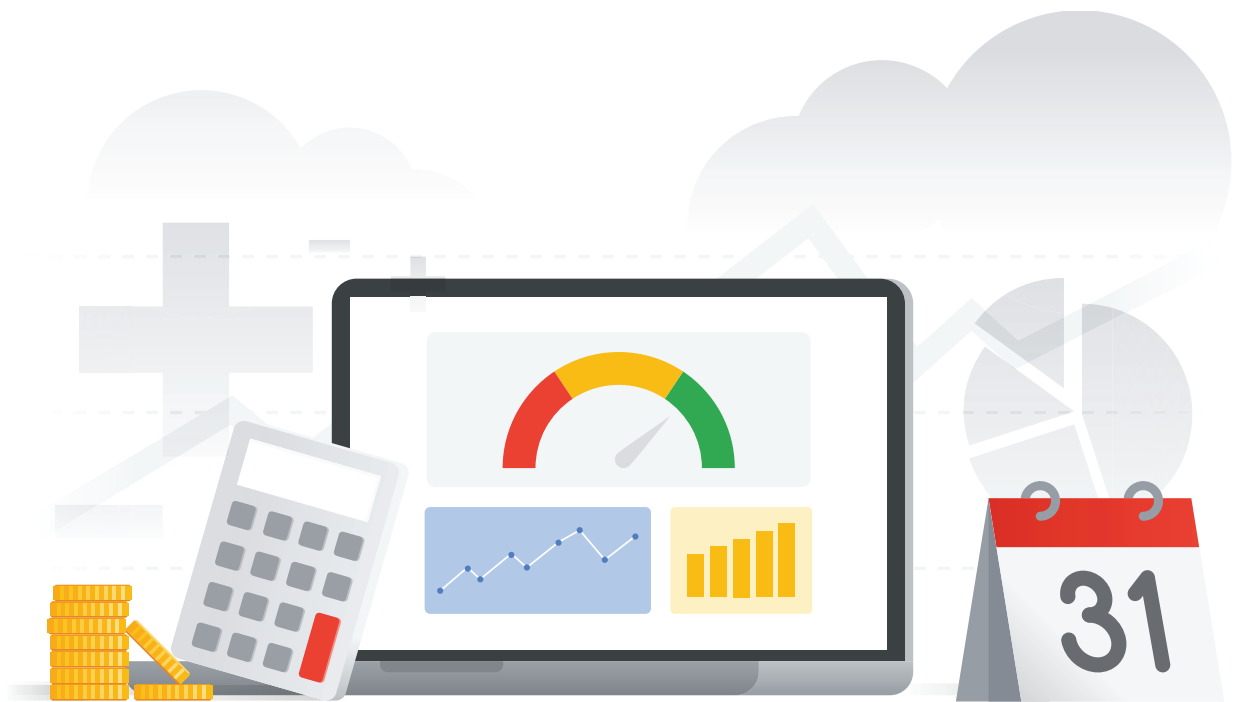




Google Cloud ホワイトペーパー

クラウド利用時におけるコスト最適化の原則



Google Cloud

目次

はじめに	1
------------	---

第 1 章 クラウドの費用最適化のための原則とプロセス	2
-----------------------------------	---

人とプロセスによる最適化
価値と費用の関係性を理解する
最初から標準化されたプロセスを実装する
最高の結果を得るための継続的なレビュー
費用最適化のツール
最適化案の優先順位のつけ方

第 2 章 コンピューティング費用の最適化	12
-----------------------------	----

費用の削減を目指すにあたって
高価なだけがコンピューティングではない

第 3 章 ストレージ費用の最適化	21
-------------------------	----

クラウド移行時にストレージをクリーンアップする
保持に関する考慮事項とヒント
アクセス パターンに関する考慮事項とヒント
パフォーマンスに関する考慮事項とヒント

目次(つづき)

第 4 章 ネットワーク費用の最適化..... 28

ネットワークトラフィックフローを理解する
「トラフィック使用量が特に多いプロセス」を見極める
VPN を使用するタイミングを決定する
階層を利用したネットワークの最適化
ネットワーク使用量の最適化

第 5 章 BigQuery によるデータ分析費用の最適化..... 34

BigQuery の基本的な料金体系を理解する
定額料金とオンデマンド料金の違いを理解する
BigQuery の費用最適化テクニック: クエリの処理
BigQuery の費用最適化テクニック: ストレージ
努力を称える
実現された費用の最適化

協力者

以下の Cloud プロフェッショナル サービス チームメンバーの貢献に深く感謝します。

- Justin Lerma
- Pathik Sharma
- Amber Yadron
- Andrew Sallaway
- Akshay Kumbhar



はじめに

ビジネスの計画や目標を立てる際、ある程度の不確実性はつきものです。顧客が製品に対してどのような反応をするか、販売予測が正確かどうかなどは、実際にはわからないものです。しかし、自社の技術システムを効率良く稼働させるための管理を行ったり、システムがどの程度ビジネス ユーザーに適切なサービスを提供しているかを確認したりすることは可能です。従来のビジネスの不確実性に加えて世界規模で不確実性が増している今、既存の技術リソースを賢く使用し、できるだけ多く活用することがいっそう重要になっています。

結局のところ、クラウド リソースを最大限に活用することがより多くの顧客へのサービス提供、より多くの問題の解決、ビジネス全体の適応性の向上につながるのです。クラウド リソースをもっと効率的に利用すれば、お客様のチームやビジネスがこのような新しい事態に適応し、より効果的に事業を進めることができます。

Google Cloud のチームは、長期にわたり世界中のさまざまな業界の IT チームや運用チームと協力し、お客様の課題、成功体験、将来計画に耳を傾けてきました。多くが変化する中、テクノロジストの適応力と成功力は変化していません。ビジネスが変化するにつれて移行、変化に対応する能力はますます重要になっています。Google Cloud のテクノロジーはそのようなアジリティと復元性をサポートするよう設計されています。おすすめのヒントやベスト プラクティスを集めました。より多くの VM、ストレージ、クエリなどの既存のリソースを最大限に活用して、ビジネス目標の達成にお役立てください。

本ホワイトペーパーには、Google Cloud に組み込まれた費用最適化を実現する機能の使用方法と、リソースを有効に活用できるヒントを多数掲載していますので、ぜひご覧ください。さらに、プロセスの設定と、機能横断型チームと協力して、クラウドの効率性と費用最適化をビジネス文化の一部とする基準の実装に関するヒントも掲載しています。

ビジネスが変化するにつれて移行、変化に対応する能力はますます重要になっています。

第 1 章

クラウドの費用最適化のための原則とプロセス

クラウドはただのコストセンターではありません。クラウドに移行することで、グローバル規模でのイノベーションを可能にし、機能の速度向上により製品化までの時間を短縮するとともに、お客様のニーズに迅速に対応することで競争優位性を高めることができます。そのため、多くの企業が組織のデジタル戦略を早急に変革しようとするのは当然のことです。クラウドを迅速に導入することは大事ですが、アプリケーションをクラウドに移行、デプロイする前に、時間をかけて重要なコンセプトを確認することも大事なことです。同様に、もしすでにクラウドに既存のアプリケーションがある場合は、環境を監査して、ベスト プラクティスに従っているかを確認する必要があります。その目的は、クラウド リソースの最も効果的かつ効率的な活用を念頭に置き、費用を最適化しつつビジネスの価値を最大限に高めることにあります。

Google は、複雑な問題を抱えた一部のお客様と協力して、Google Cloud の次世代アプリケーションやサービスの導入に取り組んできました。費用の最適化に関しては、組織が利用できるたくさんのツールやテクニックが用意されています。しかし、ツールにできるのはここまでです。クラウドを最大限に活用するためには、組織がその規模にかかわらず、従うべき高度な原則がいくつかあります。

人とプロセスによる最適化

多くのテクノロジーと同様、優れた基準は、守られてこそ意味をなします。多くの場合、制限的要因はテクノロジーの能力ではなく、関係する人とプロセスです。費用の最適化には、エグゼクティブ チーム、プロジェクト リーダー、財務部門、サイト信頼性エンジニア (SRE) のすべてが関連しあっています。最初のステップとして、これらの主な利害関係者は、必要なサービスレベルの収益性、信頼性、およびパフォーマンスを概説する会社の基準セットを設けるために、ミーティングを行う必要があります。

費用の最適化に関しては、たくさんのツールやテクニックが用意されています。

このイニシアチブを始動させる専門家チームを結成することを強くおすすめします。本書の後半では、具体的なサービスレベルの最適化案について説明しますが、フレームワークがプログラムで適用されるまでは、いずれもそのスケーラビリティや効果は望めません。

クラウドの強化された費用把握を利用する

クラウド環境の主なメリットは、利用データの可視性が強化されることです。各クラウド サービスは追跡が可能であり、個別に測定できます。これは諸刃の剣となりえます。今や何万もの SKU が存在し、誰がどのサービスをどんな理由で購入しているのかわからなければ、クラウドでデプロイされているアプリケーションやサービスの総所有費用 (TCO) を把握することが困難になるからです。

これは、お客様がオンプレミスの設備投資 (CapEx) モデルからクラウドベースの運用費用 (OpEx) モデルに最初にシフトする際によく見られる問題です。昔は、中央の財務チームが固定予算を設定し、必要なリソースを調達していました。過去の成長率などの指標に基づいて予測がなされ、次の月、四半期、年、あるいは複数年のニーズを判断するのに用いられていました。全員が揃って会社全体でその必要性を検討する機会が得られるまで、購入は行われませんでした。

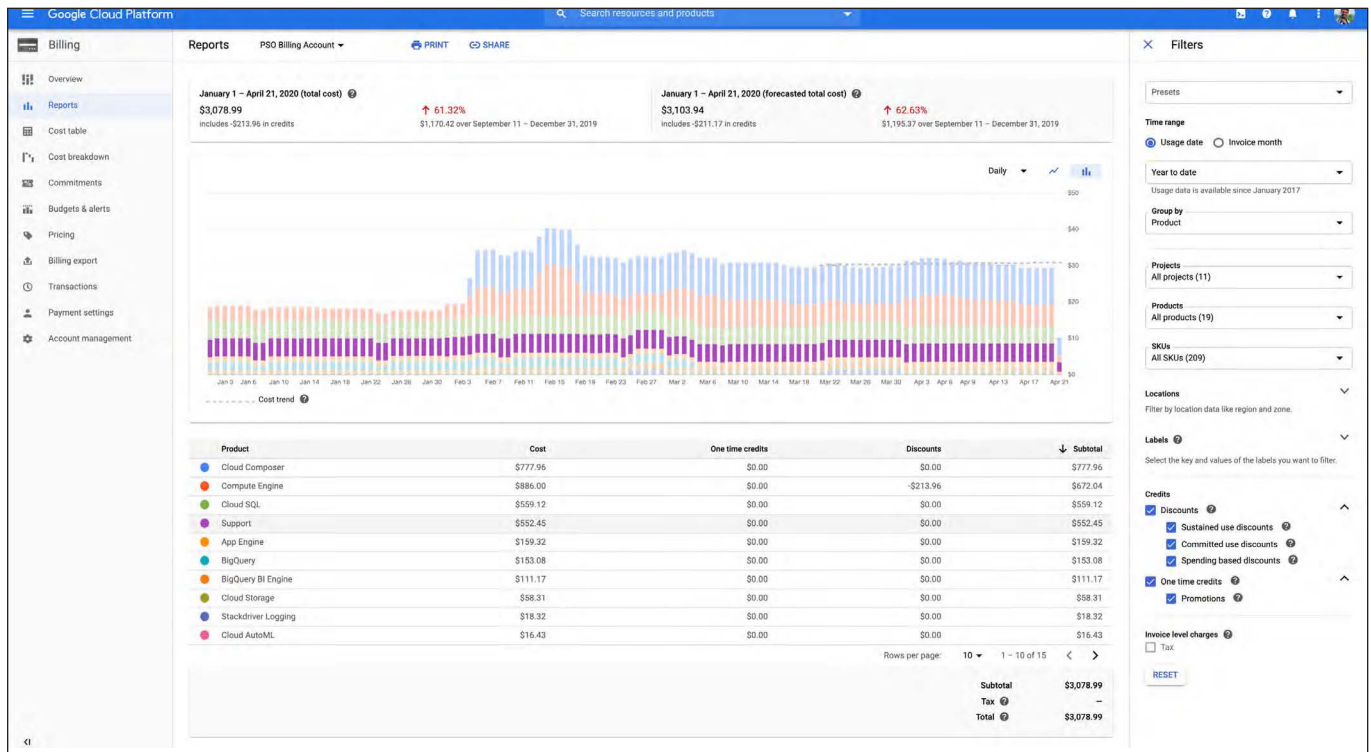
現在、OpEx 環境では、エンジニアリングチームが必要に応じてリソースを増やし、サービスを最適に走行させることができます。クラウドを利用するお客様において、予算やアラートの設定、適切なリソースのラベリング、エンジニアリングと財務の観点から費用を確認する頻度などの標準化されたガードレールの設定がないまま、エンジニアリングチームがリソースを増やすという「荒野の無法地帯」的状况が多く見受けられます。これは速度の向上を強化するものですが、サービスの費用対価値の方程式を効果的に設計するための出発点としてはあまり良いとは言えません (本質的に、サービスによって生成される価値に対して、支出の最適化がはるかに少ないため)。標準化されたラベル付けの慣行がないため、お客様が自社の環境で開発プロジェクトと生産プロジェクトの費用を特定するのに苦労しているのを目の当たりにしてきました。また、別のケースではエンジニアがパフォーマンスの問題を回避するためにインスタンスを過剰にプロビジョニングし、ピーク時以外の時間帯にはかなりのオーバーヘッドが発生しています。これでは、長い目で見るとリソースの無駄遣いになってしまいます。利用可能なリソースの種類とそれらをデプロイするタイミングについて全社的な基準を作成することは、クラウド費用を最適化するうえで最も重要です。

クラウドを使用される
多くのお客様にとって、
それはまさに「荒野の
無法地帯」です。

これまでに何度もこのような動き(変化)を見てきましたが、クラウドの最も望ましい特長の一つである柔軟性がときに問題視されてしまうことがあります。請求書の額が予想外に高い場合、費用の増加を気にされるお客様もいらっしゃいます。処理されたトランザクションやサービスが提供されたユーザー数などのビジネス指標と照らし合わせ費用増加の原因を考えない限り、クラウド費用の解釈を誤ってしまいます。多くのお客様にとって、費用増加やその原因がある特定のビジネス オーナーやグループにあることをと確認することは簡単ですが、プロジェクト オーナーに具体的な最適化案を提供できるだけの状況を理解する十分な情報を持っているとは言えません。チームはより多くのお客様に優れたサービスを提供するために、費用が増加したのかもしれないのです。逆に、週末に誰かが不要な高 CPU の VM をシャットダウンするのを忘れたために、それがオーストラリアへの不要なトラフィックの原因となり費用が上昇したのかもしれない。

この問題を解決する方法の一つは、ビジネスニーズに応じて費用を整理し、構造化することです。そして、Cloud Billing レポートを使ってサービスの内容を掘り下げ、費用を一目で把握できます。また、ラベルを使用して費用の発生元の部門やチームを確認できるようにし、独自のカスタムダッシュボードを構築することにより、環境におけるより詳細な費用ビューを取得することもできます。このアプローチでは、事前に定義されたビジネス指標に基づいてリソースにラベル付けし、時間の経過とともにその支出をトラックできます。長期的には、「先月、Compute Engine に \$X を使った」ということではなく、「\$Y の収益をもたらす顧客にサービスを提供するには \$X の費用がかかる」ということを理解することがゴールとなります。このような分析こそが、作成できるように努力すべきタイプの分析です。

クラウドの特に望ましい特長の一つである柔軟性がときに問題視されてしまうことがあります。



Google Cloud Console の課金レポートで、費用の詳細な内容が確認できます。

クラウドの主な特長の 1 つは、機能の速度向上により製品化までの時間を短縮できることであり、この柔軟性のおかげで従来のオンプレミス環境では何ヶ月も待たされていたワークロードのデプロイを数分で実施できます。実際のビジネスの成長速度が常にわかるとは限らないため、事前に費用把握モデルを確立することが不可欠です。そして、単純なサービスごとの費用の指標を把握できるようになれば、プロジェクトごとのパフォーマンス指標として、収益性のような新規ビジネス指標の測定を開始できます。

価値と費用の関係性を理解する

複雑なクラウド システムを構築する目的は、単に費用削減だけではありません。フィットネスに例えてみましょう。より健康な体を作ろうとすると、多くの人は体重を減らすことにこだわります。しかし、体重の減少は必ずしも健康であることの重要な指標とは限りません。病気や脱水状態になったとしても、体重は減少するからです。減量のような指標を目指す場合、実際に重要視するのは、総合的な健康（フィットネス）や活動時の自分の容姿や気分です。たとえば子どもの相手ができること、長生きすること、ダンスを楽しむ、といったことです。同じように、費用を最適化することは、費用を削減することだけではなく、無駄を見つけ、費やしたお金すべての価値を最大化することです。

同様に、最適化に精通したお客様は特定の費用削減数値に固執するのではなく、総合的な運用の適合性を把握するために以下のようなさまざまな質問をされます。

- 顧客に対して実際に提供しているものは何か(単位)?
- それを提供するための費用はいくらか?
- 作成されたものの単位当たりのすべての関連する支出はどのように最適化できるか?

