リソースのモニタリング

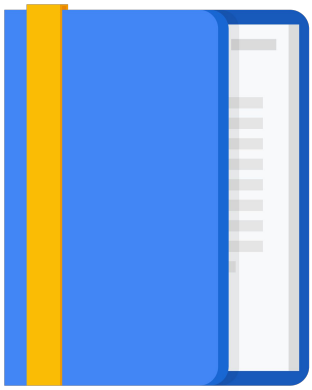
Stackdriver は Google Cloud の
オペレーション スイートになりました

このモジュールでは、Google Cloud のリソース モニタリング オプションの概要について説明します。

このモジュールで取り上げる機能は、アプリケーション向けにモニタリング、ロギング、診断機能を提供するサービスである、Google Cloud のオペレーション スイートに依存します。

アジェンダ

- Google Cloud のオペレーション スイート
- モニタリング
- ラボ
- ロギング
- エラーレポート
- トレース
- デバッグ
- ラボ



このモジュールでは、Cloud Monitoring、Cloud Logging、Error Reporting、Cloud Trace、Cloud デバッガの各サービスについて見ていきます。こうしたサービスを、このモジュールの 2 つのラボで適用しながらご紹介します。

まずは、Google Cloud のオペレーション スイートとその機能の概要を説明します。

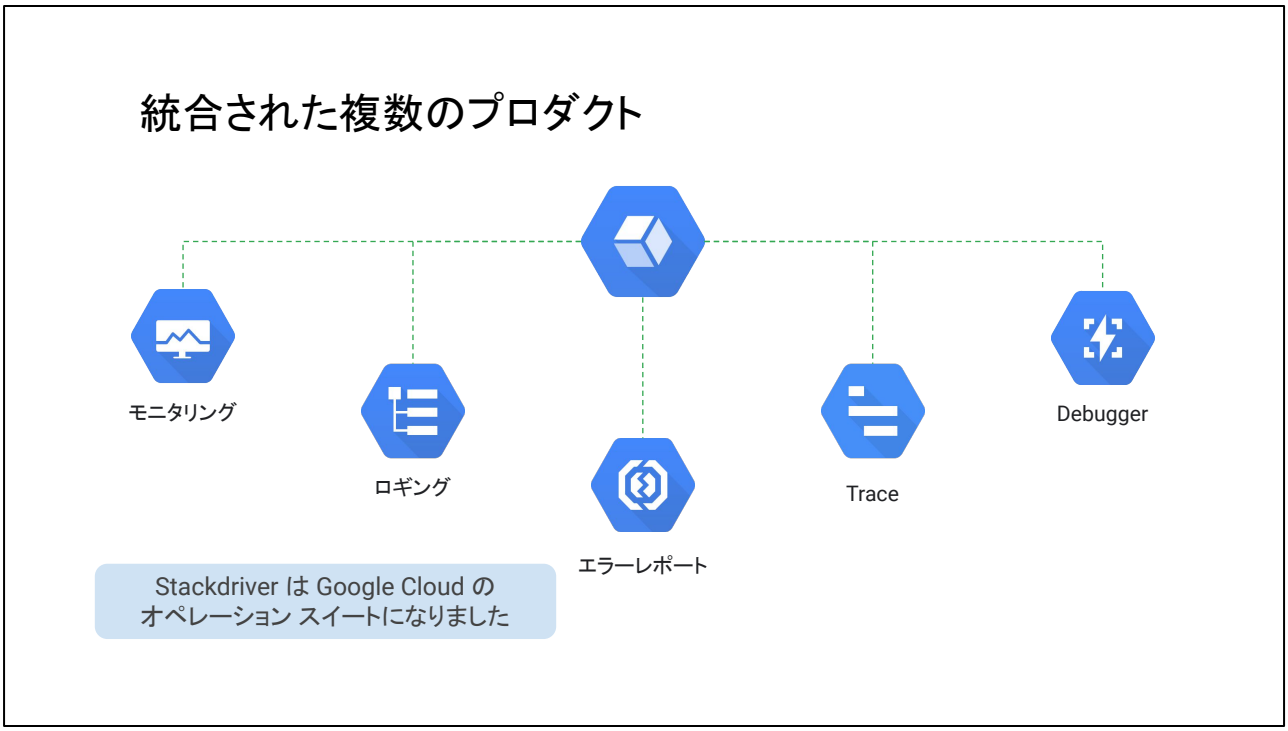
Google Cloud のオペレーション スイートの概要

- 統合されたモニタリング、ロギング、診断
- プラットフォームの枠を越えた管理
  - Google Cloud と AWS
  - Google Cloud のスマートなデフォルト設定による動的検出
  - オープンソースのエージェントと統合
- 強力なデータ / 分析ツールへのアクセス
- サードパーティ ソフトウェアでのコラボレーション

Google Cloud のオペレーション スイート
（旧 Stackdriver）

Google Cloud のオペレーション スイートは、Google Cloud やアマゾン ウェブ サービス（AWS）と緊密に統合されているため、クラウド リソースやアプリケーション サービスを動的に検出します。スマートなデフォルト設定により、クラウド プラットフォームの主要な情報を数分で可視化できます。

これにより、強力なデータ / 分析ツールにアクセスし、さまざまなサードパーティ ソフトウェア プロバイダとのコラボレーションが可能になります。



先ほどご説明したとおり、Google Cloud のオペレーション スイートには、モニタリング、ロギング、エラーレポート、フォールト トレース、デバッグなどのサービスがあります。お支払いはご利用いただいた分だけの従量制で、使用量に対する無料の割り当ても用意されているため、前払い費用や利用の確約なしで利用を開始できます。料金の詳細については、こちらの動画のリンク セクションをご覧ください。
[https://cloud.google.com/stackdriver/pricing]

