CLOUDNATIVE DAYS
TOKYO2021

New Relic

3つのフェーズで組み立てる オブザーバビリティ設計入門

Koji Aizawa @kaojiri
Solutions Consultant
Nov 4 , 2021

Koji Aizawa

Business

Building a Digital Platform and Services

Cloud Computing

AWS - EKS, ECS, Fargate, etc.

Specialities

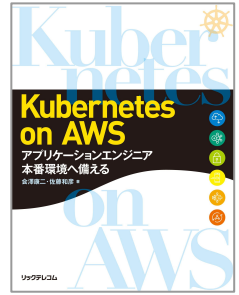
kubernetes on AWS, containers

"Containers Power to Enterprise"

CERTIFIED kubernetes ADMINISTRATOR

aws partner network

2021 APN AWS Top Engineer



Twitter: @kaojiri

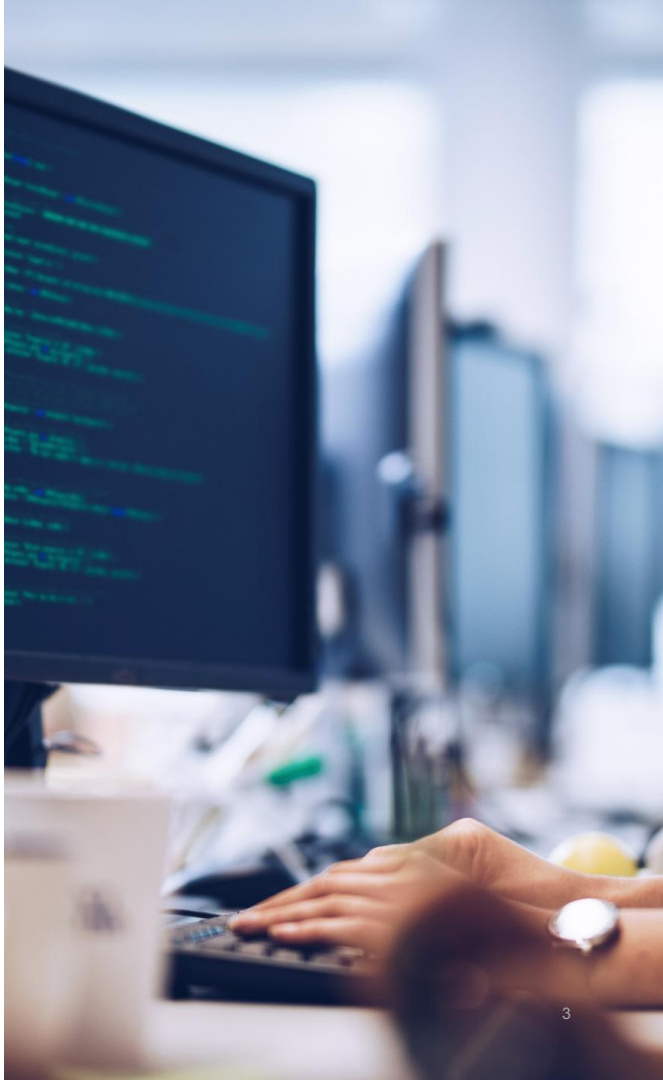
対象者とゴール

対象者

Cloud Nativeなサービスを運用するエンジニア

ゴール

- オブザーバビリティを実現するための**3つのフェーズ**を理解する
- 各フェーズを疎結合に考え、**それぞれの設計ポイントを抑えることで、Cloud Nativeなオブザーバビリティプラットフォーム**を組み立てるための勘所を掴む



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.

