ゆるふわ！？ Stackdriver Logging & Monitoring

　はじめまして、株式会社grasys 大川です(社内ではStackdriverおじさんと呼ばれています。)
Stackdriverをはじめて使ってみようとした際に日本語での解説が少なく取っ掛かりにくいという声を聞くことがありましたので、よくある監視構成についてゆるく執筆しました。Stackdriverの経験がない・少ない方の導入の一助になれば幸いです。

1. Stackdriver のちょっと雑な説明

　StackdriverはGoogle社が提供するリソース・アプリケーションのモニタリングプラットフォームです。元々StackdriverはStackdriver社が提供するサービスでしたが、2014年にGoogleが買収、以後はGoogle Cloud Platform(以下、GCP)のプロダクトStackdriver Monitoringとして提供されています。
　現在はサーバの状態監視以外にもProfiler,Traceなどアプリケーションの状態管理・解析プロダクトも追加され、GCP上で提供するサービスの維持・改善を包括的に行えるプラットフォームとして進化し続けています。

1.1 Stackdriver関連プロダクト

　2019年11月現在、GCPコンソールから選択できるStackdriver関連のプロダクトは Monitoring, Debug, Trace, Logging, Error Reporting, Profiler の6種類になります。
　本章ではこの中から**Logging**と**Monitoring**に焦点を当て解説します。

1.1.1 Logging

　GCPやAmazon Web Services(以下、AWS) からのログデータやイベントの格納、検索、分析、モニタリング、通知ができます。

1.1.2 Monitoring

　Monitoringの主な特徴としてGCP、AWSをはじめとしたマルチプラットフォームの横断的な監視管理、柔軟な設定、簡易な通知設定が挙げられます。
　例えばGCPのCloudSpannerやAWS RDSのようなマネージドサービスのメトリクスを取得することもできますし、OS(Linux etc.)にモニタリング用のエージェントをインストールし設定することによって、サーバが設置されている場所を問わず監視を行うことが出来ます。
※メトリクスの量によって課金額が異なってきますので、ご利用は計画的に、です。

2. ゆるくログ監視と通知を試した件

　それではよくある監視構成のケースとして、サーバから出力されたnginxのログをStackdriver Loggingで受け取り、statusが200,301以外だった場合にSlackへと通知する流れを解説します。

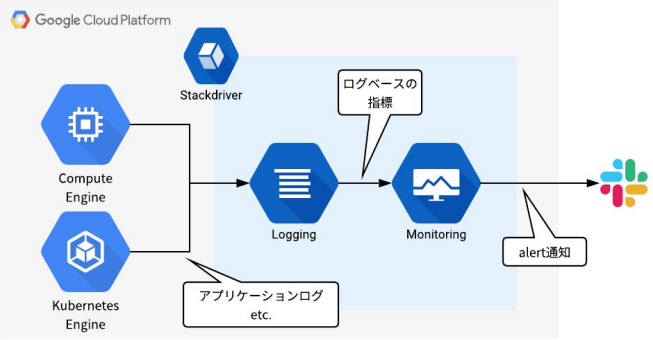


図. ログの通知フロー

2.1 Stackdriver Logging - ログの指標設定

最初に、LoggingからMonitoringに指標を渡す設定を行います。おおまかにはインスタンスからLoggingへログを渡すエージェント設定、Loggingに渡されたログをフィルタリングルールに従いMonitoringに渡すログベースの指標設定の2つが必要になります。

2.1.1 Logging エージェントのインストール

VMインスタンスのログを集約するためには、Stackdriver Logging エージェントをインストールする必要があります。通常、GCP内のVMインスタンスであれば特別な設定を行わずに自身のプロジェクトへログを送信することが出来ます。[1]またRHEL/CentOSの場合はGoogle Cloudのyumリポジトリより **google-fluentd-catch-all-config-structured** をインストールすることで、nginxをはじめとするサードパーティ製ソフトウェア向けの設定を導入できます。

