

はじめに

企業のクラウド活用は年々加速する傾向にありビジネス展開のスピードも上がってきております。それはクラウドで稼働するアプリケーションに関しても、今まで以上の速い機能強化や機能改善が必要になってくることにも繋がってきます。

クラウドの特徴のひとつは、API でインフラを操作できることです。これまで手作業で行っていたこと、ハードウェアの増設などの投資が無いと取り組めなかったことが、簡単に短時間にできるようになりました。その結果、効率よくアプリケーション開発を実施する為には、開発者もアプリケーション開発に必要なインフラレイヤーに関しても同時に考える必要があるのです。開発者がインフラに携わる機会が増えた結果、開発者がインフラ運用や監視のスキルを身に着けることも必要になったのです。

システムは開発して終わりではなく、ビジネスの変化に対応しながらメンテナンスやアップデートを行い、運用し続ける必要があります。つまり、システムを安定して稼働させるための「監視」の仕組みが必要です。

本書では、クラウド活用が進む中で監視がなぜ重要なのかを説明し、さらにクラウド移行の際に DevOps モデルへの転換を図りつつ、安定したシステム運用を継続するためのクラウド監視を始める方法を、はてなでの事例を交えて紹介します。

02 はじめに

03 監視と DevOps

[DevOps モデルとはてなの取り組み](#)

04 クラウドで変わる監視の考え方

[付録：監視はスキル](#)

05 AWS クラウド監視をはじめ

[AWS クラウドの状態を知る](#)

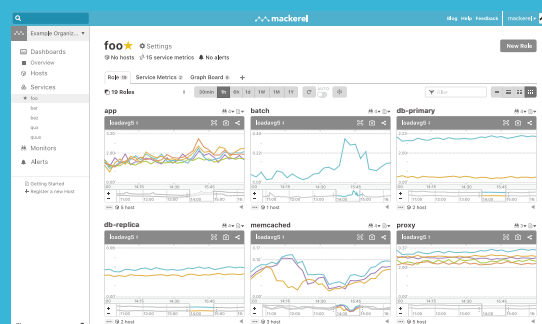
06 監視ルールを設定する

07 アラートを通知する

08 ダッシュボードでシステムの状態を可視化する

09 まとめ：AWS クラウド監視をはじめましょう

10 サーバー監視サービス「Mackerel」とは



01：サービスの様子を一望できる画面

監視とDevOps

監視とは、監視する対象であるシステムが正しく動いていることを継続的に見守る仕組みのことです。Web サービスであれば 24 時間ユーザーに使われますし、企業内システムであれば営業時間中は常に正常に動いていることが求められます。システムが正常に動くために必要なことを一番よく知っているのは、そのシステムの開発者です。つまり、システムが正しく動いていることを担保するためには、開発者も運用に携わりシステムを継続的に見守る仕組みを作ることが必要なのです。そのためには、開発と運用の垣根を超えて継続的に改善していく DevOps モデルの導入が重要になります。

DevOps モデルとはてなの取り組み

DevOps とは、単なる開発と運用の職能の壁を取り払う取り組みではなく、開発と運用が緊密に連携してアプリケーション品質を継続的に向上させていくための組織体制、プラクティス、ツールと考えます。例えば、Dev（開発）と Ops（運用）が分離した組織では責任範囲が分離され、お互いが情報共有や連携をすることなく個別に業務を進めることになります。開発と運用が分離されると、開発チームは新規技術をどんどん取り入れたい一方で、運用チームはシステムの安定化を目指して新技術の導入に慎重になるという（矛盾）が生まれ、サービスやアプリケーションの継続的な品質向上を妨げます。

開発と運用が緊密に連携すれば、クラウド技術を積極的に活用して開発速度を上げることができ、結果的にビジネス展開の速度を高めることができます。開発と運用を緊密に連携させ開発プロセスを高速化する、これが DevOps モデルの大きなメリットです。