現在このようなサービスは、他のほとんどの環境で、まったく異なるパッケージ、または緩やかに統合されたソフトウェア コレクションによって処理されています。こうした機能が、一つに統合された包括的なサービスで連携しているのをご覧になれば、それが信頼性、安定性、保守性に優れたアプリケーションを作るうえでいかに重要であるかが、おわかりいただけると思います。

また、このスライドに示すように、Google Cloud のオペレーション スイートは、テクノロジー パートナーの拡大し続ける充実したエコシステムもサポートしています。これは、Google Cloud のお客様が利用できる IT オペレーション、セキュリティ、コンプライアンス機能の拡大に役立ちます。統合について詳しくは、動画 [https://cloud.google.com/stackdriver/partners] のリンク セクションをご覧ください。

アジェンダ

Google Cloud のオペレーション スイート

モニタリング

ラボ

ロギング

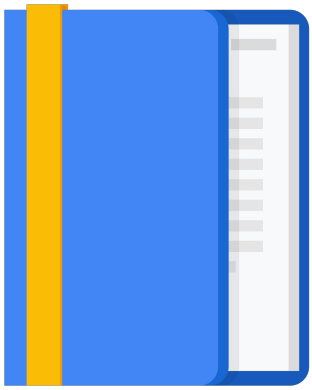
エラーレポート

トレース

デバッグ

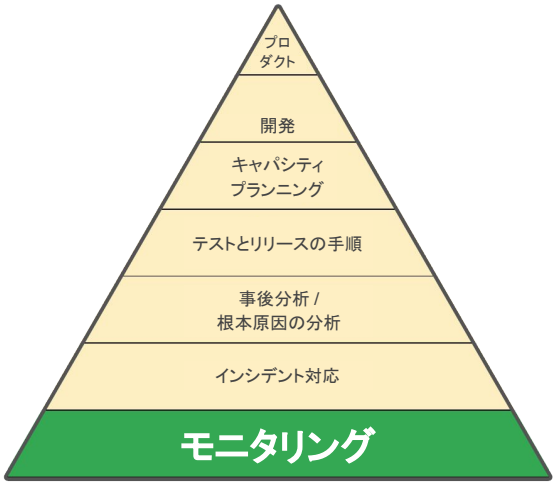
ラボ

Stackdriver Monitoring は Cloud Monitoring になりました



Google Cloud のオペレーション スイートの概要をご案内しました。次は、Cloud Monitoring を見てみましょう。

サイト信頼性エンジニアリング



Monitoring はサイト信頼性エンジニアリング（ SRE）の基礎となるため、Google にとって重要です。

SRE は、ソフトウェア エンジニアリングの側面を、極めてスケーラブルで信頼性の高いソフトウェア システムの構築を目的としたオペレーションに応用する規範です。この規範によって、Google は、世界最大級のソフトウェア システムを構築、デプロイ、モニタリング、維持できます。

SRE の詳細については、Google の SRE チームメンバーが執筆した書籍（追加料金なし）をご覧になることをおすすめします。こちらの動画の リンク セクションからご確認ください。[https://landing.google.com/sre/book.html]

モニタリング

- 動的な構成とインテリジェントなデフォルト構成
- プラットフォーム、システム、アプリケーションの指標
  - データの取り込み: 指標、イベント、メタデータ
  - ダッシュボード、チャート、アラートで分析情報を生成
- 稼働時間 / ヘルスチェック
- ダッシュボード
- アラート

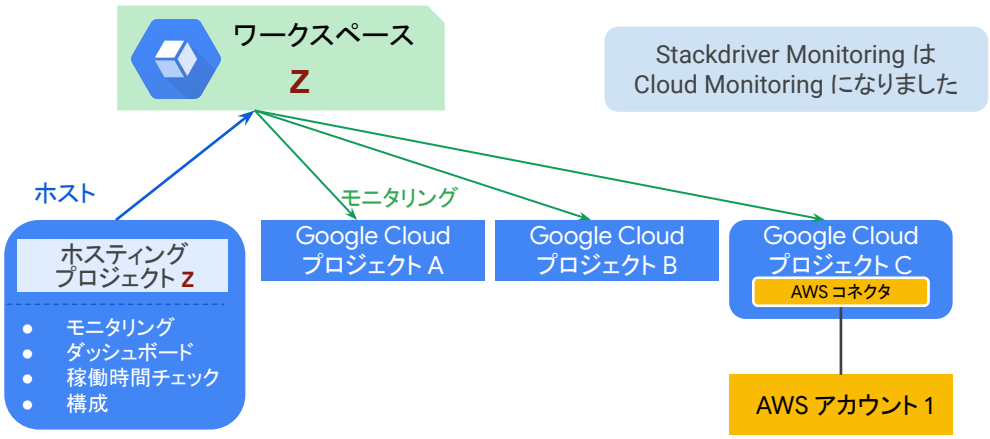
Cloud Monitoring
（旧 Stackdriver Monitoring）

Cloud Monitoring は、リソースのデプロイ後にモニタリングを動的に構成します。また、インテリジェントなデフォルト構成になっており、基本的なモニタリング活動を行うためのグラフを簡単に作成できます。

これにより指標、イベント、メタデータなどのデータを取り込み、プラットフォーム、システム、アプリケーションの指標をモニタリングできます。その後、ダッシュボード、グラフ、アラートを使用して、このデータから分析情報を生成できます。

