



ダッシュボード開発と NROLの基本編

NRU 302 - Dashboard / NRQL

©2008-20 New Relic, Inc. All rights reserved

本ウェビナーについて

本ウェビナーでは質問が可能です。

音声不具合などについてもこちらからご連絡ください。

質問は最後のQ&Aの時間にて可能な範囲で回答いたします。

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





New Relic K.K. Solutions Consultant

ネットワーク/パフォーマンス/サーバの運用監視にて エンジニアとしてのキャリアを開始。 得意な技術は、

- Content Delivery Network
- 暗号化技術
- ・ ID管理
- Cloud Security



本日のゴール

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

本ウェビナーの受講想定者

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

New Relicの知識に不安のある方はこちらを受講ください! (オンデマンド視聴可) https://newrelic.com/jp/webinar/nrb-newrelic-essentials

本日のタイムテーブル

時間	タイプ	内容
15:00 - 15:20	座学	New Relicとデータ分析 New Relicが取得するデータ
15:20 - 15:40	ハンズオン	データの理解
15:40 - 15:50	座学	NRQL (New Relic Query Language)
15:50 - 16:10	ハンズオン	分析手法の取得
16:10 - 16:15	座学	New Relic Oneダッシュボード
16:15 - 16:35	ハンズオン	ダッシュボードの作成
16:35 - 16:40	座学	高度なアラート設定
16:40 - 16:50	ハンズオン	アラートの作成
16:50 - 17:00		まとめ、アンケートご記入

New Relicとデータ分析



デフォルトのUIでもデータ分析はできます

O New Relic ONE™ Account: 2511671 - NewRelicUniversity-Japan ∽

Explorer ~ Services - APM / 🕁 📕 EC-site 🗸 NewRelicUniversity-lapan 🛇 5 Show new view 🌑 Copy permalink v (Since 60 minutes ago v > Transaction type ⑦ Web ~ Compare with none ~ Instances ⑦ All Instances ~ Summary Error traces This entity isn't producing any traces with errors for the time range you've chosen. To find traces with errors, try Web transactions time ① 870.41 ms 0.79 s End user <u>2 111 %</u> adjusting the time range. Monitor APP SERVER BROWSER PAGE LOAD ... Distributed tracing Application activity 〒 Filters ∽ 1200 ms Service map 1000 ms Critical violation opened 2:53 pm Dependencies 800 ms Web response time deviated from the baseline for at least 5 minutes on 'EC-site' Transactions 600 ms 400 ms Databases Critical violation opened 2:52 pm 200 ms Web response time deviated from the baseline for at External services least 5 minutes on 'EC-site' 02:00 PM 02:05 PM 02:10 PM 02:15 PM 02:20 PM 02:25 PM 02:30 PM 02:35 PM 02:40 PM 02:45 PM 02:50 PM 02:55 PM 'M' Critical violation opened 2:52 pm Triage PHP
 MvSOL
 Web external
 Response time Web response time deviated from the baseline for at least 5 minutes on 'EC-site' Errors Inbox 232 rpm Throughput () Error rate ① 0.029 % Apdex score ① Show more activity APP SERVER BROWSER ··· Events 1.5 k 3 % **Related** entities () About this data 2.5 % Errors 1k 2.96 Map view 0.6 1.5 % Violations 500 This entity: 1.% 0.2 0.5 % is called by 4 synthetic monitors Deployments 0 % 0000 02:00 PM 02:15 PM 02:30 PM 02:45 PM 03 02:00 PM 02:15 PM 02:30 PM 02:45 PM 02: 02:00 PM 02:15 PM 02:30 PM 02:45 PM 03 Thread profiler Web throughput Critical Violation Web errors
 All errors App server End user Critical Violation is hosted by 1 host Reports Transactions . 5 slowest transactions (by total time) SLA Add/edit related entities Transaction name 0 Slowest trace ① Error rate 🗘 Average duration 🤤 Scalability admin_homepage_sale 1.56 min 100% 1.17 min Repositories Web transactions 1 38 min 0% 1.02 min admin_logout Database Connect your first repository. 0% admin_homepage 21.5 s Background jobs Connect repository block_search_product 2.04 s 0% 852 ms

🕞 Query your data 🔐 Instant Observability 🔠 Apps 🧭 Get started 🔍 🕐 😴 🔂 107

・・・が、人によってみたい観点は全く異なる システム ビジネス ロール 計測したいことは密接に関わり合ってい る プロダクト プロダクトオー アプリ インフラ マネージャー > -開発者 ナー 運用者 売上/コスト アプリの 顧客行動 状態 Ś 0 . . . みたい観点 ユーザー 売上/コスト アプリの インフラの エクスペリエン 顧客行動 状態 状態 ユーザー ス エクスペリ インフラの エンス 状態 現在の それぞれ異なるツール 利用ツール



New Relicによるデータ分析のメリット



New Relic のObservability 成熟モデル

成熟度	特徵	КРІ	関わるロール
4 データ ドリブン	 データを基にした経営判断 マーケットへの素早い対応 CSAT/Net Promoterスコアの改善 	 顧客満足度 経営計画や経営判断と紐付いたSLOの活用 Time-To-Market (# of new products/features) 	
3 _{予測的}	 オブザーバビリティを用いた運用 顧客体験の改善 丁度良いスケーリング 	 % of 重大な障害の削減 % of エラーバジェットの利用数 % of SLOアラートの数 % of ビジネス指標のトラッキング数 	プロダクト オーナー
2 積極的	 MTTDの改善 サービスレベル改善のトラッキング デジタル顧客体験の改善施策の実施 	 平均MTTD % of SLOが定義されたサービス数 % of クリティカル・ケイパビリティの策定(DCX) 	プロダクトマネージャー
1 受動的	 MTTRの改善 アプリケーションパフォーマンスの把握 顧客に影響のある事象への対応の改善 	 平均MTTR % of パフォーマンス低下事象の発生 % of 可用性低下事象の発生 	
Getting Started	 パフォーマンスデータの収集 計測の展開計画の策定 サービスレベルの検討 	 % of New Relicの導入率 テレメトリーデータの量 	レンジャンジャン す アプリ 開発者 かた regergergergergergergergergergergergerge

データ分析を活用いただいているお客様事例

株式会社Mobility Technologies様 <u>https://newrelic.co.jp/resources/case-studies/japantaxi-jp</u>

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

江尻氏 JapanTaxiソフトウェアエンジニア

pr#-==-a-Atbox7242kEkstatuse New Relic One Øys2atter K

https://blog.newrelic.co.jp/customer-story/weathernews/

株式会社ウェザーニューズ様

STEP1

データの 理解

New Relicが取得するデータ

©2008–20 New Relic, Inc. All rights reserved () New Relic.

New Relicが取得するデータ: MELT



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

Eventとは

利用目的に応じた<u>Event名</u>と、複数の<u>属性(Attirbute)</u>から成る構造化データ New Relicが扱うデータの中核であり、以下の2つの手段で収集される

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

参考: Eventデータについて

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

Event名について

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

データソース	Event名	データの種類			
APM	Transaction	トランザクションの所要時間を記録			
	TransactionError	アプリで発生したエラーを記録			
Browser	PageView	ページがロードされた際の所要時間を記録			
	JavaScriptError	フロントエンドのエラーを記録			
Infrastructure(クラ ウド連携)	ComputeSample	計算リソースを提供するサービスのメトリックを記録 (AWS EC2等)			
	DatastoreSample	ストレージキャパシティを提供するサービスのメトリック を記録(AWS S3等)			

©2008-20 New Relic, Inc. All rights reserved **New Relic**. https://docs.newrelic.co.jp/docs/insights/insights-data-sources/default-data/insights-default-data-other-new-relic-products

18

属性(Eventの構成要素)について

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

https://docs.newrelic.co.jp/docs/using-new-relic/data/customize-data/collect-custom-attributes)。

Transaction Eventの例



属性(Eventの構成要素)について

Transaction

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

ハンズオン - 環境について

New Relicにログインしてください。

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

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

[ご注意下さい]

普段New Relicをお使いの方はセッションが残っている場合がありますのでプライベートブラウジングをお使いください。また、普段お使いのNew Relicで併せて操作いただいても問題ありません。

- Chrome:シークレットウィンドウ
- Firefox: プライベートウィンドウ
- Edge: InPrivate ウィンドウ
- IE: New Relicの一部機能はIEをサポートしていません。
 上記のいずれかのブラウザをご利用ください。

Log in to your account	🔘 New Relic.		
	Log in to you	r account	
EMAIL			
I	EMAIL		
	EMAIL		

ハンズオン

ハンズオン(1) データの理解

このハンズオンのセクションでは、以下の点を学習します。

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

Option!!



