

クラウドコストを 50%削減！ モニタリング活用 4 つの勘所

Tomas Dittmer 氏

TravelSupermarket 社、ヘッドアーキテクト
MoneySupermarket Group



DATADOG



Travel
Supermarket

MoneySupermarket
Group

著者について

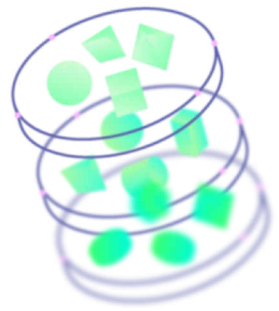
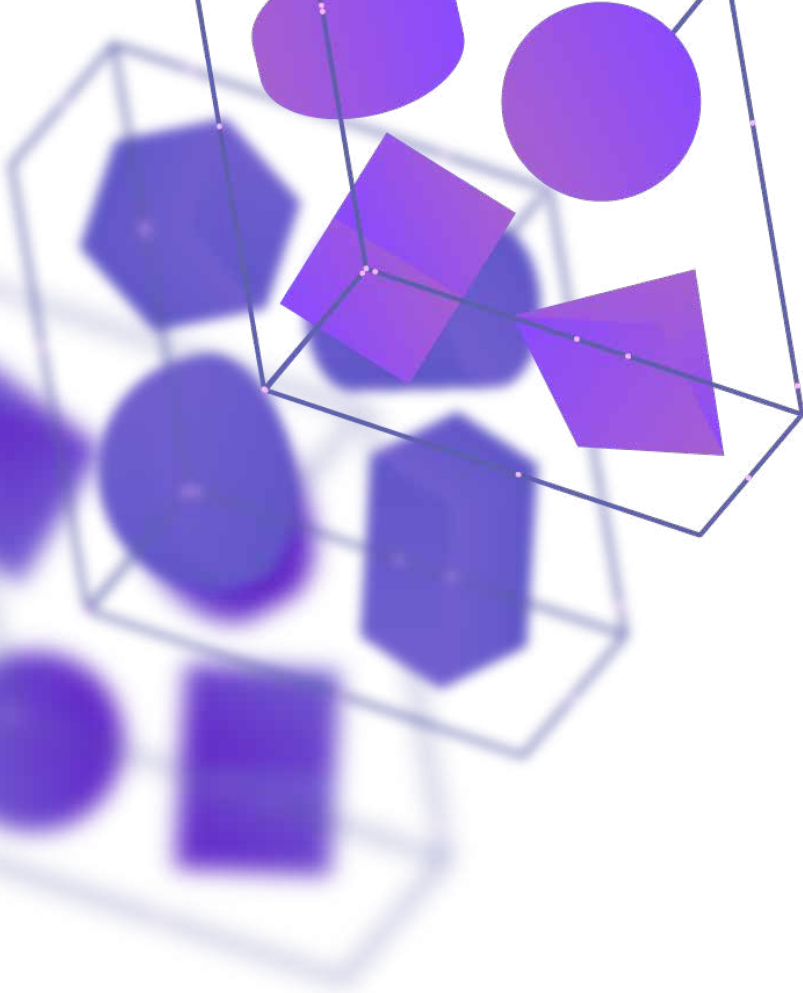


Tomas Dittmer 氏

Thomas Dittmer 氏は、価格比較サービスを提供している英国の大手企業であり、FTSE250 種総合株価指数に含まれる MoneySupermarket Group の関連企業である TravelSupermarket 社の主任アーキテクトを努めています。Dittmer 氏は、TravelSupermarket 社でベストプラクティスを推進し、同社の製品エンジニアリング組織の効率性を継続的に改善する業務を行っています。Dittmer 氏は、ソフトウェア開発、アジャイルデリバリ、技術チームのリーダーとして 15 年近くの経験があります。

目次

03	はじめに
04	本番環境の規模の適正化
05	リソースのサイズ変更
05	不要なリソースをオフにして、さらに効率的なアーキテクチャに移行する
06	キャパシティの事前購入
07	不要なテスト環境の排除
07	本番環境外のテストは本当のセーフティネットにあらず
07	複数の環境はチームの負担となる
08	本番環境の監視
08	少数の大規模なデプロイから多数の小規模なデプロイへ移行する
08	インシデントの早期解決
09	ロールバックではなくフォワードフィックス
09	パートナーとサードパーティの監視
10	チーム構造、文化、プロセスの変化に対応する
10	テクノロジー指向から製品指向へ
11	サービスについての全体的な責任
11	スペシャリストからジェネラリストへ
12	まとめ
13	謝辞



