

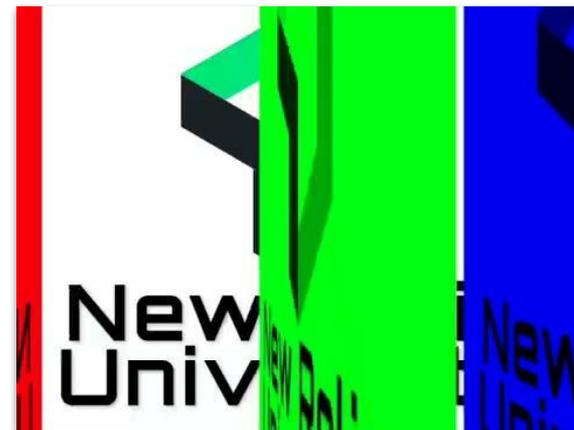


NRU 401

IDEと連携し、問題解決を加速する New Relic活用の実践

2024/03/06

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、再入室をお試しください。



ウェビナー 各種ご連絡

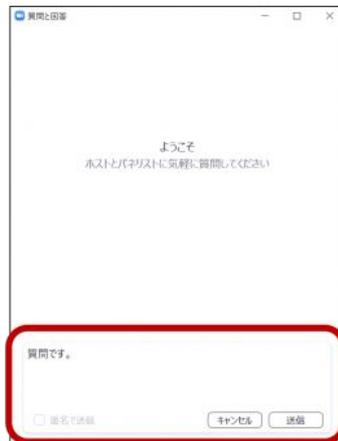
1. ご質問がある場合は、「Q&A」からご入力ください。



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
- 開発における**開発者間のコミュニケーション円滑化**
- 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

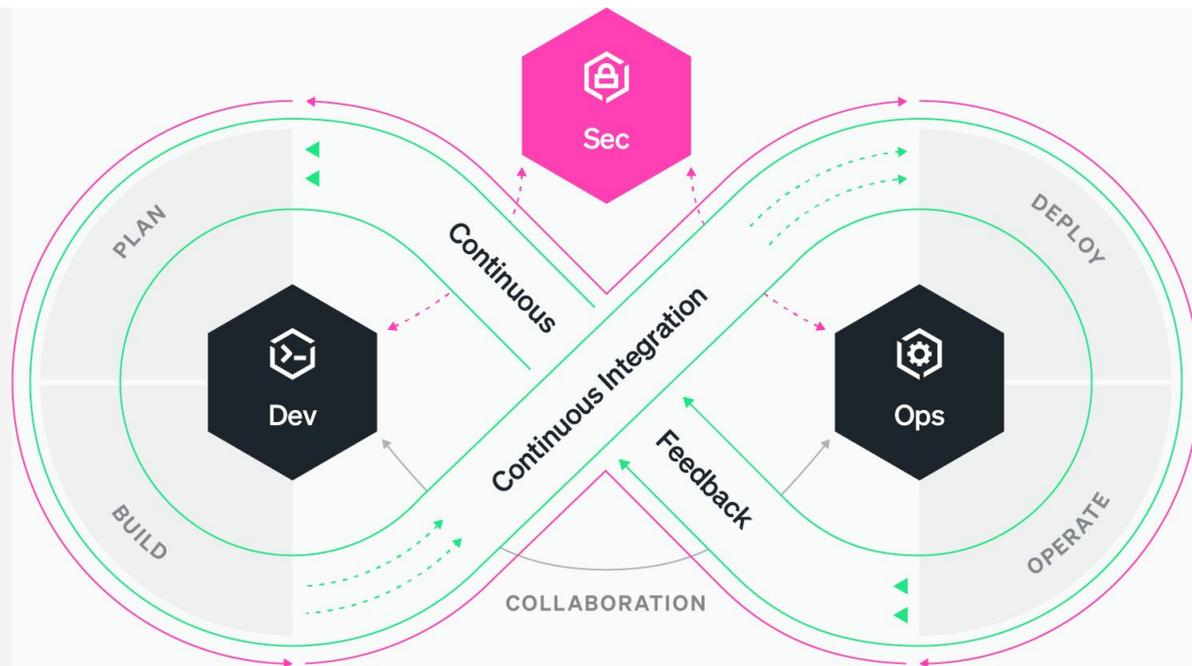
時間(目安)	内容	
15:00 - 15:30	座学	CodeStreamとは？
15:30 - 15:45	ハンズオン(0)	CodeStream環境の準備
15:45 - 16:05	ハンズオン(1)	開発者がデプロイされたシステムの状態を意識
16:05 - 16:25	ハンズオン(2)	開発者同士のコミュニケーション
16:25 - 16:45	ハンズオン(3)	開発者と運用者のコミュニケーション
16:45 - 17:00		まとめ、アンケートご記入

座学 : CodeStreamとは？

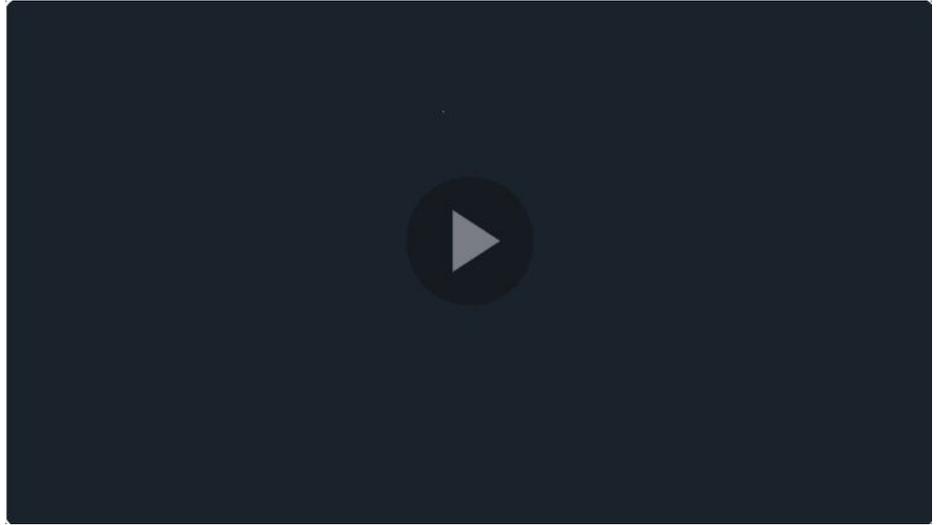


New Relic - Full Lifecycle Observability

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



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

**Focusing on
engineers and
their success**

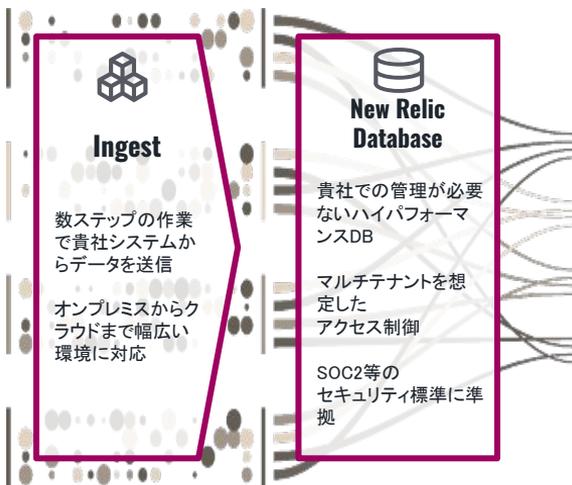
new relic. オールインワン オブザーバビリティプラットフォーム

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

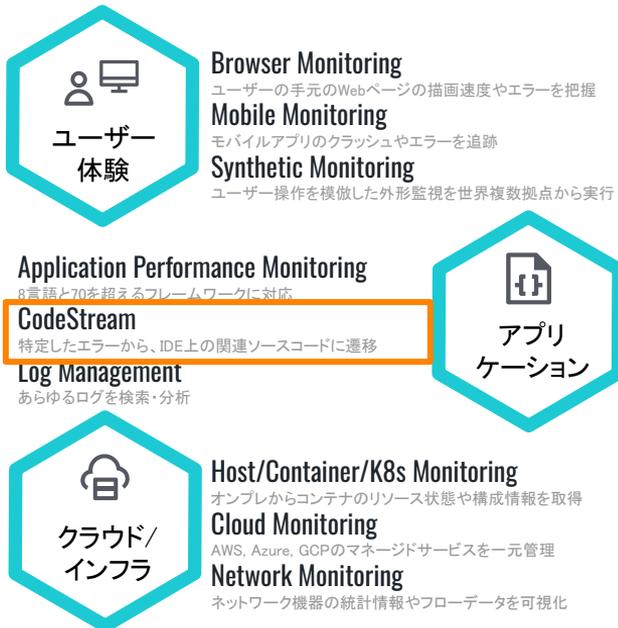


Insight Cloud – データ活用

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

開発における開発者間の コミュニケーション円滑化

リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

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

開発における開発者間の コミュニケーション円滑化

リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

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??

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！

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

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

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

メソッドレベルでもメトリクスを確認

```
11  
12  
13 @socks  
14 if @tags.find { |tag| tag == Tag.find(SockTag.find,  
15 |sock| @tags.find { |tag| tag == Tag.find(SockTag.find,  
16 }  
17 }  
18 else  
19 @socks = Sock.all  
20 end  
21 if !@page_size.nil?  
22 @socks = @socks.slice((@page_num - 1) * @page_size.to_i, @  
23 end tazuma, a year ago · fix  
24 return @socks.map { |sock| {  
25 "id": sock.sock_id,  
26 "name": sock.name
```

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

多様な開発環境で利用 & 連携可能

IDE



VS Code



VS Code
Insiders



Visual
Studio



Android
Studio



IntelliJ
IDEA



PyCharm



WebStorm



PhpStorm



RubyMine



Rider



CLion



GoLand



DataGrip



AppCode

Issue管理



GitHub



GitLab



JIRA



Azure
DevOps



Asana



Linear



Trello



YouTrack



Bitbucket



Shortcut

メッセージング



Slack



Teams

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

開発における開発者間の
コミュニケーション円滑化

リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

3

開発者と運用者の
コミュニケーション

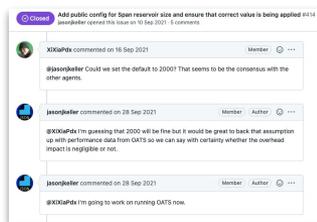
デプロイされたシステムの
エラーをブラウザとIDEで
シームレスに確認 & 対応

