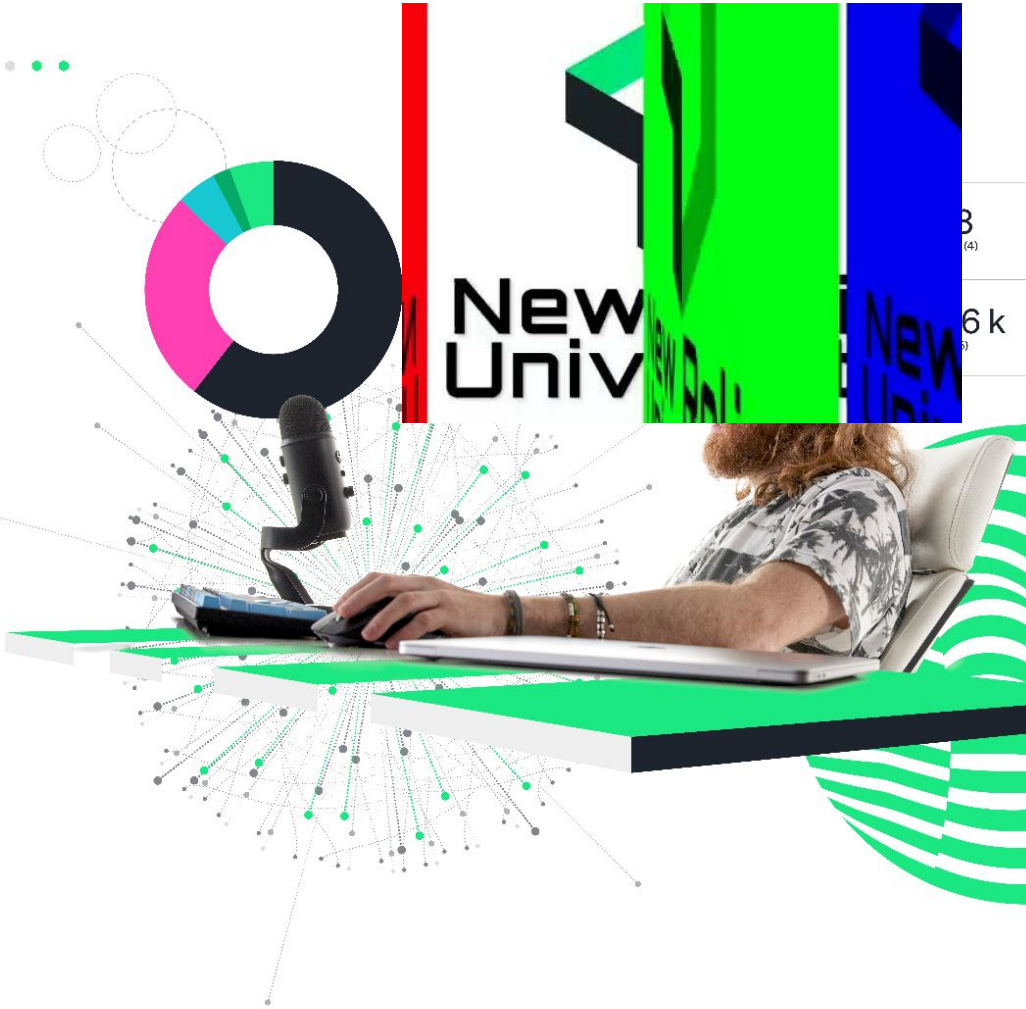
new relic

ダッシュボード開発とNRQLの基本編

NRU 302 - Dashboard / NRQL



ウェビナー　各種ご連絡

1. ご質問がある場合は、”Q＆A”からご入力ください。

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

Safe Harbor

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

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

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

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

Yoshikazu Okawa

Cloud Architect

for B2C Industry

Infrastructure, Google Cloud

New Relic

Technical Account Manager

Google Cloud Partner Top Engineer '2021

new relic

本ウェビナーの受講想定者

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

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

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

本日のゴール

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

本日のタイムテーブル

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

ハンズオンで使用するブラウザについて

⚠️ ブラウザは下記のいずれかをご利用ください。

- Chrome
- Firefox
- Edge

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

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

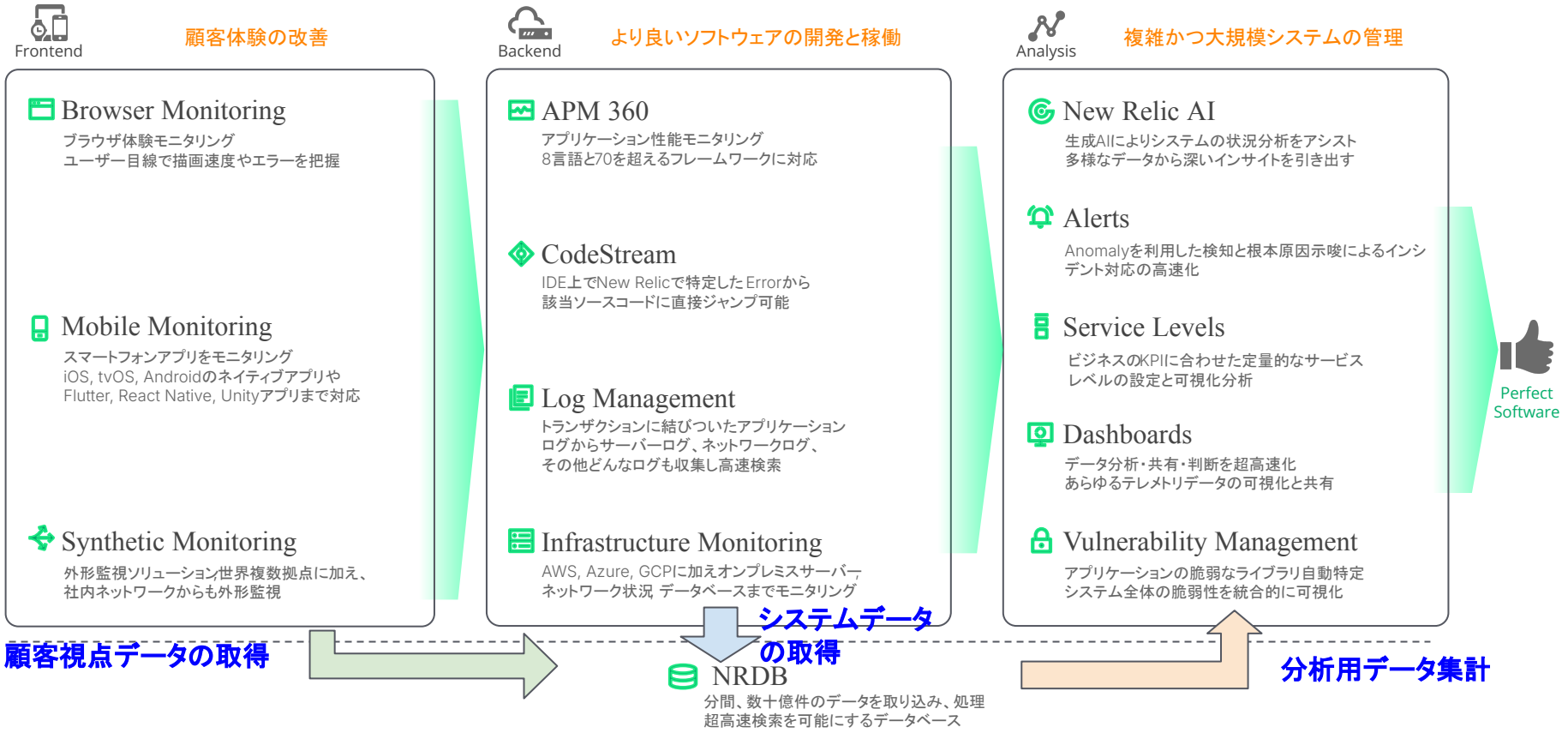
参考: New Relic UIの対応ブラウザ
https://docs.newrelic.com/jp/docs/new-relic-solutions/get-started/supported-browsers-new-relics-ui/

New Relicとデータ分析

NRU302 - 座学

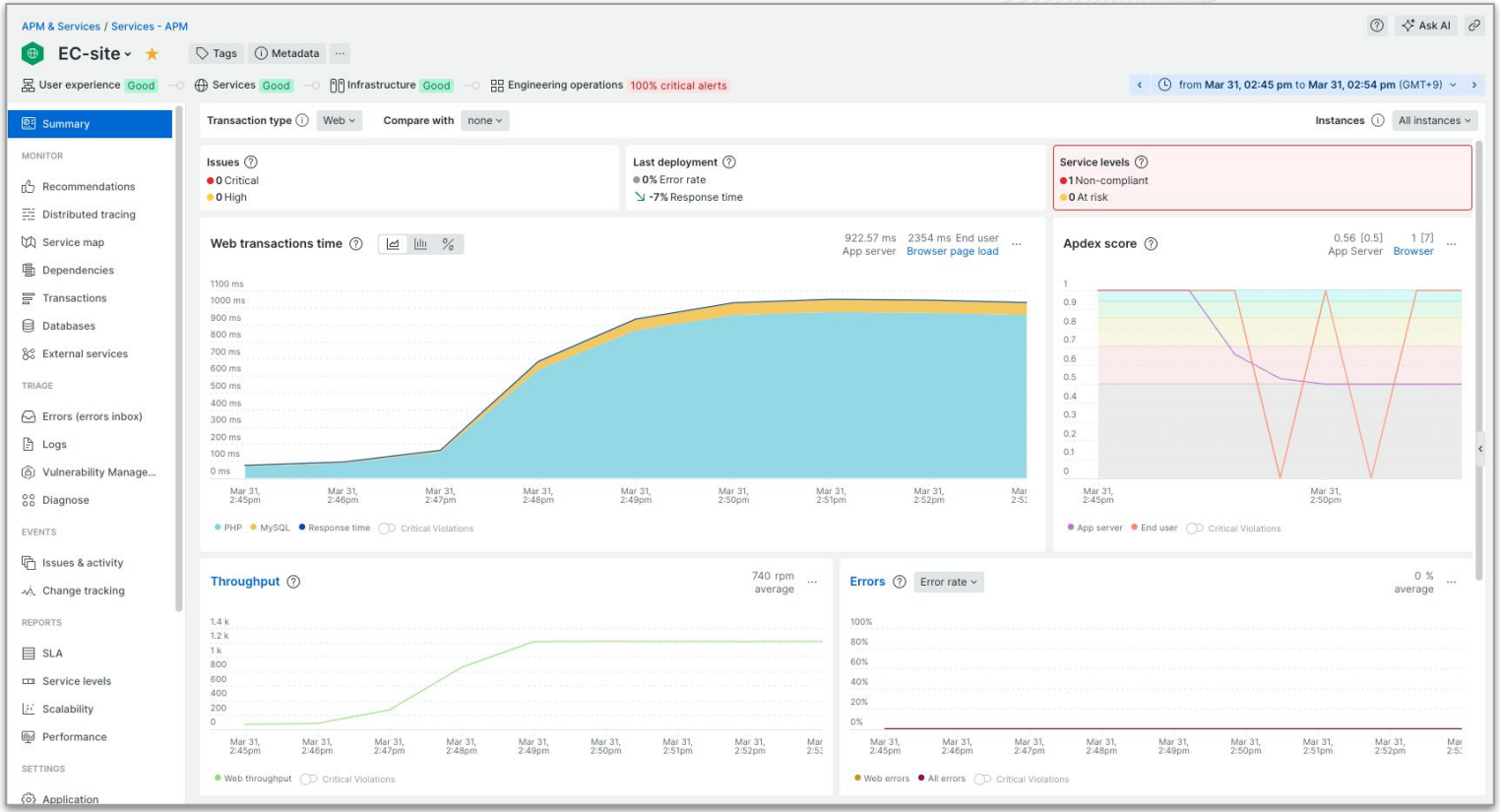
new relic

new relic オブザーバビリティプラットフォーム 全体像



new relic

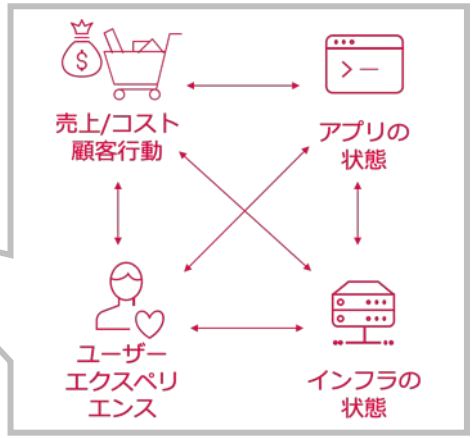
ビルトインのUIでもデータ分析はできます



new relic

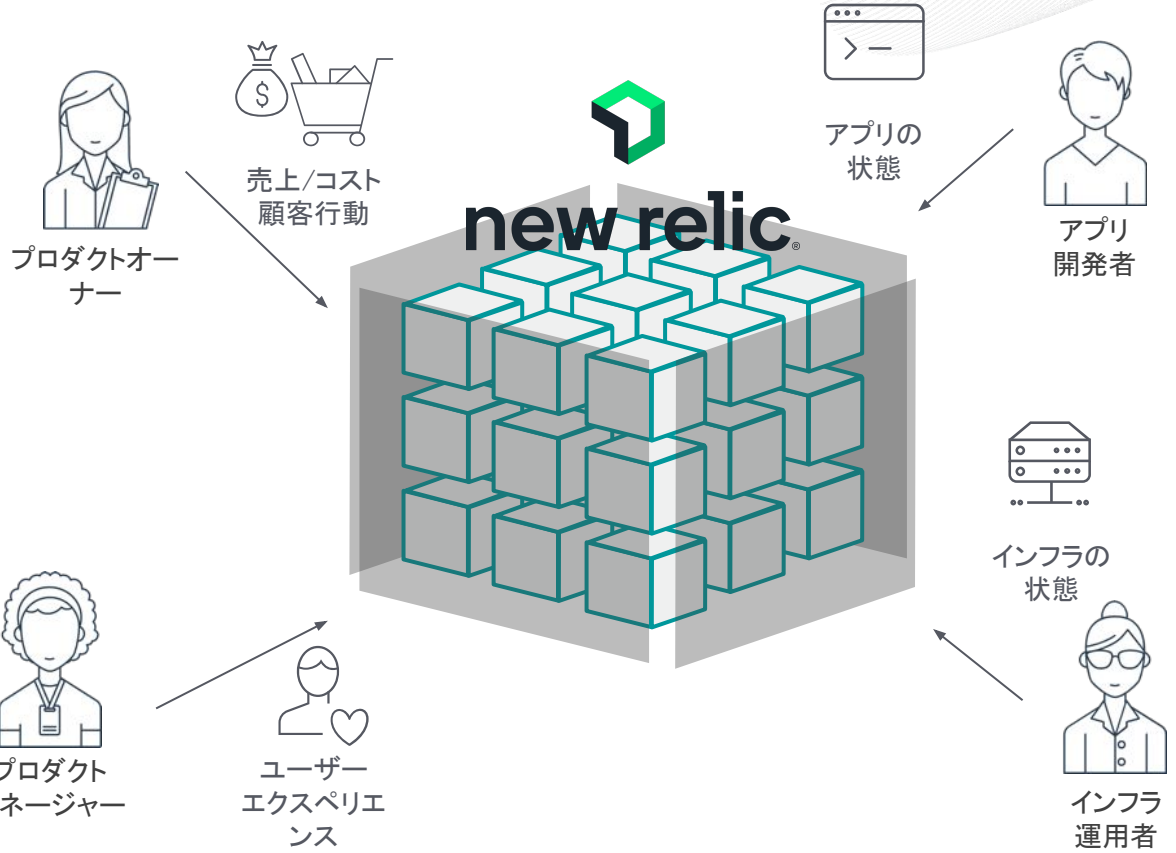
・・・が、人によってみたい観点は全く異なる

計測したいことは密接に関わり合っている



new relic

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

守り

攻め

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

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

STEP1

データの
理解

New Relicが取得するデータ

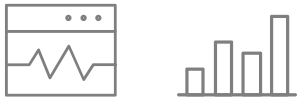
NRU302 - 座学

new relic

New Relicが取得するデータ: MELT

M:メトリック

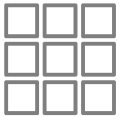
集計可能なデータの粒



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

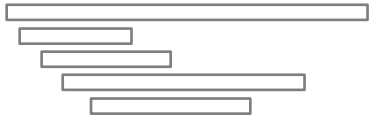
L:ログ

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

単一ト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(Attirbute)** から成る構造化データ
- New Relicが扱うデータの中核であり、以下の2つの手段で収集される
  - 各エージェント(APM,Browser,Mobile,Synthetic,Infrastructure)から送信
  - Event APIでカスタムデータを送信
    - https://docs.newrelic.co.jp/jp/docs/data-apis/ingest-apis/event-api/introduction-event-api/

- Eventデータについて
  - https://docs.newrelic.co.jp/jp/docs/data-apis/understand-data/new-relic-data-types/

Event名について

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

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

参考: New Relic の機能によって報告されるデフォルトのイベント
https://docs.newrelic.com/jp/docs/data-apis/understand-data/event-data/default-events-reported-new-relic-products/

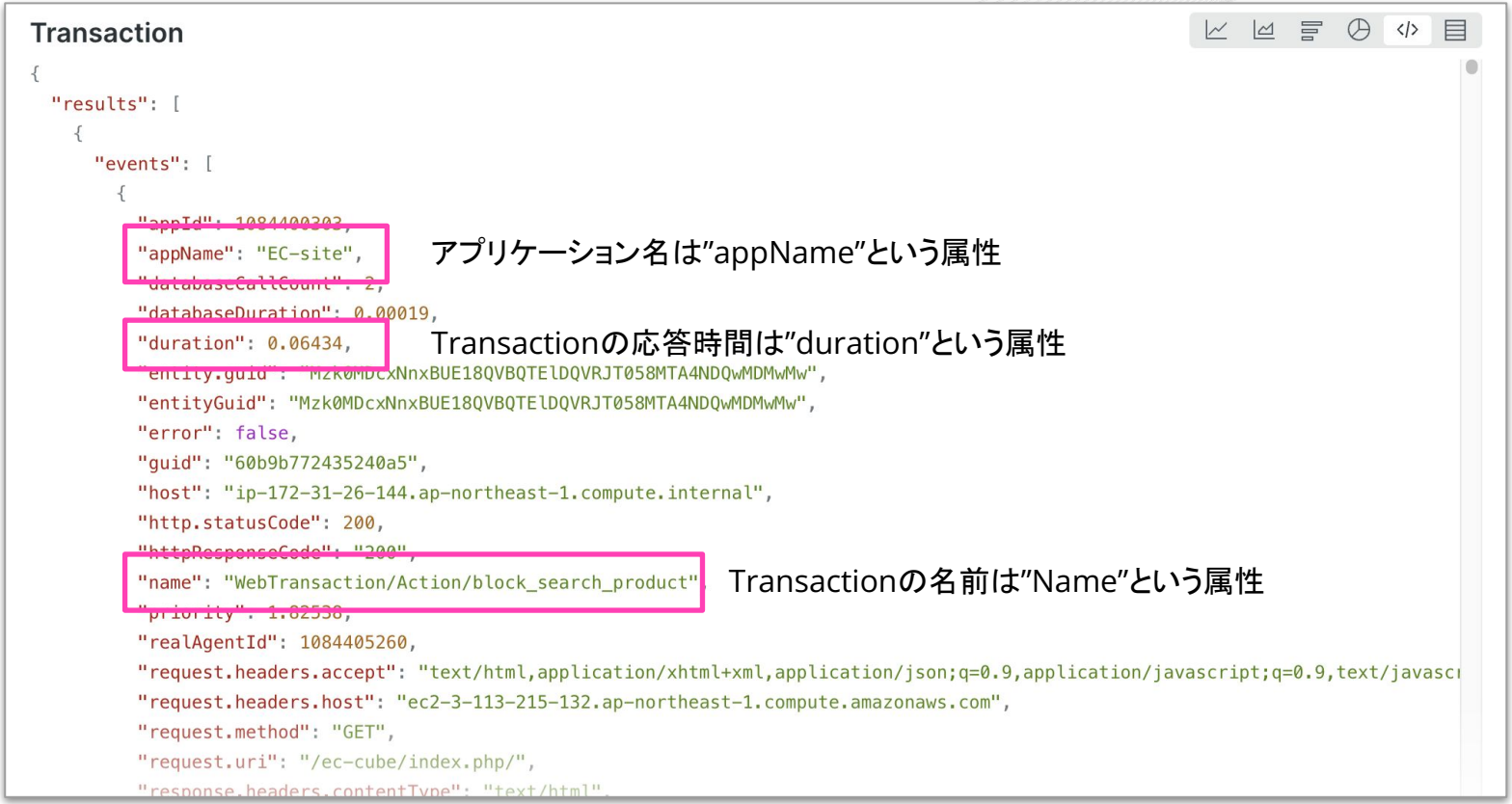
属性(Eventの構成要素)について

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

参考: カスタム属性の収集
https://docs.newrelic.co.jp/jp/docs/data-apis/custom-data/custom-events/collect-custom-attributes/

属性(Eventの構成要素)について

Transaction




```
{
  "results": [
    {
      "events": [
        {
          "appId": 1084400303,
          "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": 1.82538,
          "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",
```

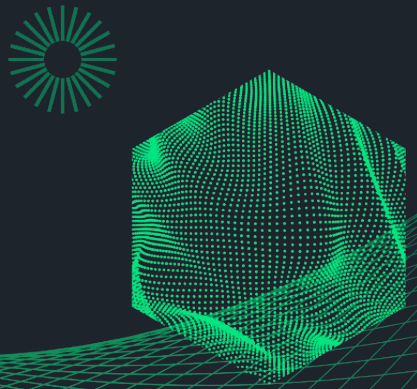

アプリケーション名は”appName”という属性

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

NRU302 - ハンズオン #1



ハンズオン - データの理解

このハンズオンでは、以下の点を学習します。

- New Relic プラットフォームにログインする
- New Relic の様々なUIに触れてみる
- Metrics & Events を活用して、収集したEventデータを参照する

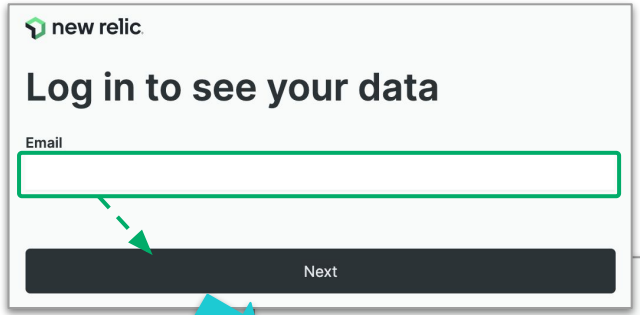
Option!!
- カスタムイベントを送信し、参照する

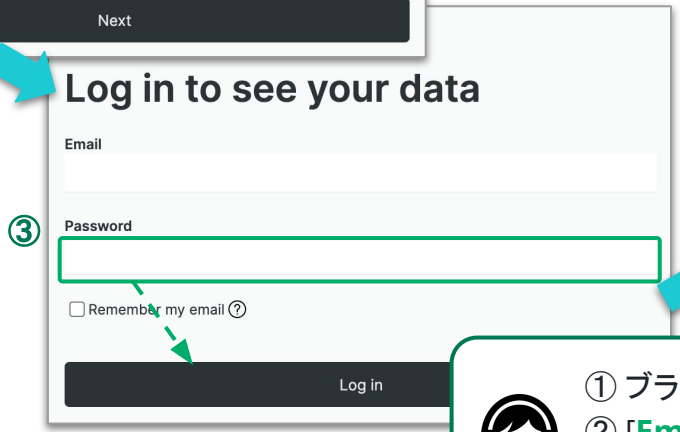
15:20 - 15:40 (20min)

