

# 最新APMガイド



# クラウドネイティブ を実現するAPMとは

イノベーションを加速し、デジタルやクラウドへの投資を守るには、クラウド向けに構築された最新のインフラストラクチャ監視ソリューションとAPM(アプリケーションパフォーマンス監視)ソリューションの導入を最優先させる必要があります。

インフラストラクチャの監視を担うAPMソリューションを導入すれば、監視と分析によってアプリケーションレイヤーを可視化し、レイヤー内の問題(比率についての懸念から、エラー、期間の問題に至るまで)に迅速に対応できるようになります。「遅さ」が「ダウンタイム」と同様に捉えられるようになった常時稼働の環境では、適切なAPMソリューションを使用してサービスアーキテクチャでのリクエストの進捗を把握できるかどうか重要です。

インフラストラクチャの監視が問題の有無を見極めるのに役立つのに対し、アプリケーションパフォーマンスの監視は問題が起きている箇所を特定できるようにします。APMツールは、アプリケーションを中断することなく、適切なサービスレベルを維持できるように設計されています。社内のエンタープライズアプリケーションであっても、消費者向けアプリケーションであっても、アプリケーションの速度とアップタイムは企業の収益に直結します。環境のどこで障害が発生しているかを把握することで、インシデントをはるかに迅速に解決できるようになり、システム停止の影響を大幅に軽減できます。

しかし、これは企業が単独で実現できるものではありません。また、部分的にしか使用できないソリューションを継ぎ合わせても、イノベーションの速度が鈍るばかりか、現状の把握を誤ることさえあります。必要なのは、包括的なアプローチによってあらゆるデータの効果的な活用を可能にする、信頼できるパートナーエコシステムです。目指すべきゴールは、全体像を詳しく、常に最新の状態で把握し、すべてを考慮して生成されるアラートを活用しながら包括的な監視を行う「オブザーバビリティ」の実現です。

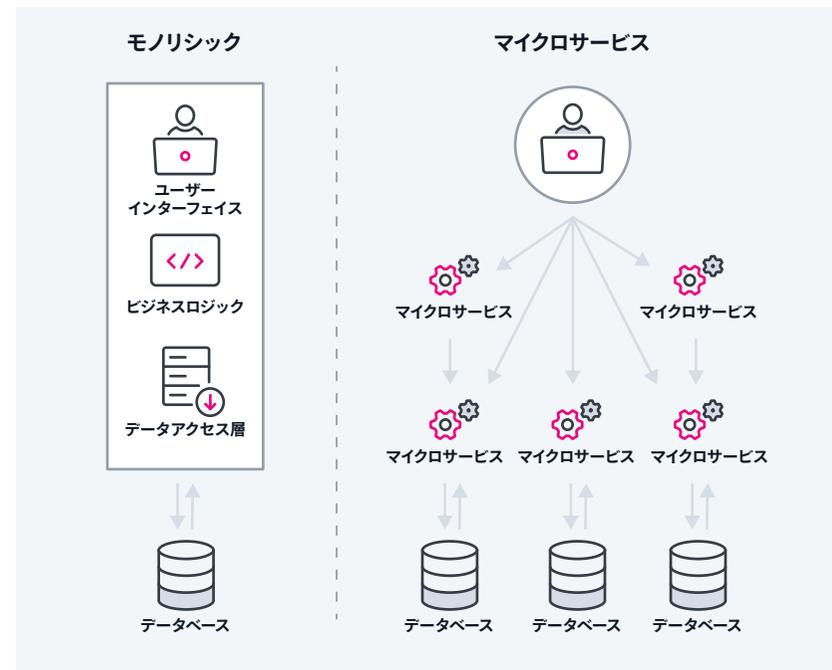
# 何が変わったのか？

多くの新興企業や先進的な企業がクラウドへの移行を加速する一方で、モノリシックなアーキテクチャで運用を続けている企業も少なくありません。モノリシックなアーキテクチャは、たとえるなら、千トンもあるような巨石です。機能や冗長なロジックが何層にも折り重なり、何千行ものコードが単一の言語で記述されています。モノリシックなアーキテクチャを小さいコンポーネントに分けるのはなかなか難しく、容易には拡張できません。必要な部分だけを拡張できず、環境全体を拡張しなければならないからです。ただし、コードプッシュは頻繁に発生しないため、監視はかなり容易です。