つまり、このようなお客様は先を行っており、独自の**ユニットエコノミクス**のモデルを作成しているのです。前もってこれらの質問を投げかけ、次にこれらの重要な質問に答えるだけでなく、彼らの行動の監査が可能なシステムの構築に取り組んでいます。このような取り組みは、「crawl(赤ちゃんのハイハイ)」段階にあるお客様には滅多に見られませんが、「walk(歩く)」段階にある多くのお客様は、このコンセプトを採用して将来のシステムを設計しています。



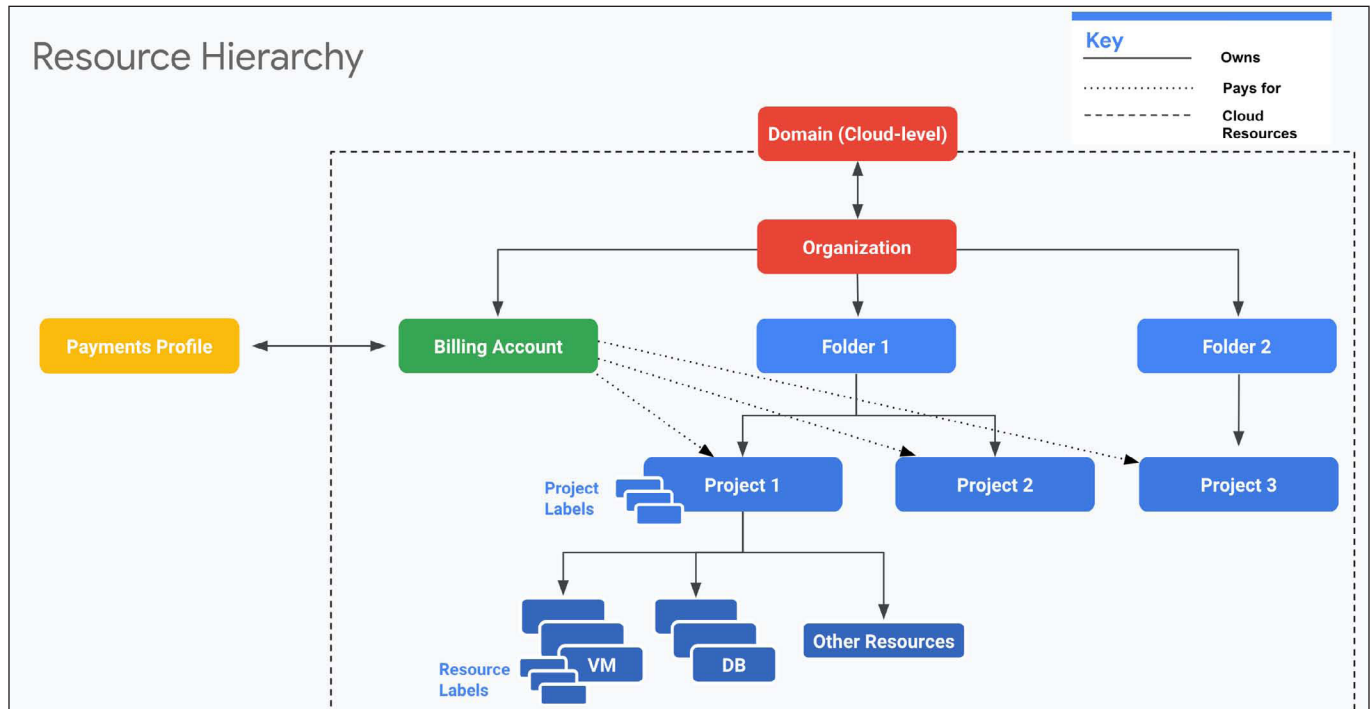
最初から標準化されたプロセスを実装する

最適化案を着実に実装するには、組織的に設計、実施する必要があります。**Terraform** や **Cloud Deployment Manager** などの自動化ツールは、クラウド リソースをデプロイする前のガードレールの設定に役立ちます。過去にさかのぼって基準を実装することは、事前に行うよりもはるかに困難です。IT Ops がタグなしのリソースを停止または停止すると脅したことや、基準を遵守しなかった者を「恥の壁」で隔離するなど、ありとあらゆるケースを見てきました。(ピザやトロフィー、さらにはピザでできたトロフィーなど、ポジティブな促進剤は大歓迎です。)

初期の段階で標準化させておきたい最適化プロセスの例には、どんなものがあるでしょうか。1 つはリソースのデプロイです。エンジニア全員がすべてのリソースをいくらかでもデプロイできるようにすべきでしょうか?おそらく違います。ここは、事前に標準化を行うことで大きな違いを生むエリアであると考えています。



リソースの構造化により効率よく費用の管理を行うことも重要です。最も良い方法は、初期の要件を満たす、できるだけシンプルな構造を採用し、要件の変化に応じてリソース階層を調整するというものです。設定ウィザードを使えば、最適化案と手順の案内を受けて最適な環境を作ることができます。このリソース階層内では、プロジェクト、フォルダー、ラベルを使用してリソースの論理グループを作成できます。こうすることで、管理と原価配分の要件をサポートできます。

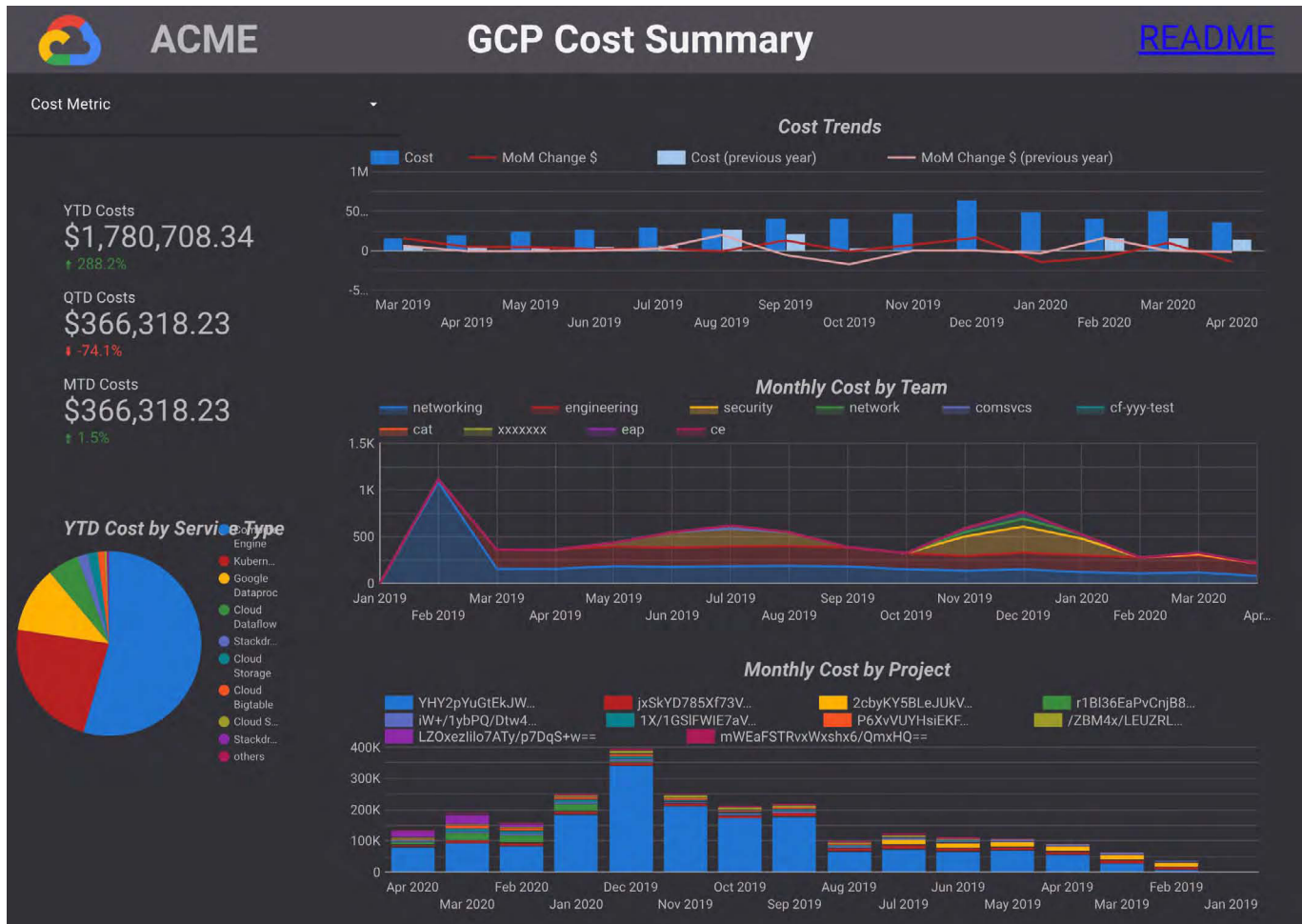


クラウドにおけるリソース階層の例

リソース階層において、リソースのラベル付けは費用管理に関心のある組織にとって最優先事項です。これは、本質的には、費用の発生元となる特定の事業、サービス、ユニット、リーダーなどを把握する能力です。リソースのラベル付けを行わなければ、特定のもの / サービスに対する支出がいくらなのかを読み解くことは非常に困難です。Compute Engine に \$36,000 を支出したと言うよりも、先月は 400,000 人のユーザーにミームを供給するために \$36,000 を支出したと言える方が望ましいでしょう。この 2 番目の報告の方が、最初の報告よりはるかに本質を突いています。標準化したラベルの作成をエンジニアリングおよび財務チームとともに実施し、できるだけ多くのリソースにラベルを使用することを強く推奨します。

最高の結果を得るための継続的なレビュー

一般的な方法としては、適切なチームと定期的に会い、使用傾向を確認し、必要に応じて予測の調整を行うことが重要です。Cloud Billing Console を使用すれば、クラウド費用のレビューおよび監査を定期的かつ簡単に行えます。また、カスタム ダッシュボードでは、支出の状況をより詳細に確認できます。定期的なレビューや適切なユニット エコノミクスの使用、および支出の可視化を実施しなければ、請求が急増したときに後手の対策を行うことしかできません。



Google データポータルでの一定期間の利用額情報の可視化

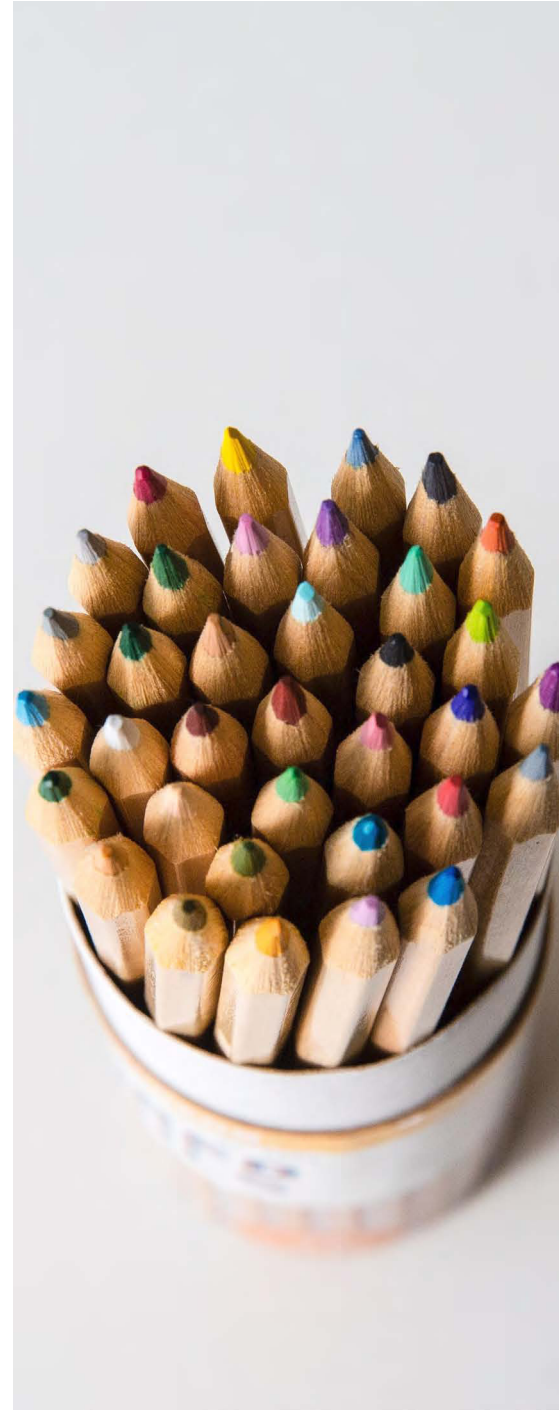
安定したお客様の場合、ビジネス戦略の調整を行う機会は Google Cloud の新しい機能とお客様のプロダクト ロードマップにおけるビジネスの変化との兼ね合いなどに依存するため、支出の見直しを行う頻度は少なめに設定できます。一方で、多くの新しいアプリケーションをデプロイしており、毎月数百万ドルを支出するようなお客様の場合、より頻繁な費用のレビューを実施するためのわずかな投資で、大幅な費用削減を短期間で実現できます。一部のケースでは、予測の対応と調整を毎日行うような、先進的なお客様もいます。毎月数百万ドルを支出しているような場合、全体的な請求の割合が少し変化するだけで、新技術の試用やエンジニアの追加雇用などを行える費用が消えてしまう可能性があります。

運用を効率化し、クラウドの価値を最大限に引き出すためには、多種多様なバックグラウンドを持つ複数のチームが協働して、特定のビジネスニーズに対応したシステム設計を行う必要があります。ベスト プラクティスとしては、クラウドにおけるビルドと支出のスピードをもとに見直しの頻度を設定する、などが挙げられます。「鉄の三角形 (Iron Triangle)」は、費用、スピード、品質の 3 つの要素を査定するために使用される一般的なフレームワークです。チームと協働して、ビジネスに有効である合意に基づいたフレームワークを設定できます。そこから、出費を抑えることも、投資を増やすことも可能です。