本章では **google-fluentd-catch-all-config-structured** をインストールしたうえで解説を進めます。

2.1.2 ログベースの指標

Stackdriver Monitoringと連携するルールの設定を行えます。今回設定する「**GCPインスタンスから出力されるnginxのログ**から**response_codeが200,301以外**をピックアップしたデータ」をモニタリングしたい場合は、GCPダッシュボード ログビューアより「高度なフィルタに変換」を選択し、下記設定のうえ「フィルタを送信」ボタンを押し決定します。

```
resource.type="gce_instance" AND
logName="projects/<project_id>[2]/logs/nginx-access" AND
jsonPayload.code!="200" OR
jsonPayload.code!="301"
```

次に、指標エディタで指標名などを指定します。
「アイテムを追加」からフィールド名 **jsonPayload.code** を指定することにより、Monitoringに送る指標にresponse_codeを含めて500系のレスポンス数と400系のレスポンス数とで通知の閾値を変えるといった運用が可能になります。
他にもフィールド名**jsonPayload.path**を指定し特定のpathを監視するなど、アイテムを使用した幅広い監視の設定が可能です。

指標の注意点として、設定された**以降**に流入したログに対してのみ有効になります。
過去に遡って指標を適用することはできませんので、ご注意ください。

※注意

Loggingから出力された指標をMonitoringで扱うためには、最低1回はMonitoringに指標を渡す必要があります。

指標の設定完了後は一度404など200,301以外のレスポンスが返されるリクエストを該当サーバに投げ、Stackdriver Loggingにログが格納されていることを確認してください。

---

[1] /etc/google-fulentd/config.d/ 以下に追記することで独自形式のログも送ることが出来ます。
[2] <project_id>の部分はご自身のproject_idに置き換えてください。

2.1.2.1 フィルタリングの基本的な表記

ログベースの指標の設定で良く使う表記をピックアップしました。その他の詳しくはloggingドキュメント**Advanced logs queries**[3]を参照してください。

記号	説明	例
=	完全一致	resource.type="gce_instance"
!=	否定	labels."compute.googleapis.com/resource_name"!="prd-web01"
:	部分一致	labels."compute.googleapis.com/resource_name": "prd-web"
> < >= <=	不等式	severity>=WARNING
OR	論理和	textPayload: ("/wp-admin/" OR "/phpMyAdmin/")

表. よく使う表記

2.1.3 ログの保存期間

おおよそのログは保存期間が30日までで、それ以前のログは削除されます。30日以上の期間ログを保存したい場合は「エクスポートを作成」からGCS/BigQuery等へ保存設定を行ってください。

2.2 Stackdriver Monitoring - 監視・通知設定

最後に、Monitoringより監視・通知の設定を行います。

2.2.1 Stackdriver Monitoring - 通知先設定

監視設定では通知先が必要なため、先にSlackチャンネルを通知先として設定します。

1. Monitoringダッシュボードのworkspace選択より「Workspace Settings」を選択
2. Settings一覧より「Notifications」を選択し、SLACKタブを指定
3. Add Slack Channelを選択し、通知したいSlackのworkspaceと連携[4]

2.2.2 Alerting Policy

Stackdriver Monitoring ダッシュボードの左のメニューより Alerting > Policies を選択し、Policies一覧画面の右上にある **Add Policy** ボタンを押します。以下の項目を入力することで、通知を設定できます。

項目	内容
Conditions	指標を設定します。 1つのPolicyに対し5つまでのConditionを作成することが出来ます。 複数条件を満たす場合のみ通知するといった設定も可能です。
Notifications(optional)	通知対象を設定します。 Slack + Email など、複数種別の通知先を設定することも可能です。
Documentation(optional)	通知本文にドキュメントを記載することもできます。 障害の概要などを書いておくと、初動の役に立つかもしれません。
Name this policy	ポリシー名です。

