



# ダッシュボード開発と NRQLの基本編

NRU 300 - Dashboard / NRQL



# ウェビナー 各種ご連絡

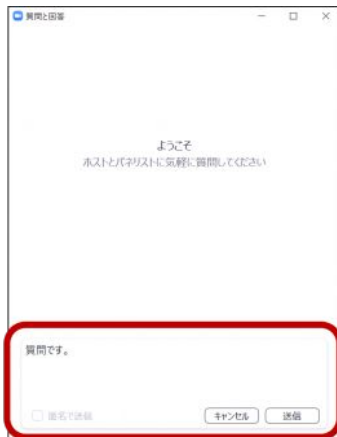
1. ご質問がある場合は、“Q&A”からご入力ください。



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.

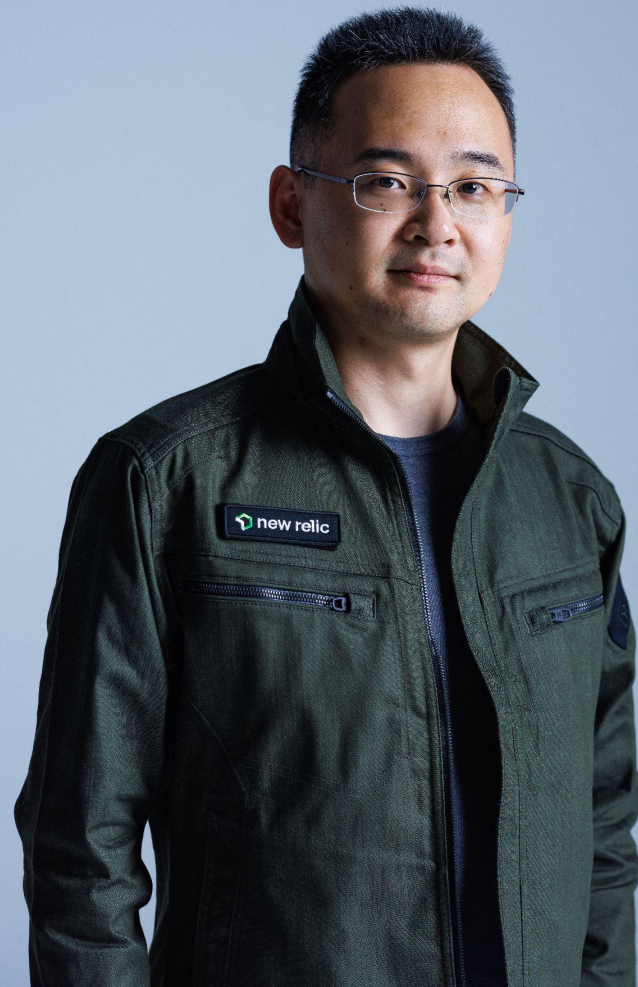
Actual results may differ materially from those expressed in these forward-looking statements, which speak only as of the date hereof, and are subject to change at any time without notice. Existing and prospective investors, customers and other third parties transacting business with New Relic are cautioned 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](http://ir.newrelic.com) or the SEC's website at [www.sec.gov](http://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.

# Satoshi Itatani

## Solutions Consultant

長年インフラエンジニアとして従事。  
オンプレミスおよびクラウドでの大規模 Webインフラの  
設計、構築、運用を専門とする。  
物理、ネットワークからアプリまで  
幅広い開発構築経験もあり、バックエンド全体の知識を有する。  
最近のブームはIoT鉄道模型



# 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)

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

© 2025 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

NRU300 - 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, GCPI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

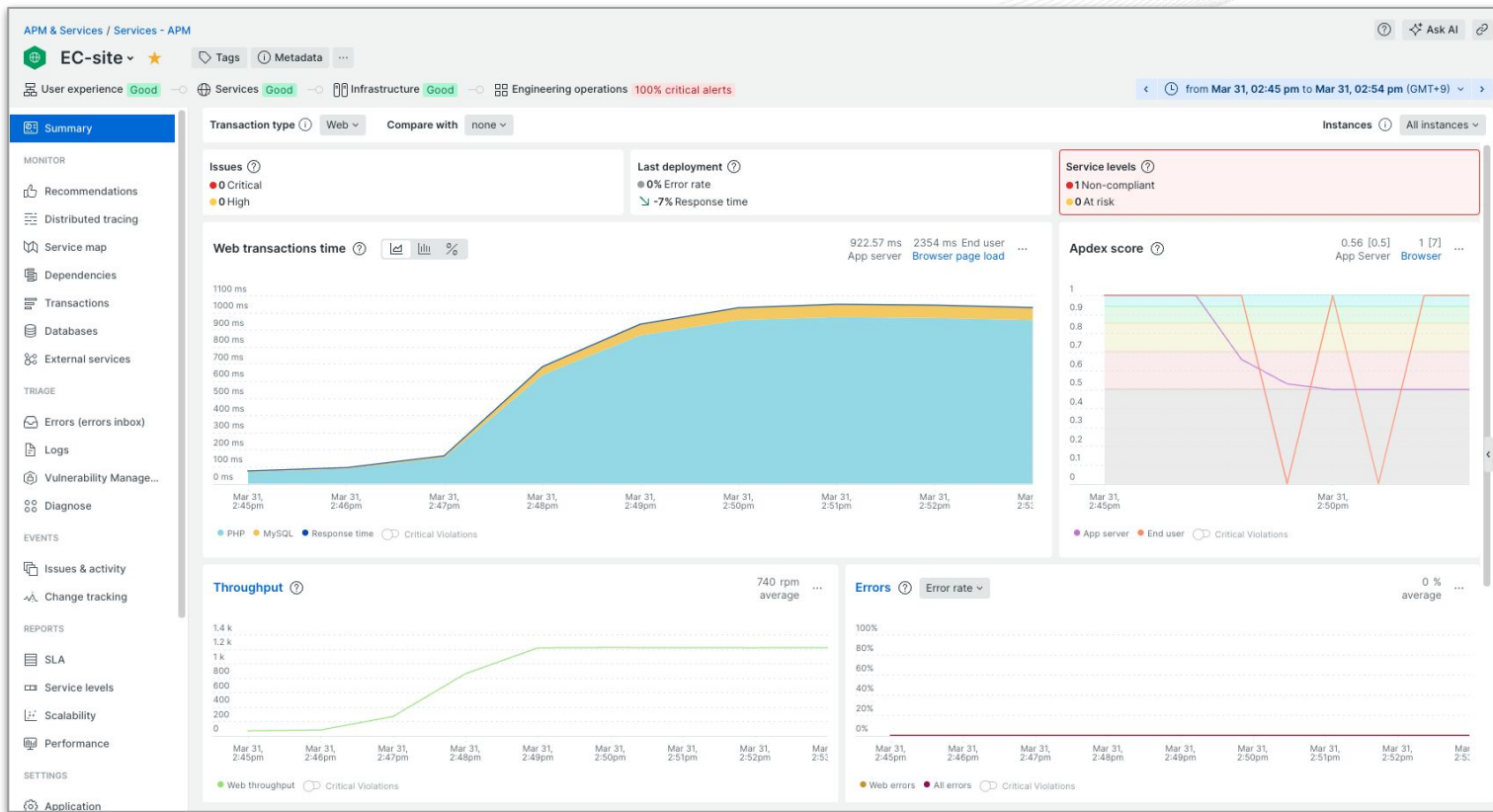


NRDB

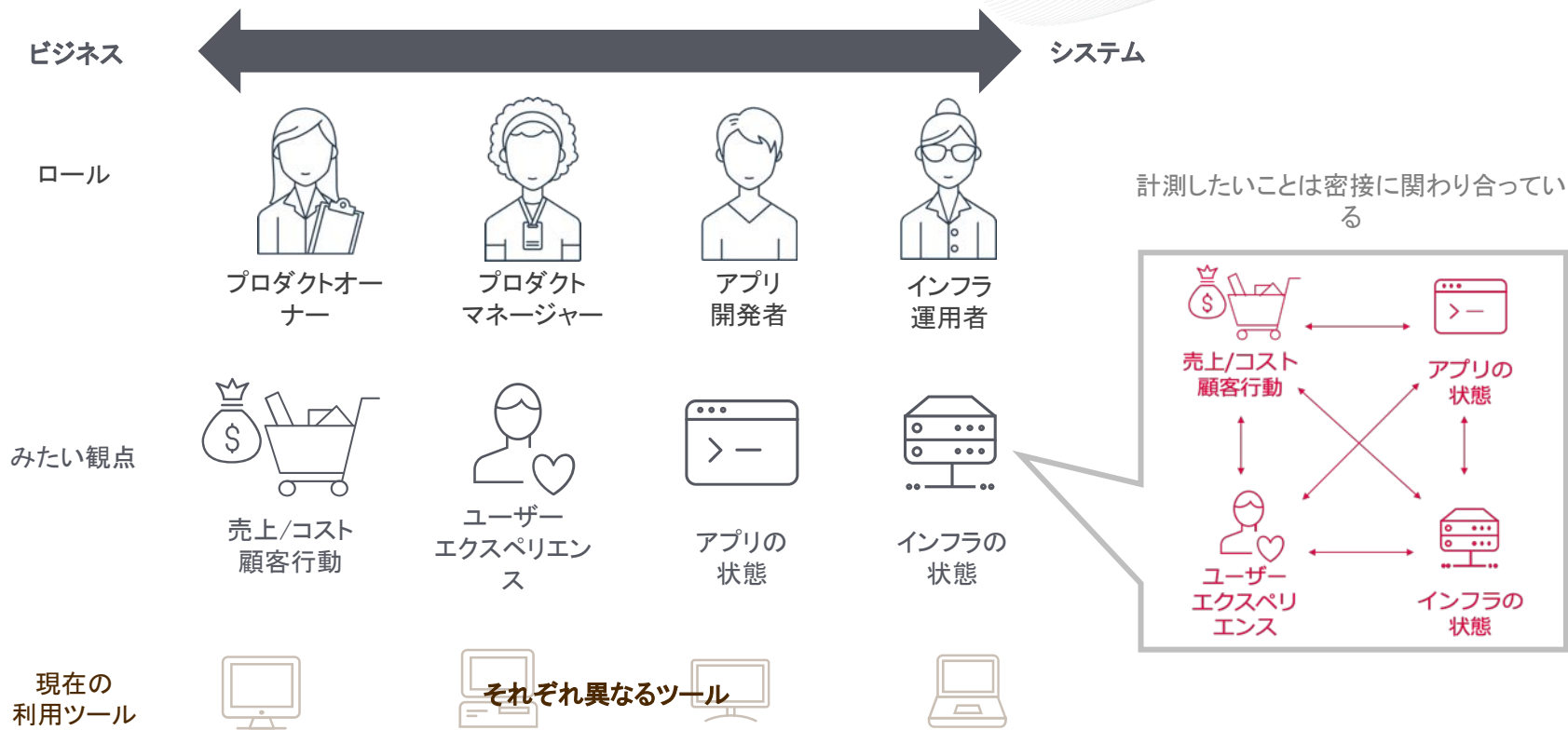
分間、数十億件のデータを取り込み、処理  
超高速検索を可能にするデータベース

分析用データ集計

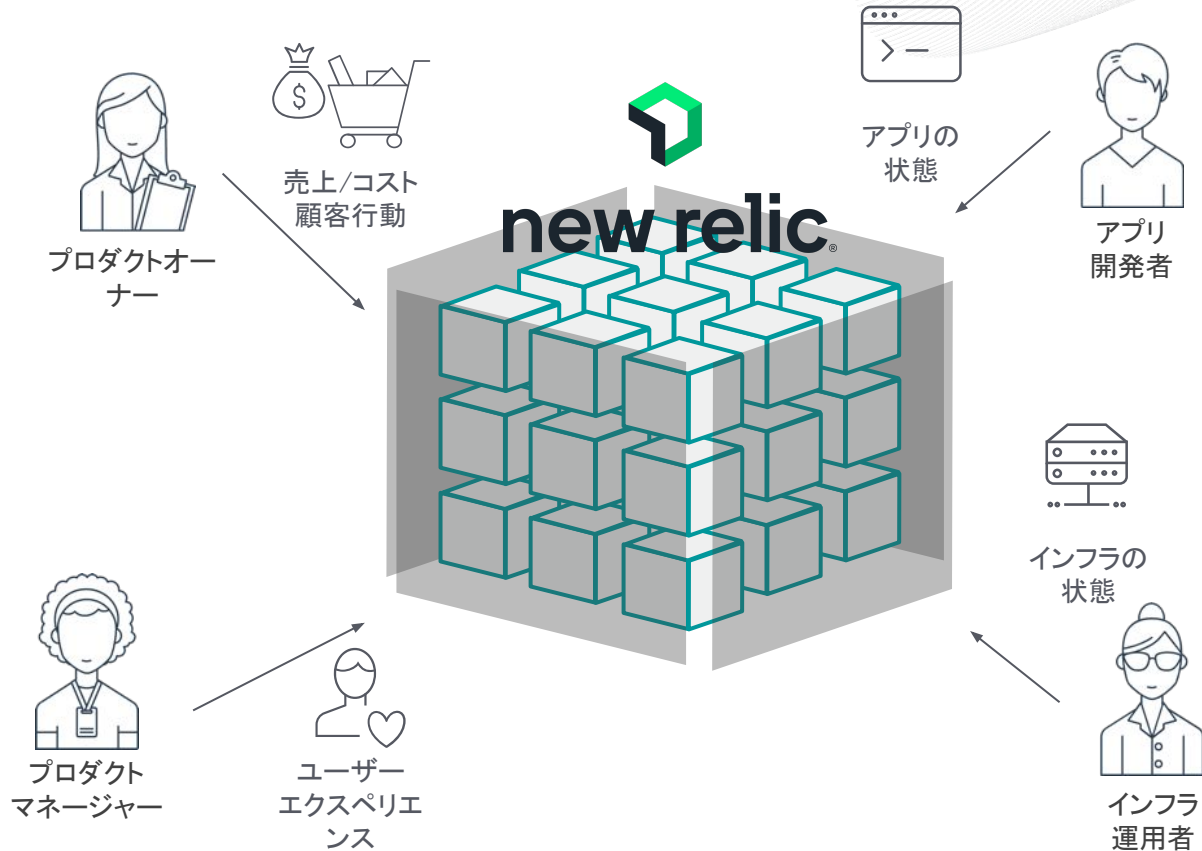
# ビルトインのUIでもデータ分析はできます



# ・・・が、人によってみたい観点は全く異なる



# 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・・・)	<ul style="list-style-type: none"><li>データ一元化率</li><li>対象システム割合</li><li>ログ監視脱却→APM中心</li></ul>
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)	<ul style="list-style-type: none"><li>サービス停止率</li><li>障害発生率</li><li>MTTR削減</li></ul>
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)	<ul style="list-style-type: none"><li>エラー発生率</li><li>レスポンスタイム</li><li>SLI/SLOの策定割合</li></ul>
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)	<ul style="list-style-type: none"><li>適正スケーリングによるコスト削減</li><li>デプロイ高速化</li></ul>
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)	<ul style="list-style-type: none"><li>顧客満足度の改善</li><li>市場投入までの時間(新製品・新機能の数)</li></ul>