一方、マイクロサービスは、その巨石から切り出された大量の小石にたとえることができます。コンポーネントは分散していて、それぞれが一連の固有のビジネス機能を担っています。マイクロサービスアーキテクチャでは、異なる機能をカプセル化するためにコンテナが多用されるため、ニーズの変化に応じて各サービスを簡単に拡張できます。

このような疎結合の分散サービスは、元のモノリシックなアーキテクチャ全体にわたって処理していた機能を実行できるわけではありません。ネットワーク上でAPIにより相互通信する独立したプロセスで複雑なアプリケーションを構成する、イベントドリブンまたはサービス指向といったアーキテクチャの原則にも沿っています。

マイクロサービスアーキテクチャのアプローチで構築されるアプリケーションは、一般にさまざまな言語で開発され、それぞれのサービスは個別に拡張できます。また、各サービスが独立しているため、導入スケジュールを分けられます。このため、俊敏性が向上し、コードを頻繁にプッシュできるようになり、ユーザーからのフィードバックを短い間隔で反映できるようになります。ただし、コードをプッシュする頻度が高まり、独立したコンポーネントを頻繁に更新するため、複雑な監視プロセスを使用することになります。

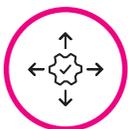


Amazon、Spotify、Uber、Grouponなど、多くの企業がモノリシックなアーキテクチャからマイクロサービス構造へとシステムを進化させてきました。Netflix社の開発者はマイクロサービスを使用して、毎日数千ものコードセクションをデプロイし、1億3,900万人の契約者と100億時間に及ぶ映画やTVドラマシリーズをサポートしています。

マイクロサービスへの切り替えが進んでいる大きな背景には、イノベーションが加速する中で、企業がビジネス上の優先事項に集中的に取り組もうとしていることがあります。同じようにスピードと成果を重視するDevOpsの広がりも、マイクロサービスへの関心と導入に拍車をかけています。

# トランスフォーメーション が起きている理由

## オンデマンドの 拡張性



アプリケーション規模を  
瞬時に拡張して、  
増大するユーザーの  
要求に対応

## 生産性の向上



新製品をより短期間で  
顧客に提供

## 問題を迅速に検出



技術者が担当サービス  
内の問題を迅速に検出

## 可用性の向上



アプリケーションを  
24時間365日稼働

## 生じている課題

従来型の監視ソリューションでは、コンテナ、Kubernetes、サーバーレス関数、マイクロサービスといった最新のテクノロジーや、DevOpsのプラクティスなどのサポートによって生じる運用上の新たな課題を解決することはできません。

しかし、なぜ企業やDevOpsチームは、このような複雑化を招くテクノロジーをわざわざ導入しているのでしょうか。

それは、クラウドやマイクロサービスへの移行を進めることで、拡張性や生産性を高め、問題をすばやく検出して解決し、アプリケーションの可用性を向上させることができるからです。そして最終的には、可能なかぎり最高のユーザーエクスペリエンスを実現できます。

監視は、マイクロサービスアーキテクチャの鍵を握る要素です。アプリケーションをコンポーネントに分割するマイクロサービスは、多くのメリットをもたらす一方、複雑化を招きます。マイクロサービスは、相互に通信する必要があり、個別に作成、更新されるそれぞれのコンポーネントは他のコンポーネントと連携しなければなりません。そのうえ、遅延を最小限に抑える必要もあります。つまり、マイクロサービスで構成されるアプリケーションの管理とは、 관련된コンポーネントからなるアプリケーション同士のネットワークを管理することです。サービス間のやりとりを効率的に監視することが、全体の信頼性の確保に欠かせません。

