



NRU 401

IDEと連携し、問題解決を加速する New Relic活用の実践

2025/03/19

15:00より開始致します。しばらくお待ちください。

音声聞こえづらい場合は退出、再入室をお試ください。



ウェビナー 各種ご連絡

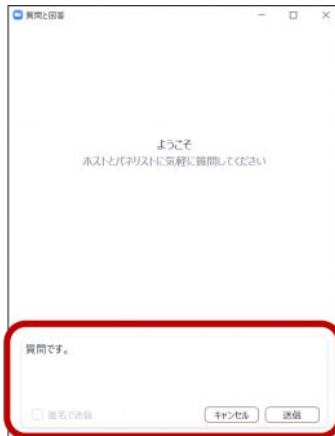
1. ご質問がある場合は、「Q & A」からご入力ください。



① 画面下
「Q&A」をクリック！

こちらにご質問をご記入し、
「送信」をクリックしてください！

②



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

Safe Harbor

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

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

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

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

本日のゴール

CodeStreamを活用して“**One Team**” を実現し、
顧客満足度の高いサービス開発 する方法を理解

- 開発において稼働中の**システムの状態** を常に意識
- 開発に必要なデータをIDEに集約し**データ分析を効率化**
- 不具合対応における**SREと開発者の円滑なコラボレーション**

アジェンダ

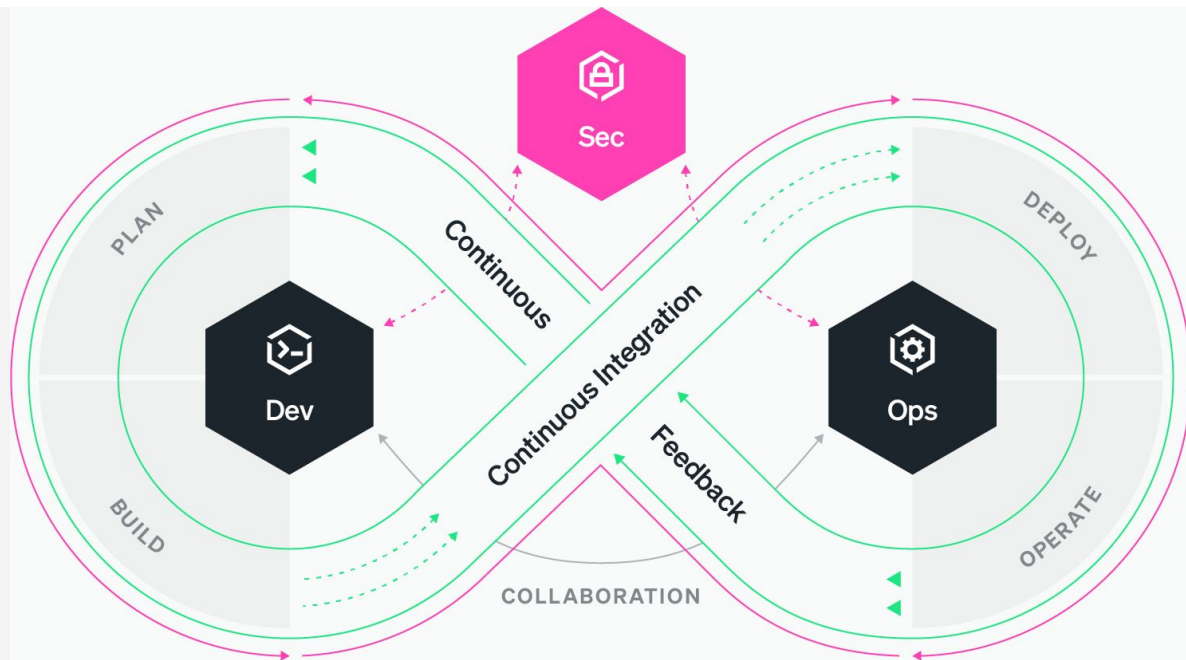
時間(目安)	内容	
15:00 – 15:30	座学	CodeStreamとは？
15:30 – 15:45	ハンズオン(0)	CodeStream環境の準備
15:45 – 16:05	ハンズオン(1)	開発者がデプロイされたシステムの状態を確認
16:05 – 16:25	ハンズオン(2)	システム状態をIDE上から分析
16:25 – 16:45	ハンズオン(3)	開発者と運用者のコミュニケーション
16:45 – 17:00	まとめ、アンケートご記入	

座学: CodeStreamとは？



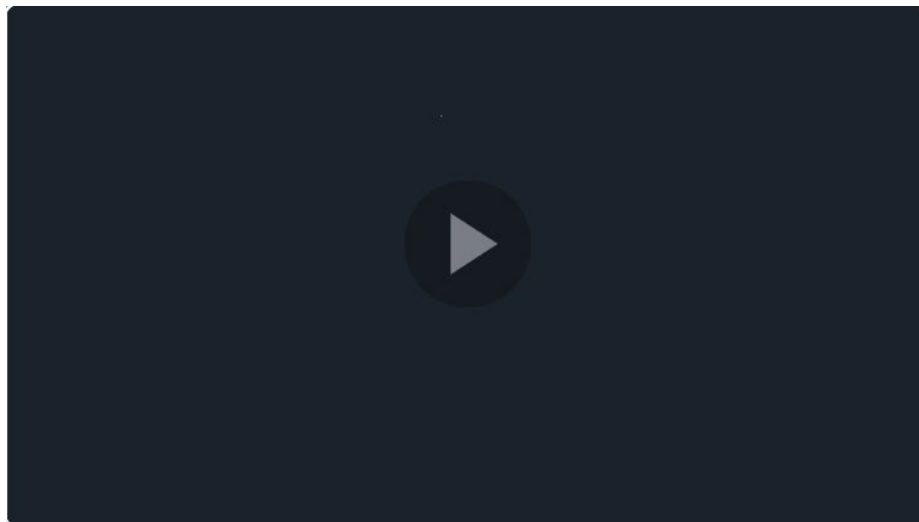
New Relic - Full Lifecycle Observability

よりよいソフトウェア開発
DevSecOpsの実現のために
O11y プラットフォームを提供



Data for Engineers

**Focusing on
engineers and
their success**



[Our New Brand Announcement](#) | [New Relic](#) | [New Relic](#)

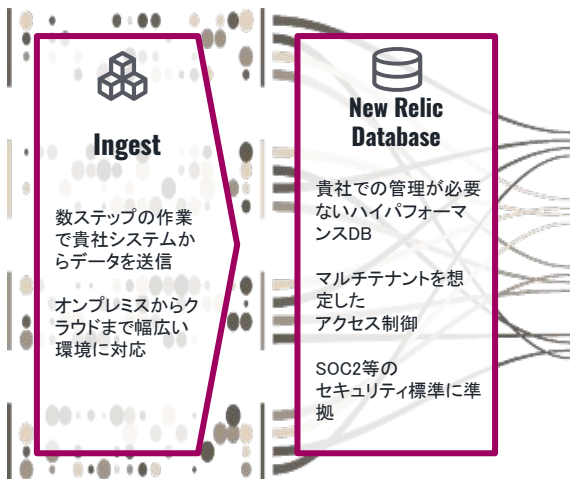
new relic オールインワン オブザーバビリティプラットフォーム

Data Cloud – データ格納

分間数十億件のデータを取り込み、自動で関連付け

Ingest & New Relic Database

システムのあらゆる場所からデータを集約

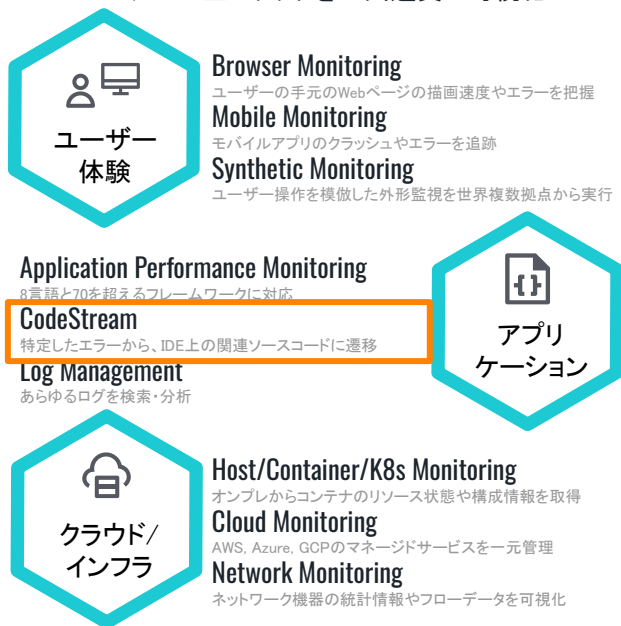


Insight Cloud – データ活用

AIベースのリアルタイム分析を、あらゆる場所、あらゆるアプリケーション開発サイクルで実行可能

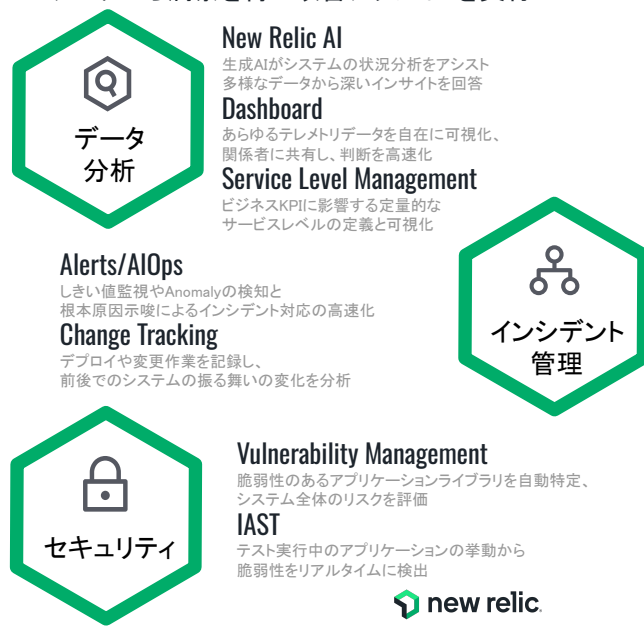
Full-Stack Monitoring

システムの全スタックを一気通貫で可視化



Observability Experience

データから洞察を得て改善アクションを実行



CodeStreamを活用しよう！

1

開発者がデプロイされた システムの状態を確認

IDE上でシステムの
パフォーマンスや脆弱性等を
常に確認しながら開発

2

システム状態を IDE上から分析

IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の コミュニケーション

デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

CodeStreamを活用しよう！

1

開発者がデプロイされた システムの状態を確認

IDE上でシステムの
パフォーマンスや脆弱性等を
常に確認しながら開発

2

システム状態を IDE上から分析

IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の コミュニケーション

デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

開発チームは開発で忙しい！



開発チーム

よし！デプロイされた！これでOK！

早く次の開発に取り掛かろう！

設計して、開発して、テストして..

よし！デプロイされた！これでOK！

早く次の開発に取り掛かろう！

設計して、開発して、テストして..

よし！デプロイされたしこれでOK！

問題なく動いているようだけど..

ほんとのところどうなんだろう？

パフォーマンスとか大丈夫かな？

動いているシステムも気にしてほしい。。

とはいえ、簡単に状況共有できない。。



運用チーム

CodeStreamでIDE上でシステムの状態を確認



開発チーム



手軽にIDE上で確認できて嬉しい！

NEW RELIC CODESTREAM

tomorinru

OBSERVABILITY catalogue

catalogue_nru NRU401 (APM)

Golden Metrics

Response time	131.41 ms
Throughput	2.86 req/m
Error rate	39.53 %

Errors

Assigned to Me

No error:

Recent

Exception

Exception

Exception

Exception

Exception

Exception

avg duration: 1281.428ms | error rate: 0.00% - 33 samples in the last 30 minutes

Method level metrics confirmation

```
def call
  @socks = Sock.all
  if @tags.find { |tag| tag == Tag.find(SockTag.find(
    |sock| @tags.find { |tag| tag == Tag.find(SockTag.find(
  })
  else
    @socks = Sock.all
  end
  if !@pageSize.nil?
    @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @
  end
  tazuma, a year ago · fix
  return @socks.map { |sock| {
    "id": sock.sock_id,
```

-- NORMAL -- 行 23、列 7 スペース: 2 UTF-8 LF Ruby Prettier

CodeStreamを活用しよう！

1

開発者がデプロイされた
システムの状態を意識

IDE上でシステムの
パフォーマンス等を
常に確認しながら開発

2

システム状態を
IDE上から分析

IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の
コミュニケーション

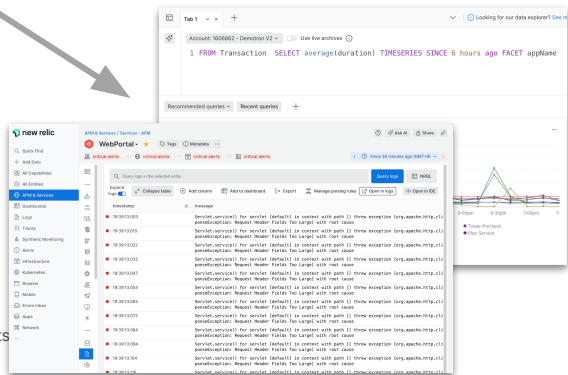
デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

複数の画面を開いて ...