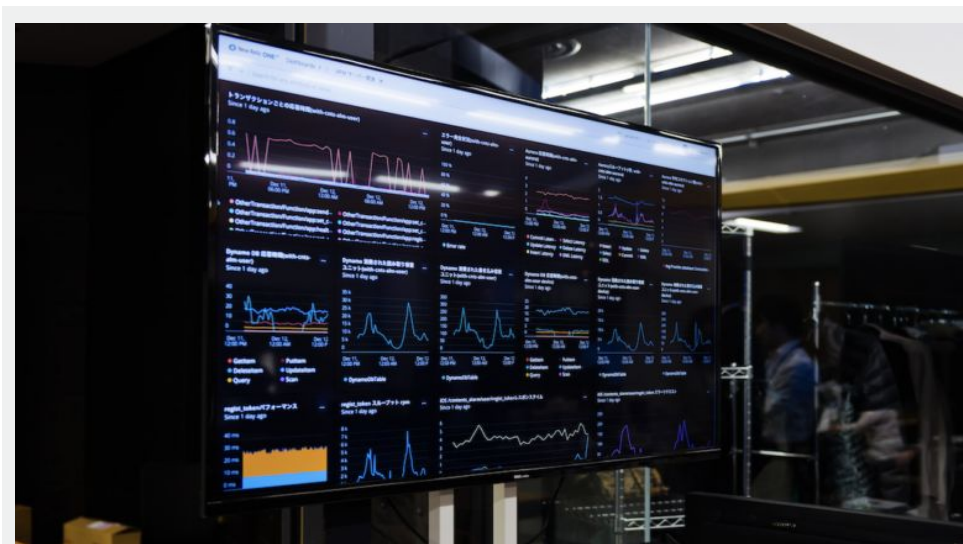
守り

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

NRU300 - 座学

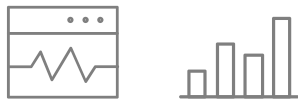
STEP1

データの  
理解

# New Relicが取得するデータ: MELT

## M:メトリック

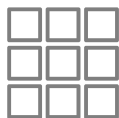
集計可能なデータの粒



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

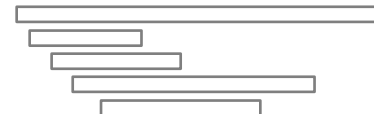
個々の出来事の記録



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.com/jp/docs/data-apis/ingest-apis/event-api/introduction-event-api/>
- Eventデータについて
  - <https://docs.newrelic.com/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": "Mzk0MDcxNnxBUE180VB0TE1DQVRJT058MTA4NDQwMDMwMw",
          "entityGuid": "Mzk0MDcxNnxBUE180VB0TE1DQVRJT058MTA4NDQwMDMwMw",
          "error": false,
          "guid": "60b9b772435240a5",
          "host": "ip-172-31-26-144.ap-northeast-1.compute.internal",
          "http.statusCode": 200,
          "http.responseCode": "200",
          "name": "WebTransaction/Action/block_search_product",
          "priority": 1.82538,
          "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/javascr",
          "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"
```

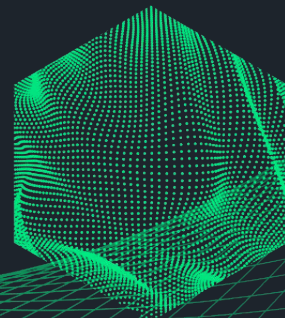
アプリケーション名は"appName"という属性

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

NRU300 - 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**<sup>®</sup>

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

**p. 25 - p. 37**



# New Relic へログイン

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 G E

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+nruc@newrelic.com](mailto:japan-handson+nruc@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、ピー、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

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

Help

Add User

New Relic University Japan

Search by entity name

Entity Type = All

Create a workload

Add data

List

Navigator

Lookout

See which parts of your system are unmonitored.

View 1 gaps

All entities (173)

Last viewed (7)

Services - APM View all (1)

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

Hosts View all (1)

Agent ve...	CPU usa...	Memory ...	Storage ...	Network ...	Network ...	
-26-144.ap-northeast-1.comp...	1.510	6.41%	58.19%	44.35%	1.12 MB/s	27.2 kB/s

ations View all (1)

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

Browser applications (1)

Synthetic monitors (4)

Workloads (1)

Synthetic

Help

Add User

New Relic University Japan

4

Help

Add User

New Relic University Japan



手順

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

# メニューの操作

New Relic ポータルの UI を確認します

The screenshot shows the New Relic 'All Entities' page. The left sidebar menu is highlighted with a green box and a circled number 5. The main content area shows a table of entities categorized by Services - APM, Hosts, and Browser applications.

**Services - APM**

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

**Hosts**

Name ↑	Agent ve...	CPU usa...	Memory ...	Storage ...	Network ...	Network ...
☆ EC-site	1.51.0	6.41%	58.19%	44.35%	1.12 MB/s	27.2 kB/s

**Browser applications**

Name ↑	Through...	Largest c...	Interacti...	Errors	Pageload...	Ajax thro...
☆ EC-site	9.4 page...	463 ms	-	-	866 ms	1.77 rpm



手順

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

# 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**

Alerts & AI

APM & Services

Apps

AWS Lambda Setup

Browser

Dashboards

Errors Inbox

IAST **New**

Infinite Tracing Settings

Infrastructure

Key Transactions

Kubernetes

Logs

Lookout

Metrics & Events

Mobile

Model Performance

Network

Query Your Data

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 'Event type' section in the main panel. In the 'Event type' section, 'Transaction' is selected with a green box and a circled '10'. A green dashed arrow points from 'Transaction' to the 'Transaction' data view. In the 'Transaction' view, the 'JSON' icon in the chart picker is highlighted with a green box and a circled '11'. A green dashed arrow points from this icon to the JSON data displayed in the main panel.

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

```
Transaction
{
  "results": [
    {
      "events": [
        {
          "appId": 1084400303,
          "appName": "EC-site",
          "databaseCallCount": 2,
          "databaseDuration": 0.00023,
          "duration": 0.05816,
          "entityGuid": "Mzk0MDcxNnxBUE18QVBQTElDQVJRJT058MTA4NDQwMDMwMw",
          "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	
April 02, 2024 13:16:26						

⑫

⑬

⑭

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

手順

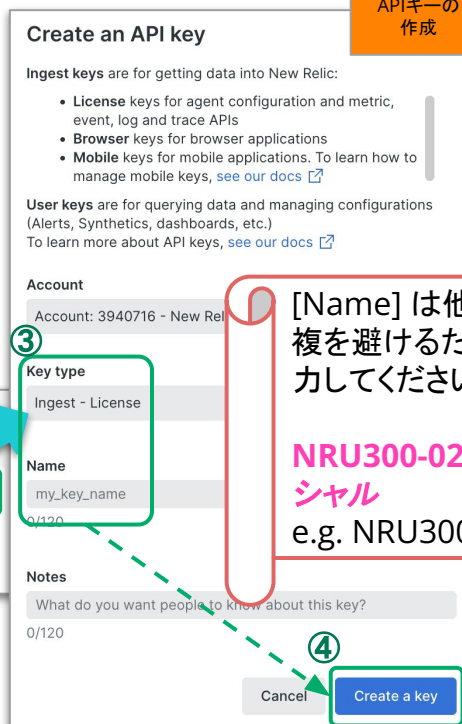
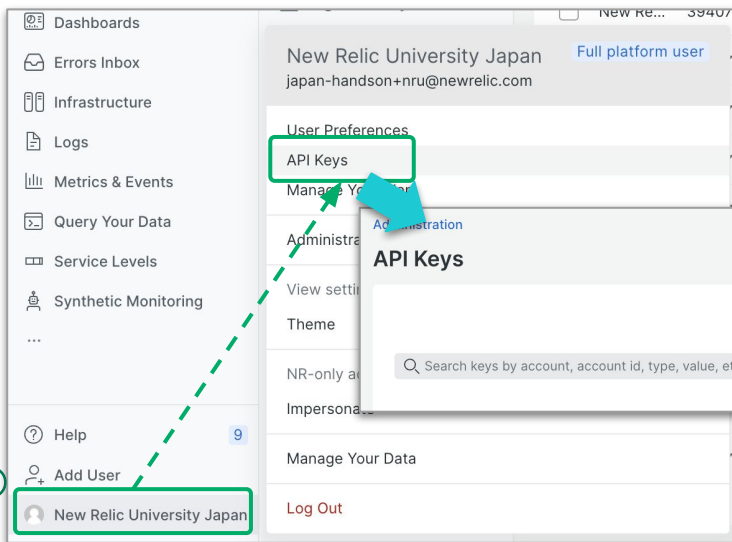
Option!!

# カスタムイベントを送信する

NRU300 - 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



[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

NRU300-0212-企業名-イニシャル  
e.g. NRU300-NEWRELIC-NR

- 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



手順

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

API keys

Create and manage API keys for your organization.

Create a key

Search keys by account, account id, type, value, etc. Only show my user keys 1-35 of 35

Copy your API Key now. You won't be able to see it again! dffbFFFFNRAL Copy Key



手順

⑤ 作成したAPIキーが表示されるので [Copy Key] をクリックし、作成されたキーをクリップボードにコピーします。

参考:

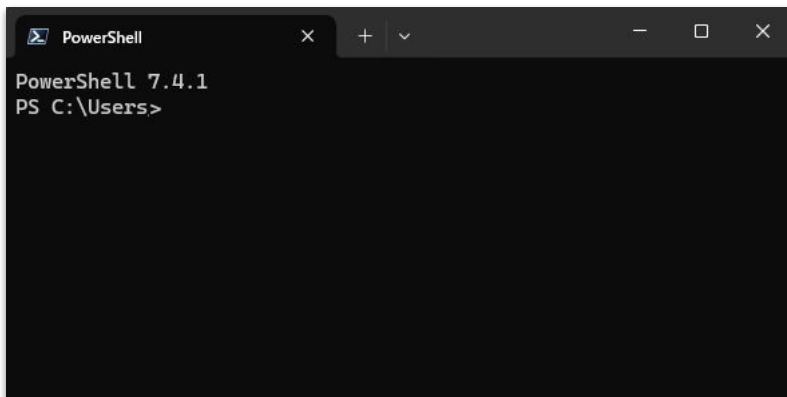
APIキーはOriginalのKeyが作成されている場合がありますが、セキュリティの観点から、新たに作成したKeyを活用することをお勧めいたします。

セキュリティ観点から既存キーのコピーはWebUIから行えなくなりました。

## Option!! ターミナルの起動

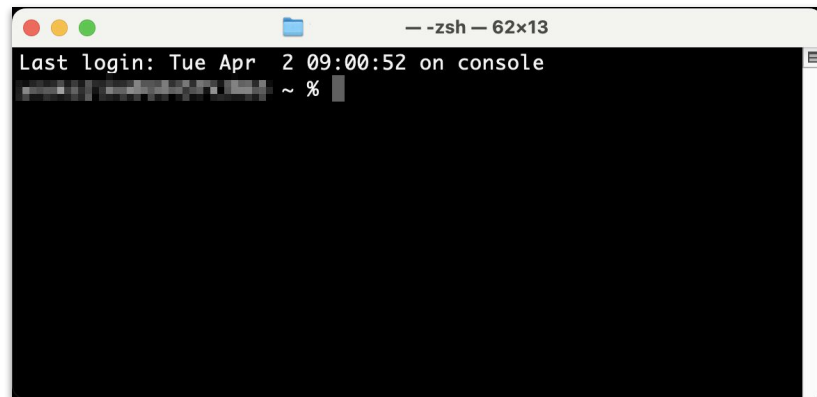
カスタムイベントを送信するため、OSのターミナルアプリを起動します

Windows - PowerShell



```
PowerShell
PowerShell 7.4.1
PS C:\Users>
```

Mac - ターミナル



```
--zsh-- 62x13
Last login: Tue Apr 2 09:00:52 on console
~ %
```



手順

⑥ 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

- ⑦ 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
- ⑧ 「**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

⑨ 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

⑩ 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 'Metrics & Events' page. In the left sidebar, 'Metrics & Events' 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. In the table view, the 'Raw' icon in the top right corner of the table is highlighted with a green box and labeled 13. The table shows one event for 'NRULab' with a timestamp of 'April 02, 2024 18:47:14', comment 'NRU302ハンズオン', initial 'NR', and labnumber '1'. A dashed green line connects the highlighted elements to the numbered steps in the legend below.