このように全体の信頼性を高めるのが、オブザーバビリティです。オブザーバビリティでは、新しいアプローチを取り入れてアプリケーションのパフォーマンスに関するインサイトを生成します。中でも多く採用されているのが、メトリクス、トレース、ログという3つのクラスで極めて詳細なデータを収集するアプローチです。オブザーバビリティは、詳細なデータを提供することによって、アプリケーションの「未知の未知」を調査できるようにし、思いもつかなかった疑問への答えを見つけたり、問題の発生を即座に特定できるようにします。また、オブザーバビリティは監視と組み合わせる必要があります。この組み合わせは、DevOpsの考え方を身につけている開発者の方にはおなじみかもしれません。マイクロサービスを成功させるには、このDevOpsの2つのプラクティスとともに、SDLC(ソフトウェア開発ライフサイクル)のあらゆる段階で自動化とコラボレーションを実現する必要があります。構成管理、CI/CDサーバー、APM、ダッシュボード、自動アラート、インシデント管理は、マイクロサービスを実行するチームにとっての基盤です。

## 従来のAPMでは 力不足



部分的な  
ソリューション



部分的な  
データ



分析に  
時間がかかる



拡張性に  
限界



ベンダー  
ロックイン

## 従来型のAPMで解決できない さまざまな問題

- ・ トレースデータが部分的にしか取り込まれないため、異常が見逃され、開発者による導入や開発者の生産性が低下
- ・ バッチ方式の分析のため、問題の検出が遅れる
- ・ 価格体系が複雑でパフォーマンスに問題があるため、必要でも拡張できず、成長が妨げられる
- ・ ベンダー固有の重量なエージェントがイノベーションの妨げとなる
- ・ ばらばらなオブザーバビリティツールがいくつもあるために、手順を繰り返さなければならない

## 精度の高いトレーシング、 メトリクス、ログ

アプリケーションについて把握できなければならないポイントは、主に2つあります。1つ目は、アプリケーションが正常に動作しているか、もう1つは、アプリケーションがこれまで期待どおりに動作してきたかどうかです。あらゆる開発作業を、これらのフィードバックに基づいて進めなければなりません。

包括的なフィードバックを一貫して入手できなければ、変更が適切に反映されたのか、特定のシステムでいつ問題が発生したのかなどを知ることはできません。コードの開発であれ、インフラストラクチャの構築であれ、フィードバックは成功に不可欠な要素です。

どのようにしたら、組織の規模に関係なく、適切なレベルのフィードバックによって、常に問題の核心に踏み込むことができるでしょうか。そして同様に重要なこととして、どのようにしたら問題をリアルタイムで特定して解決できるでしょうか。

## トレーシングと精度の高いトレーシング

パブリッククラウドやクラウドネイティブアプリケーションへの移行は、容易なスケーリング、処理能力の迅速な調整、データセンターの運用負荷の軽減など、組織に数多くのメリットをもたらす一方で、新たな課題も生み出しています。APMはその解決の一助となりますが、適切に行わなければ意味がありません。多くのAPMソリューションは、分析するデータの量をサンプリングによって減らしてから、パフォーマンスの低下やエラーを検出します。しかし、動作の問題に関するインサイトを正確に得るには、当然、すべてのデータを取り込んで分析する必要があります。

APMをユーザーエクスペリエンスの向上、ブランドイメージの維持、商品の迅速な市場投入につなげるには、すべてのトランザクションデータをリアルタイムで取り込むソリューションが欠かせません。OpenTelemetryなどのオープンスタンダードをベースとしているSplunk® APMなら、単一ベンダーによる制約からコードを開放し、最適な言語やフレームワークを使用できます。また、OpenTelemetryはオブザーバビリティデータの3つのクラスすべてをサポートしているため、データの利用が促進され、ベンダーロックインを防ぐことができます。



### メトリクス

基盤となるシステム、ネットワーク、ストレージのプロセス、アクティビティ、ステータスについてのインサイトをもたらす数値です。一般に、メトリクスは経時的に測定され、「時系列」と呼ばれることもよくあります。

- **システムメトリクス**  
(CPU使用率、メモリー使用率、ディスクI/O)
- **アプリケーションメトリクス**  
(比率、エラー、期間)
- **ビジネスメトリクス**  
(収益、顧客のサインアップ率、バウンス率、カート放棄率)



### トレース

トレースとは、アプリケーションでのリクエストの進行状況を記録したもので、アプリケーションを構成する無数のサービスすべてを対象とします。

個々のトレースでは、通常、次に関するデータを取得します。

- **スパン** (サービス名、操作名、時間などのメタデータ)
- **エラー**
- **各サービス内の重要な操作にかかった時間**
- **カスタム属性**