はじめに

クラウドインフラストラクチャのスプロール化は、個々の意思決定が積み重ねられて、時の経過とともに知らぬ間に進行します。これは、組織全体で見ると非効率な結果を生み出します。

当社では、運用コストの一部を見直すことになりましたが、その筆頭に挙げられたのはクラウドインフラストラクチャでした。MoneySupermarket Group PLC は、英国における AWS の戦略的クライアントであり、大規模な IaaS インスタンスを使用して運営しています。そのため、開発と本番環境の両方でこれらのクラウドリソースの活用を最適化することが、コストを管理するために重要となっていました。

私たちは、インフラストラクチャをさらに効率化できること、そして、ツール、プロセス、組織構造、マインドセットが最適化されていないためにコストが増加していることを把握していました。まず、「お金を生み出すためには、最初に投資が必要である」ということわざがあるように、この取り組みのための予算が必要となりました。しかし、後で節約できるようにするための投資が最初に必要となることもあります。

このような経緯から、1 年半前に、これらの IaaS コストを効率化することを目的に、フルスタックのパフォーマンスモニタリング機能に大規模な投資を行いました。モニタリングソリューションにかかる費用よりも、多くのコストを将来的に削減できるようになり、チームに広範な利益がもたらされることを説明して理解してもらいました。

投資を決定した時には、いくつかのモニタリング機能が既にありましたが、これらの旧来のツールは断片的な機能しか提供できず、堅牢ではありませんでした。さらに、本番環境のモニタリングがこれまで重視されておらず、本番環境の運用は本稼働前に実施される広範なテストに依存していました。

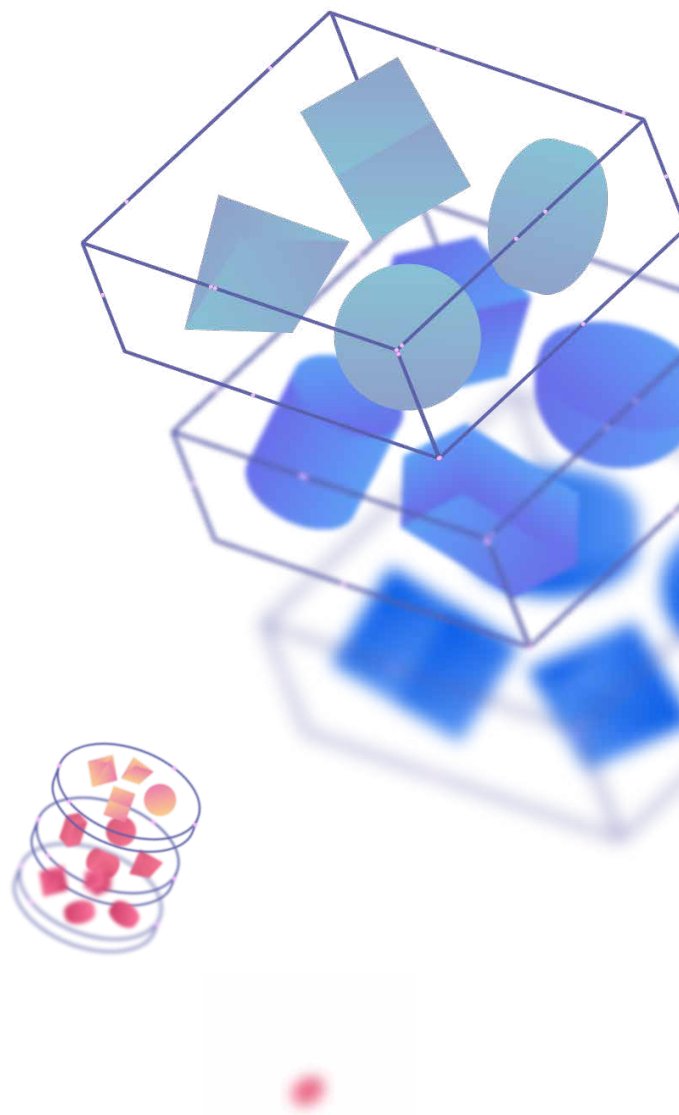
強力なモニタリングソリューションを導入し、本番環境のモニタリングを重視・強化したことで、チームの構造やデリバリモデルを根本から変革できました。以下のような利点がビジネスにもたらされました。