手順

- ⑪ メニューから [**Metrics & Events**] をクリックします
- ⑫ スコーピングセクションから [**Events**] をクリックします
- ⑬ Event type から [**NRULab**] を選択します
- ⑭ 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

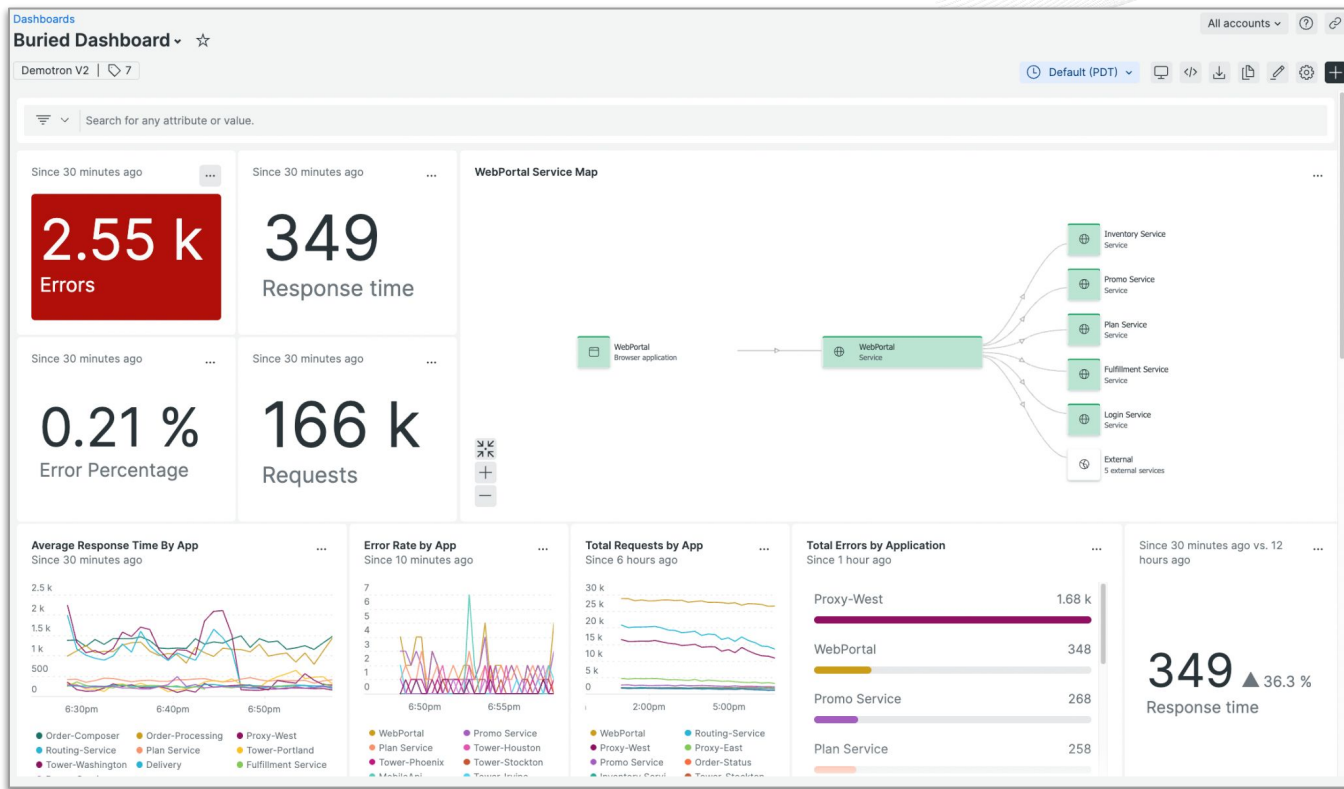
# New Relic ダッシュボード

NRU300 - 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

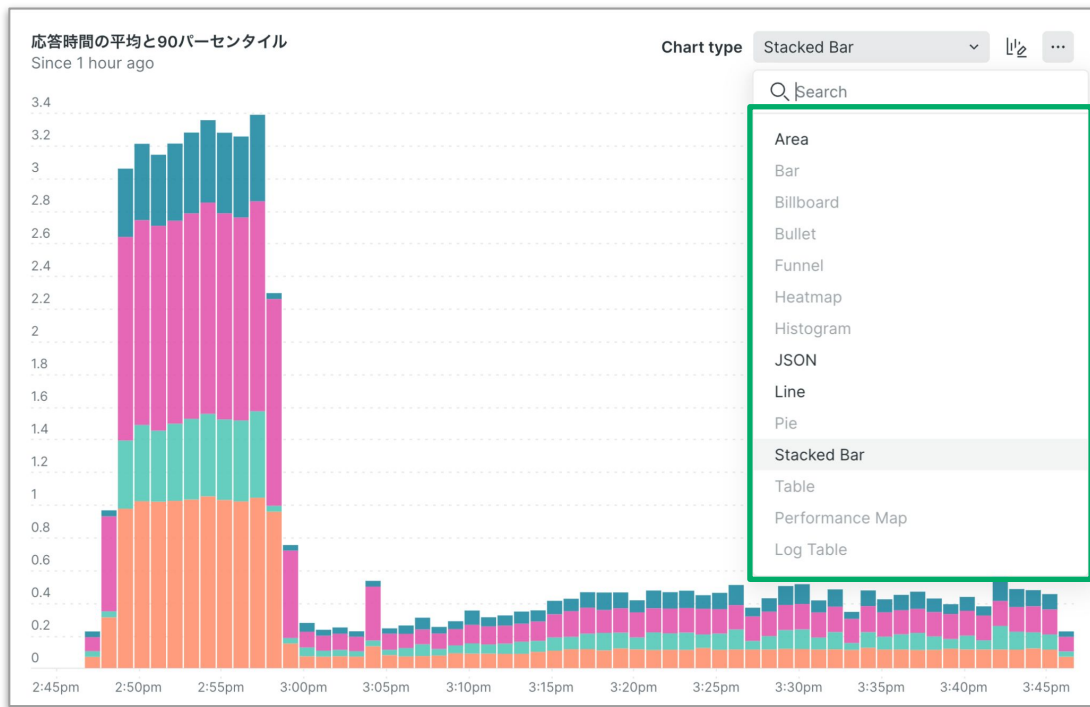


# ダッシュボードが得意とすること

- 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

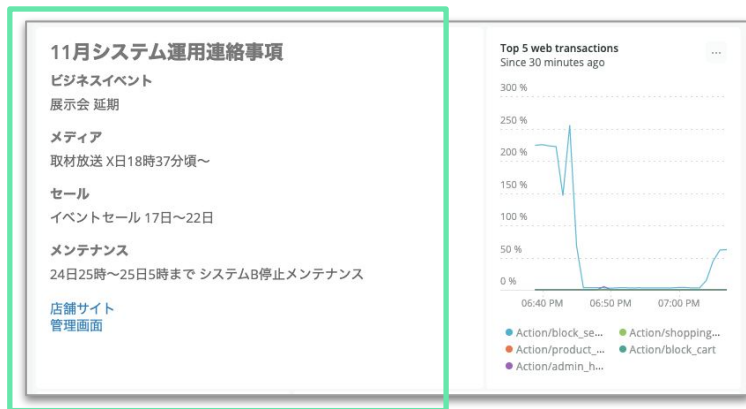


# ダッシュボードの便利機能 ①

- 表示系

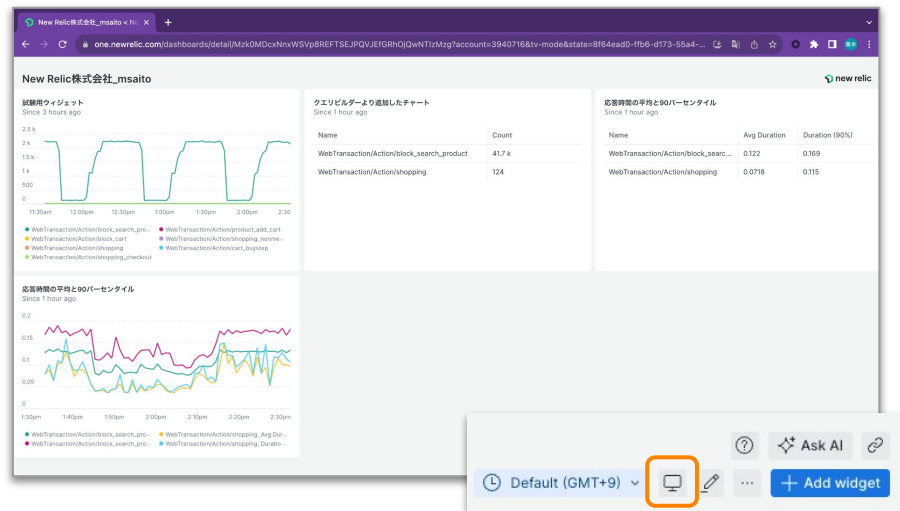
## Note機能

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)

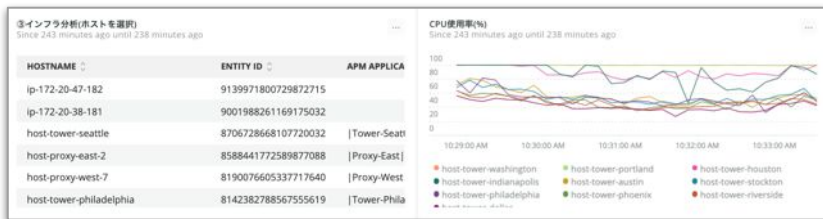


# ダッシュボードの便利機能 ②

- 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)

Default

CUSTOM [Reset to now](#)

Jul 25, 2023 02:29 pm Jul 25, 2023 02:34 pm

Asia/Tokyo [Cancel](#) [Apply](#)

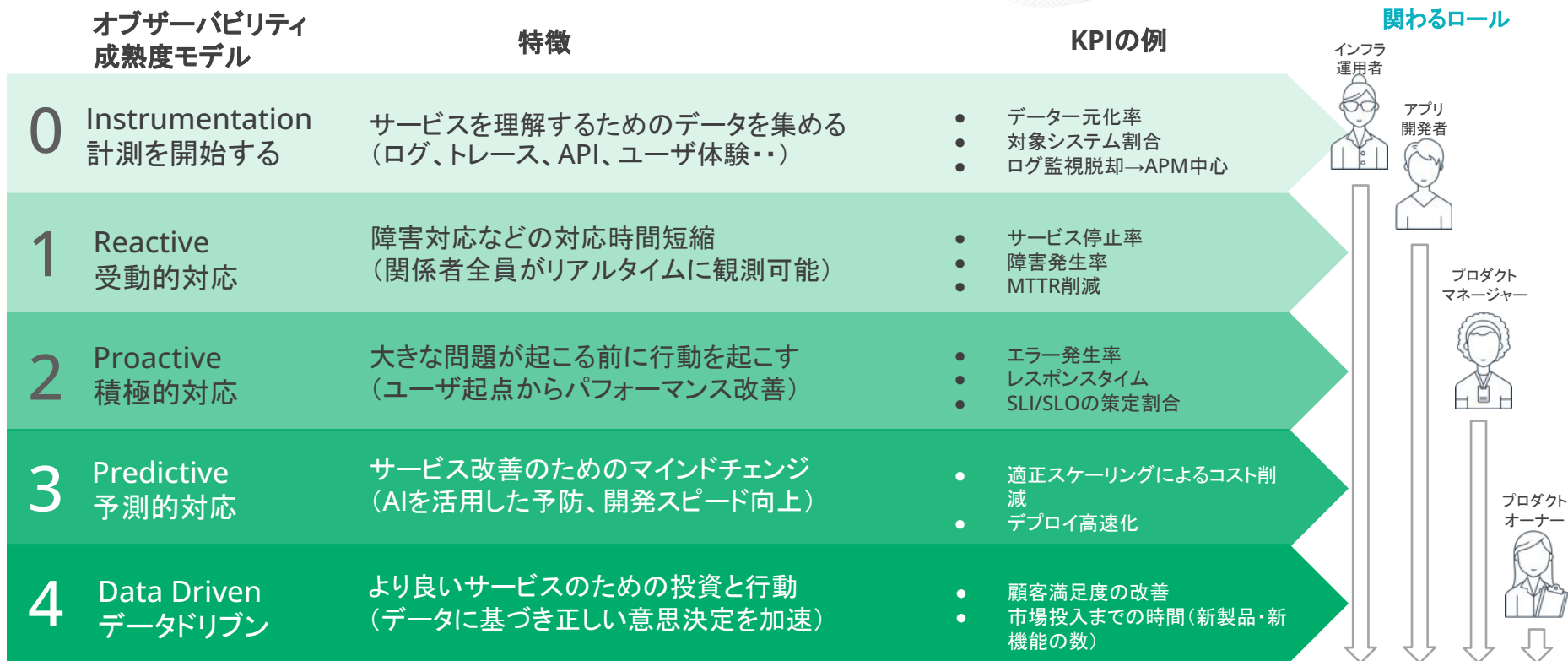
Recent

You don't have any history yet :-)

- 5 minutes
- 30 minutes
- 60 minutes
- 3 hours
- 6 hours
- 12 hours
- 24 hours
- 3 days
- 7 days
- 1 month
- 3 months
- 6 months

