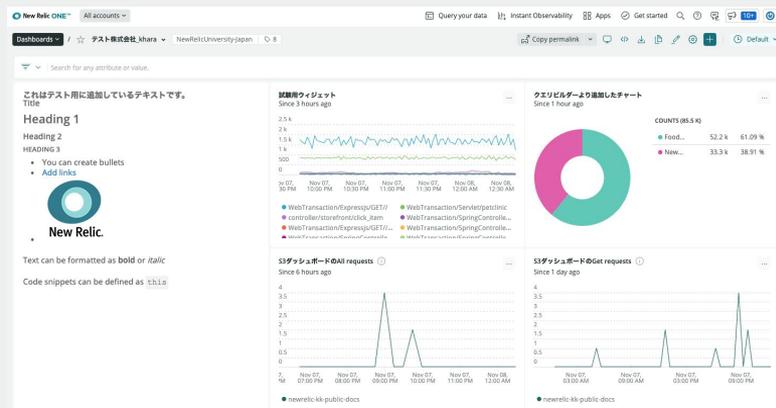
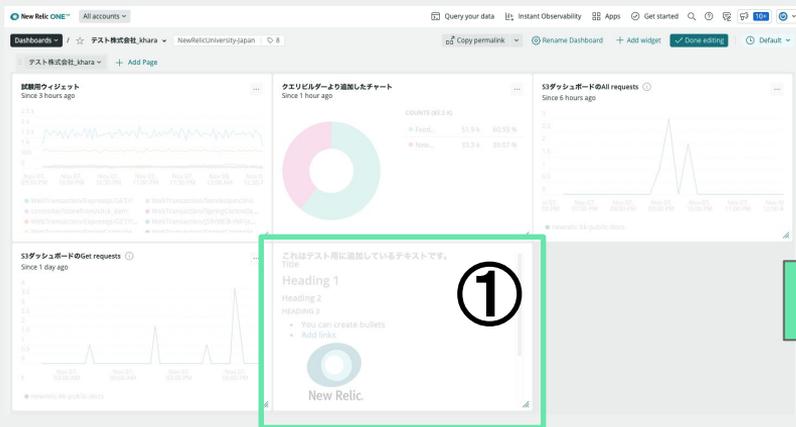
目的:

- ダッシュボードを作成する

既存のUIを用いてダッシュボードをカスタマイズする

ダッシュボード内の配置やサイズを変更します

- 追加したテキストを確認する
- チャートをドラッグすることで、配置を変更する
- 同様に、各チャートの右下をドラッグしてサイズを変更する
- Done editingを押し、変更を保存する



# ハンズオン(3) ダッシュボード の作成

目的:

- ダッシュボードを作成する

ダッシュボードにチャートを追加する

チャートの追加方法を確認する

- ダッシュボードのEditボタンをクリックする
- Add widgetをクリックする
- Add a chartをクリックする
- 右側からNRQLを指定するUIがスライドして表示されることを確認する

以降のスライドにて、この手順を用いて  
チャートの追加を行います

①

②

③

④

Add to your dashboard

- Add a chart  
Use the query builder to see what's happening with your data.
- Add text, images, or links  
Add your own content using Markdown.

NRQLをクリックし、詳細な解析を行える様にしましょう

# ハンズオン(3) ダッシュボード の作成

## 目的:

- 目的に沿ったダッシュボードを作成する

過去1週間の稼働率を表示し、前週と比較する

- アカウント: New Relic TSS
- FROM句: SyntheticCheck
- SELECT句: percentage()

## クエリ例:

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

## 解説:

- 外形監視Syntheticsのチェック結果であるSyntheticCheck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ています。
- 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、わかりやすい表示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。

# ハンズオン(3) ダッシュボード の作成

## 目的:

- 目的に沿ったダッシュボードを作成する

## URLごとのページ表示時間のヒストグラムを作成する

- アカウント: New Relic TSS
- FROM句: PageView
- SELECT句: histogram()

## クエリ例:

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

## 解説:

- Browserエージェントによって計測された PageView イベントから、応答時間を抽出しています。
- 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、それぞれにウィンドウに当てはまる応答時間をカウントしています。
- FACET句を使い、pageUrl毎にヒストグラムを作成しています。