IDE でコードを確認する

[illegible]

New Relic で確認する



IDE上でメトリクスやログのデータを確認！

IDE でコードを確認する

[illegible]

```

# In the development environment your application's role is irrelevant
# so we're going to set the role to 'dev' just to get things running
# You can't just leave this to 'null' as the server won't start without a role
config.role = 'dev'

# Do not want to log on to the database
config.logger.log_on = false

# Show full error reports
# config.consider_all_requests_local = true

# Disable debugging, by default debugging is disabled
# You can't just leave this to 'null' as the server won't start without a role
# If you're not going to use the debugger, config.debugger.config['disabled'] = true
# config.action_controller.enable_forgery_protection_logging = true

# config.cache_store = :memory_store
# config.cache_store = :redis_store, {url: 'redis://localhost:6379/1'}
# "Cache-Control" is public, no-cache (see config.action_controller.cache_control_headers)
# config.action_controller.perform_forgery_protection = false

# config.cache_store = :null_store

end

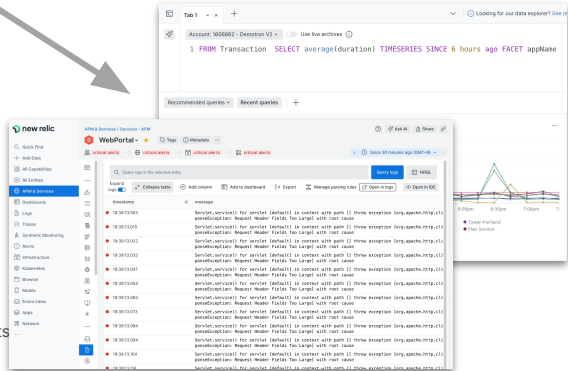
# Don't want to use the rails mailer service (see config.action_mailer.perform_caching)
# config.action_mailer.service = :local

# Don't care if the rails mailer doesn't work
# config.action_mailer.raise_delivery_errors = false

```



New Relic で確認する



new relic.
CodeStream



IDE上で直接確認できて便利！

CodeStreamを活用しよう！

1

開発者がデプロイされた
システムの状態を確認

IDE上でシステムの
パフォーマンスや脆弱性等を
常に確認しながら開発

2

システム状態を
IDE上から分析

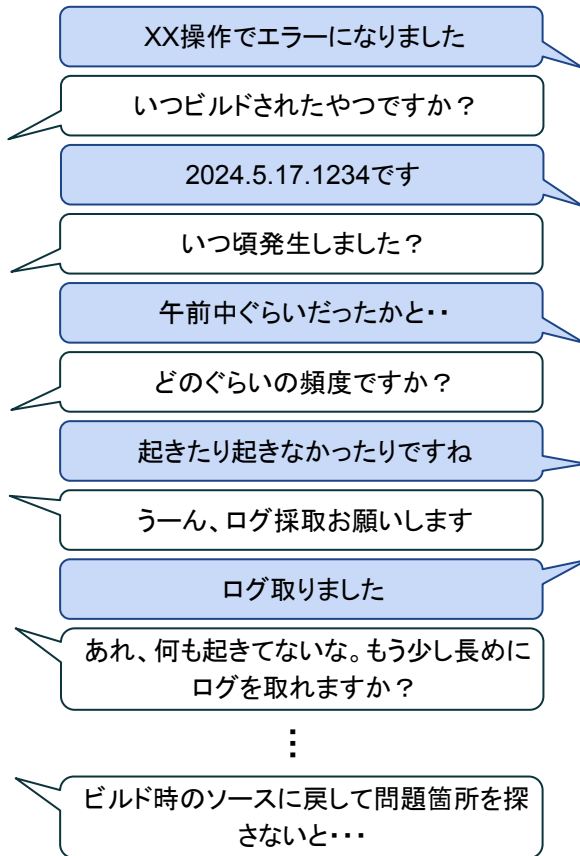
IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

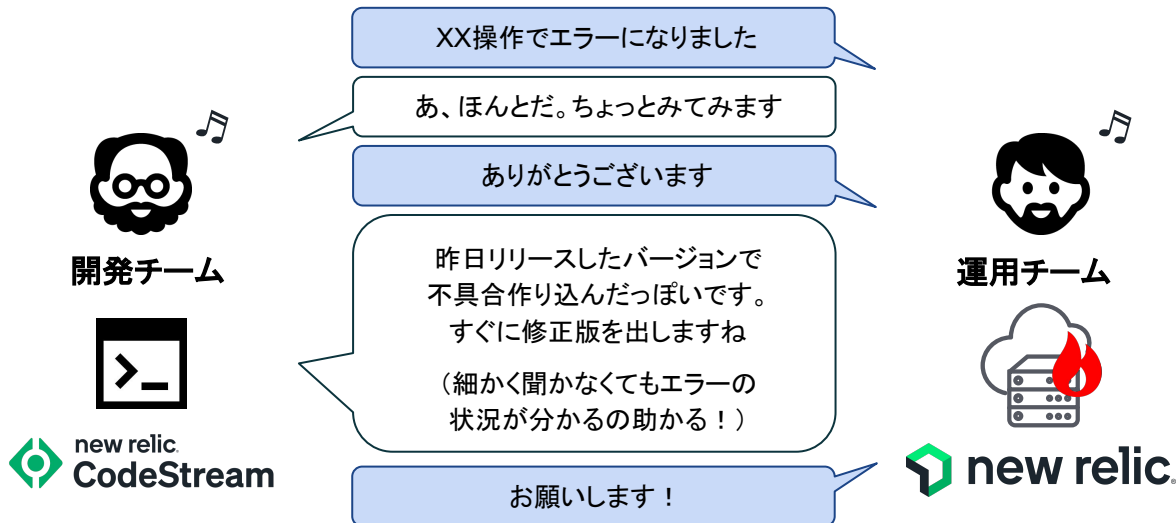
開発者と運用者の
コミュニケーション

デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

時間のかかる不具合対応 ...



IDEで問題コードを速攻で特定！

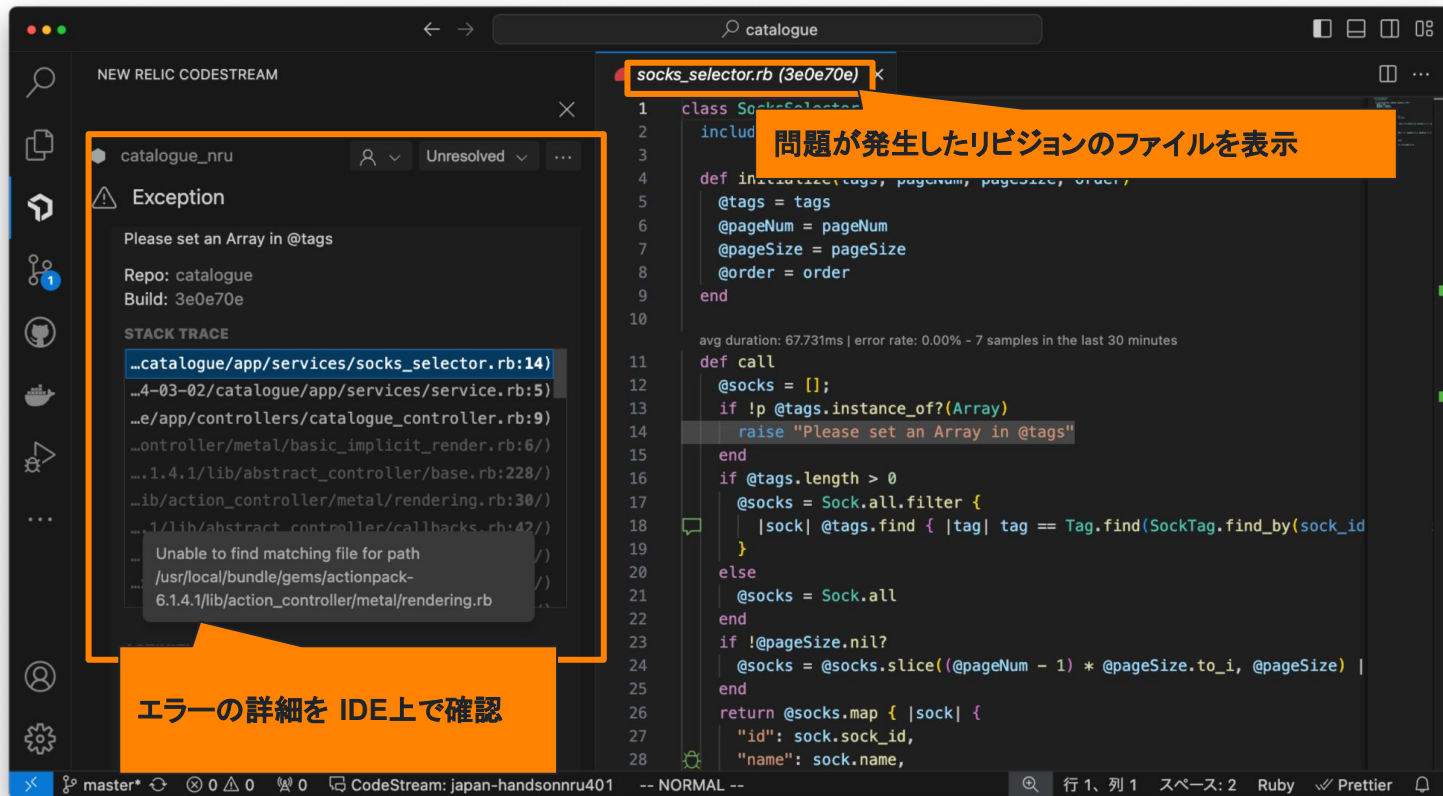


CodeStreamでIDE上でシステムの状態を確認



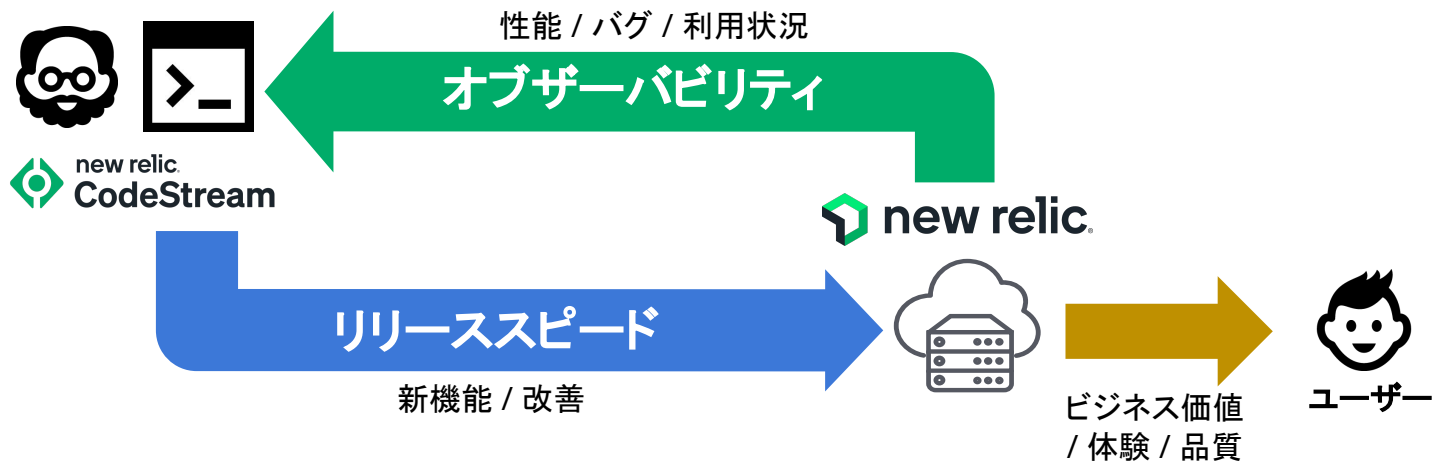
開発チーム

エラーもIDE上で
確認できて嬉しい！



CodeStreamで開発者にオブザーバビリティを提供

スピーディかつ高品質に顧客価値を提供し続けるために、
開発者自身がフィードバックを得て改善できることが重要に



オブザーバビリティと改善の効率性によって次世代の開発者スキルを獲得

“エンジニア” にはこんな世界が待ってる

- 開発チームは運用チームとの**コミュニケーションコストを削減** し、**人依存なく、短時間**で正確なバグ修正
- 開発エンジニアは、**本来の業務であるプロダクトの開発**により多くの時間を使える
- コードレベルの問題解決速度の向上により、運用チームと開発チームの**負担を最小限にしてリリースサイクルを早める**

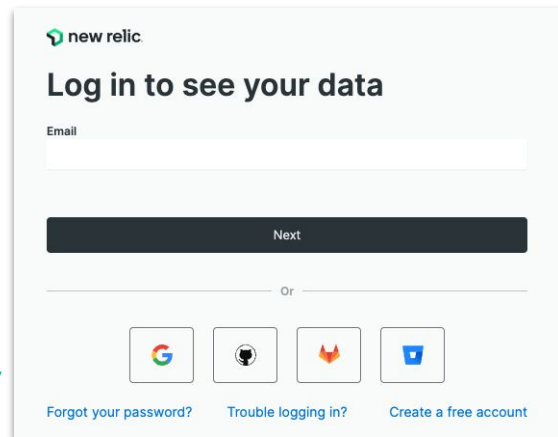
ハンズオン環境について

1. New Relicをブラウザで表示

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

2. New Relicにログイン

- ユーザー: japan-handson+nru002@learn.nrkk.technology
- パスワード: チャットで共有します



下記にご注意ください:

- 普段New Relicをお使いの方はセッションが残っている場合があります
- ブラウザは下記のいずれかのプライベートブラウジングをご利用ください
 - 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
- ユーザー名が “NRU Japan” であることを画面左下で確認ください