Set custom ▶

# 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

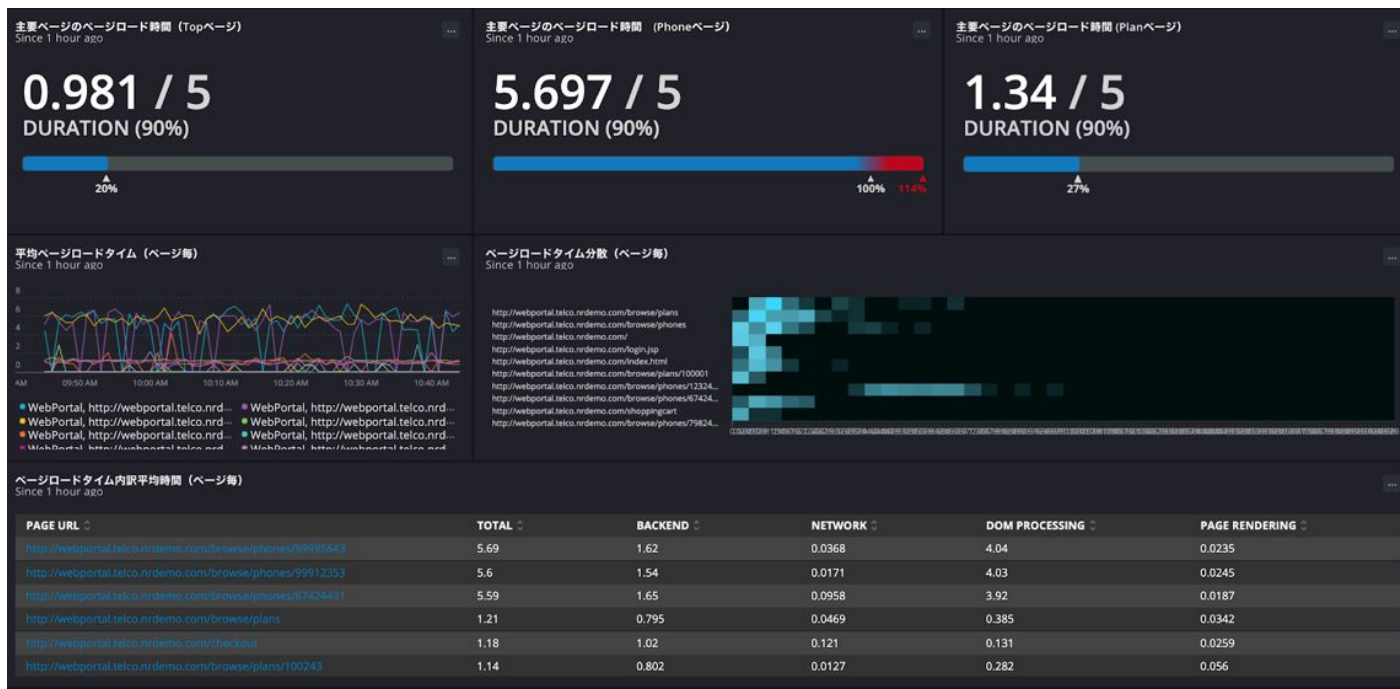
1 受動的

例1: サーバー稼働状況



# 目的に応じたダッシュボード

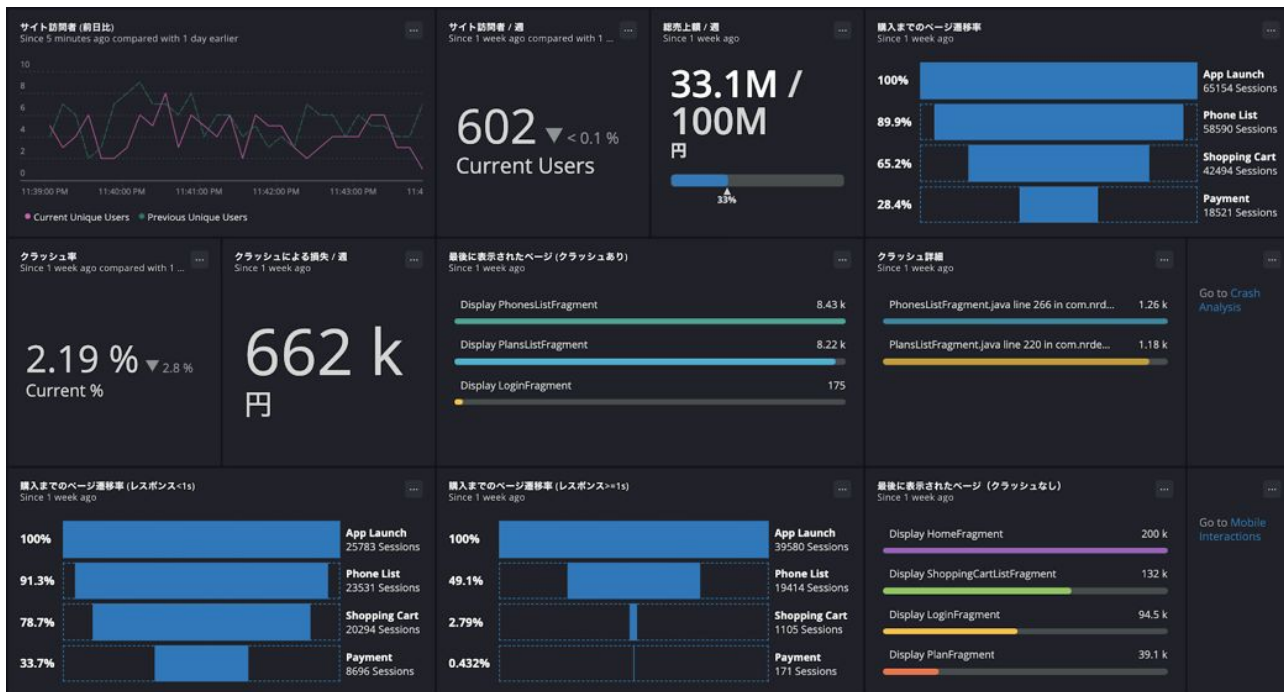
## 例2: ユーザー体験のリアルタイムモニタリング



# 目的に応じたダッシュボード

3 予測的

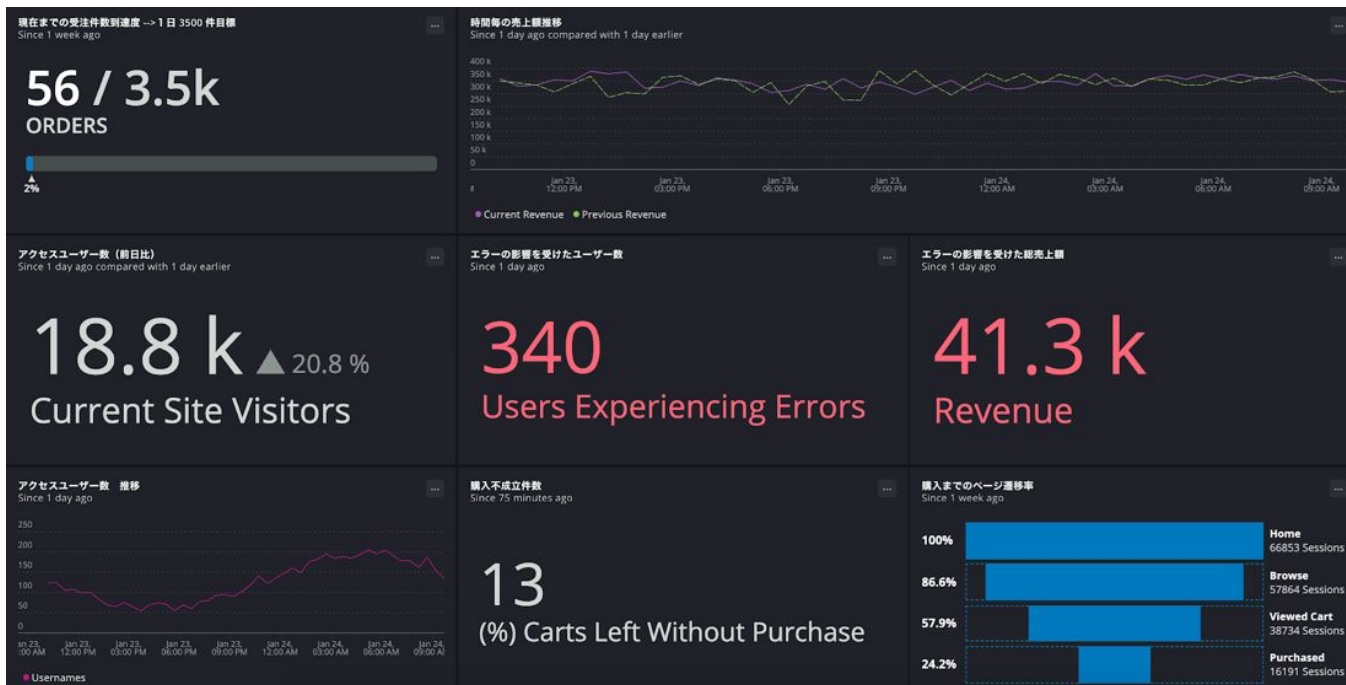
## ECサイト購買分析





# 目的に応じたダッシュボード

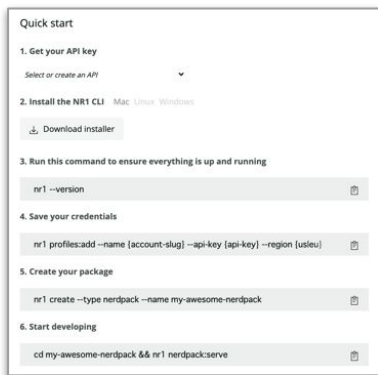
## 例4: 売上データと連動したビジネスダッシュボード



# さらに高度なビジュアルライズも

- 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



Quick start

1. Get your API key  
Select or create an API
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:cmd --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



# Pre-Build Dashboards

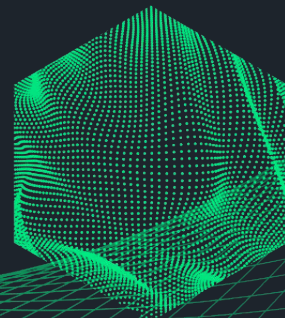
New Relicがあらかじめ準備している  
ダッシュボードがあります。

- 言語
- ミドルウェア
- クラウド

The screenshot displays the New Relic dashboard interface. At the top, there is a search bar with the text "Search for any technology". Below it, "Popular search terms" are listed: AWS, Node, Java, PHP, and Kubernetes. The "Data sources" section shows counts for "Dashboards (497)", "Alerts (372)", and "Apps & visualizations (32)". The main content area is titled "Application monitoring (12)" and features a grid of 12 pre-built dashboard cards. Each card includes a thumbnail image of the dashboard, the technology name, and the creator information. The technologies shown are: .NET, ADO.NET, Blazor Server, C, fastAPI, Golang, IBM Websphere Liberty Profile, MEXN, Netty, Node.js, PHP, and Ruby. Each card also indicates when it was last edited.

# ダッシュボードの 作成

NRU300 - 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
- 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!!

Option!!



**new relic**<sup>®</sup>

**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

The screenshot shows the New Relic Dashboards page. On the left is a navigation menu with 'Dashboards' highlighted. In the main area, the 'Create a dashboard' button is highlighted. A callout box on the right contains text about pre-built dashboards. A bottom callout box contains a numbered list of steps.

① [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。

② Create a dashboard

③ [Browse pre-built dashboards](#)  
Install a quickstart with dashboards made for what you're monitoring.

④ [Create a new dashboard](#)  
Build a dashboard from scratch and see what's happening with your data.

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

Dashboards

NRU302\_企業名\_イニシャル ☆ Tags ⓘ Metadata ⓘ Workloads ⓘ

+  
Add a new chart

[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

**NRU300-企業名-イニシャル**  
e.g. NRU300-NEWRELIC-NR

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



手順

# 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 blue box and a green circle labeled '7'. A dashed green arrow points from this menu item to the 'AWS' integration in the 'INTEGRATIONS' section, which is also highlighted with a blue box and a green circle labeled '8'. Another dashed green arrow points from the 'AWS' integration to the 'S3 dashboard' entry in the table below, which is highlighted with a blue box and a green circle labeled '9'. The table lists various AWS integrations with their respective dashboard names and configuration options.

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

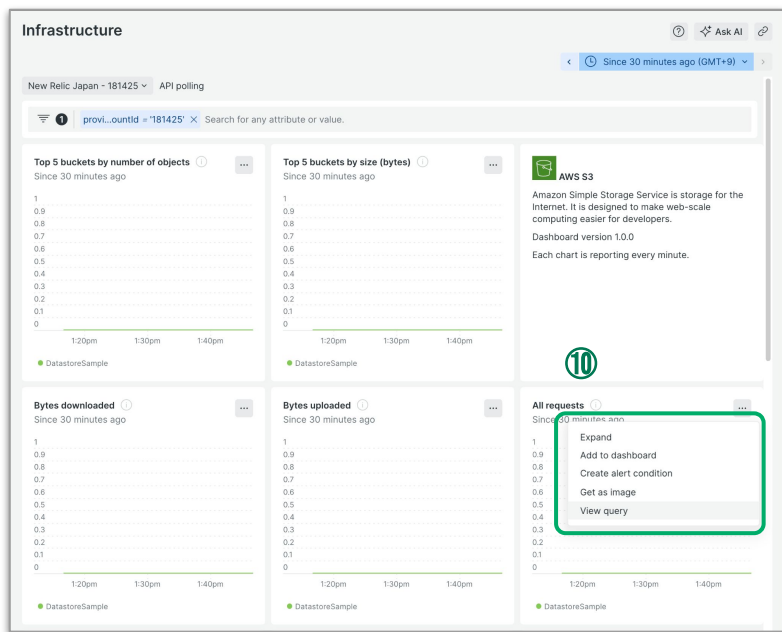
⑨ [S3 Dashboard] をクリックします

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 Rel...
New Relic 株式会社_NRU303 / Ne...	New Rel...
New Relic 株式会社_NRU303 sam...	New Rel...
New Relic株式会社_msaito / ハン...	New Rel...
New Relic株式会社_msaito / ハン...	New Rel...
NRU#ホワイトボードサンプル / NR...	New Rel...
NRU#レポートサンプル / NRU#レ...	New Rel...
test / test	New Rel...

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 **G E**

Alerts & AI

APM & Services

Browser

Dashboards

Errors Inbox

Infrastructure

Logs

**Metrics & Events**

Query Your Data

Service Levels

Synthetic Monitoring

Help 9

Add User

New Relic University Japan

Metrics & Events

Metrics **Events** Filter to All entities

Event type

New Relic Events

SyntheticCheck

SyntheticRequest

System Sample

**Transaction**

WorkloadStatus

Custom events

ApacheSample

ApplicationAgentContext

ComputeSample

Plot

Dimensions

Event type

Plot

Dimensions

Select count(\*) × From Transaction ×

NRQL SELECT count(\*) FROM Transaction FACET 'r

Transaction

800

600

400

200

0

11:25am 11:30am

request.method 3

request.uri 60

response.headers.content... 2

response.statusCode 3

sampled

tags.account New Relic University Japan

tags.accountid 3940716

request.uri

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

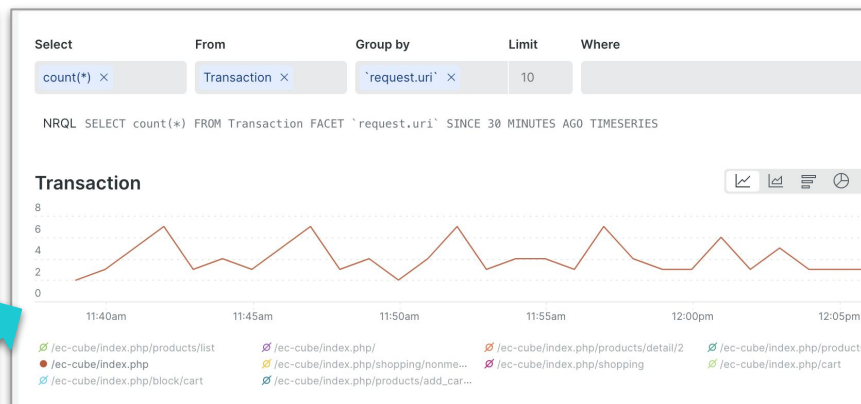
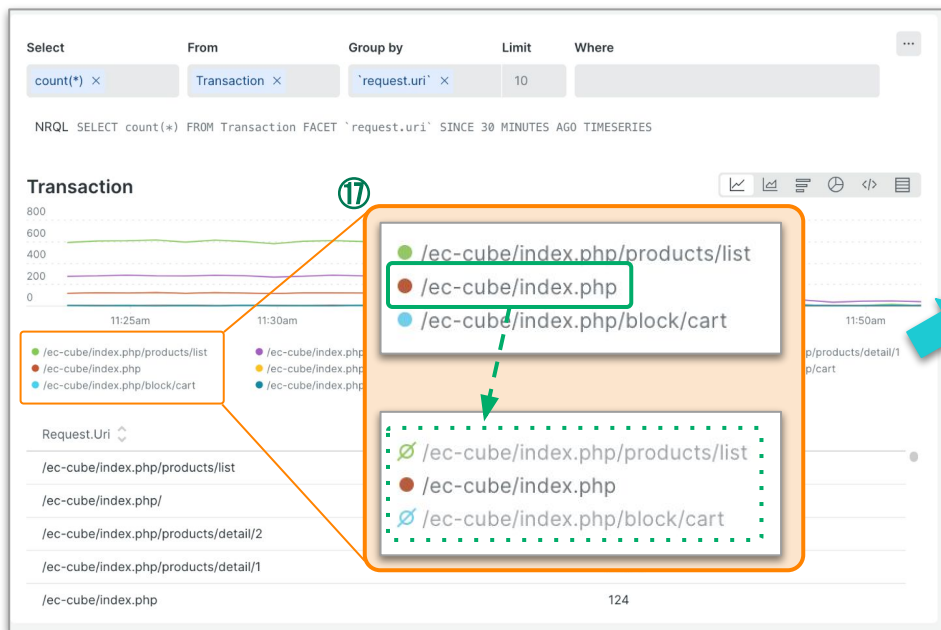


手順

- 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

①7 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

The screenshot displays the New Relic dashboard interface. On the left, the 'Dimensions' list includes 'name' which is highlighted with a green box and a circled '18'. The main chart area shows a line graph titled 'Transaction' with a time picker set to '3 hours' (circled '19'). Below the chart is a table showing transaction counts grouped by name.