IDE上でコードについて直接会話！

チャットで聞いて回る



Issue/PRを漁る



Web会議でレビュー



```
public Application.configure do
  # Settings specified here will take precedence over those in config/environments

  # In the development environment your application's code is reloaded on every request. This slows down response time but is perfect for development. If you are running your server without using Rails' daemonized mode you should never have to restart the web server when you make code changes.
  config.cache_classes = false

  # Do not eager load code on boot.
  config.eager_load = false

  # Show full error reports.
  config.consider_all_requests_local = true

  # Enable/disable caching. By default caching is disabled.
  # Run rails dev:cache to toggle caching.
  if Rails.env.development? || !config.action_controller.perform_caching?
    config.action_controller.enable_fragments_cache_logging = true
  end

  config.cache_store = :memory_store
  config.public_file_server.headers = {
    'Cache-Control' => 'public, max-age=1209600'
  }
  # Don't care if the mailer can't send.
  config.action_mailer.raise_delivery_errors = false

  # Show detailed error messages on the local file system.
  config.action_mailer.raise_delivery_errors = false
end
```

```
public Application.configure do
  # Settings specified here will take precedence over those in config/environments

  # In the development environment your application's code is reloaded on every request. This slows down response time but is perfect for development. If you are running your server without using Rails' daemonized mode you should never have to restart the web server when you make code changes.
  config.cache_classes = false

  # Do not eager load code on boot.
  config.eager_load = false

  # Show full error reports.
  config.consider_all_requests_local = true

  # Enable/disable caching. By default caching is disabled.
  # Run rails dev:cache to toggle caching.
  if Rails.env.development? || !config.action_controller.perform_caching?
    config.action_controller.enable_fragments_cache_logging = true
  end

  config.cache_store = :memory_store
  config.public_file_server.headers = {
    'Cache-Control' => 'public, max-age=1209600'
  }
  # Don't care if the mailer can't send.
  config.action_mailer.raise_delivery_errors = false

  # Show detailed error messages on the local file system.
  config.action_mailer.raise_delivery_errors = false
end
```



コード上で直接会話できて便利！
しかも、リポジトリが汚れない！

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

開発における開発者間の
コミュニケーション円滑化

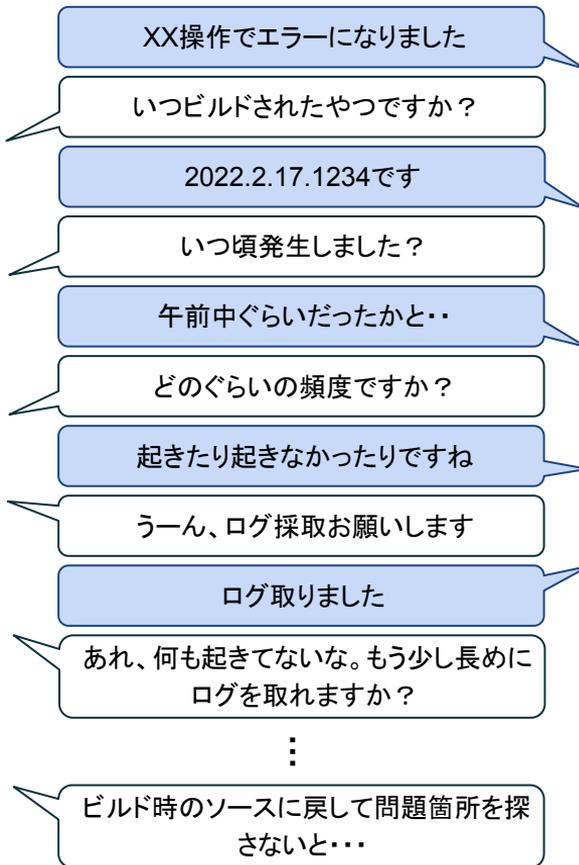
リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

3

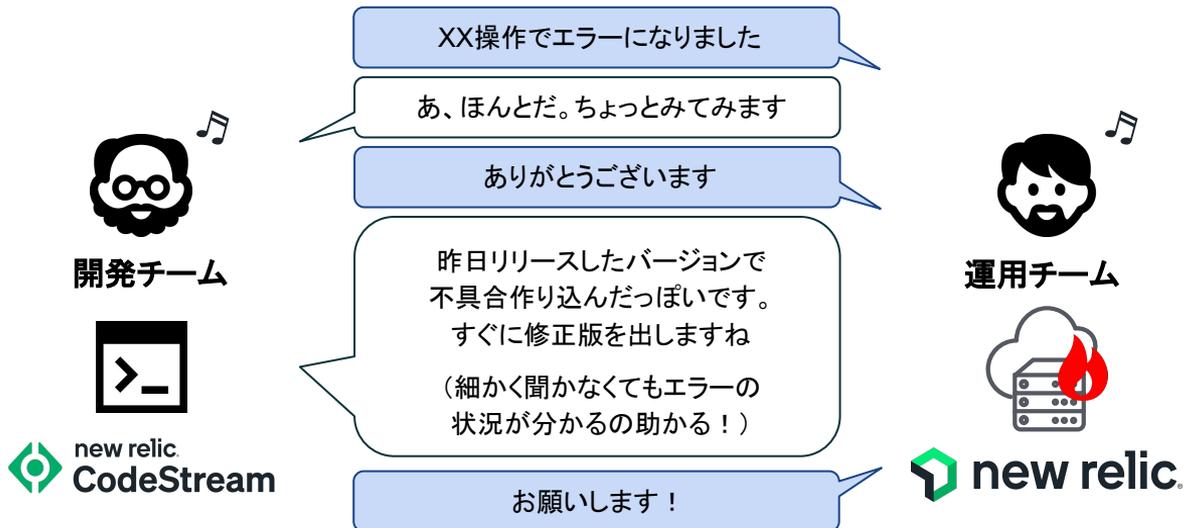
開発者と運用者の
コミュニケーション

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!

The screenshot displays the New Relic CodeStream IDE interface. On the left, a sidebar shows the project structure for 'catalogue_nru'. An 'Exception' panel is open, displaying the following details:

- Exception: Please set an Array in @tags
- Repo: catalogue
- Build: 3e0e70e
- STACK TRACE:
 - ...catalogue/app/services/socks_selector.rb:14
 - ...4-03-02/catalogue/app/services/service.rb:5
 - ...e/app/controllers/catalogue_controller.rb:9
 - ...ontroller/metal/basic_implicit_render.rb:6/
 - ...1.4.1/lib/abstract_controller/base.rb:228/
 - ...ib/action_controller/metal/rendering.rb:30/
 - ...1/lib/abstract_controller/callbacks.rb:42/
 - Unable to find matching file for path /usr/local/bundle/gems/actionpack-6.1.4.1/lib/action_controller/metal/rendering.rb

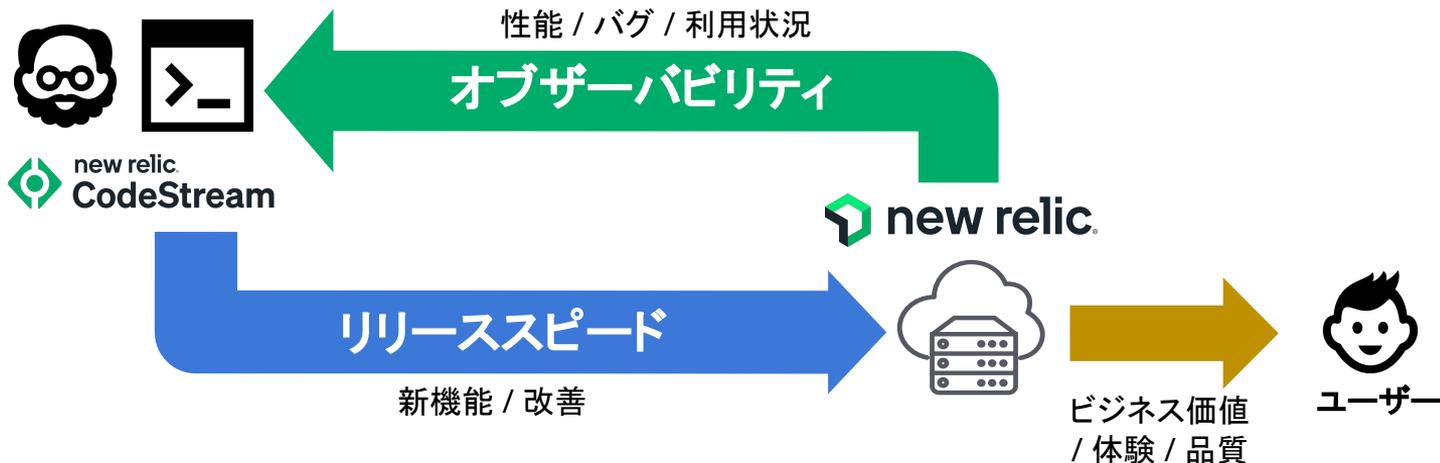
An orange callout box points to the file path in the stack trace, stating: **エラーの詳細をIDE上で確認** (Check error details in IDE).

On the right, the code editor shows the file 'socks_selector.rb (3e0e70e)'. An orange callout box points to the file name, stating: **問題が発生したリビジョンのファイルを表示** (Display file for the version where the problem occurred).

```
1 class SocksSelector
2   include ActionController::Renderable
3
4   def initialize(tags, pageNum, pageSize, order)
5     @tags = tags
6     @pageNum = pageNum
7     @pageSize = pageSize
8     @order = order
9   end
10
11   avg duration: 67.731ms | error rate: 0.00% - 7 samples in the last 30 minutes
12   def call
13     @socks = [];
14     if !@tags.instance_of?(Array)
15       raise "Please set an Array in @tags"
16     end
17     if @tags.length > 0
18       @socks = Sock.all.filter {
19         |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_by(sock_id
20       }
21     else
22       @socks = Sock.all
23     end
24     if !@pageSize.nil?
25       @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @pageSize) |
26   end
27   return @socks.map { |sock| {
28     "id": sock.sock_id,
29     "name": sock.name,
```

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(0): CodeStream環境の準備



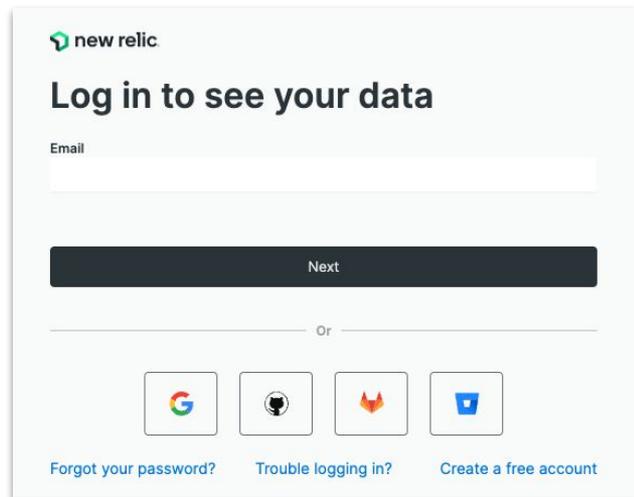
ハンズオン環境について

1. New Relicをブラウザで表示

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

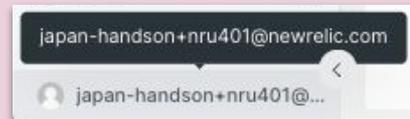
2. New Relicにログイン

- ユーザー: japan-handson+nr401@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、エネ)