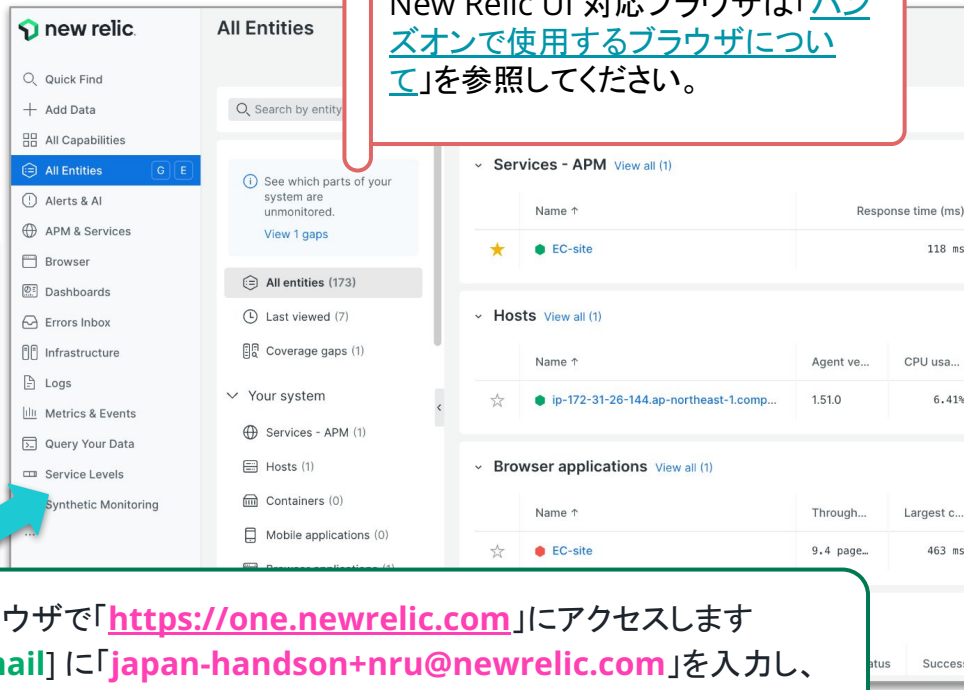
p. 25 - p. 37

New Relic へログイン

New Relic ポータル にログインします







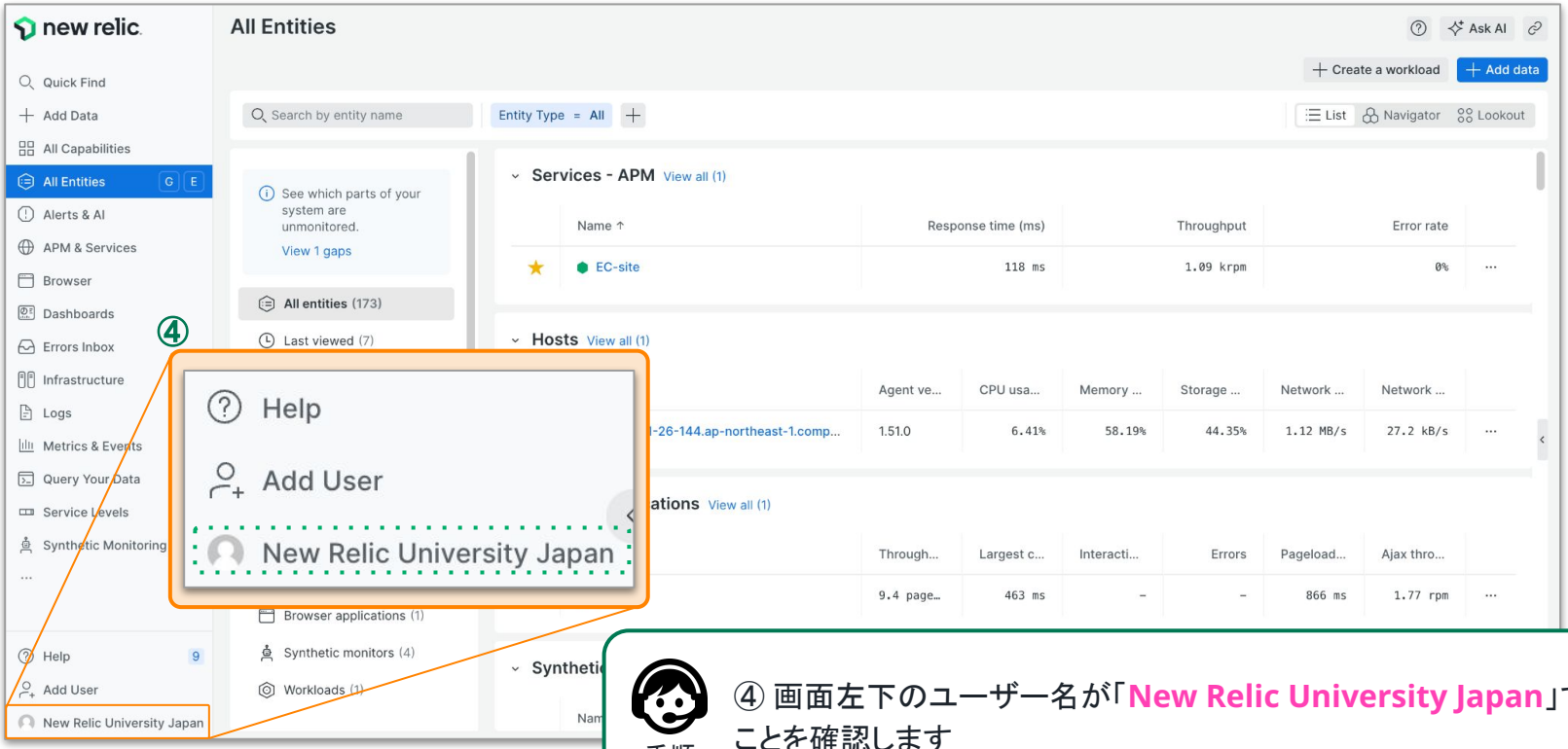
New Relic UI 対応ブラウザは「ハンズオンで使用するブラウザについて」を参照してください。

手順

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

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

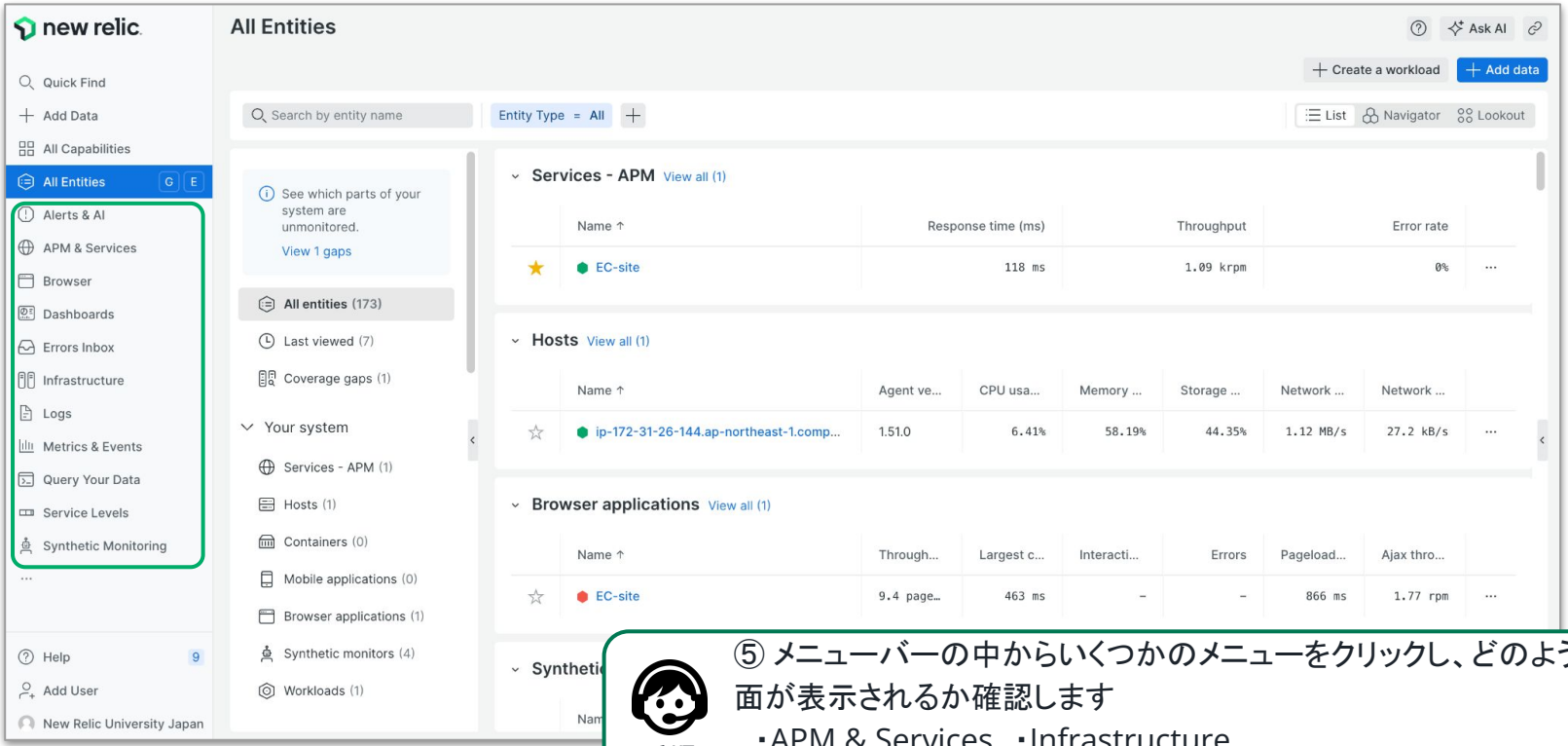


手順

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



⑤

手順

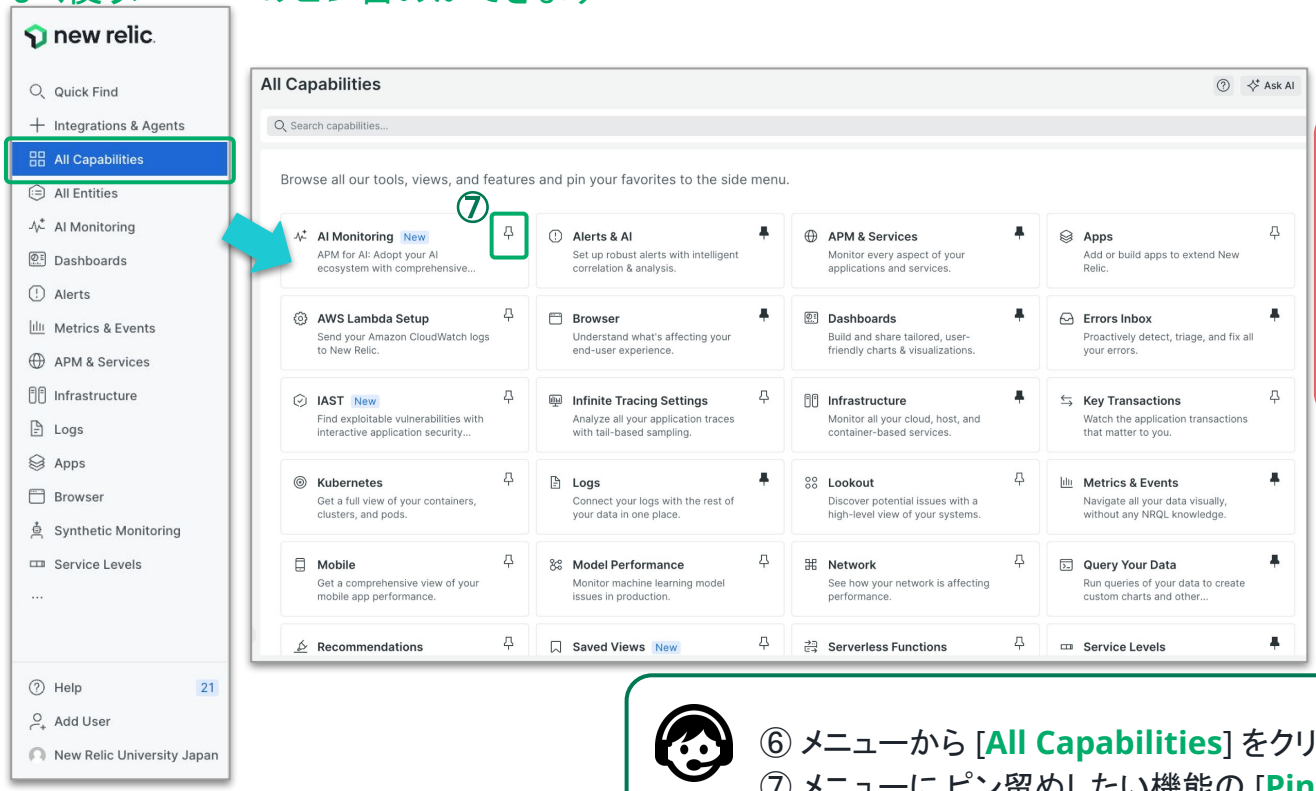
⑤ メニューバーの中からいくつかのメニューをクリックし、どのような画面が表示されるか確認します

・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



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

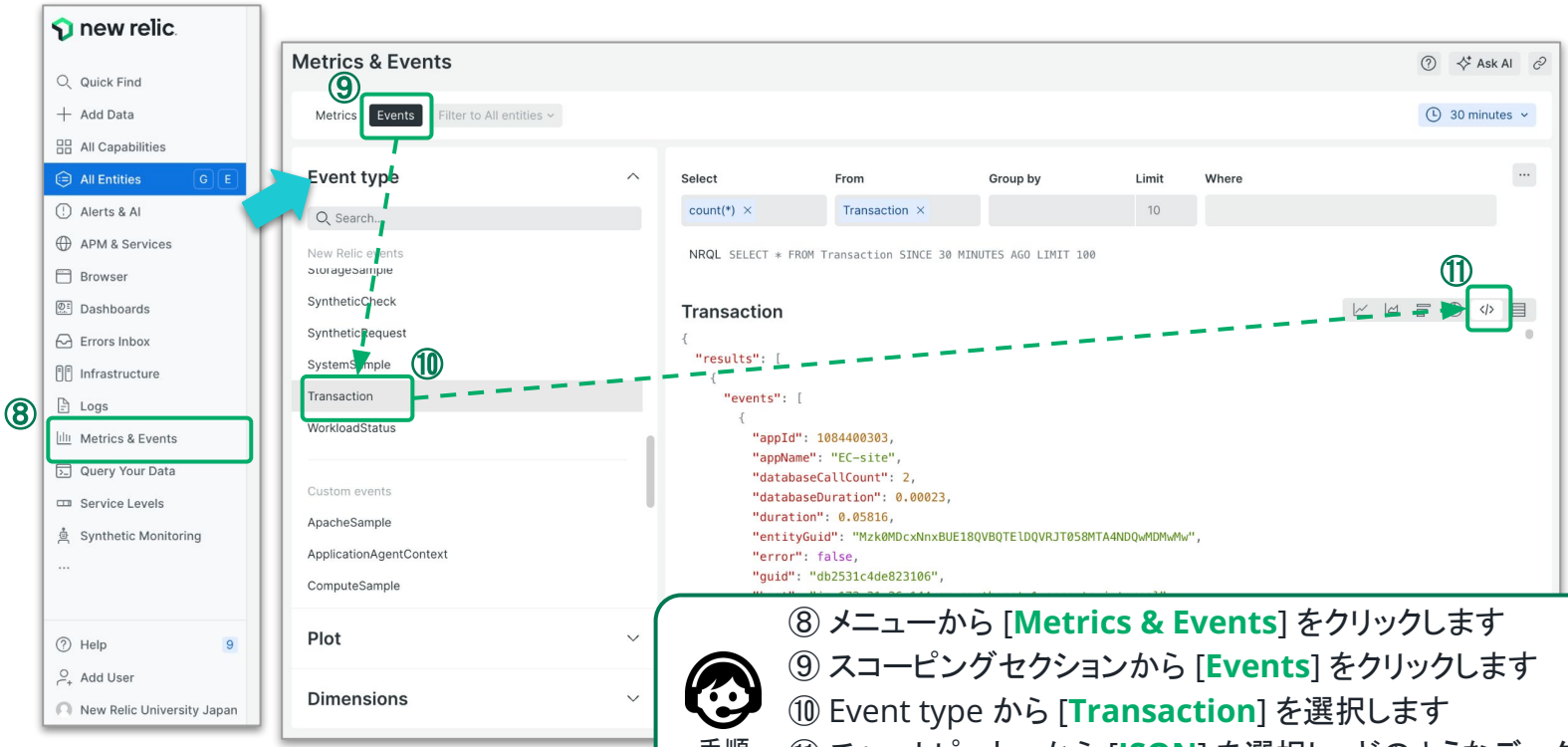
手順

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

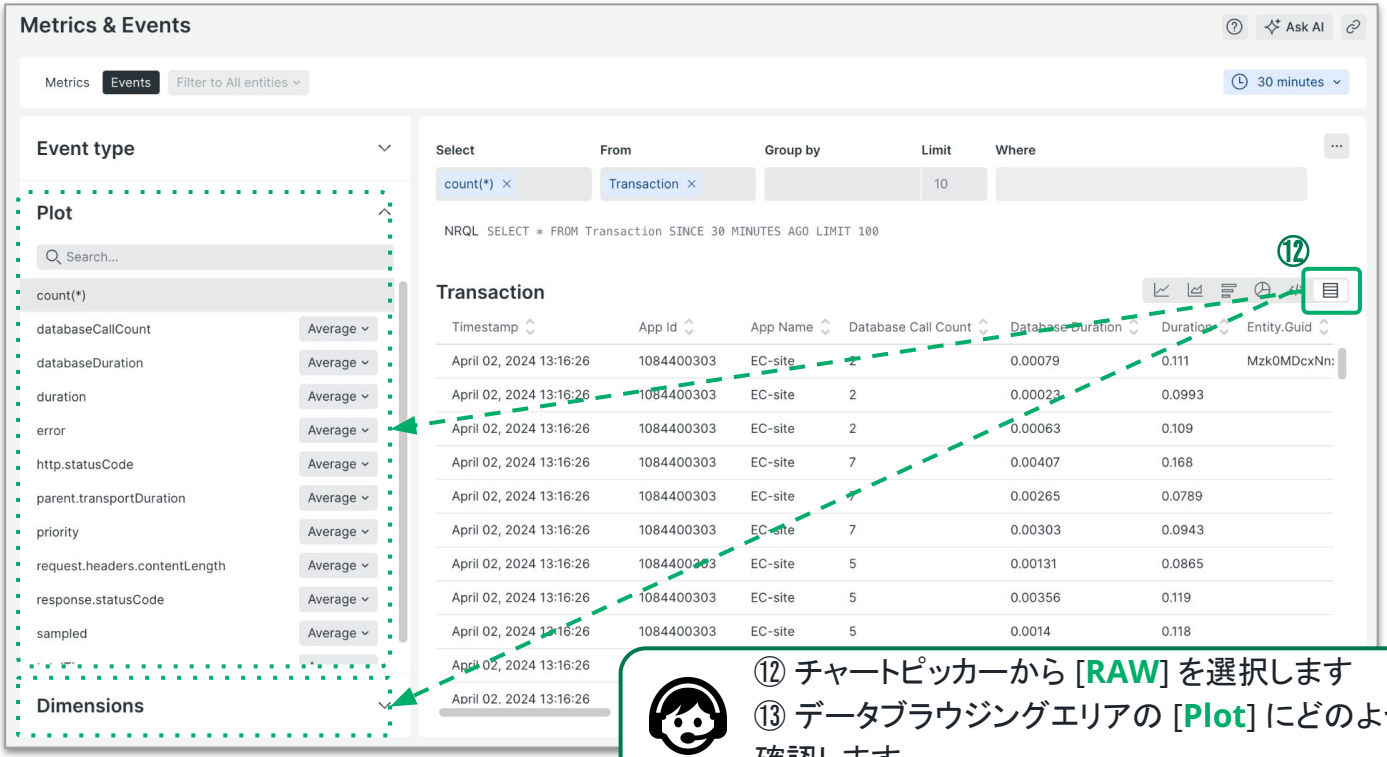


手順

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



手順

⑫ チャートピッカーから [**RAW**] を選択します
⑬ データブラウジングエリアの [**Plot**] にどのような数字属性があるか確認します
⑭ [**Dimensions**] にどのような文字属性があるか確認します

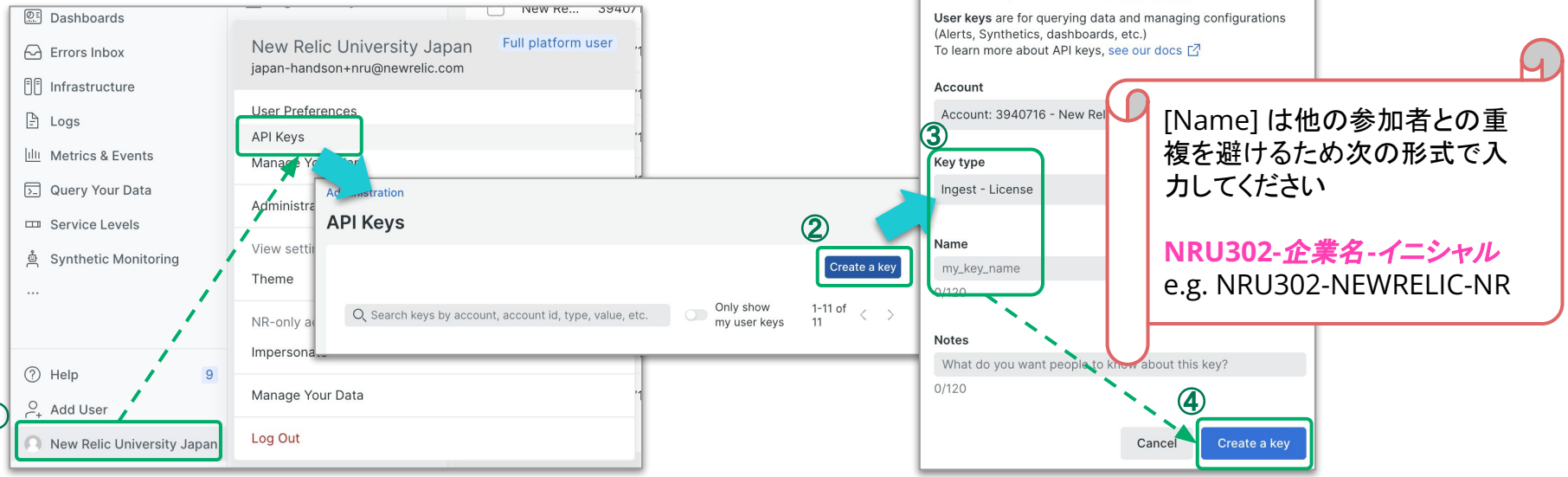
Option!!