# ハンズオン(3) ダッシュボード の作成

## 目的:

- 目的に沿ったダッシュボードを作成する

ユーザーのページ遷移の状況を解析し、ファネルで表示する

- アカウント: New Relic TSS
- FROM句: PageView
- SELECT句:funnel()

## クエリ例:

```
SELECT funnel(session, WHERE pageUrl LIKE '%home_page' AS 'ホームページ', WHERE pageUrl LIKE '%pay' AS '支払いページ', WHERE pageUrl LIKE '%check_out' AS 'チェックアウト') FROM PageView SINCE 1 day ago
```

## 解説:

- 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 を抽出しています。
- funnel()関数を用いて、pageUrlの値を条件にして、条件に当てはまる session 数をカウントしています。

# ハンズオン(3) ダッシュボード の作成

## 目的:

- 目的に沿ったダッシュボードを作成する

ハンズオンで実施したカスタムイベントの一覧を作成する

- アカウント:

**注意!!** NewRelicUniversity-japan

- FROM句: NRULab
- SELECT句:initial, comment

## クエリ例:

```
SELECT initial, comment FROM NRULab since 1 day ago
```

## 解説:

- ハンズオンで送付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ています。

STEP3-2

アラートの  
作成

# 高度なアラート設定

# 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

1. Categorize APM - Application metric

2. Select entities 1 entity

3. Define thresholds

When target application

Response time (web) has an average above

sec at least once in  minutes

Condition name

TomcatApp

2.5 ms  
2 ms  
1.5 ms  
1 ms  
0.5 ms

Oct 03, 04:00 AM Oct 03, 05:00 AM Oct 03, 06:00 AM Oct 03, 07:00 AM Oct 03, 08:00 AM Oct 03, 09:00 AM

● Critical threshold ● Critical violation

[← Back to Select entities](#)

# 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)

Incident #83551393 opened

Target

SystemSample query

Condition

NRQLテスト

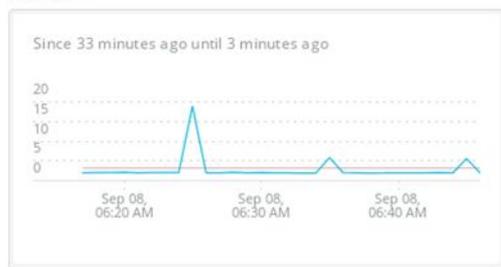
Threshold

SystemSample query result is > 3

(8 kB) ▾

Policy

Tomtest



[View Incident](#)

[NRQL Overview](#)

アラートへのリンク

アラートの詳細をNew Relic上で確認

Plan Service violated Check response time for web portal

3:16 am  
OPENED

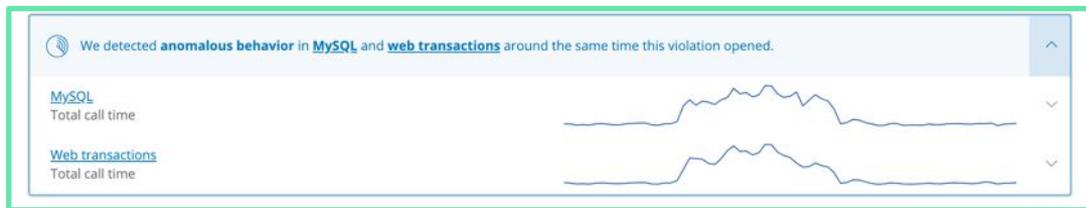
3:31 am  
CLOSED

15m  
DURATION

CRITICAL  
SEVERITY

App metric  
CONDITION TYPE

同時に変化のあったメトリックを表示



Is this helpful? [Yes](#) [No](#)

[View application in service maps](#)

[Go to application overview](#)