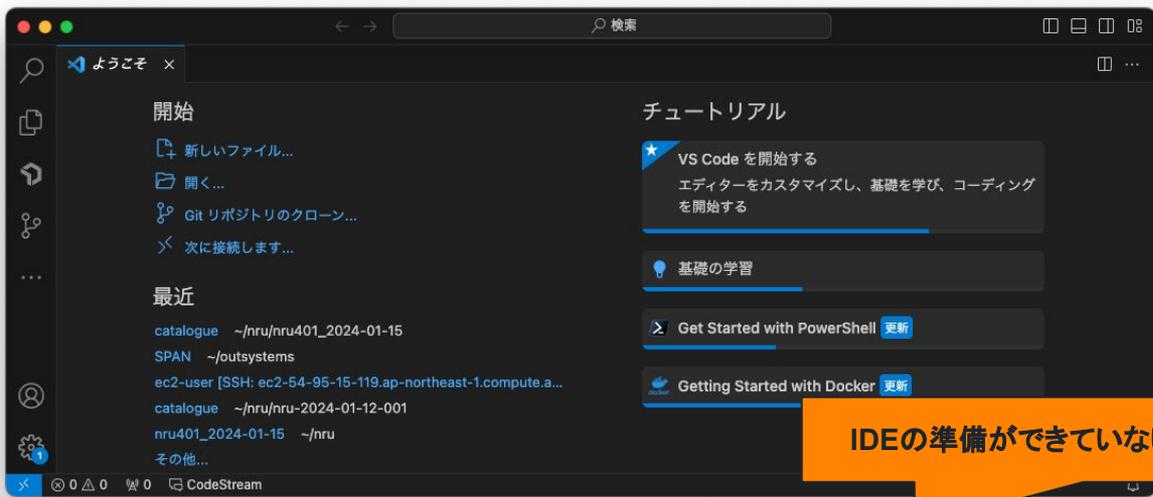
下記にご注意ください:

- 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“japan-handson+nr401@newrelic.com”であることを画面左下で確認ください



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

拡張機能: マーケットプレース

2. CodeStreamを検索

"New Relic CodeStream"

New Relic CodeStream 27ms
Bring production telemetry into your IDE to quickly identify and resolve performance issues.
New Relic

1. Extensionsをクリック

New Relic CodeStream v15.3.0
New Relic
Bring production telemetry into your IDE to quickly identify and resolve performance issues.
無効にする インストール
この拡張機能はグローバルに有効化されています。

詳細 機能のコントリビューション 変更ログ ランタイムの状態

New Relic CodeStream

Shift left by making code performance part of the earliest stages of the development process.

- Get performance data for individual methods in the code, including notifications about poorly performing methods.
- Get actionable information, such as errors, vulnerabilities and non-compliant SLOs, for the services built from your code.

カテゴリ
Other

拡張機能リソース
マーケットプレース
問題
リポジトリ
ライセンス
New Relic

master* 0 0 0 CodeStream: japan-handsonnru401 -- NORMAL --

CodeStreamハンズオン環境

The screenshot shows the CodeStream IDE interface. On the left, there is a sidebar with the New Relic logo and the GitLab logo. The main area displays a 'Sign in to...' dialog box with a 'Sign in to New Relic' button highlighted in blue. Below it, there is a link 'Don't have an account? Sign up for free.' To the right, there is a 'チュートリアル' (Tutorial) section with a 'VS Code を開始する' (Start VS Code) button highlighted in blue. The interface also shows a '最近' (Recent) section with a list of files and folders.

new relic.

GitLab

2. New Relicにサインイン

Sign in to New Relic

Don't have an account? [Sign up for free.](#)

1. GitLabからリポジトリをクローン

開始

新しいファイル...

開く...

Git リポジトリのクローン...

次に接続します...

チュートリアル

★ VS Code を開始する

エディターをカスタマイズし、基礎を学び、コーディングを開始する

基礎の学習

Get Started with Po... 更新

Getting Started wit... 更新

最近

catalogue ~/nru/nru401_2024-01-15

SPAN ~/outsystems

ec2-user [SSH: ec2-54-95-15-119.ap-no...

catalogue ~/nru/nru-2024-01-12-001

nru401_2024-01-15 ~/nru

その他...

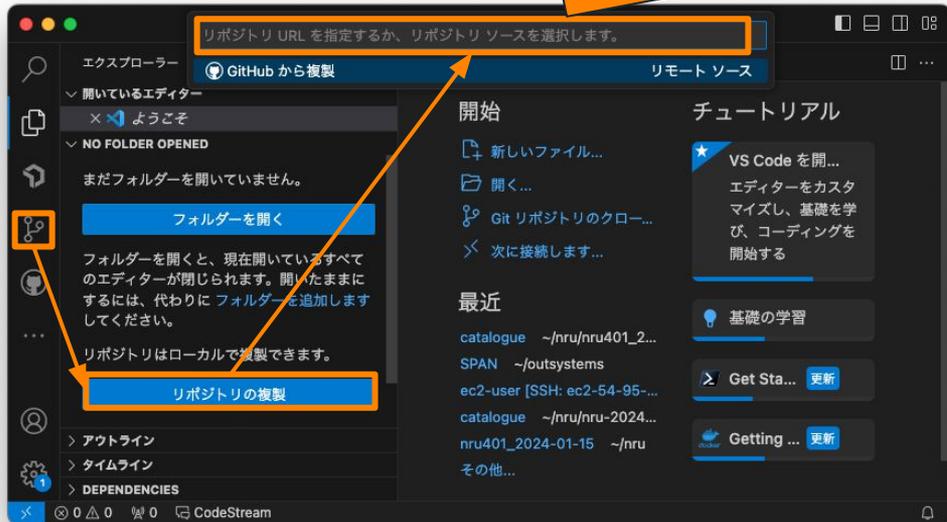
起動時にウェルカム ページを表示

※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。

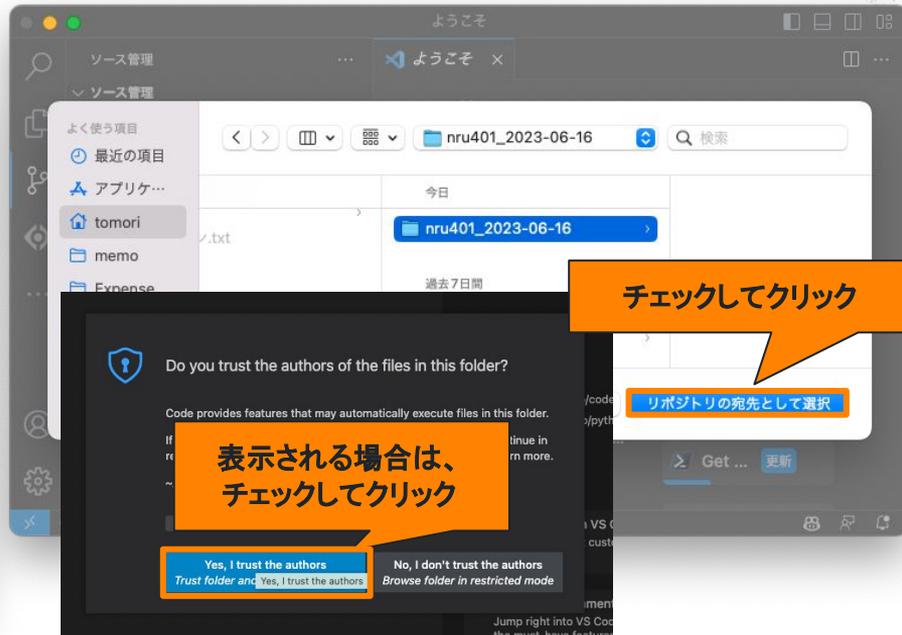
1. リポジトリをローカルにクローン

VSCodeで [リポジトリのクローン] を
クリックしてURLを入力

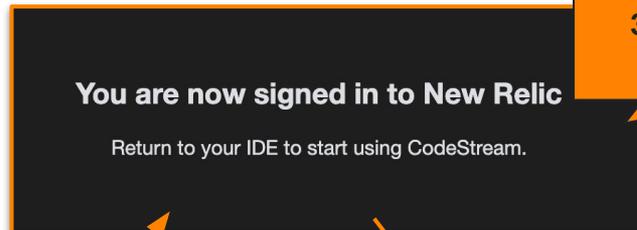
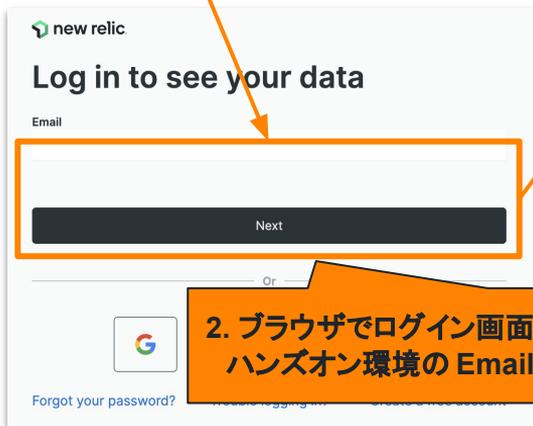
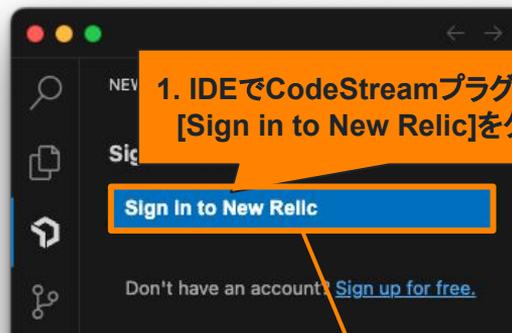
<https://gitlab.sockshop.nrkk.technology/nru/catalogue>