たとえば、稼働時間とヘルスチェックを構成、測定し、メールでアラートを送信できます。

ワークスペースはモニタリング / 構成情報が保持されるルート エンティティ



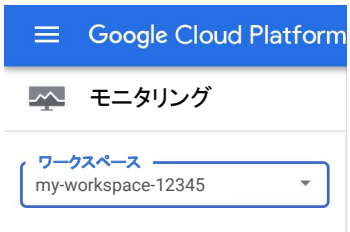
ワークスペースは、Cloud Monitoring にモニタリング / 構成情報が保持されるルート エンティティです。各ワークスペースには、1 つ以上の Google Cloud プロジェクトや任意の数の AWS アカウントなど、1～100 個のモニタリング対象プロジェクトを作成できます。ワークスペースの数に制限はありませんが、Google Cloud プロジェクトと AWS アカウントを、複数のワークスペースでモニタリングすることはできません。

ワークスペースには、モニタリング対象プロジェクトで使用するカスタム ダッシュボード、アラート ポリシー、稼働時間チェック、通知チャンネル、グループ定義が含まれています。ワークスペースはモニタリング対象プロジェクトから指標データにアクセスできますが、指標データとログエントリは個々のプロジェクトに残ります。

ワークスペース内で最初にモニタリングされる Google Cloud プロジェクトは、ホスティング プロジェクトと呼ばれます。ホスティング プロジェクトは、ワークスペースを作成する際に指定する必要があります。そのプロジェクトの名前がワークスペースの名前になります。AWS アカウントにアクセスするには、Google Cloud で、AWS コネクタを保持するようにプロジェクトを構成する必要があります。

ワークスペース:「単一の管理画面」

- モニタリングのニーズを事前に把握する
- データとコントロールを分離するために別々のワークスペース
  を使用することを検討する



Stackdriver は Google Cloud の
オペレーション スイートになりました

ワークスペースでは、すべての Google Cloud プロジェクトを 1 か所でモニタリングできます。つまり、これは「単一の管理画面」のようなもので、この画面から、複数の Google Cloud プロジェクトや AWS アカウントからリソースを見ることができます。そのワークスペースにアクセスできるすべての Google Cloud のオペレーション スイート ユーザーが、デフォルトで、すべてのデータにアクセスできます。

つまり、あるプロジェクトで 1 人の個人に割り当てられているロールは、そのワークスペースによってモニタリングされているすべてのプロジェクトに等しく適用されます。

プロジェクトごとに異なるロールを付与し、データの可視性をコントロールするには、そのプロジェクトのモニタリングを、別々のワークスペースに配置することを検討します。

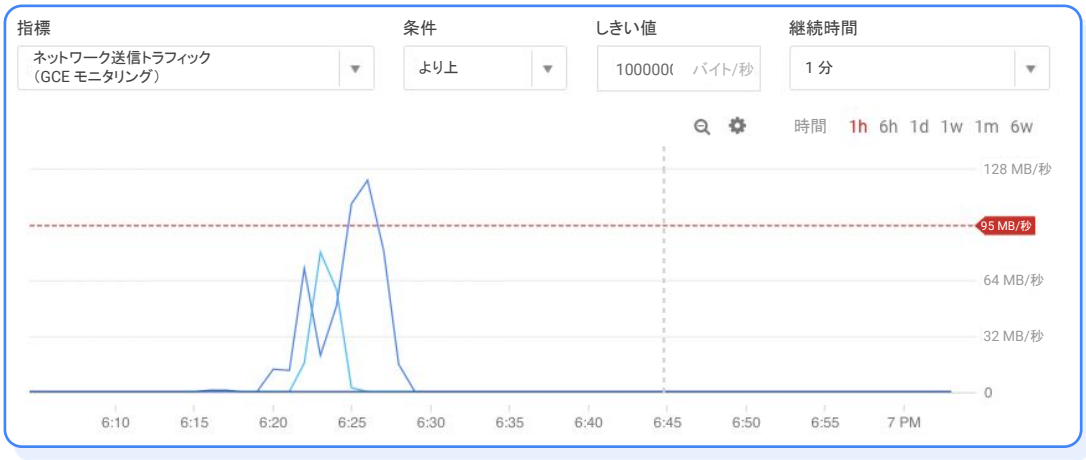
ダッシュボードで使用率やネットワークトラフィックを可視化

Cloud Monitoring を使用すると、モニタリングする指標のグラフを含むカスタム　ダッシュボードを作成できます。たとえば、インスタンスの　CPU 使用率、そのインスタンスが送受信したパケットまたはバイト数、そのインスタンスのファイアウォールによってドロップされたパケットまたはバイト数を示すグラフを作成できます。

つまり、このスライドに示すように、グラフによって、 VM インスタンスの使用率とネットワーク トラフィックが可視化されます。これらのグラフは、ノイズを除去するフィルタ、時系列の数を減らすグループ、複数の時系列をまとめる集計機能を使ってカスタマイズできます。

サポートされている指標の詳細な一覧については、こちらの動画にリンクされているドキュメントをご覧ください。[https://cloud.google.com/monitoring/api/metrics_gcp]

アラート ポリシーにより特定の状況を通知可能



さて、グラフは非常に便利ですが、そのグラフを誰かが見ている間しか分析情報が得られません。しかし、もし夜中や週末の間にサーバーがダウンしたらどうなるのでしょうか？ サーバーが利用できるかどうか、十分な容量や帯域幅が確保されているかどうかを判断するために、いつも誰かがダッシュボードを見ていることを期待できますか？