 NRU Japan

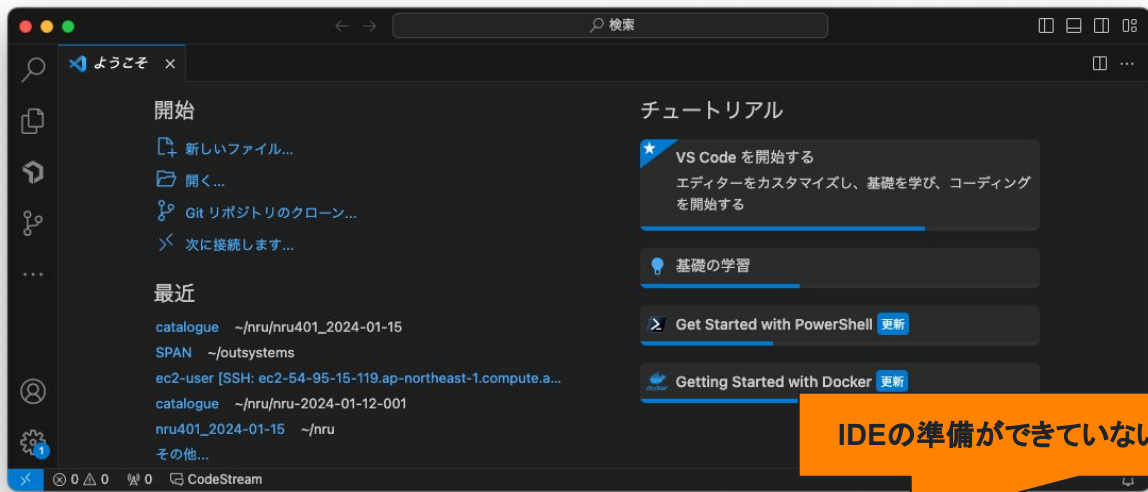


ハンズオン(0): CodeStream環境の準備



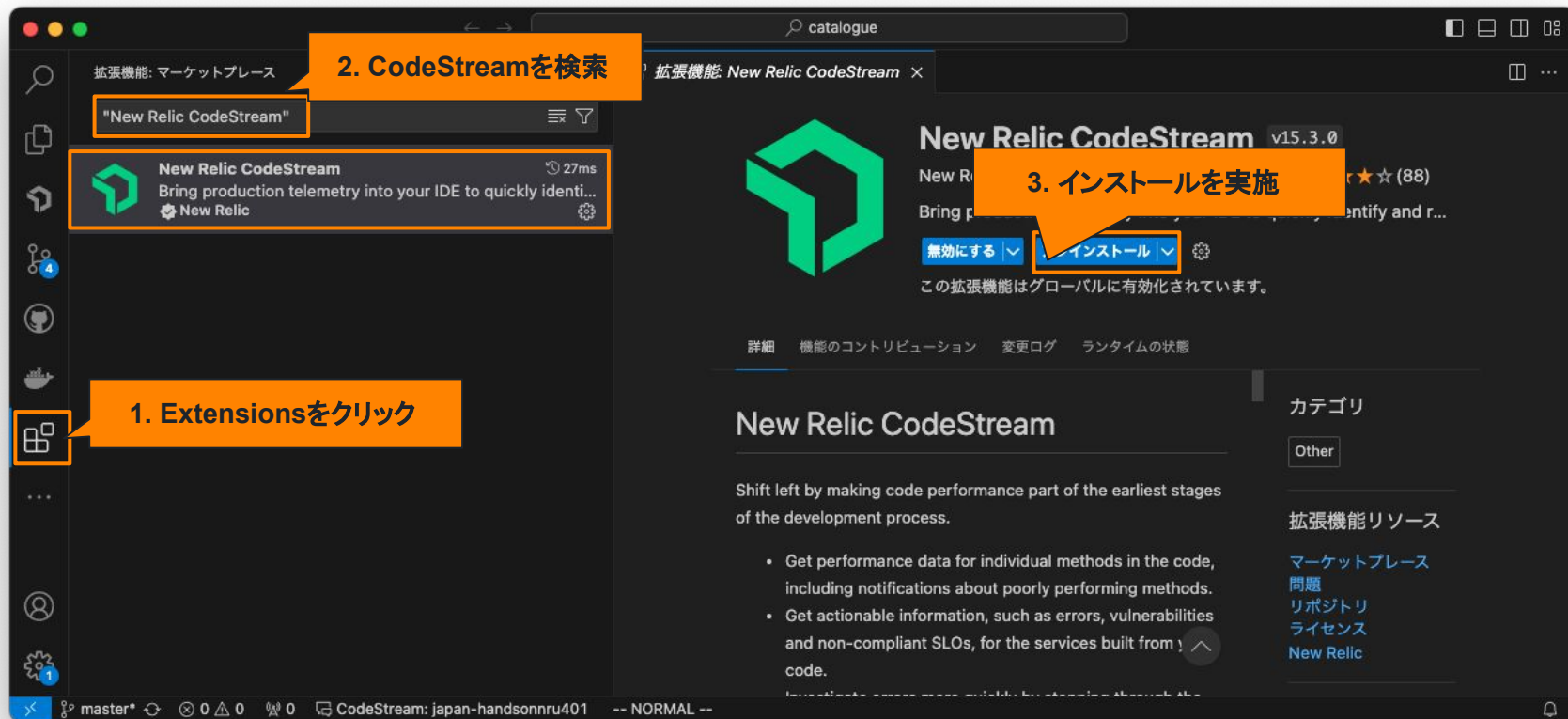
IDEの準備

- 本ハンズオンではIDEとして、**Visual Studio Codeを前提**としています。
※異なるIDEをご利用の方は適宜読み替えてください。
- **CodeStreamの機能と使い方**についてのハンズオンとなります。
※個別のCodeの記法、書き方については扱いません。

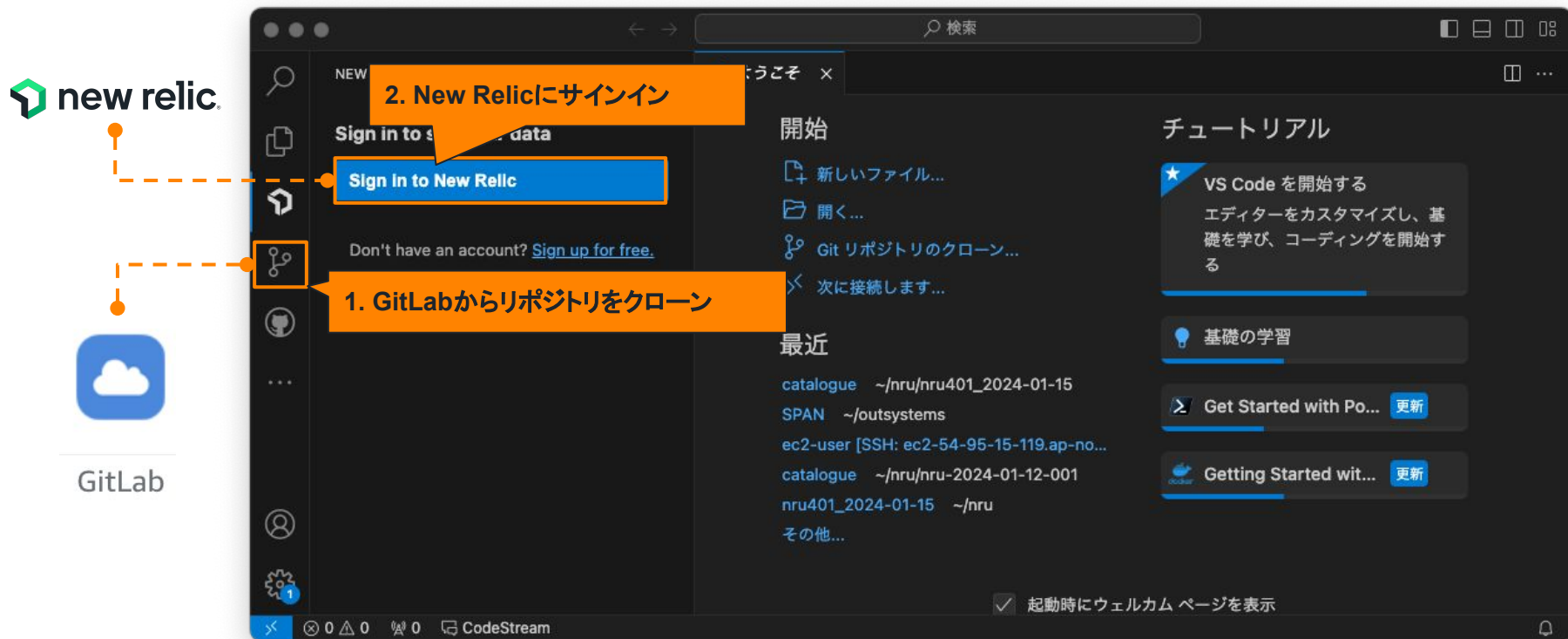


<https://code.visualstudio.com/download>

IDEにCodeStreamをインストール



CodeStreamハンズオン環境

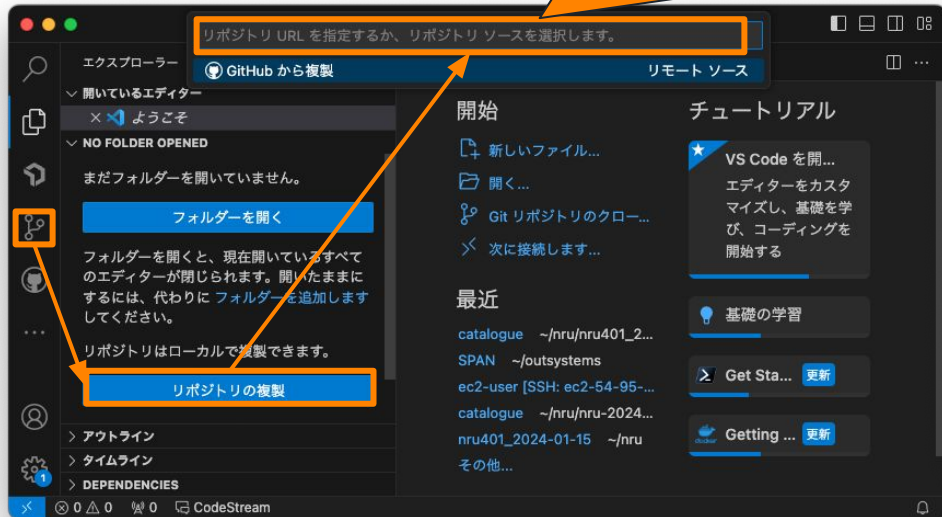


※ハンズオンでは、こちらで用意した環境をご利用ください。

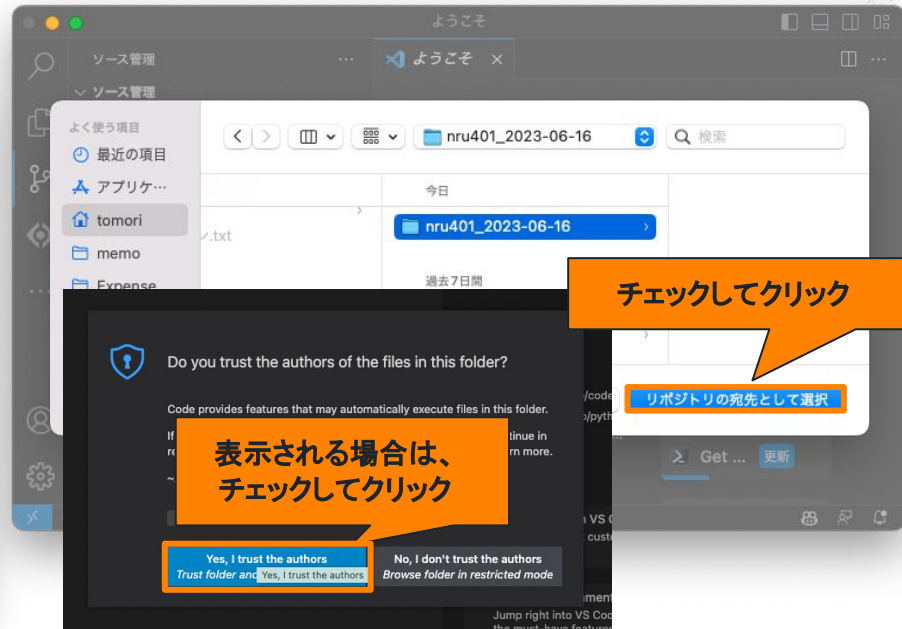
1. リポジトリをローカルにクローン

VSCodeで [リポジトリの複製] を
クリックして URLを入力

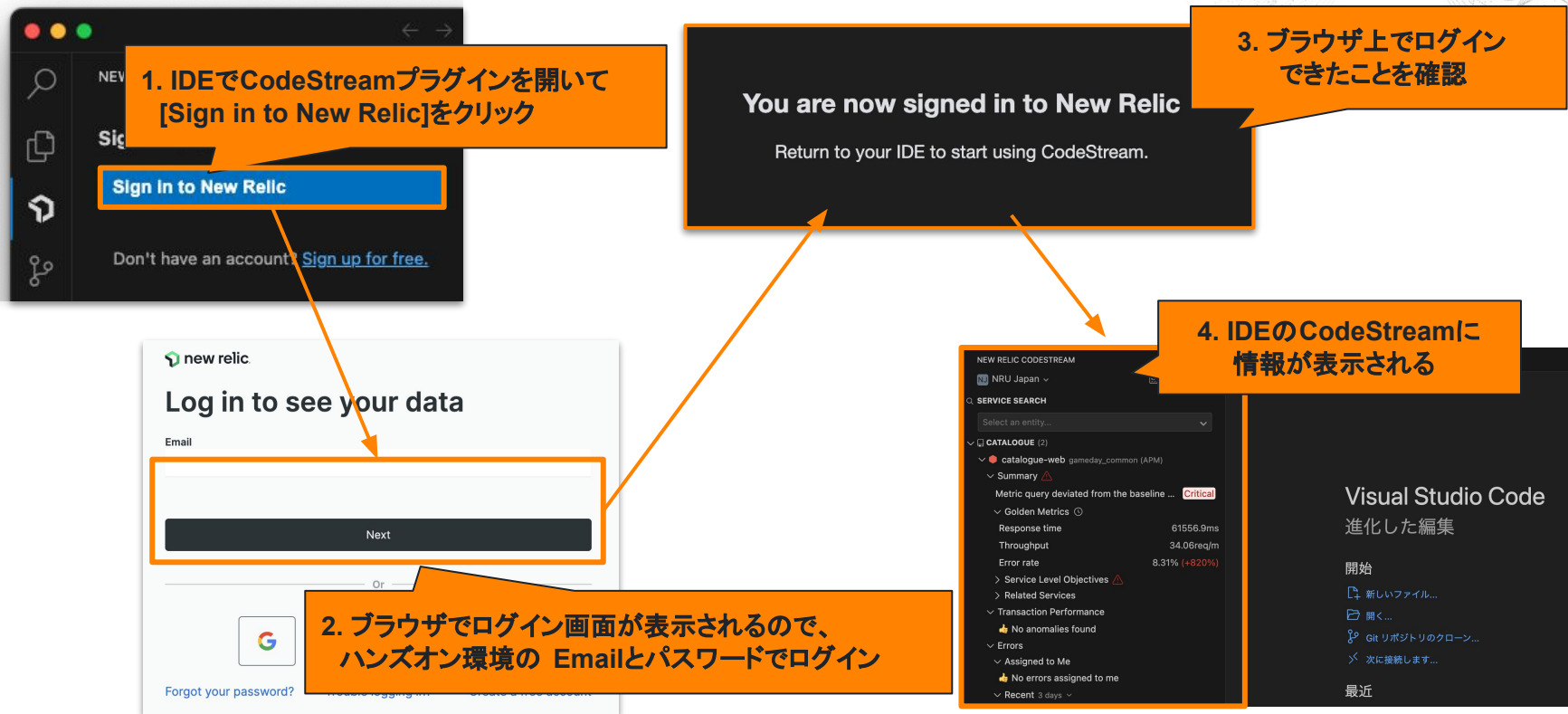
<https://gitlab.gameday.nrkk.technology/gamedaybase/catalogue>