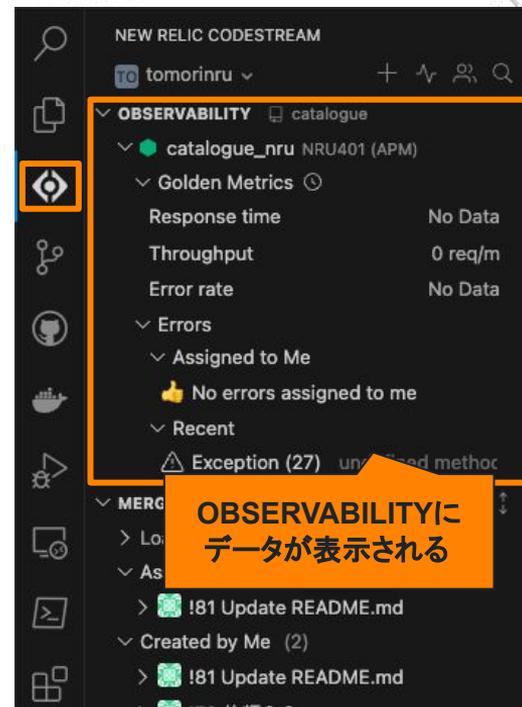
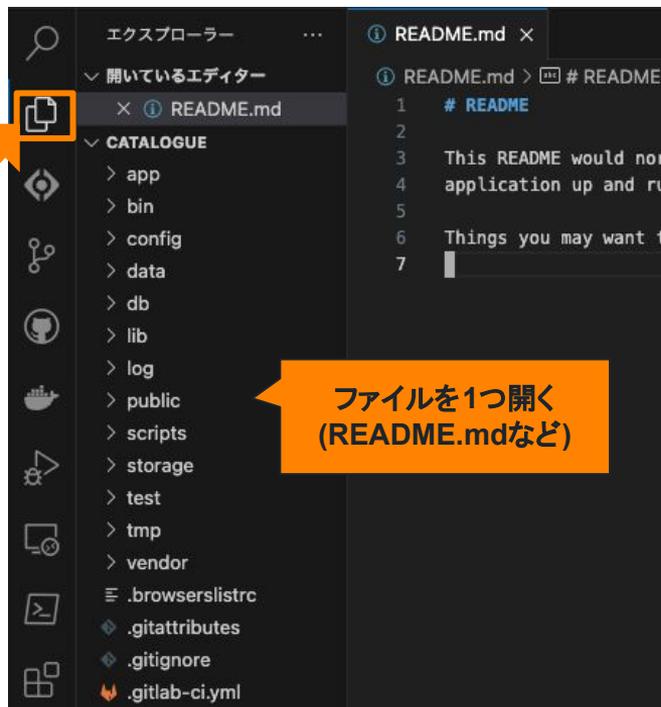
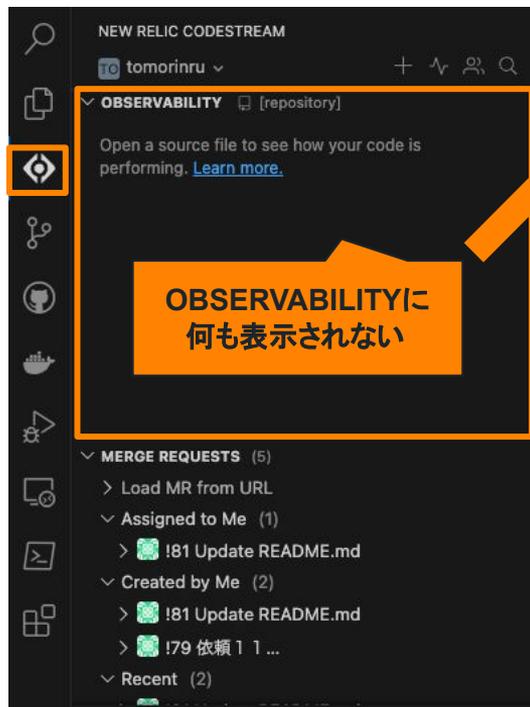
フォルダ選択してソースコードをクローン



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 システムの状態を意識



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意識

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

開発における開発者間の コミュニケーション円滑化

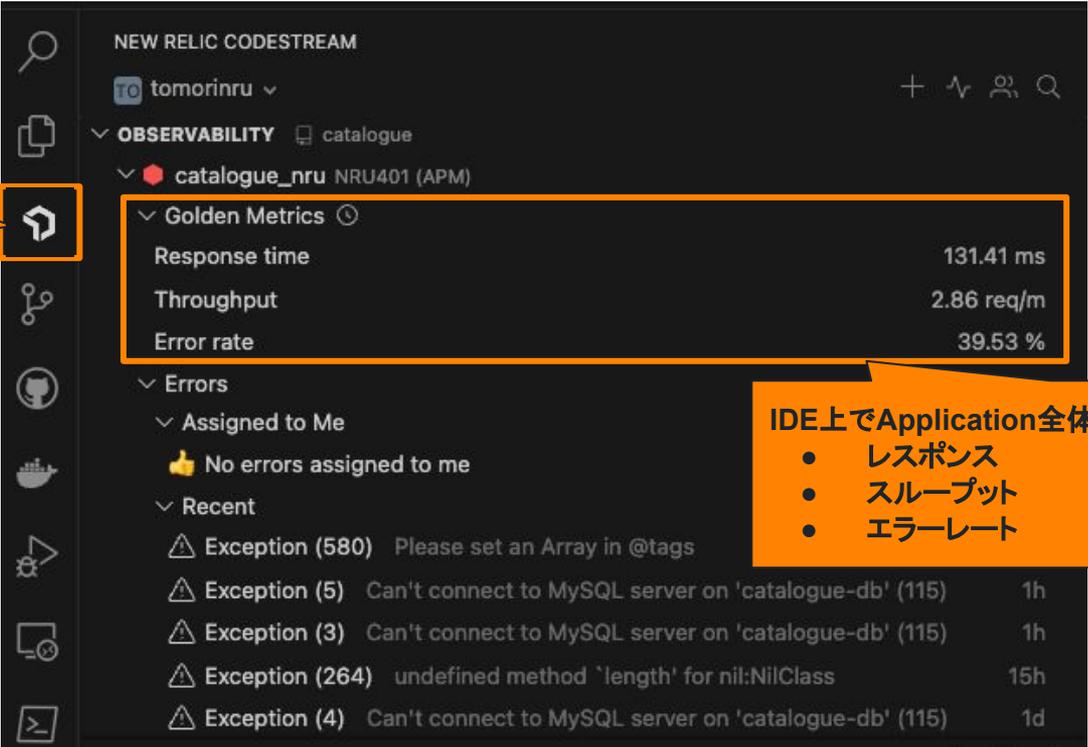
リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

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. システム全体のGolden Metricsを確認



The screenshot shows the New Relic CodeStream interface. On the left sidebar, the CodeStream icon is highlighted with an orange callout box. The main panel displays the 'OBSERVABILITY' section for a service named 'catalogue_nru NRU401 (APM)'. Under the 'Golden Metrics' section, three metrics are listed: Response time (131.41 ms), Throughput (2.86 req/m), and Error rate (39.53 %). Below this, the 'Errors' section shows a list of recent exceptions, including 'Can't connect to MySQL server' and 'undefined method `length` for nil:NilClass'.

**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. システムのログを検索

NEW RELIC CODESTREAM

japan-handsonru401

CATALOGUE ⓘ

- catalogue_nru NRU401 (APM)
- Golden Metrics ⓘ
 - Response time
 - Throughput
 - Error rate

View Logs

View Logsをクリック

ログ検索画面が表示

CodeStream (Logs) ×

Logs Learn how to search for specific log lines in your code

Service: catalogue_nru (APM Application) × ▾ 30 Minutes Ago ▾

Query logs in the selected service Query Logs

No logs found during this time range

Try adjusting your time range or [set up log management](#)

2. システムのログを検索

The screenshot displays the CodeStream Logs interface. At the top, the title bar reads "CodeStream (Logs) ×". Below it, the "Logs" section contains a search bar with the text "Service: catalogue_nru (APM Application)", a dropdown menu set to "1 Day Ago", and a search input field containing "error". A "Query Logs" button is located to the right of the search input. Below the search bar, a table of log entries is shown with columns for "timestamp" and "message". Three log entries are visible, all indicating "Completed 500 Internal Server Error".

検索対象のサービスを選択
※catalogue_nru

検索期間を選択
※1 Day Ago

検索文字列を入力
※error

検索実行

timestamp	message
> 06:53:31 PM Mar 4	[ddcecd1b-603d-42a2-9856-a401fd2fb681] Completed 500 Internal Server Error in 1ms (ActiveRecord: 0.0ms Allocations: 722)
> 06:52:57 PM Mar 4	[96eb4b0e-bd99-4431-a07e-a6823cc87797] Completed 500 Internal Server Error in 1ms (ActiveRecord: 0.0ms Allocations: 722)
> 06:51:35 PM Mar 4	[2343818c-1cc1-4341-a8a2-cfdaacde023b] Completed 500 Internal Server Error in 2ms (ActiveRecord: 0.0ms Allocations: 722)

2. システムのログを検索

クリックしてログの
他の属性値を確認

06:53:31 PM Mar 4

entity.guid	Mzk1MTEwN3xBUE18QVBQTELDQVRJT058MTA2NTA3MjU0Mg
entity.guids	Mzk1MTEwN3xBUE18QVBQTELDQVRJT058MTA2NTA3MjU0Mg
entity.name	catalogue_nru
entity.type	SERVICE
hostname	catalogue-nru-scheduled-7fb9d67ddd-8h826
level	INFO
message	[ddcecd1b-603d-42a2-9856-a401fd2fb681] Completed 500 Internal Server Error in 1ms (ActiveRecord: 0.0ms Allocations: 722)
messageId	0f0aaab8-d44d-4f17-ad61-ad48cc3041f1
newrelic.ERROR_CODE	500
newrelic.logPattern	[<UUID>] Completed<ERROR_CODE> Internal Server Error in <*> (ActiveRecord: 0.0ms Allocations: <*>)
newrelic.logs.batchIndex	4

3. Query Your DataでNRQLを実行

NEW RELIC CODESTREAM

[-] japan-handsonnru401 ▾

📖 CATALOGUE ⓘ

- ▾ 🏠 catalogue_nru NRU401 (APM)
- ▾ 📊 Golden Metrics ⓘ
 - Response time
 - Throughput
 - Error rate