カスタムイベントを送信する

NRU302 - ハンズオン

Option!! APIキーの作成

カスタムイベントを送信する際の APIキーを作成します



[Name] は他の参加者との重複を避けるため次の形式で入力してください

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

手順

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

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

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

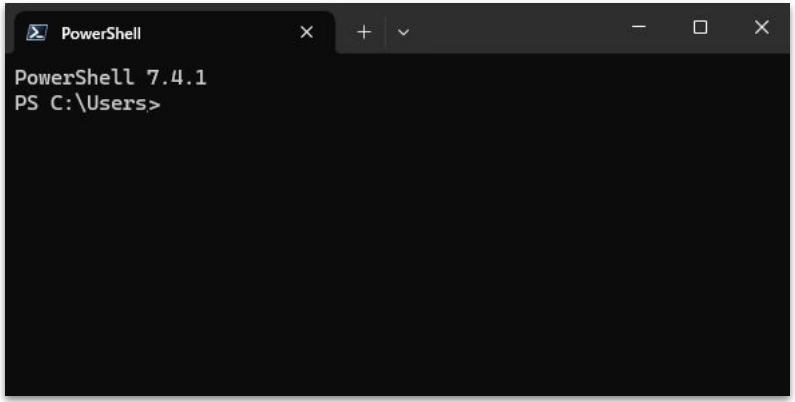
手順

⑤ 作成したAPIキー右側の [•••] をクリックします
⑥ [**Copy key**] をクリックし、APIキーをクリップボードにコピーします

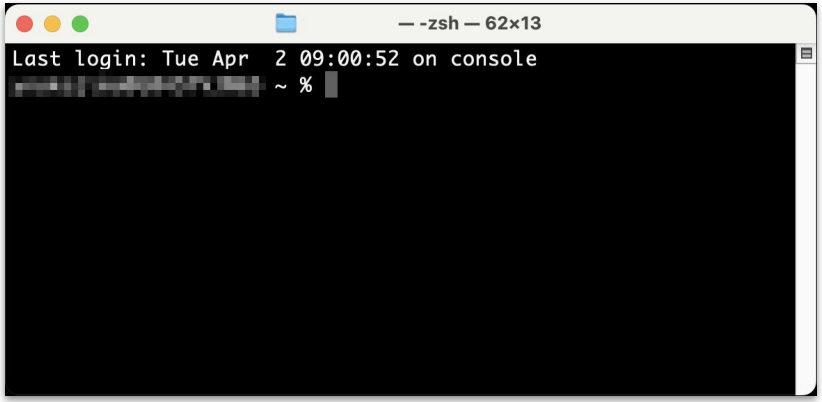
Option!! ターミナルの起動

カスタムイベントを送信するため、OSのターミナルアプリを起動します

Windows - PowerShell



Mac - ターミナル



手順

⑦ ご利用のOSのターミナルアプリケーションを起動します
Windows: PowerShell
Mac: ターミナル

Option!! カスタムイベントの送信 (Windows)

WindowsのPowerShellからカスタムイベントを送信します

GitHub - script.ps1

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

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

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

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

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

手順

⑧ 上のスクリプト内の次のパラメータを変更します
- {APIキー}: 手順⑥ でコピーしたAPIキー
- {ご自身のイニシャル} および {任意の文字列}: 任意

⑨「**script.ps1**」として保存し、PowerShellから実行します

Option!! カスタムイベントの送信 (Mac)

Macのターミナルからカスタムイベントを送信します

GitHub - nrulab.json

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

GitHub - script.sh

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

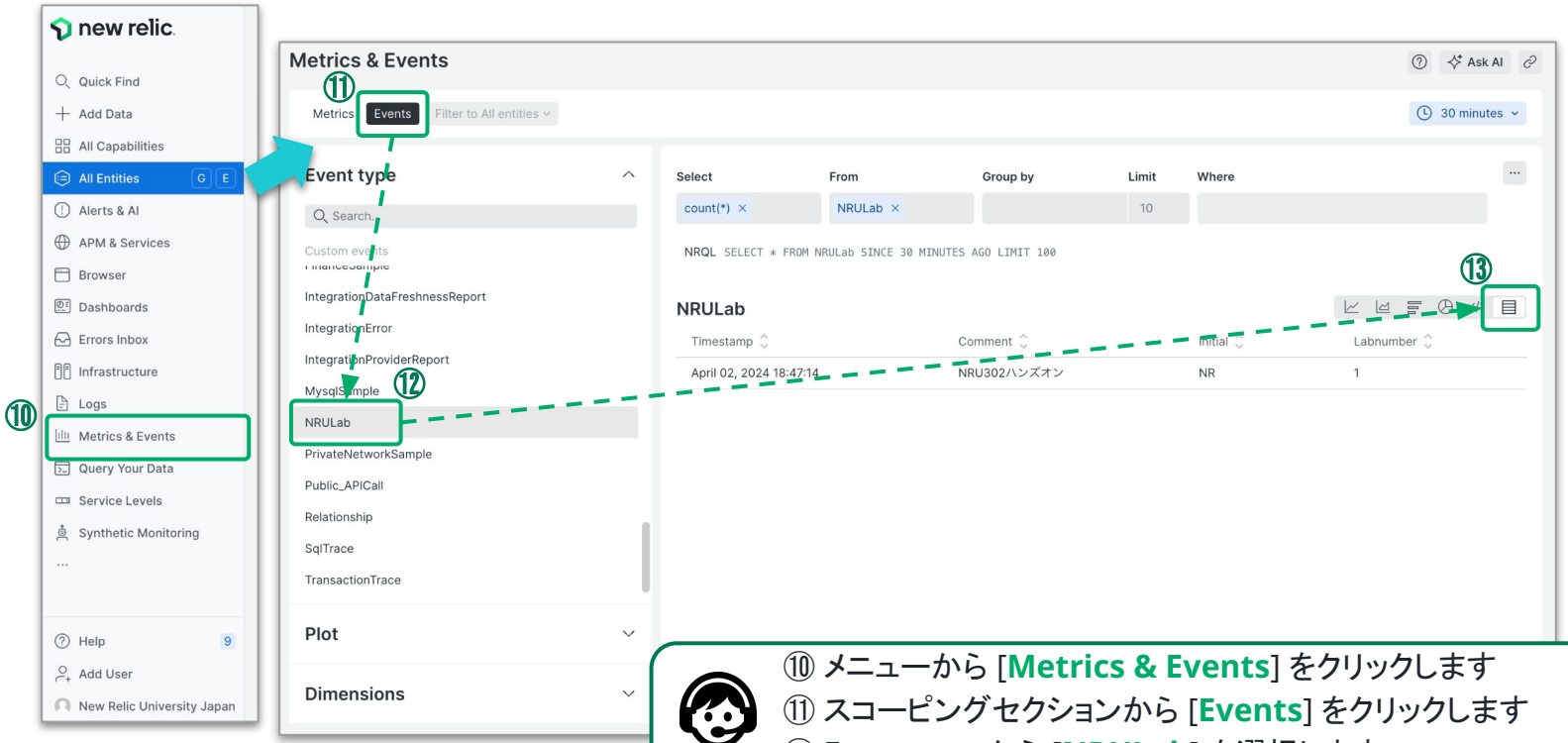
手順

⑧ 上のJSON内の次のパラメータを変更し、「**nrulab.json**」として保存します
- {ご自身のイニシャル} および {任意の文字列}: 任意

⑨ コマンド内の次のパラメータを変更し、ターミナルから実行します (nrulab.json が保存されたディレクトリで実行)
- {APIキー}: 手順⑥ でコピーしたAPIキー

Option!! カスタムイベントの確認

カスタムイベントが正しく送信されていることを確認します



手順

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

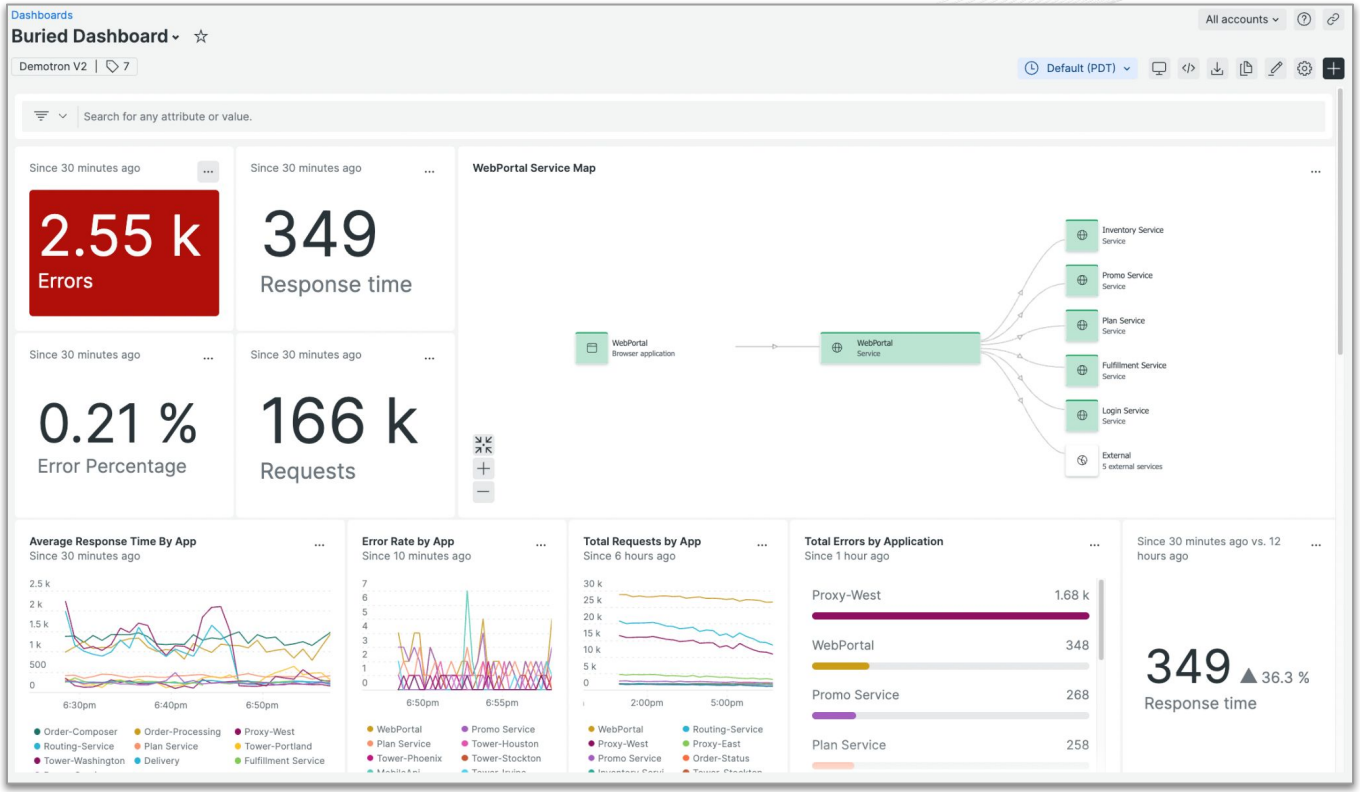
STEP2

ダッシュボードの作成

New Relic ダッシュボード

NRU302 - 座学

New Relic ダッシュボード



new relic

ダッシュボード活用シーン

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

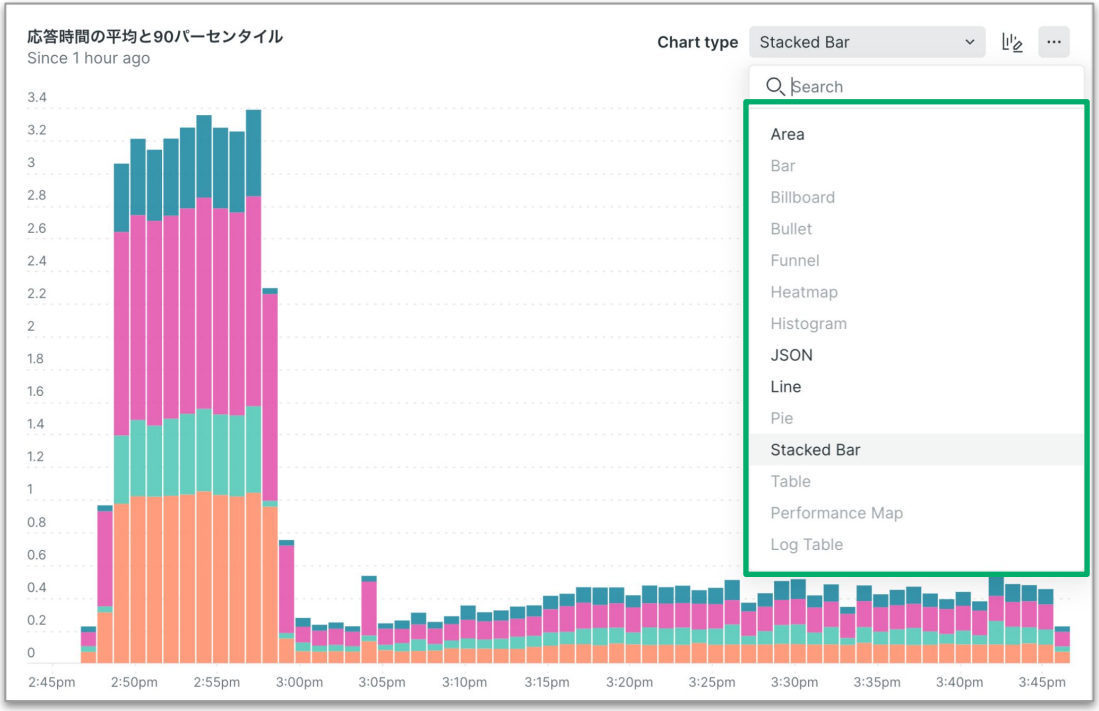
など

ダッシュボードが得意とすること

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

データのビジュアライズ

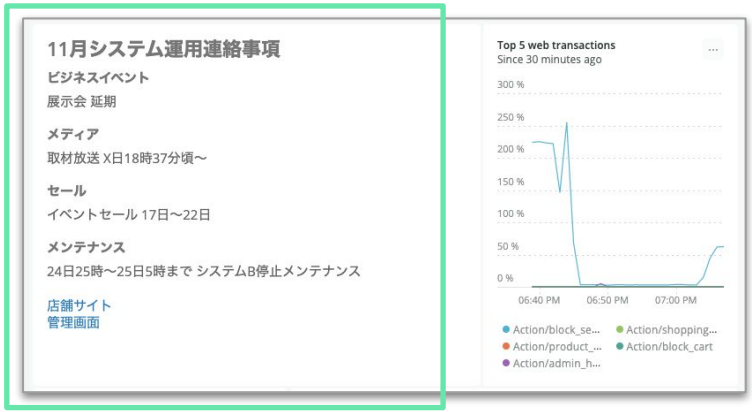
- 加工したデータを様々なチャートタイプで表示



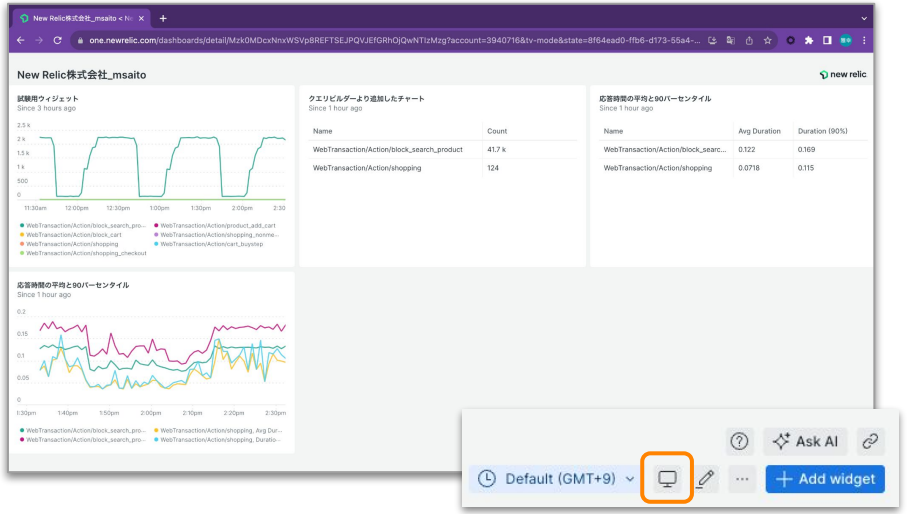
ダッシュボードの便利機能 ①

- 表示系

Note機能
任意の文字や画像をダッシュボードに埋め込む



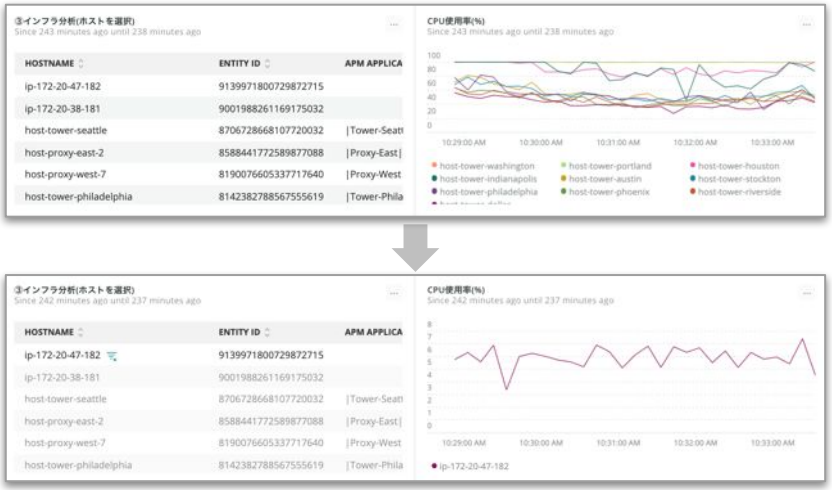
TVモード
全画面表示にする
（オフィスのディスプレイに表示するなど）



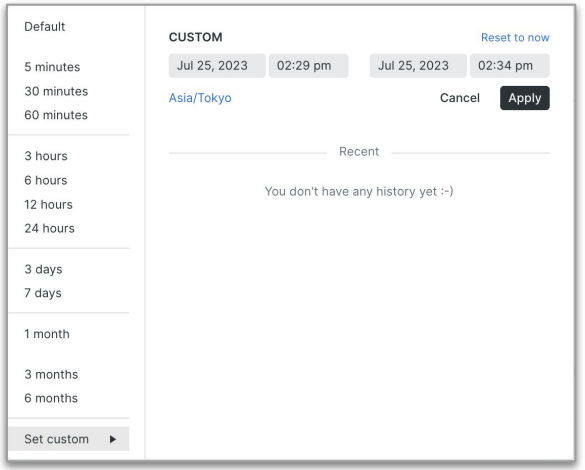
ダッシュボードの便利機能②

- 分析系

フィルタ機能
あるチャートで選んだ要素
に基づいて他のチャートの
情報が絞り込まれる

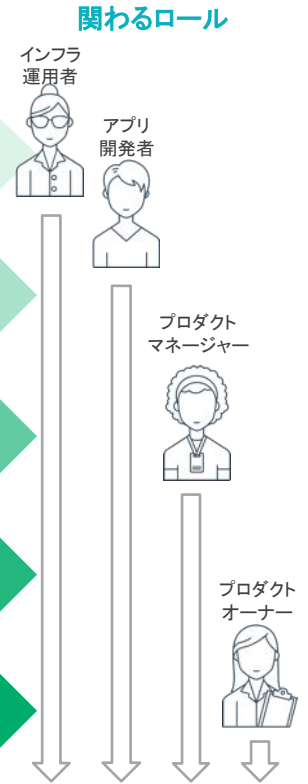


時間指定
任意の時間を指定する
(チャートからドラッグで指定するこ
とも可能)