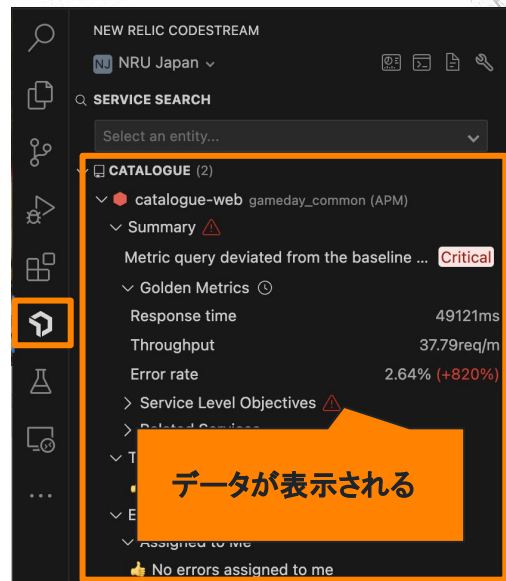
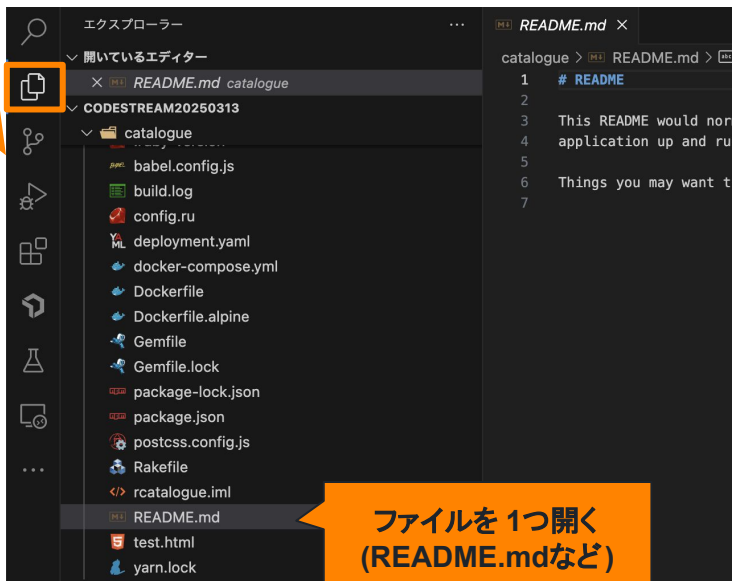
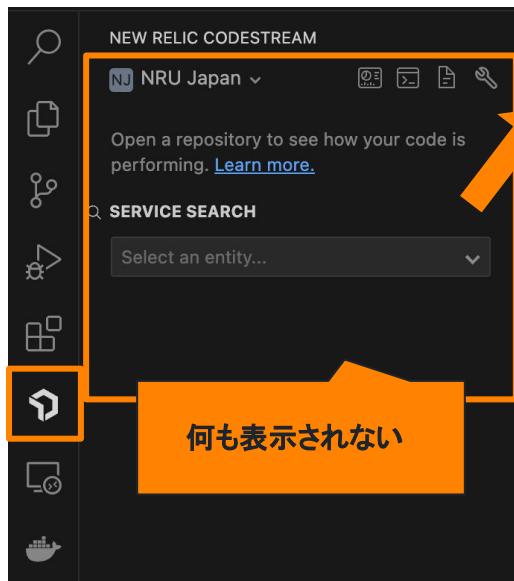
フォルダ選択してソースコードをクローン



2. CodeStream アカウントへのログイン



<参考情報> 情報が表示されない場合は、 クローンしたリポジトリのファイルを開く



ハンズオン(1): 開発者がデプロイされた システムの状態を確認



ハンズオン(1): 開発者がデプロイされたシステムの状態を確認

1

開発者がデプロイされた システムの状態を確認

IDE上でシステムの
パフォーマンスや脆弱性等を
常に確認しながら開発

2

システム状態を IDE上から分析

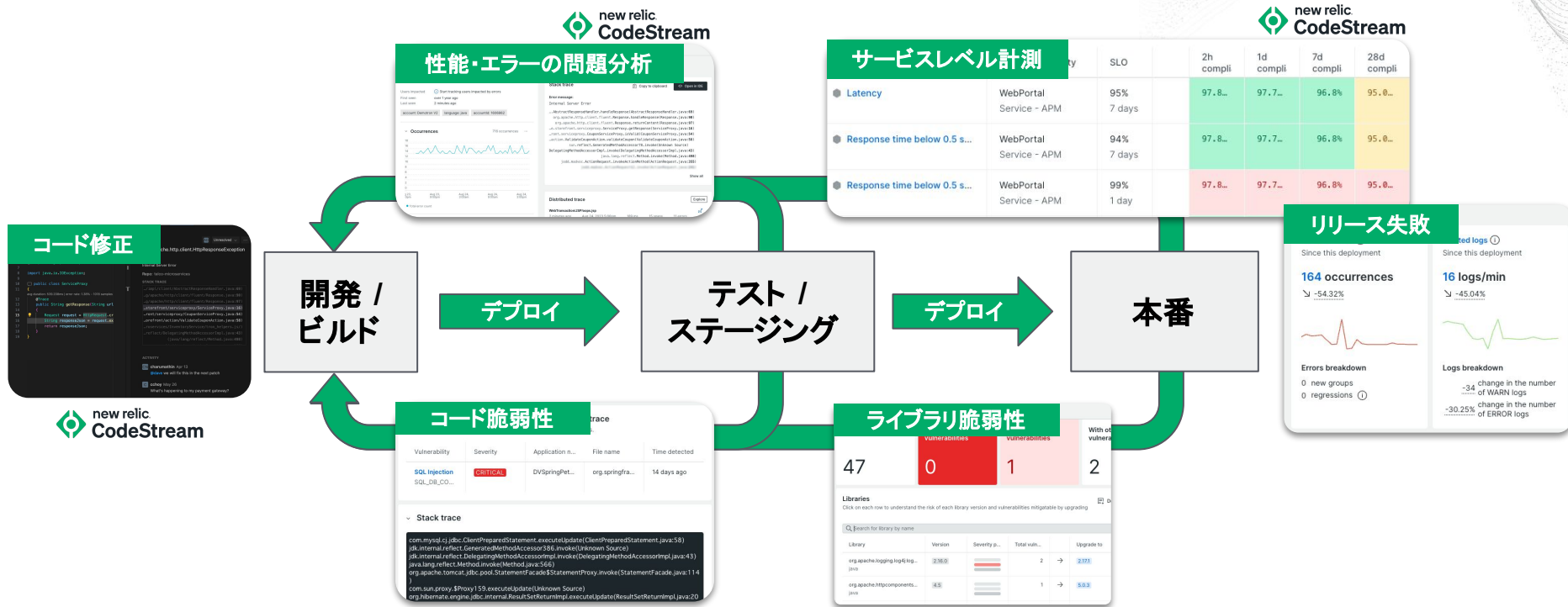
IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の コミュニケーション

デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

リアルタイムにアプリケーションの 状況把握と意思決定を実現



1. システム全体のGolden Metricsを確認

NEW RELIC CODESTREAM

NJ NRU Japan

SERVICE SEARCH

Select an entity...

CATALOGUE (2)

catalogue-web gameday_common (APM)

Summary ⚠

Metric query deviated from the baseline for at least 5 minutes on 'catalog... Critical

Golden Metrics ⓘ

Response time	11721.92ms
Throughput	80.23req/m
Error rate	0.95% (+820%)

> Service Level Objectives ⚠

> Related Services

Transaction Performance

CodeStream
プラグインを確認

IDE上でApplication全体のGolden Metricsを確認

- レスポンス
- スループット
- エラーレート

2. システムに設定されたサービスレベルを確認

The screenshot shows the New Relic IDE interface. On the left is a sidebar with icons for search, documents, links, settings, and a highlighted icon representing service levels. The main panel displays the 'NEW RELIC CODESTREAM' environment for 'NRU Japan'. Under 'SERVICE SEARCH', there is a dropdown menu 'Select an entity...'. Below this, the 'CATALOGUE (2)' section is expanded, showing 'catalogue-web gameday_common (APM)'. Under this, 'Summary' is expanded, showing 'Golden Metrics' and 'Service Level Objectives'. The 'Service Level Objectives' section is highlighted with an orange box and contains the following data:

Service Level Objective	Value
catalogue-web - Latency	94.17% / 7d
catalogue-web - Success	98.8% / 7d
catalogue-web レイテンシ	96.55% / 7d
カタログトップ画面 稼働レベル	100% / 7d

Below the table, there are sections for 'Related Services', 'Transaction Performance' (showing 'No anomalies found'), and 'Errors'.

IDE上でApplicationのサービスレベルを確認

カタログトップ画面 稼働レベル

> Related Services

[View on New Relic](#)

該当の画面を1クリックで開く

3. システムで発生しているエラーを確認

NEW RELIC CODESTREAM

NJ NRU Japan

Q SERVICE SEARCH

Select an entity...

▼ CATALOGUE (2)

▼ catalogue-web gameday_common (APM)

> Summary ⚠

> Transaction Performance

▼ Errors

> Assigned to Me

▼ Recent 3 days

- ⚠ Exception undefined method `map' for blue,magic:String @socks = Sock.joins(sock_tags: :tag).where 1233
- ⚠ Exception undefined method `find' for action:String |sock| @tags.find { |tag| tag == Tag.find(SockTag.i 858
- ⚠ Exception Mysql2::Error: Server shutdown in progress 365
- ⚠ Exception undefined method `map' for action:String @socks = Sock.joins(sock_tags: :tag).where(tag: 265
- ⚠ Exception undefined method `find' for action,black:String |sock| @tags.find { |tag| tag == Tag.find(Soc 207

▼ Vulnerabilities ⚠ Critical, high, medium and low

> rack 2.2.10 -> 2.2.13 (3) high

IDE上でApplicationのエラーを確認

4. 使用しているライブラリの脆弱性を確認

NEW RELIC CODESTREAM

NJ NRU Japan

Q SERVICE SEARCH

Select an entity...

▼ CATALOGUE (2)

- ▼ catalogue-web gameday_common (APM)
 - > Summary
 - > Transaction Performance
 - > Errors
 - ▼ Vulnerabilities critical, high, medium and low
 - > rack 2.2.10 -> 2.2.13 (3) high
 - > actionpack 6.1.7.10 -> 7.0.8.7 (2) low
 - > nokogiri 1.18.2 -> 1.18.3 (2) low
 - > net-imap 0.5.5 -> 0.5.6 (1) medium
 - View Logs
- ▼ catalogue-web gameday_common (Browser)
 - > Add another service

IDE上でApplicationのライブラリと脆弱性を確認

Local File Inclusion in Rack::Static

Severity: HIGH

CVE Id: CVE-2025-27610

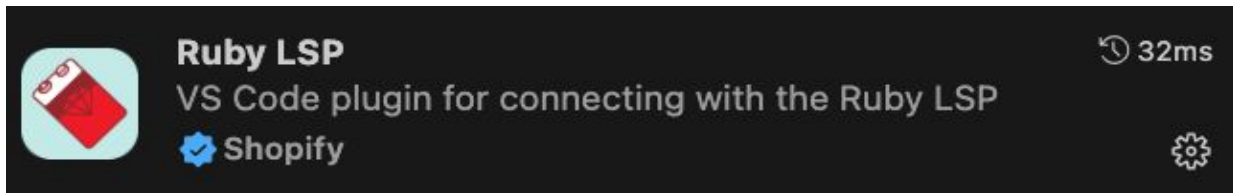
CVSS score: 7.5

CVSS vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

<参考情報> Code Level Metricsの要件

1. 直近30 分間にデータを収集している必要あり
2. 分散トレーシングの有効化が必要
3. 言語ごとに要件あり
 - [こちらを参照](#)
 - 今回のハンズオン環境では、VSCodeで**Ruby LSP**を有効化