Cloud Native してますか？

クラウドネイティブ とは

CNCF Cloud Native Definition v1.0

抜粋: CNCF Cloud Native Definition v1.0
https://github.com/cncf/toc/blob/main/DEFINITION.md

日本語版：

クラウドネイティブ技術は、パブリッククラウド、プライベートクラウド、ハイブリッドクラウドなどの近代的でダイナミックな環境において、スケーラブルなアプリケーションを構築および実行するための能力を組織にもたらします。 このアプローチの代表例に、コンテナ、サービスメッシュ、マイクロサービス、イミュータブルインフラストラクチャ、および宣言型APIがあります。

これらの手法により、回復性、管理力、および可観測性のある疎結合システムが実現します。 これらを堅牢な自動化と組み合わせることで、エンジニアはインパクトのある変更を最小限の労力で頻繁かつ予測どおりに行うことができます。

Cloud Native Computing Foundationは、オープンソースでベンダー中立プロジェクトのエコシステムを育成・維持して、このパラダイムの採用を促進したいと考えてます。 私たちは最先端のパターンを民主化し、これらのイノベーションを誰もが利用できるようにします。

手段

状態

目的

こちらの
プラクティスは
盛り沢山

疎結合なシステム

可観測性

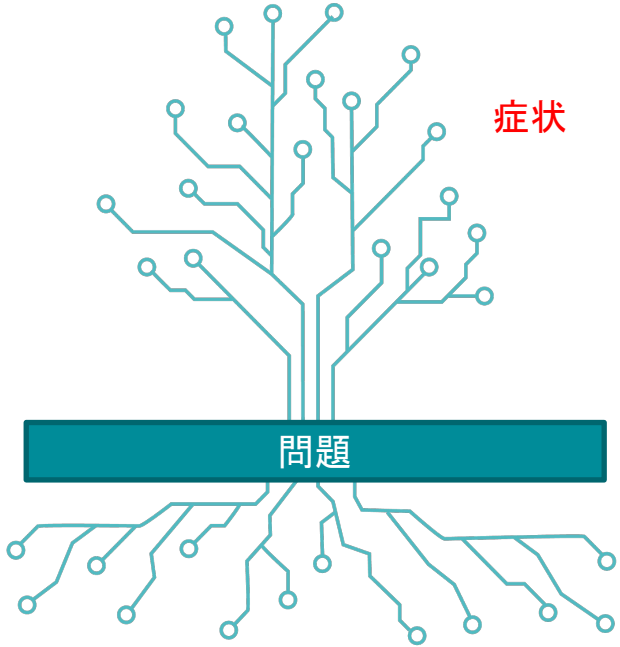
疎結合なシステム

可観測性

今日はこちらを話します

オブザーバビリティ（可観測性）とは

- Observability = 可観測性
  = 必要なデータを観測できている状態
- 「何がおかしいか」に答えられる監視から
  「なぜおかしいか、どうなおせばいいか」
  がわかるオブザーバビリティへ
- 根本原因を突き止め、
  改善につなげるアクションを取れる

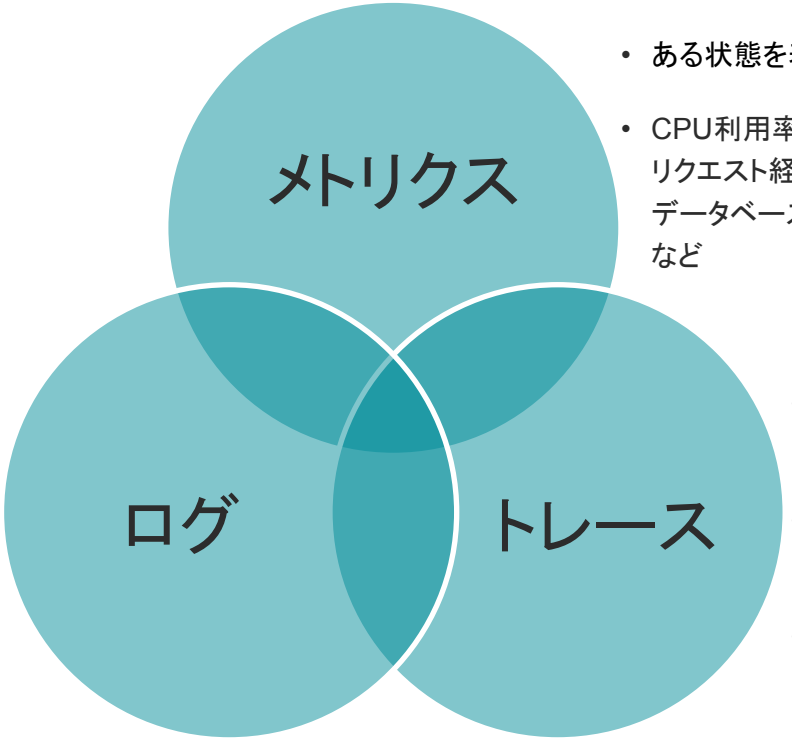


New Relic.