15:20 - 15:40 (20min)



ハンズオン(1) データの理解

イベントデータを確認する

New Relic OneポータルのUIIに触れ てみる

New Relic Oneにログインし、メ ニューを触ってみる

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

● New Relic ONE [™] All accounts ∽		Duery your data	⊞ Apps ⊘ Get started Q ⑦ 🛱 🔂 10+ 0	<u>ں</u> و
Explorer Browse data Dashboards Aler	erts & Al Errors Inbox APM Browser Infrastructure Logs Mobile Synthetics I	Nore v Co	py permalink 👻 + Create a workload + Add more da	ata
\Xi Filter by name, type, tags (e.g. entity	ty type = Hostj		I List	w
Favorite saved views View all	✓ Hosts View all (1)			
*	Name C Account C	Agent Version $\hfill \bigcirc$ CPU Utilization $\hfill \bigcirc$ Memory	Used 🗘 Disk Used Perc 🗘 🔋 Network Traffic 🗘	
Now you can save your filters as views and access them here.	★ ■ ip-172-31-26-144.ap-northeast-1.compute.internal NewRelicUniversi	1.20.7 3.32 %	54.99 % 39.56 % 105 kB/s …	
Your system				
☆ Favorite entities (1)				

ハンズオン(1) データの理解

イベントデータを確認する

イベントデータを確認するための手順 を実施する

New Relic Oneにログインし、 Explorerを参照する

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

All	l accounts ~		ハンズオン
(1) .	2, Search		
	Cross-account visibility	<u>()</u> _	
n A	All accounts		
w: N	Most recent al		または、
N	NewRelicUniversity-Japan 2511671		0
N	NewRelicUniversity-Japan 3139014	>_	Ouery your data
N	New Relic TSS 1336182		
NI-	:e		
New Relic ONE All accounts	v	D Query your dat	Instant Observability 끎 Apps ⓒ Get started 〇, ⓒ 및 당 101 🔘
Explorer Browse data Dashboar	ards Alerts & Al Errors Inbox APM Browser Infrastructure Logs M	obile Synthetics More - 🦉	🛱 Copy permalink 👻 + Create a workload + Add more data
😇 Filter by name, type, tags ((e.g. entityType = Host)		\
Favorite saved views View all	II V Hosts Viewall (1)		
*	Name 0	Account 0 Agent Version 0 C	CPU Utilization \bigcirc Memory Used \bigcirc Disk Used Perc \bigcirc Network Traffic \bigcirc
Now you can save your filters as views and access them here.	🜟 🔳 ip-172-31-26-144.ap-northeast-1.compute.internal	NewRelicUniversi 1.20.7	3.32 % 54.99 % 39.56 % 105 kB/s
Your system			
☆ Favorite entities (1)			

ハンズオン

ハンズオン(1) データの理解

イベントデータを確認する

イベントデータを確認するための Data explorerを利用する

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

どのようなデータを参照できるかを確認します

次のページに続きます

• Plotをクリックします

Data explorer Query builder							Copy permalink v Close
Events Metrics Logs Traces	Account: 2511671 - N	ewRelicUniversity-Japar	Filter to All entities ~				📇 30 minutes 🗸
Event type	^	SELECT	FROM	GROUP BY	LIMIT	WHERE	
Q. Search		count(*) \times	${\rm Transaction}~\times$		10		
New Relic events	1	NRQL SELECT co	unt(*) FROM Transaction	SINCE 30 MINUTES AGO	TIMESERIES		(3) –
AjaxRequest							
BrowserInteraction		Transaction					(👌 JSON Ra v data
nfrastructureEvent		{ "results": [
.og		{ "events":	C				
NetworkSample		("appI	1": 445000097,				
PageView		"appN "data	ame": "EC-site", paseCallCount": 2,				
PageViewTiming		"data "dura	<pre>baseDuration": 0.00019 tion": 0.05017,</pre>				
ProcessSample		"enti "enti	ty.guid": "MjUxMTY3MXx tyGuid": "MjUxMTY3MXxB	BUE18QVBQTE1DQVRJT0 UE18QVBQTE1DQVRJT05	58NDQ1MDAwMD 8NDQ1MDAwMDk	k3", 3",	
Span		"erro "guid	r": false, ": "5aa09dd74b167eb0",				
StorageSample		"host "http	": "ip-172-31-26-144.a .statusCode": 200,	p-northeast-1.compu	ite.internal"	<i>.</i>	
SyntheticCheck		"http: "name	ResponseCode": "200", ": "WebTransaction/Act	ion/block_search_pr	oduct",		
yntheticRequest		"prio "real	rity": 1.00043, AgentId": 445000099,				
oystempampie		"requ	est.headers.accept": " est.headers.host": "ec	text/html,applicati 2-3-113-215-132.ap-	on/xhtml+xml northeast-1.	,application/json;q=0. compute.amazonaws.com"	9,application/javascript;
Transaction		"requ	est.method": "GET", est.uri": "/ec-cube/in	dex.php/",			
		"resp	onse.neaders.contentTy onse.statusCode": 200,	pe": "text/html",			
ransactionError							
ransactionError		"samp "time	led": true, stamp": 1636265812673,				

ハンズオン(1) データの理解

目的:

イベントデータを確認する

イベントデータを確認するための Data explorerを利用する

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

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

									(17) an i
Events Metrics Logs Traces	Account: 2511671 - N	lewRelicUniversity-Japan ~	Filter to All (entities ~					30 minu
Event type	~	SELECT	FROM		GROUP BY	LIMIT	WHERE		
		count(*) ×	Transactio	in ×		10		6	3
Plot	^	NRQL SELECT count(*) FROM Tran	saction SINCE	30 MINUTES AGO TI	MESERIES		('	1)
Q Search									<u> </u>
count(*)		Transaction						4 🖬 🛛	JSO I Raw da
databaseCallCount	Average ~	TIMESTAMP 🗘		APP ID 🗘	APP NAME 🗘	DATABASE C	ALL COUNT 🗘	DATABASE DURATION	0 DURATION 0
databaseDuration	Average 🛩	November 07, 202	15:28:29	4450000	EC-site	25		0.00706	0.0667
duration	Average ~	November 07, 202	15:28:28	4450000	EC-site	2		0.0002	0.0488
error	Average 🛩	November 07, 202	15:28:27	4450000	EC-site	19		0.00492	0.0387
externalDuration	Average ~	November 07, 202	15:28:27	4450000	EC-site	50		0.012	0.0644
http.statusCode	Average 🛩	November 07, 202	15:28:24	4450000	EC-site	18		0.007	0.0761
priority	Average ~	November 07, 202	15:28:24	4450000	EC-site	11		0.00476	0.0707
request.headers.contentLength	Average ~	November 07, 202	15:28:22	4450000	EC-site	10		0.00296	0.0509
response.statusCode	Average ~	November 07, 202	15:28:21	4450000	EC-site	12		0.00376	0.0578
sampled	Average ~	November 07, 202	15:28:20	4450000	EC-site	10		0.00305	0.0338
totalTime	Average ~	November 07, 202	15:28:20	4450000	EC-site	12		0.00379	0.0447
		November 07, 202	15:28:18	4450000	EC-site	11		0.00527	0.0746
Dimensions	~	November 07, 202	15:28:18	4450000	EC-site	2		0.0002	0.0525
		November 07, 202	15:28:14	4450000	EC-site	2		0.0002	0.0535
		November 07, 202	15.28.10	4450000	EC-site	2		0.00035	0.0516