Query your data

CodeStream (NRQL) ×

Query Your Data 💡 Save and share queries with .nrql files

Account: 3951107 - NRU401 ▾ Recent queries ▾

FROM

Clear Run

3. Query Your DataでNRQLを実行

CodeStream (NRQL) x

Query Your Data Save and share queries with .nrql files

Account: 3951107 - NRU401

NRQLを入力

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

FROM Transaction SELECT * SINCE 1 DAY AGO

Clear Run

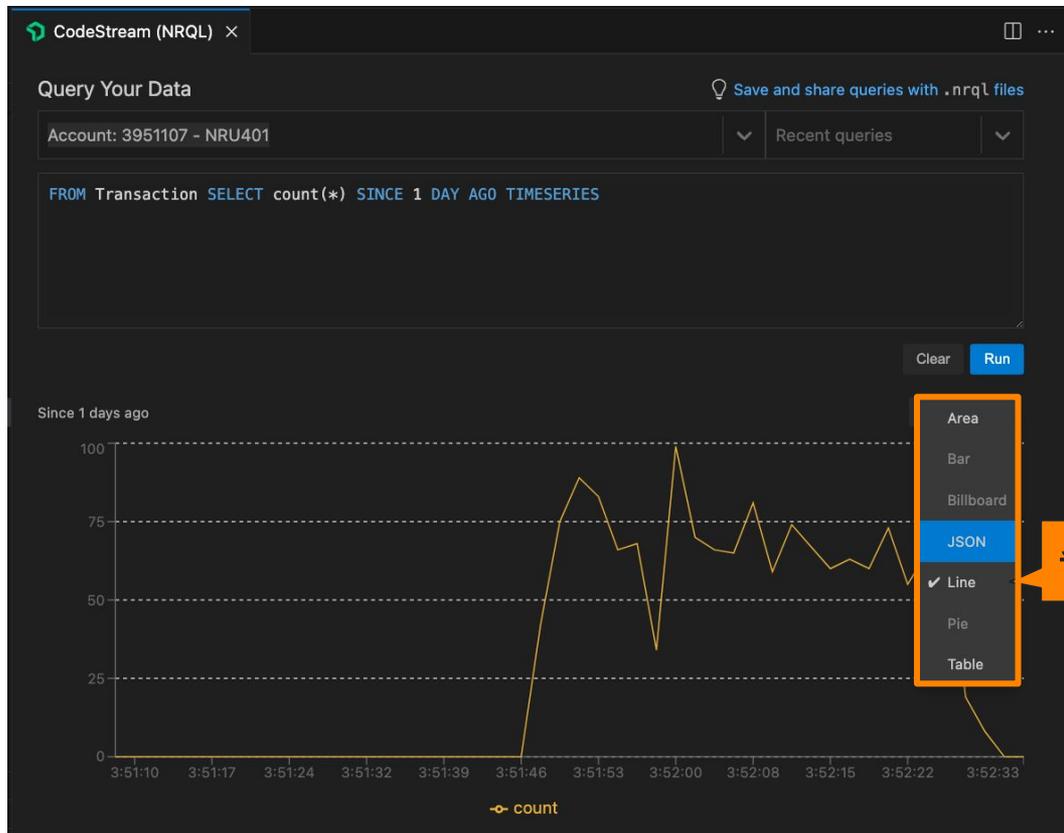
Runを実行

Since 1 days ago Table ↓

結果を表示

timestamp	request.headers.host	guid	clusterName	tags.trustedId
1709548343546	catalogue	1162ea94b5c1c57f	newrelic-catalogue	3940716
1709548335172	catalogue	3fc25ff719c0b057	newrelic-catalogue	3940716
1709548304507	catalogue	adc9431afdde02a3	newrelic-catalogue	3940716

3. Query Your DataでNRQLを実行



4. NRQLファイルからNRQLを実行

新しいファイルをクリック

.nrql拡張子でファイルを作成

RunをクリックするとNRQLが実行される

```
demo.nrql U
demo.nrql >
1 FROM Transaction SELECT * SINCE 1 DAY AGO
2
Run ▶
3 FROM Transaction SELECT count(*) SINCE 1 DAY AGO TIMESERIES
4
```

NRQLを入力

例)

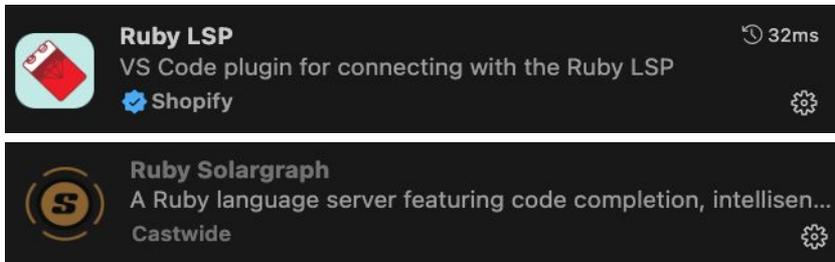
```
FROM Transaction SELECT * SINCE 1 DAY AGO
```

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

<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**か**Ruby Solargraph**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 VS Code interface with the following elements:

- Explorer (Left):** A file tree showing the project structure. The file `socks_selector.rb` is selected and highlighted. An orange callout box points to the Explorer icon with the text "エクスプローラーを開く". Another orange callout box points to the selected file with the text "CATALOGUE > services > socks_selector.rbを開く".
- Code Editor (Right):** The code for `socks_selector.rb` is displayed. A performance summary box is overlaid on the code, showing: "avg duration: 21.238ms | error rate: 0.00% - 4 samples in the last 30 minutes". An orange callout box points to this summary with the text "メソッドレベルで Golden Metricsを確認".
- Code Content:**

```
app > services > socks_selector.rb > SocksSelector > call
2  class SocksSelector
5    def initialize(tags, pageNum, pageSize, order)
9      @order = order
10   end
11
12   def call
13     @socks = [
14     if @tags.l
15     @socks = [
16     |sock|
17     }
18   else
19     @socks = Sock.all
20   end
21   if !@pageSize.nil?
22     @socks = @socks.slice((@pageNum - 1) * @pageSize.
23   end
24   return @socks.map { |sock| {
25     "id": sock.sock_id,
26     "name": sock.name,
```

<参考情報> Code Level Metricsを確認

The screenshot shows the New Relic IDE interface. On the left, a detailed view for the 'call telemetry' is displayed, showing the entity 'catalogue_nru', repository 'catalogue', and file path '/app/services/socks_selector.rb'. Below this, a line graph titled 'AVERAGE DURATION (MS)' shows data points with a peak at 97.64ms. An orange box highlights this detailed view, with a callout bubble containing the text '詳細画面が表示' (Detailed view is displayed) and an arrow pointing to the view. On the right, the code editor shows the 'socks_selector.rb' file. A callout bubble with the text 'クリック' (Click) points to a line in the code. Below the code, a summary box shows 'avg duration: 21.238ms | error rate: 0.00% - 4 samples in the last 30 minutes'. The bottom status bar indicates 'CodeStream: japan-handsonnru401 -- NORMAL --' and other settings like 'スペース: 2', 'UTF-8', 'LF', and 'Ruby'.