Web response time > 600 milliseconds for at least 10 minutes  
THRESHOLD

# アラートのカスタマイズ

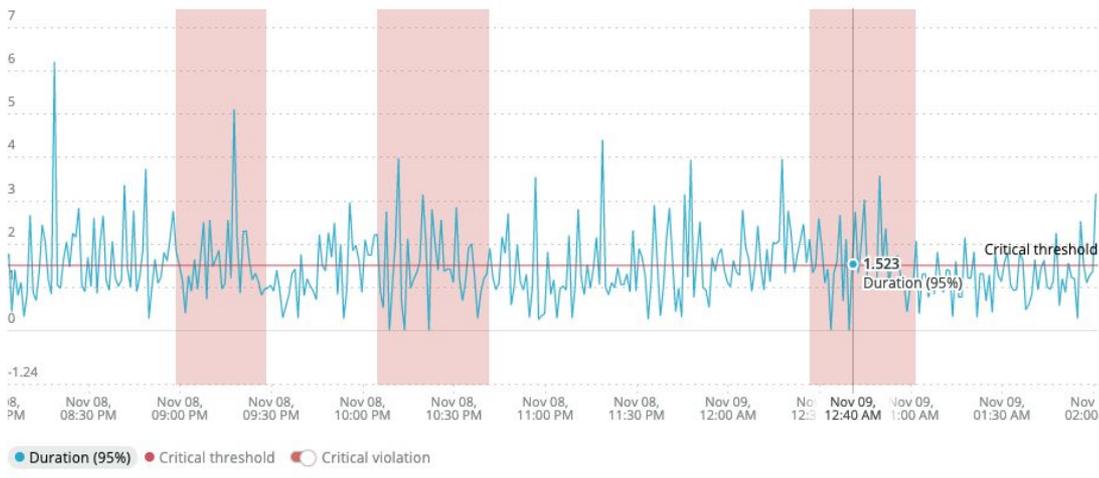
- NRQLの結果をしきい値にアラートを設定することも可能

▼ Define your signal

Enter NRQL Query ⓘ

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

For help with null values [🔗](#), loss of signal [🔗](#), or other query options, [see our docs](#) [🔗](#).



🚨 3 critical violations

Signal loss violations and filled data gaps are currently not reflected in the chart. [See our docs](#) [🔗](#)

- 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、閾値の設定を行えます

# ハンズオン(4) アラートの作成

このハンズオンのセクションでは、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



**16:40 - 16:50 (10min)**

# ハンズオン(4) アラートの作成

## 目的:

- アラート設定のための手順を理解する

## アラート設定を行うUIにアクセスする

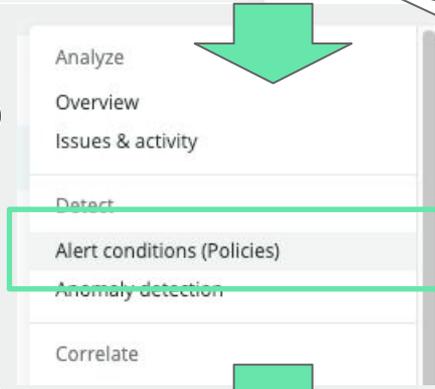
- メニューより、Alerts & AIをクリックする
- 左ペインからAlert condition(Policies)をクリックする
- ダッシュボードハンズオン用アラートポリシーをクリックする

[次のスライドに続きます]

①



②



メニュー上にポインタを移動させるとプルダウンメニューがでます。こちらから直接Policiesに移動することもできます。

③



# ハンズオン(4) アラートの作成

## 目的:

- アラート設定のための手順を理解する

ポリシー内にコンディションを追加します

- 画面右下のAdd a conditionをクリックする
- New condition UI上で、NRQLを選択する
- Next, define thresholdをクリックする

[次のスライドに続きます]

①

ダッシュボードハンズオン用アラート  Connect to Incident Intelligence  Incident preference: By policy  Delete this policy

id: 1065477

1 Alert condition 0 Notification channels [Add a notification channel to receive alerts](#) Last modified Sep 30, '20 2:12 am by Akhiro Ito

Search conditions [Add a condition](#)

NRQL QUERY: NRU302\_alert\_lab Last modified 4:37 pm by NRU-User [Edit](#) [Copy](#) [Delete](#)  On

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

[PageView query result is > 1.0 unit for at least 5 mins](#)

[Add a warning threshold](#)

[Add a condition](#)



⊕ Add a condition

New condition [Cancel](#)

1. Categorize

Select a product

NRQL  APM  Browser  Mobile  Synthetics  Infrastructure

②

[Next, define thresholds](#)

③

2. Enter a NRQL query and thresholds

# ハンズオン(4) アラートの作成

## 目的:

- アラート設定のための手順を理解する

ポリシー内にコンディションを追加します

- Enter condition nameに任意の名前を入力する
- Enter NRQL Queryを入力する
- 画面下部にあるSave conditionボタンをクリックする

Create an alert condition ×

Account: 2511671 - NewRelicUniversity-Japan Policy: 1065477 - ダッシュボードハンズオン用ア...