ハンズオン

ハンズオン(1) データの理解

カスタムイベントを送信する

カスタムイベントを送信するための事 前準備を実施します。

APIキーを作成します。

- NewRelicUniversity-Japanアカ ウントを指定し、Inget - License キーを作成する
 - NameやNotesには、イニ シャルや企業名などを用 いて重複を避ける様に設 定して下さい。

Q ⑦ ₺ 10+ ◎ t started 83% NRU-User iapan-handson+2021@newrelic.com Copy permalink API kevs := List Account settings **View Settings** O Memory Use... O Disk Use Theme Light Dark Auto 55.47 % NRQL console Show Hide Add more data Manage vour data View your usage Support Logout Create a key

ハンズオン

Create an API key

Ingest keys are for getting data into New Relic:

 License keys for agent configuration and metric, event, log and trace APIs

オプション演習

- Browser keys for browser applications
- Mobile keys for mobile applications

User keys are for querying data and managing configurations (Alerts, Synthetics, dashboards, etc.) To learn more about API keys, see our docs \square

	Account
	Account: 2511671 - NewRelicUniversity-Japan ~
	Key type
	Ingest - License 🗸
	Name
	my_key_name
	Notes
_ \	What do you want people to know about this key?
_ /	
	Cancel Create a key

参考:

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

ハンズオン(1) データの理解

カスタムイベントを送信する

作成したKeyの取得

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

Q Search keys by account, t	type, value, etc.				Create a key
Account 🗘	Туре 🗘	Name 🗘	Value 🗘	Notes	
NewRelicUniversity-Japan (INGEST - LICENSE	extra api key for optional h	-		
NewRelicUniversity-Japan	INGEST - LICENSE	License Key for NewRelicU	******	Copy key	
NewRelicUniversity-Japan	INGEST - BROWSER	Browser Key for NewRelicU	******	Edit	
NewRelicUniversity-Japan (INGEST - LICENSE	License Key for NewRelicU	******	Delete	
NewRelict Iniversity-lanan (Rrowser Key for NewPelic I	******	Original account browser	· kov

ハンズオン

オプション演習

補足:

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

ハンズオン(1) データの理解

- カスタムイベントを送信する
- ターミナルにて作業を行います。
 - Windows: Powershellを開く
 - Mac: ターミナルを開く



ハンズオン



ハンズオン(1) データの理解

カスタムイベントを送信する

Windows上での作業

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

Windowsの方はこちらです

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

ハンズオン

オプション演習

\$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"

ハンズオン(1) データの理解

カスタムイベントを送信する

Mac上での 作業

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

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

"nrulab.json"のあるディレクトリで実行する

Macの方はこちらです



ハンズオン

ハンズオン(1) データの理解

カスタムイベントを送信する

送信したカスタムイベントを参照しま す。

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

補足:トップ画面に戻ります。 または、 Query your data New Relic ONE[™] O New Relic ONE™ Account: 2511671 - NewRelicUniversity-Japan ~ 网些品 Get started ______ 83% 🔍 🖓 🛱 10+ 🔕 (i) New Relic CodeStream Debug prod and discuss code straight from your IDE. Browse data Dashboards Alerts & Al Errors Inbox APM Browser Infrastructure Logs Mobile Synthetics More -Copy permalink ~ + Create a workload Filter by name, type, tags... (e.g. entityType = Host) ∃ List A Navigator So Lookout Save as view Hosts View all (1) Favorite saved views View all Name ^ Account 0 Agent Version O CPU Utilizati... O Memory Use... O Disk Used Pe... O Network Tra... O tip-172-31-26-144.ap-northeast-1.compute.internal NewRelicUnive... 1.20.6 22.65 % 55.1 % 38 51 % 3 MB/s ... Now you can save your filters as views and access them here.

ハンズオン

ハンズオン(1) データの理解

カスタムイベントを送信する

送信したカスタムイベントを参照しま す。

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

New Relic ONE™ Data explorer Query builder							E∃ Copy perma	link ~ Close X
Events Metrics Logs Traces	Account: 2511671 - N	ewRelicUniversity-Japan	✓ Filter to All entities ✓				É	🖞 30 minutes 🗸
Event type	^	SELECT	FROM	GROUP BY	LIMIT	WHERE		
Q Search		count(*) ×	NRULab \times		10			
Custom events NrdbQuery		NRQL SELECT cou	nt(*) FROM NRULab SINCE	30 MINUTES AGO TIM	MESERIES		_	
NrIntegrationError		NRULab				2	🗐 🕗 jion	Raw data
NIMIDConsumption		1 0.95						
NRULab		0.9 0.85						
NrUsage		0.8						
PrivateNetworkSample		0.75 0.7						
Public_APICall		0.65						
QueueSample		0.55						
Relationship		0.45						
ServerlessSample	1	0.4						
SesRegionSample		0.3 0.25						
SqlTrace		0.2						
SyntheticsPrivateLocationStatus		0.15						
Plot	~	0 :00 PM	05:05 PM	05:10 PM	05:15 PM	05:20 PM	05:25 PM	05:3
Dimensions	~	 NRU Labs 						

ハンズオン

STEP2

分析手法 の習得

NRQL (New Relic Query Language)

©2008–20 New Relic, Inc. All rights reserved 🛛 🔘 New Relic,

NRQLとは

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



Data Explorer

• GUIで実データを見ながらデータを選択し、分析することができる機能

Data explorer Query builder							
Account: 1336182 - New Relic TSS ~ D	oata type Events	Metrics Logs 🗗 Trace	es 🗠				🛗 30 minutes ~
Event type	~	SELECT	FROM	GROUP BY	LIMIT	WHERE	
Plot	^	count(*) ×	Transaction ×		10		
Q. Search		NRQL SELECT count()	 FROM Transaction SINCE 30 MINU 	TES AGO TIMESERIES			
count(*)		Transaction					In In E D ISON Raw data
build	Average ~	1.7 k					
customer	Average ~	1.5 k	$\wedge \wedge$	\land	\wedge		
databaseCallCount	Average ~	1.4 k		\		$ \land \land$	
databaseDuration	Average 🗸	1.3 k		\vee		~ / / / / / / / / / / / / / / / / / / /	
duration	Average ~	1.1 k		V		\vee	V
error	Average ~	1 k					
externalDuration	Average ~	800					
items	Average ~	700					
memcacheDuration	Average ~	500					
offerUsed	Average ~	400	結果	を右メニューで	「表示		
port	Average ~	300	лч цл,				
price	Average ~	200					

見たいデータを左メニューで選択
クエリビルダー(Basic)

ドロップダウンでデータを選択し、分析することができる機能 •

View a chart wit	Enter an event type,	, attribute, keyword, or select a shortcut		Narrow results to								
Facet by Select	SHORTCUTS		1	Compare no comparison	View a chart with Records time - average	7208			~ =	E v Narrow results to Enforce	attribute of	vilue o
	APM Trans APM Resp APM Error	saction time ionse time r rate	>		Facet by name	×	Limit 10	⊟ Last 30 mins	~	Compare no comparison		⊗ Ck
	APM Thro Browser SPA I	ughput load times percentile	>		Since 30 minutes ago						Share	v
	Browser SPA I Browser SPA I Browser Brow	load times histogram throughput vser vs. Synthetics performance	>		55. 50 45 40							
	Host CPU Host Load	usage I average	>		35 30 25		~					
	Host Mem	nory usage	>		20 15 10	\sim		and a	X			
	見たい	データをドロッフタ	ウンで選		5 0 5 PM 01:00 PM	01:05 PM	0	1:10 PM	1:15 PM	01:20 PM		01:25
	仈				WebTransaction/Servlet/petclinic WebTransaction/Expressis/GET// WebTransaction/SpringController/ow WebTransaction/SpringController/ow WebTransaction/SpringController/ow	mers (GET) mers/find (GET) frome isn		WebTransaction/Spring WebTransaction/Spring WebTransaction/Spring WebTransaction/Expre WebTransaction/Expre	Controller/ Controller/ Controller/ ssjs/POST//	/owners/(ownerid)/edit (GET) /owners/(ownerid) (GET) /vets (GET) /api/order es found)		