費用最適化のツール

クラウドで費用を最適化する方法をしっかりと把握できた後は、自由に使用できる多種多様なツールについて検討しましょう。高いレベルでは、Google Cloud の費用の管理に必要なツールは 3 つの種類に大きく分けられます。

1. **費用の把握** - これには、支出の詳細や特定のサービスに対する請求を把握すること、どのようなプロセスと目的で特定の金額を支出してビジネスの成果を挙げたのかを確認できる機能が含まれます。ここでは、アカウントビリティの共有や、頻繁的な費用の見直し、傾向の分析、アクションがもたらす影響をほぼリアルタイムに可視化するなどの機能が重要となります。[リソースの組織化](#)のための標準化された戦略を使用すれば、費用を正確に組織の運用構造にマッピングして、チャージバックやショーバックのモデルを作成できます。また、予算アラートや割り当てなどの費用管理を使用して、費用の抑制を経時的に行えます。
2. **リソース使用の最適化** - 使用方法を最適化することで環境における無駄を削減します。目標は、環境内の費用とパフォーマンスの適切な交差を描く特定の基準セットを実装することです。これは、リソースの使用を見直すための一つのレンズであり、アイドル状態のリソースがないか、アプリケーションをデプロイすべきより良いサービスはあるか、はたまたカスタム VM 構成を採用した方が適切であるかどうか、などを吟味します。無駄を削減できている企業の多くが、リソース使用の最適化を分散的な方法で実行しています。これは、たいていの場合、それぞれのアプリケーションの所有者はそのワークロードに深く精通しているためリソースのシャットダウンやサイズ変更最適に最適な人材だからです。また、[Recommender](#) を使用することで、VM インスタンスのプロビジョニングが過少または過剰である場合や、リソースがアイドル状態である場合などの問題を検出できます。このような最適化案がチームで自然に出てくるような環境を作ることが、多大な労力をかけて最適化に取り組むうえで目指すべきことです。



3. **価格効率** - これには、特定のサービスにかかる料金を最適化できる継続利用割引、確約利用割引、定額料金、秒単位の課金制、その他使用量に応じた割引などの機能が含まれます。これらの機能は企業内の一元化されたチームにとって最適です。たとえば、Cloud Center of Excellence (CCoE) や FinOps チームなどはすべての事業部門を対象範囲とする使用量に応じた割引により料金の最適化を実践しながら、無駄が発生するリスクを減らします。これについては、クラウドへの移行前だけでなく導入後の定期的な見直しが必要です。

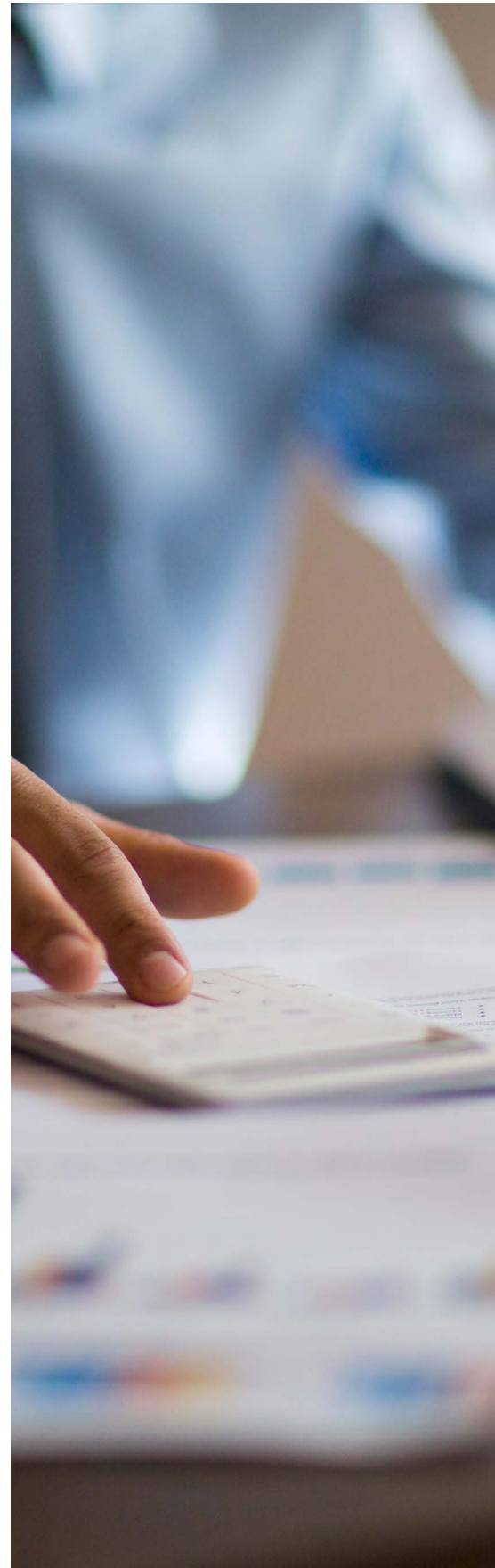
人とプロセスの両方を考慮することで、規定した基準が有用かつビジネスのニーズに整合していることを確実にできるでしょう。同じように、Google Cloud における費用の把握、リソース使用の最適化、価格効率を理解することで、お客様の技術とチーム全体にわたり費用を最適化できるツールを活用できます。

最適化案の優先順位のつけ方

競合するイニチアチブが多くある中で、どの費用最適化案を優先させるかを判断し、見直しを実施する時間を常に確保することは困難です。エンジニアリング作業量と潜在的な費用削減を把握することで、この優先順位づけが容易になります。お客様の中には、何年にもわたって技術革新と移行のスピードのみに重点を置き、長期間にわたり最適化にとって悪い習慣が重なったことから、相当な無駄が発生しているケースもあります。本来ならば、この資金は新たな機能の開発や、インフラストラクチャの追加購入、開発速度向上を目指したエンジニアの雇用などに充当できたはずで、費用と開発速度のバランスを見極め、片方に過度な重きを置いた場合の影響を把握しておくことが重要です。

お客様の費用最適化案の優先順位付けを支援するために、お客様に提示するすべての最適化案に以下の2つの特徴を意味するタグをつけています。

- **実装努力:** お客様がリソースを整理し費用の最適化案を実装するために必要な作業の推定レベル (週単位)。
- **節約量:** お客様が費用の最適化案を実装したことにより実現する可能性のある潜在的な節約量 (サービスあたりの割合)。



実装努力	期間	節約量	サービスあたりの割合
低	2 週間未満	低	0～10%
中	2～6 週間	中	10～20%
高	6 週間を超過	高	20% 超え

テストを実施する前に、実装した費用削減策によってどれだけの節約が可能であるかを正確に見積もることが常にできるとは限りませんが、知識に基づいて対策の結果を予想してみることは重要です。たとえば、ある変更を加えることでプロジェクト X での Cloud Storage の使用料金が 60% 削減できることがわかれば、優先順位付けの土台作りやチーム内でエンジニアリングの優先事項を決定するのに十分に役立ちます。場合によっては、実際の節約量を推定できることもあります。特に購入オプションにおいては、FinOps チームが、特定規模のインフラストラクチャに適用される確約利用割引などを活用して潜在的な節約量を推定できます。このような手法を実行することで、エンジニアリングの方向性についてチームが十分な情報に基づいた決定ができ、サービス向上にむけてエネルギーを集中させることができるでしょう。

第 2 章

コンピューティング費用の最適化

イントロダクションで説明した 3 つの費用最適化の原則 - 費用の把握、リソース使用の最適化、価格効率 - が適用できるのは、クラウド インフラストラクチャのさまざまな部分における基準を評価、設定するときです。まず初めに、クラウド インフラストラクチャを支えるコンピューティングから説明します。

Google Cloud に移行するお客様にとっての最初のステップは、たいていの場合 [Compute Engine](#) の導入です。これにより、高いコンピューティング処理能力が備わったクラウドの仮想マシン (VM) を簡単に調達、設定できます。Compute Engine は 2012 年にリリースされ、複数のマシンタイプと、多くの革新的な機能を提供しており、23 の地域と 70 のゾーンで利用できます。

Compute Engine の事前定義のマシンタイプとカスタム マシンタイプがあることにより、お客様のオンプレミス インフラストラクチャに最も近い VM を簡単に選ぶことができ、ワークロードの移行プロセスを費用効率よく加速化できます。Cloud では従量課金制のメリットが得られます。また、[継続利用割引](#)を使用して頻繁にコンピューティングを活用することにより大幅な費用の削減が実現できます。以下では、Google が企業のお客様と協働して月々の費用を分析し、最適化の機会を模索した経験をもとに作成した最適化案を紹介します。

費用の削減を目指すにあたって

費用の最適化を目指すにあたり、[VM インスタンスの料金体系](#)について理解を深めることが大切です。Compute Engine の課金モデルとリソースベースの料金設定について詳しく把握しておきたい場合はこのページをご参照ください。費用の把握を深めるプロセスにおいて大いに役立つでしょう。



Compute Engine の費用を把握するために重要な次のステップは、Google Cloud Console にある Cloud Billing レポートを活用することです。プロジェクト、ラベルなどのフィルタリングやグループ化を行うことでデータ表示をカスタマイズできます。ここでは、Compute Engine の複数あるマシンタイプや、確約利用割引、使用状況の確認方法などについて主に説明します。さらに高度なデータ表示が必要な場合は、より細やかな分析が可能な [BigQuery に Compute Engine の使用状況の詳細をエクスポート](#) します。こうすることで、データストアにクエリを実行して、vCPU の使用傾向や回収可能な vCPU の数が把握できます。プロジェクトあたりのコア数に対してしきい値を定義しておけば、使用傾向によって異常値の特定ができ、それに応じて事前の対策が容易になります。この対策には、VM のサイズ適正化やアイドル状態の VM の回収などがあります。

明確な費用の把握ができるようになったら、Compute Engine リソースを最適化するための 5 つの方法について見ていきます。これを実施することで、即時のメリットがもたらされるでしょう。

1. コンピューティング料金は必要な分だけ支払う

実装努力 ●●● 節約量 ●●●

アイドル状態の VM (とディスク) を特定する: Google Cloud の請求を減らす最も簡単な方法は、使用していないリソースを削除することです。優先順位が低くなった概念実証プロジェクトや、削除しようとも思っていないゾンビ インスタンスの存在について考えてみてください。Google Cloud では、このようなリソースの最適化に役立つ Recommender が提供されています。これには、非アクティブな VM と永続ディスクを使用率の指標に基づいて特定する、アイドル VM Recommender があります。

ただし、VM の削除は必ず慎重に行いましょう。リソースを削除する前に、削除したことによって起こりうる影響や必要になった際の再作成の方法などを確認してください。インスタンスを削除すると、基盤となるディスクとそのすべてのデータが削除されます。ベスト プラクティスの 1 つは、削除する前にインスタンスのスナップショットをとっておくことです。代替案として、VM を停止しても良いでしょう。これによって、インスタンスは終了しますが、ディスクや IP アドレスなどのリソースは接続解除または削除しない限り残ります。

Reduce VM resource cost Switch VM resources with low CPU or memory usage to a recommended machine type. Rightsize to save \$31.67/month + 239 more	Cost savings \$14,255.65/month estimate
→ View all	

Unused Compute Engine resources Back up and delete unused resources to reduce costs. Shut down VM to save \$204.95/month + 238 more	Cost savings \$23,023.17/month estimate
→ View all	

Unused Compute Engine resources HISTORY Recommendation Hub

 **Cost savings**
 Back up and delete unused resources to reduce costs. [Learn more](#)

Impact
\$23,023.17/month

Recommendations DISMISS

Filter table

Recommendation ↓	Resource name	Recommended action	Location	Refreshed
☐ Shut down VM to save \$324.62/month	n1-highcpu-8-idling-southamerica-east1-c	Shut down	southamerica-east1-c	Apr 22, 2020, 2:43:17 AM
☐ Shut down VM to save \$324.45/month	n1-highcpu-8-idling-southamerica-east1-b	Shut down	southamerica-east1-b	Apr 22, 2020, 2:13:53 AM
☐ Shut down VM to save \$319.13/month	n1-highcpu-8-idling-southamerica-east1-a	Shut down	southamerica-east1-a	Apr 22, 2020, 2:39:11 AM
☐ Shut down VM to save \$292.12/month	n1-highcpu-8-idling-australia-southeast1-a	Shut down	australia-southeast1-a	Apr 22, 2020, 2:52:44 AM
☐ Shut down VM to save \$292.12/month	n1-highcpu-8-idling-australia-southeast1-b	Shut down	australia-southeast1-b	Apr 22, 2020, 2:54:19 AM
☐ Shut down VM to save \$291.03/month	n1-highcpu-8-idling-australia-southeast1-c	Shut down	australia-southeast1-c	Apr 22, 2020, 2:34:44 AM
☐ Shut down VM to save \$288.05/month	n1-highcpu-8-idling-europe-west6-b	Shut down	europe-west6-b	Apr 22, 2020, 2:21:44 AM
☐ Shut down VM to save \$288.05/month	n1-highcpu-8-idling-asia-east2-c	Shut down	asia-east2-c	Apr 22, 2020, 2:49:39 AM
☐ Shut down VM to save \$288.05/month	n1-highcpu-8-idling-europe-west6-a	Shut down	europe-west6-a	Apr 22, 2020, 2:45:26 AM
☐ Shut down VM to save \$287.58/month	n1-highcpu-8-idling-europe-west6-c	Shut down	europe-west6-c	Apr 22, 2020, 2:22:46 AM

Rows per page: 10 1 - 10 of 238 < >

Google Cloud の Recommender では、費用削減のためのガイダンスが用意されています。

Shut down VM to save \$324.62/month Feedback?