期待できないのなら、特定の条件が満たされたときに通知するアラート　ポリシーを作成する必要があります。

たとえば、このスライドに示すように、特定の期間にわたって　VM インスタンスの下り（外向き）ネットワークがしきい値を超えたときに警告するアラート　ポリシーを作成できます。この条件が満たされたとき、この問題のトラブルシューティングを行えるように、メール、SMS などのチャンネルを通じて、ご自身または他のユーザーに自動的に通知できます。

アラート ポリシーを作成して Google Cloud のオペレーション スイートの使用状況をモニタリングし、料金のしきい値に近づいたらアラートを受け取ることもできます。詳細については、こちらの動画のリンク セクションをご覧ください。
[https://cloud.google.com/stackdriver/pricing#alert-usage]

通知ポリシーの作成

これはアラート ポリシーの作成例です。 左側に summer01 インスタンスの HTTP チェック条件が示されています。これにより、右のドキュメント セクションの内容でカスタマイズされたメールが送信されます。

アラートを作成するときのベスト プラクティスをいくつかご紹介します。

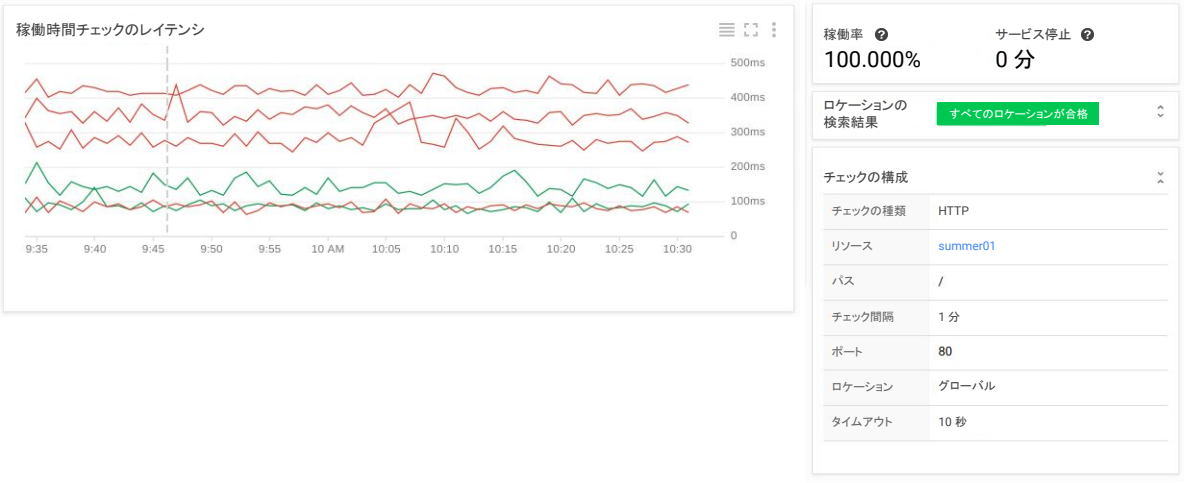
- アラートの対象は症状にすることをおすすめします。必ずしも原因にする必要はありません。たとえば、データベースのクエリの失敗をモニタリングすることで、データベースがダウンしているかどうかを特定します。
- 次に、メール、SMS などの複数の通知チャンネルを使用していることを確認します。これはアラート戦略における単一障害点を防止するのに役立ちます。
- また、必要なアクションや、検証が必要なリソースを記述することで、オーディエンスのニーズに合わせてアラートをカスタマイズすることをおすすめします。
- 最後に、ノイズを避けてください。ノイズが原因で時間経過とともにアラートが解除されてしまいます。具体的には、モニタリング アラートが実用的なものになるように調整します。むやみやたらにアラートを設定しないようにしてください。

稼働時間チェックでパブリック サービスの可用性をテスト

このスライドのとおり、世界各地のパブリック　サービスの可用性をテストするように、稼働時間チェックを構成できます。稼働時間チェックの種類は、 HTTP、HTTPS、TCP のいずれかに設定できます。App Engine アプリケーション、Compute Engine インスタンス、ホストの URL、AWS インスタンス、AWS ロードバランサなどのリソースをチェックできます。

稼働時間チェックごとに、アラート　ポリシーを作成し、各グローバル　ロケーションのレイテンシを表示できます。

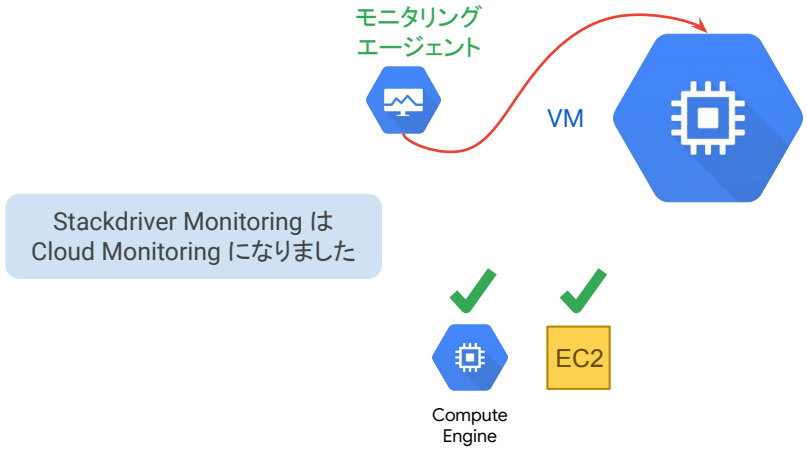
稼働時間チェックの例



これは HTTP 稼働時間チェックの例です。リソースは　1 分ごとにチェックされ、10 秒間のタイムアウトがあります。このタイムアウト期間内にレスポンスがない稼働時間チェックは失敗とみなされます。