時間が余った方は
試してみてください



<https://docs.newrelic.com/jp/docs/codestream/observability/code-level-metrics>

<参考情報> Code Level Metricsを確認

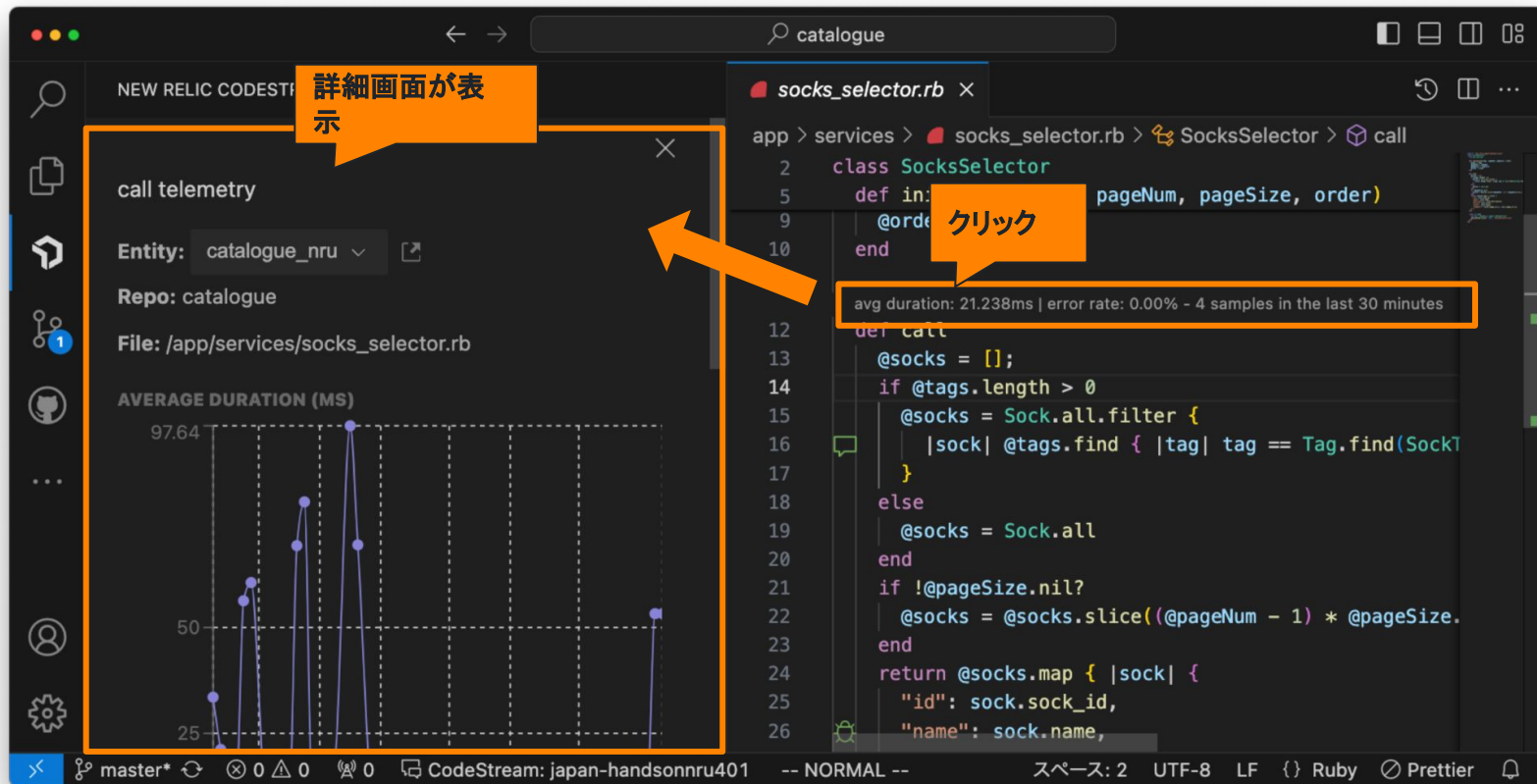
The screenshot shows the New Relic IDE interface. On the left, the file explorer is open, showing the project structure. The file `socks_selector.rb` is selected under the path `app > services`. An orange callout box points to the file explorer icon with the text "エクスプローラーを開く". Another orange callout box points to the file `socks_selector.rb` with the text "CATALOGUE > app > services > socks_selector.rbを開く".

The main editor displays the code for `socks_selector.rb`. An orange callout box points to the code level metrics displayed above the code, which reads: "avg duration: 21.238ms | error rate: 0.00% - 4 samples in the last 30 minutes". Another orange callout box points to the code with the text "メソッドレベルで Golden Metricsを確認" and a list of metrics:

- レスポンス
- スループット
- エラーレート

The bottom status bar shows the current file is `socks_selector.rb`, the Ruby version is `2.7.0`, and the encoding is `UTF-8`.

<参考情報> Code Level Metricsを確認



ハンズオン(2): システム状態をIDE上から分析



ハンズオン(2): システム状態をIDE上から分析

1

開発者がデプロイされた
システムの状態を意識

IDE上でシステムの
パフォーマンス等を
常に確認しながら開発

2

システム状態を
IDE上から分析

IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の
コミュニケーション

デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

1. システムのログを検索

The image shows the New Relic CodeStream interface. On the left, a sidebar contains a search icon, a 'SERVICE SEARCH' section with a dropdown menu, and a 'CATALOGUE (2)' section with various service metrics. An orange box highlights the 'View Logs' button (represented by a document icon) in the top right of the sidebar. An orange arrow points from this button to the 'CodeStream (Logs)' window. This window displays a search interface for logs, including a dropdown for the selected entity ('catalogue-web (APM)'), a time range selector ('30 Minutes Ago'), and a search bar. Below the search bar, a table of log entries is shown.

View Logsをクリック

ログ検索画面が表示

timestamp	message
> 06:29:28 PM May 30	[7f569c73-4553-450d-9ec2-47b305178114] Completed 200 OK in 1ms (Views: 0.2ms ActiveRecord: 0.0ms Allocations: 330)
> 06:29:28 PM May 30	[7f569c73-4553-450d-9ec2-47b305178114] Processing by HealthController#index as */*
> 06:29:28 PM May 30	[289e83c0-3843-49a3-9061-fcf512974b92] Completed 200 OK in 1ms (Views: 0.2ms ActiveRecord: 0.0ms Allocations: 329)

1. システムのログを検索

The screenshot shows the New Relic Logs interface. Annotations in orange boxes highlight the following elements:

- 検索対象のサービスを選択 ※catalogue-web**: Points to the service selection dropdown menu.
- 検索期間を選択 ※1 Day Ago**: Points to the time range selection dropdown menu.
- 検索文字列を入力 ※fatal**: Points to the search input field containing the text "fatal".
- 検索実行**: Points to the "Query Logs" button.

The logs table displays the following data:

timestamp	message
> ■ 2025-03-18 20:12:37.428	[20f71aec-52e8-455c-bef9-0e44d88f63c7] [20f71aec-52e8-455c-bef9-0e44d88f63c7] ActionController::RoutingError (No route matches [GET] "/"): [20f71aec-52e8-455c-bef9-0e44d88f63c7]
> ■ 2025-03-18 20:12:37.428	[20f71aec-52e8-455c-bef9-0e44d88f63c7] [20f71aec-52e8-455c-bef9-0e44d88f63c7] ActionController::RoutingError (No route matches [GET] "/"): [20f71aec-52e8-455c-bef9-0e44d88f63c7]
> ■ 2025-03-18 20:10:26.253	[1da9f022-4b00-41cf-9286-4eab6a9c925f] [1da9f022-4b00-41cf-9286-4eab6a9c925f] ActionController::RoutingError (No route matches [GET] "/login.rsp"): [1da9f022-4b00-41cf-9286-4eab6a9c925f]
> ■ 2025-03-18 20:10:26.253	[1da9f022-4b00-41cf-9286-4eab6a9c925f] [1da9f022-4b00-41cf-9286-4eab6a9c925f] ActionController::RoutingError (No route matches [GET] "/login.rsp"): [1da9f022-4b00-41cf-9286-4eab6a9c925f]
> ■ 2025-03-18 19:47:03.657	[092485e2-4d43-4cee-a44b-a07d693da284] [092485e2-4d43-4cee-a44b-a07d693da284] ActionController::RoutingError (No route matches [GET] "/"): [092485e2-4d43-4cee-a44b-a07d693da284]
> ■ 2025-03-18 19:47:03.657	[092485e2-4d43-4cee-a44b-a07d693da284] [092485e2-4d43-4cee-a44b-a07d693da284] ActionController::RoutingError (No route matches [GET] "/"): [092485e2-4d43-4cee-a44b-a07d693da284]
> ■ 2025-03-18 19:39:00.654	[1b46b647-51f7-47d5-a90b-147a6cea7384] [1b46b647-51f7-47d5-a90b-147a6cea7384] ActionController::RoutingError (No route matches [GET] "/"): [1b46b647-51f7-47d5-a90b-147a6cea7384]

1. システムのログを検索

timestamp	message	
 2025-03-18 20:12:37.428	cluster_name	newrelic
	container_name	catalogue-web
	entity.guid	NDA5NzE4NHxBUE18QVBQTElDQVRJT058MTUxMjg3Njg3Nw
	entity.guids	NDA5NzE4NHxBUE18QVBQTElDQVRJT058MTUxMjg3Njg3Nw
	entity.name	catalogue-web
	entity.type	SERVICE
	hostname	catalogue-web-7d65c97bf6-br9m2
	level	FATAL
	message	[20f71aec-52e8-455c-bef9-0e44d88f63c7] [20f71aec-52e8-455c-bef9-0e44d88f63c7] ActionController::RoutingError (No route matches [GET] "/"): [20f71aec-52e8-455c-bef9-0e44d88f63c7]
	messageId	9c9415d2-fb1b-4a3e-a3ce-1dd7285699e9
	namespace_name	catalogue

クリックしてログの
他の属性値を確認

2. Query Your DataでNRQLを実行

The image shows the New Relic interface with a focus on the 'Query your data' feature. An orange callout box with the text 'Query your data をクリック' (Click 'Query your data') points to a button in the top navigation bar. Another orange callout box with the text 'Query Your Data の画面が表示' (The 'Query Your Data' screen is displayed) points to the 'Query Your Data' panel on the right. The panel shows the account '4097184 - gameday_common' and a text area with the word 'FROM' in purple. The left sidebar shows the 'CATALOGUE' section with 'catalogue-web' selected.

NEW RELIC CODESTREAM

NJ NRU Japan ▾

Query your data

Query your data

Select an entity...

▼ CATALOGUE (2)

▼ catalogue-web gameday_cor

▼ Summary ⚠

▼ Golden Metrics ⌚

Response time

Throughput

Error rate

> Service Level Objectives ⚠

> Related Services

CodeStream (NRQL) ×

Query Your Data

💡 Save and share queries with .nrql files

Account: 4097184 - gameday_common ▾ Recent queries ▾

FROM

Clear Run

2. Query Your DataでNRQLを実行

CodeStream (NRQL) ×

Query Your Data

Account: 4097184 - gameday_common

FROM Transaction SELECT * SINCE 1 DAY AGO

NRQLを入力
例1) FROM Transaction SELECT * SINCE 1 DAY AGO
例2) FROM Transaction SELECT count(*) SINCE 1 DAY AGO TIMESERIES