Recommendation

 We recommend shutting down the VM instance to save \$324.62/month

Resource name

n1-highcpu-8-idling-southamerica-east1-c

Location

southamerica-east1-c

VIEW INSTANCE

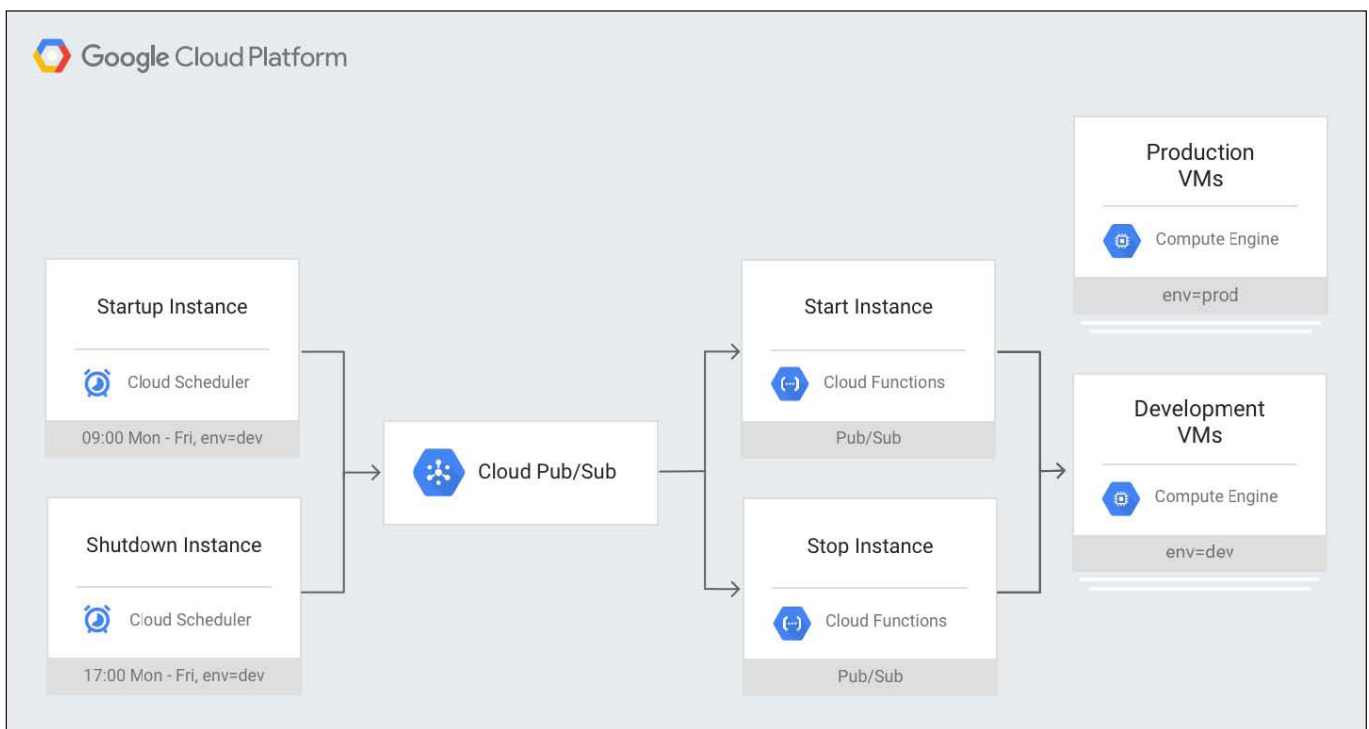
DISMISS

CANCEL

Recommender では、VM のシャットダウンに関する詳細なヒントが提供されます。

詳しくは、[Recommender のドキュメントをご覧ください](#)。また、ポートフォリオに使用量ベースの Recommender を随時追加する予定です。

VM の自動起動 / 停止をスケジュールする: [Compute Engine](#) のようなプラットフォームを使用するメリットは、使用したリソース分のみの支払いが可能なおことです。本番環境では 24 時間 365 日の稼働がほとんどですが、開発、テスト、個人用途といった環境の VM は営業時間内のみ使用することが一般的です。そのため、VM を停止することで費用を大幅に削減できます。たとえば、VM が 1 日 10 時間、月曜日から金曜日まで稼働するとして、常時稼働と比較して、毎月の費用を 75% 減少させることが可能です。



クラウド インフラストラクチャ全体で VM がどのようにセットアップ、サイズ設定されているか調査し、費用削減のチャンスを見つけることは非常に有用です。

VM のサイズ適正化: Google Cloud では、お客様のニーズに合わせて CPU と RAM の量が適切なカスタム マシンタイプを作成するだけでも大幅に費用削減が可能です。しかし、ワークロードの要件は時間とともに変わる可能性があります。一度最適化したインスタンスの対応ユーザー数とトラフィック量が減少することもあるかもしれません。このため、[サイズ適正化の推奨案](#)では、vCPU と RAM の使用状況の変化に基づいてマシンタイプを効率的にダウンサイジングする方法について紹介しています。インスタンスのマシンタイプ (または[マネージド インスタンス グループ](#)) におけるサイズ適正化の推奨案は、過去 8 日間に Cloud Monitoring で収集したシステム指標を使用して生成されます。

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

START

STOP

RESET

DELETE

393 instances could be resized to save you up to an estimated \$11,225 per month and increase performance. [Learn more](#)

Filter VM instances

Columns

☐ Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
☐ ☒ n1-highcpu-8-idling-europe-west5-a	europe-west5-a	Save \$133 / mo		10.0.0.4 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-europe-west5-b	europe-west5-b	Save \$133 / mo		10.0.0.35 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-europe-west5-c	europe-west5-c	Save \$133 / mo		10.0.0.68 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-europe-west6-a	europe-west6-a	Save \$144 / mo		10.0.0.115 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-europe-west6-b	europe-west6-b	Save \$144 / mo		10.0.0.89 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-europe-west6-c	europe-west6-c	Save \$144 / mo		10.0.0.82 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-northamerica-northeast1-a	northamerica-northeast1-a	Save \$113 / mo		10.0.0.16 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-northamerica-northeast1-b	northamerica-northeast1-b	Save \$112 / mo		10.0.0.131 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-northamerica-northeast1-c	northamerica-northeast1-c	Save \$113 / mo		10.0.0.84 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-southamerica-east1-a	southamerica-east1-a	Save \$160 / mo		10.0.0.29 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-southamerica-east1-b	southamerica-east1-b	Save \$162 / mo		10.0.0.35 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-southamerica-east1-c	southamerica-east1-c	Save \$162 / mo		10.0.0.83 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central1-a	us-central1-a	Save \$102 / mo		10.0.0.156 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central1-b	us-central1-b	Save \$103 / mo		10.0.0.155 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central1-c	us-central1-c	Save \$103 / mo		10.0.0.31 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central1-d	us-central1-d			10.0.0.38 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central1-f	us-central1-f	Save \$103 / mo		10.0.0.147 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central2-a	us-central2-a	Save \$103 / mo		10.0.0.71 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central2-b	us-central2-b	Save \$103 / mo		10.0.0.119 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central2-c	us-central2-c	Save \$103 / mo		10.0.0.55 (nic0)	None	SSH ▾ ⋮
☐ ☒ n1-highcpu-8-idling-us-central2-d	us-central2-d	Save \$102 / mo		10.0.0.31 (nic0)	None	SSH ▾ ⋮

クラウド VM のサイズ適正化に関する提案をご確認ください。

組織が環境を管理するコードとしてインフラストラクチャを使用している場合は、VM のサイズ適正化を大規模に展開する方法を説明する[こちらのガイド](#)を確認してください。

2.コミットメントを購入する

実装努力 ●●● 節約量 ●●●

価格効率は、クラウド最適化に関する取り組みの一環として採用すべき、もう 1 つの重要な概念です。コミットメントを購入し、関連する割引を受ける機能は、クラウド費用の真の最適化を大きく前進させます。

Google のお客様は、さまざまな可用性要件に基づき、多様なワークロードを Google Cloud 上で実行しています。お客様は、VM フリートの管理に関して 70/30 のルールに従っています。年間を通じた使用率は約 70% であり、休日や特別なイベント時には約 30% の季節的なバーストがあります。

お客様がこれに当てはまる場合、おそらくピーク時の容量に合わせてリソースをプロビジョニングしているはずです。しかし、Google Cloud 移行後は、使用量を基準として、コンピューティングワークロードの割引をさらに活用できます。1 年または 3 年のコミットメントを購入することで VM の使用に対する大幅な割引を得られるため、確約利用割引は、予測可能な定常状態のワークロードがある場合に理想的です。最近リリースされた Cloud Console の [確約利用割引分析レポート](#) では、購入したコミットメントの有効性を理解、分析し、過去のデータに基づいたリソースフロアの状況予測に役立ちます。

October 1 – December 31, 2019 commitments and average usage

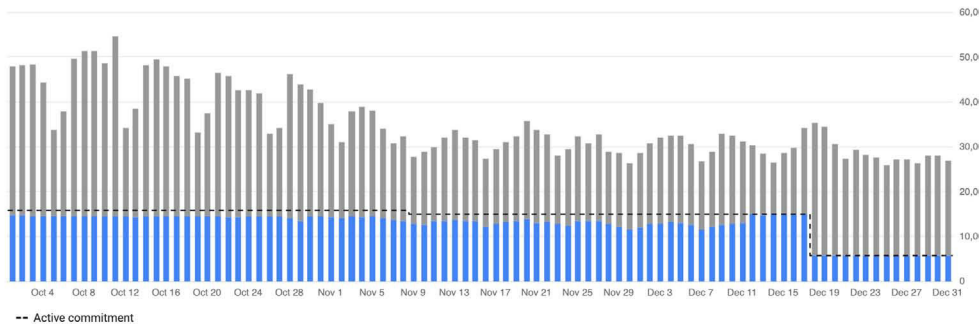
Commitment scope: Billing account EDIT

< **VCPU** RAM COMPUTE-OPTIMIZED VCPU COMPUTE-OPTIMIZED RAM N2 VCPU N2 RAM N2D VCPU N2D RAM >

Region
All regions
with vCPU commitments

Active commitments
5,702 vCPU
decreased from 14,977 on 12/18/19

Commitment utilization
90%



Usage	Average usage per microsecond	Highlights
All usage	35,046.36 vCPU	View usage costs
Commitment covered usage	12,507.08 vCPU	36% of all usage
Eligible on-demand usage	22,539.27 vCPU	64% of all usage

View

- ☒ Aggregate
☐ By region

Scope

Time range

Custom range

Usage data is available since January 2017

From 10/1/19

To 12/31/19

Include data from regions with

- ☐ vCPU commitments
☒ Any vCPU usage

Filter

Region
All regions (16)

Projects
All projects (131)

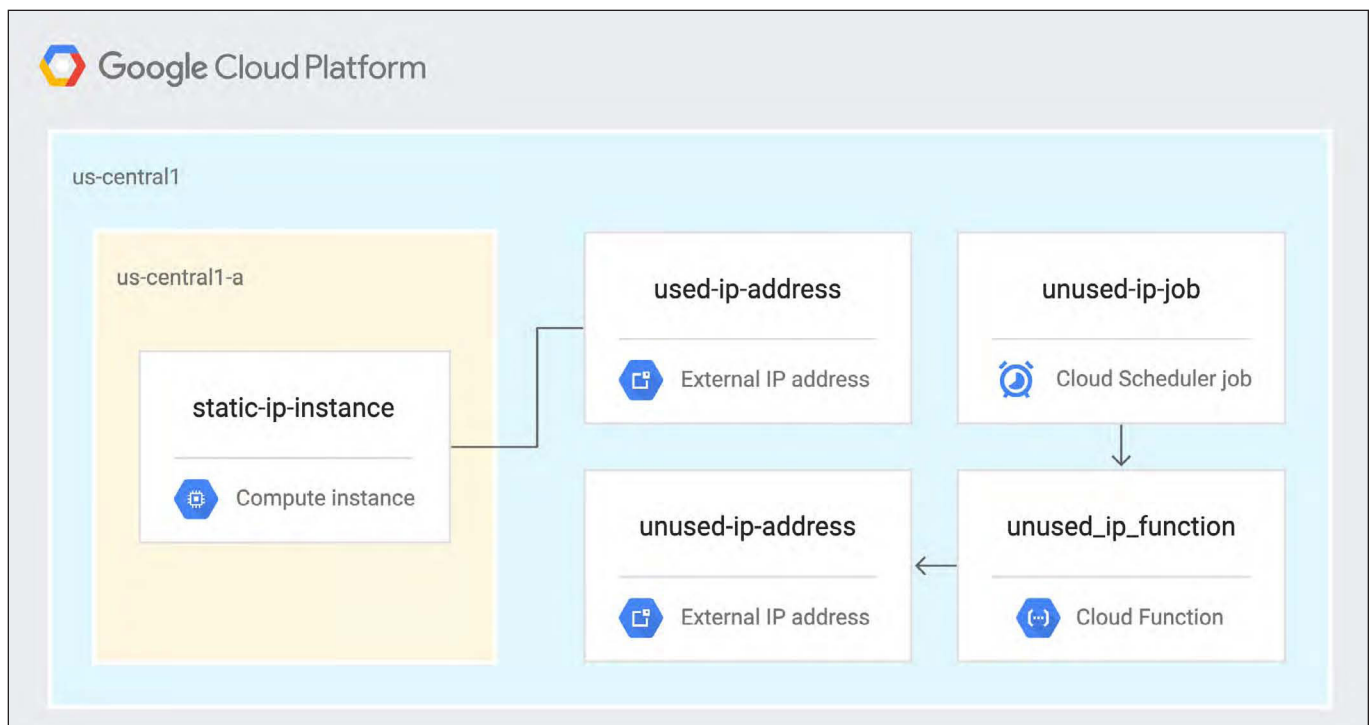
確約利用割引レポートには、費用削減の可能性が示されます。

このレポートに加えて、社内チームで使用状況について話し合い、確約利用割引がワークロードに対して意味があるかどうかを把握します。社内チームと積極的に協力することで、確約利用割引の適用範囲を拡大し、費用を最大限に削減できます。

3.費用の最適化を自動化する

実装努力 ●●● 節約量 ●●●

チームが常に費用最適化のベスト プラクティスに従うようにするためには、自動化を行い、手動作業を減らすのが一番です。自動化は、ラベルを使用することで大幅に簡素化されます。ラベルとは、さまざまな Google Cloud サービスに適用される Key-Value のペアです。たとえば、営業時間中に開発者のみが使用するインスタンスに、「env: development」のラベルを付けたとします。その後、Cloud Scheduler を使用してサーバーレスの Cloud Function をスケジュールすることで、週末または営業時間後にはシャットダウンし、必要に応じて再起動するようにできます。このアーキテクチャ図とコードサンプルを使用すれば、これをお客様自身で行えます。これは、リソースの使用を最適化する能力を大きく飛躍させてくれます。



ラベルを使用することで、費用の最適化を自動化できます。

Cloud Functions を使用して他の Compute Engine リソースのクリーンアップを自動化すると、エンジニアリングにかかる時間と費用も削減できます。たとえば、お客様は接続されていない（孤立した）永続ディスクや未使用の IP アドレスについて忘れてしまうことがよくあります。仮想マシン インスタンスに接続されていない場合、これらには費用が発生します。削除ルールのオプションが「ディスクを保持する」に設定されている VM の場合、VM が削除された後も永続ディスクが保持されます。これは、後でそのディスクにデータを保存する必要がある場合には有効です。しかし、孤立した永続ディスクはすぐに増えてしまいます。この Google Cloud ソリューションの記事では、Cloud Functions、Cloud Scheduler、Cloud Monitoring を使用してこれらの孤立したディスクを自動的に検索し、それらのスナップショットを作成してから削除するためのアーキテクチャとサンプルコードについて説明します。このソリューションは、未使用の IP アドレスのクリーンアップやアイドル状態の VM の停止など、他の費用自動化のブループリントとして使用できます。

4. プリエンプティブル VM を使用する

実装努力 ●●● 節約量 ●●●

HPC、ビッグデータ、メディアコードの変換、CI / CD パイプライン、ステートレス ウェブ アプリケーションといったフォールトトレラントなワークロードがある場合、プリエンプティブル VM を使用してそれらをバッチ処理すると、リソース使用量の最適化と価格効率化戦略の両方を使用して大幅な費用の削減を実現できます。たとえば、Google のお客様である Descartes Labs は、プリエンプティブル VM を使用して衛星画像を処理し、企業や政府が世界の食料供給を予測できるようにすることで、分析費用を 70% 以上削減しました。

プリエンプティブル VM は有効期間が短いため、実行されるのは最大 24 時間です。24 時間が経過する前にシャットダウンされる場合もあります。VM を再利用する必要がある場合、30 秒のプリエンプション通知がインスタンスに送信されます。シャットダウンスクリプトを使用して、その 30 秒の間にクリーンアップを行えます。プリエンプティブル VM のワークロードでの使用を検討する場合は、必ず[規定のリスト全体](#)を十分に確認してください。すべてのマシンタイプが、プリエンプティブル VM として使用できます。gcloud コマンドラインに「preemptible」を追加するか、Cloud Console からオプションを選択するだけで、起動できます。

プリエンプティブル VM のアーキテクチャでの使用は、割引料金でコンピューティングのスケールリングを行う優れた方法ですが、VM を再利用する必要がある場合は、ワークロードが潜在的な中断を処理できることを確認する必要があります。これに対応する方法の 1 つは、アプリケーションがデータを処理するときにチェックポイントを設定することです。つまり、Google Cloud Storage やデータベースなど、VM 自体の外部のストレージに書き込むようにします。例として、この[シャットダウン スクリプトを使用するサンプルコード](#)を試用し、チェックポイント ファイルを Cloud Storage バケットに書き込みます。ロードバランサの背後にあるウェブ アプリケーションの場合は、30 秒のプリエンプション通知を使用してその VM への接続をドレインし、トラフィックを別の VM にシフトできるようにすることを検討してください。複数の VM を一緒に起動した場合にそれらが同時にシャットダウンすることを避けるため、24 時間の期間が終了する前に、プリエンプティブル VM のローリング ベースでのシャットダウンを自動化することを選択するお客様もいます。