本書を執筆する株式会社はてな（以下、はてな）は 2001 年に創業して以来、オンプレミス環境でサービスを提供してきました。インフラ運用チームがシステム基盤を構築し運用することで、多くのコミュニティサービスを効率的に提供する仕組みを作っていました。その後クラウド化が進み、サービスの幅が広がるにつれて、技術選択の幅も広がっていきました。しかし同時に、クラウド時代に徐々に合わなくなった共通基盤が選択できる技術の幅を狭めていき、多くの技術領域に対応する必要がある運用チームの負荷が高まっていきました。

そこで 2018 年から DevOps モデルを採用し、開発チームにインフラ運用まで含めてシステム全体にオー

ナーシップを持つように組織体制を変更しました。開発チームがインフラ領域まで責任を持つことで、システムに適した技術選択の自由度を増やし、開発速度を上げることを目指しました。この実現のため、開発チームへの運用スキル移転と同時に、開発チームに SRE 職のメンバーを配置し、チームで運用できる体制を強化していきました。

2020 年現在、はてなでは開発チーム全員が運用について考え、障害対応や定期的な振り返りなど、運用改善にもチーム全体で取り組むことに成功、その中核を担うツールとして開発されたのが監視サービスの Mackerel なのです。

クラウドで変わる監視の考え方

オンプレミス環境では、1 台 1 台に搭載されたサーバーリソースがリッチ（集約型）で、1 台のサーバーに複数の機能を持たせることもありましたが、サーバー自体を資産として所有しているため、物理的な障害が発生しても修復して使い続ける必要がありました。通常これらの運用業務は、専任の運用者によって実施されてきました。

一方で、クラウド環境では、1 台 1 台のリソースを用途や負荷に応じて柔軟に変更できるため、サーバーごとに機能を分割でき、システム負荷を分散できます。また物理的な障害に対応する必要がなく、必要に応じてすばやく作り直すことが可能になります。

クラウド監視はサーバー 1 台 1 台を丁寧に監視・管理して利用するオンプレミスから、サーバー（そしてマネージドサービス）を一つの機能として考えて監視する必要があります。クラウドではオートスケーリングや API の提供により、動的にシステム環境が変動します。つまり、物理的なサーバー単位やマネージドサービスの単位ではなく、個々の機能や役割ごとに管理することが、クラウド環境を監視するポイントとなります。

クラウド環境では、リソースを必要な分だけ調達して利用することができるので、システムを小さく作り、運用しながら育てていくことが可能です。クラウド監視も同様に、まず小さくはじめて、育てていくことが大事なのです。

付録：監視はスキル

Mike Julian 著、松浦 隼人 訳『入門 監視』（O'Reilly Japan, 2019）では、「監視とは役割でなくスキルであり、チーム内の全員がある程度のレベルに至っておくべきです。（中略）監視は他の仕組みから孤立した仕組みではなく、サービスのパフォーマンスのために重要なのです。」と記載されています。クラウド運用を開始する際の組織体制を検討するために重要な考え方です。

Mackerel は、豊富な API や使いやすい UI により開発者にとっても扱いやすい監視サービスです。これまで監視に携わっていなかった方でも、Mackerel を通じて監視の第一歩を踏み出し、育てていくことで監視のスキルを身に付けられます。