ここまでは 100% の稼働時間で、停止もありません。

Monitoring エージェント



Cloud Monitoring は、Monitoring エージェントがなくても、CPU 使用率、一部のディスク トラフィック指標、ネットワーク トラフィック、稼働時間情報など、いくつかの指標にアクセスできます。

ただし、追加のシステム リソースやアプリケーション サービスにアクセスするには、Monitoring エージェントをインストールする必要があります。

Monitoring エージェントは、Compute Engine インスタンスと EC2 インスタンスに対応しています。

Monitoring エージェントのインストール

Monitoring エージェントのインストール（例）

```
curl -sSO https://dl.google.com/cloudagents/add-monitoring-agent-repo.sh
sudo bash add-monitoring-agent-repo.sh
```

Monitoring エージェントは、この 2 つの簡単なコマンドでインストールし、起動スクリプトに含めることができます。

これは、Linux を実行している VM インスタンスがワークスペースによってモニタリングされ、エージェントに対する適切な認証情報がインスタンスに設定されていることを想定しています。最新のコマンドについては、こちらの ドキュメントをご覧ください。

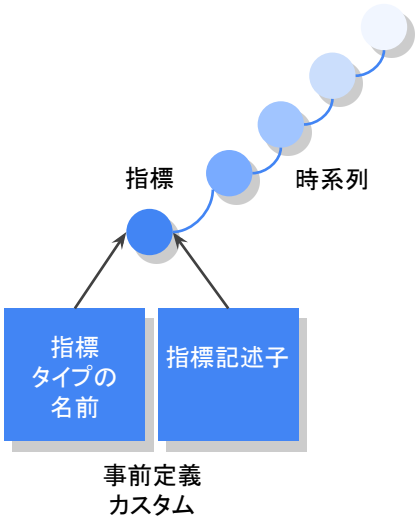
カスタム指標

Python でのカスタム指標の例:

```
client = monitoring.Client()
descriptor = client.metric_descriptor(
    'custom.googleapis.com/my_metric',

metric_kind=monitoring.MetricKind.GAUGE,
    value_type=monitoring.ValueType.DOUBLE,
    description='This is a simple example
of a custom metric.')
descriptor.create()
```

Stackdriver Monitoring は
Cloud Monitoring になりました



Cloud Monitoring が提供する標準指標がニーズに合わない場合は、カスタム指標を作成できます。

たとえば、容量が 50 ユーザーのゲームサーバーがあるとします。スケーリング　イベントをトリガーするには、どのような指標インジケーターを使用できるでしょうか？インフラストラクチャの観点からは、CPU 負荷やネットワーク トラフィックの負荷を、ユーザー数とある程度相関のある値として利用することを検討できます。しかし実際のところ、カスタム指標を使えば、アプリケーションから直接、現在のユーザー数を　Cloud Monitoring に渡すことができるのです。

カスタム指標の作成を開始するには、こちらの動画のリンク　セクションをご覧ください。
[https://cloud.google.com/monitoring/custom-metrics/creating-metrics#monitoring-create-metric-python]

ラボ

リソースのモニタリング

Stackdriver Monitoring は
Cloud Monitoring になりました

ここまでに説明したモニタリングのコンセプトをラボで実践してみましょう。

このラボでは、Cloud Monitoring を使用して、Google Cloud で実行するアプリケーションの分析情報を取得する方法を学びます。具体的には、 Cloud Monitoring を有効にして、ダッシュボードにグラフを追加するほか、アラート、リソース　グループ、稼働時間チェックを作成します。

ラボの復習

リソースのモニタリング

Stackdriver Monitoring は
Cloud Monitoring になりました

このラボでは、Cloud Monitoring の概要について説明しました。プロジェクトのモニタリング、複数の条件を指定したアラートの作成、ダッシュボードへのチャートの追加、リソースグループの作成、サービスの稼働時間チェックの作成について学びました。

モニタリングはアプリケーションの稼働状況のチェックに不可欠な機能です。 Cloud Monitoring には、インフラストラクチャのモニタリング、モニタリング　データの可視化、アラートとイベントのトリガーを行うための豊富な機能が備わっています。

この後、ラボのチュートリアルを参照できますが、 Google Cloud のユーザー インターフェースは変更されることがあるため、実際の環境は見た目が少し異なる場合があります。

アジェンダ

Google Cloud のオペレーション スイート

モニタリング

ラボ

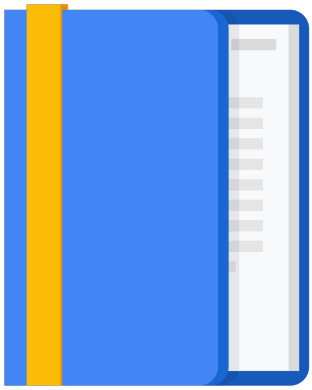
ロギング

エラーレポート

トレース

デバッグ

ラボ



Google Cloud のオペレーション スイートの基礎となるのはモニタリングですが、このサービスは、ロギング、エラーレポート、トレース、デバッグも提供しています。ロギングについて見ていきましょう。

ロギング

- プラットフォーム、システム、アプリケーションのログ
  - ログに書き込むための API
  - 30 日間保存
- ログの検索、表示、フィルタ
- ログベースの指標
- ログのイベントにモニタリング アラートを設定可能
- Cloud Storage、BigQuery、Pub/Sub へのデータのエクスポートが可能