Name	Count
WebTransaction/Action/block_search_product	133 k
WebTransaction/Action/product_add_cart	378
WebTransaction/Action/block_cart	
WebTransaction/Action/shopping	
WebTransaction/Action/shopping_che...	
WebTransaction/Action/cart_buystep	
WebTransaction/Action/shopping_non...	
WebTransaction/Action/shopping...	



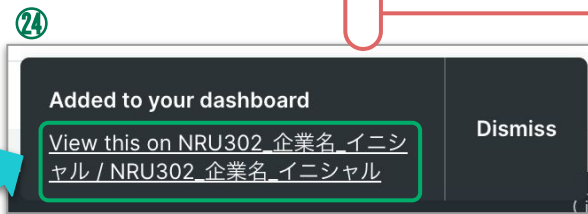
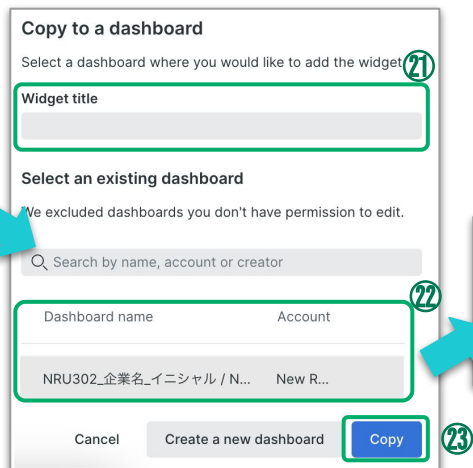
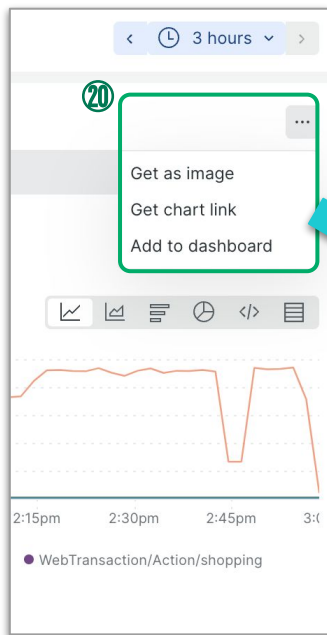
⑱ Dimensionsから [name] を選択し、チャートが変化することを確認します

手順

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
- ②① [Widget title] に任意のチャート名を入力します
- ②② [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
- ②③ [Copy] をクリックします
- ②④ 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 process is shown in three steps:

- Step 25: A chart titled "Hands on 2-2" is shown. A menu is open over the top right corner, and the "Duplicate" option is highlighted.
- Step 26: The duplicated chart is shown. A menu is open over the top right corner, and the "Edit" option is highlighted.
- Step 27: The "Edit widget" screen is shown. The "See customization options" button is highlighted.



手順

- 追加したチャート右上の [...] - **[Duplicate]** をクリックします
- 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
- 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

CE 3 HOURS AGO **TIMESERIES**

Clear **Run**

**Chart type** Pie

Counts (133 k)

- WebTransaction/Action/block\_search\_product
- WebTransaction/Action/product\_add\_cart
- WebTransaction/Action/block\_cart
- WebTransaction/Action/shopping
- WebTransaction/Action/cart\_buystep
- WebTransaction/Action/shopping\_checkout
- WebTransaction/Action/shopping\_nonmember

Area

**Bar**

Billboard

Bullet

Funnel

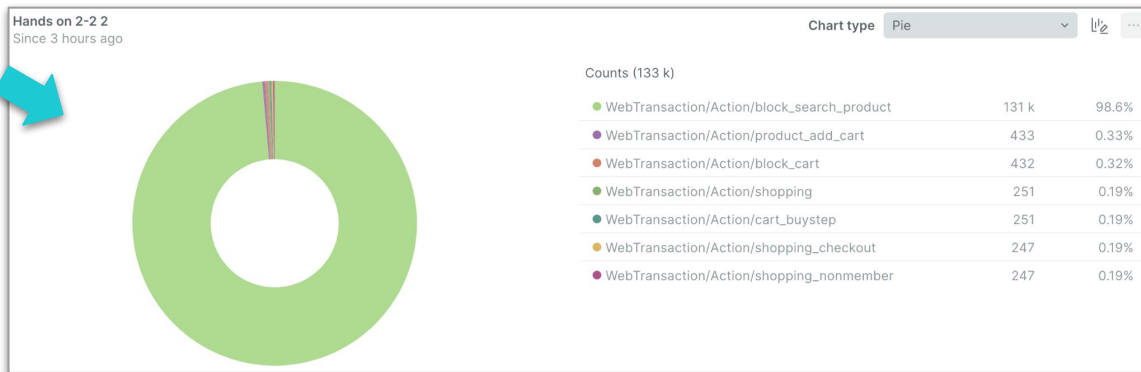
Heatmap

Histogram

JSON

**Pie**

Stacked Bar



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

The image shows the configuration process for facet filtering. On the left, the 'Basic information' panel is visible, with the 'Facet Linking' section expanded. The 'Filter the current dashboard' toggle is turned on (indicated by a blue circle 30). The 'Save' button is highlighted with a blue circle 31. A blue arrow points from the 'Save' button to the dashboard view on the right. The dashboard view shows a donut chart and a table of counts. A blue circle 32 highlights the 'Filter this dashboard' button in the table. A second blue arrow points from the table to the final filtered dashboard view on the right, which shows the donut chart and table with the first row highlighted in blue.



手順

③⑩ [Filter the current dashboard] を有効化 (  ) します

③⑪ チャートを保存します

③⑫ 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

① + Add widget

Add a widget 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.

Add text, images, or links

Markdown

```

1 # Hola! नमस्ते! Hello! やあ! ![New Relic logo](https://newrelic.com
  /static-assets/images/icons/avatar-newrelic.png)
2 ## Use this to add context for the data in your dashboards
3 > It can be `code`, important, slightly highlighted, ~no
  longer relevant~.
4
5 Add your bullets—for unordered items:
6 - You can navigate within the platform by [adding links like this]
  (https://one.newrelic.com/dashboards)
7 - Or you can link [external websites like your repositories](https
  ://github.com/newrelic)
8 ---
9 Or a TODO list:
10 - [x] Add links to related dashboard pages
11 - [ ] Create a widget facet by environment to use it as a filter
12
13 ### Add your own context—[here's what we support](https://github
  .com/newrelic)

```

Preview

**Hola! नमस्ते! Hello! やあ!**

Use this to add context for the data in your dashboards

It can be `code`, **important**, *slightly highlighted*, no-longer-relevant.

Add your bullets—for unordered items:

- You can navigate within the platform by [adding links like this](https://one.newrelic.com/dashboards)
- Or you can link [external websites like your repositories](https://github.com/newrelic)

Or a TODO list:

- Add links to related dashboard pages
- Create a widget facet by environment to use it as a filter

Add your own context—[here's what we support](https://github.com/newrelic)

Cancel Save

時間に余裕のある方は  
Markdownを編集してく  
ださい

参考: GitHub 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>

© 2025 New Relic, Inc. All rights reserved.



手順

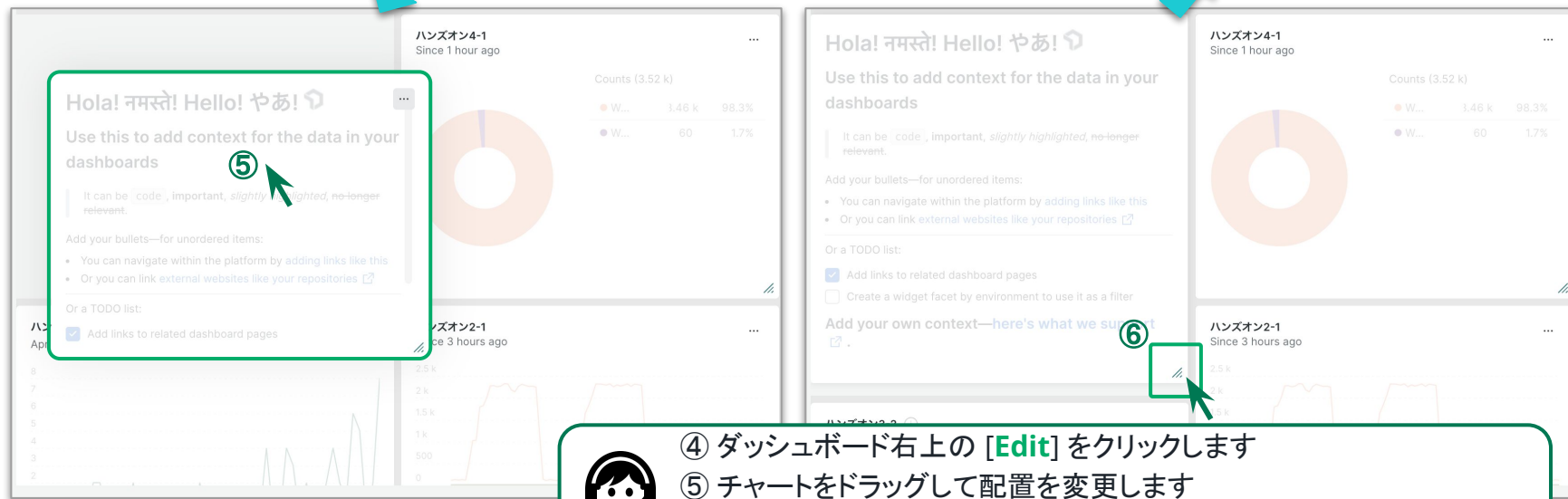
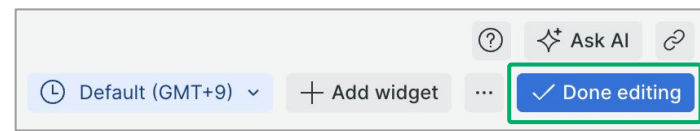
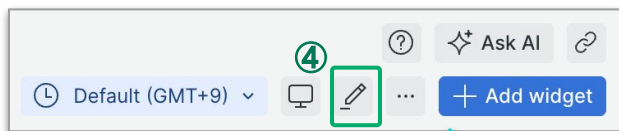
① ダッシュボード右上の [+ Add widget] をクリックします

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



手順

- 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)

NRU300 - 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



```
1 SELECT average(duration) FROM Transaction WHERE appName = 'EC-site' FACET name SINCE 30 minutes ago TIMESERIES
```

Copy query link ...

Recommended queries ▾ 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: 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できる(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

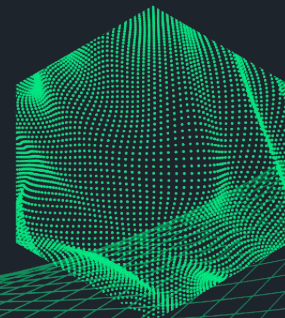
## WHERE句について

- 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

NRU300 - 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**<sup>®</sup>

**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

The image shows two screenshots from a dashboard. The first screenshot shows a line chart titled 'All requests' with a context menu open over it. The menu includes options like 'Expand', 'Edit', 'Duplicate', 'Copy', 'Delete', 'Create alert condition', 'Get as image', and 'View query'. A red circle with the number '1' is placed over the 'View query' option. The second screenshot shows the NRQL query in a query builder interface. A red circle with the number '2' is placed over the 'FROM' clause of the query. A red arrow points from the 'View query' option in the first screenshot to the query in the second screenshot.

①

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

②

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



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'
```



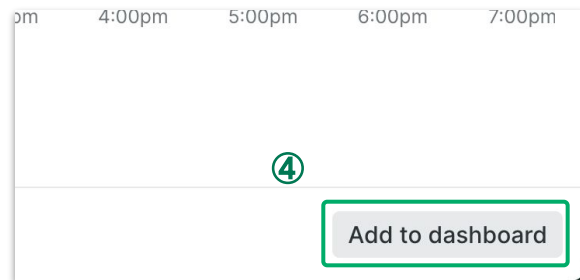
変更後

```
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

The screenshot shows the New Relic dashboard. In the left-hand navigation menu, the 'Query your data' option is highlighted with a green box and a circled number 5. The main content area shows a dashboard titled 'New R...' with a search bar and a line chart titled 'All requests' for the entity 'newrelic-kk-public-docs'. The chart displays request volume over time from 0pm to 7:00pm, with several peaks reaching up to 3.8.

本ページ以降でいくつかのNRQLを入力し結果の確認を行っていきます

- 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



手順

⑧ 変更後のNRQLを実行し結果を確認します

# 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



手順

⑨ 変更後のNRQLを実行し結果を確認します



# 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]



手順

⑪ 変更後のNRQLを実行し結果を確認します

# 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

Tab 1

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

Recommended queries Recent queries

Since 1 hour ago

Chart type Table

Name	Count
WebTransaction/Action/block_search_product	3.48 k
WebTransaction/Action/shopping	62

4,745 events inspected

**34**

Add to dashboard

Dashboards

NRU302\_企業名\_イニシャル ☆

+ Add variable Search for any attribute or value.

ハンズオン2-1  
Since 3 hours ago

ハンズオン2-2  
Since 1 hour ago

Name	Count
WebTransaction/Action/...	41.5 k
WebTransaction/Action/...	61

Legend:

- WebTransaction/Action/block\_search\_product
- WebTransaction/Action/product\_add\_cart
- WebTransaction/Action/block\_cart
- WebTransaction/Action/shopping



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

ハンズオン2-2  
Since 1 hour ago

⑬

Name
WebTransaction/Action/...
WebTransaction/Action/...