—— Mike Julian 著、松浦 隼人 訳『入門 監視』（O'Reilly Japan, 2019）

AWSクラウド監視をはじめめる

Mackerel で AWS クラウド監視をはじめめるための最初のステップは 4 つです。

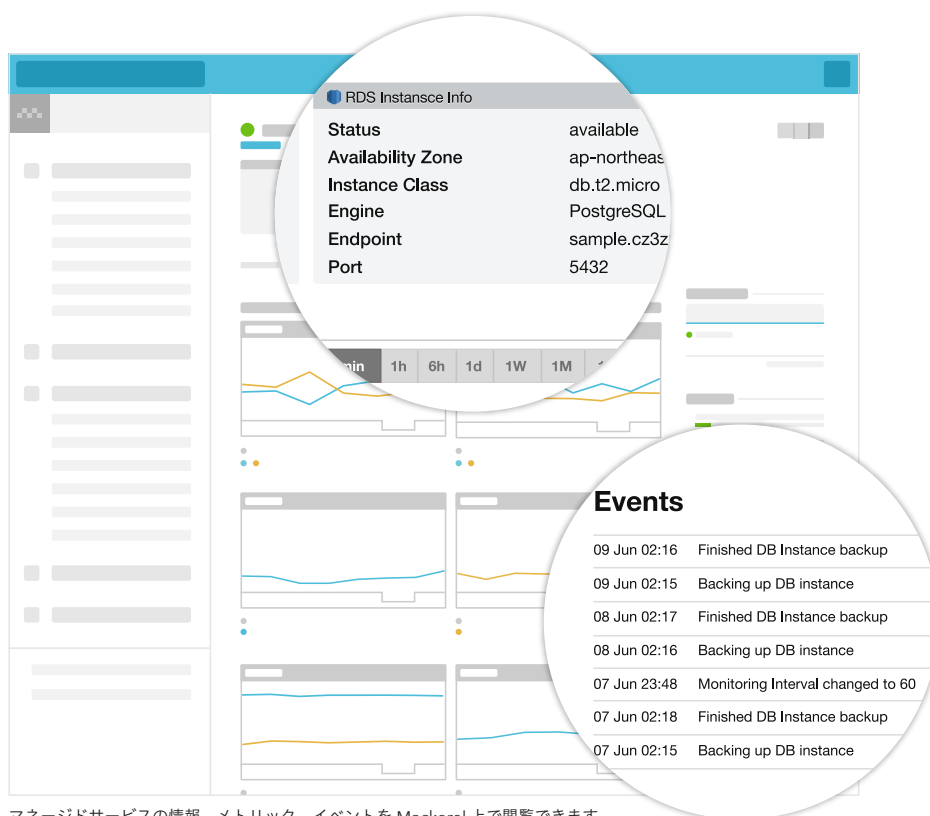
1. AWS クラウドの状態を知る
2. 監視ルールを設定する
3. アラートを通知する
4. ダッシュボードでシステムの状態を可視化する

1. AWS クラウドの状態を知る

AWSクラウド監視をはじめめるには、まずAWSクラウドの状態を知る必要があります。Amazon Elastic Compute Cloud (Amazon EC2) を利用してサーバーを構築する場合、Mackerelはオンプレミスの場合と同様にサーバーにエージェントをインストールしてリソース情報を取得します。クラウド環境では、データベースやロードバランサなど、マネージドサービスを多く利用します。例えば、よく利用される Amazon Relational Database Service (Amazon RDS) や Elastic Load

Balancingなどの場合、MackerelではAWSインテグレーション※1を設定すると連携サービスの情報を自動で取得し、サーバーと同じ「ホスト」という単位で扱うことができます。Amazon CloudWatchのメトリックを収集することでマネージドサービスを含めたシステムの状態をMackerel上で知ることができます。

こうして、まず監視したいシステムの状態を知ることができれば、監視を始めるための準備は完了です。



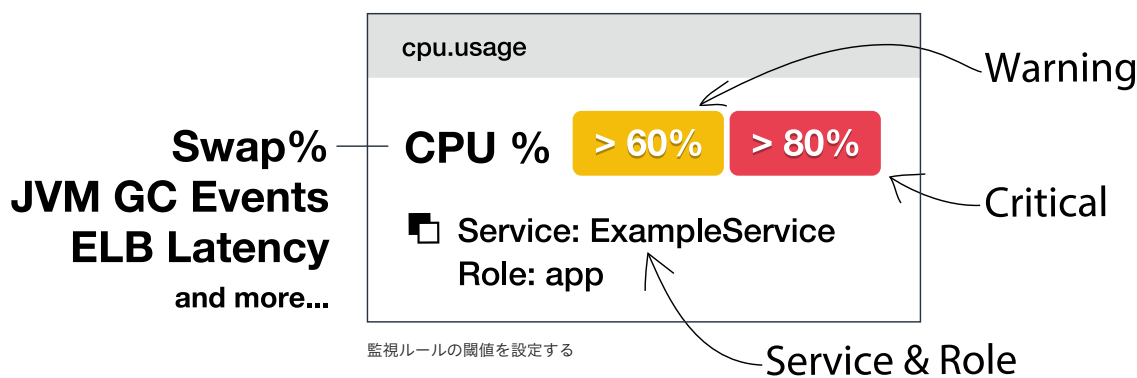
マネージドサービスの情報、メトリック、イベントを Mackerel 上で閲覧できます