1. クラウドインフラのコストを 50% 削減
2. 低レイテンシにより収益性が数パーセント向上。レイテンシの低下は、トラフィックが増大するオンライン横断検索ビジネスでは非常に大きなアドバンテージ。
3. チームのサイクルタイムの短縮、インシデントの減少、修正の迅速化

本番環境の規模の適正化

最初に、簡単な問題から解決することにしました。利用状況とニーズを把握して、本当に必要なリソースのみを購入するようにしました。この取り組みでは、主

次のセクションでは、本番環境でのパフォーマンスモニタリングへの投資により、プロセス、組織構造、マインドセットがどのように変化し、クラウドのコスト効率がどうして大幅に向上したかを説明します。

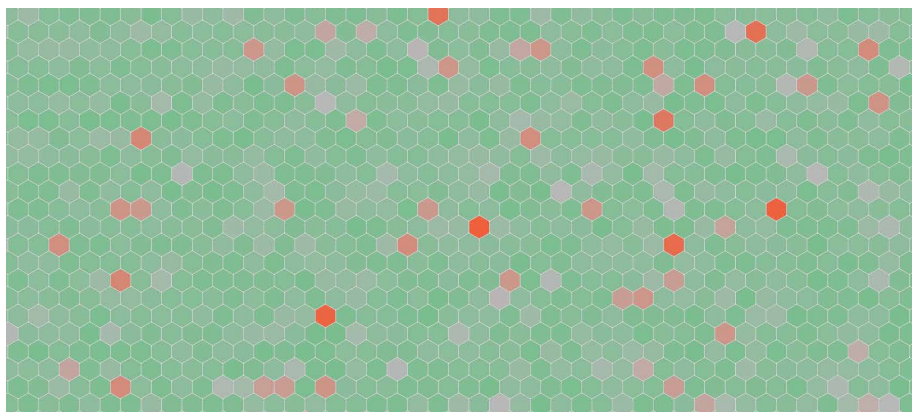


に、リソースのサイズ変更、より効率的なアーキテクチャへの移行、キャパシティの事前購入の 3 つの戦術を用いました。

リソースのサイズ変更

平均で CPU はわずか 1~2% の使用率で稼働していました。ワークロードで必要とされる以上の強力なインスタンスを選択しており、過剰なリソースを割り当

てている（「オーバースペック」）ことが分かりました。インスタンスのサイズを変更し、キャパシティを集約することで、大幅なコスト削減を実現しました。

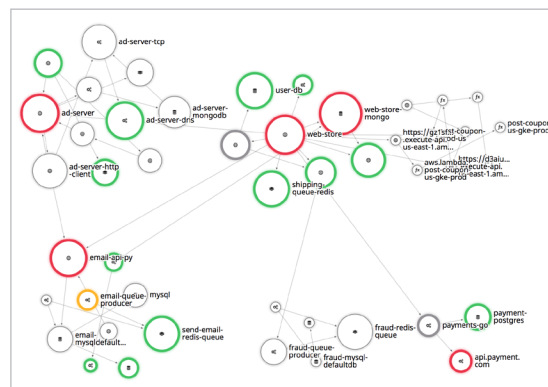
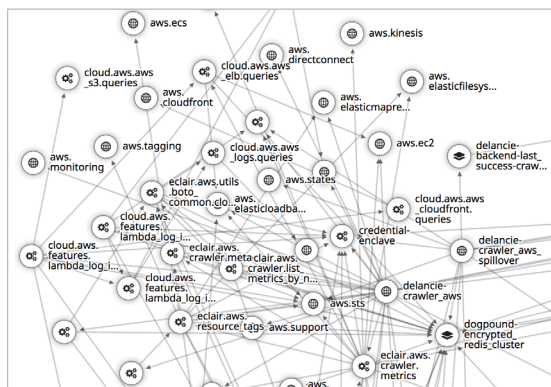