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

Tab 1 × Tab 2 × +

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



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 [, 何パーセントイルにするかの数値 ])**: 指定パーセント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]

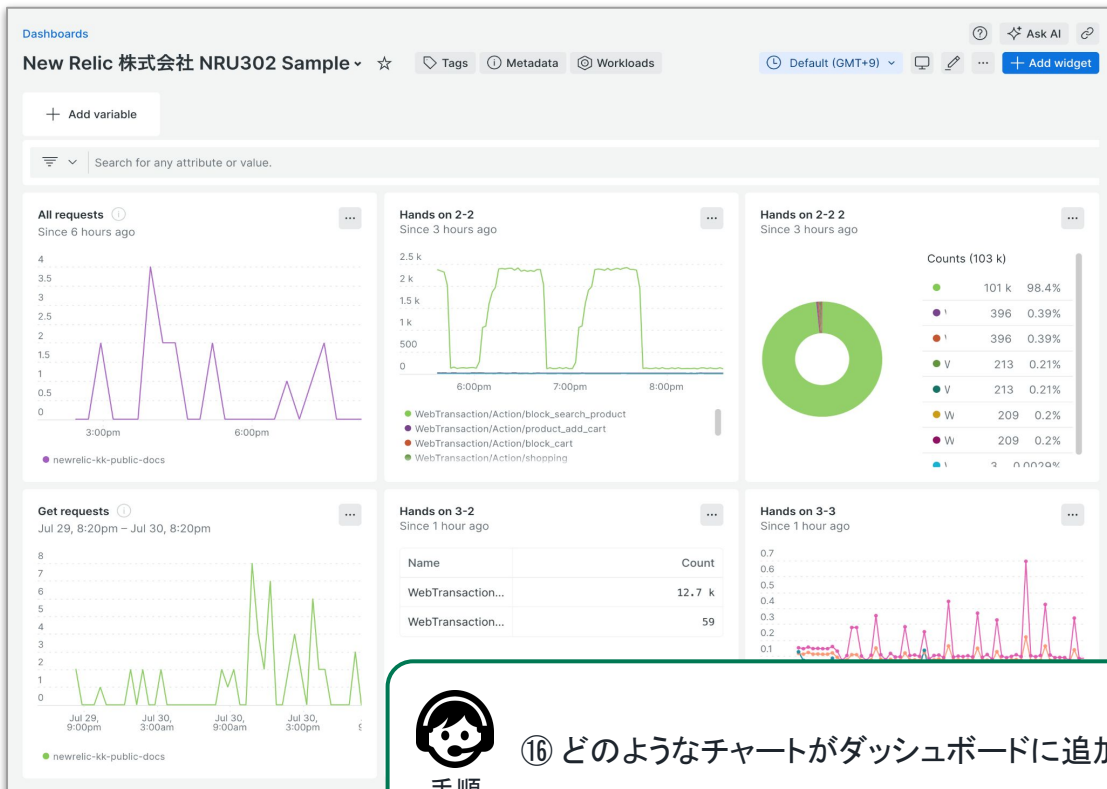


手順

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!!

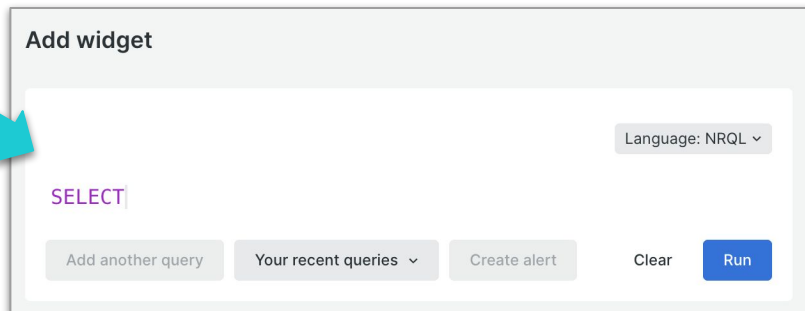
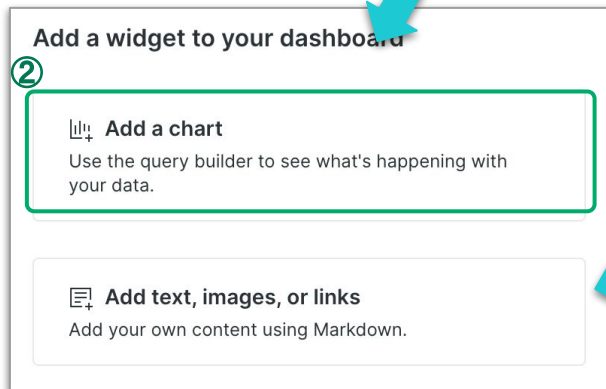
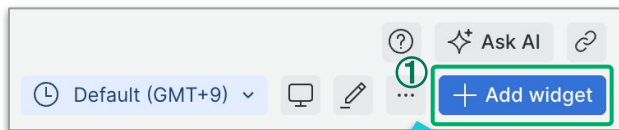
# NRQLを用いて様々な チャートを作成する

NRU300 - 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...)	<ul style="list-style-type: none"> <li>データ元化率</li> <li>対象システム割合</li> <li>ログ監視範囲-APM中心</li> </ul>
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)	<ul style="list-style-type: none"> <li>サービス停止率</li> <li>障害発生率</li> <li>MTTR削減</li> </ul>
2 Proactive 積極的対応	大きな問題が起こる前に行動を起こす (ユーザー起点からパフォーマンス改善)	<ul style="list-style-type: none"> <li>エラー発生率</li> <li>レスポンスタイム</li> <li>SLI/SLOの測定割合</li> </ul>
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)	<ul style="list-style-type: none"> <li>適正スケーリングによるコスト削減</li> <li>デプロイ高速化</li> </ul>
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)	<ul style="list-style-type: none"> <li>開発品質の改善</li> <li>市場投入までの時間 (新製品、新機能の数)</li> </ul>



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

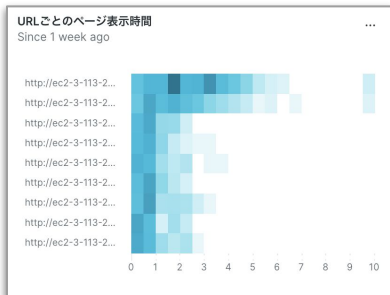


```
SELECT percentage(count(result), where result = 'SUCCESS') AS '稼働率(週)'  
FROM SyntheticCheck  
SINCE 1 week AGO  
COMPARE WITH 1 week AGO
```

- 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

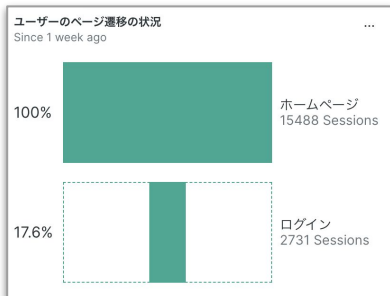


```
SELECT histogram(duration,10,20)
FROM PageView
FACET pageUrl
SINCE 30 MINUTES AGO
```

- 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

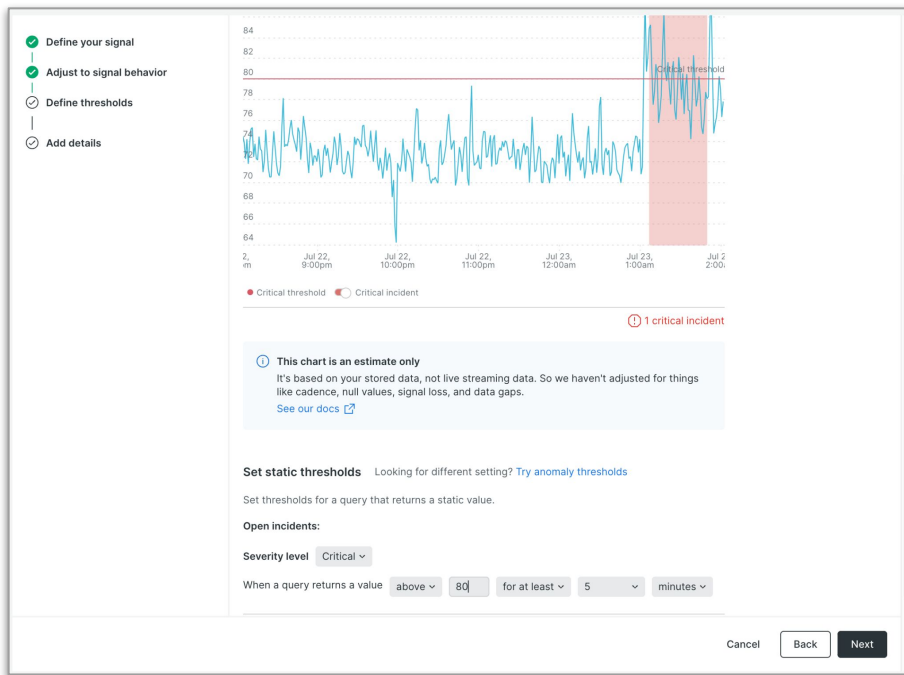
アラートの  
作成

# 高度なアラート設定

NRU300 - 座学

# アラート機能

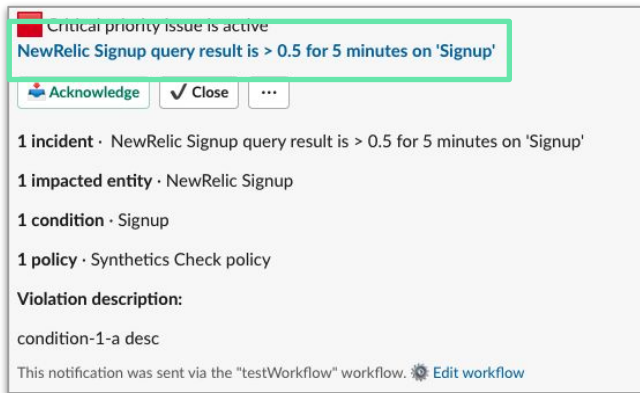
- New Relicが取得しているデータを使ってしきい値(動的/静的)を設定し、アラートを発報することが可能



# 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

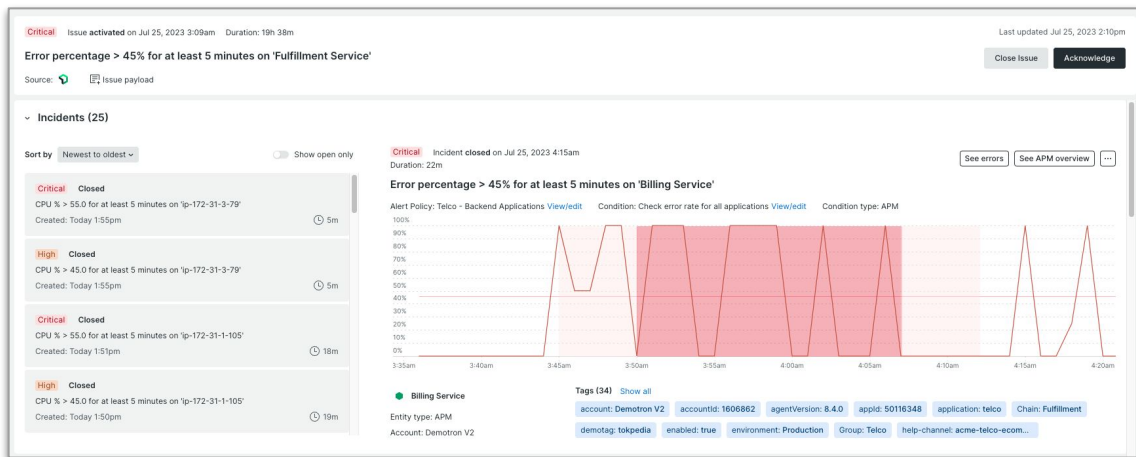
1 condition · Signup

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



Critical Issue activated on Jul 25, 2023 3:09am Duration: 19h 38m  
Error percentage > 45% for at least 5 minutes on 'Fulfillment Service'

Source: [Issue payload](#)

Last updated Jul 25, 2023 2:10pm [Close issue](#) [Acknowledge](#)


Incidents (25)

Sort by [Newest to oldest](#)  Show open only

- Critical Closed**  
CPU % > 55.0 for at least 5 minutes on 'ip-172-31-3-79'  
Created: Today 1:55pm 5m
- High Closed**  
CPU % > 45.0 for at least 5 minutes on 'ip-172-31-3-79'  
Created: Today 1:55pm 5m
- Critical Closed**  
CPU % > 55.0 for at least 5 minutes on 'ip-172-31-1-105'  
Created: Today 1:51pm 18m
- High Closed**  
CPU % > 45.0 for at least 5 minutes on 'ip-172-31-1-105'  
Created: Today 1:50pm 19m

**Critical Incident closed on Jul 25, 2023 4:15am**  
Duration: 22m [See errors](#) [See APM overview](#)

Alert Policy: Telco - Backend Applications [View/edit](#) Condition: Check error rate for all applications [View/edit](#) Condition type: APM



100%  
90%  
80%  
70%  
60%  
50%  
40%  
30%  
20%  
10%  
0%

3:35am 3:40am 3:45am 3:50am 3:55am 4:00am 4:05am 4:10am 4:15am 4:20am

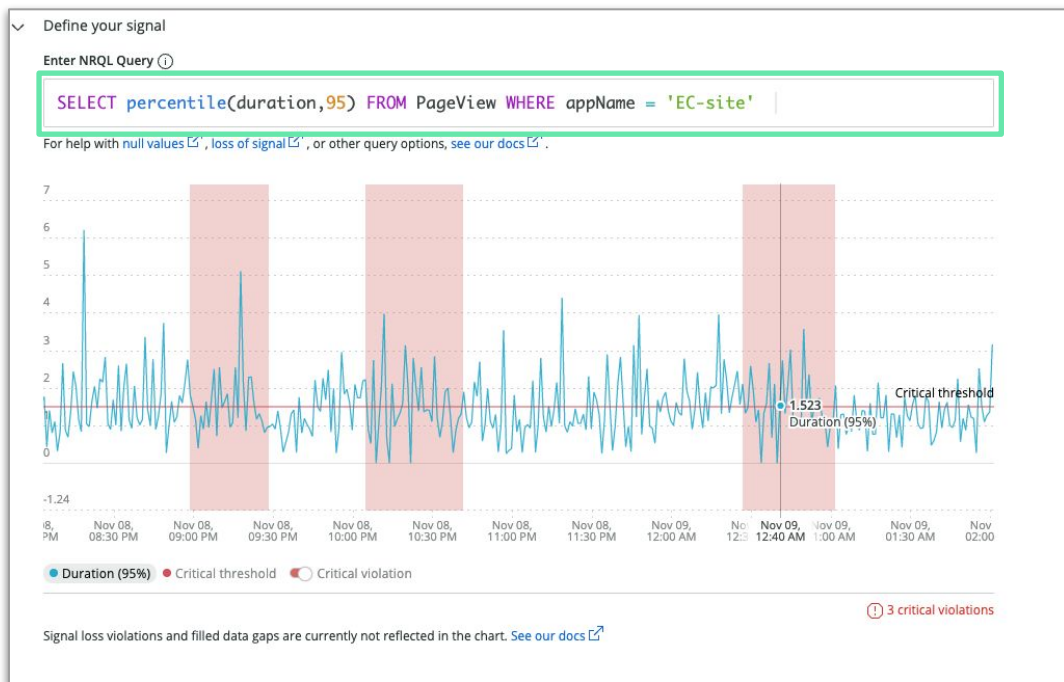
● Billing Service

Tags (34) [Show all](#)  
account: Demotron V2 accountid: 1606862 agentVersion: 8.4.0 appld: 50116348 application: telco Chain: Fulfillment  
Entity type: APM  
Account: Demotron V2  
demotag: tokpedia enabled: true environment: Production Group: Telco help-channel: acme-telco-ecom...