表. Alerting Policy設定内容

---

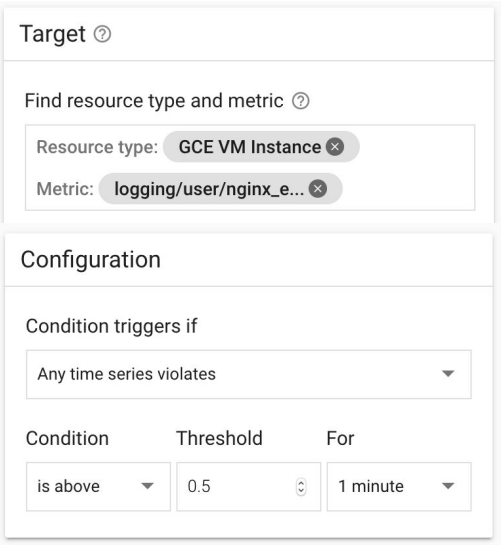
[3] https://cloud.google.com/logging/docs/view/advanced-queries

[4] 退社するなどして連携を行ったSlackのアカウントが削除された場合は、Alertが発報されなくなります。
Slackアカウントの運用にはご注意ください。

2.2.3 Alerting - Condition

それでは実際にConditionを設定します。例として、**200,301以外のログが1分間に30回以上出力されたら発報**を設定します。

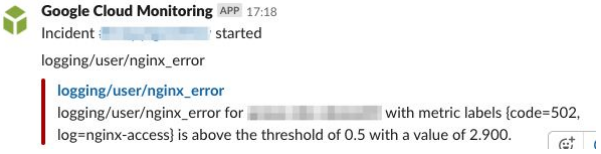
1. 項目**Conditions**から**Add Condition**を選択
2. TargetのResource Typeに**GCE VM Instance**を指定
3. Metricに **logging/user/nginx_error (logging.googleapis.com/user/nginx_error)** を指定
4. Aggregatorに**none**を設定
5. ConfigurationのThresholdに0.5を設定
6. 他はデフォルトのまま
7. save でconditionを保存



注意点としては、Thresholdで判定される値は**per sec**のため、**1分間に30回以上の場合は0.5**に設定する必要があります。condition保存後にPolicyの保存を行うことで、通知設定を完了します。

2.2.4 Alert Test

ロギング設定を行ったサーバにアクセスし、404などエラーを発生させ通知を確認します。



2.3 Monitoring エージェントについて

本章ではログの監視に説明を絞ったため事後の解説となりますが、Stackdriver Monitoringエージェントについても簡単に触れます。

GCE VMインスタンスの監視項目はMonitoring API のメトリクス[5]である程度の取得できますが、例えばメモリ使用率のようにメトリクスが存在しないため取得することが出来ない項目があります。そのような場合、Monitoringエージェントを導入することでAPIで取得できない監視項目を補うことが出来ます。また、サーバプロセス(例: nginx)の稼働状況を監視したい場合も、エージェントをインストールすることにより監視が可能になります。

エージェントは **collectd** [6]にGCP連携のための設定を追加したものが使用されています。多くのミドルウェアは事前定義されているためそのままメトリクスを収集できますが、自社製サーバプログラムなどを監視したい場合などは別途カスタムメトリクスを作成する必要があります。

3. おわりに

駆け足となりますが、Stackdriverによるログの監視の入り口についてふんわり説明しました。ログの監視は奥深く、ステータスだけではなくpath単位でのエラーの出現頻度を割り出したり、インスタンスだけではなくLoadbalancerをはじめとするマネージドサービスのログを元に応用することが出来ます。

また、Stackdriverには他にも様々なプロダクトがありますので、障害監視をはじめアプリケーションのレスポンス改善など、開発・運用するサービスの品質向上にお役立ていただけますと幸いです。

---

[5] https://cloud.google.com/monitoring/api/metrics_gcp#gcp-compute
[6] https://collectd.org/