### ログ

ある期間に発生した各イベントの永続的な記録です。イベントログはプレーンテキスト、構造化テキスト、またはバイナリ形式で存在します。

- **システムログとサーバーログ**  
(syslog、journald)
- **ファイアウォールのログと侵入検知システムのログ**
- **ソーシャルメディアフィード**  
(Twitterなど)
- **アプリケーション、プラットフォーム、サーバーのログ**  
(log4j、log4net、Apache、MySQL、AWS)

## メトリクス

メトリクスは、基盤となるシステム、ネットワーク、ストレージのプロセス、アクティビティ、ステータスについてのインサイトをもたらす数値です。一般に、メトリクスは経時的に測定され、「時系列」と呼ばれることもよくあります。メトリクスが経時的に測定されるのは、システムが停滞することなく常に変化しているからです。

サービスから収集できるメトリクスは、メトリクスキー (対象サービス)と値です。これがタイムスタンプ(いつ)と組み合わせられて時系列データとなり、値を一連のデータポイントとして一定の時間間隔で図表化できるようになります。

インサイトをもたらすログやメトリクスは、アプリケーション以外からも得ることができます。ファブリック(たとえばクラウドインフラストラクチャ)、データベース、キャッシュ、キュー、サーバーなどあらゆるものがテレメトリ(アプリケーションやインフラストラクチャからの出力データ)を生成し、さまざまなインサイトをもたらします。

適切なAPMツールは、すべてを考慮して実態を正確かつ包括的に示し、誤検知を低減しながら、あらゆるアラートで詳しいコンテキストを提供します。

## トレース

トレースとは、アプリケーションでのリクエストの進行状況を記録したもので、アプリケーションを構成する無数のサービスすべてを対象とします。アプリケーションは複雑です。特定の1人がシステムのフロー全体を把握できる時代は過ぎ去りました。つまり、トレースに頼らなければ、あるリクエストがアプリケーションでどのように処理されているかを把握できなくなっているのです。

トレースからはメトリクスも得られます。この種のメトリクスは、一般に、REDと呼ばれる最新の監視プラクティスの形で提供されます。RED (比率、エラー、期間)は、個々のサービスとアプリケーションのパフォーマンスについて示唆に富んだ視点をすばやくもたらします。AIや機械学習でREDを使用すると、問題を容易に特定して、適切に対応できるようになります。

## ログ

ログエントリは、システム、アプリケーション、環境などで発生した個別のイベントの記録です。ログの形式には、Rawデータ、構造化などさまざまなものがあり、ソースもシステムソフトウェア、サードパーティソフトウェア、独自のソフトウェアなど多岐にわたります。

オブザーバビリティを獲得するには、メトリクス、トレース、ログのいずれも詳細に分析できなければなりません。メトリクス、トレース、ログは、オブザーバビリティの3つの柱です。メトリクスからは、問題があるかどうかわかります。トレースは、問題のトラブルシューティングに役立ちます。ログは問題の原因を特定する手段となります。

この3つの柱をすべて組み込んだオブザーバビリティソリューションなら、問題を検出し、根本原因を突き止めたうえでトラブルシューティングできるため、MTTRを短縮し、システムの稼働を維持できます。

# 監視を超えるオブザーバビリティへの移行

クラウドネイティブへの取り組みが進むにつれて、従来の監視方法はその役割を果たせなくなっていきます。データソースとそこから得られるデータが増え、サービスも頻繁に変更され、需要の変動が大きくなります。マイクロサービスの分散によって依存関係が複雑になるとともに、インフラストラクチャが流動化し、コードをプッシュする頻度が増え、監視はさらに困難になります。

オブザーバビリティの文化を確立し、最新のオブザーバビリティプラットフォームを活用する以外に、クラウドネイティブへの取り組みを進めて、ビジネスの成功に必要なスピードを獲得する方法はありません。必要なのは、エンドツーエンドで完全忠実な可視性を実現し、リアルタイムのストリーミング分析に基づいた結果を提供するとともに、ビジネスインサイトを提示し、AIやMLを活用してトラブルシューティングの負担を軽減する適切なソリューションです。