必要なのは根本原因の発見と継続的な改善

症状	CPU利用率90%以上	メトリクス	レスポンス遅延による パフォーマンス低下が 発生している	メトリクス
問題	（問題なし）		全体のトランザクションの うち、データベースから の取得処理を行っている サービスで時間がかかっ ている	トレース
根本原因			インデックスが貼られて おらず、フルスキャンが 走っている	ログ
改善 アクション	CPU利用率に加えて別のデータを計測し 問題ないことを把握できるようにする		インデックス設計の見直し	

オブザーバビリティに必要なデータ

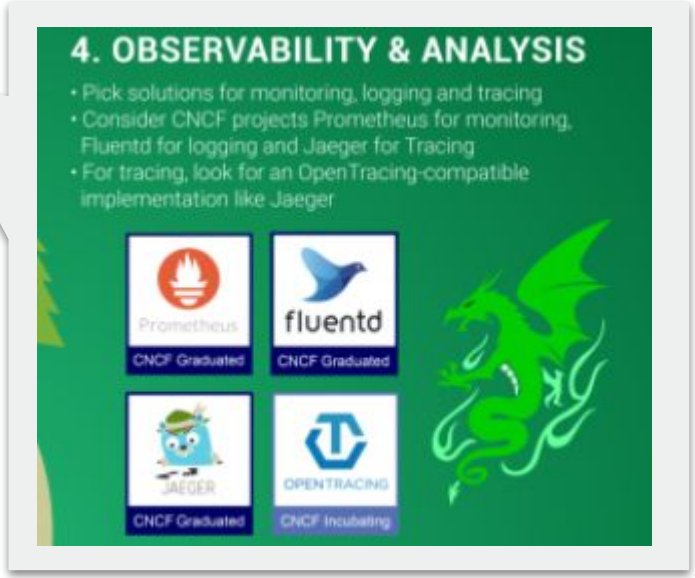


- ある状態を表す **集約**された統計値
- CPU利用率、
  リクエスト経過時間、
  データベースクエリ数
  など

- タイムスタンプ付きイベント
- プロセスから生じた
  詳細なデバッグ情報
- Error: XXX happened
  Info: YYY invoked
  など

- システムへの **リクエスト**の
  ライフサイクル情報
- 分散トレースは特に複数のシステム
  をまたいだライフサイクル
- エラーやパフォーマンス問題が起きた
  場所を特定する情報を含む