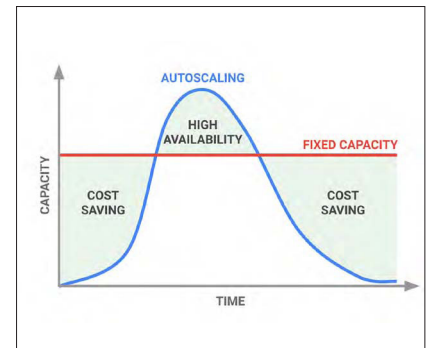
プリエンプティブル VM を Compute Engine 環境に制限する必要はありません。Dataproc の GPU、GKE クラスタ、セカンダリ インスタンスもプリエンプティブル VM を使用できます。また、[フレキシブル リソース スケジューリング](#)を使用して通常のインスタンスにプリエンプティブル VM を追加することで、Cloud Dataflow バッチ分析の費用を削減することもできます。

5. 自動スケールリングを試す

実装努力 ●●● 節約量 ●●●

費用を削減するもう 1 つの優れた方法は、必要なときに必要な容量だけを実行することです。これは、リソース使用を最適化するための重要な原則です。前述のように、通常、定常状態での使用には容量の約 70% が必要ですが、追加の容量が必要になった場合に、追加分がすぐに利用可能な状態であることが重要です。

オンプレミス環境では、追加の容量を事前に購入して、提供されるまで待ってからラックマウントし、スタックする必要があります。クラウドでは、自動スケーリングを使用して、必要な場合にのみ容量を自動的に増やすことができます。Compute Engine マネージド インスタンス グループは、Google Cloud でこの自動スケーリング機能を提供します。トラフィックの増加を処理するために適切にスケールアップし、インスタンスの必要性が低下したときには、自動的に再びスケールダウンできます（ダウンスケーリング）。CPU 使用率、HTTP 負荷分散容量、または Cloud Monitoring 指標に基づいてスケーリングを行えます。これにより、アプリケーションにとって何が最も重要かに基づいてスケールする柔軟性が得られます。



高価なだけがコンピューティングではない

前述したように、Compute Engine 費用の最適化には、さまざまな方法があります。最適なオプションを最初に理解し、時間をかけてベースライン費用をモデル化するための鍵となるのは、環境をモニタリングして、使用パターンを理解することです。ワークロードと現在の運用モデルに応じて、実装する戦略は多種多様です。



第 3 章

ストレージ費用の最適化

数十億ドル規模の複合企業の社員が前期の売上データを確認する場合でも、ピアノを弾いている猫の動画をアップロードする場合でも、必要なはそのデータを保存する場所です。

Google Cloud のお客様にとっては、通常、統合オブジェクト ストアである Cloud Storage がその場所となります。Cloud Storage は、さまざまなサービスと統合できる堅牢な API を備えています。この API にはまた、使用の最適化、費用の把握、価格効率といった費用最適化の原則に従うために役立つ機能も含まれています。オブジェクトをクラウドに保存すること自体は簡単な作業ですが、現在の状況に対して最も適切なアプローチを確立するには、もう少し先を見通す必要があります。

スケーラブルかつ無制限のストレージ サービスを利用する利点の 1 つは、無限にスケールできる屋根裏部屋を手に入れたのと同じことで、ボックスやアイテム（またはバケットやオブジェクト）を深く考える必要なく簡単に持ち続けられるということです。しかし、これはあまり良い考えとは言えません。時間の経過とともに、これらのアイテムの保管には費用がかかります。ビジネス目的としてそれらは必要なのか。それとも、お気に入りの木製のヌンチャクのように、いつか役に立つかもしれないという万一の可能性に期待してアイテムを囲い込んでいるだけなのか。最初のステップは、ビジネスに対するオブジェクトまたはバケットの有用性を識別する方法を構築することです。それでは、ほうきとちりとりを持って、掃除を始めましょう。

クラウド移行時にストレージをクリーンアップする

ストレージ費用の最適化を検討する際には、考慮すべき要素がいくつかあります。ここで秘訣となるのは、パフォーマンスに悪影響を及ぼさないようにすること、そしてコンプライアンス、法務、または単純にビジネス上の価値の観点から、今後の目的のために保持する必要があるものを捨てないようにすることです。データがビジネスにおける最も重要な商品として浮上している中、長期的な分析のためだけでなく、短期的にも適切なストレージ クラスを使用する必要があります。Cloud Storage では、多様な費用、耐久性、復元性を備えた、さまざまなストレージ クラスをご用意しています。

スケーラブルかつ無制限のストレージ サービスを使用するということは、バケットやオブジェクトを簡単に持ち続けられることを意味しますが、これはあまり良い考えとは言えません。

クラウド アーキテクチャに関して言えば、万能の解決策というものはほとんどありません。ただし、お客様と一緒に取り組む中で、何度も浮かび上がってくるいくつかのテーマに気づきました。これらの教訓は、画像の保存から高度な機械学習モデルの構築まで、あらゆる環境に適用できます。

自然な出発点はまず、Cloud Storage を利用する場合に「何にお金がかかるか」を理解することです。[料金ページ](#)は非常に便利ですが、Cloud Storage の使用状況を分析する場合には、次のことも考慮する必要があります。

1. 保持
2. アクセス パターン
3. パフォーマンス

費用に影響するその他のユースケースにはさまざまな可能性があります。まずはこれらのテーマに関する推奨事項に焦点を当ててみましょう。ここからは、それぞれのテーマを詳しく説明していきます。

保持に関する考慮事項とヒント

実装努力 ●●● 節約量 ●●●

データの種類の検討する際にまず考慮すべきことは、その保持期間です。適切な[ライフサイクル ポリシー](#)を決定するには、「なぜこのオブジェクトに価値があるのか」、「どれくらいの期間、価値があるのか」を考えることが重要です。ライフサイクル ポリシーを設定すると、特定のオブジェクトまたはバケットにタグを付け、[一連の条件](#)に基づいて特定のオブジェクトまたはバケットタイプのストレージ クラスを削除または変換する自動ルールを作成できます。これは、屋根裏部屋を体系的に整頓し、清掃してくれる、執事のようなものだと考えてください。ただし、この執事はお金をかけずに、むしろこういったオペレーションにかかるお金を節約してくれます。

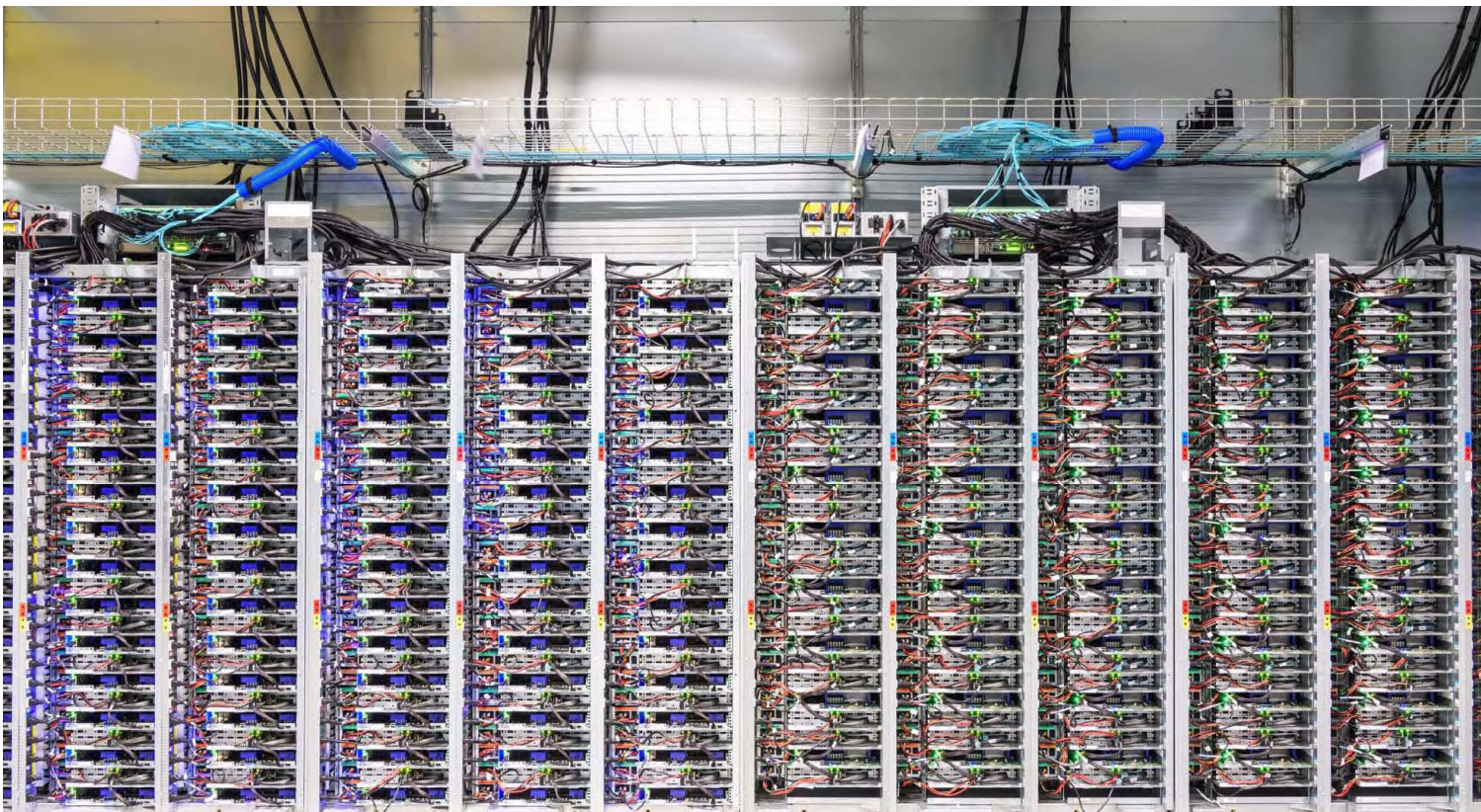
クラウド アーキテクチャに関して言えば、万能の解決策というものはほとんどありません。

データの種類の検討する際にまず考慮すべきことは、その保持期間です。

さまざまな方法でライフサイクル ポリシーを利用したお客様が、大きな成功を収めています。法的情報開示のコンプライアンスでは、特に力を発揮します。業界やデータタイプによっては、保持するデータタイプと保持期間を規制する法律が存在します。Cloud Storage ライフサイクル ポリシーを使用すると、法令遵守の必要性の最小しきい値に達したオブジェクトをすぐに削除対象としてタグ付けできます。これにより、必要以上に長く保持することで料金がかかることを防ぎ、またどのデータがいつ期限切れになるのかを覚えておく必要もなくなります。

Cloud Storage 内で、ポリシーを設定してストレージ タイプを別のクラスに変換することもできます。これは、短期間においては比較的頻繁にアクセスするけれど、長期間の頻繁なアクセスは必要ないデータで特に役立ちます。これら特定のオブジェクトを、法的またはセキュリティ上の目的で、あるいは通常の長期的なビジネス価値のために、長期間保持したい場合もあるでしょう。これを実践するのにぴったりなのが、ラボ環境です。テストが完了したら、近い将来に結果を詳細に分析する必要があります。しかし、長期的に見れば、そのデータに頻繁にアクセスすることはありません。このストレージを 1 か月後に Nearline または Coldline Storage クラスに変換するようポリシーを設定しておけば、長期的なデータ費用を削減できます。

さまざまな方法でライフサイクル ポリシーを利用したお客様が、大きな成功を収めています。





Add object lifecycle rule

gcp-cost-optimization-idle-bucket

After you add or edit a rule, it may take up to 24 hours to take effect.


Select object conditions


The action will be triggered when all selected conditions are met.

☒ **Age**
 Time elapsed since objects uploaded to current bucket.

☐ **Creation date**

☒ **Storage class**
 All objects with any of the selected storage classes.

☒ **Regional**
☒ **Standard**
☐ **Durable Reduced Availability**
☐ **Nearline**
☐ **Coldline**

☒ **Newer versions**
 Applies only to versioned objects. All objects with at least this many newer versions.

☐ **Live state**

Continue


Select action


☒ **Set to Nearline**
☐ **Set to Coldline**
☐ **Set to Archive**
☐ **Delete**

Continue

Save

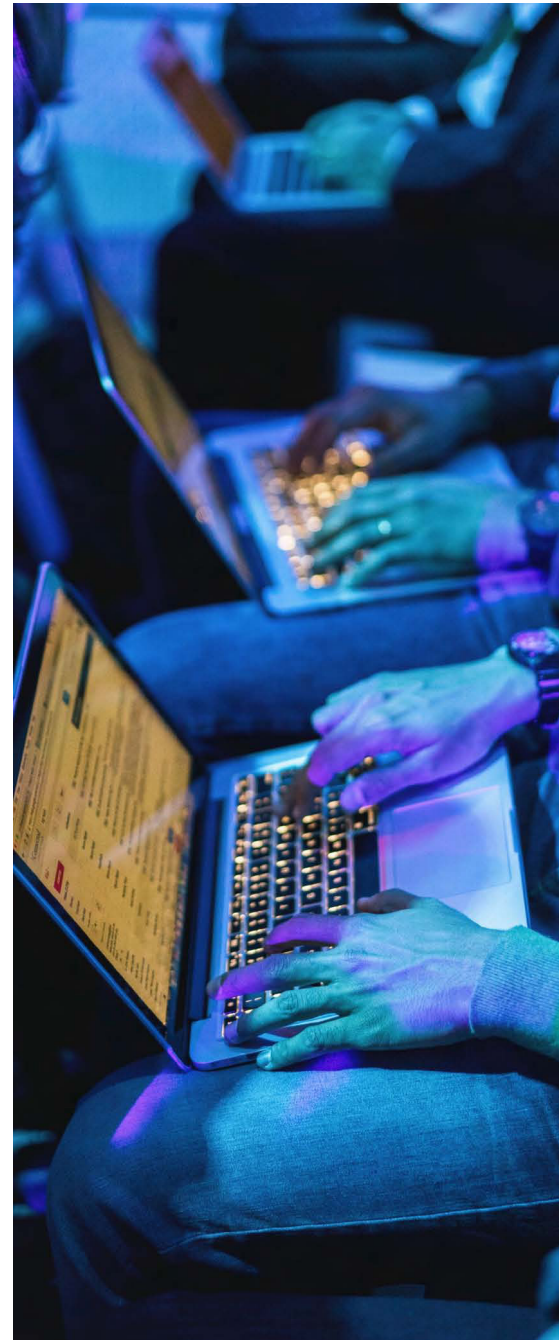
Cancel

データを自動的に移動するよう、Cloud Storage ポリシーを設定します。

ストレージ環境における、よくある無駄の原因のもう 1 つが、データの重複です。もちろん、重複が必要な場合もあります。たとえば、現地のチームがすばやくアクセスできるよう、データセットを複数のリージョンに複製したい場合もあります。しかし、お客様とともに取り組んできた経験から言うと、重複データの多くは緩いバージョン管理の結果であることがほとんどです。こうした重複データは管理が面倒で、費用がかかってしまいます。