# 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、閾値の設定を行えます

## NRUハンズオンのお知らせ

2025.02.26 (Wed) 15:00 - 17:00

NRU300 - アラート設計の基本と活用

<https://newrelic.com/jp/events/2025-02-26/nru300alt>

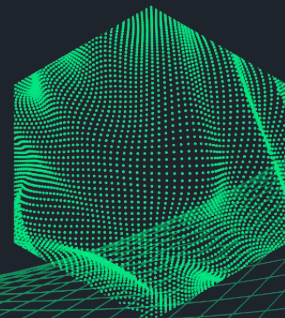


このハンズオントレーニングでカバーするトピック

- 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

NRU300 - 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**<sup>®</sup>

**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 red circle and a red arrow pointing to the 'Alert Policies' sub-menu item. On the right, the 'Alert Policies' page is displayed, showing a table of 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 red box and a red circle, with a red arrow pointing to it from the menu.

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	...

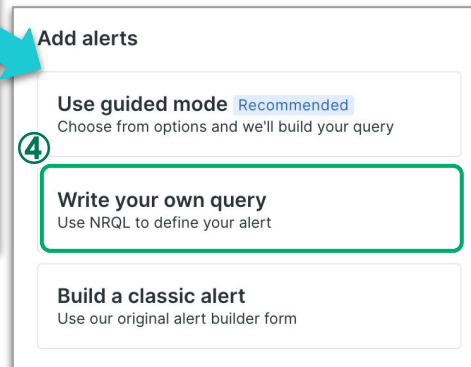
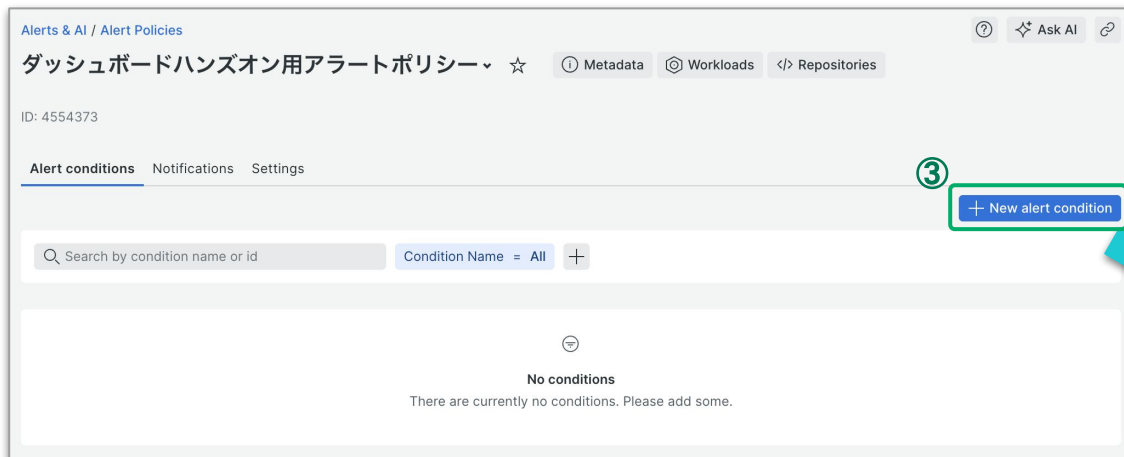


手順

- 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'`

Query the data you want to monitor ⓘ

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

Clear **Run**

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

5.5  
5  
4.5  
4  
3.5  
3  
2.5  
2

Cancel **Next**

⑥

⑦

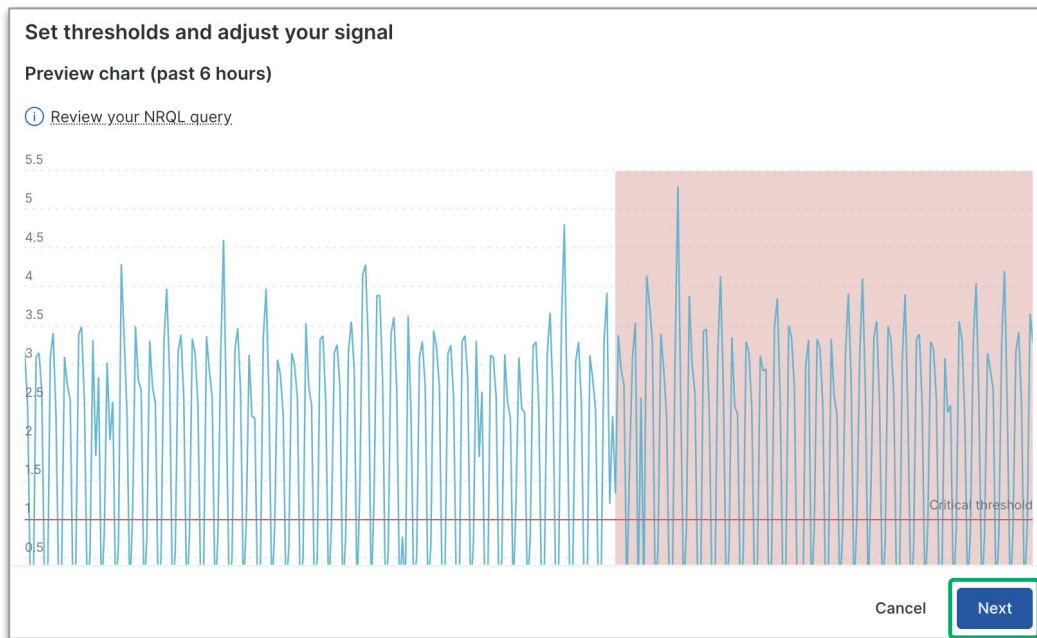


手順

- 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。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

**NRU300-企業名-イニシャル**  
e.g. NRU300-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

NRU300

# Dashboard と NRQL の 便利な機能

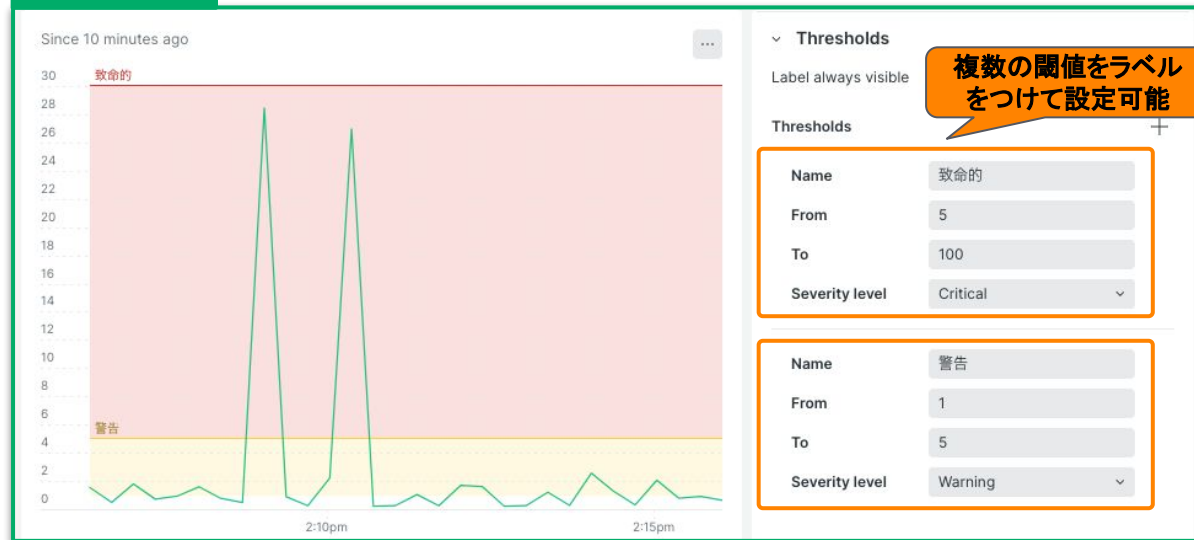
NRU300

## 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た配色**、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	From	To	Severity level
Avg External Duration	0.1	1000	Critical

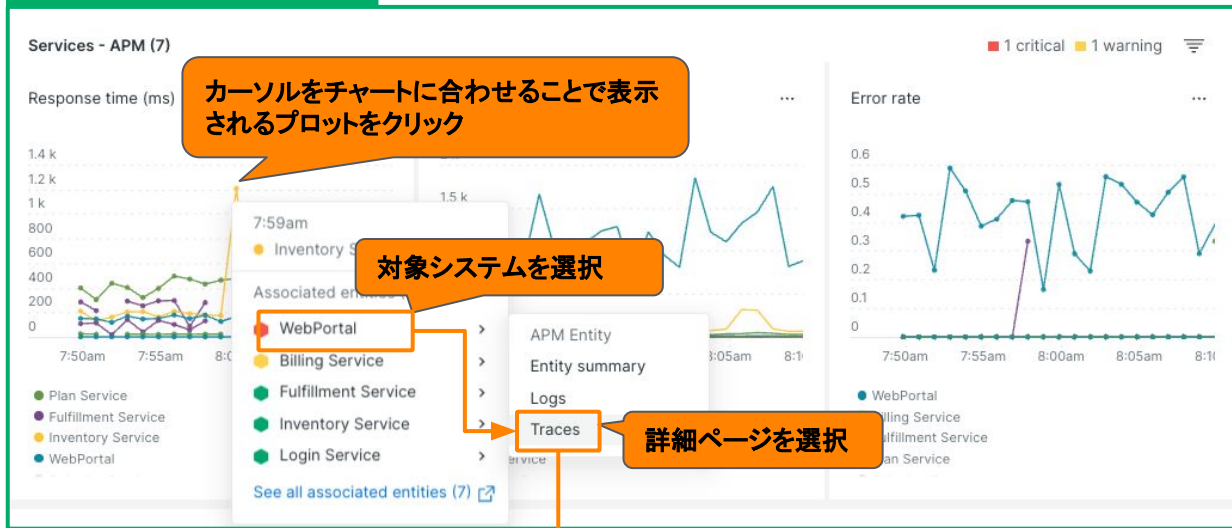
## チャートからシームレスにドリルダウンして詳細分析

- 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

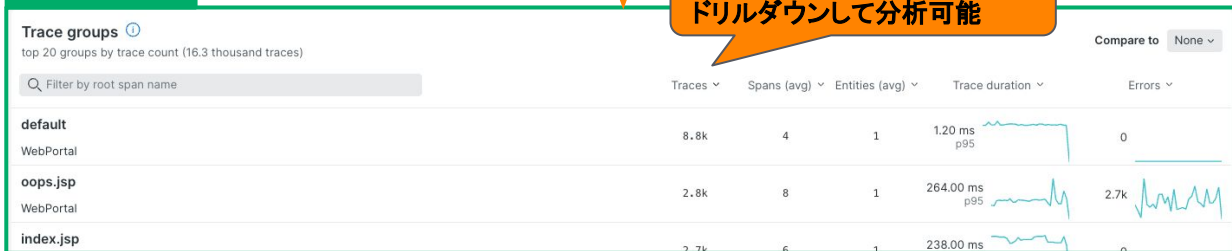
†WHERE句でentityGuid、entity.guid、appName、entity.name、または、entityNameを指定

参照：[What's new](#)

### 例) Workloads > Activity



### 例) Traces

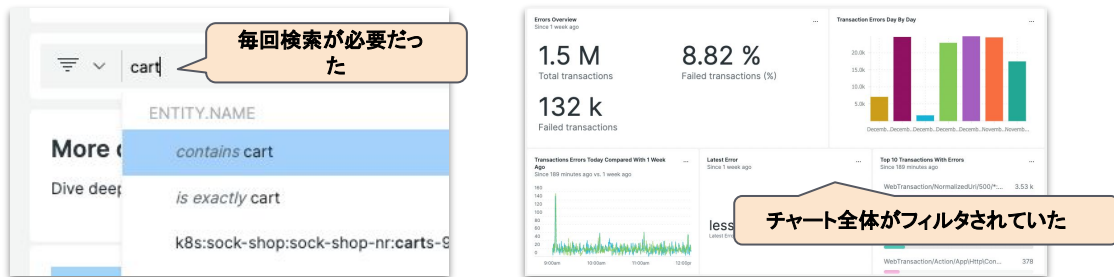


# ダッシュボード テンプレート変数

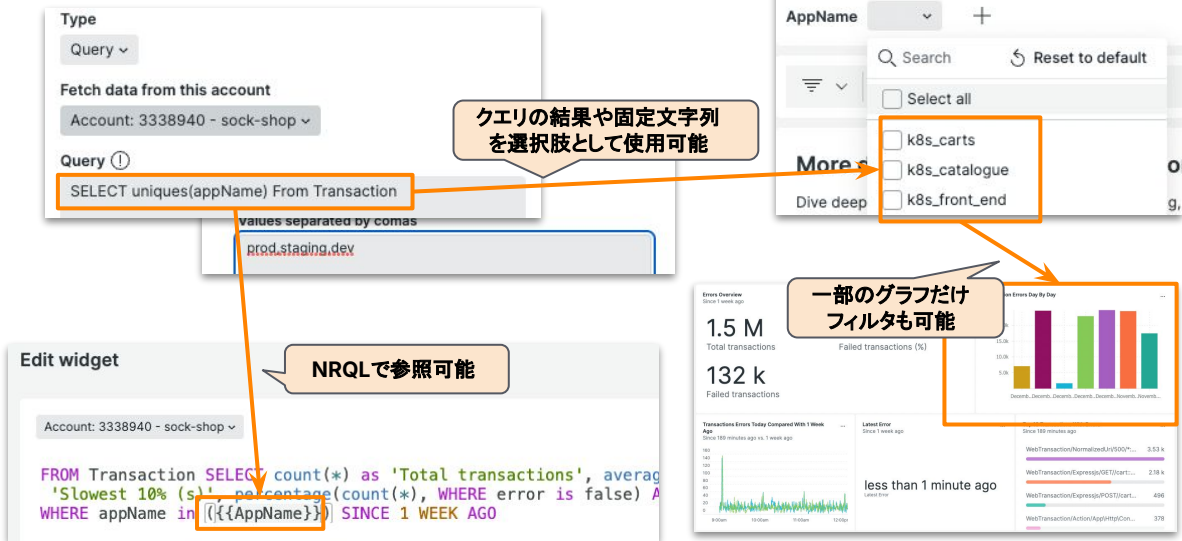
- あらかじめ変数を用意することで、フィルタ項目の検索が不要になります
- 個々のチャートをそれぞれ別々の変数でフィルタリングできます
- 固定リストとNRQLのクエリ結果を選択肢にできます