New Relic オブザーバビリティ 成熟モデル

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



new relic

目的に応じたダッシュボード

1 受動的

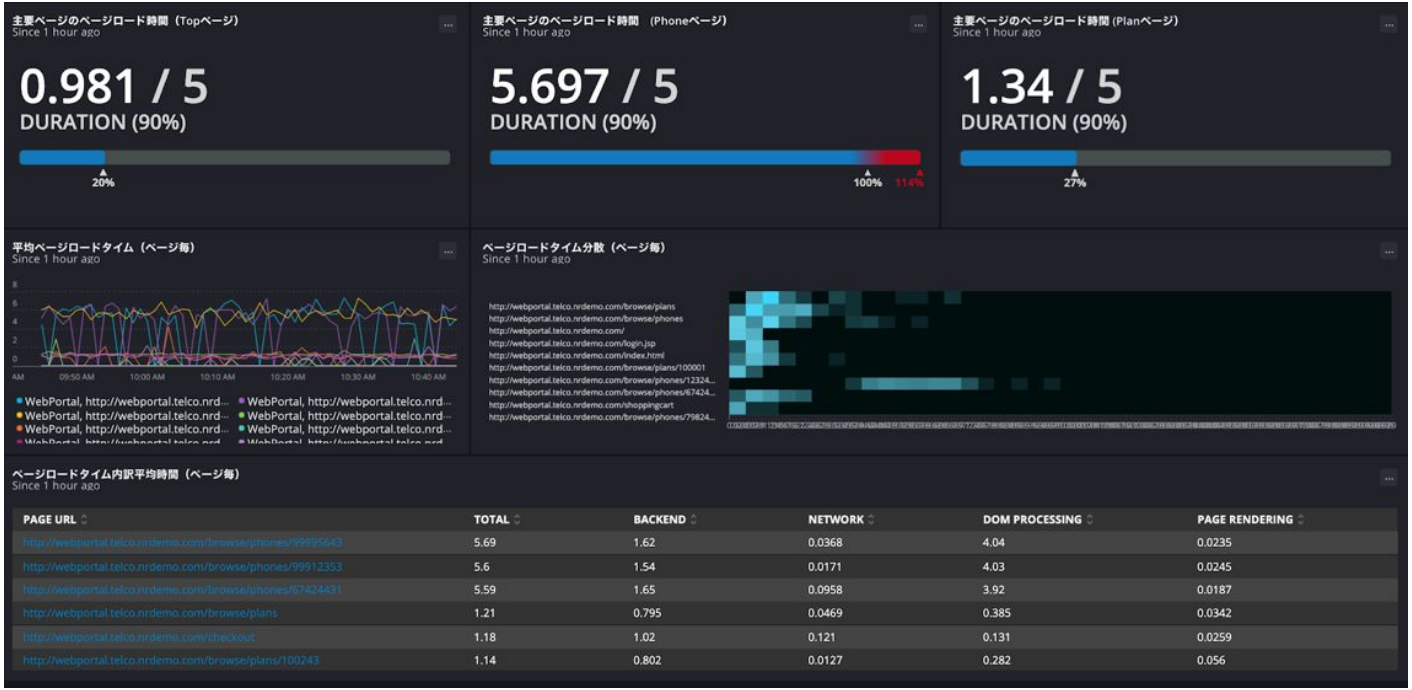
例1: サーバー稼働状況

new relic

目的に応じたダッシュボード

2 積極的

例2: ユーザー体験のリアルタイムモニタリング

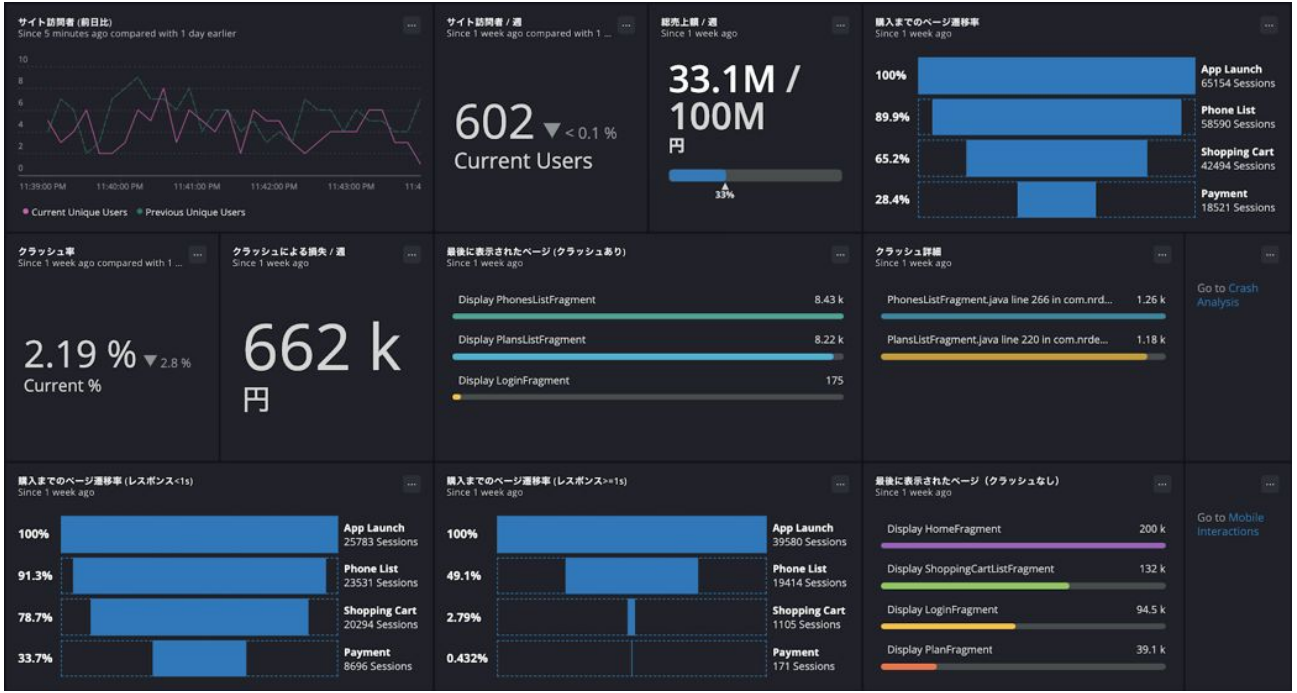


new relic

目的に応じたダッシュボード

3 予測的

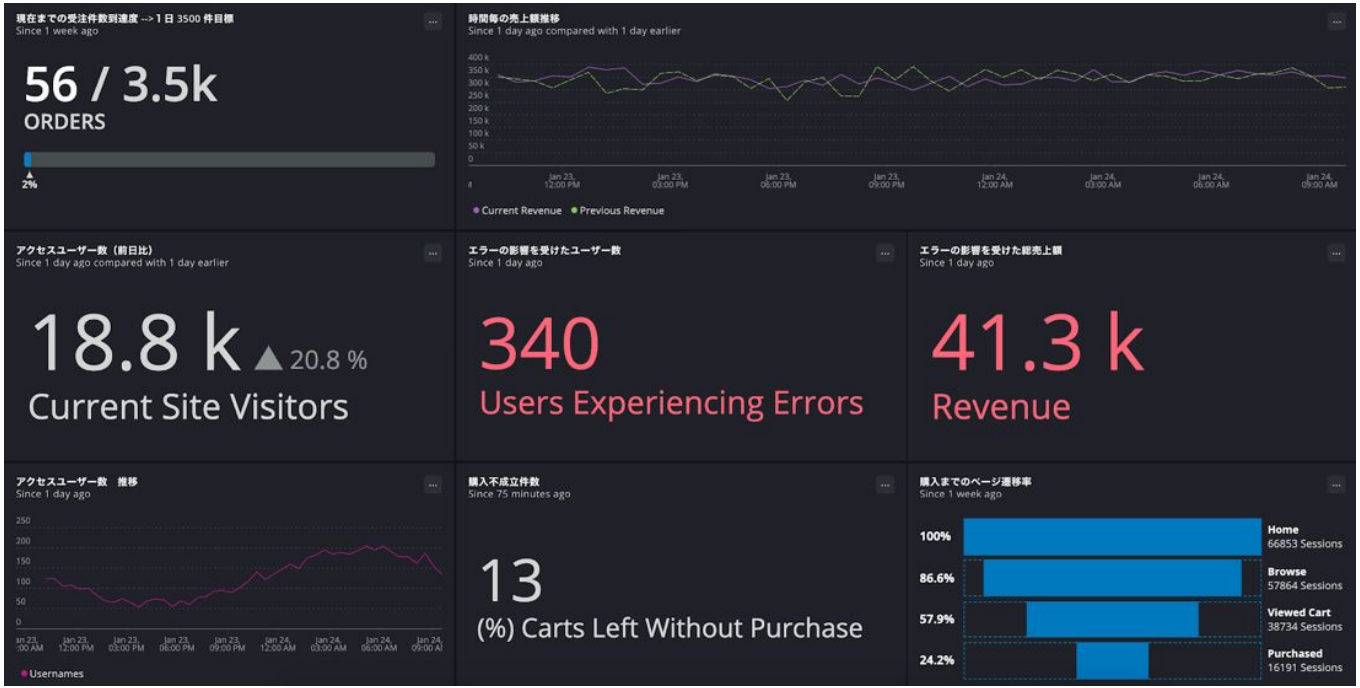
ECサイト購買分析



目的に応じたダッシュボード

4 データドリブン

例4: 売上データと連動したビジネスダッシュボード



さらに高度なビジュアライズも

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

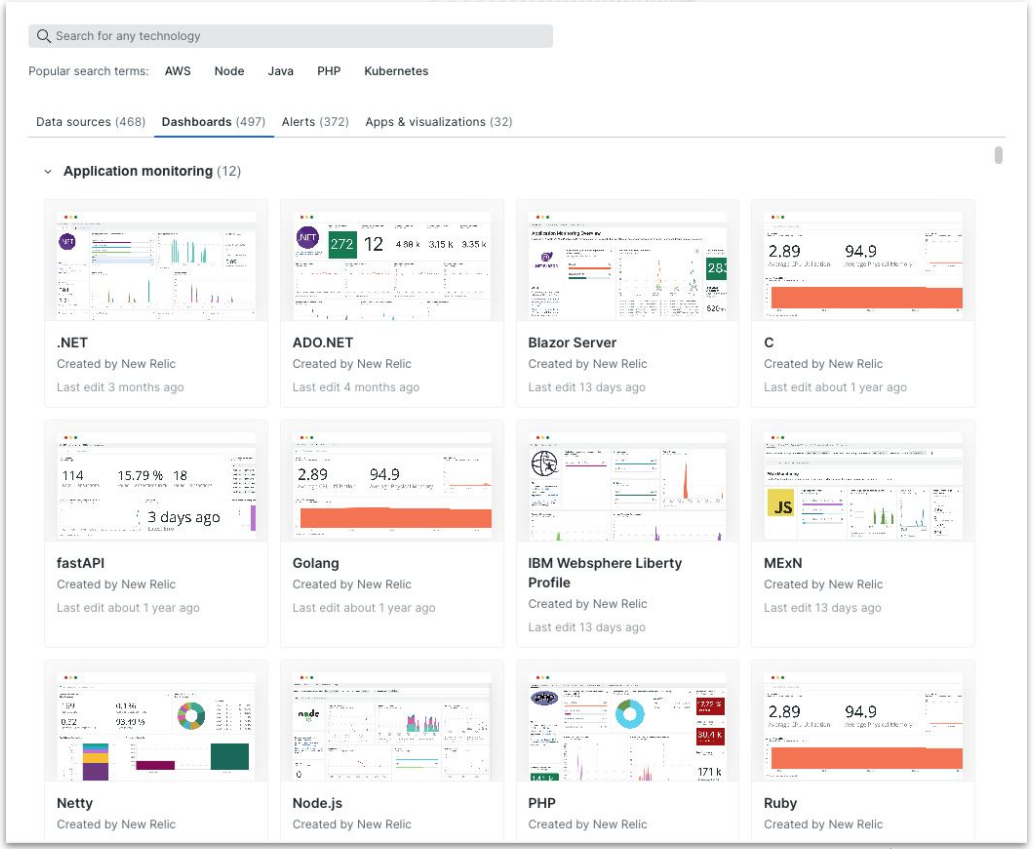
環境をセッティングして開発

オリジナルのダッシュボードを作成可能

Pre-Build Dashboards

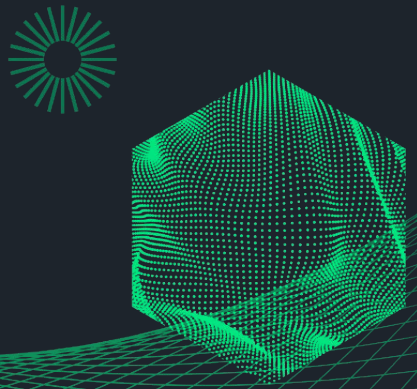
New Relicがあらかじめ準備している
ダッシュボードがあります。

- 言語
- ミドルウェア
- クラウド



ダッシュボードの作成

NRU302 - ハンズオン #2



ハンズオン - ダッシュボードの作成

このハンズオンでは、以下の点を学習します。

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

Option!! - NOTE機能を利用する

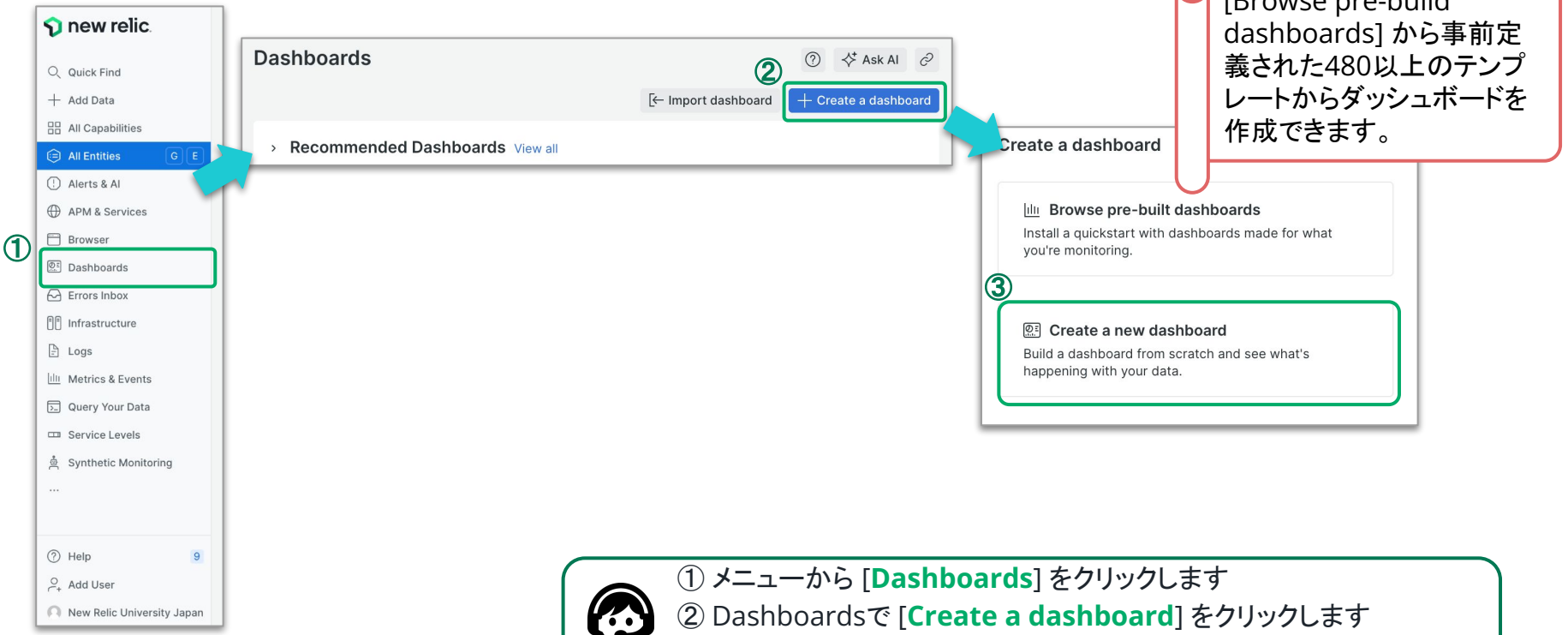
Option!! - チャートのサイズや配置を変更する

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

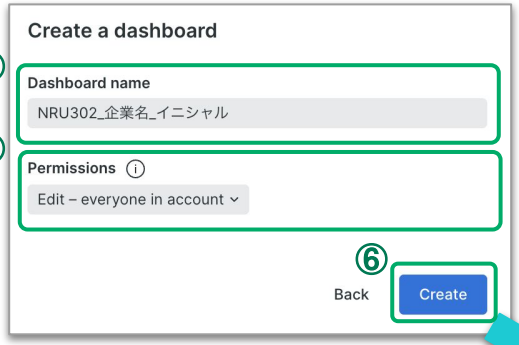


手順

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

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



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

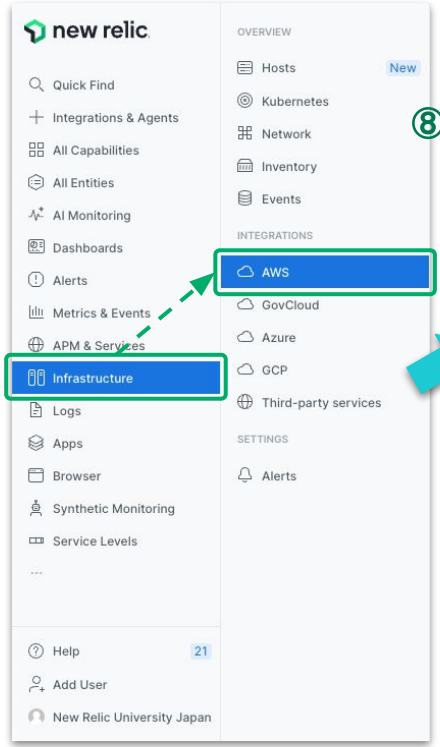
NRU302-企業名-イニシャル
e.g. NRU302-NEWRELIC-NR

手順

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

Amazon S3 Dashboardの確認

既存チャートを Amazon S3 Dashboard から選択します



手順

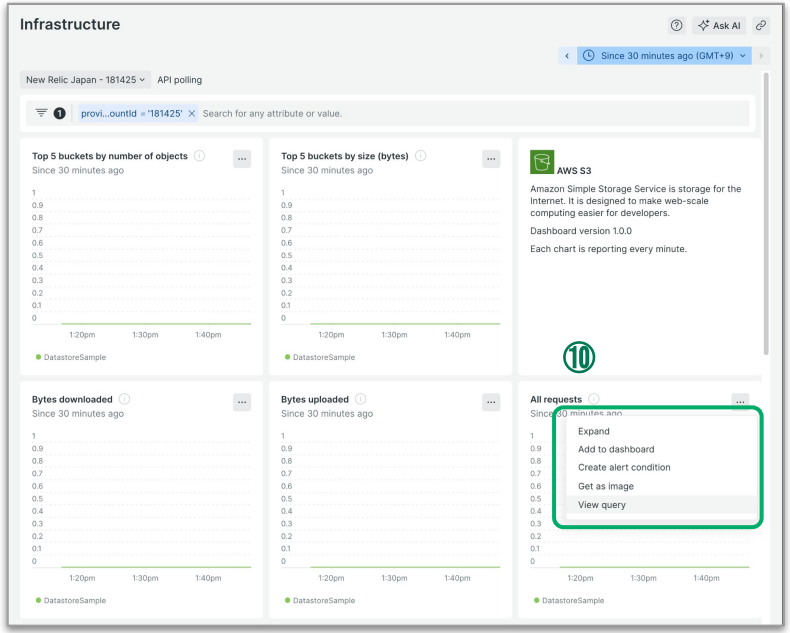
⑦ メニューから [**Infrastructure**] - [**AWS**] をクリックします

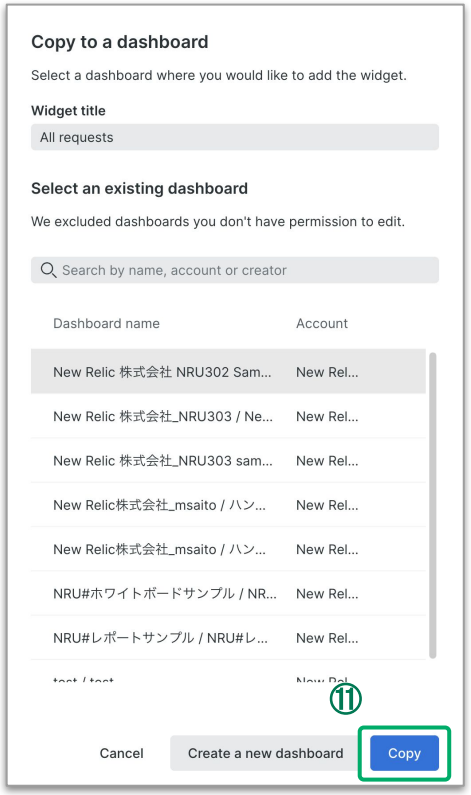
⑧ [**Select a provider account**] から「**New Relic Japan - 181425**」を選択します

⑨ [**S3 Dashboard**] をクリックします

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





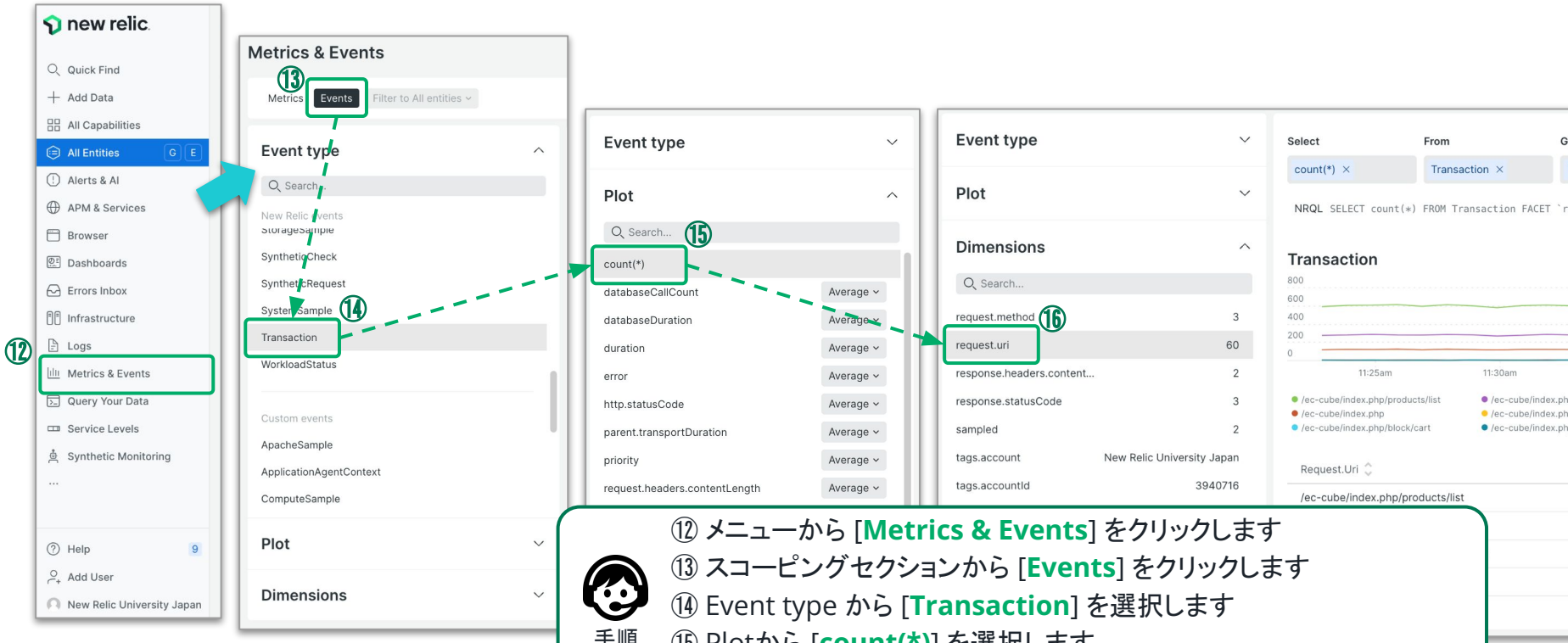
手順

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

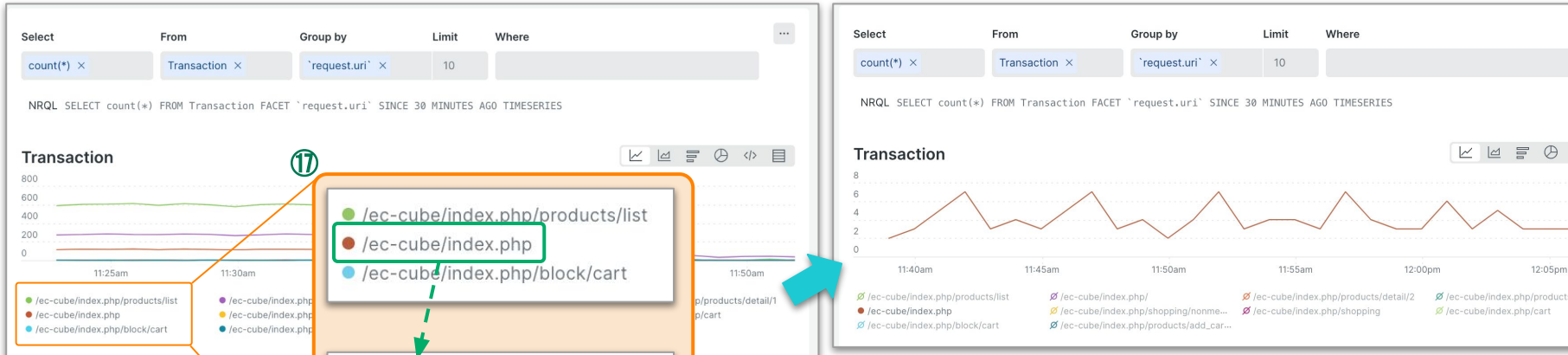


手順

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.url**] 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