Clear Run

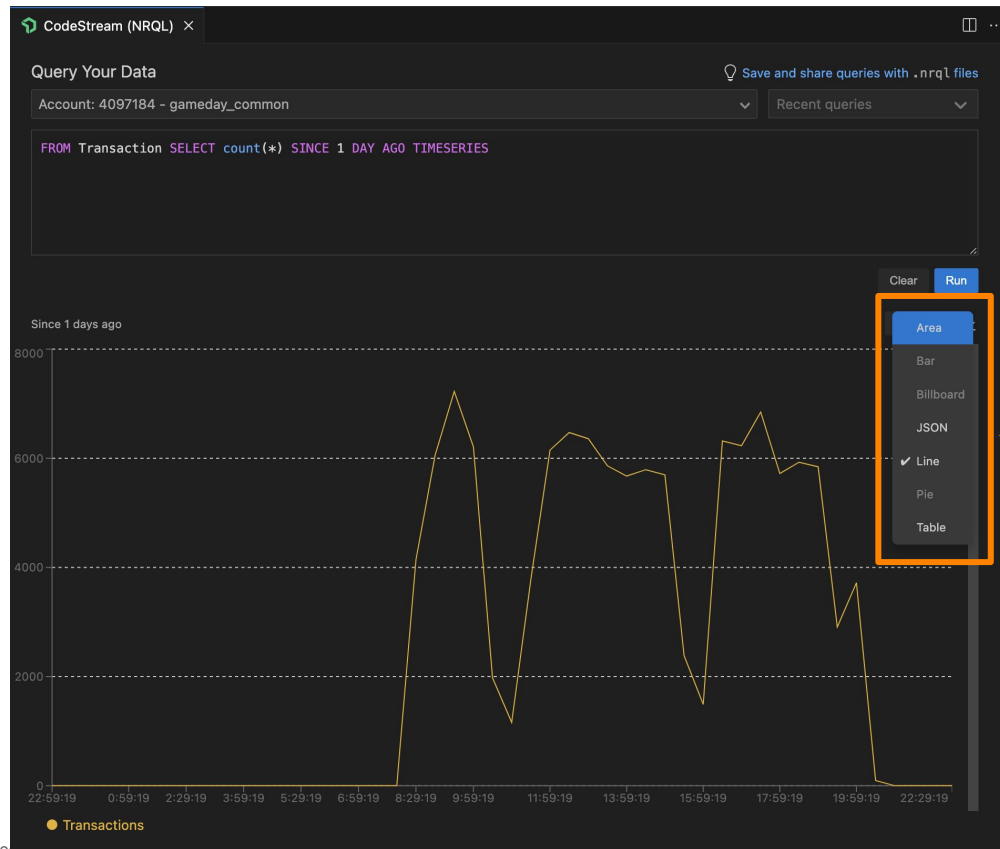
Runを実行

Since 1 days ago Table ↓

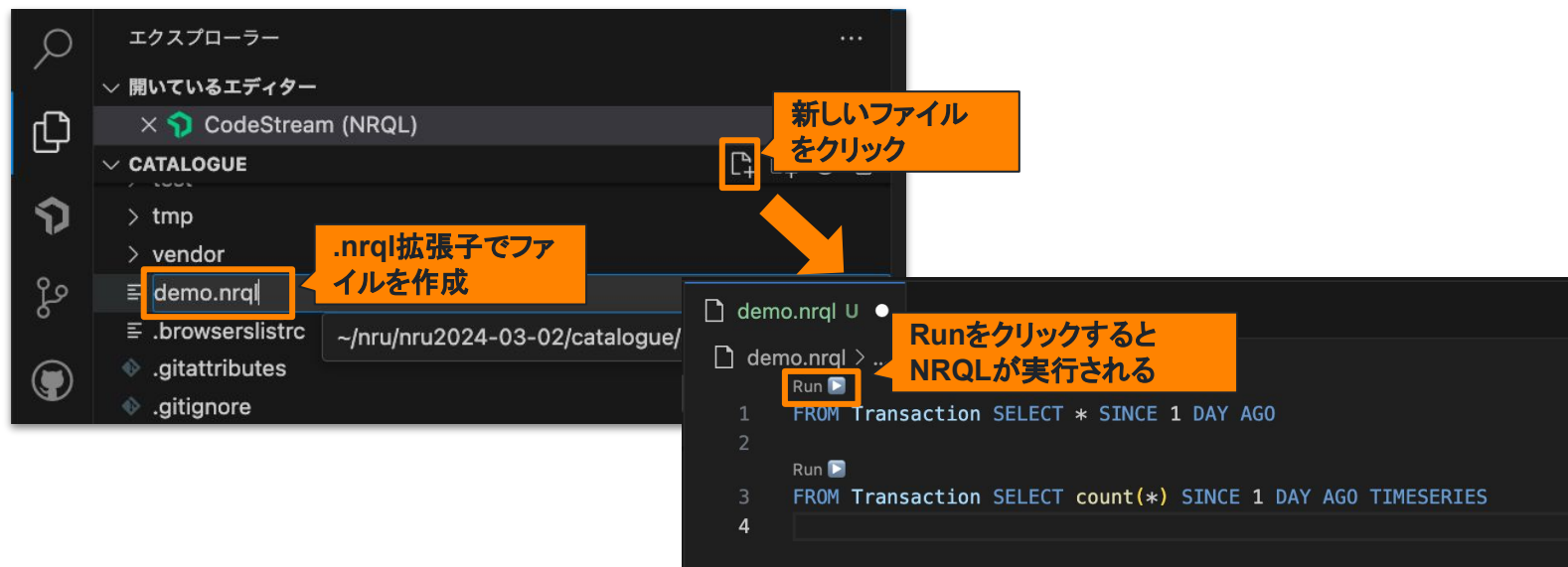
結果を表示

timestamp	request.headers.host	guid	cluster
1717066805496	catalogue.demo.learn.nrkk.tech...	430d2818fe821559	newreli
1717066803048	front.demo.learn.nrkk.technolo...	7cd00c038a0019b2	newreli
1717066803029	catalogue.demo.learn.nrkk.tech...	f6513cb7cd13e630	newreli

2. Query Your DataでNRQLを実行



3. NRQLファイルからNRQLを実行



NRQLを入力

例)

```
FROM Transaction SELECT * SINCE 1 DAY AGO
```

```
FROM Transaction SELECT count(*) SINCE 1 DAY AGO TIMESERIES
```

4. Dashboard の参照

NEW RELIC CODESTREAM

NJ NRU Japan ▾

SERVICE SEARCH

Select an entity...

CATALOGUE (2)

- catalogue-web gameday
- Summary ⚠
- Golden Metrics 🕒
 - Response time
 - Throughput

Dashboards

CodeStream (Dashboards) ✕

Dashboards

Search dashboards by name Search

	Name	Account	Created By	Last Edited	Created On
★	SaaS Engineering Dashboard	gameday_com...	tazuma+nrpro...	Jan 16, 2025	Oct 23, 2024
☆	2025-01-16 SaaS Engineering Dashboard	gameday_com...	yokawa+gamed...	Feb 13, 2025	Feb 13, 2025
☆	2025-01-16 SaaS Engineering Dashboard	gameday_com...	japan-handson...	Oct 10, 2024	Oct 10, 2024
☆	2025-01-16 SaaS Engineering Dashboard	gameday_com...	japan-handson...	Jun 14, 2024	Mar 22, 2024
☆	DevOps 4 keys	gameday_com...	tazuma+nrpro...	Feb 13, 2025	Jan 22, 2025
☆	Kubernetes Dashboard	gameday_com...	tazuma+nrpro...	Feb 13, 2025	Jun 20, 2024

4. Dashboard の参照



ハンズオン(3): 開発者と運用者のコミュニケーション



ハンズオン(3): 開発者と運用者のコミュニケーション

1

開発者がデプロイされた
システムの状態を確認

IDE上でシステムの
パフォーマンスや脆弱性等を
常に確認しながら開発

2

システム状態を
IDE上から分析

IDE上でシステムの稼働データを
クエリを使って分析したり、
ログを調査したりすることが可能

3

開発者と運用者の
コミュニケーション

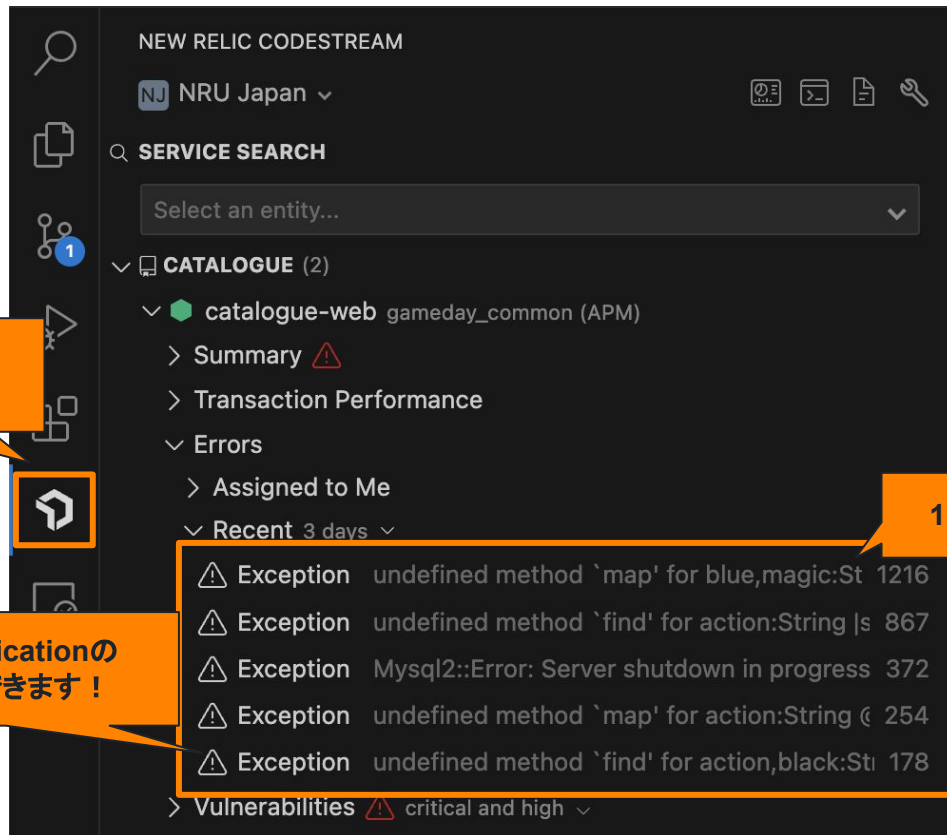
デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

1. CodeStreamでErrorsを確認

CodeStream
プラグインを確認

IDE上でApplicationの
エラーを確認できます！

1つ選んでクリック



1. CodeStreamでErrorsを確認

エラー詳細が表示される

エラーが発生した環境のファイルがリビジョン指定で表示される

スタックトレースを選択する

関連する Code 行が自動的にフォーカス

```
Exception
undefined method `map' for blue,magic:String @socks = Sock.joins(sock_tags: :tag).where(tag: { name: @tags.map { |tag| tag.strip! ? tag.strip! : tag }}).distinct ^^^^^
Did you mean? tap

Re
Bu

STACK TRACE
NRU/codestream20250313/catalogue/app/services/socks_selector.rb:15)
~/msaito/NRU/codestream20250313/catalogue/app/services/service.rb:5)
~/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:339/)
~/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:74/)
~/wrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37/)
~/bundle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/tracer.rb:357/)
~/wrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37/)
~/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:72/)
~/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:334/)

View related logs

DISCUSSION
AI
The stack trace indicates an error where the map method is being called on a String object, which is not valid. The error occurs in the context of processing tags, suggesting that the @tags variable might not be an array as expected. The code provided needs to be analyzed to ensure that @tags is indeed an array before calling map on it.

def call
  @socks = []
  if @tags.length > 0
    @socks = Sock.all.filter {
      |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_by(so
    }
  }
  @socks = []
  if @tags.is_a?(Array) && @tags.length > 0
    @socks = Sock.all.filter do |sock|
      @tags.find { |tag| tag == Tag.find(SockTag.find_by(so
    end
  end
```

```
socks_selector.rb (a0cb5df)
require 'new_relic/agent'
class SocksSelector
  include Service

  def initialize(tags, pageNum, pageSize, order)
    @tags = tags
    @pageNum = pageNum
    @pageSize = pageSize
    @order = order
  end

  def call
    @socks = []
    if @tags.length > 0
      @socks = Sock.joins(sock_tags: :tag,
        strip! : tag })).distinct
    else
      @socks = Sock.all
    end
    if !@pageSize.nil?
      @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @pageSize) || []
    end
    return @socks.map { |sock| {
      "id": sock.sock_id,
      "name": sock.name,
      "description": sock.description,
      "price": sock.price,
      "count": sock.count,
      "imageUrl": [sock.image_url_1, sock.image_url_2]
    } }
  end

  class << self
    include ::NewRelic::Agent::MethodTracer
    add_method_tracer :call, 'SocksSelector/call'
  end
end
```

<参考情報>エラー発生時のリビジョンのファイル

NEW RELIC CODESTREAM

catalogue-web

Exception

undefined method `length` for nil:NilClass

Repo: catalogue
Build: 3a20ab0

STACK TRACE

../CodeStream/gameday-NRU/catalogue/app/services/socks_selector.rb:14
...ito/work/CodeStream/gameday-NRU/catalogue/app/services/service.rb:5
...e/gems/newrelic_rpm-8.8.0/lib/new_relic/agent/method_tracer.rb:335/
...le/gems/newrelic_rpm-8.8.0/lib/new_relic/agent/method_tracer.rb:74/
...newrelic_rpm-8.8.0/lib/new_relic/agent/method_tracer_helpers.rb:41/
...l/bundle/gems/newrelic_rpm-8.8.0/lib/new_relic/agent/tracer.rb:351/
...newrelic_rpm-8.8.0/lib/new_relic/agent/method_tracer_helpers.rb:41/

socks_selector.rb (3a20ab0)

```
1 require 'new_relic/agent/method_tracer'
2 class SocksSelector
3   include Service
4
5   def initialize(tags, pageNum, pageSize, order)
6     @tags = tags
7     @pageNum = pageNum
8     @pageSize = pageSize
9     @order = order
10  end
11
12  def call
13    @socks = []
14    if @tags.length > 0
15      @socks = Sock.all.filter {
16        |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_by(sock_id: sock.sock_id)
17      }
18    else
19      @socks = Sock.all
```

同一ファイルの異なるリビジョン

NEW RELIC CODESTREAM

catalogue-web

Exception

undefined method `strip!' for [blue, magic]:Array

Repo: catalogue
Build: 3a2bed3

STACK TRACE

../CodeStream/gameday-NRU/catalogue/app/services/socks_selector.rb:15
../CodeStream/gameday-NRU/catalogue/app/services/socks_selector.rb:15
../CodeStream/gameday-NRU/catalogue/app/services/socks_selector.rb:15
...ito/work/CodeStream/gameday-NRU/catalogue/app/services/service.rb:5
...am/gameday-NRU/catalogue/app/controllers/catalogue_controller.rb:15
...ack-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6/

socks_selector.rb (3a2bed3)

```
1 require 'new_relic/agent/method_tracer'
2 class SocksSelector
3   include Service
4
5   def initialize(tags, pageNum, pageSize, order)
6     @tags = tags
7     @pageNum = pageNum
8     @pageSize = pageSize
9     @order = order
10  end
11
12  def call
13    @socks = [];
14    if @tags.length > 0
15      @socks = Sock.joins(sock_tags: :tag).where(tag: { name: @tags.map { |tag| tag.str
16    else
17      @socks = Sock.all
18  end
```

<参考情報>エラー発生時のリビジョンのファイル

リビジョン情報を表示するためには、commit shaおよび/またはreleaseタグの設定が必要

- <https://docs.newrelic.com/jp/docs/codestream/how-use-codestream/performance-monitoring/#buildsha>
- <https://newrelic.com/jp/blog/how-to-relic/troubleshooting-settings#toc-3-apm->

設定例)

Dockerfile

```
34 ARG COMMIT_SHA="sha"
35 ARG RELEASE_TAG="dev"
36 ENV NEW_RELIC_METADATA_COMMIT=$COMMIT_SHA
37 ENV NEW_RELIC_METADATA_HOGE=fuga
38 ENV NEW_RELIC_METADATA_RELEASE_TAG=$RELEASE_TAG
```

.gitlab-ci.yml

```
30     echo "Running on branch '$CI_COMMIT_BRANCH': tag = $tag"
31   fi
32   - docker build . --tag public.ecr.aws/g7c8o4c7/catalogue:${tag} --build-arg COMMIT_SHA=${CI_COMMIT_SHA}
33   - docker push public.ecr.aws/g7c8o4c7/catalogue:${tag}
34   # Run this job in a branch where a Dockerfile exists
```

2. New RelicでErrorsを確認

new relic

APM & Services [Save view](#)