幸い、データの重複を防ぐ方法や、データが誤って削除されるのを防ぐツールが数多くあります。次のことを考慮してください。

- 信頼できる唯一の情報源で復元力を維持しようとしている場合、さまざまなバケットに複数のコピーを作成するのではなく、[マルチリージョン バケット](#)を使用する方が理にかなっている可能性があります。この機能を使用すると、保存されたオブジェクトに対して地理的な冗長性を有効にできます。これにより、データが 2 か所以上で非同期に複製され、自然災害が発生した場合の地域的な障害から保護されます。
- 重複データの多くは、Cloud Storage オブジェクトのバージョンング機能を適切に使用していないことが原因で作成されます。オブジェクトのバージョンングにより、データの上書きや誤った削除を防ぐことができますが、作成される重複データが増加します。そのデータは本当に、コピーが 5 つ必要でしょうか？保護されていれば、1 つでも十分でしょう。ロールバックできなくなるのが心配ですか？オブジェクトのバージョンング ポリシーを設定すれば、適切な数のコピーを確保できます。それでも、誤って何かを失うことが心配ですか？特定の日時より前にアイテムが削除されないようにする、[バケットロック](#)機能の使用を検討してください。この機能は、いくつかの重要な規制へのコンプライアンスを実施するのに非常に役立ちます。オブジェクトのバージョンングを使用する場合に、不必要にスペースを無駄にすることなくデータを安全に保つために利用できる、いくつかの機能をご用意しています。



アクセス パターンに関する考慮事項とヒント

実装努力 ●●● 節約量 ●●●

Cloud Storage は、異なる料金設定と、それぞれに最適なユースケースを持つ、さまざまなストレージ クラス (Standard、Nearline、Coldline、Archival) を提供しています。現在 Standard クラスのみを使用しているなら、ワークロードを見直して、データへのアクセス頻度を再評価するときかもしれません。経験から言うと、多くの企業がアーカイブ目的として Standard クラスのストレージを使用していますが、Nearline または Coldline クラスのストレージを利用することで、支出を削減できる可能性があります。また、法的情報の検索といったコールド ストレージのユースケースでオブジェクトを保持している場合、Archival クラスのストレージを使用することでさらに費用を抑えられる可能性があります。

オブジェクトをより低料金のストレージ クラスに変換する機能は強力なツールですが、使用には注意が必要です。アクセス頻度の低いオブジェクトは長期ストレージのほうが低料金で維持できますが、よりアクセス頻度の低い (コールド) ストレージのオプションに移動したデータまたはメタデータに頻繁にアクセスする必要が出てきた場合には、追加料金が発生します。特定のストレージ クラスからそのデータを削除しようとした場合もまた、費用に影響します。たとえば現状では、オブジェクトが Nearline Storage に配置されるまでに、30 日以上かかります。頻繁にデータにアクセスする必要がある場合は、代わりに Regional Storage クラスにコピーを作成することで、アクセス料金の増加を回避できます。

費用削減の機会を考える際には、そのデータが長期的にアクセスする必要があるのかどうか、また必要時にどのくらいの頻度でアクセスするのかということも考える必要があります。例えば、あなたは CFO で、クラウド費用の四半期報告書を見ているとします。その情報は 3 か月に 1 度だけ参照すればいい場合、データ抽出のための費用の増加は気にする必要はないかもしれません。Regional バケットで年間を通してストレージを維持するよりはまだ安いからです。しかし、長期的なストレージ クラスの取得費用の中には相当高価なものもあるため、決定する際には慎重にレビューする必要があります。

多くの企業が
Nearline や
Coldline クラスの
Storage を活用すること
で、出費を抑えるこ
とができています。

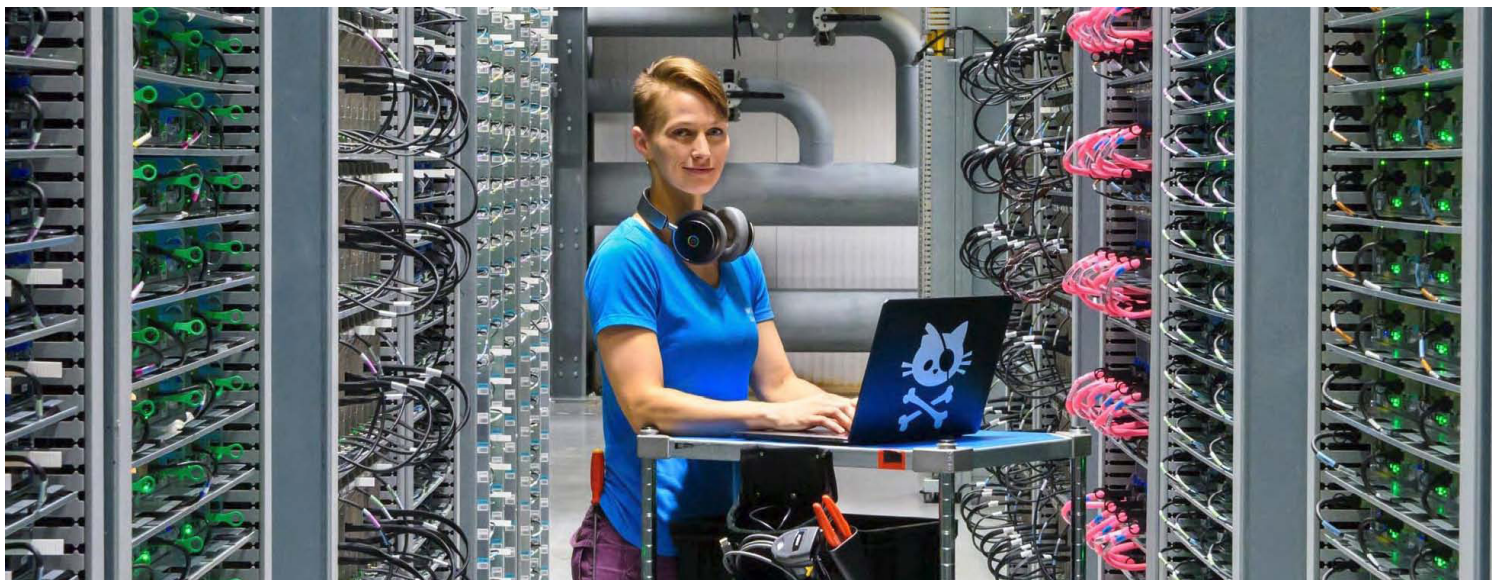
パフォーマンスに関する考慮事項とヒント

実装努力 ●●● 節約量 ●●●

「このデータはどこからアクセスされるのか」というのは、パフォーマンスを考慮しつつ特定のユースケースに最適なストレージ クラスを確立しようとしたときに考えるべき大きな問題です。局所性は、選択したストレージ ロケーションへのコンテンツのプッシュとそこからの取得の速さに直接影響します。例えば、グローバルな利用率を持つ「ホット オブジェクト」（例えば、従業員のタイムトラッキング アプリケーションのような頻繁にアクセスされるデータベース）は、マルチリージョン ロケーションでの使用に適しており、オブジェクトを複数の場所に保存できます。これは、全体的な可用性を高めるだけでなく、コンテンツとエンドユーザーの距離を近づけるものになりえます。別の例としては、ユーザーの地理的分布が広範囲にわたるゲーム アプリケーションがあります。コンテンツとユーザーの距離を近づけてより良い体験（ラグの軽減）を提供し、最後に保存したファイルが複数の場所に分散して保管されるようにすることで、地域的な障害が発生しても、苦労して稼いだ戦利品を失うことはありません。

このオプションを検討する際に留意すべき点は、[マルチリージョン ロケーションへストレージを配置する](#)ことで、パフォーマンスと可用性が改善されますが、アプリケーションの設計によっては、プレミアムが発生し、ネットワーク下り（外向き）料金が增加する可能性があるということです。アプリケーションの設計段階において、これは考慮すべき重要な要素です。パフォーマンスを考慮する場合のもう 1 つのオプションは、リージョン ロケーションにあるバケットです。エンドユーザーに比較的近いリージョンの場合には良い選択となります。データの保管場所として特定のリージョンを選択することで、そのリージョン内での冗長性が保証されます。特定の地域で作業するチームがあり、比較的高い頻度でデータセットにアクセスする場合には、このロケーション タイプが一般的に安全です。これは、ほとんどのワークロードのニーズに対応できるため、最も一般的に使用されているストレージのロケーション タイプです。アクセスが速く、リージョン内での冗長性があり、オブジェクト ストアとしては全体的に手頃な価格となっています。

バケットのような単純に聞こえるものに対して、クラウドベースのオブジェクト ストレージは、実際には膨大な可能性を提供しており、費用やパフォーマンスに与える影響もさまざまです。ご覧のように、自社のストレージのニーズを微調整し、考え抜かれた自動化によってスペースと費用を節約する方法は実にたくさんあります。Google Cloud への投資を最大限に活用していただくため、多くの機能を用意しています。



第 4 章

ネットワーク費用の最適化

すべてのクラウドのデプロイメント環境には、データを転送するネットワークが必要です。ネットワークがなければ、猫の動画を視聴することも、自撮りをアップロードすることもできません。マイクロサービス間のやり取りも不可能です。

Google Cloud では、クラウドベースのワークロードとサービスのために[スケーラブルで柔軟なグローバル ネットワーク](#)をご用意しています。そのネットワークの利用方法が、デプロイの 4 つの重要な側面、つまり、費用、セキュリティ、パフォーマンス、可用性に影響を及ぼします。信頼性と安全を備えた費用対効果の高いネットワーク アーキテクチャを設計するには、社内に複数のチームを用意して、こうした 4 つの要素に重点を置きながら、優先順位を決める必要があります。今回ご紹介するヒントでは、ネットワークソリューションの設計時に考慮すべきいくつかの検討事項を取り上げます。

社内に複数のチームを用意して、アーキテクチャの要素に重点を置きながら、優先順位を決める必要があります。

ネットワークトラフィックフローを理解する

全体的なネットワーク費用戦略の確認でまず行うのは、使用している対象、つまり、Google Cloud 環境を出入りするトラフィックを把握することです。これは、VM インスタンスとの間で送受信されるネットワークフローの記録を保持する VPC フローログを使用することで簡単に行えます。各フローログ エントリはネットワーク接続ごとに送受信された送信元 IP、送信先 IP、バイト数などの詳細を記録します。これは、ネットワークトラフィックの把握に必要な情報です。これらのログは Cloud Logging で収集され、[BigQuery にエクスポート](#)することでトレンドの可視化に役立ちます。

VPC フローログのユースケースとしては、ネットワークのモニタリング、フォレンジック、リアルタイム セキュリティ分析、そして今回の目的である費用の最適化などがあります。ネットワーク費用の最適化について言えば、VPC フローログで最も関連性の高い情報は次のとおりです。

- リージョンとゾーン間のトラフィック
- インターネットにおける特定の国へのトラフィック
- トラフィック使用量が特に多いプロセス

VPC フローログを有効にする方法について、こちらで[詳細な手順](#)をご紹介します。

「トラフィック使用量が特に多いプロセス」を見極める

実装努力 ●●● 節約量 ●●●

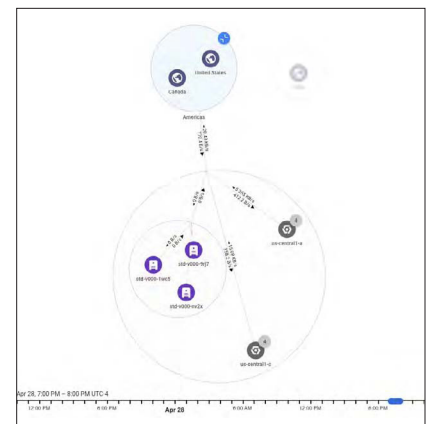
VPC フローログから得られる情報は、既存のネットワーク費用のどこを削減できるかを特定するのに役立ちます。たとえば、地理的なロケーションは、費用を最適化して設計する際に考慮すべき重要な要素です。すべてのネットワーク料金が均等に発生するわけではありません。リージョンが異なるとネットワーク費用も変わります。

また、ネットワークのレイアウトや、アプリケーションとユーザー間のトラフィックの流れを知ることも重要です。ネットワークトポロジは、Network Intelligence Center のモジュールの 1 つで、トポロジの組織全体のビューや関連するネットワーク パフォーマンス指標を含め、グローバルな Google Cloud の展開とそのパブリック インターネットとの相互作用を包括的に可視化します。これにより、非効率的なデプロイメントを見つけ出し、必要な措置を講じて、リージョンおよび大陸間のネットワーク下り(外向き)費用を最適化できます。

インターネットにコンテンツを提供するウェブサーバーのグループなど、一般的なインターネットの下り(外向き)料金の場合、サーバーが配置されているリージョンによって料金が異なることがあります。たとえば GB あたりの料金を比較すると、us-central1 は asia-southeast1 よりも低価格です。別の例として、Google Cloud のリージョン間を流れるトラフィックがありますが、これはインターネットの下りでなくても、リージョンのロケーションによって大きく変わることがあります。たとえば、asia-south1 (インド) と asia-east1 (台湾) の間でデータを同期する費用は、us-east1 (サウスカロライナ) と us-west1 (オレゴン) の間でトラフィックを同期する費用の 5 倍です。

リージョンに関する検討事項と同様、ワークロードがどのゾーンにあるかを考慮することも重要です。可用性の要件によっては、無料でゾーン内のネットワークトラフィックを使用するように設計することができかもしれませんが、この無料利用のメリットが大きいことは、誰の目にも明らかです。同じリージョンまたはゾーンに配置されているものの、パブリック外部 IP アドレスを使って通信している VM について考えてみてください。内部 IP アドレスを使用して通信するように構成することで、外部 IP アドレスを使用して通信するトラフィックに対する費用を削減できます。

すべてのネットワーク料金が均等に発生するわけではありません。リージョンが異なるとネットワーク費用も変わります。



ネットワークを使用したトポロジの例
トポロジ、Network Intelligence Center

実現できる可能性のあるネットワーク費用の削減と、シングルゾーン アーキテクチャの可用性への影響を比較検討する必要があります。シングルゾーンのみへのデプロイは、高可用性を必要とするワークロードにはおすすめしませんが、特定のサービスで同じゾーン内の virtual Private Cloud (VPC) ネットワークを使用することが適切な場合もあります。例としては、アジアのような費用が高いリージョンではシングルゾーン アプローチを使用し、費用が低い北米ではマルチゾーンやマルチリージョン アーキテクチャを使用することが挙げられます。

標準的な月のネットワーク費用が決まったら、費用を適切に割り当てるための各種のアプローチをいくつか検討してみましょう。アプリケーションをユーザーベースに合わせるためにソリューションを再設計するお客様もいれば、トラフィック量とレイテンシを削減するために Cloud CDN を使用するお客様もいます。Cloud CDN は低コストなため、コンテンツ提供時においてそのメリットを実感されている方もおられるでしょう。どちらの方法も、費用削減とパフォーマンス向上を両立できる実用的な選択肢です。

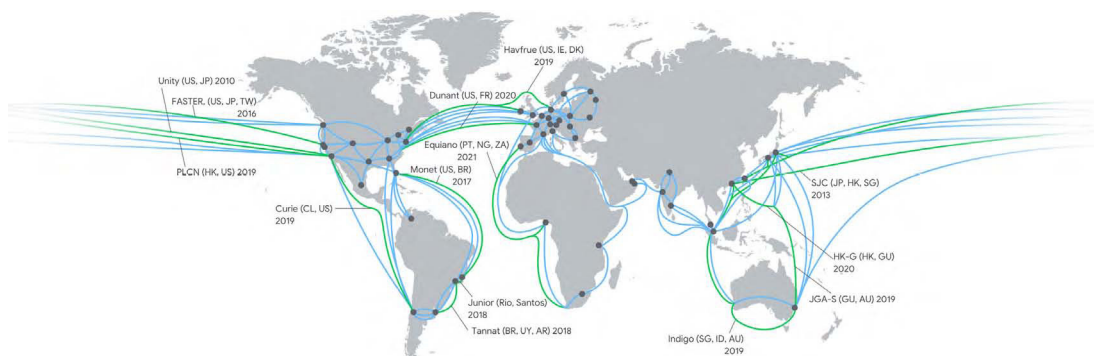
VPN を使用するタイミングを決定する

実装努力 ●●● 節約量 ●●●

全体的なネットワーク費用の確認で次に検討するのは、転送バイト数の合計です。VPCフローログを使用すると、環境内での「トラフィック使用量が特に多いプロセス」を確認できます。大量のデータ(テラバイトやペタバイトを想定)を push する場合は、利用できる割引を活用するようにします。