216.98 thousand events inspected in 125 ms (1.7 MEPS)

クエリビルダー(Basic)の使い方



NRQLを直接書く

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

Account Demotron V2 V	Switching to basic mode will erase any changes you make here.	🗄 Basic
SELECT average(duration) FROM Transac	tion WHERE appName = 'Plan Service' TIMESERIES FACET `name` LIMIT 10 SINCE 1800 seconds ago EXTRAPOLATE	
My recent queries $\ arphi$	⊗ 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: <u>https://docs.newrelic.com/docs/query-data/nrql-new-relic-query-language/getting-started/introduction-nrql</u>

NRQL使いこなしTips①

- この句だけは覚えましょう!
 - FROM イベント名: どの名前のイベントから情報を収集するか
 - **SELECT 属性**: どの属性の情報を収集するか

※数値データは集計関数が使える (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Tips②

• 数値データの集計関数を覚えましょう!

平均值: 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)

- 最大值、最小值: max(属性), min(属性)
- 合計: sum(属性)
- 最新值: latest(属性)

ヒストグラム: histogram(属性, データ最大値, スロット数)

ある条件に合致するものの割合: percentage(関数(属性), WHERE 条件)

属性のバリエーション数のカウント: uniquecount(属性)

NRQL使いこなしTips③

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

ハンズオン(2) 分析手法の習得

このハンズオンのセクションでは、以下の点を学習します。

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



15:50 - 16:10 (20min)

ハンズオン

ハンズオン(2) 分析手法の習 得

目的:

NRQLの利用方法を取得する

ダッシュボード作成を通して NRQLの 利用方法を身につける

New Relic Oneにログインし、メ

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



ハンズオン(2)分析手法の習得

目的:

NRQLの利用方法を取得する

ダッシュボード作成を通して NRQLの利用方法を身につける

以下の点に注意して、ダッシュボードの作成を進めます

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

Createボタンをクリックします





NRQLの利用方法を取得する

ダッシュボード作成を通して NRQLの 利用方法を身につける

右の様な画面が表示されれば、準備が完了です



この画面が出てきたら、準備完了です。

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

ハンズオン

準備



ハンズオン(2) 分析手法の習 得

NRQLの利用方法を取得する

Data explorerで解析した結果をダッ シュボードに追加する

ハンズオン上のポイント

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

li	Q Search		
	Cross-account visibility		
r	All accounts		
0	Most recent)
	NewRelicUniversity-Japan	2511671	
	NewPolicUniversity Japan Dera	2120014	4
	New Relic TSS	1336182	

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

- 以下のデータを抽出します
- トランザクション総数を用います

(count(*))

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



NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerを起動し、解析する データを取得する

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



ハンズオン



Dimensions

Q. Search...

apdexPerfZone

appName

containerId

entityGuid

error

host

目的:

NRQLの利用方法を取得する

Data explorerで解析した結果をダッ シュボードに追加する

Data explorerを起動し、解析する データを取得する

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





Dimensi

Q Search

appld appName containerld

entityGuid

httpRespons

httpRespon name

request.hea

目的:

NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerを起動し、解析する データを取得する

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

าร	^						
	3						
	2						
	2						
	2						
	2						
	2						
	false	SELECT	FROM	GROUP BY	LIMIT	WHERE	
	2	count(*) ×	Transaction V	name. V	10	~	
de	5	NROL SELECT count(+)	EROM Transaction EACET na	THE STACE 30 MINUTES AGO	TIMESEDIES	2)	
essage	4	NAGE SECECT COUNT(*)	FROM THANSACCION FACEL HA	HE SINCE SU HINDLES AGE	TIMESERIES	y	
	11	Transaction					Ø JSON Raw data
	3	1.2 k					
accent	E	1 k 800	$\wedge \sim$	\sim	~ /	\wedge	
accept		600	\checkmark \lor \sim	~~	\checkmark		
contentLe	12	400			\sim	\sim	
			04:55 PM	05:00 PM		05:10 PM	05:15 PM
		04:50 PM			05:05 PM		
		04:50 PM • WebTransaction/Expres • WebTransaction/Expres	sjs/GET//	GED	• WebTransaction	Servlet/petclinic	
		04:50 PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring	sjs/GET// Controller/owners/{ownerld} Controller/owners (GET)	(GET)	05:05 PM • WebTransaction/ • WebTransaction/ • WebTransaction/	'Servlet/petclinic 'Expressjs/GET//api/restaurant JSP/WEB-INF/jsp/welcome.jsp	
		04:50 PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring	sjs/GET// Controller/owners/{ownerid} Controller/owners (GET) Controller/owners/{ownerid}/	(GET) edit (GET)	05:05 PM • WebTransaction/ • WebTransaction/ • WebTransaction/	Servlet/petclinic Expressjs/GET//api/restaurant JSP/WEB-INF/jsp/welcome.jsp SpringController/vets (GET)	
)	04:50 PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring	sjs/GET// Controller/owners/{ownerld} i Controller/owners (GET) Controller/owners/(ownerld)/	(GET) edit (GET)	05:05 PM • WebTransaction, • WebTransaction, • WebTransaction,	Servlet/petclinic Expressjs/GET//api/restaurant JSP/WEB-INF/jSp/welcome.jsp SpringController/vets (GET)	
	/	04:50 PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring NAME © WebTransaction/Expr	sjs/GET// Controller/owners/(ownerid) ; Controller/owners (GET) Controller/owners/(ownerid)/ essjs/GET//	(GET) edit (GET)	05:05 PM • WebTransaction, • WebTransaction, • WebTransaction,	Servlet/petclinic Expressjs/GET//api/restaurant JSP/WEB-INF/jsp/welcome.jsp SpringController/vets (GET) COUNT © 23.9 k	
		DdSD PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring NAME © WebTransaction/Expr WebTransaction/Serv	sjs/GET// Controller/owners/(ownerld) Controller/owners (GET) Controller/owners/(ownerld)/ essjs/GET// let/petclinic	(GET) edit (GET)	05:05 PM • WebTransaction. • WebTransaction. • WebTransaction.	Servlet/petclinic Expressjs/GET//api/restaurant JSP/WEB-INE/jsp/welcome.jsp SpringController/vets (GET) 23.9 k 12.6 k	
		Odso PM WebTransaction/Expres WebTransaction/Spring WebTransaction/Spring WebTransaction/Spring WebTransaction/Expr WebTransaction/Expr WebTransaction/Serv WebTransaction/Serv WebTransaction/Spring	sja/GET// Controller/owners/(ownerld) Controller/owners/(ownerld)/ Controller/owners/(ownerld)/ essjs/GET// let/petclinic ngController/owners/(own	(GET) edit (GET) nerfd) (GET)	05:05 PM • WebTransaction, • WebTransaction, • WebTransaction,	Servlet/petclinic Expressis/GET//api/restaurant JSP/WEB-INE/jsp.welcome.jsp SpringController/vets (GET) 23.9 k 12.6 k 1.08 k	
		DdSD PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Expr WebTransaction/Serv WebTransaction/Serv WebTransaction/Spring	sjørGETI/ Controller/owners/(ownerld) Controller/owners/(ownerld)/ Controller/owners/(ownerld)/ essjs/GET// let/petclinic ngController/owners/(own essjs/GET//api/restaurant	(GET) edit (GET) ner1d) (GET)	05:05 PM • WebTransaction/ • WebTransaction/ • WebTransaction/ • WebTransaction/	Servlet/petclinic Expressis/GET//api/restaurant JSP/WEB-INE/JSp/welcome_JSp SpringController/vets (GET) 23.9 k 12.6 k 1.08 k 655	
		0450 PM • WebTransaction/Expres • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Expr WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring	sjs/GET// Controller/owners/(ownerld) Controller/owners/(GET) Controller/owners/(ownerld)/ let/petclinic egcontroller/owners/(own essjs/GET//api/restaurant WEB-INF/sp/welcome.jsp	(GET) edit (GET) herld} (GET)	05:05 PM • WebTransaction/ • WebTransaction/ • WebTransaction/ • WebTransaction/	Servlet/petclinic Expressis/GETI/api/restaurant JSP/WEB-INE/JSp/welcome_JSp SpringController/vets (GET) 23.9 k 12.6 k 1.08 k 6555 599	
		0450 PM • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring • WebTransaction/Spring We	sja/GET// Controller/owners/(ownerd) Controller/owners/(GET) Controller/owners/(ownerd)/ let/petclinic gcontroller/owners/(own essjs/GET//api/restaurant WEB-INF/jsp/welcome.jsp ngController/owners (GET	(GET) edit (GET) herid) (GET)	05:05 PM • WebTransaction/ • WebTransaction/ • WebTransaction/	Servlet/petclinic Expressis/GETI/api/restaurant JSP/WEB-INF/JSpJ/welcome_JSp SpringController/vets (GET) 23.9 k 12.6 k 1.08 k 6555 599 597	