CPU 使用率別に色分けされたインスタンス

不要なリソースをオフにして、さらに効率的なアーキテクチャに移行する

静的なアーキテクチャダイアグラムは、通常、システムの実際の動作を正確に示すことはありません。堅牢なモニタリングソリューションによって、リアルタイムのアーキテクチャマップを取得でき、不要なツールやサービス、さらには非効率なアーキテクチャパターンも検出できました。

たとえば、個々のレコードを読み取ることを目的とするデータベースがホスティングされていたことも特定できました。コスト削減のために、Amazon S3 に切り替えました。S3 は、AWS に障害対策と運用要件をアウトソースするソリューションであり、自社でデータベースクラスタを運用するよりもはるかに安価です。



導入前と導入後：複雑で非効率で高価なアーキテクチャと、シンプルで効率的で安価なアーキテクチャを示すサービスマップ

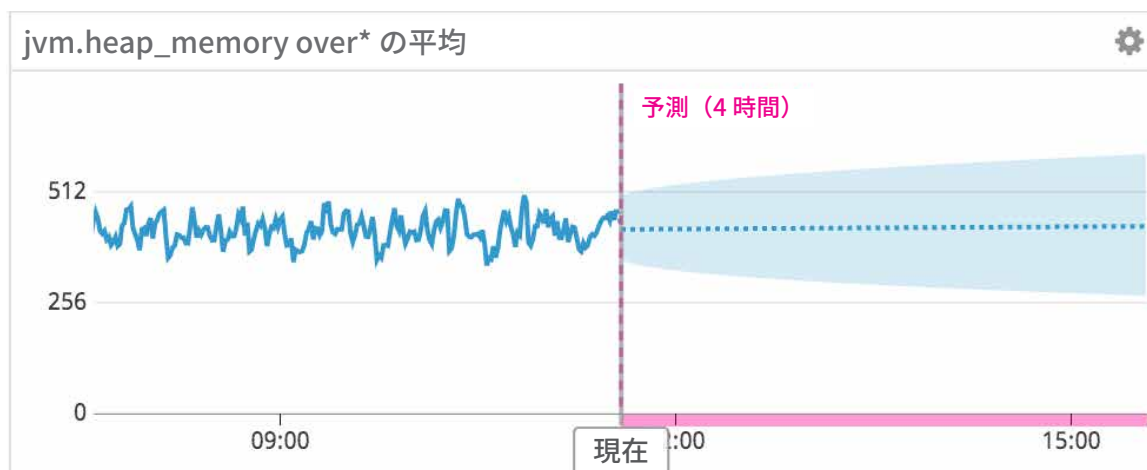
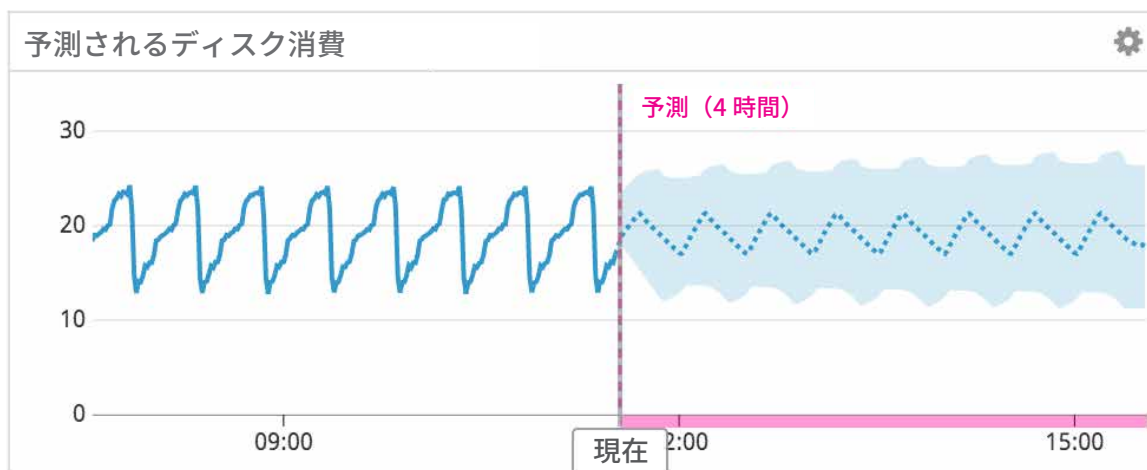
当社では、新しいツールの導入については非常に慎重に検討するようになっていました。直接的なコストと管理作業が増えて間接費も増大する恐れがあるため