多くのお客様は日頃から、大量のデータをオンプレミス環境から Google Cloud に push しています。そうしたアップロードの手段として、VPN の使用や、(SSL で暗号化されたトラフィックの) インターネットを介した直接接続が利用されています。また、データベースを専用のオンプレミス ハードウェア上で実行し、フロントエンド アプリケーションでのリクエスト処理は Google Cloud で行われている場合もあり、お客様ごとに実装方法は異なります。お客様の環境が上記の事例と同様な場合は、[Dedicated Interconnect](#) か [Partner Interconnect](#) の利用をご検討ください。大量のデータを常に push する場合、パブリック インターネットを通過するトラフィックや VPN を使用するトラフィックにかかわる費用と比べて、専用接続費用のほうが安くなる可能性があります。

Interconnect を選択する際に[確認すべきアーキテクチャ上の検討事項](#)について、詳細をご覧ください。



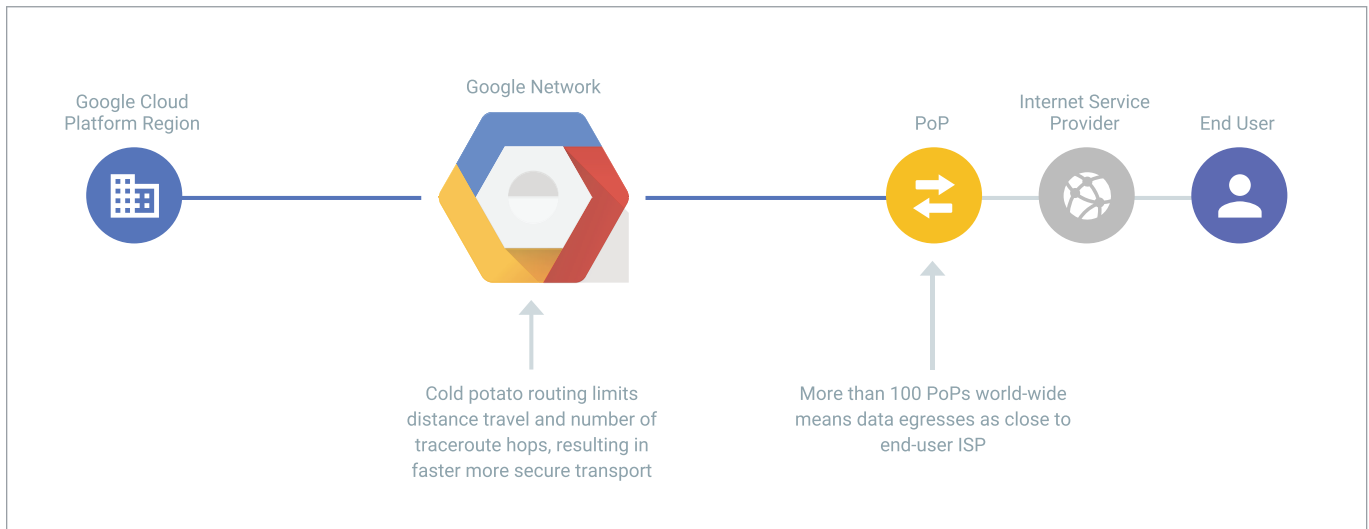
階層を利用したネットワークの最適化

実装努力 ●●● 節約量 ●●●●

Google Cloud の大きな特長の 1 つは、Google のプレミアム ネットワーク バックボーンにアクセスできることです。これはすべてのサービスにおいてデフォルトで使用されます。ただし、このようなパフォーマンスと低レイテンシはすべてのサービスに必要なわけではありません。たとえば、日次販売レポートの配信は世界中で即時入手できなくても構いません。パフォーマンスと費用のトレードオフが必要なサービスの場合は、[Network Service Tiers](#) のご利用をおすすめします。



スタンダード ネットワーキング ティアを選ぶことで、費用を削減できます。



プレミアム ネットワーキング ティアを選ぶと、パフォーマンス向上と低レイテンシが望めます。

スタンダード ティアかプレミアム ティアを選択すると、サービス間で適切な接続を割り当て、アプリケーションのニーズに合わせてネットワークを細かく調整できる他、レイテンシの要件が厳しくないサービス (SLA が設定されていないサービスなど) において、費用の削減を図れる場合があります。

料金面でのメリットのためにスタンダード ティアを利用する場合、いくつかの制約があります。おおまかに言うと、これらの制約にはパブリック インターネットを通過するトラフィックだけでなく、HTTP(S)、SSL プロキシ、TCP プロキシ負荷分散、Cloud CDN の使用に関するコンプライアンス ニーズが含まれます。いくつかの推奨事項を確認したら、チームと協力してサービスを見直し、低価格なスタンダード ティアを利用して料金面でのメリットを活かしながらも外部向けサービスのパフォーマンスへの影響を抑えられるか判断できます。

ネットワーク使用量の最適化

実装努力 ●●● 節約量 ●●●

上記のトピックは、ネットワーク費用を見直す際に活用できる大きな特長の一部です。ただし、基本的には、クラウドの大きなメリットの一つである従量制をご利用ください。この点を考慮し、Google Cloud への投資を最大限にご活用いただけるように、以下を確認することをおすすめします。

- Cloud Logging** - 不要になったログを除外することで、ネットワークトラフィックの可視性を制御できます。安全に除外できるログの**一般的な例**をご確認ください。これは、大量になり追加の費用が発生する可能性のある、データアクセス監査ログでも行えます。たとえば、開発プロジェクトでは、データアクセス監査ログを記録する必要はおそらくないでしょう。VPC フローログと Cloud Load Balancing では、サンプリングを有効にもできます。これにより、データベースに書き込まれるログトラフィックの量を劇的に減らすことができます。これは 1.0 (ログエントリの 100% を保持) から 0.0 (0%、ログは保持されない) まで設定できます。トラブルシューティングやカスタム ユースケースでは、特定の VPC ネットワーク、あるいはサブネットのテレメトリーを収集するか、さらにドリルダウンして特定の VM インスタンスまたは仮想インターフェースをモニタリングするかをいつでも選択できます。
- エンタープライズ向け、または大容量をご使用のお客様向けのプライベート アクセス** - 可能な場合、限定公開の Google アクセスを利用して費用を削減し、セキュリティ体制を改善します。
- 外部 IP アドレス** - 2020 年以降、無料枠にあてはまらない外部 IP アドレスには**若干の費用が発生するようになります**。ただし、一般的なセキュリティのベスト プラクティスとして、適切な場合は内部 IP アドレスを使用することをおすすめします。内部 IP に移行する方法については、**プライベート VM のインターネット接続の構築**や **Google Kubernetes Engine での限定公開クラスタの設定に関するガイド**をご覧ください。



上記を確認すると、設計における無駄な出費をなくし、クラウドベースのソリューションを十分に活用できます。パフォーマンス、可用性、セキュリティと費用との間でバランスを取ることは簡単なことではなく、多くの場合、複数のチームによるコラボレーションが必要です。Google は、検討すべき多くのアプローチがあると考えています。通常、費用の最適化は 1 回限りの確認というより、アプリケーション チームの原則とすべきものです。チームと企業にとって最適な方法を見つけてください。

第 5 章

BigQuery によるデータ分析費用の最適化

あらゆる業務のデータが各所に分散する今日の状況において、従来のデータ ウェアハウスの運営、管理は厄介で手間のかかる作業となりがちです。こうしたデータの増加に対応してシステムをスケールし、日々の運用を維持することは、これまでになく大きな課題となっています。課題はそれだけではありません。データ ウェアハウスをアップグレードするときにダウンタイムをできるだけ短くする、ML や AI に向けた取り組みを支えてビジネスニーズに応えるなどの必要にも迫られています。Google Cloud のサーバーレス、エンタープライズ向けデータ ウェアハウスである BigQuery は、インフラストラクチャ管理に手間を取られず分析作業に集中できるという点が評価され、数々の企業に導入されています。

BigQuery を使用すれば、非常に高速なクエリを実行して、ストリーミング データでリアルタイムの分析情報を取得し、機械学習機能が組み込まれた高度な予測分析の利用を開始できます。[Enterprise Strategy Group \(ESG\) の分析](#)により、BigQuery は他のクラウドデータ ウェアハウスと比べて、3 年間の総所有コスト (TCO) を最大 26%~34% 削減できることが明らかになりました。とは言え、BigQuery に格納したデータを最適化する余地がないわけではありません。クラウド コンピューティングの時代において、テクノロジーの導入を決める最大の要素の一つは費用です。そのため、当然の流れとして、料金の詳細や、費用を継続的に最適化する方法がお客様の関心事となります。

ここでは、経験とプロダクトの知識をもとに費用を最適化し、結果として収益の増加をもたらす方法をまとめて紹介します。BigQuery はサーバーレス アーキテクチャなので、費用を最適化することによってパフォーマンスも改善できるという特長があります。パフォーマンスと費用のどちらか 1 つを選ぶという苦渋の決断に迫られることはありません。



BigQuery の基本的な料金体系を理解する

まず、BigQuery の料金体系を見ていきましょう。続けて、料金をサブカテゴリごとに掘り下げて、BigQuery の費用を抑える方法を紹介します。

BigQuery の料金 は、ロケーションにかかわらず、以下のような内訳になっています (以下で詳細を確認できます)。

- **クエリの処理**
 - オンデマンド: 実行した各クエリで処理するデータ量に基づきます。
 - 定額料金: 費用の予測可能性を重視するお客様に最適です。お客様はクエリ処理専用のリソースを購入するため、個々のクエリに対して請求は発生しません。
- **ストレージ**
 - アクティブストレージ: テーブルまたはパーティションで、過去 90 日間で変更されたデータに対する月額料金。
 - 長期保存ストレージ: テーブルまたはパーティションで、過去 90 日間で変更されていないデータに対する月額料金。料金は安めに設定されています。
 - ストリーミング挿入

BigQuery ML については、[BigQuery ML の料金体系](#)をご覧ください。

BigQuery Data Transfer Service については、[BigQuery Data Transfer Service の料金体系](#)をご覧ください。

それぞれについて詳しく見ていく前に、まず、どのロケーションでも無料の BigQuery オペレーションをおさえておきましょう。

- BigQuery へのデータの一括読み込み
- 自動再クラスタリング (セットアップ、メンテナンス不要)
- データのエクスポート オペレーション
- テーブル、ビュー、パーティション、関数、データセットの削除

```
-- No cost, since no table is created.
DECLARE x DATE DEFAULT CURRENT_DATE;
-- Incurs the cost of scanning the table.
DECLARE y STRING DEFAULT 'foo';
-- Incurs the cost of copying the data into the table.
-- table is created, you can run the script.
CREATE TEMP TABLE t AS SELECT * FROM dataset.table;
-- Incurs the cost of scanning the table.
SELECT column1 FROM t;
-- No cost, since y = 'foo'.
IF y = 'foo' THEN
  -- Incurs the cost of scanning the table.
  -- y was equal to 'foo'.
  SELECT * FROM dataset.table;
ELSE
  -- Incurs the cost of scanning the table.
  -- y was not equal to 'foo'.
  UPDATE dataset.different_table
  SET col = 10
  WHERE true;
END IF;
-- Incurs the cost of scanning the table.
-- iteration of the loop.
WHILE x < (SELECT MIN(date) FROM dataset.table)
  -- No cost, since the table is already created.
  SET x = DATE_ADD(x, INTERVAL 1 DAY);
  -- No cost, since the table is already created.
  IF true THEN
    -- LEAVE has no associated cost.
    LEAVE;
  END IF;
  -- Never executed, since the loop condition is false.
  -- a cost.
  SELECT * FROM dataset.table;
END WHILE;
```

- メタデータ オペレーション
- キャッシュされたクエリ
- エラーを返したクエリ
- ストレージは毎月 10 GB まで無料
- クエリで処理するデータは、毎月 1 TB まで無料 (オンデマンド料金の場合の特典)

定額料金とオンデマンド料金の違いを理解する

実装努力 ●●● 節約量 ●●●

BigQuery でのデフォルトの料金プランは、各クエリ処理のバイト数をもとに請求額が決定されるオンデマンド料金体系です。ワークロードが安定しており大容量をご利用のお客様の場合、オンデマンドから定額料金への切り替えでより高い費用対効果を期待できる場合があります。定額料金体系では、予測可能な固定費で無制限のバイト数を処理できます。定額料金プランの場合、スロット コミットメントを購入します。これは、BigQuery スロットとして計算される専用のクエリ演算能力です。BigQuery Reservations を導入すれば、お客様は簡単かつ柔軟なセルフサービス式の方法で BigQuery の定額料金プランを活用できます。購入するスロット数を判断するには、Cloud Monitoring を使って前月のスロット使用量を可視化してみることをおすすめします。

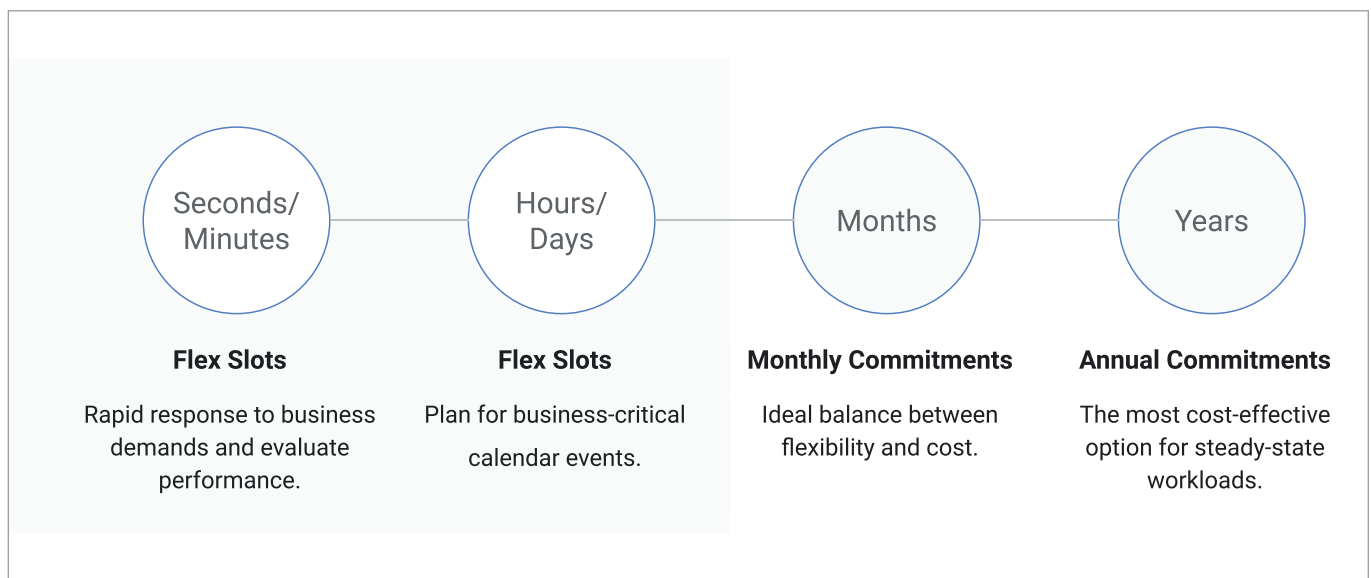
注: 定額料金の演算能力を超えるクエリを実行した場合は、必要なスロットが使用可能になるまでの間、一定の比率で処理速度が落ちます。