ハンズオン(2) 分析手法の習 得

NRQLの利用方法を取得する

Data explorerで解析した結果をダッシュボードに追加する

Data explorerで解析したグラフをダッシュボードに追加する

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

[次のスライドに進みます]





NRQLの利用方法を取得する

Data explorerで解析した結果をダッ シュボードに追加する

Data explorerで解析したグラフをダッシュボードに追加する

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

Widget title 試験用ウィジェット	
Select an existing dashboard	
We excluded dashboards you don't have p	ermission to edit.
$\boldsymbol{Q}_{\boldsymbol{x}}$ Search by name, account or creator	
Dashboard Name	Account
טערפיואר אוייט איין אוייט איין אוייט איין אייט איין אייט איין אייט איין אייט איין אייט איין אייט אייט	NCW I\CH
SP-SummerGames / SP-SummerGa	New Reli
SV-SummerGames / SV-SummerGa	New Reli
tanaka-20211104-test / tanaka-2021	NewReli
TRA-SummerGames / TRA-Summer	New Reli
VV-SummerGames / W-SummerGa	New Reli
テスト株式会社_khara / テスト株式会…	NewReli
蜆-dashboard / 蜆-dashboard	NewReli

Or create a new dashboard

×



ハンズオン

 View this on テスト株式会社 khara/ テスト株式会社 khara
 Dismiss

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

ハンズオン



目的:

NRQLの利用方法を取得する

ダッシュボードからNRQLを確認し、 編集する

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





NRQLの利用方法を取得する

ダッシュボードからNRQLを確認し、 編集する

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



Ð

Account: 1336182 - New Relic TSS ~

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





ハンズオン

		~
Account:1336182-New RelcTSS ~ SELECT count(-) FROM Transaction FACET name SINCE 12	HOURS AGO TIMESERIES	
0 ,AM 10:00 AM 11:00 AM 12:00 PM 01:00 PM 02:00 PM 0	03:00 PM 04:00 PM 05:00 PM 05:00 PM 07:00 PM 08:00 PM 09:00 PM	
WebTransaction/Expressio/GET// WebTransaction/SpringController/owners/(ownerld) (GET) WebTransaction/SpringController/vets/aurant WebTransaction/SpringController/vets (GET)	WebTransaction/Serviet/petclinic controller/storefront/click,item WebTransaction/SPWEB-NF/spAvelcome.jsp WebTransaction/SpringController/owners/find (GET)	
		Add to dashboard Open in Query Builder
3		
Add to da	Shboard Open in Query Builde	
	©2008–20 New Relic. Inc. All rights	reserved 🚺 New Relic.



ハンズオン(2) 分析手法の習 得

NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集 し、解析結果をダッシュボードに追加 する

クエリビルダー上で、NRQLを編集し ます

 Query your dataからクエリクエ リビルダーを開きます

(3)

- 右のNRQLをクエリビルダーに 入力します
- 右下にあるRunボタンをクリック し、チャートが描画されることを 確認します

1		
Data explor r Query builder		
Account: 2511671 - NewRelicUniversity-Japan ~ SELECT	PromQL-style	🗄 Basic
Add another query Your recent queries V Create alert	Clear	Run

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





NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集 し、解析結果をダッシュボードに追加 する

クエリビルダー上で、NRQLを編集し ます

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



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

Account: 1336182 - New Relic TSS ~			🕒 PromQL-style 🗄 Basic
SELECT count(*) FROM Transaction FACET appName SINCE	1 HOUR AGO WHERE appName IN [(]'Foo	dMe','New Relic Pet Clinic'[)]	
Add another query Your recent queries ~ Create alert			Clear Run
Since 1 hour ago		··· Chart n	ame Enter a chart name
APP NAME 🗘	COUNT 🗘	Chart ty	/pe Table ~
FoodMe	52.9 k		More visualizations in I/O L
New Relic Pet Clinic	33.3 k	Other a	groups ⑦

ハンズオン(2) 分析手法の習 得

NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集 し、解析結果をダッシュボードに追加 する

作成したチャートをダッシュボードに 追加する

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







NRQLの利用方法を取得する

クエリビルダー上でNRQLを直接編集 し、解析結果をダッシュボードに追加 する

作成したチャートをダッシュボードに 追加する

- 追加したチャートから View queryを選択し、SELECT句を 変更する
 - 応答時間の平均と90
 パーセンタイルを求める
 ように変更する

● 時系列データとして表示する

nce 1 hour ago		
APP NAME 🗘	cou	Expand Get as image
FoodMe	53	Get chart link
New Relic Pet Clinic	33.	Export as CSV
		Create alert condition
		View query
		Сору
		Edit
		Duplicate
		Delete

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

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

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

STEP3-1

ダッシュ ボードの 作成

New Relic One ダッシュボード

©2008–20 New Relic, Inc. All rights reserved () New Relic.

New Relic One ダッシュボード



ダッシュボード活用シーン

- みんなが見えるようにずっとモニタに映しておく (ご紹介した2社様の事例)
- キャンペーン中のリアルタイムな状況把握のために使う
- カスタマーサポートが問い合わせに応じて状況確認のために使う
- サービスに関わるチーム間ミーティングで共通認識を得るために
 使う
- 定期報告書代わりに使う

など

ダッシュボードが得意とすること

(データの選択と集約)

複数アプリケーションやアプリケーションとインフラのメト リックの相関関係など、様々なソースからのデータを一つ の画面で把握したい場合

- (データの加工)
 - チームで定めたKPIに対する実測値を把握したい場合
- (データのビジュアライズ) 集めたデータを目で見てわかりやすい形式で表示したい 場合

データのビジュアライズ

加工したデータを様々なチャートタイプで表示

Since 30 minutes ago	Share 😪	chart NAME Enter a chart name	
WebTransaction/Action/App\Http\Controllers\GetPlansController@getPlans	0.334	Chart type Bar	~
WebTransaction/Action/App\Http\Controllers\GetPlansController@getPlan	0.134	JSON	
WebTransaction/Action/App\Http\Controllers\TronController@statusMessage	0.0639	Area Bar	
WebTransaction/Action/App\Exceptions\Handler@render	0.0586	Bullet Billboard	
		Table Pie	
		Histogram Heatmap	

ダッシュボードの便利機能(1/2)

表示系

Note機能 任意の文字や画像をダッ シュボードに埋め込む

11月システム運用連絡事項	Top 5 web transactions Since 30 minutes ago
ビジネスイベント	300 %
 長示会 延期	
	250 %
<アイア 	200 %
又材放送 X日18時37分頃~	
z—16	150 %
イベントセール 17日~22日	100 %
〈ンテナンス	50 %
4日25時~25日5時まで システムB停止メンテナンス	0 %
「「語サイト」	06:40 PM 06:50 PM 07:00 PM
記法回闭	Action/block_se
	Action/product Action/block_cart