2. 監視ルールを設定する

システムの状態を知ることができるようになったら、次は実際に監視をしてみましょう。Mackerel にはいろいろメトリックが設定されていて、最初はどのメトリックに監視を設定するといいか難しいかもしれません。まずは明らかに障害だろうと思うところを思い浮かべ、設定してみましょう。サーバーの CPU 使用率が 90% 以上を障害と定義するケースもありますし、ロードバランサーでエラー応答が出ていることを検知したいケースもあるでしょう。考えられる障害シナリオを思い浮かべ、リソース情報に対して閾値を設定して異常を検出します。

最初から閾値を難しく考える必要はありません。まず仮の値を設定してみて、負荷試験や実際の運用の中で閾値を調整し、監視設定を育てていくことをおすすめします。Mackerel では見やすいグラフとわかりやすい UI で簡単に設定変更することができます。

運用中の突発的な負荷上昇など、「いつもと違う」状況を見つけたいというケースも考えられます。Mackerel は手動での閾値設定のほか、機械学習を用いた異常検知によりこういった異常を見つける機能も提供しています。



Mackerel はこのほかにも「URL 外形監視」の機能を提供しています。特定の URL へのレスポンスタイムやステータスコード、さらに証明書の有効期限やレスポ

ンスに含まれる内容などを監視し、異常を検出することができます。

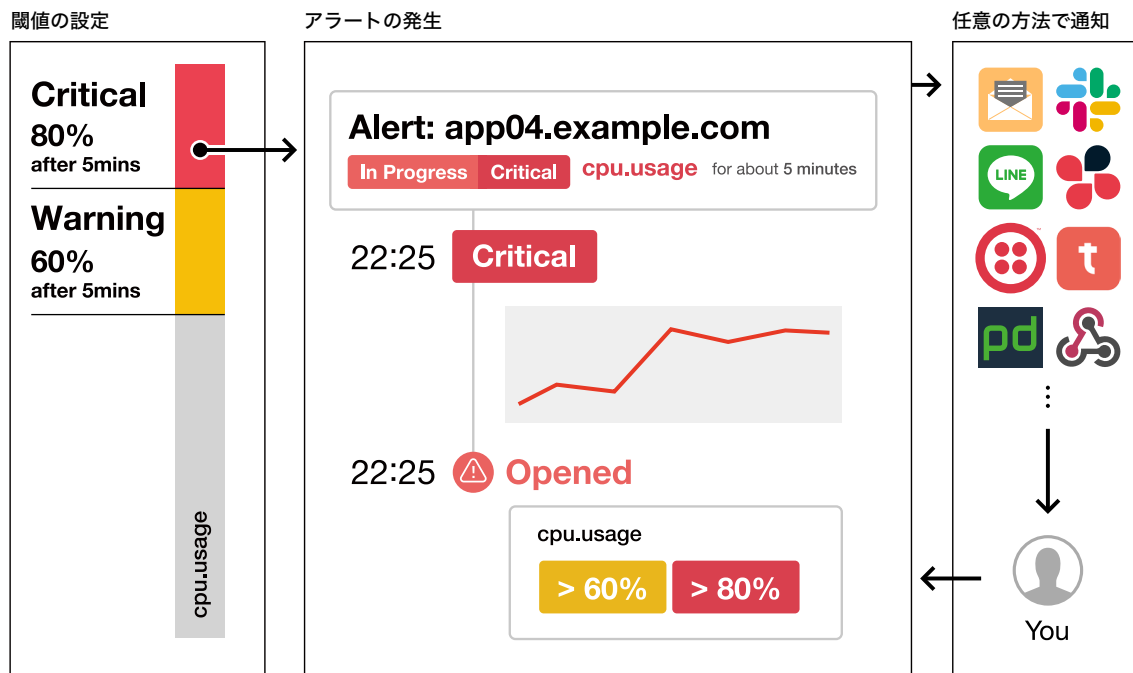


URL 外形監視でレスポンスタイムやステータスコードを監視する

3. アラートを通知する

監視ルールを設定してシステムの異常を検出できるようになったら、次はそのアラートに気付けるように通知設定をしましょう。Mackerel はメールへの通知はもちろん、Slack などのコミュニケーションツールへの通知