Account: 4097184 - gar

Quick Find

Integrations & Agents

All Capabilities

All Entities

Query Your Data

APM & Services

Logs

Dashboards

Traces

Search by entity name

Entity Type = Service - APM, Service - OpenTelemetry

Team = All

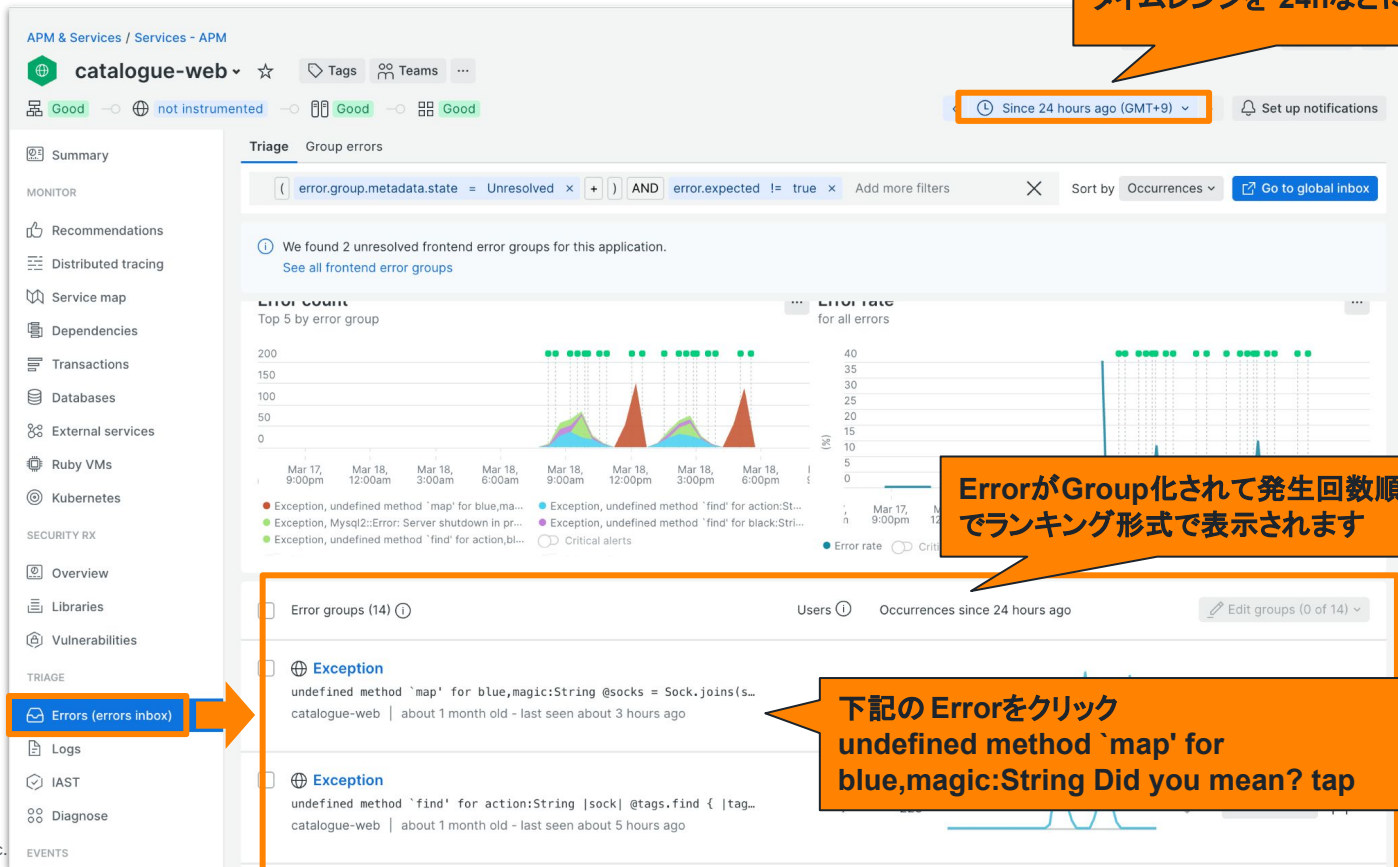
Services - APM [View all \(2\)](#)

	Name ↑	Team	Account	Response time
☆	front-service	--	gameday_common	2834 ms
	catalogue-web	--	gameday_common	9 s

ログインしたら APMから catalogue-webをクリック

2. New RelicでErrorsを確認

エラーが表示されない場合は
タイムレンジを 24hなどに変更



ErrorがGroup化されて発生回数順
でランキング形式で表示されます

下記の Errorをクリック
undefined method `map` for
blue,magic:String Did you mean? tap

2. New RelicでErrorsを確認

Exception Unresolved  First < 1 of 156 > Last

catalogue-web

Error message: undefined method `map' for blue,magic:String

Sock.join? tag.strip! ? tag.strip! : tag

4-qdwt8 technology

Show all Show more attributes

View discussions (1)

Error group metrics

Users impacted 111 users impacted since 24 hours ago ⓘ
— users impacted since 1 month ago ⓘ

First seen about 1 month ago in v0.3.13.b

Last seen about 3 hours ago in v0.3.13.b

account:gameday_common accountId:4097184 language:ruby

Occurrences 156 occurrences ⓘ ...

50
40
30
20
10
0

Mar 17, 9:00pm Mar 18, 12:00am Mar 18, 3:00am Mar 18, 6:00am Mar 18, 9:00am Mar 18, 12:00pm Mar 18, 3:00pm Mar 18, 6:00pm

Total error count Users impacted

Stack trace

Explain this error ⓘ Copy to clipboard <> Open in IDE

Error message:
undefined method `map' for blue,magic:String

@socks = Sock.joins(sock_tags: :tag).where(tag: { name: @tags.map { |tag| tag.strip! ? tag.strip! : tag }}).distinct
^^^^
Did you mean? tap

```

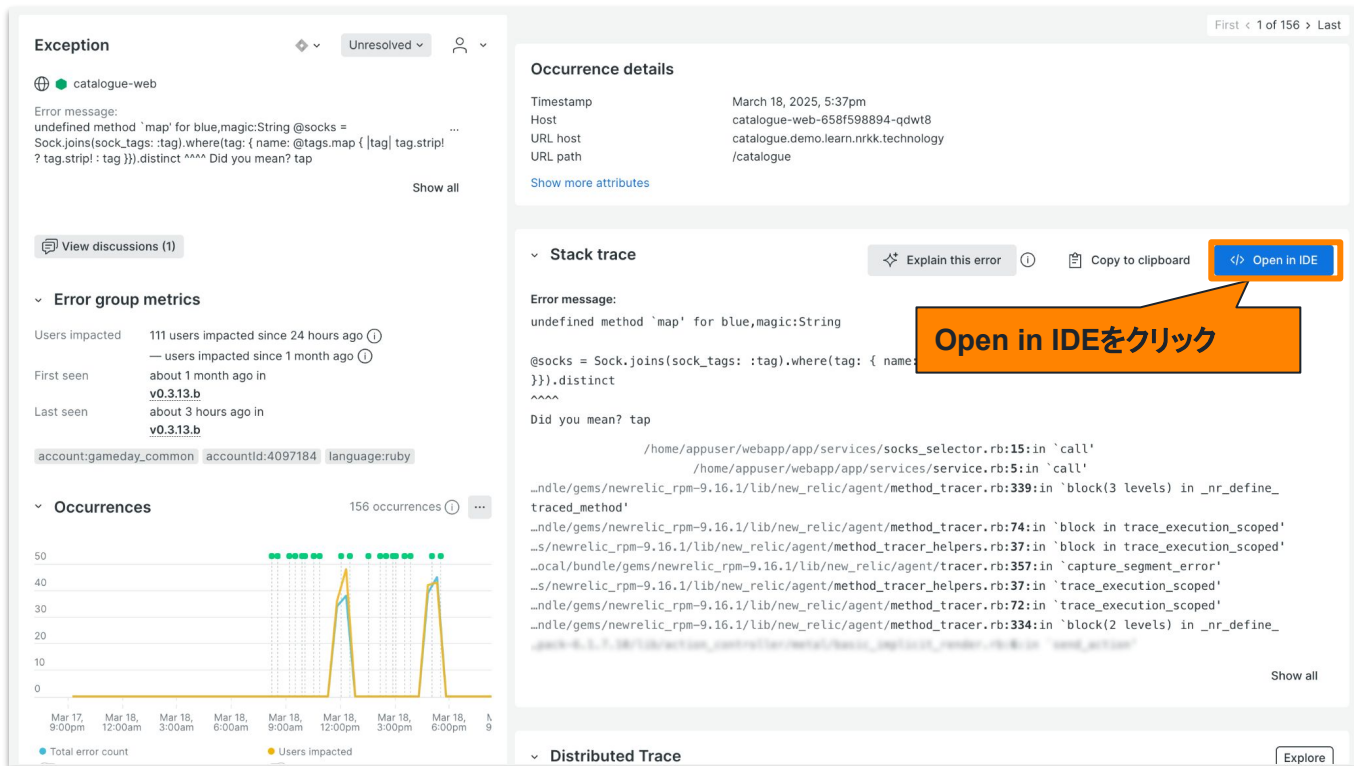
/home/appuser/webapp/app/services/socks_selector.rb:15:in `call'
/home/appuser/webapp/app/services/service.rb:5:in `call'
...ndle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:339:in `block(3 levels) in _nr_define_traced_method'
...ndle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:74:in `block in trace_execution_scoped'
...s/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37:in `block in trace_execution_scoped'
...ocal/bundle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/tracer.rb:357:in `capture_segment_error'
...s/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37:in `trace_execution_scoped'
...ndle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:72:in `trace_execution_scoped'
...ndle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:334:in `block(2 levels) in _nr_define_traced_method'

```

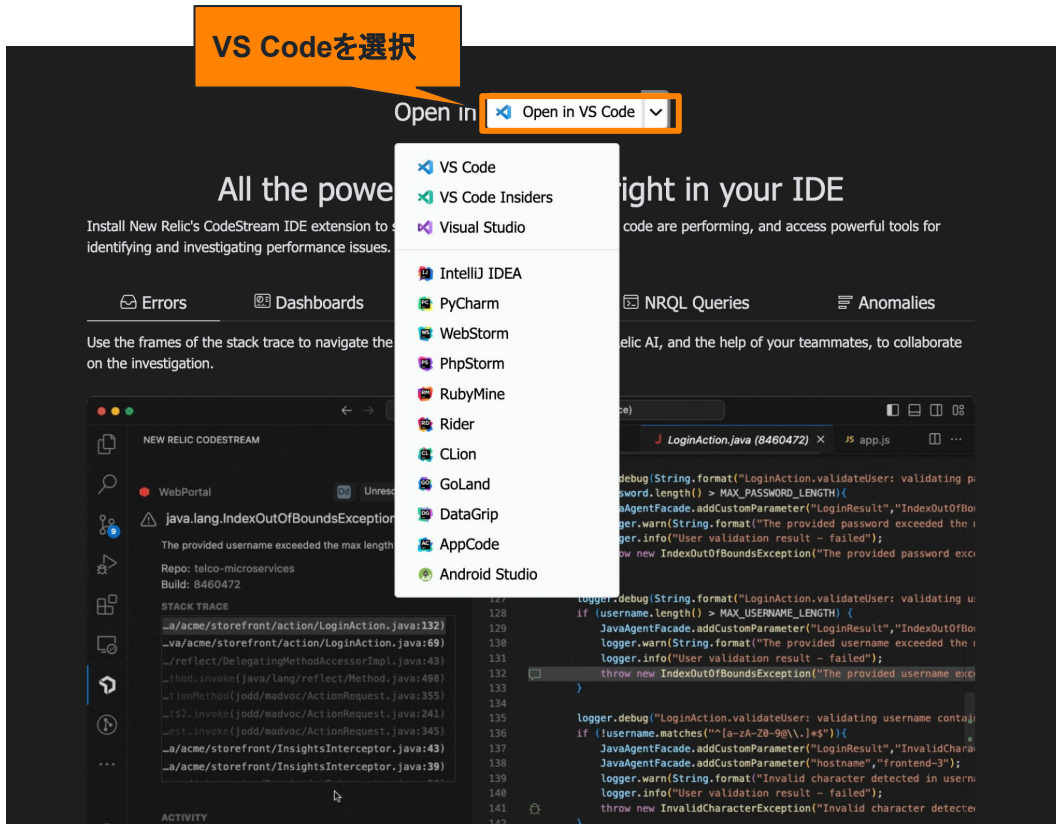
Show all

Distributed Trace Explore

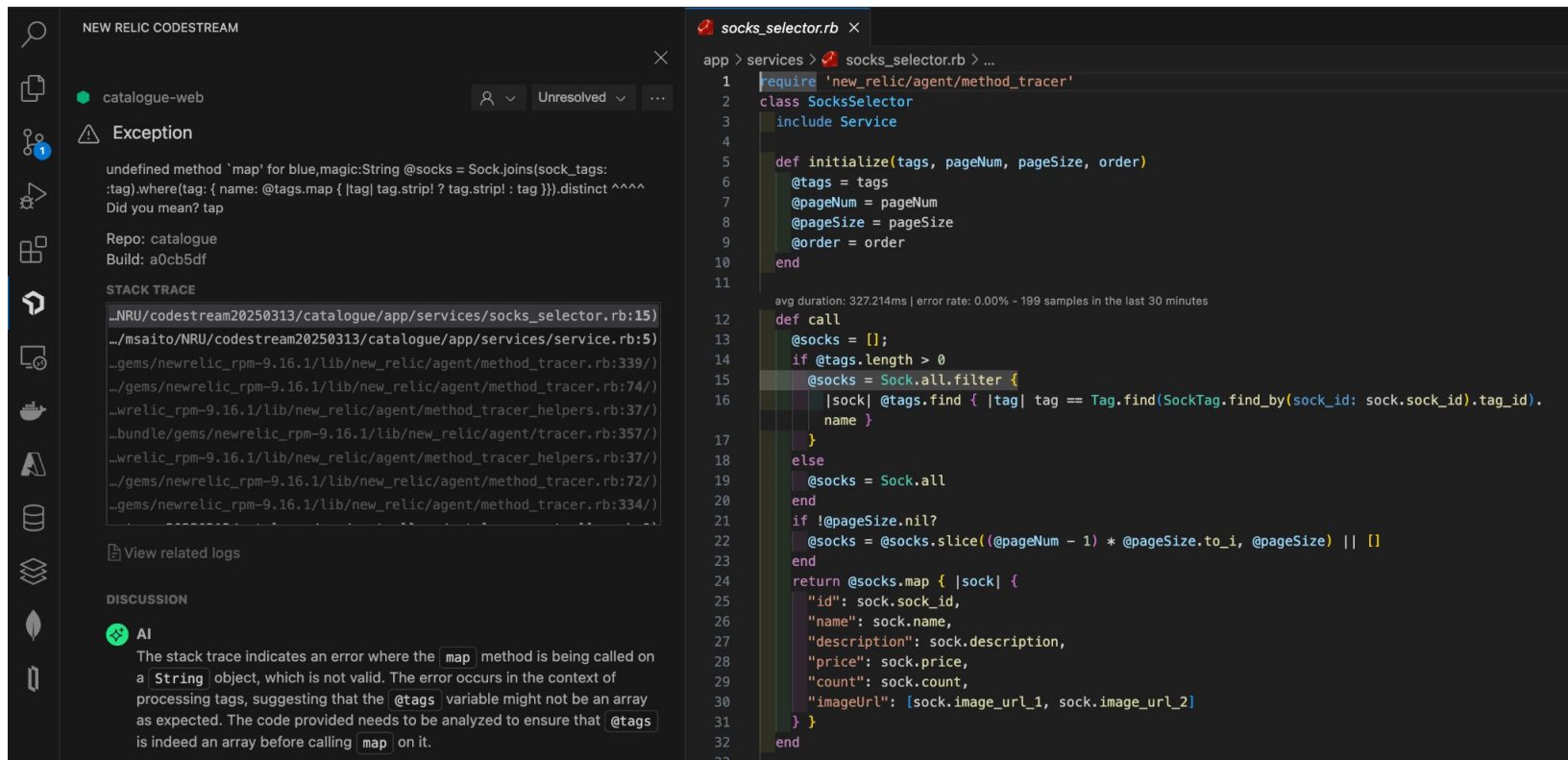
2. New RelicでErrorsを確認



2. New RelicでErrorsを確認



2. New RelicでErrorsを確認



The screenshot displays the New Relic interface with an exception and its stack trace. The left sidebar shows the 'Exception' tab, and the main area displays the error details and the corresponding code in `socks_selector.rb`.