す。モニタリングソリューションは、ツールやサービスの利用を合理化し、状況を正確に把握するのに役立ちます。

キャパシティの事前購入

堅牢なモニタリングソリューションがあれば、現在のニーズを把握し、将来のニーズを予測できます。各インスタンスの正確な利用状況と、全インスタンスを集計した情報を正確に把握できるようになりました。過去のデータをベースにして機械学習を活用し、需要予

測を参照できることは、非常に有用です。予測によって、インスタンスを先行予約して大幅なディスカウントを受けることが可能になるなど、経済的にも有利な選択が可能となります。

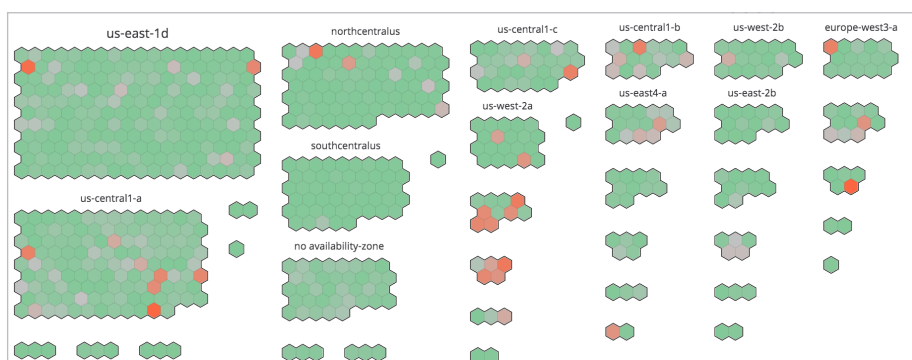


キャパシティプランニングに役立つ予測アルゴリズム

不要なテスト環境の排除

また、開発とテスト環境の数も削減できました。私たちのチームは、多くのテスト環境があり、異なる環境で新しいコードを稼働しても価値がないことに気がきました。さらに、多くの異なる環境を運用する中で、

(1) 本番環境以外のテストは本当のセーフティネットにはならないこと、そして (2) 複数のテスト環境があると、管理上およびコスト面の負担が生じるという 2 つの重要な教訓を学びました。



テスト環境が多すぎると、コストが増大し、効率性が低下します。また、そのテスト結果が良好であっても本番環境で生かされない恐れがあります。

本番環境外のテストは本当のセーフティネットにあらず

このような人工的なテスト環境では、本番環境でコードが実際にどのように実行されるかを十分に理解できない恐れがあります。この理由の 1 つには、実際のライブトラフィックを再現できないことです。回帰テストや他のテストを実行しても安心材料とはならず、本当のセーフティネットにはなりません。それに、これだけ多くの環境を保持すると、費用対効果も低くなり

ます。そのために、本番前のテストを重視することを止め、本番環境の監視とアラートへと移行しました。プロアクティブにアラートを発行し、異常を監視できるようになったことで、回帰テストを実施するよりも、安心して運用できるようになり、リスクの高い領域に対するテストを合理的に進めることが可能になりました。

複数の環境はチームの負担となる

当社の開発者は、複数の環境にデプロイする作業の複雑さに不満を感じていました。特に、手動による構成作業が多く求められる場合には、特に不満を募らせていました。各環境は技術的に決して同じではなく、これらの環境を適切に維持することも容易ではありません。環境の数が増えるほど、複雑化し、コストも増大します。問題は、インフラストラクチャにかかるコストだけではなく、チームは、いくつものテスト

環境にデプロイし、最終的には本番環境にデプロイしなければなりません。これにより、プロセスのコストが増大し、デリバリサイクルの時間が膨れ上がることになります。品質保証のレベルを維持しつつ、余分なステップを取り除くことで、チームの効率性を向上し、コストを削減できました。現在当社には、テスト環境と本番環境がそれぞれ 1 つあるだけです。