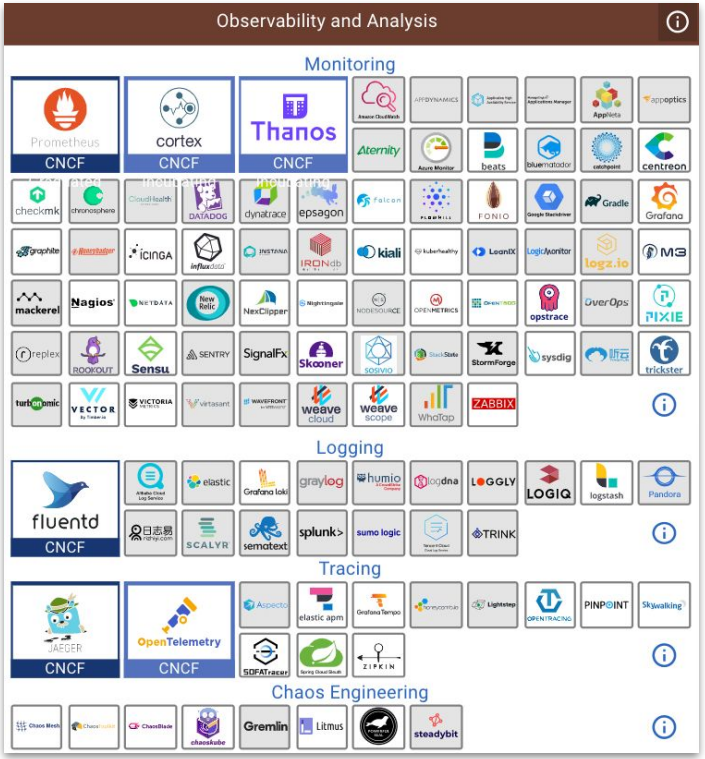
CNCFのTrail Mapによるオブザーバビリティ



https://github.com/cncf/landscape/blob/master/README.md#trail-map

CNCFによるオブザーバビリティ実現の選択肢

各フェーズを独立して設計可能な各種ソリューションが登場

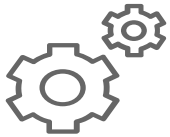


https://landscape.cncf.io/

オブザーバビリティに必要な3つのフェーズ

どうやって**収集**するか？ どこに**集約**するか？ どうやって**分析**するか？

Collect **Store** **Analyze**



オブザーバビリティプラットフォーム

モニタリングは、データを収集、集約、分析して、アプリの動作の理解を深めるためにアプリをインストルメント化することです