Exception Details:

- Exception:** undefined method `map' for blue,magic:String @socks = Sock.joins(sock_tags: :tag).where(tag: { name: @tags.map { |tag| tag.strip! ? tag.strip! : tag }}).distinct ^^^^ Did you mean? tap
- Repo:** catalogue-web
- Build:** a0cb5df
- STACK TRACE:**
_NRU/codestream20250313/catalogue/app/services/socks_selector.rb:15)
../msaito/NRU/codestream20250313/catalogue/app/services/service.rb:5)
_gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:339(/)
_gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:74(/)
_wrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37(/)
_bundle/gems/newrelic_rpm-9.16.1/lib/new_relic/agent/tracer.rb:357(/)
_wrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer_helpers.rb:37(/)
_gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:72(/)
_gems/newrelic_rpm-9.16.1/lib/new_relic/agent/method_tracer.rb:334(/)

DISCUSSION:

- AI:** The stack trace indicates an error where the `map` method is being called on a `String` object, which is not valid. The error occurs in the context of processing tags, suggesting that the `@tags` variable might not be an array as expected. The code provided needs to be analyzed to ensure that `@tags` is indeed an array before calling `map` on it.

Code Snippet (socks_selector.rb):

```
1 require 'new_relic/agent/method_tracer'
2 class SocksSelector
3   include Service
4
5   def initialize(tags, pageNum, pageSize, order)
6     @tags = tags
7     @pageNum = pageNum
8     @pageSize = pageSize
9     @order = order
10  end
11
12  avg duration: 327.214ms | error rate: 0.00% - 199 samples in the last 30 minutes
13  def call
14    @socks = []
15    if @tags.length > 0
16      @socks = Sock.all.filter {
17        |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_by(sock_id: sock.sock_id).tag_id).
18          name }
19      }
20    else
21      @socks = Sock.all
22    end
23    if !@pageSize.nil?
24      @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @pageSize) || []
25    end
26    return @socks.map { |sock| {
27      "id": sock.sock_id,
28      "name": sock.name,
29      "description": sock.description,
30      "price": sock.price,
31      "count": sock.count,
32      "imageUrl": [sock.image_url_1, sock.image_url_2]
33    } }
34  end
35 end
```

<参考情報>New Relic AIにエラー内容確認

Exception

undefined method 'length' for nil:NilClass

Repo: catalogue
Build: 1f93390

STACK TRACE

```
...ru/nru2024-03-02/catalogue/ap  
...tomori/nru/nru2024-03-02/cata  
...24-03-02/catalogue/app/contro  
...4.1/lib/action_controller/me  
...gems/actionpack-6.1.4.1/lib/a  
...ionpack-6.1.4.1/lib/action_cc  
.../actionpack-6.1.4.1/lib/abstr  
...s/activesupport-6.1.4.1/lib/a  
...dle/gems/actiontext-6.1.4.1/l
```

ACTIVITY

AI 2 hours ago

The error message `undefined method 'length' for nil:NilClass` indicates that the `length` method is being called on a `nil` object. In the provided code, the `length` method is called on `@tags`. This suggests that `@tags` is `nil` at the time of the method call.

```
def call
  @socks = [];
  if @tags.length > 0
    if @tags && @tags.length > 0
      @socks = Sock.all.filter {
        |sock| @tags.find { |tag| tag == Tag.find(SockTag.f
      }
    else
      @socks = Sock.all
    end
  if !@pageSize.nil?
```

Apply Fix

The fix involves adding a check to ensure `@tags` is not `nil` before calling the `length` method on it. This is done by adding `@tags &&` before `@tags.length > 0`. This way, if `@tags` is `nil`, the condition will be `false` and the `length` method will not be called, thus avoiding the

New Relic AIがエラー内容と対応方法を提示

@AIで質問できます！

japan-handsonnru-
@AI 日本語でお願い。

AI 2 hours ago

説明

エラーメッセージ `undefined method 'length' for nil:NilClass` は、`nil` オブジェクトに対して `length` メソッドが呼び出されていることを示しています。提供されたコードでは、`length` メソッドは `@tags` に対して呼び出されています。これは、メソッド呼び出し時に `@tags` が `nil` であることを示しています。

修正は、`length` メソッドを呼び出す前に `@tags` が `nil` でないことを確認することで行います。これは、`@tags.length > 0` の前に `@tags &&` を追加することで行います。このようにすると、`@tags` が `nil` の場合、条件は `false` となり、`length` メソッドは呼び出されず、エラーを回避することができます。

まとめ

CodeStreamを活用して“**One Team**” を実現し、
顧客満足度の高いサービス開発 する方法を理解

- 開発において稼働中の**システムの状態** を常に意識
- 開発に必要なデータをIDEに集約し**データ分析を効率化**
- 不具合対応における**SRE**と開発者の円滑なコラボレーション

NRU GUIDE

NRU401

New Relic University

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

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

2025年4-6月期開催のハンズオンセミナー

2025年4月9日(水)15:00-17:00

NRU300 - アプリケーションとインフラ性能観測の基本

2025年4月23日(水)15:00-17:00

NRU300 - Webサイト上のユーザー体験観測の基本



2025年5月14日(水)15:00-17:00

NRU300 - ダッシュボード開発とNRQLの基本

2025年5月28日(水)15:00-17:00

NRU300 - アラート設計の基本と活用

2025年6月4日(水)15:00-17:00

NRU300 - SLI/SLO 設計の基本

2025年6月11日(水)15:00-17:00

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

2025年6月18日(水)15:00-17:00

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

【開催形式】

全てZoomによるウェビナー
(参加費無料)

こちらからも一覧でご覧いただけます！

new relic. University	new relic. University	new relic. University
ワークショップ NRU301 - アプリケーションとインフラ性能観測の基本	ワークショップ NRU302 - ダッシュボード開発とNRQLの基本	ワークショップ NRU303 - SLI/SLO 設計の基本
📍 Webinar 🕒 15:00 - 17:00 📅 2024年10月9日 (水)	📍 Webinar 🕒 15:00 - 17:00 📅 2024年10月23日 (水)	📍 Webinar 🕒 15:00 - 17:00 📅 2024年11月6日 (水)
登録	登録	登録

NRU300の新ウェビナーのご紹介

NRU300 - Webサイト上のユーザー体験観測の基本

<https://newrelic.com/jp/events/2025-04-23/nru300dem>

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

このハンズオントレーニングでカバーするトピック

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

こんな方におすすめ

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

2025年4-6月期開催の資格取得向けセミナー

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

開催日程:2025年6月11日(水)15:00-17:00

開催形態:Zoomを使用したオンラインの座学形式

参加費:無料

▼お申し込みはこちらから

<https://newrelic.com/jp/events/2025-06-11/nrufsoexm>

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

後述の認定試験の合格に向けた最短の道になります！

Observabilityのスペシャリストを目指せ！ Full Stack Observability Practitioner認定試験

【この認定試験を通じて身につくスキル】

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

【認定試験に向けた準備】

- New Relicの基本的な操作経験
- ラーニングパスに沿った学習

【合格者特典】

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



認定試験受験
サイトは [こちら](#)

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

認定資格取得までのラーニングパス(出題範囲別)

認定試験の
出題範囲

APM

Infrastructure

フロントエンド

データ分析

アラート

NRU200
バックエンド機能

NRU300
ハンズオントレーニング :
アプリケーションとインフラ性
能観測の基本

NRU200
フロントエンド機能

NRU300
ハンズオントレーニング : Web
サイト上のユーザー体験観測
の基本

NRU200
データ分析 (NRQL)

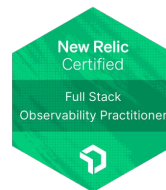
NRU300
ハンズオントレーニング : ダッ
シュボード開発と NRQL の基本

NRU200
アラート

NRU300
ハンズオントレーニング :
アラート設計の基本と活用

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

いざ受験 !



ルール別 New Relic ラーニングパス

フロントエンド
開発エンジニア

サーバーサイド
開発エンジニア

インフラ運用
エンジニア

SRE

ネットワーク
エンジニア

セキュリティ
エンジニア

プロジェクト
マネージャー

プロダクト
マネージャー

Observability
/New Relicを
知る

Observability入門

New Relic入門

サインアップ (アカウント開設)

New Relicを
使い始める

エージェント導入を体験

APMハンズオン
(Progate社提供学習タスク)

Java バッチの計測ハンズオン

ネットワークモニタリングハンズオン

初心者向けハンズオン ※オフライン開催

ご自身の環境へのエージェント導入

New Relicの
主要機能を学
ぶ

フロントエンド機能 ※サーバーサイド/インフラ担当は特にSynthetics受講を推奨

(興味に応じて左記から受講)

バックエンド機能 ※フロントエンド担当は興味に応じて受講

分散トレーシング、Errors Inbox、
CodeStream

セキュリティ
機能 (作成中)

データ分析 (NRQL) / アラート / サービスレベルマネジメント

New Relicの
使い方を体感
する

ハンズオントレーニング : アプリケーションとインフラ性能観測の基本
ハンズオントレーニング : Webサイト上のユーザー体験観測の基本
ハンズオントレーニング : ダッシュボード開発と NRQLの基本

ハンズオントレーニング : アラート設計の基本と活用

ハンズオントレーニング: IDEと連携し、
問題解決を加速するNew Relic活用の実践

ハンズオントレーニング : SLI/SLO設計の基本

© 2025 New Relic, Inc. All rights reserved.

資格を得る

フルスタックオブザーバビリティ認定試験 (受験に向けた講座あり) ※また、受験のためには上記コンテンツをできるだけ網羅的に学習することを推奨

 new relic

お知らせ

NRU401

New Relic サインアップのご案内

- 全機能が無料で使い放題（1名）！
- 転送データが 100GB/月まで無料！
- 必要事項記入ですぐに利用開始！
- クレジットカード記入不要！
- 利用期限なし！

※ New Relic フリープランで始めるオブザーバビリティ！

サインアップされた方に抽選で
New Relic Tシャツプレゼント！！

 応募フォーム



 newrelic.com/jp/sign-up-japan

New Relic 学習オンラインコンテンツのご案内

「New Relic を使ってアプリケーションを改善しよう」 がオンライン学習サービス Progate Path で公開されました！！
エンジニアの実務に即した体験ができる実践的なコンテンツ になっており、無料で学習 することができます！！



学習コンテンツは
[こちらから](#)

[学べること]

- New Relic APM / Browser の導入方法
- アプリケーションのエラーを特定して修正する流れを理解する
- アプリケーションのパフォーマンスのボトルネックを見つけて改善する流れを理解する

progate path

New Relic を使ってアプリケーションを改善しよう

オンラインIDEを起動する

このタスクについて

このタスクは New Relic 株式会社 と共同で開発したコラボタスクになっており、オブザーバビリティプラットフォームである New Relic の実務での活用方法が学べます。

概要

Express(Node.js) の API と React のフロントで構成された SNS のアプリケーションを題材に、New Relic APM と New Relic Browser を導入し、New Relic の機能を活用してアプリケーションを改善する方法を学びます。

DB (MySQL)

New Relic APM

API (Node.js/Express)

New Relic Browser

フロント (React)

目次

- 概要
- 手順
- 1. セットアップ
- 2. New Relic について
- 3. New Relic のセットアップ
- 4. データを見てみよう
- 5. エラーの発生箇所を特定して修正しよう
- 6. パフォーマンスの課題を見つけて改善しよう

チェックアイテム

タスクノート

New Relic実践入門 第2版 オブザーバビリティの基礎と実現



発売日: 2023年12月11日

価格:3,410円(税込み)

翔泳社、Amazon等から販売中

<https://www.shoeisha.co.jp/book/detail/9784798184500>

最新情報のキャッチアップにぜひ活用ください！

X(Twitter)

New Relicに関する情報発信



<https://twitter.com/NewRelicJapan>

Qiita Organization

やってみた系の記事を公開



<https://qiita.com/organizations/newrelic>

最新情報はこちら

- [What's new](#)
- [リリースノート](#)
- [公式ブログ](#)

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。

ConnpassのNRUGページより
ご登録ください。

[NRUG 本部](#)



NRUGのご紹介

イベント関連ページ：

<https://nrug.connpass.com/event/345288/>

開催日時：

2025年3月19日(水)
19:00 ~ 21:30

開催形式：

オンライン



イベントのご紹介

イベント関連ページ：

<https://newrelic.com/jp/events/2025-05-13/new-relic-now-live-tokyo>

開催日時：

2025年5月13日(火)
13:30開場、14:00~19:00

開催場所：

虎ノ門ヒルズフォーラム 4F