Cloud Logging
（旧 Stackdriver Logging）

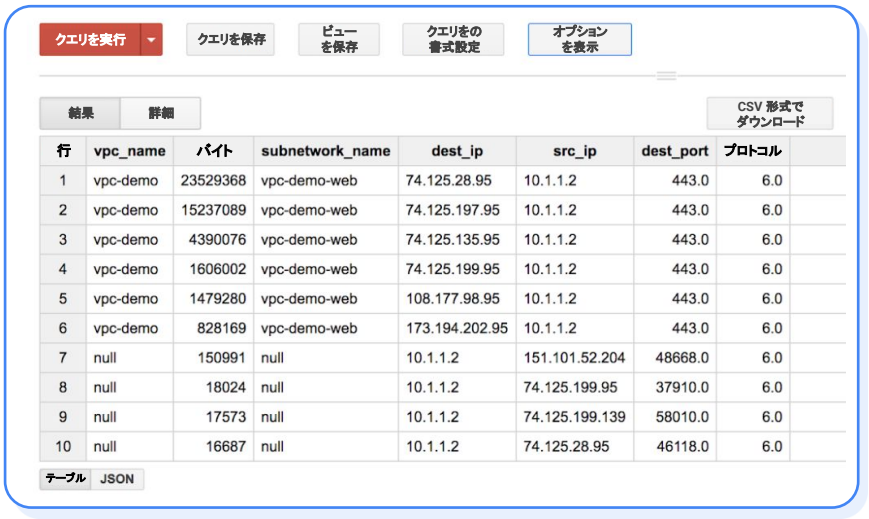
Cloud Logging では、Google Cloud と AWS から取得したログデータとイベントについて、保存、検索、分析、モニタリング、通知を行うことができます。これは、何千もの　VM からアプリケーションとシステムのログデータを取り込んで、大規模に処理できるフルマネージド サービスです。

Logging は、ログのストレージ、ユーザー　インターフェース（ログビューア）、ログをプログラムで管理する API から構成されます。このサービスでは、ログエントリの読み取りと書き込み、ログの検索とフィルタリング、ログベースの指標の作成が可能です。

ログの保持期間は 30 日ですが、Cloud Storage バケット、BigQuery データセット、Pub/Sub トピックにログをエクスポートできます。

Cloud Storage へのログのエクスポートは、ログを　30 日以上保存するときに意味があります。しかし、BigQuery や Pub/Sub にエクスポートする必要があるのは、なぜでしょうか？

BigQuery でログを分析し、データポータルで可視化



行	vpc_name	バイト	subnetwork_name	dest_ip	src_ip	dest_port	プロトコル
1	vpc-demo	23529368	vpc-demo-web	74.125.28.95	10.1.1.2	443.0	6.0
2	vpc-demo	15237089	vpc-demo-web	74.125.197.95	10.1.1.2	443.0	6.0
3	vpc-demo	4390076	vpc-demo-web	74.125.135.95	10.1.1.2	443.0	6.0
4	vpc-demo	1606002	vpc-demo-web	74.125.199.95	10.1.1.2	443.0	6.0
5	vpc-demo	1479280	vpc-demo-web	108.177.98.95	10.1.1.2	443.0	6.0
6	vpc-demo	828169	vpc-demo-web	173.194.202.95	10.1.1.2	443.0	6.0
7	null	150991	null	10.1.1.2	151.101.52.204	48668.0	6.0
8	null	18024	null	10.1.1.2	74.125.199.95	37910.0	6.0
9	null	17573	null	10.1.1.2	74.125.199.139	58010.0	6.0
10	null	16687	null	10.1.1.2	74.125.28.95	46118.0	6.0

テーブル JSON

ログを BigQuery にエクスポートすると、ログを分析し、データポータルで可視化できます。

BigQuery は、ギガバイトからペタバイト規模のデータに対して SQL クエリを超高速で実行します。これによりネットワーク トラフィックなどのログを分析できるため、トラフィックの増加を把握して容量を予測し、ネットワークの使用状況を把握してネットワーク トラフィックの費用を最適化し、ネットワーク フォレンジックによってインシデントを分析することができます。

たとえば、このスクリーンショットでは、ログを照会して、自分のウェブサーバーとやり取りしたトラフィックが特に多い IP アドレスを特定しました。その IP アドレスがどこにあるのか、誰に属しているのかに応じて、インフラストラクチャの一部を再配置してネットワーク費用を節約することや、自分のウェブサーバーにアクセスしてほしくない IP アドレスを拒否することができます。

ログを可視化する必要がある場合は、BigQuery テーブルをデータポータルに接続することをおすすめします。データポータルは元データを指標やディメンションに変換します。これを使用して、わかりやすいレポートやダッシュボードを作成できます。

ログを Cloud Pub/Sub にエクスポートできることにも触れました。これにより、ログをアプリケーションやエンドポイントにストリーミングできます。

Logging エージェントのインストール

Logging エージェントのインストール

```
curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
sudo bash install-logging-agent.sh
```

Cloud Monitoring エージェントと同様、Logging エージェントもすべての VM インスタンスにインストールすることがベスト プラクティスとして推奨されます。Logging エージェントは、この 2 つの簡単なコマンドでインストールし、起動スクリプトに含めることができます。