本番環境の監視

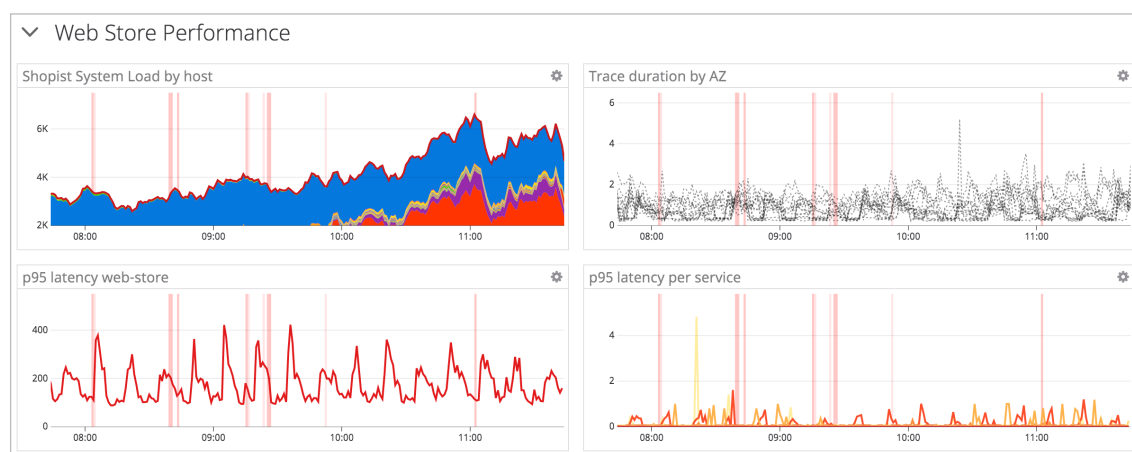
テスト環境と一部のテストツールを排除したため、コストは削減されましたが、これらのツールが担っていた機能のギャップを解消する必要がありました。導入したモニタリングソリューションの広範な機能によっ

て、これらのギャップを解消することができました。しかし、デリバリープロセスも変更する必要がありました。

少数の大規模なデプロイから、多数の小規模なデプロイへ移行する

当社は、成果物のバッチサイズを小さくすることに注力しました。変更を可能な限り小さくすることで、サイクルタイムを短縮し、発生する可能性のある問題を簡単に切り分けることができました。小規模な変更ではうまくいかない場合は、すぐにロールバックすることも、フォワードフィックス（修正しながら前進

する）を行うこともできます。さらに、堅牢な計測機能を利用できるため、アーキテクチャを詳細に把握でき、モノリシックアプリケーションを安全に分解し、依存関係が少なく、適切に稼働できる可能性が高い、よりシンプルなマイクロサービスパターンに移行する方法を特定できるようになります。