参照: [テンプレート変数](#)

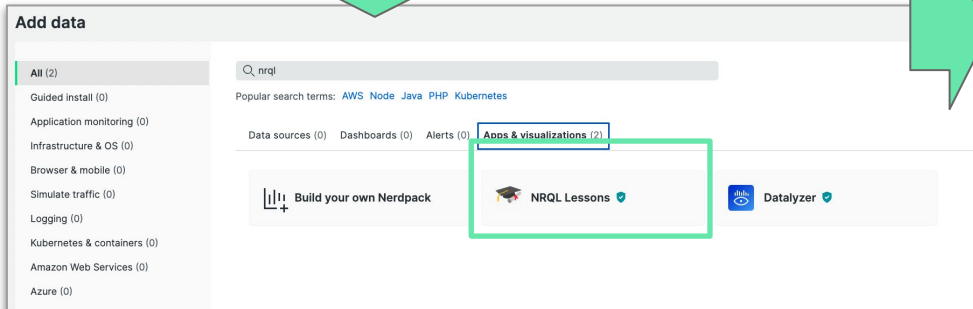
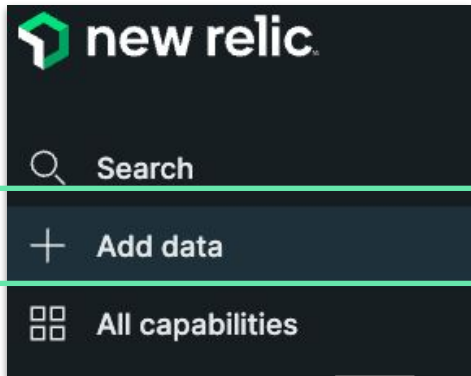
## 従来のフィルタ機能



## 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



# 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 for a service named 'WebPortal'. The interface includes a top navigation bar with 'APM & Services / Services - APM', 'WebPortal', 'Tags', and 'Metadata'. Below this is a filter bar with 'critical alerts', 'Good', and 'critical alerts' filters, and a time range of 'Since 30 minutes ago (UTC)'. The main content area is titled 'Query your data' and features a search bar with 'Query using AI - Experimental' and a 'Generate query' button. A dropdown menu shows 'Accounts: Demotron V2, ...'. The query editor contains the following NRQL: `FROM Transaction SELECT count(*) FACET accountId() TIMESERIES`. Below the query editor is a 'Recommended queries' section with a 'Run' button. A line chart displays data for 'Since 1 hour ago' with a y-axis from 0 to 4k and an x-axis from 6:40am to 7:30am. The chart shows multiple lines representing different accounts. A 'Customize this visualization' panel on the right includes 'Basic information' with fields for 'Name' and 'Chart type' (set to 'Line').

1 最下部のバーをクリックすることで Query your data画面を表示

2 New Relic AIでクエリを生成可能

3 おすすめクエリを選択可能

4 NRQLを入力

5 複数アカウントを対象としたクエリが可能に

6 アカウントごとにFACETも可能



# 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, ①
  toTimestamp(jparse(builds)[0][started_at]) as StartedAt, ②
  toDatetime(toTimestamp(jparse(builds)[0][finished_at]), 'yyyy/MM/dd') as FinishedAt, ③
  toTimestamp(jparse(builds)[0][finished_at]) - toTimestamp(jparse(builds)[0][started_at]) as Duration,
  encode(builds, 'base64'), ④
  builds ⑤
FROM Log SINCE 40 MINUTES AGO WHERE builds is not null
```

①	②	③	④	⑤		
フィールドを取得	文字列から日時に変換	変換後計算可能	エンコード	元のデータ		
timestamp	Stage	StartedAt	FinishedAt	Duration	encode(builds, ...)	builds
13:33:29.611	build	1707117637233	2024/02/05	40189	W3siaWQjOjEwODU...	[{"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](#)

UI上からCSVをアップロード

NRQLのlookup関数で内容を確認可能

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

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<sup>+</sup>でも使える!
- Query Builderで利用可能

参照:

- [公式ドキュメント](#)

<sup>+</sup>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
```

ホスト名	アプリ応答時間
Host	Avg Duration
order-composer-5495f9b947-m9t6	5.71
order-composer-5495f9b947-2gt6z	5.67
order-composer-5495f9b947-dr6pl	5.53
order-composer-5495f9b947-n8x52	5.45
order-processing-695b9d958f-bzzvn	4.73

## インフラ由来のデータ (SystemSample Event)

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

ホスト名	インフラリソース使用率	
Hostname	Avg Cpu Percent	Avg Memory Used Percent
host-tower-portland	99.8	39.8
host-tower-washington	99.8	32.8
host-tower-houston	96.6	66.1
host-tower-indianapolis	95.1	79.2
host-tower-stockton	72.2	59.9
host-tower-austin	71.8	61.2

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

```
FROM Transaction JOIN (FROM SystemSample 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
```

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

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

NRU300

# New Relic University(NRU)とは

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
<p><b>New Relic</b> を使い始める</p> <p>New Relic 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</p> <p><a href="#">APM / Browser / Infrastructure / Logs / Mobile (iOS/Android) / AWS統合 / Azure統合 / GCP統合</a> インストール手順</p>	<p><b>Observability/New Relic</b> を知る</p> <p>New Relic やオブザーバビリティに関する基礎知識を座学にて学習</p> <p>NRU Practitioner オブザーバビリティ入門</p> <p>NRU 101 New Relic 入門</p> <p>Hands on for Beginners 初心者向けハンズオン(オフライン)</p>	<p><b>New Relic</b> の主要機能を学ぶ</p> <p>New Relicに含まれる主要機能に含まれる54の機能群を動画で説明</p> <p>フロントエンド、バックエンド、アラート、データ分析、ユースケースに応じた機能</p>	<p><b>New Relic</b> の使い方を体感する</p> <p>New Relic 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</p> <p><b>NRU 300</b></p> <p>アプリケーションとインフラ性能観測の基本</p> <p>Webサイト上のユーザー体験観測の基本</p> <p>ダッシュボード開発とNRQLの基本</p> <p>アラート設計の基本と応用</p> <p><b>NRU 400</b></p> <p>IDEと連携し、問題解決を加速するNew Relic活用の実践</p> <p>SLI/SLO設計の基本</p>	<p>資格を得る</p> <p>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</p> <p><b>NRU for Exam</b> New Relic Full-Stack Observability 認定試験対策講座</p> <p>フルスタックオブザーバビリティ認定試験</p>
<p>▶サインアップ方法</p> <p><a href="https://newrelic.com/jp/blog/how-to-relic/create-new-account">https://newrelic.com/jp/blog/how-to-relic/create-new-account</a></p> <p>▶インストールガイド</p> <p><a href="https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide">https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide</a></p>	<p>▶オンデマンドセミナー</p> <p><a href="https://newrelic.com/jp/resources/presentations/nru-practitioner2022">https://newrelic.com/jp/resources/presentations/nru-practitioner2022</a></p> <p><a href="https://newrelic.com/jp/resources/presentations/nru101-2022">https://newrelic.com/jp/resources/presentations/nru101-2022</a></p>	<p>▶主要機能解説動画</p> <p><a href="https://newrelic.com/jp/learn">https://newrelic.com/jp/learn</a></p>	<p>▶開催スケジュール</p> <p><a href="https://newrelic.com/jp/events">https://newrelic.com/jp/events</a></p>	<p>▶受験サイト</p> <p><a href="https://learn.newrelic.com/full-stack-observability-exam-jp">https://learn.newrelic.com/full-stack-observability-exam-jp</a></p>

今回ご紹介する  
セミナー

# 2025年1-3月期開催のハンズオンセミナー

2025年1月22日(水)15:00-17:00

**NRU300 - 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/2025-01-22/nru300aib>



2025年2月5日(水)15:00-17:00

**NRU300 - Webサイト上のユーザー体験観測の基本**

<https://newrelic.com/jp/events/2025-02-05/nru300dem>

2025年2月12日(水)15:00-17:00

**NRU300 - ダッシュボード開発とNRQLの基本**

<https://newrelic.com/jp/events/2025-02-12/nru300dqj>

2025年2月26日(水)15:00-17:00

**NRU300 - アラート設計の基本と活用**

<https://newrelic.com/jp/events/2025-02-26/nru300alt>

2025年3月19日(水) 15:00-17:00

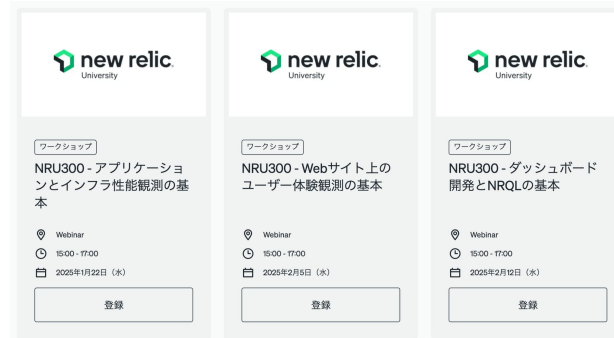
**NRU400 - IDEと連携し、問題解決を加速するNew Relic活用の実践**

<https://newrelic.com/jp/events/2025-03-19/nru400cds>

**【開催形式】**

**全てZoomによるウェビナー  
(参加費無料)**

こちらからも一覧でご覧いただけます！



# 新ウェビナーのご紹介

## NRU300 - Webサイト上のユーザー体験観測の基本

<https://newrelic.com/jp/events/2025-02-05/nru300dem>

あなたのシステム運用、インフラ監視だけで満足していませんか？近年のより複雑化したシステム運用においては、ユーザー体験を観測し改善することが品質の維持、改善の鍵となります。本ウェビナーでは New Relicを用いたユーザー体験の観測がいかに簡単で効果的かをハンズオンで学ぶことができます。

### このハンズオントレーニングでカバーするトピック

- ユーザー体験観測の重要性
- RUM(Real User Monitoring)と外形監視の概要
- New Relic BrowserによるRUMの実践
- New Relic Syntheticsによる外形監視の実践

### こんな方におすすめ

- New Relic を知ってはいるがまだ触ったことがないエンジニアの方
- New RelicのAPMやInfrastructureは触ったことがあるが、BrowserやSyntheticsは使っていないエンジニアの方



# 2025年1-3月期開催の資格取得向けセミナー

## New Relic Full Stack Observability Practitioner 認定試験対策講座

開催日程:2025年3月5日(水)15:00-17:00

開催形態:Zoomを使用したオンラインの座学形式

参加費:無料

▼お申し込みはこちらから

<https://newrelic.com/jp/events/2025-03-05/nrufsoexm>

※試験対策講座の内容をより理解いただき、実践力を身につけていただくために、後述のラーニングパスにそった学習を推奨しています

**後述の認定試験の合格に向けた最短の道になります！**

# Observabilityのスペシャリストを目指せ！ 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】

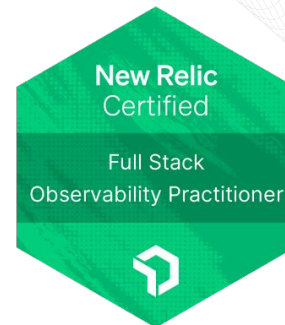
- Observabilityの実現のためにNew Relicが取得するデータの理解と、目的に応じたデータ分析やアラート設定
- バックエンドおよびフロントエンドの問題発見とトラブルシューティング

## 【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

## 【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

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)

New Relic  
ブログ

お客様事例講演  
※作成中

お客様事例紹介

New Relic  
セミナー

初めての方向け

日本語で提供

英語でのみ提供

# お知らせ

NRU300

# 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 screenshot shows the New Relic sign-up page. The header includes the New Relic logo and navigation links. The main heading is 'New Relic 無料サインアップ'. Below this, there is a sub-heading '仮登録' (Temporary Registration) and a note: '無料アカウントでは以下の内容が含まれています: 1名フルユーザーアクセス、100GB/月のデータ保存容量'. The form includes fields for 'Name', 'Company Email', and 'Password'. A 'Start Now' button is visible at the bottom right of the form area.

## 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 screenshot shows the 'Create your account' page. It features the New Relic logo and the heading 'Create your account.'. There are input fields for 'Name' and 'Company Email', followed by a 'Start Now' button. To the right, there are three bullet points: 'Access to everything', 'Scale as you grow', and 'Data in one place'. A large blue arrow points from this page towards the right.

## ④ パスワード設定

The screenshot shows the 'Create your password' page. It includes the New Relic logo and a confirmation message: 'We verified your email'. The heading is 'Create your password'. Below this, there is a note: 'Use at least 8 characters with at least 1 letter and 1 number or special character.'. There are input fields for 'Email' (pre-filled with 'scarlet5.ht@gmail.com') and 'Password'. A 'Save password' button is at the bottom.

## ⑤ データ保存先を選択

The screenshot shows the 'Tell us where to store your data' page. It includes the New Relic logo and the heading 'Tell us where to store your data'. Below this, there is a note: 'The data center doesn't have to match your physical location.'. There is a 'Data region' section with two options: 'United States' (selected with a checkmark) and 'Europe'. 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.**  
NRU300