ハンズオン(2): 開発者同士のコミュニケーション



ハンズオン(2): 開発者同士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

開発における開発者間の
コミュニケーション円滑化

リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

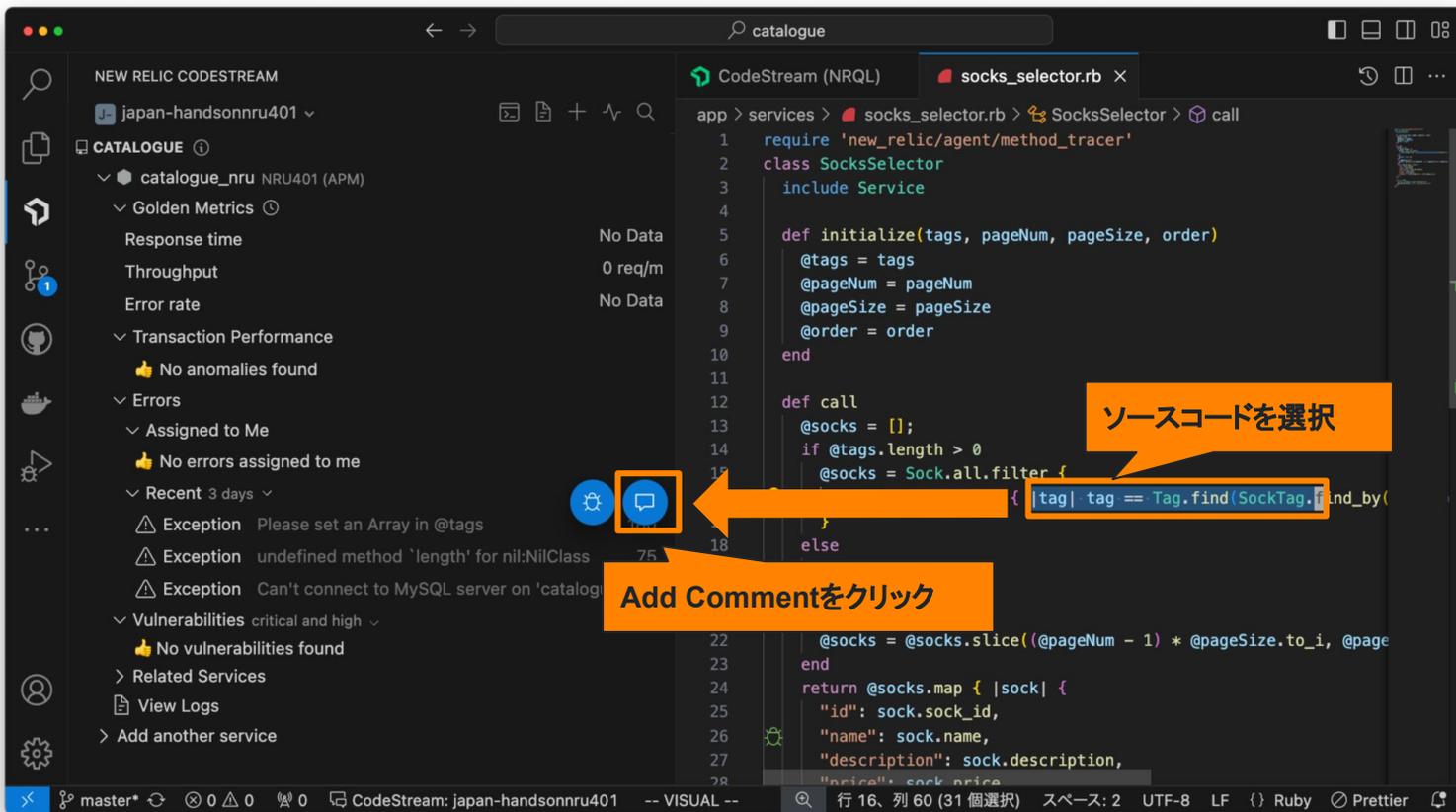
3

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

※実際にコメントの追加や参照を試してみてください

1. CODEMARKSでコメント追加

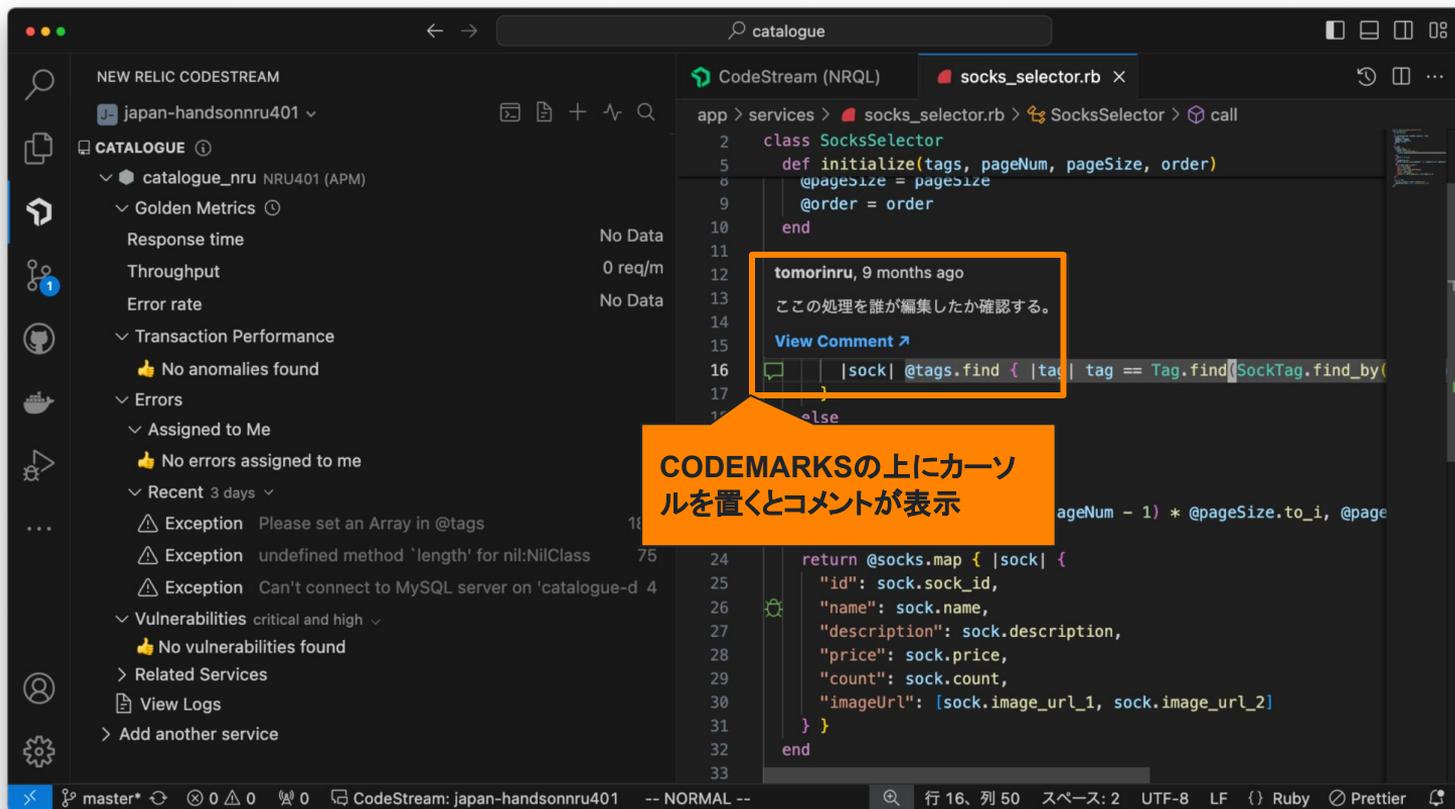


1. CODEMARKSでコメント追加

The screenshot shows the New Relic CodeStream interface. The main editor displays the code for `socks_selector.rb`. A comment dialog box is open, showing a comment being added to line 16 of the file. The comment text is `|tag| tag == Tag.find(SockTag.f`. The dialog box includes a 'Submit' button and a 'Cancel' button. An orange callout box points to the comment text with the text: **ソースコードを修正せずにコメントを追加してチーム内でコミュニケーション！**. Another orange callout box points to the 'Submit' button with the text: **Submitで追加**.

```
1 require 'new_relic/agent/method_tracer'
2 class SockSelector
3   attr_reader :page_num, :page_size, :order
4   attr_reader :socks
5   def initialize(page_num, page_size, order)
6     @page_size = page_size
7     @order = order
8   end
9   def call
10    @socks = []
11    if @tags.length > 0
12      @socks = Sock.all.filter {
13        |sock| @tags.find { |tag| tag == Tag.find(SockTag.f
14      }
15    else
16      @socks = Sock.all
17    end
18    if !@page_size.nil?
19      @socks = @socks.slice((@page_num - 1) * @page_size.to_i, @page
20    end
21    return @socks.map { |sock| {
22      "id": sock.sock_id,
23      "name": sock.name,
24      "description": sock.description.
25    }
26  end
27 end
```

1. CODEMARKSでコメント追加



2. ISSUESを追加

The screenshot displays the New Relic CodeStream interface. On the left, the 'CATALOGUE' sidebar shows a tree view for 'japan-handsonnru401' with various metrics and error logs. An orange callout box with the text 'Create Issueをクリック' (Click Create Issue) points to the gear icon in the sidebar. In the center, the code editor shows the 'socks_selector.rb' file with a Ruby class definition. An orange callout box with the text 'ソースコードを選択' (Select source code) points to a highlighted section of code in the editor. At the bottom, another orange callout box highlights a JSON object in the code: `{ "name": sock.name, "description": sock.description, "price": sock.price, "count": sock.count, "imageUrl": [sock.image_url_1, sock.image_url_2] }`.

2. ISSUESを追加

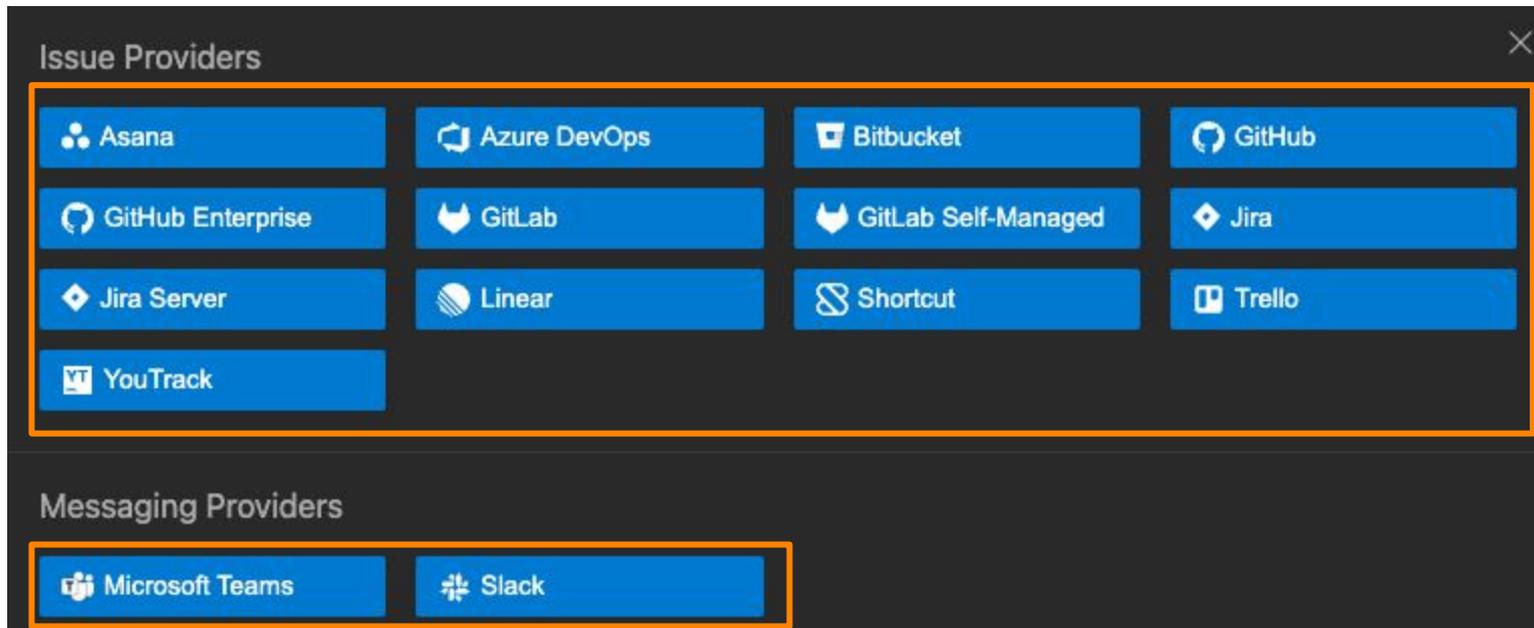
The screenshot shows the New Relic CodeStream interface. On the left, a sidebar contains navigation icons. The main area is split into two panes. The left pane displays the 'Open an Issue' dialog, which is highlighted with an orange border. The dialog has a title 'tomorinru issue in socks_selector.rb (Lines 24-29)' and a 'Create an issue on CodeStream (click for more)' button. Below this is a dropdown menu listing various issue management tools: Asana, Azure DevOps, Bitbucket, GitHub, GitHub Enterprise, GitLab, GitLab Self-Managed, Jira, Jira Server, Linear, Shortcut, Trello, and YouTrack. The 'CodeStream' option is selected and highlighted in blue. Below the dropdown, there are fields for 'Title (required)', 'Assignees (optional)', and 'Description (optional)'. At the bottom of the dialog, there is a 'Submit' button, which is also highlighted with an orange box and a callout bubble containing the text 'Submitで追加'. The right pane shows a code editor with the file 'socks_selector.rb' open, displaying Ruby code for a 'SocksSelector' class. The code includes a 'Service' module and an 'initialize' method that sets '@tags'. A 'call' method is also visible, which filters and maps 'sock' objects based on tags and pagination parameters.

CodeStreamはもちろん、
他のチケット管理ツール上にも作成可能

Submitで追加

<参考情報> CodeStreamとツールの連携

CodeStreamと他のツールを連携させることが可能



<https://docs.newrelic.com/jp/docs/codestream/code-discussion/discuss-code/>

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

開発における開発者間の
コミュニケーション円滑化

リポジトリを汚さずに
コードに関するやり取りや
チケット起票が可能

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

The screenshot shows the New Relic CodeStream interface. On the left sidebar, the CodeStream icon is highlighted with an orange callout box. The main panel displays the 'Errors' section for the 'catalogue_nru' application, showing a list of recent errors. The top error is selected, and its details are shown in a table below. An orange callout box points to the first error in the list, and another points to the 'Click once' instruction.