Enter condition name **①**

Enter a name that's easy to identify

Define your signal

Enter NRQL Query <sup>①</sup>

**②** SELECT

For help with [null values](#), [loss of signal](#), or other query options, see our [docs](#).

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



コンディションを設定するためのさまざまな項目についてご興味のある方は、是非、NRU304のウェビナーをご受講下さい。

お疲れさまでした!!

ご質問がございましたら、チャットにご記入ください。

# ぜひNew Relicをお試ください！

## 弊社サイトからサインアップ

<https://newrelic.com/jp/sign-up-japan>

※このフォームは仮登録フォームです。入力後に表示されるURLまたは短くEメールより、本登録をお願いします。また、仮登録と本登録は同じメールアドレスを入力ください。

無料アカウントでは以下の内容が含まれています：

無料ユーザーアカウント

月間100GBまでのデータ容量。全機能にアクセスできるフルアクセスユーザー1名 / ベーシックユーザーは無制限

メトリクス、ログ、イベント、トレースのすべてデータを統合可能なプラットフォーム

ペタバイト規模のデータを高速で処理。月間100GBを超えてからも1GBあたり0.25ドルでデータ取り込みが可能

ソフトウェアスタック全体を可視化、分析し、迅速なトラブルシューティングを実現

APM、インフラストラクチャ監視、ログ監視、デジタルエクスペリエンス監視、AIOPsなど様々な機能を通じて完璧なソフトウェア構築をサポート

### サインアップ

姓：\*

山田

名：\*

太郎

会社名：\*

New Relic, Inc.

従業員数：\*

選択してください。

メールアドレス（勤務先）：\*

有効なメールアドレスを指定し

携帯/電話番号：\*

03-0000-0000 / 090-0000-0000

\*日中にご連絡可能な電話・携帯電話番号を入力下さい。

国：\*

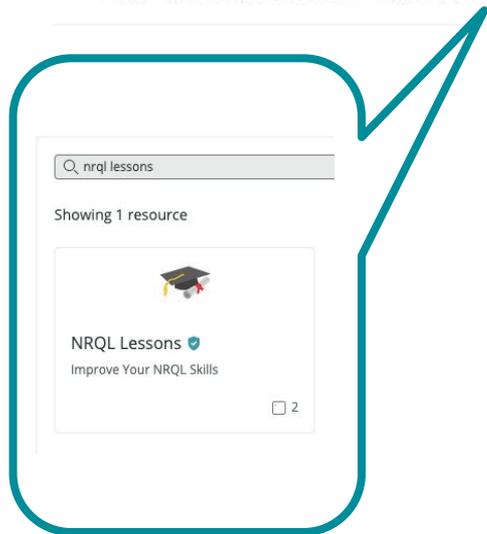
日本

New Relic 製品、サービス、イベント、キャンペーンに関連する情報を New Relic および関連会社から受け取ることに同意します。この登録の解除はいつでも可能です。お客様の情報は、[一般データプライバシー](#)に従って処理されます。

サインアップ

# NRQL Lessons - NRQL学習ツール

Query your data Instant Observability Apps Get started ?



メニューから、I/Oを選択し、  
"NRQL Lessons"で検索する

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

## 初めてのクエリ

まず、New Relic APMで収集される、**Transaction**イベントすべての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は可能です。

レベル1: コツを掴む

1. ようこそ
2. **初めてのクエリ**
3. データの集約
4. 時間範囲
5. 時系列クエリ
6. WHERE句
7. ファセットクエリ
8. まとめ

NRQL

```
SELECT * FROM Transaction
```

Result

TIMESTAMP	APDEX PERF ZONE
November 15, 2021 16:54:56	5

# New Relic University

トレーニング、認定資格、書籍など知識へのアクセス

## 簡単にはじめる

### Guided Install | [In-Product](#)

New Relic One インストール方法を製品内部でわかりやすくガイド

### Install Guide | [Document & Video](#)

New Relic One の各機能のインストール手順をドキュメントと動画でガイド

- Sign-up 方法と初期設定ガイド
- 7言語ごとのAPM計装ガイド
- OSごとのInfrastructure計装ガイド
- OSごとのMobile計装ガイド
- Browser計装ガイド
- Synthetics計装ガイド