TVモード 全画面表示にする (オフィスのディスプレイに表示するなど)



ダッシュボードの便利機能(2/2)

分析系

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

(3インフラ分析(ホストを選択) Since 243 minutes ago until 238 minutes ago			CPU使用率(%) Since 243 minutes ago until 2	238 minutes ago		
HOSTNAME ()	ENTITY ID	APM APPLICA	100		- 10	~
ip-172-20-47-182	9139971800729872715		60 X220		VX	
ip-172-20-38-181	9001988261169175032		20			
host-tower-seattle	8706728668107720032	Tower-Seatt	0			
host-proxy-east-2	8588441772589877088	Proxy-East	10.29:00 AM 10.30:00	10:31:00 AM	10:12:00 AM	10.33.00 AM
host-proxy-west-7	8190076605337717640	Proxy-West	 host-tower-washington host-tower-indianapolis 	host-tower-portland host-tower-austin	 host-tow host-tow 	er-houston er-stockton
host-tower-philadelphia	8142382788567555619	Tower-Phila	host-tower-philadelphia host-tower-deliadelphia	host-tower-phoenix	host-tow	er-riverside

レインフラ分析(ホストを選択) ince 242 minutes ago until 237 minutes ago			CPU使用率(%) Since 242 minutes a	go until 237 mini	utes ago		
HOSTNAME	ENTITY ID 🔅	APM APPLICA	8				
ip-172-20-47-182 🛒	9139971800729872715		: ~~	~	$ \land \land$	m	~
ip-172-20-38-181	9001988261169175032		4	/		•	•
host-tower-seattle	8706728668107720032	Tower-Seatt	1				
host-proxy-east-2	8588441772589877088	Proxy-East	0				
host-proxy-west-7	8190076605337717640	Proxy-West	10:29:00 AM	10:30:00 AM	10:31:00 AM	10:32:00 AM	10:33:00
host-tower-philadelphia	8142382788567555619	Tower-Phila	• ip-172-20-47-182				

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		
Last 30 mins	Start date	Start time	-	Jun 18, 2020	02:34 pm
Last 60 mins					
Last 3 hrs	Asia/Tokyo			Cancel	Apply
Last 6 hrs		R	lecer	nt	
Last 12 hrs		Mary share to be seen		- history	
Last 24 hrs				y nistory yet :-)	
Last 3 days					
Last 7 days					
Custom >					

New Relic のObservability 成熟モデル

成熟度	特徴	КРІ	関わるロール
4 データ ドリブン	 データを基にした経営判断 マーケットへの素早い対応 CSAT/Net Promoterスコアの改善 	 顧客満足度 経営計画や経営判断と紐付いたSLOの活用 Time-To-Market (# of new products/features) 	
3 _{予测的}	 オブザーバビリティを用いた運用 顧客体験の改善 丁度良いスケーリング 	 % of 重大な障害の削減 % of エラーバジェットの利用数 % of SLOアラートの数 % of ビジネス指標のトラッキング数 	プロダクトオーナー
2 積極的	 MTTDの改善 サービスレベル改善のトラッキング デジタル顧客体験の改善施策の実施 	 平均MTTD % of SLOが定義されたサービス数 % of クリティカル・ケイパビリティの策定(DCX) 	プロダクト マネージャー
1 受動的	 MTTRの改善 アプリケーションパフォーマンスの把握 顧客に影響のある事象への対応の改善 	 平均MTTR % of パフォーマンス低下事象の発生 % of 可用性低下事象の発生 	
Getting Started	 パフォーマンスデータの収集 計測の展開計画の策定 サービスレベルの検討 	● % of New Relicの導入率 ● テレメトリーデータの量	アプリ 開発者 hts reports 運用者 66

目的に応じたダッシュボード



例1: サーバー稼働状況



目的に応じたダッシュボード



例2: ユーザー体験のリアルタイムモニタリング

主要ページのページロード時間 (Topページ) Since 1 hour ago	主要ページのページロード Since 1 hour ago	時間 (Phoneページ)		主要ページのページロード時間 (Planペ Since 1 hour ago	-אר (ע
0.981 / 5 DURATION (90%)	5.697 DURATION (9	/5 ^{00%)}		1.34 / 5 DURATION (90%)	
20%			100% 1145	27%	
平均ページロードタイム(ページ句) … sine 1 hour ago	ページロードタイム分散(4 Since 1 hour ago	ページ毎)			
WebPortal, http://webportal.teico.nrd WebPortal, http://webportal.teico.nrd	Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com Ntg://webportal.takco.mdemo.com	ubrowszipians ubrowszipians wi wiego jap ukoswa pianor 10001 ubrowszipians 10001 ubrowszipians 17324 ubrowszipians 17324			c mananan jara manga jara kang jara kang jara manga jara manga jara kang jara kang jara kang jara kang jara ka
ページロードタイム内訳平均時間(ページ毎) Since 1 hour ago					
PAGE URL 🛇	TOTAL 0	BACKEND	NETWORK	DOM PROCESSING	PAGE RENDERING
http://webpurtal.telco.virdemo.com/browse/phones/99995643	5.69	1.62	0.0368	4.04	0.0235
http://webportal.telco.nrdemo.com/browse/phones/99912353	5.6	1.54	0.0171	4.03	0.0245
http://webportal.te/co.nrdemo.com/browse/phones/67424491	5.59	1.65	0.0958	3.92	0.0187
http://webportal.telco.nrdemo.com/browse/plans	1.21	0.795	0.0469	0.385	0.0342
http://webportal.telco.nrdemo.com/checkout	1.18	1.02	0.121	0.131	0.0259
http://webportal.telco.ordemo.com/browse/plans/100243	1.14	0.802	0.0127	0.282	0.056

68

目的に応じたダッシュボード



例3: ECサイト購買分析

サイト訪問者 (約日比) Since 5 minutes ago compared with 1 day earlier		サイト訪問者 / 週 Since 1 week ago compared with 1	能先上版 / 通 Since 1 week ago	購入までのページ連移率 Since 1 week ago	
10 10			33.1M /	100%	App Launch 65154 Sessions
: ON MARAN		602 ▼ <0.1%	100M	89.9%	Phone List 58590 Sessions
		Current Users	#	65.2%	Shopping Cart 42494 Sessions
11:39:00 PM 11:42:00 PM 11:41:00 PM 11:42:00 PM Current Unique Users Previous Unique Users			33 ⁴ W	28.4%	Payment 18521 Sessions
クラッシュ率 Since 1 week ago compared with 1 Since 1 week ago		最後に表示されたページ (クラッシュあり) Since 1 week ago		クラッシュ評議 Since 1 week ago	
		Display PhonesListFragment	8.43 k	PhonesListFragment.java line 266 in com.nrd 1.26 k	Go to Crash Analysis
2 19 % 66	2 k	Display PlansListFragment	8.22 k	PlansListFragment.java line 220 in com.nrde 1.18 k	
Current % 円		Display LoginFragment	175		
購入までのページ遷移車 (レスポンス<1s) Since 1 week ago		購入までのページ運移率 (レスポンス>=1s) Since 1 week ago		最後に表示されたページ(クラッシュなし) Since 1 week ago	
100%	App Launch 25783 Sessions	100%	App Launch 39580 Sessions	Display HomeFragment 200 k	Go to Mobile Interactions
91.3%	Phone List 23531 Sessions	49.1%	Phone List 19414 Sessions	Display ShoppingCartListFragment 132 k	
78.7%	Shopping Cart 20294 Sessions	2.79%	Shopping Cart 1105 Sessions	Display LoginFragment 94.5 k	
33.7%	Payment 8696 Sessions	0.432%	Payment 171 Sessions	Display PlanFragment 39.1 k	

reserved 🔘 New Relic.