https://landscape.cncf.io/guide#observability-and-analysis--monitoring

CNCFによるフェーズごとの選択肢

CNCFによるオブザーバビリティ実現の選択肢：https://landscape.cncf.io/

CNCFによるフェーズごとの選択肢

CNCFによるオブザーバビリティ実現の選択肢：https://landscape.cncf.io/

1. 収集フェーズ

- 特徴
  - アプリケーションへのセットアップ時の影響が最も高い
  - アプリケーションによって言語やフレームワーク、基盤が異なる

- 課題
  - アプリケーションによって採用している方式が異なる
  - ツール変更など、何らかの変化が起こった時にアプリケーションへの影響が大きい

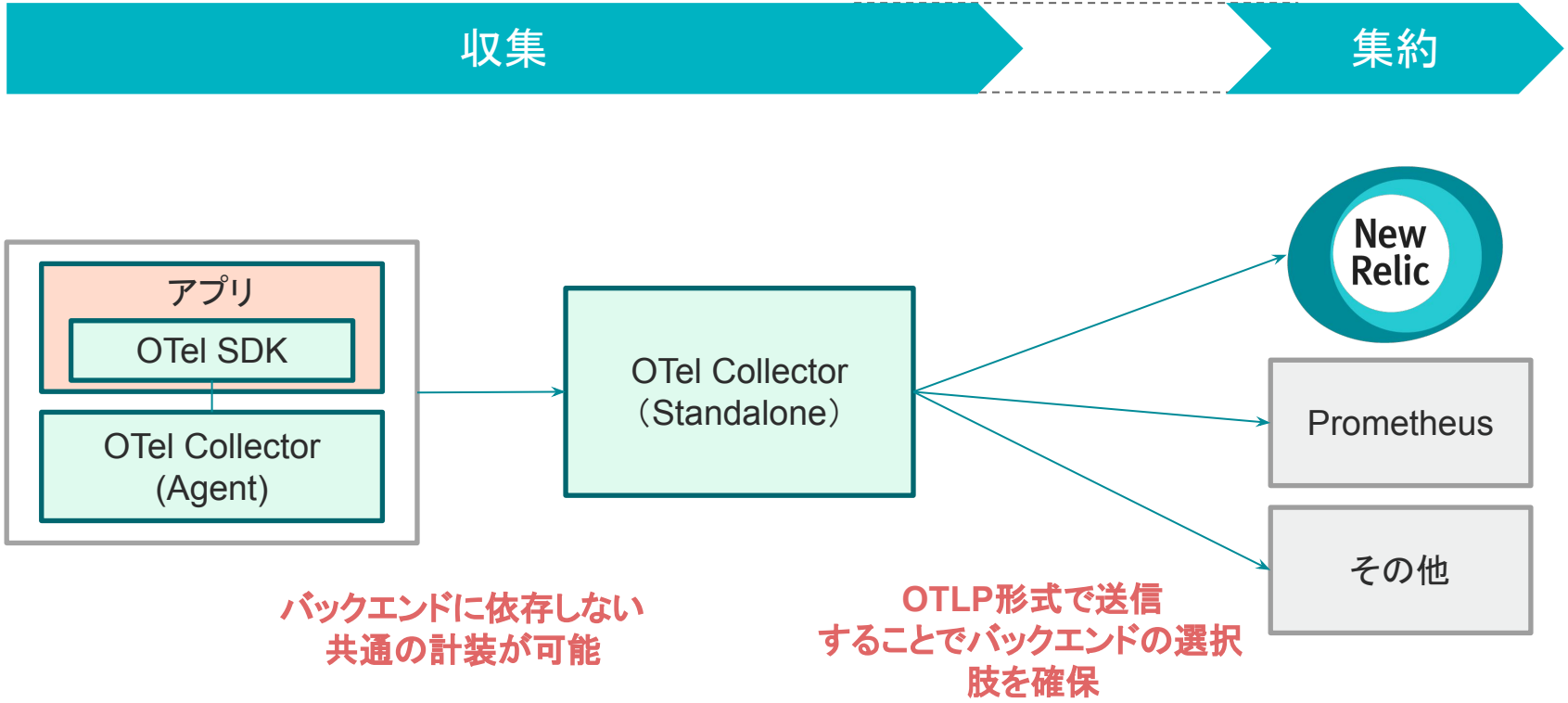
- 設計ポイント
  - 計装方式の汎用性と負荷
  - データ転送方式の汎用性