このエージェントは、Compute Engine インスタンスと EC2 インスタンスに対応しています。

アジェンダ

Google Cloud のオペレーション スイート

モニタリング

ラボ

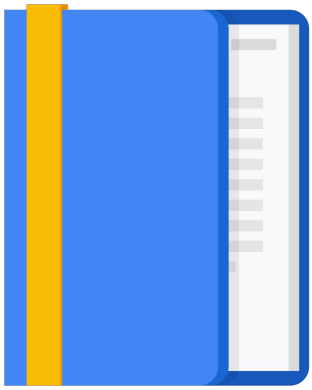
ロギング

エラーレポート

トレース

デバッグ

ラボ



Google Cloud のオペレーション スイートのもう一つの機能である Error Reporting を見ていきましょう。

エラーレポート

クラウド サービスの実行中に発生した
エラーを集計して表示

- エラー通知
- エラー ダッシュボード
- App Engine、Apps Script、Compute Engine、Cloud Functions、Cloud Run、GKE、Amazon EC2
- Go、Java、.NET、Node.js、PHP、Python、Ruby

エラー
レポート

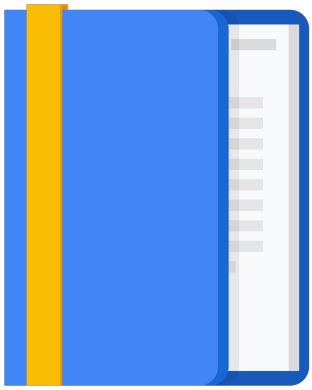
Error Reporting は、稼働中のクラウド サービスで発生したエラーの回数をカウントし、分析と集計を行います。結果は、並べ替え機能やフィルタリング機能を備えた、一元化されたエラー管理インターフェースに表示されます。また、リアルタイム通知を設定して、新しいエラーが検出されたときに通知することもできます。

プログラミング言語については、例外スタック トレースのパーサーが、Go、Java、.NET、Node.js、PHP、Python、Ruby を解析できます。

ちなみに、今 App Engine について話しているのは、次のラボで App Engine にデプロイされたアプリで Error Reporting を使用するためです。

アジェンダ

- Google Cloud のオペレーション スイート
- モニタリング
- ラボ
- ロギング
- エラーレポート
- **トレース**
- デバッグ
- ラボ



トレースは、Google Cloud に統合された Cloud Operations 機能の一つです。

トレース

トレース システム

- ほぼリアルタイムでデータを表示
- レイテンシ レポート
- URL ごとのレイテンシのサンプリング

レイテンシ データの収集

- App Engine
- Google HTTP(S) ロードバランサ
- Cloud Trace SDK でインストルメントされたアプリケーション

Cloud Trace
（旧 Stackdriver Trace）

Cloud Trace は分散トレース システムであり、アプリケーションからレイテンシ　データを収集し、Cloud Console に表示します。アプリケーション内でやり取りされるリクエストをトラックし、ほぼリアルタイムでパフォーマンスの分析情報を提供できます。

Cloud Trace は、アプリケーションのすべてのトレースを自動的に分析し、パフォーマンス低下の原因となるレイテンシの詳細レポートを生成します。また、 App Engine、HTTP(S) ロードバランサ、Cloud Trace API でインストルメントされたアプリケーションからトレースを取得できます。

アプリケーションが受信リクエストを処理し、オペレーションの実行にかかる時間を管理することは、アプリケーションの全体的なパフォーマンスの管理において重要な役割を果たします。実際、Cloud Trace は、Google がきわめて大規模なサービスを運用し続けるために使用しているツールをベースにしています。

アジェンダ

Google Cloud のオペレーション スイート

モニタリング

ラボ

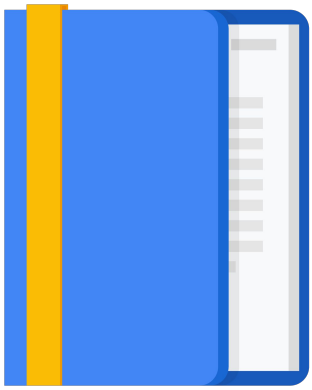
ロギング

エラーレポート

トレース

デバッグ

ラボ



最後に、このモジュールの最後の Google Cloud のオペレーション スイート機能である、デバッガについて説明します。

デバッグ

- アプリケーションの停止や大幅な速度低下を発生させずに検査を実施
- デバッグ スナップショット:
  - 実行中のアプリケーションのコールスタックとローカル変数をキャプチャ
- デバッグ ログポイント:
  - サービスを停止させずに、そのサービスにロギングを挿入。
- Java、Python、Go、Node.js、Ruby、PHP、.NET Core

Cloud デバッガ
（旧 Stackdriver Debugger）

Cloud デバッガは、アプリケーションの停止や実行速度の低下を発生させずに、実行中のアプリケーションの状態をリアルタイムで調査できる　Google Cloud の機能です。具体的には、デバッガでは、アプリケーションの状態がキャプチャされているときに、 10 ミリ秒未満のリクエスト レイテンシが発生します。ほとんどの場合、ユーザーは気付きません。

この機能を使用すると、本番環境でのコードの動作を把握し、状態を分析して、見つけにくいバグを特定できます。数回クリックするだけで、実行中のアプリケーションの状態のスナップショットをとったり、新しいロギング　ステートメントを挿入したりできます。

Cloud デバッガは、Java、Python、Go、Node.js、Ruby など、複数の言語をサポートしています。

ラボ

エラーレポート とデバッグ

Stackdriver は Google Cloud の オペレーション スイートになりました

先ほど学習したロギング、エラーレポート、デバッグを、ラボで実践してみましょう。

このラボでは、小さな「 Hello, World」アプリケーションを App Engine にデプロイします。その後、アプリケーションにバグを追加して、エラーレポート機能とデバッグ機能を使ってみます。