69

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

現在までの受注件数到達度>1日 3500 件目標 Since 1 week ago	時間毎の先上額推移 Since 1 day ago compared with 1 day earlier	-
56 / 3.5k	4001 3001 3001 3001 3001 3001 3001 3001 3001 4 12007Au 4 12007Au 4 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007Au 10007 10007Au 10	u 12007Au ditto:7Au ditto:7Au
	Current Revenue Previous Revenue	
アクセスユーザー数(第日比) Since 1 day ago compared with 1 day earlier	エラーの影響を受けたユーザー数 Since 1 day ago	エラーの影響を受けた認先上調 … Since 1 day ago
18.8 k a 20.8 % Current Site Visitors	340 Users Experiencing Errors	41.3 k Revenue
アクセスユーザー数 推移 Since 1 day ago	關入不成立件数 Since 75 minutes ago	購入までのページ通修率 … Since 1 week ago
250 200 150 50 50 50 50 50 50 50 50 50 50 50 50 5	13 (%) Carts Left Without Purchase	Home Home 66853 Sessions 66853 Sessions 86.6% Browse 578.6% Viewed Cart 38734 Sessions 38734 Sessions
Usernames		24.2% Purchased 16191 Sessions

New Relic.
 ⁷⁰

さらに高度なビジュアライズも

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



オリジナルのダッシュボードを作成可能

環境をセッティングして開発

ハンズオン(3) ダッシュボードの作成

このハンズオンのセクションでは、以下の点を学習します。

- S3ダッシュボードのチャートを利用する
- S3のチャートをカスタマイズする
- NOTE機能を利用する
- チャートのサイズや配置を変更する

□ption!!● NRQLを用いてチャートを作成する



16:15 - 16:35 (20min)


ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

S3ダッシュボードを用いて、ダッシュ ボードをカスタマイズする

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

D						鳼	ҍ備
New Relic ONE ^{**} All accounts ~		🗊 Query your data	네: Instant Observability	🗄 Apps	⊘ Get starte	1005	2 🕫 10+ 🔘
Dashboards - / ☆ テスト株式会社_khara - NewRelicUniversity-Japan 🛛 🖓	8		En Copy permalir	ik v 🖵	ቀ ⊉ 🗅	_⁄ ⊚ +	🕒 🕒 Default 🗸
च ✓ Search for any attribute or value.							
試験用ウィジェット Since 3 hours ago	クエリビルダーより追加したチャート Since 1 hour ago						
2.5 k	APP NAME 🗘	COUNT 🗘					
1.5k ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	FoodMe	52.8 k					
1k 500	New Relic Pet Clinic	33.2 k					
0 I8:00 PM 08:30 PM 09:00 PM 09:30 PM 10:00 PM 10:30 PM 11:00 Pf							
WebTransaction/Zepresjs/GET// WebTransaction/Servlet/petclinic WebTransaction/Servlet/petclinic WebTransaction/Servlet/Petclinic WebTransaction/Servers/Servlet/Net/servlet/ WebTransaction/Servlet/Servlet/Net/servlet/ WebTransaction/Servlet/Petclinic WebTransaction/Servlet/Petclinic							

Infrastructure \rightarrow AWS \rightarrow S3 dashboard







1)

(2)

(3)

目的:

ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

S3ダッシュボードを用いて、ダッシュ ボードをカスタマイズする

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

All requests				
Since 1 hour ago.Japan (UTC 9)				
1	Expand			
0.8	Get as image			
0.6	Ceteborellisk			
0.4	Copy to query builder			
0.2	·			
0				
Add to Dashboard	クエリビル? も必要あり	ダー上 ⁻ ません	ଟେମ୍ବ	変更は何
Add to Dashboard	クエリビル も必要あり	ダー上 ⁻ ません	ଟେଡ଼ି	変更は何 ^{535ッシュポードのAll reques} Since 6 hours ago
Add to Dashboard	クエリビル も必要あり	ダー上 ⁻ ません	ሮወ፤	変更は何 変更は何
Add to Dashboard	クエリビルチ も必要あり ^{クエリビルチーより強烈したチャート} Since 1 hour ago RP NAME C FoodMe	ダー上 ⁻ ません ^{COUNT 0} 525 k	ଟେଡ଼ି	変更は何 変更は何 510ce 6 hours ago 4 35 35 32 32
Add to Dashboard	クエリビルチ も必要あり ^{クエリビルチーとり強いしたチャート} Since 1 hour ago RP NAME C FoodMe New Relic Pet Clinic	ダー上 ⁻ ません ^{count 0} 525k 33.3k	ଟେଡ଼ି	変更は何 変更は何 Since 6 hours ago 4 35 35 35 35 35 35
Add to Dashboard Add to Dashboard Image: Add to Dashboard <	クエリビルチ も必要あり ^{クエリビルチーとり強いしたチャート} Since 1 hour ago RP NAME C FoodMe New Relic Pet Clinic	ダー上 ⁻ ません ^{count 0} 525k 33.3k	ල ග : 	変更は何 変更は何 Since 6 hours ago 4 35 25 25 25 25 25 25 25 25 25 25 25 25 25

08:00 PM 09:00 PM

10-00 PM



ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

S3ダッシュボードを用いて、ダッシュ ボードをカスタマイズする

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

Since 6 hours ago	
4	Expand
3.5	Get as image
3	Get chart link
2.5	Create alert condition
1.5	View query
	Сору
0	Edit
00 PM 06:00 PM 07:00 PM 08:00 PM 09:00	Duplicate
	Delete

	\bigcirc	
Account: 2511671 - NewRelicUniversity-Japan		🔘 PromQL-style 🛛 🖹 Basic
SELECT sum(`provider.allReq since 6 hours ago TIMESERI	uests.Sum) from DatastoreSample v ES auto facet entityName	where provider='S3BucketRequests' where providerAccountId = '26802'
Add another query Your recent querie	s 🐱 Create alert	Clear Run

ハンズオン



ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

S3ダッシュボードを用いて、ダッシュ ボードをカスタマイズする

 Requestの対象をallからgetに 絞る (属性 名:provider.getRequests.Sum)

- データ参照時間を1日に変更する
- Save as...ボタンを押し、ダッシュボードに保存します



ハンズオン



目的:

ダッシュボードを作成する

既存のUIを用いてダッシュボードを力 スタマイズする

作成したチャートの表示スタイルを変 更する

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

[次のスライドに続きます]



ハンズオン



目的:

ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

作成したチャートの表示スタイルを変 更する

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







ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

ダッシュボードにテキストや画像など チャート以外の情報を追加する

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

[次のスライドに続きます]





ダッシュボードを作成する

既存のUIを用いてダッシュボードをカ スタマイズする

ダッシュボード内の配置やサイズを変 更します

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





ダッシュボードを作成する

ダッシュボードにチャートを追加する

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



以降のスライドにて、この手順を用いて チャートの追加を行います

ハンズオン(3) ダッシュボード の作成

- 目的に沿ったダッシュボードを 作成する
- 過去1週間の稼働率を表示し、前週と 比較する
 - アカウント: New Relic TSS
 - FROM句: SyntheticCheck
 - SELECT句: percentage()

クエリ例:

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

解説:

- 外形監視Syntheticsのチェック結果である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ていま す。
- AS句を使うと、クエリ結果をダッシュボードで表示させる際に、わかり やすい表示に置き換えることができます。
- COMPARE WITH句を使ってその前の週と比較しています。



ハンズオン

受動的

ハンズオン(3) ダッシュボード の作成

 目的に沿ったダッシュボードを 作成する

URLごとのページ表示時間のヒストグ ラムを作成する

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

クエリ例:

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

解説:

- Browserエージェントによって計測された PageViewイベントから、応答時間を抽出しています。
- histogram()関数を用いて、応答時間10秒を20個のウィンドウに分割し、それぞれにウィンドウに当てはまる応答時間をカウントしています。
- FACET句を使い、pageUrl毎にヒストグラムを作成しています。



ハンズオン(3) ダッシュボード の作成

- 目的に沿ったダッシュボードを 作成する
- ユーザーのページ遷移の状況を解析 し、ファネルで表示する
 - アカウント: New Relic TSS
 - FROM句: PageView
 - SELECT句:funnel()