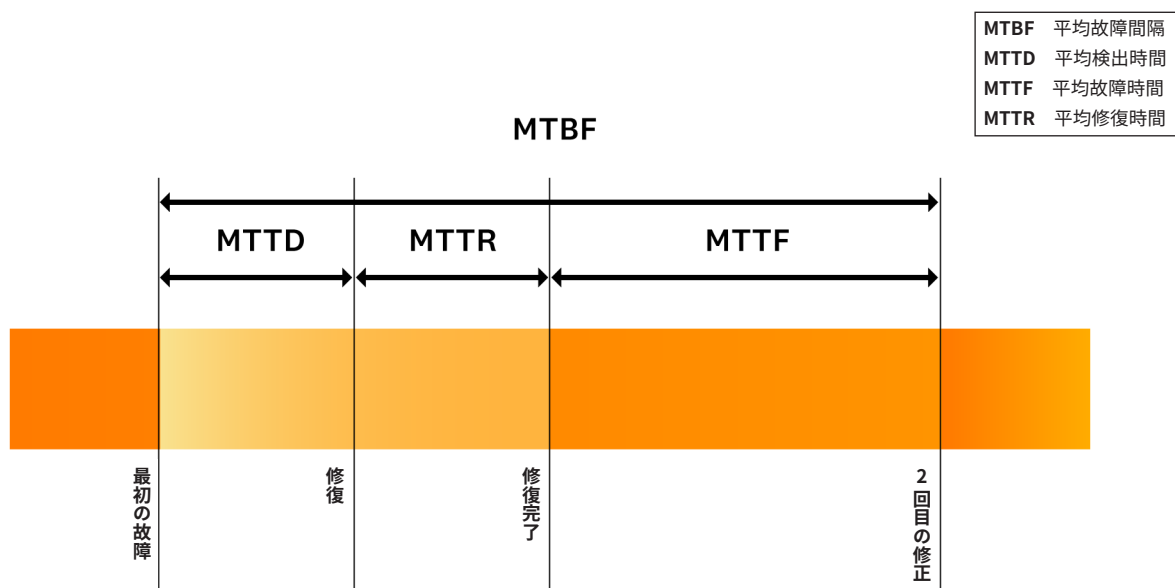


小規模な各デプロイ（ピンクの縦棒）の影響をすぐに把握できます

インシデントの早期解決

デプロイを小規模にすることで複雑さは解消されますが、問題を迅速に解決できるようにする必要もありました。これまでは、使用していたモニタリングツールは統合されておらず、特にプレッシャーのかかる状況で、真夜中や週末に問題を解決することが難しくなっていました。高度に統合されたモニタリングソリューションに投資することで、インシデントを解決するまでの時間が短縮され、インシデントによるチームへの

ストレスも軽減されました。MTTD（平均検出時間）と MTTR（平均修正時間）を厳密に計算しているわけではありませんが、発生したチケット数、P1、P2、P3 カテゴリのチケット数を追跡したところ（これらのカテゴリは、各インシデントの優先度を示しています）、想定されるよりも解決までに長い時間を要したチケット数（たとえば、P3 インシデントの場合は 60 日以内）が減少しました。

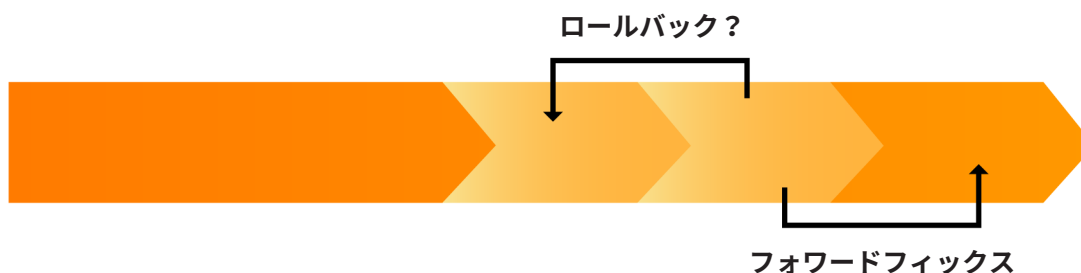


ロールバックではなくフォワードフィックス

最近では、ロールバックよりもフォワードフィックスを採用する傾向にあります。フォワードフィックスでは、問題を解決しながら前進させることができるため、デリバリサイクルの勢いを維持できます。

しかし、フォワードフィックスは、発生している問題を迅速に特定して修正する能力に自信がなければ、採

用できない手法です。堅牢なモニタリングソリューションがあれば、ログ、イベント、インフラストラクチャメトリクス、その他の監視データをシームレスに相関でき、問題の根本原因を迅速に特定して解決することができます。



パートナーとサードパーティの監視

TravelSupermarket のコアサービスは、ホリデーシーズンや長期休暇用のトラベルパッケージの検索と比較サービスです。当社の Web サイトで検索を実行すると、パートナー企業の情報が呼び出されます。トラフィックや時間帯にもよりますが、レスポンスが遅いパートナーが存在します。最も処理の遅いパートナーのレスポンスは、ユーザーの待ち時間に影響します。

そのため、パートナーのレスポンス時間は、当社のビジネスにとって非常に重要です。

新しいモニタリングソリューションの機能により、すべてのパートナーのレスポンス時間をグラフ化し、パートナーの状況を把握できるようになりました。このデータによって、パートナーがダウンしている場合、そのパートナーの情報を自動的に待機しないよ

うにでき、全体的な待ち時間を短縮できました。また、パートナーがダウンしている場合、アラートを発行できるようになりました。さらに、パートナーとプロアクティブに協力してレスポンス時間を短縮し、トラフィックの需要に対応できるようになりました。

パートナーの処理が遅くなったり、停止したりするとユーザーはすぐに他のサイトへ移ってしまうため、これらの効果は当社のビジネスに非常に大きな影響があり、収益を数パーセント押し上げることができました。