OpenTelemetryとは？

CLOUD NATIVE COMPUTING FOUNDATION

OpenTelemetry(OTel)は、
アプリケーションやサービスからテレメトリーデータを収集し、
選択したオブザーバビリティプラットフォームに送信する方法を標準化
するAPI、ライブラリ、統合、およびコレクターサービスの単一セットを提
供します

OpenTelemetryによるオープンな標準化の波



CNCFによるフェーズごとの選択肢

CNCFによるオブザーバビリティ実現の選択肢：https://landscape.cncf.io/

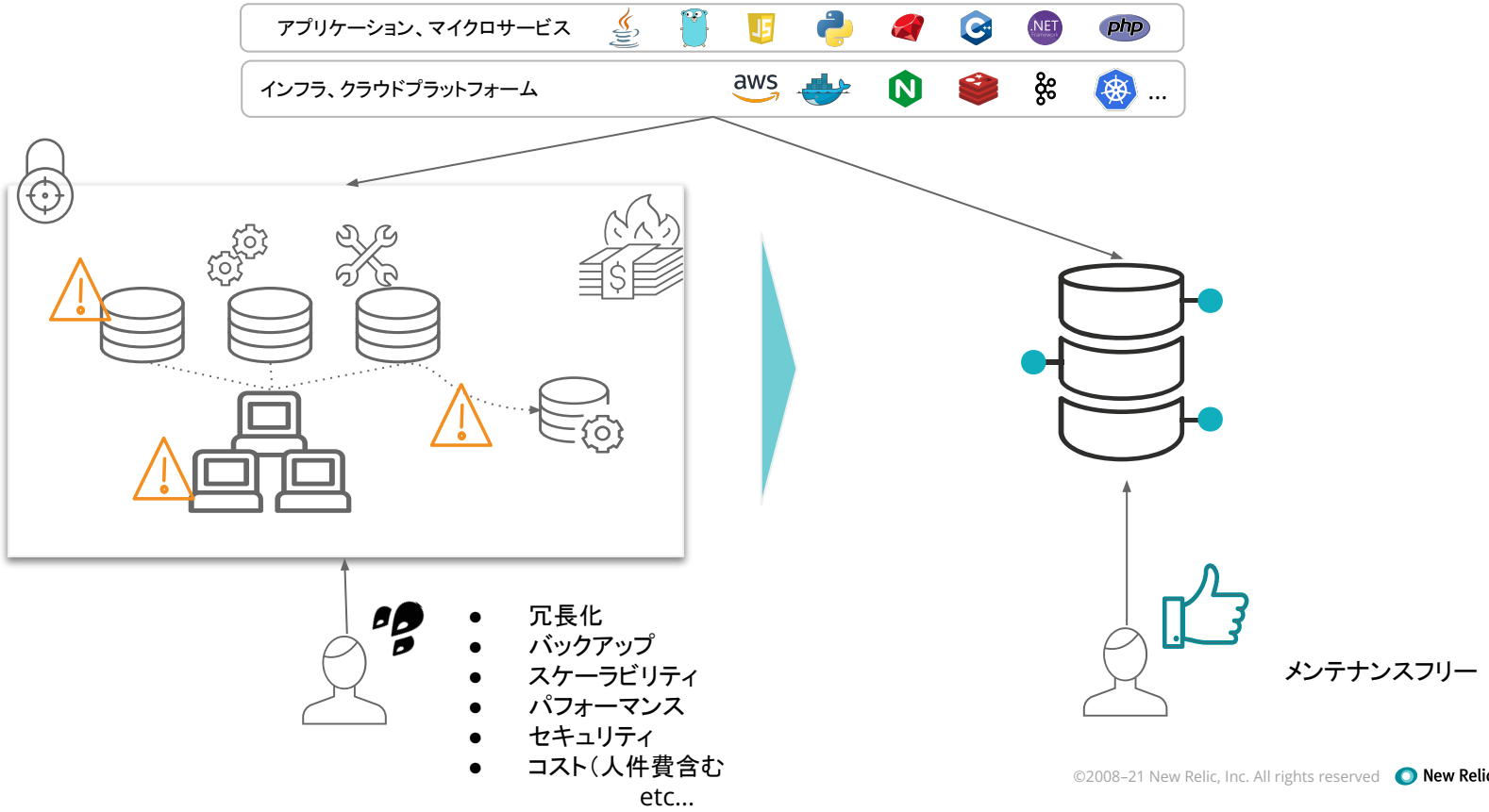
2. 集約フェーズ

- 特徴
  - 最も"利用"という観点からは遠いフェーズ
    - ストレージ、データベースの運用

- 課題
  - 最初は何も考えずに始められるが、サービスが大きくなるにつれて考えなければならないことが増え、運用負荷が増大しやすい

- 設計ポイント
  - 中長期的な運用負荷や運用コスト
    - 冗長化
    - バックアップ
    - スケーラビリティ
    - パフォーマンス
    - 運用コスト、セキュリティ etc..