機能を標準で備えています。他にも多くの通知先を選ぶことができ、例えば Twilio と連携しての架電通知や、PagerDuty を使ったインシデント管理など、通知したい経路に合わせた設定が可能です。



アラートはコミュニケーションツールに簡単に通知できます

4. ダッシュボードでシステムの状態を可視化する

ここまでで、クラウド上にあるシステムの状態を把握して異常が起きた際にはアラートが通知され、気づくことができる仕組みが整いました。日々の運用では、システム全体の長期的な傾向見る、障害箇所の特定のためにクリティカルな指標を並べるなど、システム全体の状態を把握できるようにしておくことが便利です。Mackerel 上に保存されている任意のグラフ情報や、グラフから抽出した数値情報、サービスのアラート状況、

任意のテキストなどをダッシュボード上に自由に配置できます。

ダッシュボードは、「常時確認する重要項目」「定期ミーティング」「障害発生時に注意したいグラフ集」など用途に応じて作成すると役立ちます。キャンペーンサイトへのアクセス数などビジネス的に重要な指標も配置することで、ビジネスメンバーも含めて確認しやすいダッシュボードを作成することも可能です。



50 インチの大型モニターに Mackerel の監視画面を出して、その隣に Google Analytics の情報も合わせて出して「どのページにアクセスがきているぞ」とわかるようにしています。そうすると、アプリケーションエンジニアが「いつもとパターンが違うぞ」と言うてくることがあります。

大型モニターがアプリケーションエンジニアからも見える場所に置いてあります。それで、アプリケーションエンジニアの方から「グラフの傾向がいつもと違うぞ」と言うてきて、お互い調べ始めたりします。

いろいろな人が通る場所に置いてあるので、そこからコミュニケーションが生まれたりしています。

WILLER 株式会社 Marketing & Technology Business Unit. システム デベロップメント Dept. 田崎様、松井様、廣田様

業務システムをクラウドで内製、Mackerel ダッシュボードと大型モニターをコミュニケーションツールに ~ WILLER ~
Mackerel ブログ <https://mackerel.io/ja/blog/entry/customers/willer>

ダッシュボードを作成してシステム全体の状態を把握する

Mackerel では、「サービス」「ロール」という単位でシステムを管理するモデルを提供しており、「サービス」全体を見ることでシステム全体の状況を把握できるダッシュボードが自動で作成されます。さらに、カスタムダッシュボード機能も提供しており、柔軟に、簡単な操作で

更新できます。注視したいグラフやシステム構成はシステムを開発・運用していく中で変化していきます。一度作ったダッシュボードを使い続けるだけでなく、定期的にダッシュボードを更新して、クラウドの変更に合わせて監視を育てていきましょう。