Slowest API Endpoints	
seconds	15.19 ProductsController#show
	3.26 ShoppingCartController#apply_coupon
	2.27 MerchantsController#index
	2.04 ProductsController#index
	0.99 ShoppingCartController#checkout

処理が遅い、あるいはパフォーマンスが低下しているパートナーは、ビジネスの足かせとなる恐れがあります

チーム構造、文化、プロセスの変化に対応する

このセクションでは、クラウドのコスト効率に直接関係していないと思われるチーム、文化、プロセスの変化について説明します。先に述べたように、実際には、文化、プロセス、マインドセットに影響される個々の小さな選択肢が、時間の経過とともに積み重なった結果として、クラウドのスプロール化が発生します。次の変更点は、これまで説明してきた利点のすべてを可能にするため、コスト効率を向上するために不可欠なものです。

これまでは、中央集権的なチームがあり、フロントエンド、ミドルウェア、バックエンドのような個別のテクノロジーに特化した、労働集約型で分業体制で作業を行っていました。現在では、主要な製品分野別にチームが編成されており、スペシャリストではなくテクノロジージェネラリストを中心にチームが構成されています。このシフトを可能にする上で、本番環境のパフォーマンスモニタリングが重要な役割を果たしました。

テクノロジー指向から製品指向へ

各チームが、それぞれ独自の製品分野を専門に管理するようになりました。これらのチームは、目的意識と使命感を持って業務にあたっており、カスタマージャーニーに対して非常に情熱的に取り組むことが可能になりました。たとえば、トラフィックを専門とするチームは、ユーザーをサイトに誘導したり、セー

ルスファネルに顧客を誘導したりする対策を実施し、ページのスピードや UX の向上に情熱を注いでいます。チームの誰もが、ビジネスの価値を理解し、その仕事ができるように全体のビジネス目標を達成するのに役立つのかを理解していない限り、業務に着手することはありません。

サービスについての全体的な責任

製品指向の取り組みを機能させるためには、各チームは担当するサービスに対して全体的な責任を持たなければなりません。コードを書いたら終わりではありません。独自のモニタリングとアラートを設定し、問題のトラブルシューティング、メンテナンス、サービスの強化を行わなければなりません。堅牢で使いやすい

モニタリングソリューションによって、当社のチームは自律的にこのような課題に取り組むことが可能になりました。リリースが承認されれば、チームのほぼ全員がデプロイでき、モニタリングソリューションを使用して、本番環境に何も負の影響がないことをすぐに確認できます。

スペシャリストからジェネラリストへ

製品指向をエンドツーエンドで実現するためには、組織がサイロ化されておらず、チームがスペシャリストではなくジェネラリストで主に構成される必要があります。ジェネラリストから構成されるチームが高い実績を発揮するためには、簡明性と堅牢なモニタリングの 2 つの要素が重要です。スタックがシンプルであれば、チームが技術的な細部で行き詰まることはありません。また、スタックを十分に把握していない場合でも、モニタリングによってスタックを把握して、自信を持って（そして何も壊すことなく）変更をデプロイできます。

ジェネラリストから構成されるチームが高い実績を発揮するためには、簡明性と堅牢なモニタリングの 2 つの要素が重要です。

特定のテクノロジー分野ではスペシャリストが今も活躍していますが、今では、全員がある程度ジェネラリストとして働いています。このような変革の中で、何人かのスタッフは離れていきましたが、当社に残った人材は、果敢に挑戦し、多様性を許容し、顧客を中心にしてお仕事を進めています。



「I 型」

1 つの特定分野の専門家



ジェネラリスト

さまざまな分野に対応できるが、
いずれの分野の専門家ではない



「T 型」

さまざまな分野に対応でき、
これらの 1 分野における専門家

「I 型」から「T 型」へと変わった当社のチーム

まとめ

クラウドのパフォーマンスモニタリングにより、チーム、プロセス、テクノロジーを、よりシンプルで、無駄がなく、効率的になるように調整できるようになりました。デリバリモデルを進化させたことで、デリバリの高速化、クラウドインフラストラクチャコストの

50% 削減、事業部門の収益の増加など、企業全体にとって大きな成果をもたらしました。取り組むべきことはまだ多くありますが、これらの利点は、モニタリングソリューションへの投資が十分に価値のあるものであったことを示しています。