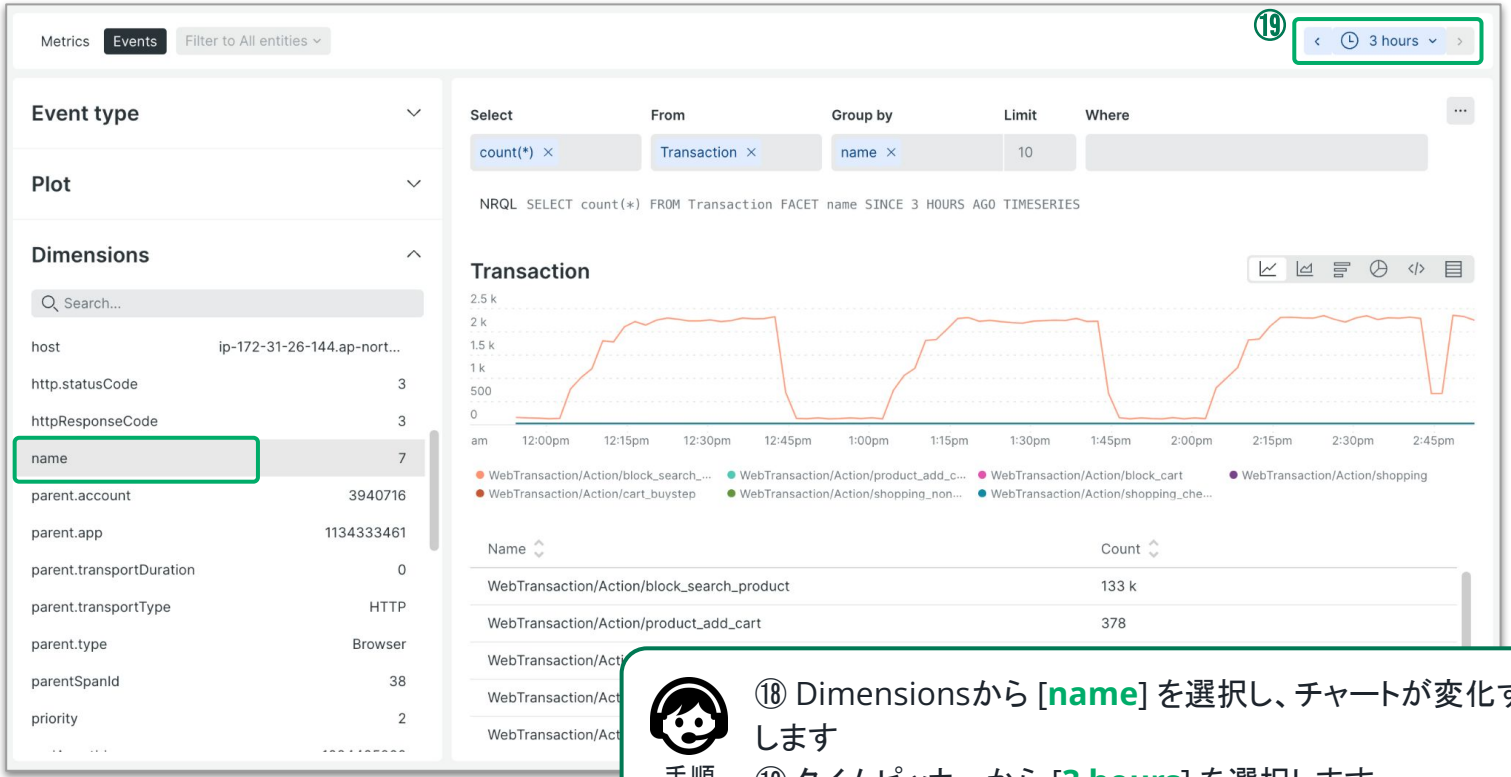


手順

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



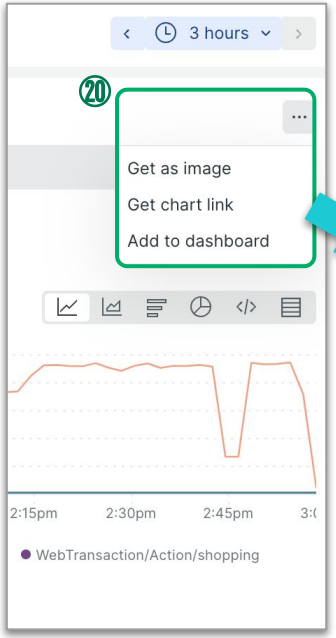
手順

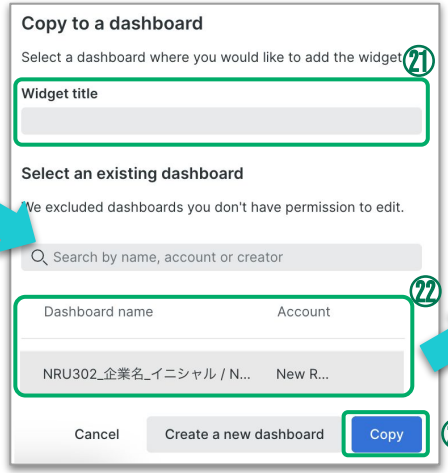
⑱ Dimensionsから [**name**] を選択し、チャートが変化することを確認します

⑲ タイムピッカーから [**3 hours**] を選択します

チャートの追加

ご自身のダッシュボードにチャートを追加します





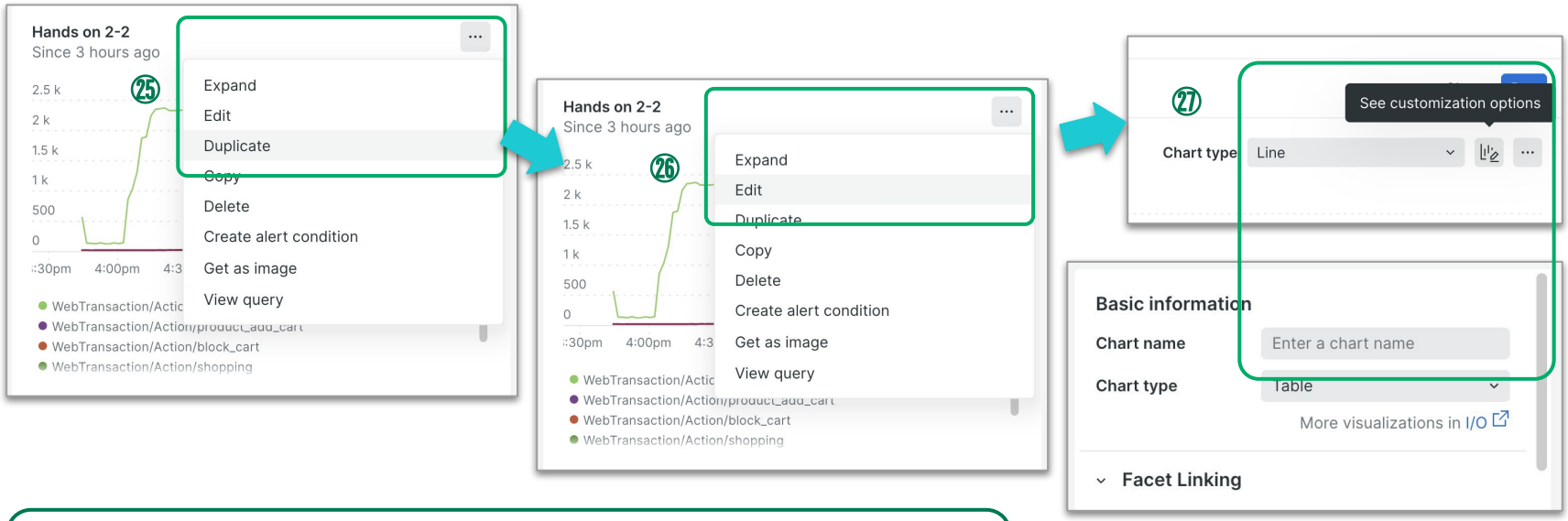
手順㉔でリンクをクリックできなかった場合は、メニューの [Dashboards] から選択してください

手順

⑳ 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



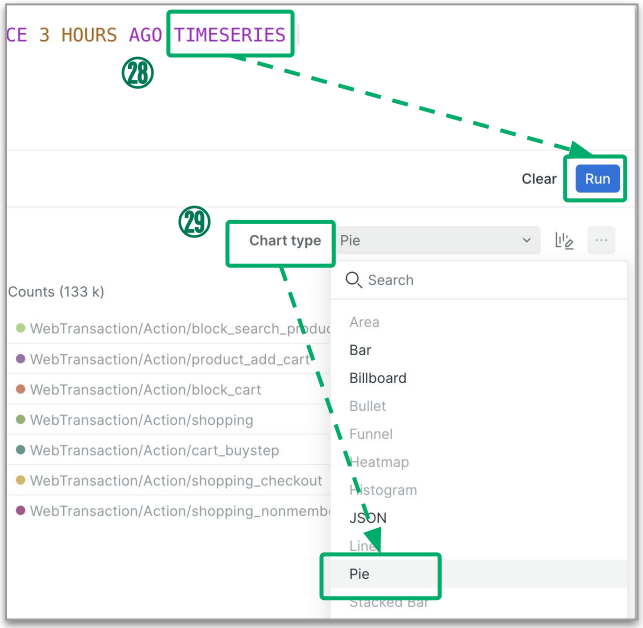
手順

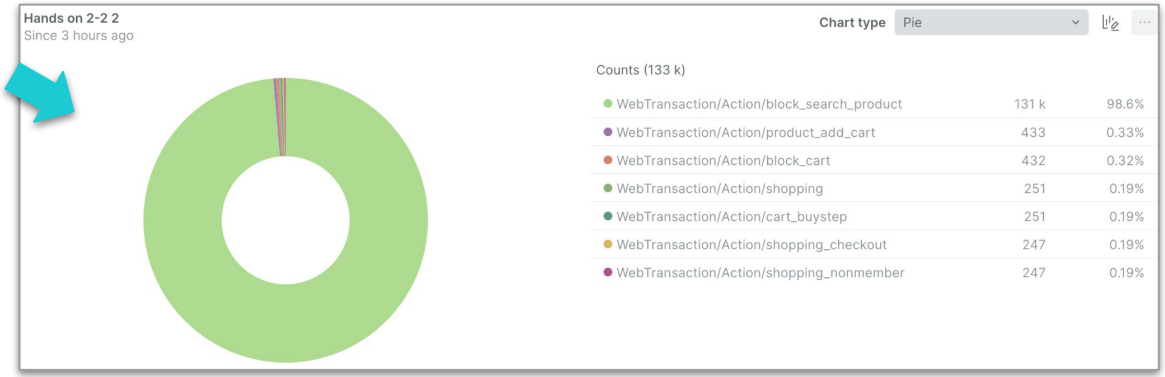
㉕ 追加したチャート右上の [•••] - [**Duplicate**] をクリックします㉖ 画面左にチャートが複製されますので、[•••] - [**Edit**] をクリックします
㉗ Edit widget画面の[See customization options] を表示させ [**Chart name**] にチャート名を入力します

複製後のチャートは元のチャートと同じ名前になるので変更しておきます

チャートタイプの変更

チャートのタイプを Line から Pie に変更します





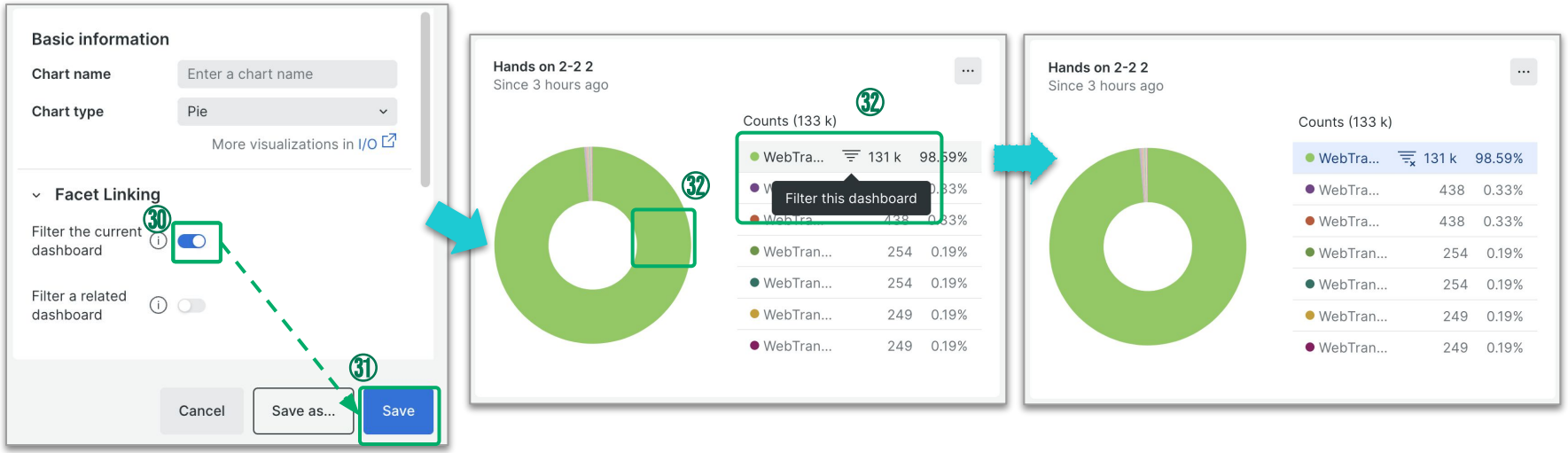
手順

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

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

ファセットフィルタリングの設定

FACET 句で指定した属性によるフィルタを行えるよう設定します



手順

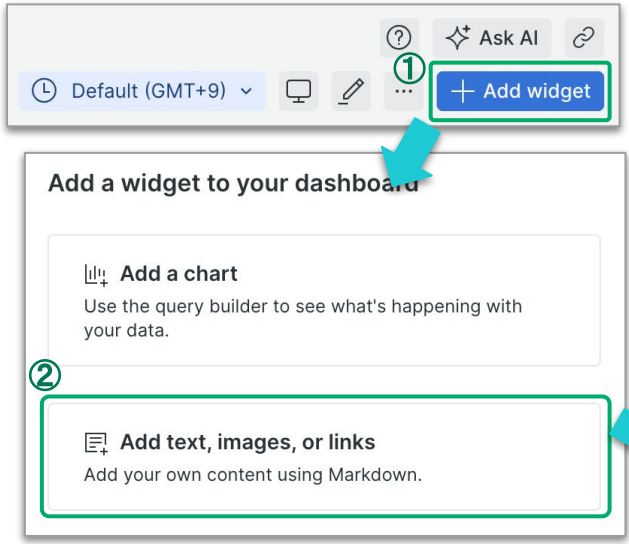
㉚ [Filter the current dashboard] を有効化 ( )します

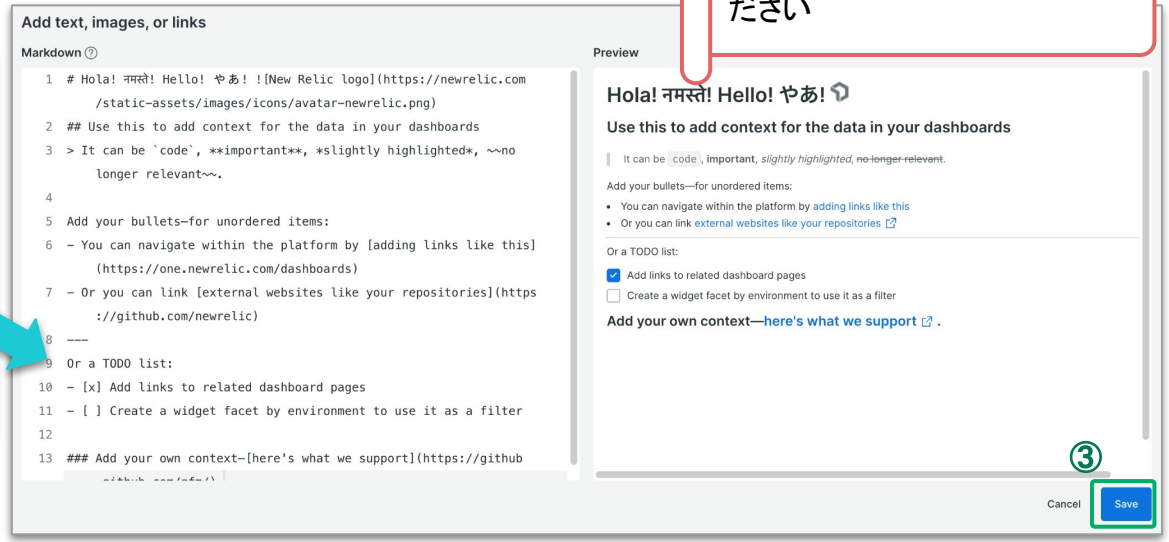
㉛ チャートを保存します

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

Option!! NOTE機能の利用

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





時間に余裕のある方は
Markdownを編集してみてください

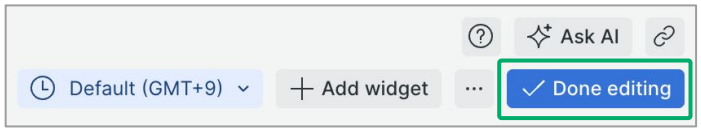
参考: GtiHub マークダウンガイド
https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax

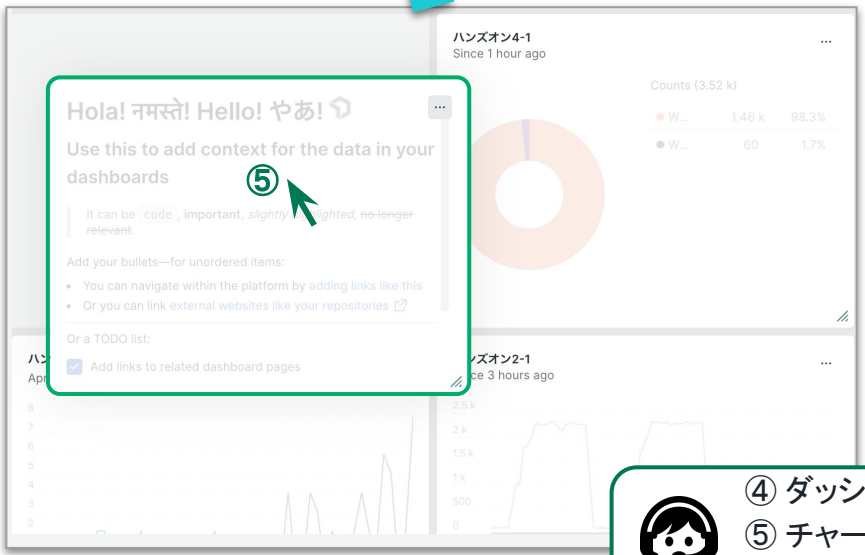
手順

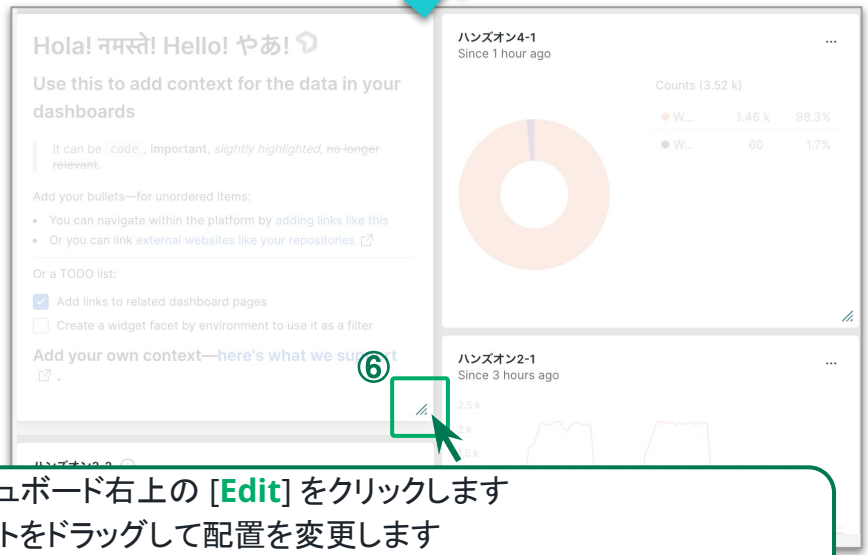
① ダッシュボード右上の [+ Add widget] をクリックします
② [Add text, images, or links] をクリックします
③ Markdownの編集画面が表示されることを確認し、[Save] をクリックします

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、編集を終了します

STEP3

分析手法の習得

NRQL
(New Relic Query Language)

NRU302 - 座学

new relic

NRQLとは

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

NRQLを直接書く

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

NRQL構文

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

参考: New Relic の機能によって報告されるデフォルトのイベント
https://docs.newrelic.com/jp/docs/nrql/get-started/introduction-nrql-new-relics-query-language/