CodeStream プラグインを確認

IDE上でApplicationのエラーを確認できます！

Error Message	Count	Time
Exception (580) Please set an Array in @tags	580	3m
Exception (5) Can't connect to MySQL server on 'catalogue-db' (115)	5	
Exception (3) Can't connect to MySQL server on 'catalogue-db' (115)	3	
Exception (264) undefined method `length' for nil:NilClass	264	15h
Exception (4) Can't connect to MySQL server on 'catalogue-db' (115)	4	1d

1つ選んでクリック

※操作をして対象ファイルが開かれることを確認ください

1. CodeStreamでErrorsを確認

エラー詳細が表示される

Exception
undefined method `length` for nil:NilClass
Repo: catalogue
Build: 1f93390

スタックトレースを選択する

STACK TRACE
_ll(/Users/tomori/nru/catalogue/app/services/socks_selector.rb:13)
call(/Users/tomori/nru/catalogue/app/services/service.rb:5)
_s/tomori/nru/catalogue/app/controllers/catalogue_controller.rb:9)
_k-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6)
_ndle/gems/actionpack-6.1.4.1/lib/abstract_controller/base.rb:228)
_s/actionpack-6.1.4.1/lib/action_controller/metal/rendering.rb:30)
_/gems/actionpack-6.1.4.1/lib/abstract_controller/callbacks.rb:42)
_e/gems/activerecord-6.1.4.1/lib/active_record/callbacks.rb:117)
_l/bundle/gems/actiontext-6.1.4.1/lib/action_text/rendering.rb:20)

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

```
1 class SocksSelector
2   include Service
3
4   def initialize(tags, pageNum, pageSize, order)
5     @tags = tags
6     @pageNum = pageNum
7     @pageSize = pageSize
8     @order = order
9   end
10
11  def call
12    @socks = [];
13    if @tags.length > 0
14      @socks = Sock.all.filter { |sock| @tags.find { |tag| tag.id == sock.tag_id }.name }
15    }
16  else
17    @socks = Sock.all
18  end
19  end
20  if !@pageSize.nil?
21    @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @pageSize) || []
22  end
23  return @socks.map { |sock| {
24    "id": sock.sock_id,
25    "name": sock.name,
26    "description": sock.description,
27    "price": sock.price,
28    "count": sock.count,
29    "imageUrl": [sock.image_url_1, sock.image_url_2]
30  } }
31  end
32 end
```

関連するCode行が自動的にフォーカス

<参考情報>エラー発生時のリビジョンのファイル

socks_selector.rb (9d378f5) — catalogue

NEW RELIC CODESTREAM

catalogue_nru

Exception

Please set an Array in @tags

Repo: catalogue
Build: 9d378f5

STACK TRACE

```
...ori/nru/nru401_2023-06-16/catalogue/app/services/socks_selector.rb:14  
...sers/tomori/nru/nru401_2023-06-16/catalogue/app/services/service.rb:5  
...nru401_2023-06-16/catalogue/app/controllers/catalogue_controller.rb:9  
...npack-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6
```

```
1 class SocksSelector  
2   include Service  
3  
4   def initialize(tags, pageNum, pageSize, order)  
5     @tags = tags  
6     @pageNum = pageNum  
7     @pageSize = pageSize  
8     @order = order  
9   end  
10  
11   def call  
12     @socks = []  
13     if !p @tags.instance_of?(Array)  
14       raise "Please set an Array in @tags"  
15     end
```

socks_selector.rb (9d378f5) x

同一ファイルの異なるリビジョン

socks_selector.rb (1f93390) — catalogue

NEW RELIC CODESTREAM

catalogue_nru

Exception

undefined method `length` for nil:NilClass

Repo: catalogue
Build: 1f93390

STACK TRACE

```
...ori/nru/nru401_2023-06-16/catalogue/app/services/socks_selector.rb:13  
...sers/tomori/nru/nru401_2023-06-16/catalogue/app/services/service.rb:5  
...nru401_2023-06-16/catalogue/app/controllers/catalogue_controller.rb:9  
...npack-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6
```

```
1 class SocksSelector  
2   include Service  
3  
4   def initialize(tags, pageNum, pageSize, order)  
5     @tags = tags  
6     @pageNum = pageNum  
7     @pageSize = pageSize  
8     @order = order  
9   end  
10  
11   def call  
12     @socks = []  
13     if @tags.length > 0  
14       @socks = Sock.all.filter {  
15         |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_
```

socks_selector.rb (1f93390) x

<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

Account: 3951107 - NRU401

+ Create a workload + Add data

Entity Type IN Click to select more filter: List Navigator Lookout

Name ↑	Response time...	Throughput	Error rate
catalogue_nru	NRU401	-	-

ログインしたら APM から catalogue_nru 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

APM & Services / Services - APM
catalogue_nru ☆

NRU401 | 16 Show new view

Since 24 hours ago (UTC) Set up notifications

Triage Group errors

error.group.metadata.state = Unresolved AND error.expected != true Add more filters Sort by Occurrences Go to global inbox

Errors overview

Error groups (3)	New	Users impacted ?	Occurrences ?	ops (0 of 3)
Exception Please set an Array in @ta... catalogue_nru 10 days old - last seen ab...		-	203	
Exception undefined method `length' ... catalogue_nru 10 days old - last seen ab...				
Exception Can't connect to MySQL ser... catalogue_nru 19 days old - last seen ab...			20	

Errors (errors inbox) New

下記のErrorをクリック
undefined method `length' for nil:NilClass

2. New RelicでErrorsを確認

Error group summary

Unresolved 

catalogue_nru... Jun 18, 2023, 9:53am
Scheduled-8d889765d-66nbc

Only show errors with: Stack traces First < 1 of 71 > Last

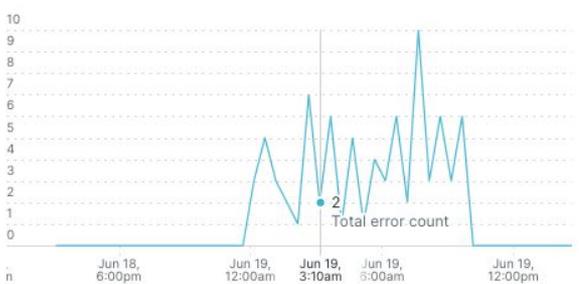
Exception
undefined method `length' for nil:NilClass

Users impacted
First seen 10 days ago
Last seen about 5 hours ago

account: NRU401 accountId: 3951107 language: ruby

**このエラーを対応するメンバーをアサインすることが可能
※ハンズオンの中ではアサインしないでください**

Occurrences 79 occurrences



Total error count

Stack trace

Copy to clipboard [Open in IDE](#)

Error message:
undefined method `length' for nil:NilClass

```

/home/appuser/webapp/app/services/socks_selector.rb:13:in `call'
/home/appuser/webapp/app/services/service.rb:5:in `call'
/home/appuser/webapp/app/controllers/catalogue_controller.rb:9:in `index'
...actionpack-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6:in `send_action'
...sr/local/bundle/gems/actionpack-6.1.4.1/lib/abstract_controller/base.rb:228:in `process_action'
...bundle/gems/actionpack-6.1.4.1/lib/action_controller/metal/rendering.rb:30:in `process_action'
...cal/bundle/gems/actionpack-6.1.4.1/lib/abstract_controller/callbacks.rb:42:in `block in process_action'
...local/bundle/gems/activerecord-6.1.4.1/lib/active_record/callbacks.rb:117:in `block in run_callbacks'
/usr/local/bundle/gems/actiontext-6.1.4.1/lib/action_text/rendering.rb:20:in `with_renderer'
/usr/local/bundle/gems/actiontext-6.1.4.1/lib/action_text/engine.rb:59:in `block(4 levels) in <class
...local/bundle/gems/activerecord-6.1.4.1/lib/active_record/callbacks.rb:128:in `process_action'

```

Show all

2. New RelicでErrorsを確認

The screenshot shows the New Relic error group summary page. On the left, the 'Error group summary' section displays the exception 'undefined method `length` for nil:NilClass' and a line graph of occurrences. On the right, the 'Discussions (2)' section shows two user comments. Two orange callout boxes highlight the 'Discussions (2)' tab and the discussion content.

Annotations:

- Discussionsをクリック** (Click Discussions)
- New RelicやCodeStream上でコミュニケーション可能** (Communication possible on New Relic or CodeStream)

Left Panel: Error group summary

- catalogue_nru
- Exception: undefined method `length` for nil:NilClass
- Users impacted: Start tracking users impacted by errors (New)
- First seen: 10 days ago
- Last seen: about 5 hours ago
- account: NRU401 | accountId: 3951107 | language: ruby

Bottom Left: Occurrences

79 occurrences

Time	Total error count
Jun 18, 6:00pm	0
Jun 19, 12:00am	4
Jun 19, 6:00am	9
Jun 19, 12:00pm	0

● Total error count

Right Panel: Discussions (2)

- tomori+nrु@newrelic.com 5 minutes ago: こちら対応おねがいできますか？
- tomori+nrु@newrelic.com 4 minutes ago: 最近頻発しているようです。

Add a comment

Submit

2. New RelicでErrorsを確認

Error group summary

catalogue_nru

Exception

undefined method 'length' for nil:NilClass

Users impacted [Start tracking users impacted by errors](#) New

First seen 10 days ago
Last seen about 5 hours ago

account: NRU401 accountId: 3951107 language: ruby

Occurrences

79 occurrences

Date	Total error count
Jun 18, 6:00pm	0
Jun 19, 12:00am	0
Jun 19, 3:10am	2
Jun 19, 6:00am	0
Jun 19, 12:00pm	0

Occurrences

Occurrencesをクリックする

Timestamp	June 19, 2023, 9:53am
Host	catalogue-nru-scheduled-8d889765d-66nbc
URL host	catalogue
URL path	/catalogue

Only show errors with: Stack traces

First < 1 of 71 > Last

Stack trace

Copy to clipboard [Open in IDE](#)

Error message:
undefined method 'length' for nil:NilClass

```
.../home/appuser/webapp/app/services/socks_selector.rb:13:in `...'
.../home/appuser/webapp/app/services/service.rb:5:in `...'
.../home/appuser/webapp/app/controllers/catalogue_controller.rb:9:in `index'
...actionpack-6.1.4.1/lib/action_controller/metal/basic_implicit_render.rb:6:in `send_action'
...sr/local/bundle/gems/actionpack-6.1.4.1/lib/abstract_controller/base.rb:228:in `process_action'
...bundle/gems/actionpack-6.1.4.1/lib/action_controller/metal/rendering.rb:30:in `process_action'
...cal/bundle/gems/actionpack-6.1.4.1/lib/abstract_controller/callbacks.rb:42:in `block in process_action'
...local/bundle/gems/activerecord-6.1.4.1/lib/active_record/callbacks.rb:117:in `block in run_callbacks'
/usr/local/bundle/gems/actiontext-6.1.4.1/lib/action_text/rendering.rb:20:in `with_renderer'
/usr/local/bundle/gems/actiontext-6.1.4.1/lib/action_text/engine.rb:59:in `block(4 levels) in <class .../usr/local/bundle/gems/actiontext-6.1.4.1/lib/action_text/engine.rb:59:in `with_renderer'`
```

Show all

Open in IDEをクリック

2. New RelicでErrorsを確認

Opening link in New Relic CodeStream...