### Japanese Document | [Document](#)

New Relic One の全機能利用解説ドキュメントは日本語

-docs.newrelic.com

### New Relic Book | [Paper Book](#)

物理的な書籍としてもNew Relic One 活用方法ガイドを提供

-New Relic 実践入門 (翔栄社)

## 無償トレーニング

### NRU Practitioner | [Fundamentals](#)

オブザーバビリティの基礎知識を学習

-オブザーバビリティ 入門

### NRU 10X | [Fundamentals](#)

New Relic One の製品概要と基礎知識を学習

-New Relic One 入門

### NRU 20X | [Product Basics](#) (11月公開予定)

New Relic One の各機能を学習

- Full Stack Observability 入門
- Applied Intelligence 入門
- Telemetry Data Platform 入門

### NRU 30X | [Hands-On](#)

SLI/SLO 設計やダッシュボード開発、AIOps の最適なアラート設計などを学習

- コア機能実践ハンズオン
- SLI/SLO 設計ハンズオン
- AIOps とアラート設計ハンズオン
- ダッシュボード開発とNRQL実践ハンズオン

### NRU 40X | [Advanced Hands-On](#) (AWS特設ページ内限定開催)

AWS など環境特化での実運用を学習

- New Relic One for Amazon EC2
- New Relic One for Amazon ECS
- New Relic One for Amazon EKS

## 認定試験

### Full Stack Observability | [Exam](#) (英語限定)

Full Stack Observability の基礎知識を試験

### Programmability | [Certification](#) (英語限定)

Programmability スキルを認定

# New Relic 実践入門

希望者に無償提供中

ついに発売された New Relic の全てを理解できる 330 ページにわたる技術書籍。オブザーバビリティの基本から New Relic One の基本機能、さらには 16 のオブザーバビリティ実装パターンまで含めた、初心者から応用を理解したい上級者まで対象にした New Relic のパーフェクトガイドブック。

無償提供希望はこちらのGoogle Form から

<https://forms.gle/jqiYmWRYt8Hf1nHk8>



# New Relic User Group

New Relicを利用するユーザー同士交流するためのグループです。  
5月13日(金) 19:00~にて、SRE支部の第1回イベントが開催予定です！

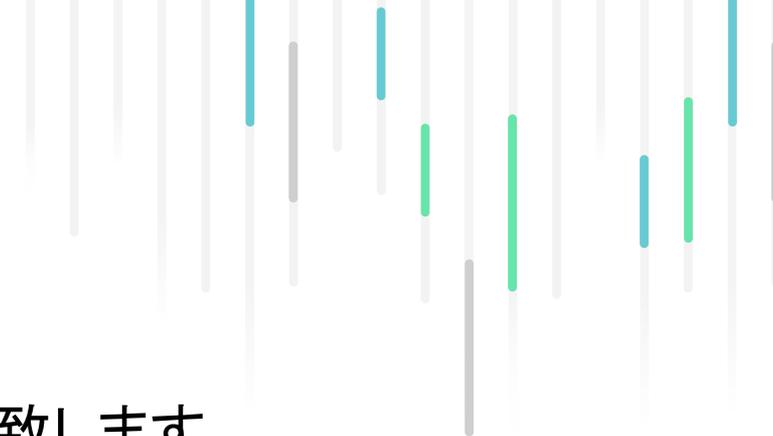
参加申し込み: <https://nrug-sre.comnpass.com/event/242783/>



NRUGのslackもあります！

[https://join.slack.com/t/nrug/shared\\_invite/zt-y0de03za-0vCEQdEoGjKXoOrdrHIRzA](https://join.slack.com/t/nrug/shared_invite/zt-y0de03za-0vCEQdEoGjKXoOrdrHIRzA)





最後となりますが、  
是非、アンケートへのご協力をお願い致します。

また、もっと詳しい話を聞きたい方は、  
その旨アンケートにご記載ください

Thank You