NRQL使いこなしTips ①

この句だけは覚えましょう！

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

NRQL使いこなしTips ②

数値データの集計関数を覚えましょう！

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

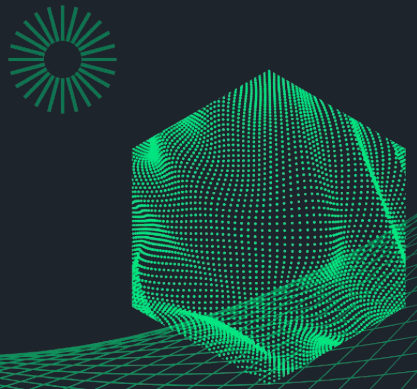
NRQL使いこなしTips ③

WHERE句について

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

分析手法の習得

NRU302 - ハンズオン #3



ハンズオン - 分析手法の習得

このハンズオンでは、以下の点を学習します。

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

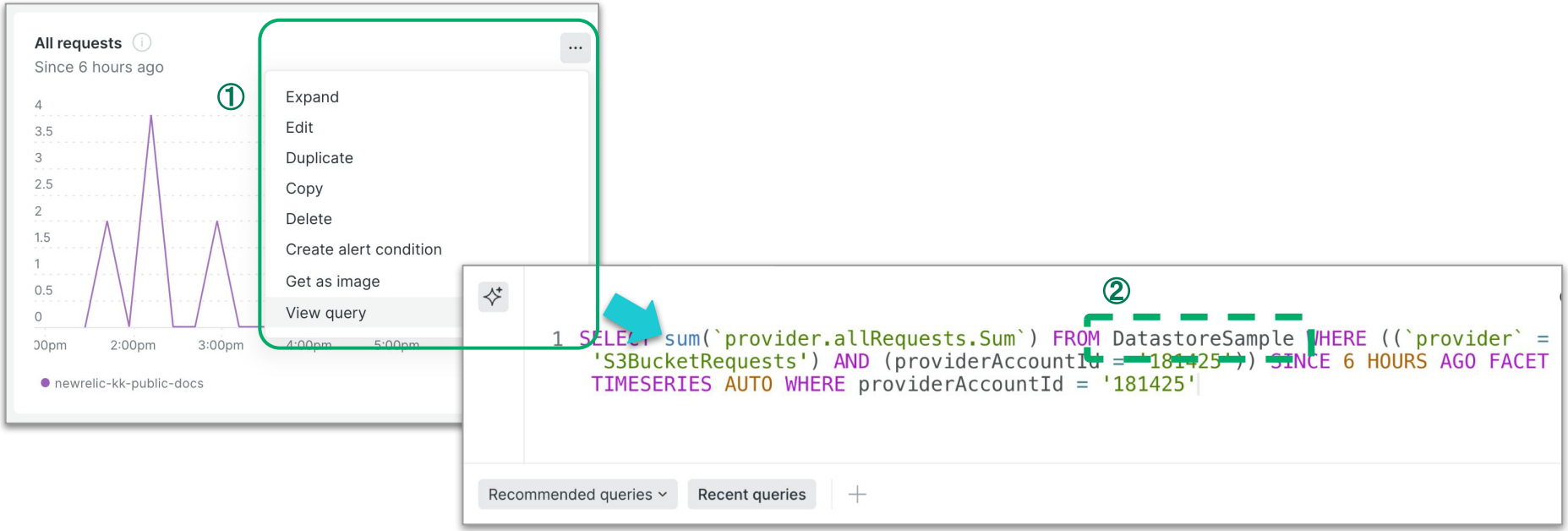
Option!!
- NRQLを用いて様々なチャートを作成する

16:15 - 16:35 (20min)

p. 75 - p. 93

NRQLの確認

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



手順

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

② NRQLのFROM句で指定されているデータタイプを確認します

NRQLの変更

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

変更前

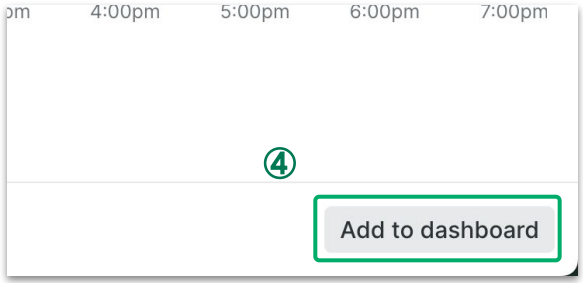
```
SELECT sum(`provider.allRequests.Sum`) FROM DatastoreSample
WHERE ((`provider` = 'S3BucketRequests') AND (providerAccountId = '181425'))
SINCE 6 HOURS AGO FACET entityName TIMESERIES AUTO WHERE providerAccountId = '181425'
```

変更後

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

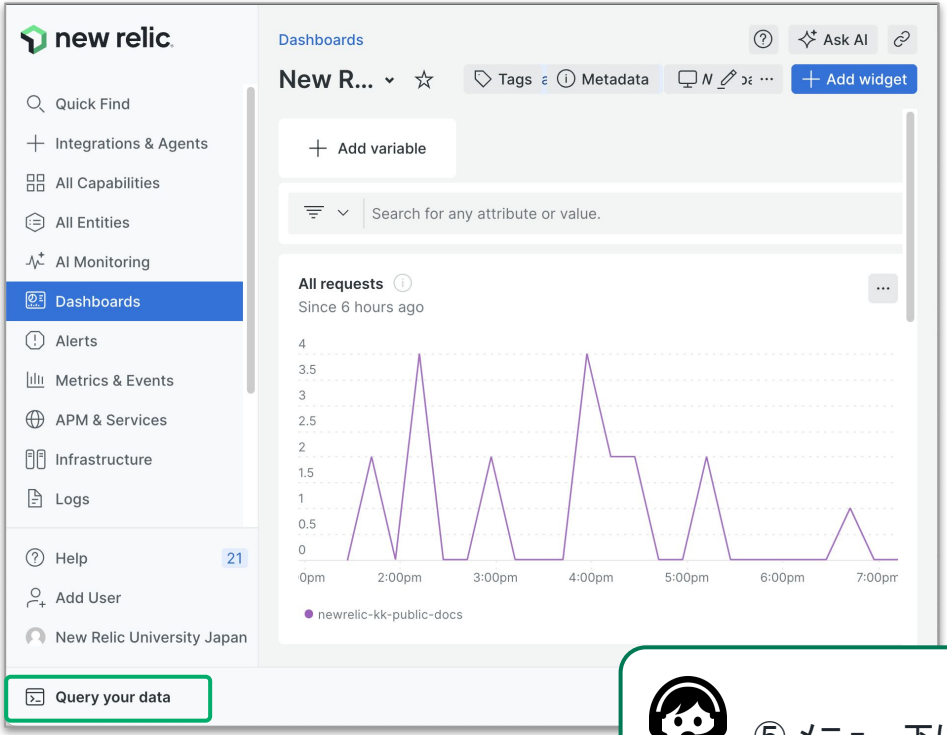
手順

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、チャートをダッシュボードに追加します



Query builderの表示

NRQL を操作するため Query builder を開きます



本ページ以降でいくつかのNRQLを入力し結果の確認を行っていきます
- 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

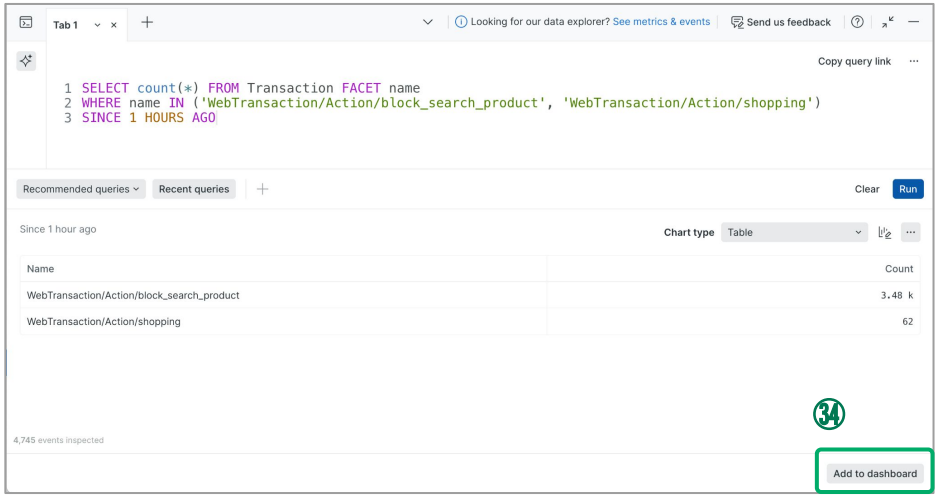
- SINCE *時間* AGO (**TIMESERIES** [***時間***]) : 検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]

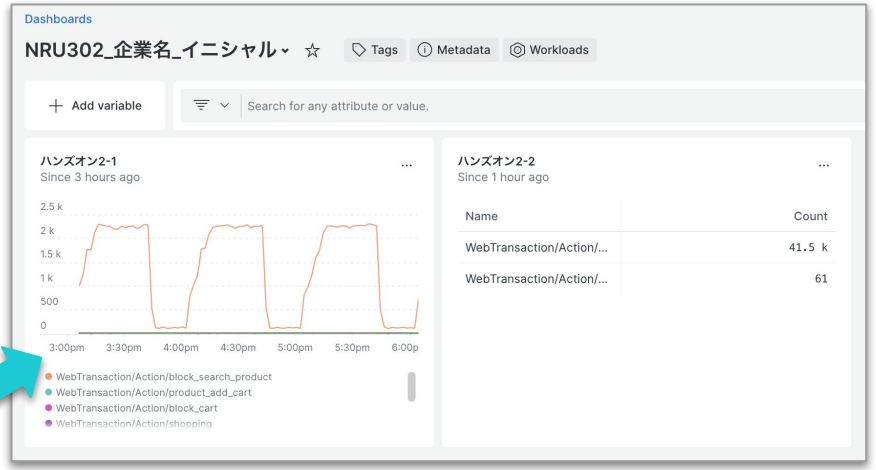
手順

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



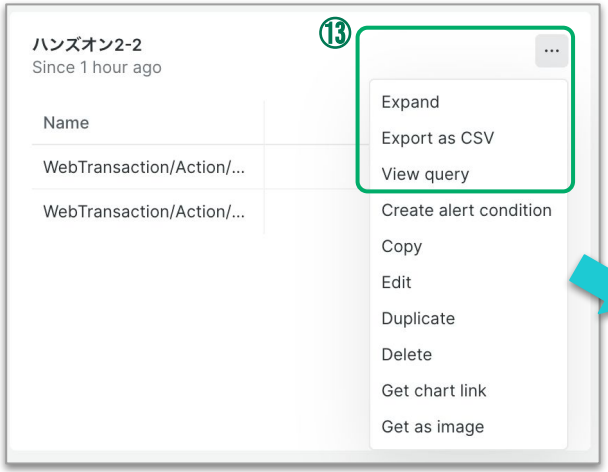


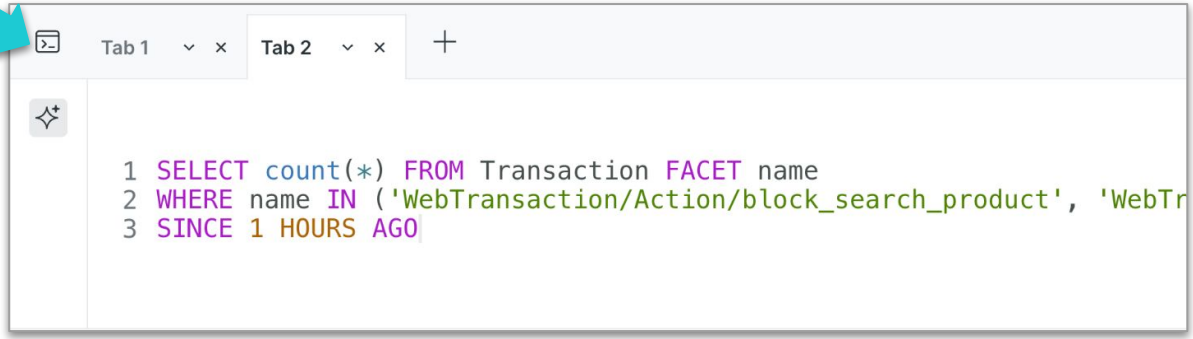
手順

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

さまざまなチャートの追加

追加したチャートをベースにして、他のチャートもダッシュボードに追加します





手順

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

さまざまなチャートの追加 #1

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

変更前

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

変更後

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

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

手順

⑭ 変更後のNRQLを実行し、ダッシュボードに追加します

さまざまなチャートの追加 #2

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

変更前

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

▼

変更後

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

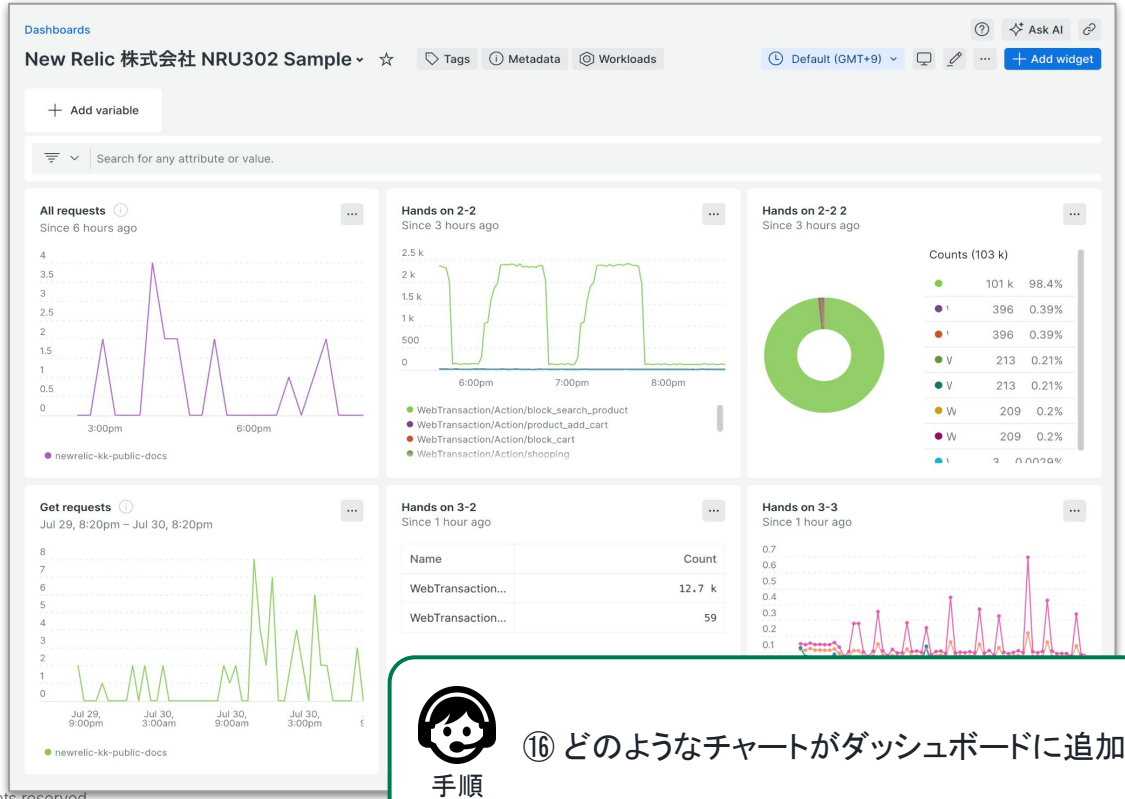
- SINCE *時間* AGO (**TIMESERIES** [***時間***]) : 検索するデータの時間範囲、および時系列データにするかどうかの指定[とそのデータ粒度]

手順

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

ダッシュボードの確認

これまで追加したチャートをダッシュボードで確認します



手順

⑯ どのようなチャートがダッシュボードに追加されたか確認します

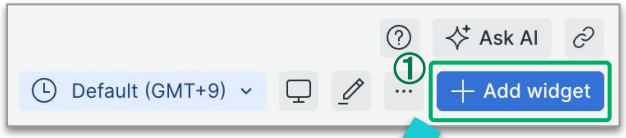
Option!!

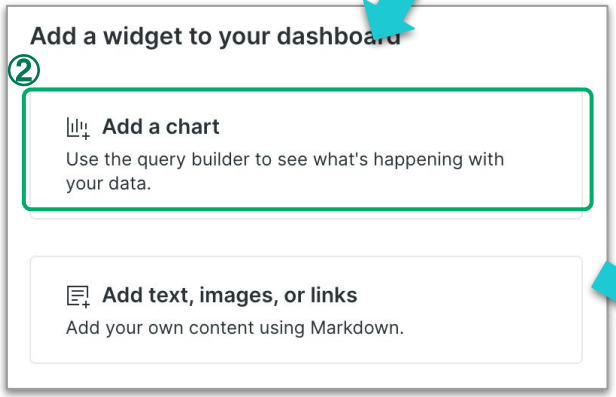
NRQLを用いて様々なチャートを作成する

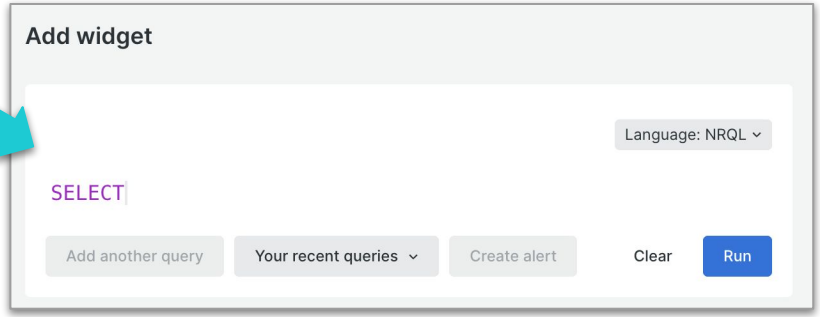
NRU302 - ハンズオン

Option!! NRQLを用いてチャートを作成する

オブザーバビリティ成熟度モデルに例にした 4つのチャートを作成します







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

1 受動的

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

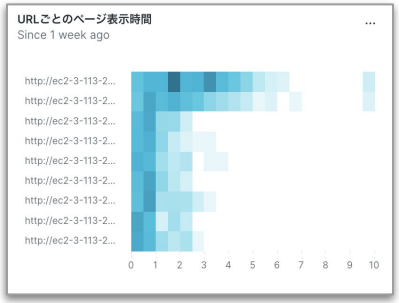

```
SELECT percentage(count(result), where result = 'SUCCESS') AS '稼働率(週)'
FROM SyntheticCheck
SINCE 1 week AGO
COMPARE WITH 1 week AGO
```

- 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

2 積極的

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



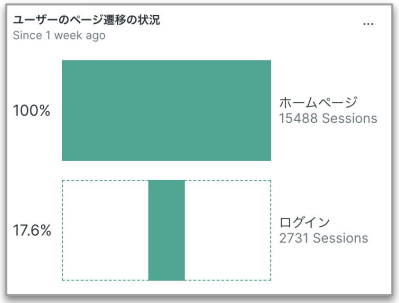

```
SELECT histogram(duration,10,20)
FROM PageView
FACET pageUrl
SINCE 30 MINUTES AGO
```

- 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

3 予測的

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 pageUrl 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 pageUrl 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、pageUrl の値を条件にして、条件に当てはまるsession 数をカウントします
- Chart Typeは **Funnel** を選択します

4 データドリブン

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


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

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

STEP4

アラートの作成

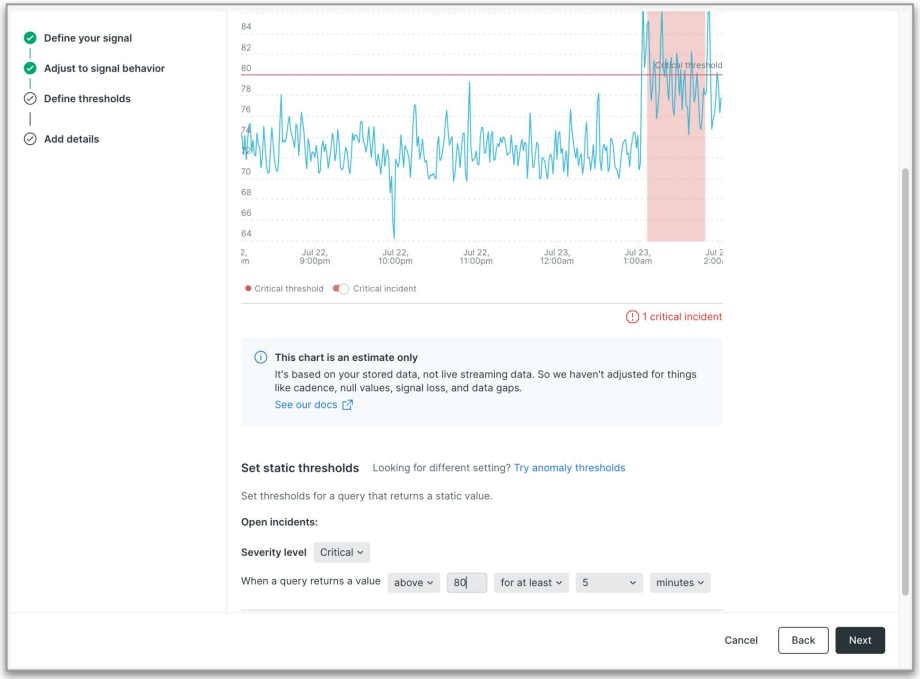
高度なアラート設定

NRU302 - 座学

new relic

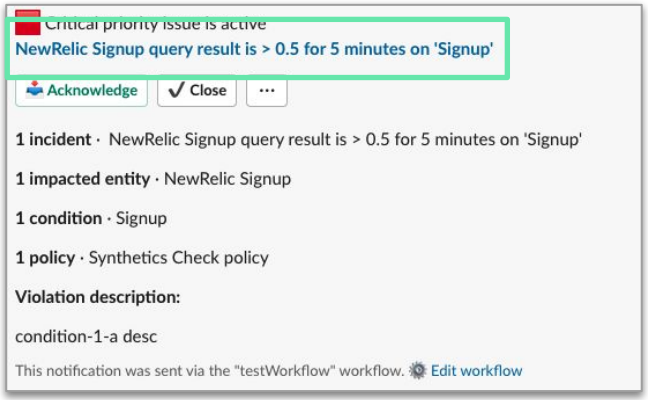
アラート機能

- 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



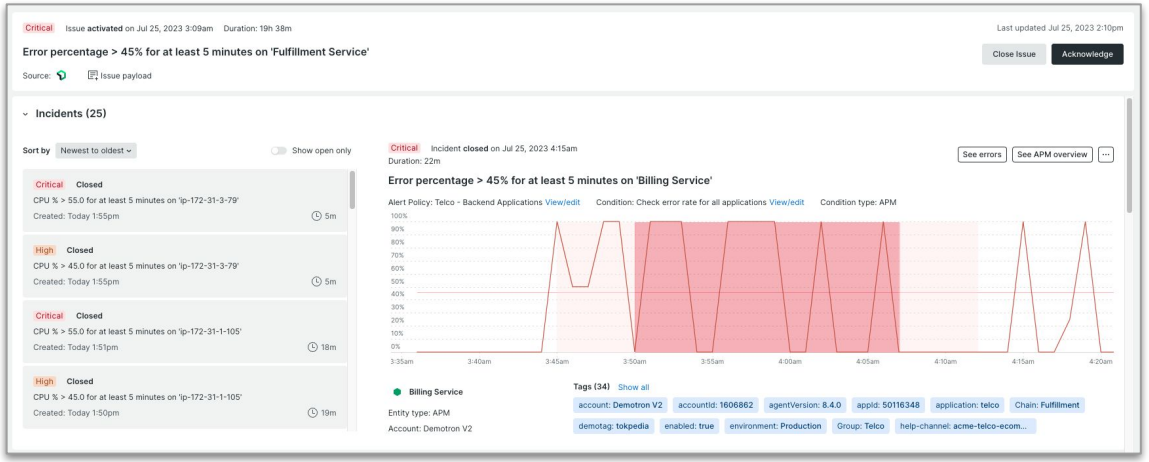
アラートの通知と確認

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)