Open in VS Code

VS Code

VS Code Insiders

Visual Studio

IntelliJ IDEA

PyCharm

WebStorm

PhpStorm

RubyMine

Rider

CLion

GoLand

DataGrip

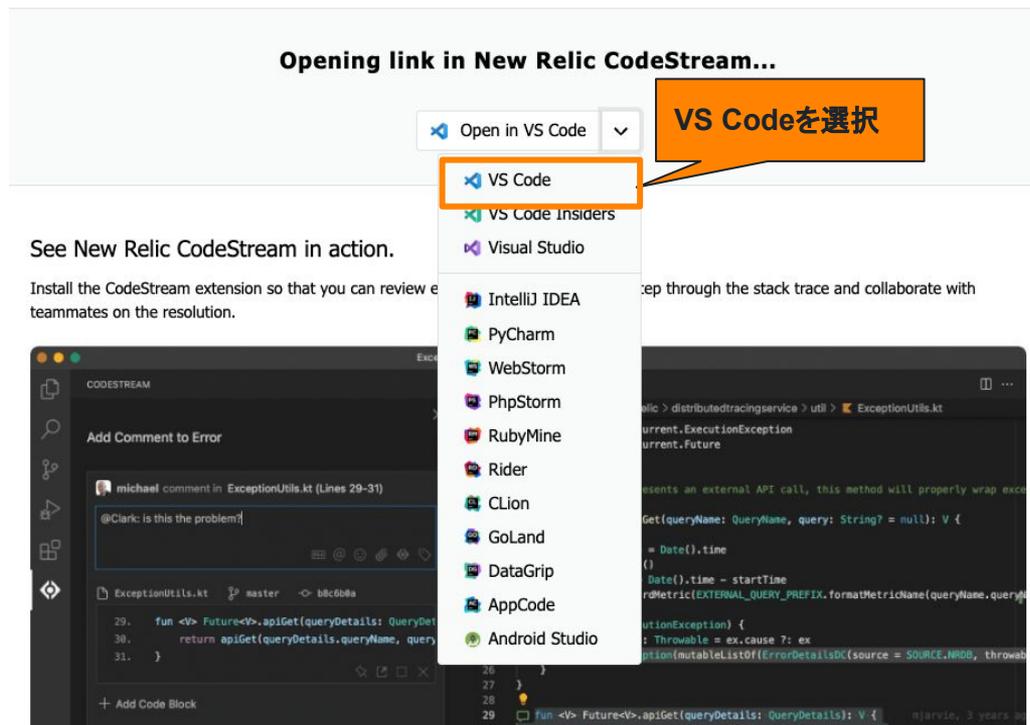
AppCode

Android Studio

VS Codeを選択

See New Relic CodeStream in action.

Install the CodeStream extension so that you can review errors and collaborate with teammates on the resolution.



The screenshot displays the New Relic CodeStream interface. On the left, there's a sidebar with a search icon and a comment section titled 'Add Comment to Error'. A comment from 'michael' is visible, mentioning 'ExceptionUtils.kt (Lines 29-31)'. The main area shows a code editor with a stack trace and a code snippet. A dropdown menu is open, listing various IDEs: VS Code, VS Code Insiders, Visual Studio, IntelliJ IDEA, PyCharm, WebStorm, PhpStorm, RubyMine, Rider, CLion, GoLand, DataGrip, AppCode, and Android Studio. The 'VS Code' option is highlighted with an orange box, and a callout bubble points to it with the text 'VS Codeを選択'.

2. New RelicでErrorsを確認

The screenshot displays the New Relic IDE interface. On the left, the 'NEW RELIC CODESTREAM' sidebar shows an 'Exception' for 'undefined method `length' for nil:NilClass' in the 'catalogue_nru' repository. The 'STACK TRACE' section highlights the error location in 'socks_selector.rb:13'. The main editor shows the code for the 'SocksSelector' class, with line 13 highlighted: `if @tags.length > 0`. The 'ACTIVITY' section at the bottom left shows recent messages from 'tomorinru'.

```
1 class SocksSelector
2   include Service
3
4   def initialize(tags, pageNum, pageSize, order)
5     @tags = tags
6     @pageNum = pageNum
7     @pageSize = pageSize
8     @order = order
9   end
10
11  def call
12    @socks = [];
13    if @tags.length > 0
14      @socks = Sock.all.filter {
15        |sock| @tags.find { |tag| tag == Tag.find(SockTag.find_by(sock_id: sock.sock_id).tag_id).name }
16      }
17    else
18      @socks = Sock.all
19    end
20    if !@pageSize.nil?
21      @socks = @socks.slice((@pageNum - 1) * @pageSize.to_i, @pageSize) || []
22    end
23    return @socks.map { |sock| {
24      "id": sock.sock_id,
25      "name": sock.name,
26      "description": sock.description,
27      "price": sock.price,
28      "count": sock.count,
29      "imageUrl": [sock.image_url_1, sock.image_url_2]
30    } }
31  end
32 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。

© 2024 New Relic, Inc. All rights reserved.

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
- 開発における**開発者間のコミュニケーション円滑化**
- 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**

New Relic

サインアップ & 応募
抽選で

New Relic
Tシャツ



プレゼント！



bit.ly/3CLGgZC

New Relicサインアップ登録で
抽選でNew Relic Tシャツが当たります！
(毎月抽選)

応募ステップは下記①と②だけ！

- ① New Relic 無料サインアップ ([Link](#))
- ② New Relic Tシャツ応募フォーム ([Link](#))



New Relic (ずっと) 無料サインアップ [Link](#)

1名のフルユーザーアクセス、100GB/月のデータ保存容量

① 仮登録への入力

The screenshot shows the New Relic sign-up page. The main heading is 'New Relic 無料サインアップ'. Below it, there is a sub-heading '仮登録' (Temporary Registration) and a note that all New Relic features are accessible with a credit card. The form fields include 'Name' (山田), 'Company Email' (太郎), and 'New Relic, Inc.'. There is a 'Sign Up' button.

② 本登録へ遷移

仮登録ありがとうございます。続いて本登録をお願い致します。

以下のボタン、またはご登録いただいたメールに届いたリンクより、サインアップのための本登録をお願いします。

※ 仮登録と本登録は同じメールアドレスをご入力ください。

本登録はこちら

③ 本登録への入力

The screenshot shows the New Relic sign-up page. The main heading is 'Create your account.'. Below it, there is a sub-heading '本登録' (Permanent Registration) and a note that no credit card is required. The form fields include 'Name' and 'Company Email'. There is a 'Start Now' button. To the right, there is a list of benefits: 'Access to everything', 'Scale as you grow', and 'Data in one place'.

④ パスワード設定

The screenshot shows the New Relic sign-up page. The main heading is 'Create your password'. Below it, there is a sub-heading 'パスワード設定' (Password Setup) and a note that the password must be at least 8 characters with at least 1 letter and 1 number or special character. The form fields include 'Email' (scarlet5.ht@gmail.com) and 'Password'. There is a 'Save password' button.

⑤ データ保存先を選択

The screenshot shows the New Relic sign-up page. The main heading is 'Tell us where to store your data'. Below it, there is a sub-heading 'データ保存先を選択' (Data Region Selection) and a note that the data center doesn't have to match your physical location. The form field is 'Data region' with options 'United States' and 'Europe'. There is a 'Save' button.



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>

New Relic University

<https://newrelic.com/jp/learn>

New Relicについて基本から応用まで学べるコンテンツです

New Relic University

New Relic One を学ぶ

自分のレベルに合わせて学びはじめよう

弊社では、New Relic Oneについて学べるコンテンツを New Relic University として無償で公開しています。これから使い始める方も、既に習熟されている方も、お客様のペースで、お好きなところから始めて頂けます。



インストール方法を学ぶ

New Relic One を実際にインストールして実証するための手順書や必要な情報について学びます。



New Relic One の全体概要を学ぶ (1月中旬公開予定)

New Relic One の全体像やライセンス形態をオンデマンドの動画視聴で学びます。



New Relic One の主要機能を学ぶ

New Relic One に含まれる TDP/FSON/NI の3レイヤーにおける主要機能を動画で学びます。



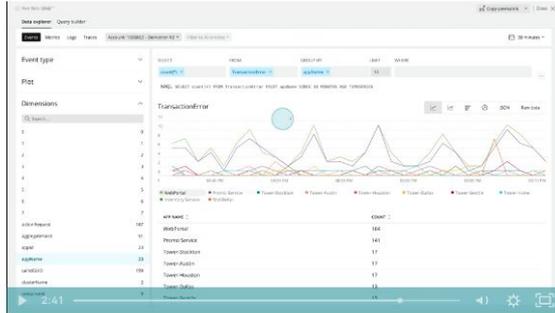
New Relic One の実践方法を学ぶ

New Relic のエンジニアが提供するハンズオントレーニングを通じて実践操作方法を学びます。

IN THIS ARTICLE

- ▶ Telemetry Data Platform >
- Event & Metrics >
- OSS Data Explorer >
- Dashboard >
- ▶ Full Stack Observability >
- Explorer >
- APM >
- APM for Serverless >
- Browser >
- Mobile >
- Synthetics >
- Infrastructure >
- Distributed Tracing >
- Workloads >
- Errors Inbox >
- Resiliency >

Event & Metrics



Data Explorer でデータを確認する

New Relic に送信されたデータを確認する方法の一つであるデータエクスプローラーをご紹介します。データエクスプローラーは New Relic One の画面上で、確認したいデータや条件を選択することで、New Relic にあるデータを簡単に表示することが出来る機能です。

New Relic University (詳細)

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

Install

New Relicを 使い始める

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

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

- ▶ [サインアップ方法](#)
- ▶ [インストールガイド](#)

NRU 100

Observability/New Relicを知る

New Relic One やオブザーバビリティに関する基礎知識を座学にて学習

[NRU Practitioner](#) オブザーバビリティ入門

[NRU 101](#) New Relic One 入門

- ▶ [オンデマンドセミナー \(practitioner\), \(nru101\)](#)

NRU 200

New Relicの主要 機能を学ぶ

New Relic One に含まれる3つの主要機能に含まれる54の機能群を動画で説明

[NRU201](#) Telemetry Data Platform
[NRU202](#) Full Stack Observability
[NRU203](#) Applied Intelligence

- ▶ [主要機能解説動画](#)

NRU 300/400

New Relicの使い方を 体感する

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

[NRU 301](#) アプリケーションとインフラ性能観測の基本
[NRU 302](#) ダッシュボード開発とNRQLの基本
[NRU 303](#) SLI/SLO設計の基本
[NRU 304](#) AIOps とアラート設計の基本
[NRU 401](#) CodeStream によるDevOps を想定したエラー分析対応の基本

- ▶ [開催スケジュール](#)

Exam

資格を得る

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

[フルスタックオブザーバビリティ認定試験](#)

- ▶ [受験サイト](#)

ロール別 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

New Relicを
使い始める

サインアップ(アカウント開設)

エージェント導入を体験

Relicstraurant(セルフハンズオンラボ)

Java パ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

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：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：SLI/SLO設計の基本

ハンズオントレーニング：AIOps とアラート設計の基本

ハンズオントレーニング：
CodeStream(エラー分析対応の基本)

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

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。

(<https://nrug.connpass.com/>)





本当にお疲れさまでした。



最後となりますが、
是非、アンケートへのご協力をお願いいたします。

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.