まとめ

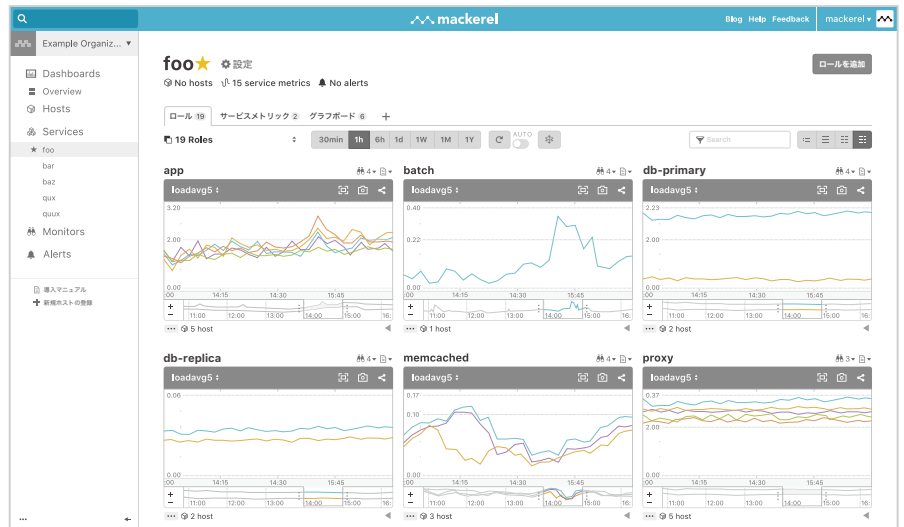
AWSクラウド監視をはじめましょう

クラウドでのアプリケーション開発が増加したことにより、開発者がインフラを含めたシステム全体を操ることになり、開発者自身が監視を行う必要が増大しました。クラウドは必要なサイズから小さく始めることができるのと同様に、監視も小さく始め、運用の中で育てていくことが重要です。Mackerel はクラウド監視をはじめするための第一歩を提供しています。Mackerel を利用して、開発者と運用者で一体となって監視を始め、協力して育てていくことで DevOps 文化を醸成し、安定したシステム運用を実現しましょう。

Mackerel でクラウド監視をはじめてみようと思った方は、すべての機能を無料で試せるトライアルプラン（<https://mackerel.io/signup>）をお試してください。Mackerel を利用して、クラウド監視をはじめましょう。

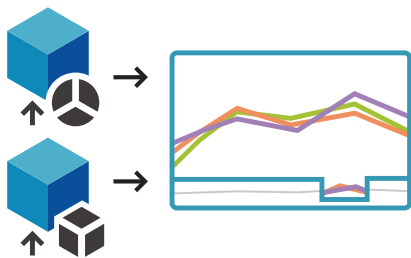
サーバー監視サービス「Mackerel」とは

Mackerel は、株式会社はてなが提供する日本製のサーバー監視サービスです。登録ユーザー数 876 万人、月間ユニークブラウザ数 1.96 億人を誇るはてなのサービス（はてなブログやはてなブックマークなど）も Mackerel によって支えられています。自社のサービス運用基盤を Mackerel で運用し、そのノウハウを詰め込むことで、Mackerel はクラウド監視に必要な機能を提供し続けています。



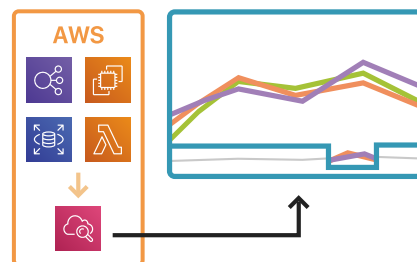
Mackerelの強み

Mackerelはオンプレミス・クラウド環境を問わず、はてなが提供するモデルに従って簡単に始めることができ、使いやすさや充実したサポートが特徴のサービスです。



導入が簡単ですぐ監視をはじめられる

スタートガイドに従えば十数分あれば導入できます。取得したメトリックは自動的にグラフで可視化され、クラウドの状態をひと目で確認できます。



クラウドサービスへの対応

AWS のリソース情報は、AWS インテグレーションを使って自動で取得し、サーバーと同じように扱うことができます。システムの基盤となる主要なコンポーネントに対応しているので、追加開発なしでクラウド監視をはじめられます。



日本語テクニカルサポートの提供

Mackerel は日本製のサーバー監視サービスです。専門のエンジニアがテクニカルサポートをしており、きめ細やかなサポートを提供します。また日本語ドキュメントや FAQ も豊富に提供しています。



導入支援プログラムの提供

Mackerel はエンジニアによる導入支援プログラムを提供しています。Mackerel をご導入いただいたお客様向けに、導入および基本機能の活用を支援いたします。

- これまで使っていた監視ツールに慣れており、Mackerel の使い方が腹落ちしていない
 - 実装したい内容を Mackerel の設定にどう落とし込めばよいのかわからない
 - 他のチームメンバーに基本的な Mackerel の使い方・運用のコツをインプットしたい
- というお客様の課題をエンジニアが解決いたします。