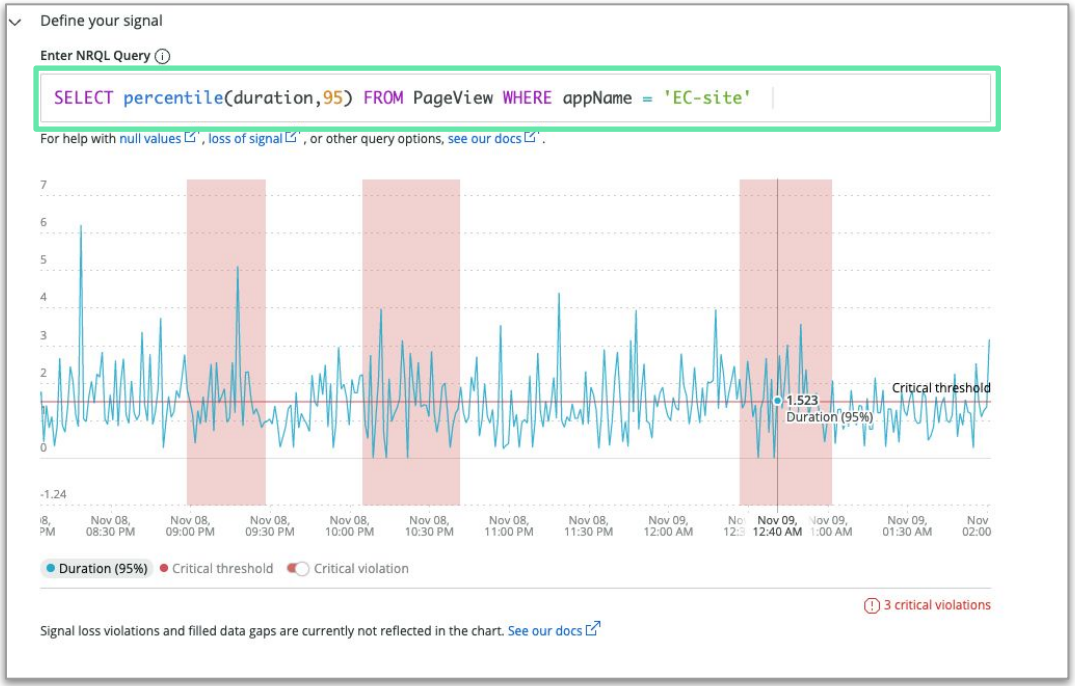
アラートへのリンク

アラートの詳細をNew Relic上で確認



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

new relic University

NRUハンズオンのお知らせ

2024.11.20 (Wed) 15:00 - 17:00

NRU304 - アラート設計の基本と活用

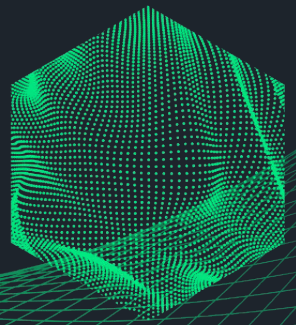
https://newrelic.com/jp/events/2024-11-20/nru304

このハンズオントレーニングでカバーするトピック

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

アラートの作成

NRU302 - ハンズオン #4



ハンズオン - アラートの作成

このハンズオンでは、以下の点を学習します。

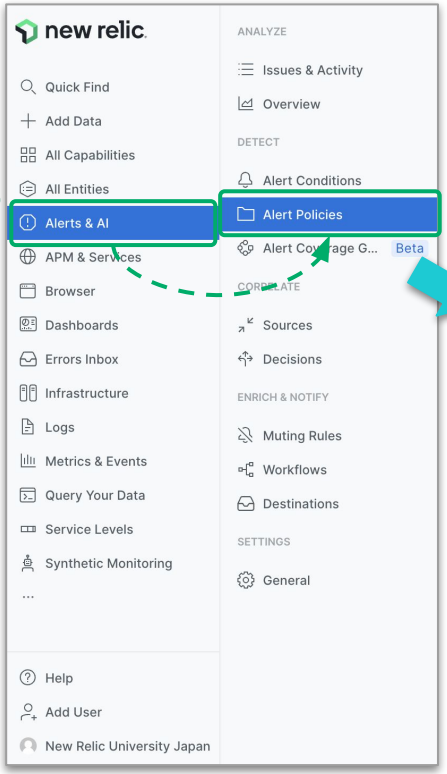
- アラートのためのコンディション作成

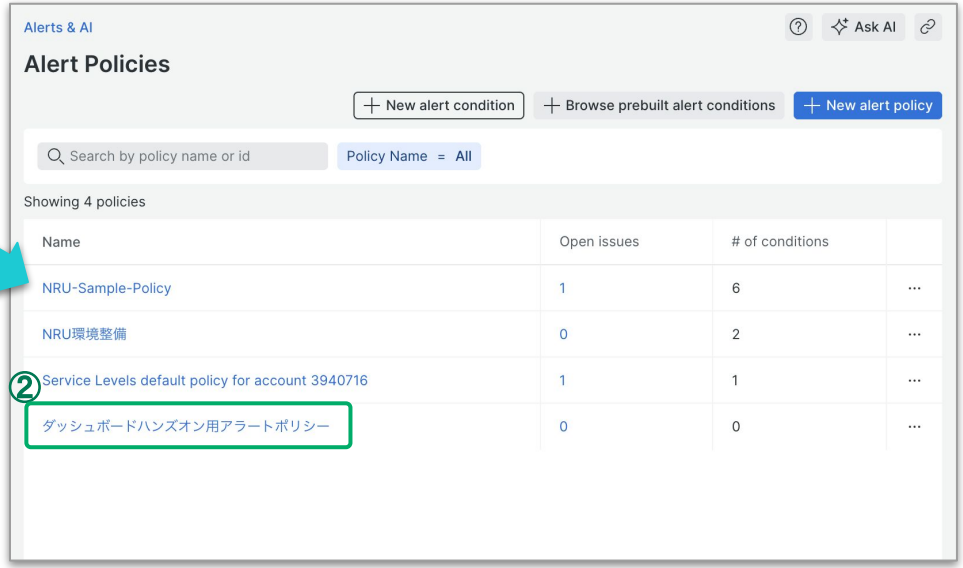
16:40 - 16:50 (10min)

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



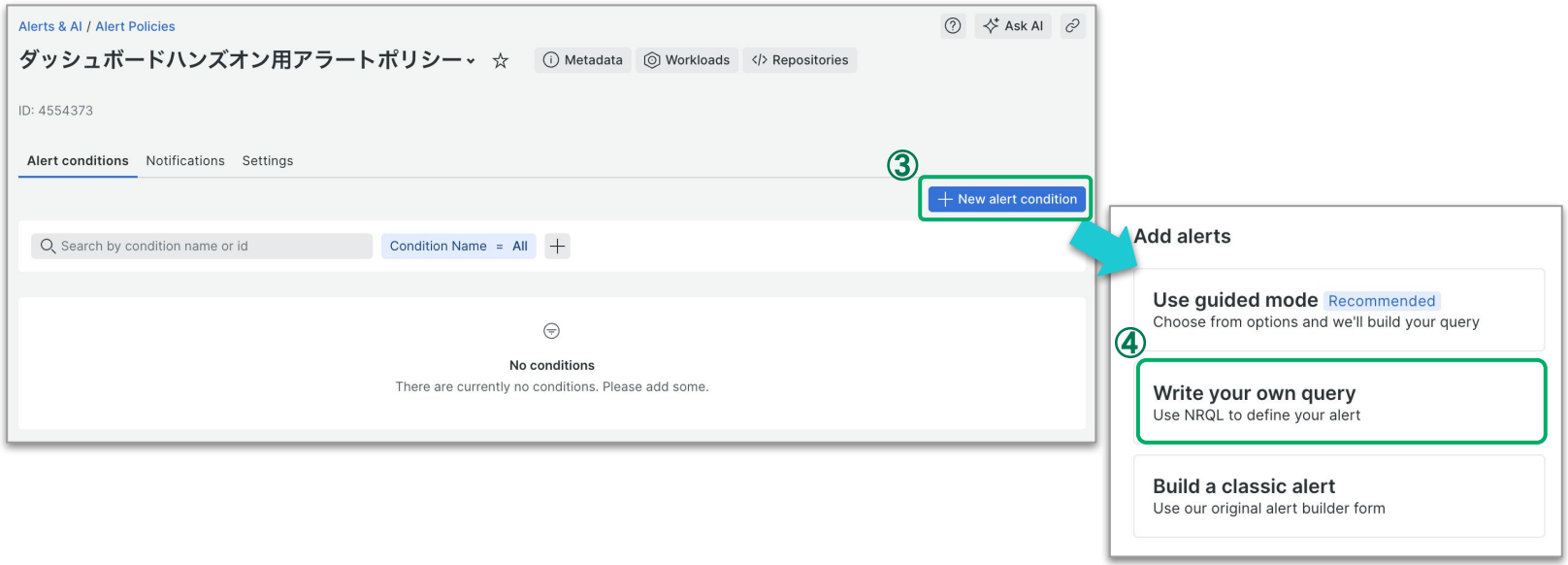


手順

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

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

NRQLの入力

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

NRQL

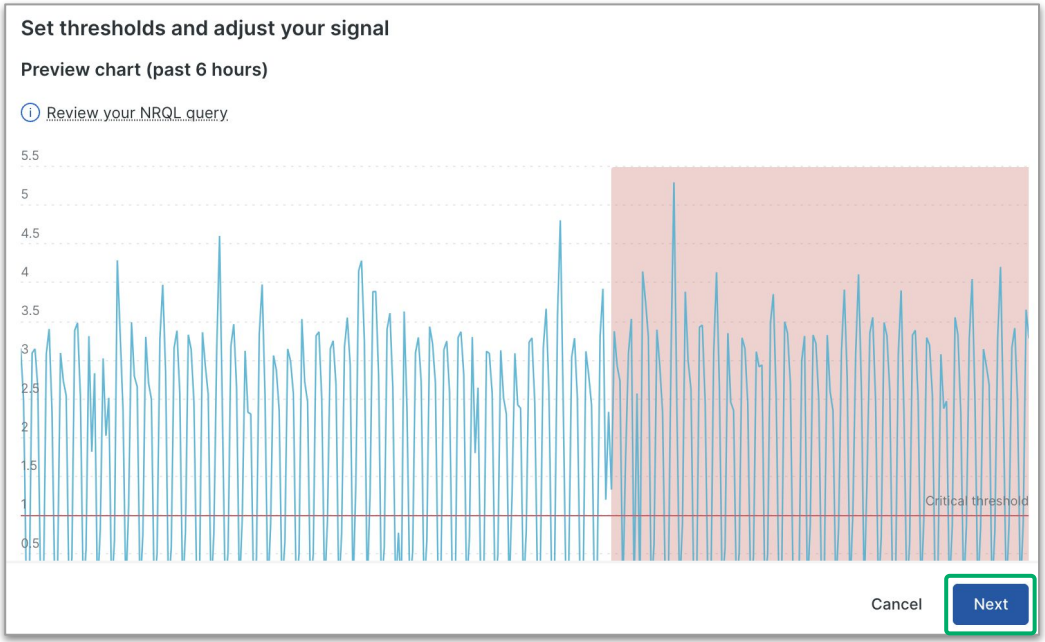
```
SELECT percentile(duration,95) FROM PageView WHERE appName = 'EC-site'
```

手順

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

スレッシュホールドの設定

アラートコンディションの閾値などを設定します



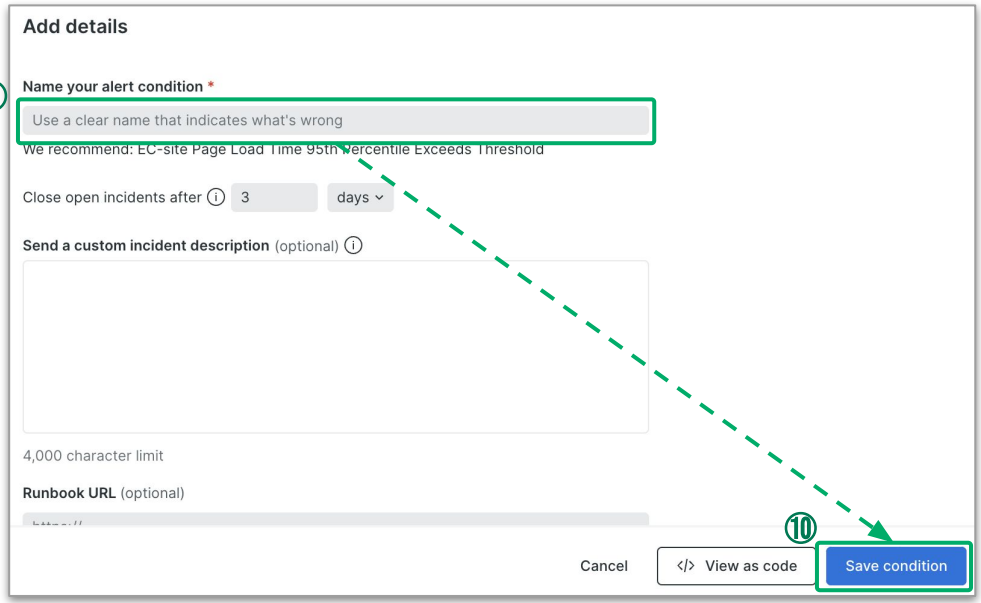
⑧

手順

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

アラートコンディションの保存

作成したアラートコンディションを保存します



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

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

手順

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

さいごに

NRU302

Dashboard と NRQL の 便利な機能

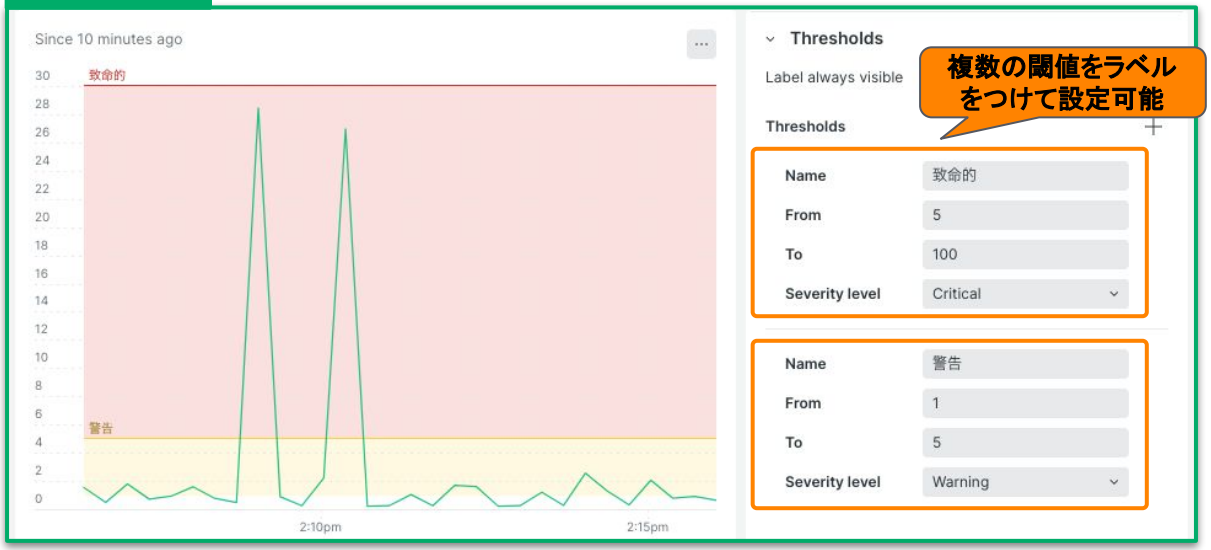
NRU302

LineとTableチャートで直感的に問題を発見

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

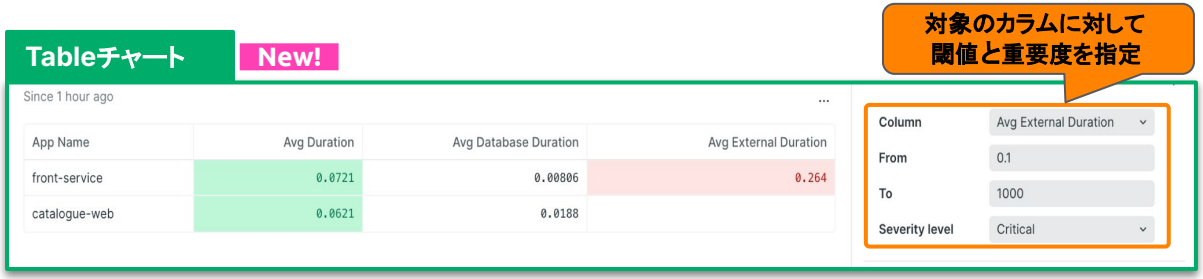
参照：公式ドキュメント

Lineチャート



Tableチャート

New!



new relic

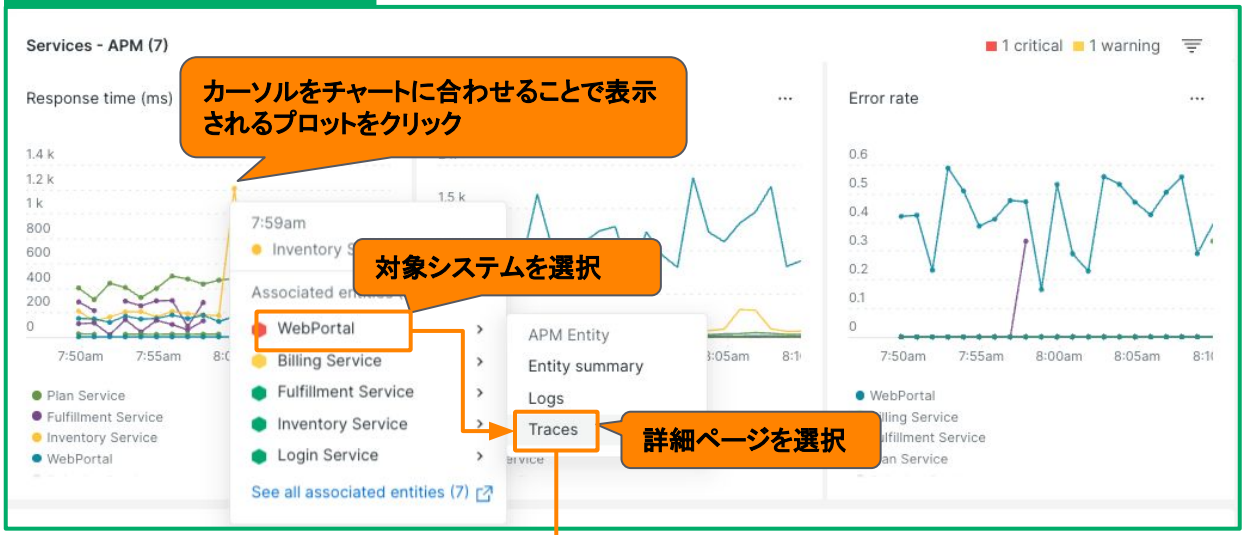
チャートからシームレスにドリルダウンして詳細分析

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

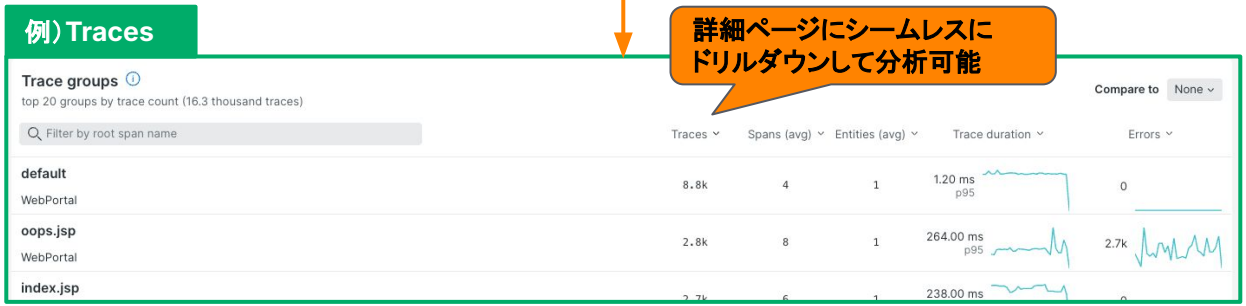
†WHERE句でentityGuid、entity.guid、appName、entity.name、または、entityNameを指定

参照：What's new

例）Workloads > Activity



例）Traces

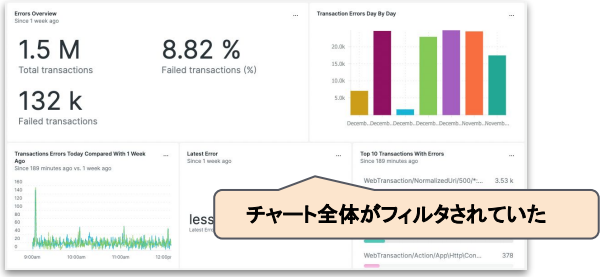


ダッシュボード テンプレート変数

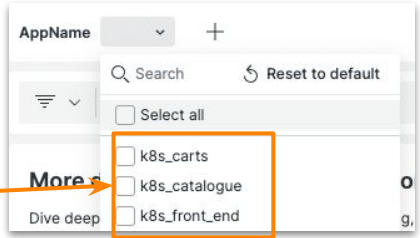
- あらかじめ変数を用意することで、フィルタ項目の検索が不要になります
- 個々のチャートをそれぞれ別々の変数でフィルタリングできます
- 固定リストとNRQLのクエリ結果を選択肢にできます