## カスタマーエクスペリエンスの向上

顧客に影響が出る前に問題を特定し、MTTRを向上



## 運用の予測性の向上

アラートを数秒で生成し、MTTDとインシデント対応に要する時間を短縮



## リソース効率の向上

可視性を高め、システムとビジネス健全性のメトリクスを関連付け



## 開発者の生産性の向上

開発の俊敏性と効率を高め、コードのリリースを迅速化

# Splunkのオブザーバビリティスイート

|                            |                   |                |                   |
|----------------------------|-------------------|----------------|-------------------|
| インフラストラクチャ監視               | アプリケーションパフォーマンス監視 | デジタルエクスペリエンス監視 |                   |
| ログ調査                       |                   | インシデント対応       |                   |
| NoSample™による精度の高い確実なトレーニング | リアルタイムのストリーミング    | 卓越した拡張性        | 機械学習に基づくAIドリブンの分析 |
| Open Telemetry             |                   |                |                   |
| ログ   メトリクス   トレース          |                   |                |                   |

緊密に統合された単一のユーザーエクスペリエンス

シームレスなワークフローで監視、トラブルシューティング、調査、問題解決に対応

# APMの最新化がもたらす メリット

## オブザーバビリティ

NoSample™の精度の高い確実なトレーシングを使用すれば、異常を見逃すことなく検出できます。しかも、ストリーミングアーキテクチャにより、ユーザーに影響が及ぶ前にリアルタイムで異常を検出できます。自動または手動のインストルメンテーション機能を使用して、さまざまプログラミング言語やオープンソースフレームワークでトレースデータを収集することもできます。また、問題が生じて、OpenTelemetryプロジェクトの主要な貢献者であるSplunkの世界トップクラスのエキスパートによるサポート体制が整っています。あらゆる規模のインフラストラクチャを処理、分析できるSplunk® Infrastructure Monitoringの依存関係分析により、新しいリリースが他のサービスに与える影響を即座に把握し、個々のユーザーやコンポーネントに与える影響を簡単に特定することもできます。

## 制御

あらゆるデータに対応し、ベンダーロックインを回避して、軽量なオープンソースエージェントとオープンスタンダードベースのインストルメンテーションを利用することで、価値を早期に実現することができます。一般的なOSS、クラウドインフラストラクチャ、サービスですぐに使用できる数百の統合機能が用意されているため、サービスから標準的なメトリクスを自動的に収集して、事前に構築されたダッシュボードに入力し、状況をクリアかつ迅速に可視化できます。クリック操作だけで設定できる動的なアラート条件と、サービスを自動で検出するスマートエージェント機能が備わっているため、適応性が高く、Kubernetes監視を完全に自動化することもできます。このため、トラブルシューティングや根本原因の分析にかかる時間を短縮できます。

## スピード

AIドリブンのトラブルシューティングと、わずか数秒で問題を特定して即時に生成されるアラートによって、問題をすばやく解決できます。サーバーレス関数の監視もわずか数秒しかかかりません。最も重要なメトリクスが反映された、使いやすい高解像度のダッシュボードとグラフはリアルタイムで更新され、インフラストラクチャ全体のライブヒートマップを、インテグレーションされた単一の画面で確認することもできます。このため、パフォーマンスの低下を最小限に抑えて、アプリケーションが提供する最高のエクスペリエンスをユーザーに届けるという最終的な目標を達成できます。

# 今すぐ始めましょう

ここまで述べてきたことから、かろうじてニーズを満たすAPMツールと、現在と将来にわたってニーズを満たす適切なAPMツールの違いを理解していただけたことと思います。前述したすべての条件を満たすソリューションを選ぶことしか、ゴールの見えない運用を終わらせる方法はありません。

最高レベルのビジネスパフォーマンスを維持し、ダウンタイムを最小限に抑えて、世界トップクラスのデジタルエクスペリエンスを提供できるように支援するソリューションの詳細を、ぜひご覧ください。

詳細情報

splunk®

© 2020 Splunk Inc. 無断複写・転載を禁じます。Splunk, Splunk>, Data-to-Everything, D2EおよびTurn Data Into Doingは、米国およびその他の国におけるSplunk Inc.の商標または登録商標です。他のすべてのブランド名、製品名、もしくは商標は、それぞれの所有者に帰属します。

20-16346-SPLK-The Guide to Modern APM-JA-202102