クエリ例:

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

解説:

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



ハンズオン(3) ダッシュボード の作成

- 目的に沿ったダッシュボードを 作成する
- ハンズオンで実施したカスタムイベン トの一覧を作成する
- アカウント:

注意!! NewRelicUniversity-japan

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

クエリ例:

SELECT initial, comment FROM NRULab since 1 day ago

解説:

 ハンズオンで送付したカスタムイベントの NRULabイベントから必要な 情報を抽出しています。



STEP3-2

アラートの 作成

高度なアラート設定

©2008–20 New Relic, Inc. All rights reserved () 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

1. Categorize	APM - Application metric
2. Select entities	1 entity
3. Define thresholds	TomcatApp \sim
Response time (web) v has an average above v	2.5 ms
 0.001 sec at least once in 5 minutes Add a warning threshold Condition name 	2 ms 1.5 ms
Response time (web) (High)	Oct 03,
Add runbook URL	Critical threshold Critical violation
	Back to Select entities Create condition

87

アラートの通知と確認

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

Incident #83551393 opened

Target

SystemSample query

Condition

NRQLテスト

Policy Tomtest

Threshold

SystemSample query result is > 3

(8 kB) -



アラートの詳細をNew Relic上で確認

Plan Service violated Check response time for web portal



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

アラートのカスタマイズ

(!) 3 critical violations

- NRQLの結果をしきい値にアラートを設定することも可能
- ✓ Define your signal



- アラートの設定画面で直接 NRQL
 を記述します
- <u>SELECT句, FROM句および</u>
 <u>WHERE句(オプション)のみ</u>で記述 します
- Ulを確認しながら、閾値の設定を 行えます

Signal loss violations and filled data gaps are currently not reflected in the chart. See our docs 🗹

ハンズオン(4) アラートの作成

このハンズオンのセクションでは、以下の点を学習します。

アラートのためのコンディション作成



16:40 - 16:50 (10min)

ハンズオン(4) アラートの作成

- アラート設定のための手順を理 解する
- アラート設定を行うUIにアクセスする
 - メニューより、Alerts & Alをク リックする
 - 左ペインからAlert condition(Policies)をクリックする
 - ダッシュボードハンズオン用ア ラートポリシーをクリックする

[次のスライドに続きます]



ご注意事項: Alerts & AIのUIが変更中の状況でございます。

ハンズオン(4) アラートの作成

 アラート設定のための手順を理 解する

ポリシー内にコンディションを追加しま す

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

[次のスライドに続きます]



ハンズオン



 アラート設定のための手順を理 解する

ポリシー内にコンディションを追加しま す

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

Save condition

Cre	eate an alert condition ount: 2511671 - NewRelicUniversity-Japan Policy: 1065477 - ダッシュボードハンズオン用ア	×
~	Enter condition name Enter a name that's easy to identify	
~	Define your signal Enter NRQL Query ①	
	SELECT For help with null values ビ', loss of signal ビ', or other query options, see our docs 드'.	
(2)	
EI	LECT percentile(duration,95) FROM PageView WHERE appName = 'EC-si	ite'
(3	

ついてご興味のある方は、

©2008–20 New Relic, Inc. All rights reserved 🛛 🔘 New Relic.

コンディションを設定するためのさまざまな項目に

是非、NRU304のウェビナーをご受講下さい。

お疲れさまでした!!

ご質問がございましたら、チャットにご記入ください。

©2008–20 New Relic, Inc. All rights reserved () New Relic.

ぜひNew Relicをお試しください!

弊社サイトからサインアップ <u>https://newrelic.com/signup</u>

New Relic.

Free access to all of New Relic. Forever.

With your new account, you get:

Perpetually free access 100 GB/month of free data ingest. 1 free full access user. Unlimited free basic users.

One data platform for all metrics, logs, events, and traces Petabyte scale. Millisecond speed. Pennies per gigabyte (beyond free tier).

Easily visualize, analyze & troubleshoot your entire stack APM, Infrastructure Monitoring, Digital Experience Monitoring, Applied Intelligence, and more.

NAME

e.g. Katherine Johnson

EMAIL

name@company.com

By signing up, you're agreeing to Terms of Service and Services Privacy Notice.

Sign up

Have an account? Log in

NRQL Lessons - NRQL学習ツール



©2008–20 New Relic, Inc. All rights reserved **O** New Relic.

New Relic University

簡単にはじめる

New Relic One インストール方法を製品内部でわかりやすくガイド

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

トレーニング、認定資格、書籍など知識へのアクセス

無償トレーニング

NRU Practitioner | Fundamentals オブザーバビリティの基礎知識を学習 -オブザーバビリティ入門

NRU 10X | Fundamentals

New Relic One の製品概要と基礎知識を学習 -New Relic One 入門

NRU 20X | Product Basics (11月公開予定)

New Relic One の各機能を学習 -Full Stack Observability 入門 -Applied Intelligence 入門 -Telemetry Data Platform 入門

Japanese Document | Document New Relic One の全機能利用解説ドキュメントは日本語 -docs.newrelic.com

Guided Install | In-Product

-Sign-up 方法と初期設定ガイド

-7言語ごとのAPM計装ガイド -OSごとのInfrastructure計装ガイド

-OSごとのMobile計装ガイド

-Browser計装ガイド

-Svnthetics計装ガイド

Install Guide | Document & Video

New Relic Book | Paper Book 物理的な書籍としてもNew Relic One 活用方法ガイドを提供 -New Relic 実践入門 (翔栄社)

NRU 30X | Hands-On

SLI/SLO 設計やダッシュボード開発、AlOps の最適なアラート設計などを学習 -コア機能実践ハンズオン -SLI/SLO 設計ハンズオン -AlOps とアラート設計ハンズオン -ダッシュボード開発とNRQL実践ハンズオン

NRU 40X | Advanced Hands-On (AWS特設ページ内限定開備 AWS など環境特化での実運用を学習 -New Relic One for Amazon ECS -New Relic One for Amazon EKS New Relic One for Amazon EKS

認定試験

Full Stack Observability | Exam (英語限定) Full Stack Observability の基礎知識を試験

Programmability | Certification (英語限定) Programmability スキルを認定

New Relic 実践入門

希望者に無償提供中(12月15日まで)

ついに発売された New Relic の全てを理解できる 332 ページにわたる技術書籍。オブザーバビリティの 基本から New Relic One の基本機能、さらには 16 のオブザーバビリティ実装パターンまで含めた、初心者 から応用を理解したい上級者まで対象にした New Relic のパーフェクトガイドブック。

無償提供希望はこちらのGoogle Form から

https://forms.gle/jqiYmWRYt8Hf1nHk8

アフリケーション開発者 フロントエンド開発者 モバイルアフリ開発者 インフラ管理者 プロジェクトマネージャー プロダクトマネージャー

SHOFISHA

New Relic.

New Relic 実践入門

監視からオブザーバビリティへの変革

松本 大樹、佐々木 千枝、田中 孝佳、伊藤 覚宏、清水 毅、 齊藤 恒太、瀬戸島 敏宏、小口 拓、東 卓弥、会澤 康二 🕷

^{##1/10/02/-17/79###126} すべてを観測し、開発の高度化と 信頼性を高める運用を実現する

 AWS
 Azure
 GCP
 モバイル
 Serverless
 Kubernetes

 ログ管理
 高精度アラート
 AlOps
 SRE
 DevOps
 OSS 連携

 C
 Go
 Java
 .NET
 Node,js
 PHP
 Python
 Ruby

New Relic User Group

New Relicを利用するユーザー同士交流するためのグループです。 12月15日(水) 16:00~にて、第2回目のユーザ会開催予定となっています。

参加申し込み: https://nrug.connpass.com/event/230940/



最後となりますが、 是非、アンケートへのご協力をお願い致します。

また、もっと詳しい話を聞きたい方は、 その旨アンケートにご記載ください

Thank You