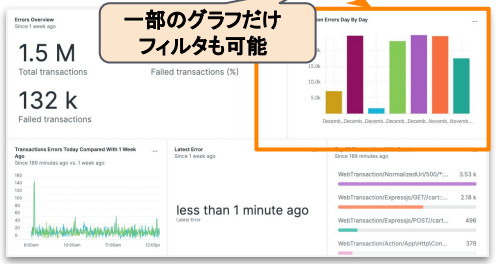
参照:テンプレート変数

従来のフィルタ機能



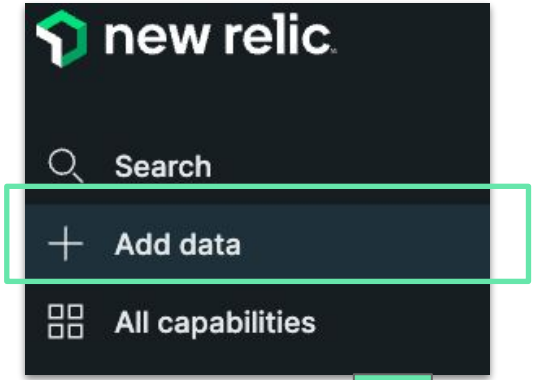
追加のフィルタ機能

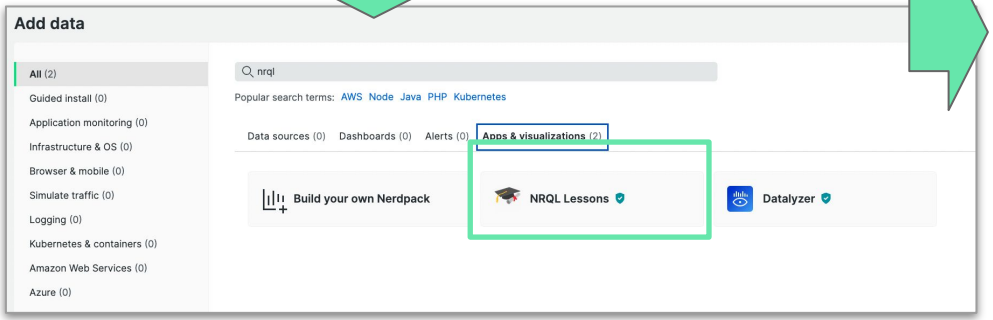




new relic

NRQL Lessons - NRQL学習ツール

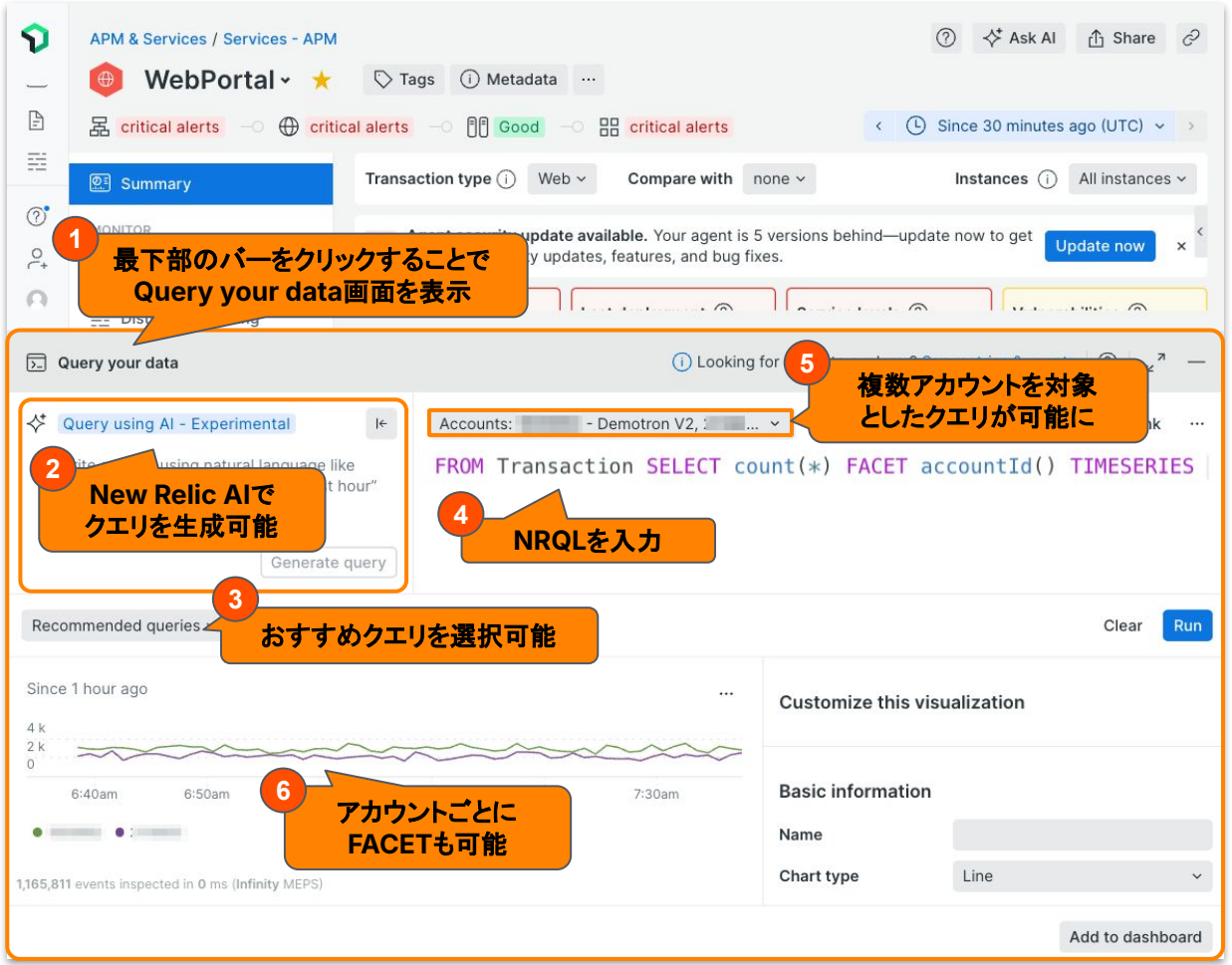




NRQLに関するさまざまな利用方法を学習することができます

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**に問い合わせてクエリを生成可能
- **おすすめのクエリのリスト** を提供



参照：What's new

NRQLで使える関数が大幅に増えました

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

参照：What's new

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

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

1 フィールドを取得
2 文字列から日時に変換
3 変換後計算可能
4 エンコード
5 元のデータ

timestamp	Stage	StartedAt	FinishedAt	Duration	encode(builds, ...	builds
13:33:29.611	build	1707117637233	2024/02/05	40189	W3siaWQiOjEwODU...	[{"id":10853,"stage":"build","name":

例）cidrAddressの使用例

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

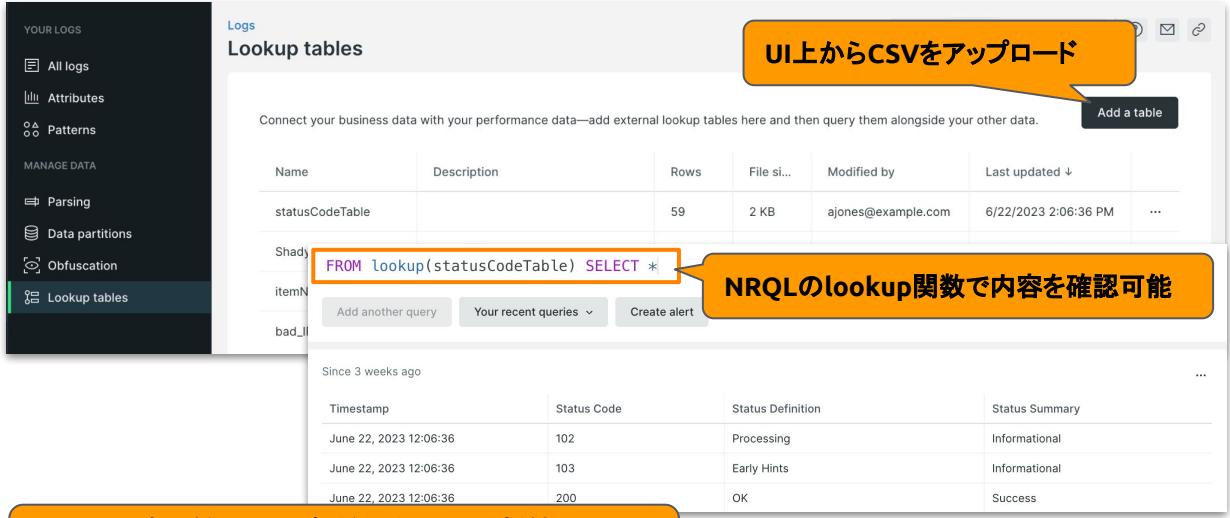
/10	/12	/24
172.0.0.0/10	172.16.0.0/12	172.27.10.0/24

CIDRを取得

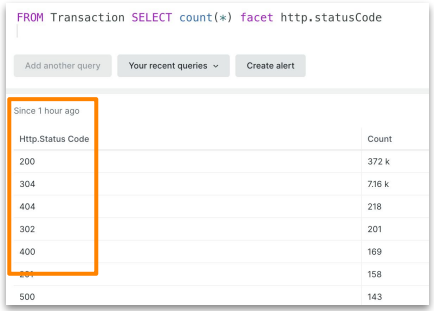
Lookupテーブルを アップロードし、 NRQLで分析可能に

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

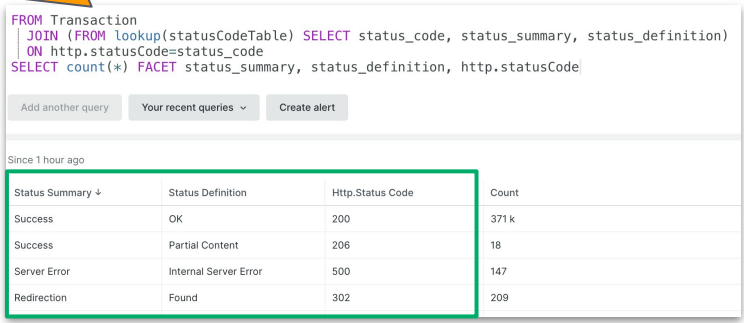
参照：公式ドキュメント



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†でも使える！**
- **Query Builder**で利用可能

参照:
- 公式ドキュメント

†Keyとなるデータは最大100件まで対応

アプリケーション由来のデータ (Transaction Event)

```
FROM Transaction SELECT average(duration) FACET host
```

Add another query | Your recent queries | Create alert

ホスト名 / アプリ応答時間

Host	Avg Duration
order-composer-5495f9b947-m9ts6	5.71
order-composer-5495f9b947-2gt6z	5.67
order-composer-5495f9b947-dr6pl	5.53
order-composer-5495f9b947-n8×52	5.45
order-processing-695bbd958f-bzzvn	4.73

インフラ由来のデータ (SystemSample Event)

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

Add another query | Your recent queries | Create alert

ホスト名 / 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
```

Add another query | Your recent queries | Create alert | Clear | Run

ホスト名 / アプリ応答時間 / インフラリソース使用率

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

Basic information
Chart name: Enter a chart name
Chart type: Table
More visualizations in I/O
Other groups (1)

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

new relic

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
```

条件の説明を追記できる

コメントアウトでデバッグ可能

NRU GUIDE

NRU302

New Relic University

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

Install

New Relic を使い始める

New Relic One へのサインアップやエージェントインストールの方法などのガイドを提供

APM / Browser / Infrastructure / Logs / Mobile (iOS/Android) / AWS統合 / Azure統合 / GCP統合 インストール手順

▸サインアップ方法

https://newrelic.com/jp/blog/how-to-relic/create-new-account

▸インストールガイド

https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide

NRU 100

Observability/New Relic を知る

New Relic One 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

NRU Practitioner オブザーバビリティ入門

NRU 101 New Relic One 入門

▸オンデマンドセミナー

https://newrelic.com/jp/resources/presentations/nru-practitioner2022

https://newrelic.com/jp/resources/presentations/nru101-2022

NRU 200

New Relic の主要機能を学ぶ

New Relic One に含まれる3つの主要機能に含まれる5つの機能群を動画で説明

NRU201 Telemetry Data Platform

NRU202 Full Stack Observability

NRU203 Applied Intelligence

▸主要機能解説動画

https://newrelic.com/jp/learn

NRU 300/400

New Relic の使い方を体感する

New Relic One を実際に操作し、主要機能を利用できる状態にするためのトレーニング

NRU 301 アプリケーションとインフラ性能観測の基本

NRU 302 ダッシュボード開発とNRQLの基本

NRU 303 SLI/SLO設計の基本

NRU 304 AIOps とアラート設計の基本

NRU 401 IDEと連携し、問題解決を加速するNew Relic活用の実践

▸開催スケジュール

https://newrelic.com/jp/events

Exam

資格を得る

New Relicの知識を有していることを証明するための試験、合格すると資格バッジを授与

フルスタックオブザーバビリティ認定試験

▸受験サイト

https://learn.newrelic.com/full-stack-observability-exam-jp

2024年10-12月期開催のハンズオンセミナー

2024年10月9日（水）15:00-17:00
NRU301 - アプリケーションとインフラ性能観測の基本
https://newrelic.com/jp/events/2024-10-09/nru301

2024年10月23日（水）15:00-17:00
NRU302 - ダッシュボード開発とNRQLの基本
https://newrelic.com/jp/events/2024-10-23/nru302

2024年11月6日（水）15:00-17:00
NRU303 - SLI/SLO 設計の基本
https://newrelic.com/jp/events/2024-11-06/nru303

2024年11月20日（水）15:00-17:00
NRU304 - アラート設計の基本と活用
https://newrelic.com/jp/events/2024-11-20/nru304

2024年12月11日（水）15:00-17:00
NRU401 - IDEと連携し、問題解決を加速するNew Relic活用の実践
https://newrelic.com/jp/events/2024-12-11/nru401

[開催形式]
全てZoomによるウェビナー

こちらからも一覧でご覧いただけます！

new relic

Observability Engineerに！

Full Stack Observability Practitioner認定試験

【この認定試験を通じて身につくスキル】

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

【合格者特典】

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

【認定試験に向けた準備】

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

認定試験受験サイトは こちら

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

new relic

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

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

New Relic セミナー

お客様事例紹介

NRUG
(ユーザーグループ)

ユースケース
実践的な情報

初めての方向け

日本語で提供

英語でのみ提供

お知らせ

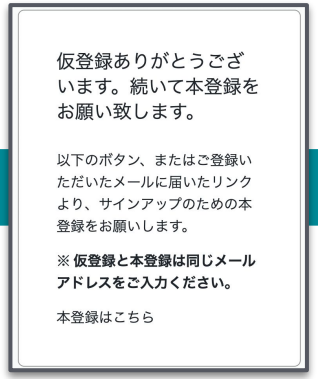
NRU302

New Relic [ずっと] 無料サインアップ (Link)

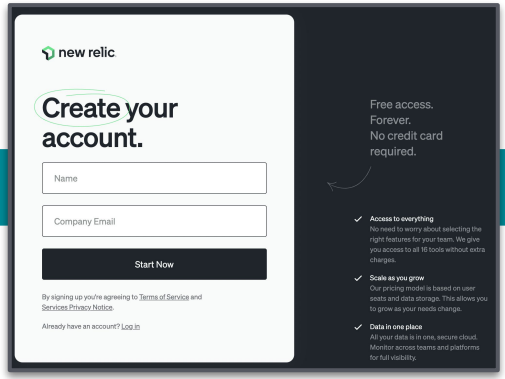
1名のフルユーザーアクセス、100GB/月のデータ保存容量

① 仮登録への入力

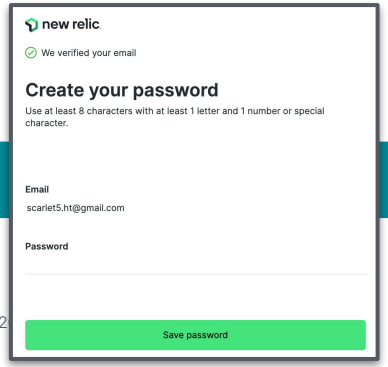
② 本登録へ遷移



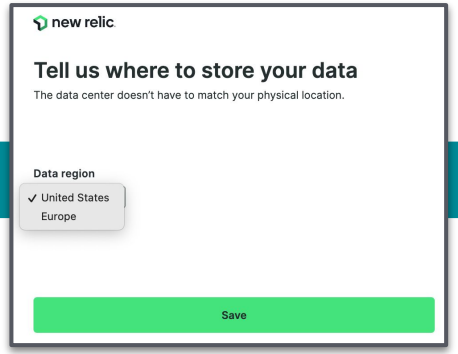
③ 本登録への入力



④ パスワード設定



⑤ データ保存先を選択



実践的なオブザーバビリティを学べるガイドブック

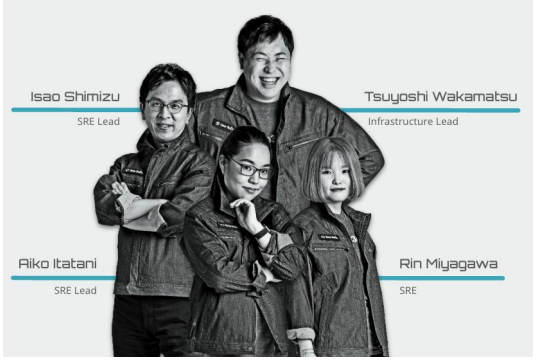
■Part 1：New Relicを知る
・第1章：オブザーバビリティの重要性
・第2章：New Relicの全体像
■Part 2：New Relicを始める
・第3章：New Relic Synthetic Monitoring
・第4章：New Relic Mobile
・第5章：New Relic Browser
・第6章：New Relic APM
・第7章：New Relic Infrastructure
・第8章：New Relic NPM
・第9章：New Relic Log Management
・第10章：New Relic Alerts & AI ①：New Relic Alerts
・第11章：New Relic Alerts & AI ②：AI
・第12章：DevSecOps
・第13章：ビジュアライゼーション
■Part 3：New Relic活用レシピ

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

NRUG
ぬるぐで学ぶ

New Relic User Group

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



NRUG 本部

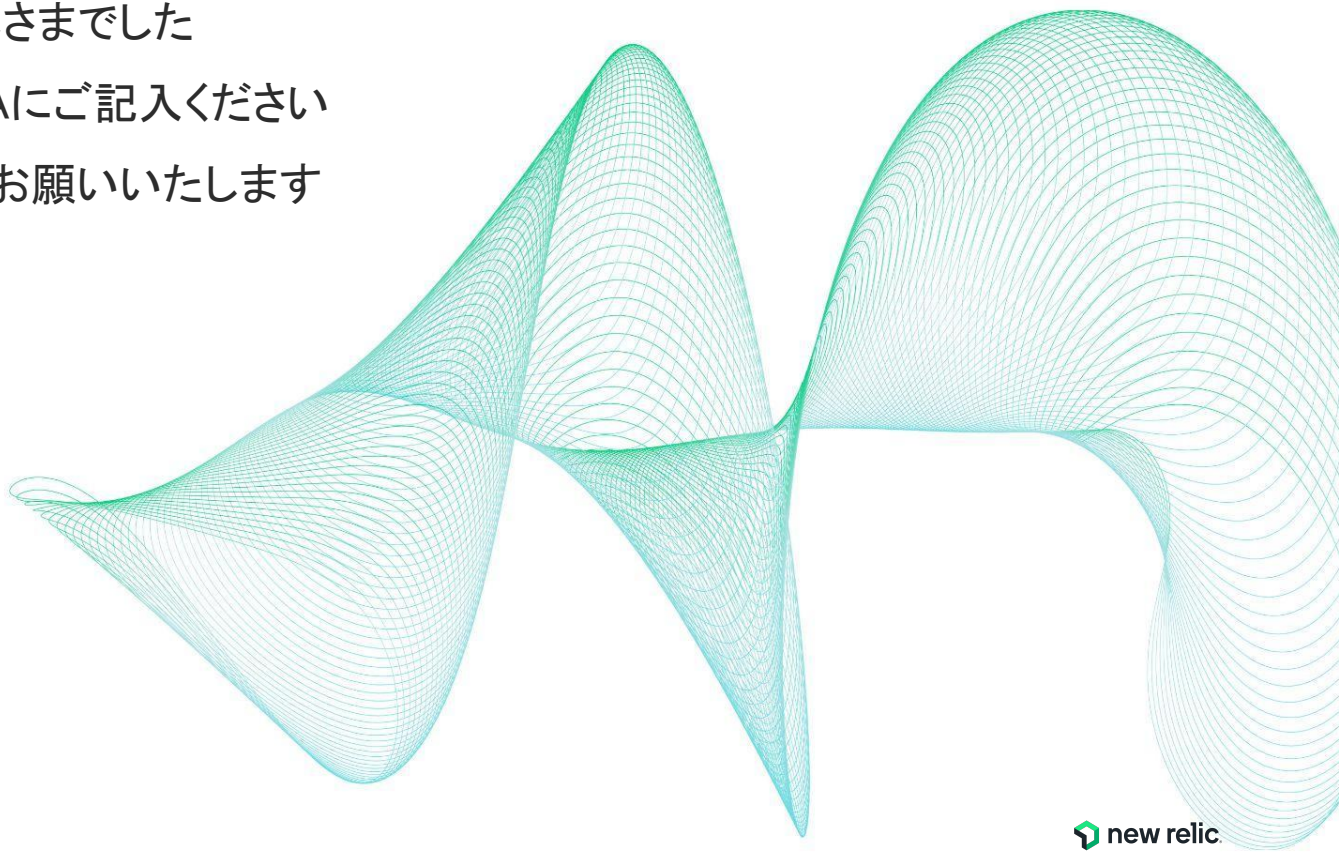
NRUG SRE支部

NRUG 沖縄支部

以上、お疲れさまでした

ご質問があればQ&Aにご記入ください

アンケートにご協力お願いいたします



Thank you.

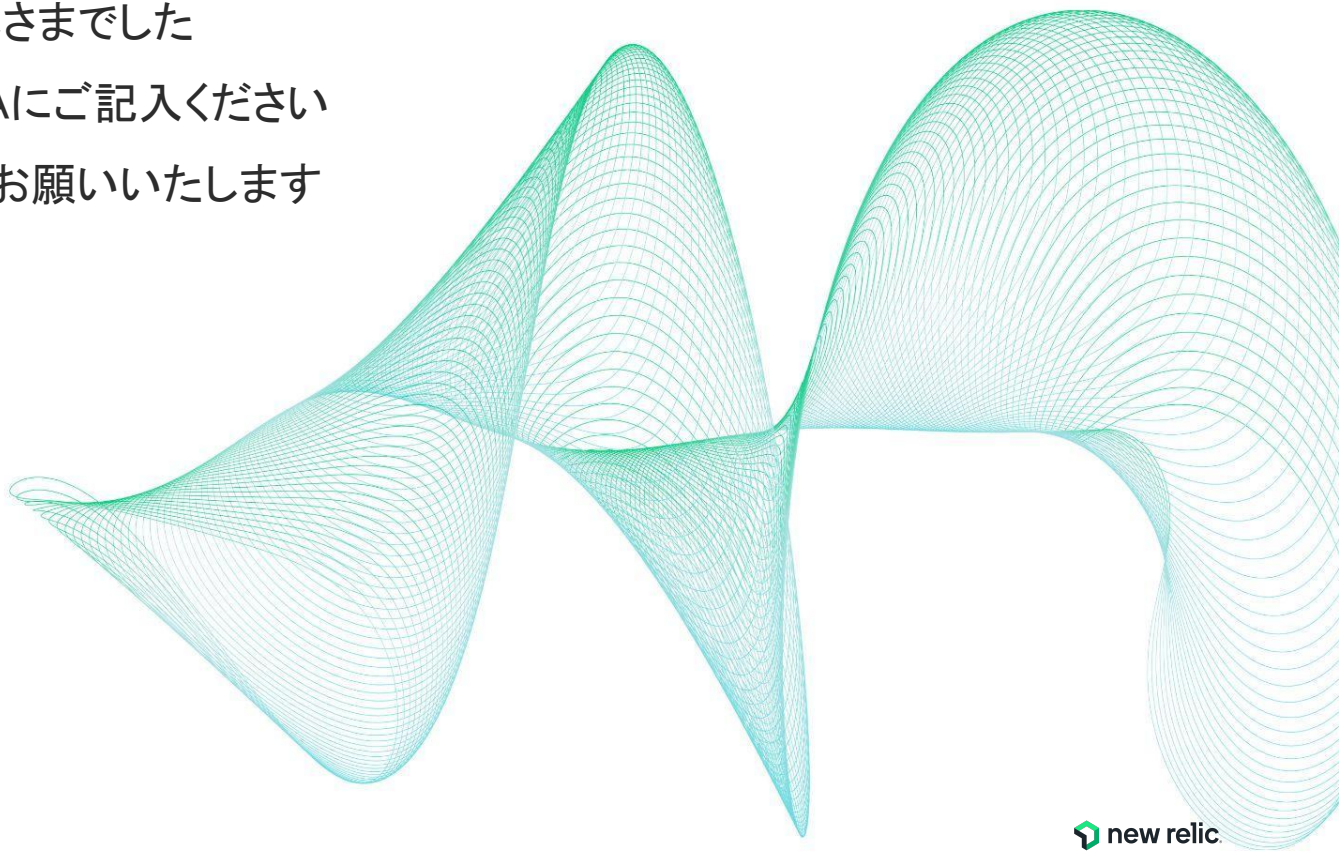
NRU302

new relic

以上、お疲れさまでした

ご質問があればQ&Aにご記入ください

アンケートにご協力お願いいたします



Thank you.

NRU302

new relic