2. 集約フェーズ



CNCFによるフェーズごとの選択肢

CNCFによるオブザーバビリティ実現の選択肢：https://landscape.cncf.io/

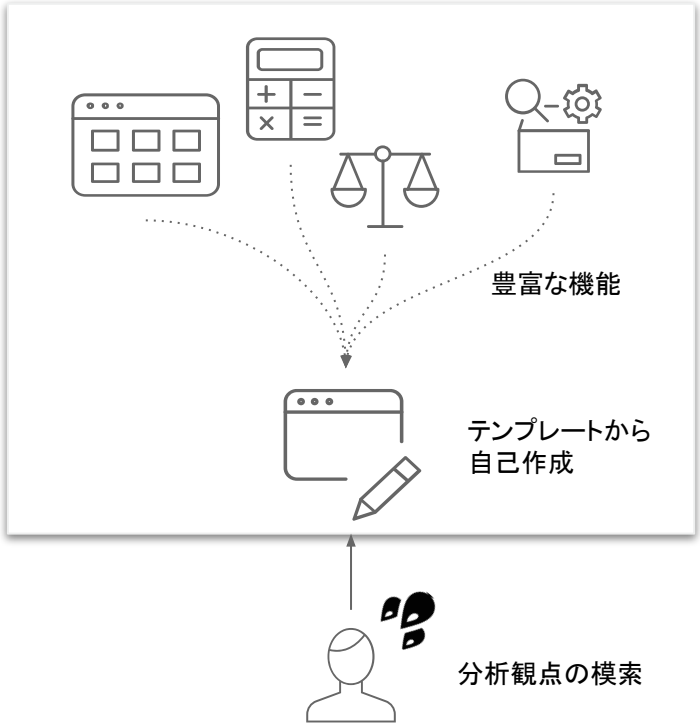
3. 可視化フェーズ

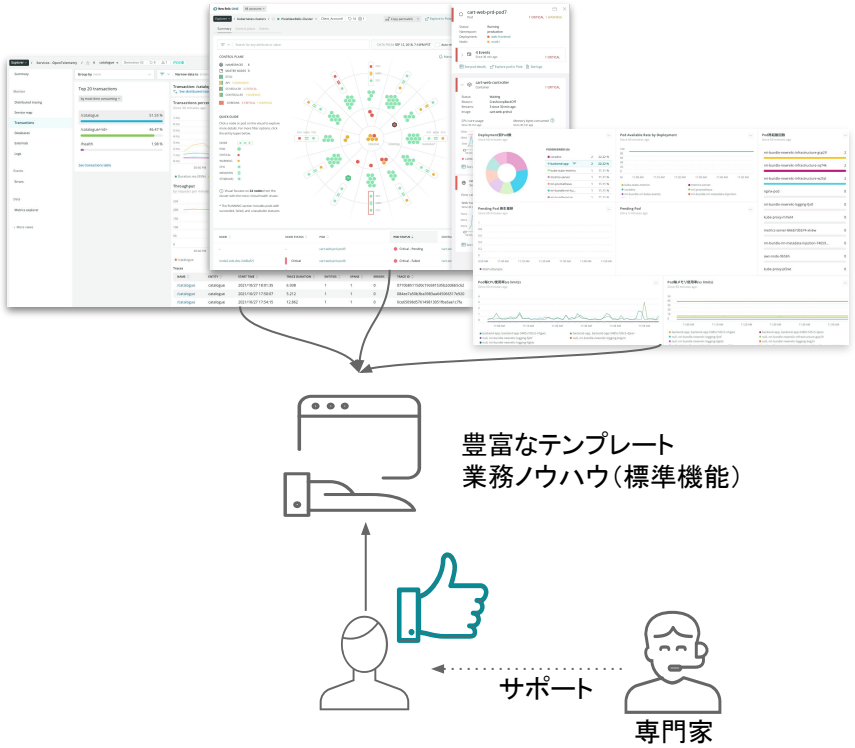
- 特徴
  - 最も関係者の目に触れやすい
  - 具体的にどう活用していくかが鍵になるフェーズ

- 課題
  - データを収集したは良いがどう分析すれば良いか分からない

- 設計ポイント
  - テンプレートの充実性や可視化柔軟性
  - 対応クエリ
  - 運用ノウハウ／サポート体制

3. 可視化フェーズ





3つのフェーズにおける設計のポイント

設計のポイント

収集	集約	可視化
● **計装負荷・方式** ● **転送方式**	● **運用負荷** ● **バックアップ** ● **スケーラビリティ** ● **パフォーマンス**	● **可視化柔軟性** ● **対応クエリ** ● **テンプレートや ノウハウの充実性**

選択肢

収集	集約	可視化
ベンダー製エージェント ・自動計装／計装負荷低 ・データ形式もベンダー固有 オープンソース ・手動計装／計装負荷高 ・任意の形式での転送が可能	ベンダーマネージドストレージ ・運用負荷低減 ・利用コストがかかる オープンソース ・利用コストはかからない ・運用負荷高	ベンダーサービス ・ベンダー独自クエリ & PromQL etc… ・コストはかかるが専門家からのノウハウを 享受可能 オープンソース ・ツール独自言語 ・テンプレートで自走可能ではあるものの 運用ノウハウを自分で開拓する必要あり

まとめ

- **オブザーバビリティを実現するために重要な3つのフェーズ**
  - 収集、集約、可視化（分析）
- **各フェーズで独立して選択できるようにすることで進化に追随する**
  - ベンダー利用＝ロックインではなくなる時代
  - 選択肢を持ち続け、適材適所でうまくベンダーを活用

各フェーズ疎結合に考え
Cloud Nativeな
オブザーバビリティプラットフォームを
実現しよう

Thank You

kaizawa@newrelic.com • @twitter kaojiri

New Relic.