ラボの復習

エラーレポート
とデバッグ

Stackdriver は Google Cloud の
オペレーション スイートになりました

このラボでは、アプリケーションを App Engine にデプロイしました。その後、アプリケーションが機能しないバグをコードに追加しました。 Error Reporting で問題を特定して分析し、Cloud デバッガで根本原因を判別しました。最後に、コードを変更して問題を修正しました。

これらのツールをすべて GCP に統合することで、コードとこれに関連するトラブルシューティングに集中できるようになります。

この後、ラボのチュートリアルを参照できますが、 GCP のユーザー インターフェースは変更されることがあるため、実際の環境は見た目が少し異なる場合があります。

まとめ

リソースのモニタリング

Stackdriver は Google Cloud の
オペレーション スイートになりました

このモジュールでは、Google Cloud のオペレーション スイートと、そのモニタリング、ロギング、エラーレポート、フォールト トレース、デバッグ機能の概要について学習しました。このすべてを GCP に統合することで、サイト信頼性エンジニアリング（ SRE）と呼ばれるアプリケーションの運用、保守が可能になります。

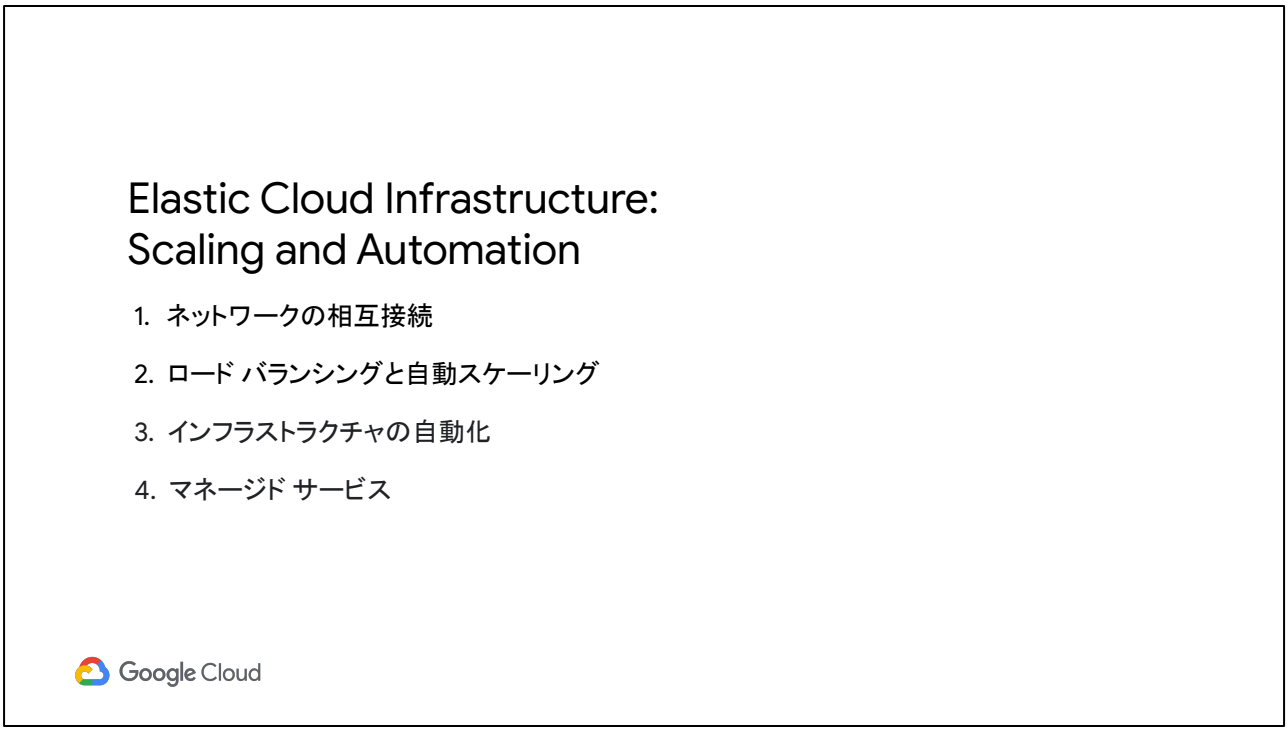
SRE について詳しくは、書籍または一部の SRE コースをご覧ください。

まとめ

Essential Cloud
Infrastructure: Core Services

Google Cloud

「Essential Cloud Infrastructure: Core Services」コースを受講いただき、ありがとうございます。IAM の管理、GCP のさまざまなデータ ストレージ サービスの選択、GCP リソースに対する請求の管理、そのリソースのモニタリングについてご理解を深めていただけたかと思います。デモとラボを通じて、今回取り上げた各　GCP サービスは使いやすいと感じていただけかとも思います。



次に、「Architecting with Google Compute Engine」シリーズの「Elastic Cloud Infrastructure: Scaling and Automation」コースを受講することをおすすめします。

1. このコースでは、まず、ネットワークを相互接続するさまざまな方法について説明し、インフラストラクチャを GCP に接続できるようにします。
2. 次に、GCP のロード バランシングと自動スケーリングのサービスについて説明し、実際に操作してもらいます。
3. その後、Terraform などのインフラストラクチャ自動化サービスについて確認し、GCP インフラストラクチャ サービスのデプロイを自動化できるようにします。
4. 最後に、GCP で活用できるその他のマネージド サービスについて説明します。

このコースをぜひご活用ください。