定額料金にすれば、クエリの最適化について一切心配しないで済むように思えるかもしれませんが、実際には、パフォーマンスも考慮する必要があります。同じスロット数でも、クエリの実行速度が速いほど、一定の時間内で完了できるジョブ数が多くなります。つまり、それ自体が、費用の最適化になるというわけです。

定額料金にすれば、クエリの最適化について一切心配しないで済むように思えるかもしれませんが。

購入するスロット数が少なすぎるとパフォーマンスが低下し、多すぎると演算能力がフル活用されず、費用の無駄になります。適切なスロット数を判断するには、まず、月額料金プランを開始してみましょう。このプランなら、30 日後にスロット数を減らすことも、キャンセルすることも可能です。必要なスロット数の見当がいたら、年額プランに切り替えることでさらに節約できます。また、[BigQuery Reservations](#) を使用することで、より効率的な定額料金プランの利用と出費の計画が簡単に行えます。

月ごと、年度ごとの定額料金契約のほかに、急激に変化するビジネスのニーズに合わせて、Flex Slots が最近導入されました。これは、最低 60 秒という短い時間単位で BigQuery スロットをご購入いただける新しいオプションです。

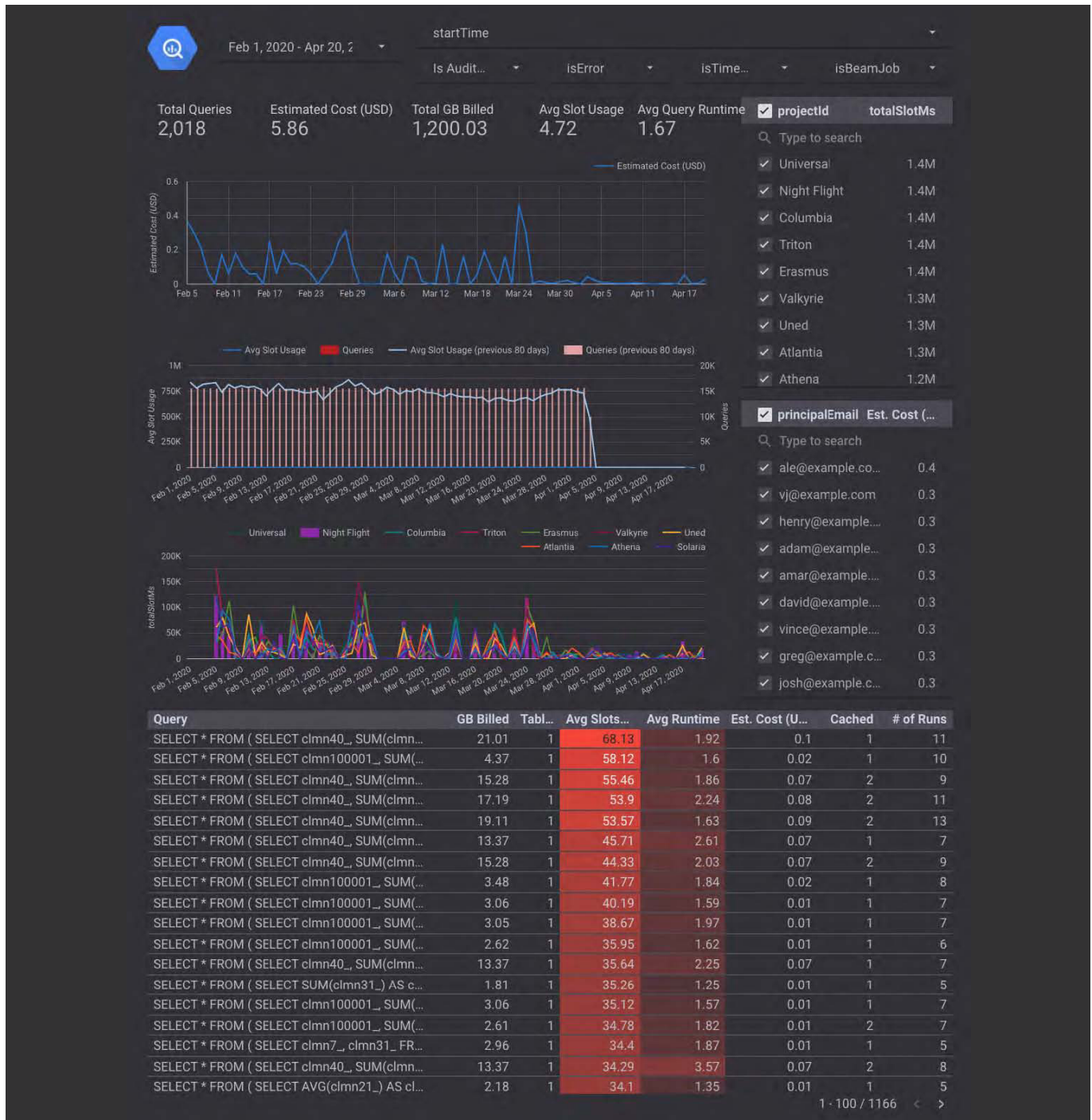


新しい Flex Slots を使えば、ニーズに柔軟に対応できます。

オンデマンドと定額料金プランを組み合わせることで、変化する分析需要に迅速かつ費用対効果の高い方法で対応できます。BigQuery の基本的な料金体系については、これでご理解いただけたと思います。次に費用の最適化の始め方について見ていきましょう。

BigQuery の費用を可視化する

組織内の費用を把握するには、まずは前月の BigQuery の使用状況について簡単なレポートを作成し、費用の現状を認識する必要があります。これには、Cloud Console の Billing レポートを使用するか、単純に課金データを BigQuery にエクスポートします。詳細な情報を確認できるデータポータルダッシュボードも利用可能です。これを使用すれば、費用の高いクエリを特定して、費用とクエリのパフォーマンスを最適化できます。また、ワークロードに関連するお客様の使用傾向やリソース使用率について有益な情報が入手できます。以下で説明する詳細な手順にそって、ダッシュボードを作成しましょう。



BigQuery のレポートで、使用状況が一目で確認できます。

BigQuery の費用最適化テクニック: クエリの処理

BigQuery データのクエリを実行する目的には、予測分析、リアルタイムの在庫管理など、分析やビジネスのユースケースへの利用が考えられます。

目的は何であれ、最初は BigQuery をオンデマンド料金で使い始めるケースが一般的です。オンデマンドの場合、BigQuery や外部データソースに格納されているデータ量にかかわらず、クエリで処理するデータ量 (バイト数) に応じて課金されます。従って、クエリで処理するデータ量を減らす方法を取り入れると効果的です。ここでは、クエリ (SQL コマンド、ジョブ、ユーザー定義関数など) の実行費用を削減するベスト プラクティスを見ていきましょう。

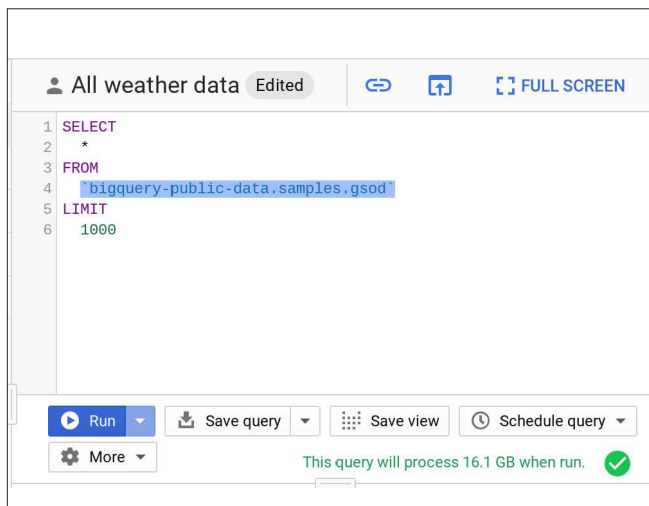
1. クエリの対象データを絞る。(特におすすめの方法)

実装努力 ●●● 節約量 ●●●

BigQuery は、データがカラム形式で格納されているため、パフォーマンスが高いのが特長です。ただし「SELECT *」を使用すると、テーブル中のすべての列に対してクエリが実行されるため、費用が最も高くなります。この中には、不要なデータも含まれているかもしれません (前月に「SELECT *」を使った回数を数えて罪悪感に囚われるのは、ご自身だけではありません)。

では、クエリで処理されるデータ量の具体例を見てみましょう。以下は、BigQuery の一般公開されている天気データセットに対してクエリを実行する例です。

ご覧のとおり、必要な列を絞り込むことで、処理データのバイト数が約 8 分の 1 にまで減り、費用の削減に直結していることがわかります。また、LIMIT 句を適用しても、特に費用には影響しないことにご注目ください。

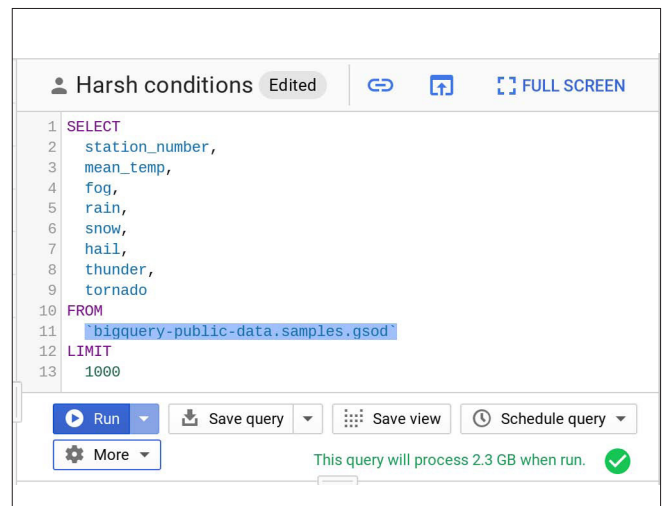


```

1 SELECT
2   *
3 FROM
4   `bigquery-public-data.samples.gsod`
5 LIMIT
6   1000
  
```

Run Save query Save view Schedule query More

This query will process 16.1 GB when run. ✓



```

1 SELECT
2   station_number,
3   mean_temp,
4   fog,
5   rain,
6   snow,
7   hail,
8   thunder,
9   tornado
10 FROM
11   `bigquery-public-data.samples.gsod`
12 LIMIT
13   1000
  
```

Run Save query Save view Schedule query More

This query will process 2.3 GB when run. ✓

BigQuery での一般公開されている天気データセットのクエリ実行例

データについて深く掘り下げたり、その背景情報を理解したりする必要がない場合は、無料のデータ プレビュー オプションの使用をおすすめします。

また、クエリ実行の最初のステージで処理されたバイト数だけが課金対象となることも重要なポイントです。中間ステージで処理されるバイト数を減らすために、複数のステージから成る複雑なクエリを作成することは、費用の面では意味がないので避けましょう（ただし、パフォーマンスが改善される可能性はあります）。

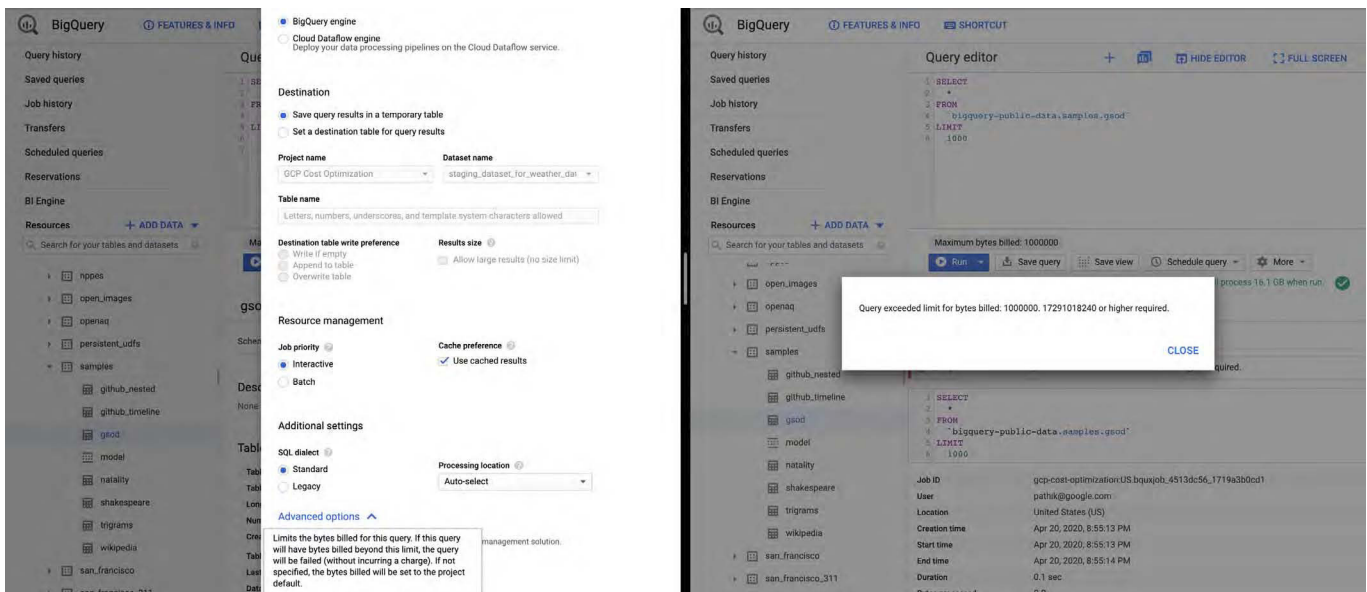
クエリのできるだけ早い段階でフィルタリングを行うことで、BigQuery の費用を削減すると同時にパフォーマンスも改善できます。

2.人為的なミスを防ぐために、上限を設定する。

実装努力 ●●● 節約量 ●●●

上記のクエリで処理されるデータ量は GB 規模なので、たとえ間違いがあっても無駄になるのは数円程度で、ビジネスには大きく影響しません。しかし、データセット テーブルの規模が TB、PB におよぶ場合、そのすべての列に対して、しかも多数のユーザーがクエリを実行すると、いつの間にか費用が膨れ上がる恐れがあります。

このような事態を防ぐために、課金される最大バイト数を設定しておくようにしましょう。この上限を越えると、以下のようにクエリのエラーとなり、費用がそれ以上発生しなくなります。



The left screenshot shows the BigQuery console interface. On the left is a sidebar with navigation links like 'Query history', 'Saved queries', 'Job history', 'Transfers', 'Scheduled queries', 'Reservations', 'BI Engine', and 'Resources'. The main area shows the 'Destination' and 'Resource management' settings for a query. The 'Destination' section has options for 'Save query results in a temporary table' and 'Set a destination table for query results'. The 'Resource management' section has options for 'Job priority' (Interactive, Batch) and 'Cache preference' (Use cached results). The 'Additional settings' section has options for 'SQL dialect' (Standard, Legacy) and 'Processing location' (Auto-select). The 'Advanced options' section has a checkbox for 'Limit the bytes billed for this query'.

The right screenshot shows the BigQuery console interface with a query editor. The query editor shows a SQL query: `SELECT * FROM bigquery-public-data.samples.goad LIMIT 1000`. Below the query editor, there is a table with query results. An error message is displayed: 'Query exceeded limit for bytes billed: 1000000. 17291018240 or higher required.' The error message has a 'CLOSE' button.

以前、カスタム コントロールの重要性について、お客様に尋ねられたことがあります。たとえば、米国（マルチリージョンのロケーション）に 10 TB のデータがあるとしましょう。ストレージの料金は、1 か月あたり約 \$200 です。このデータに対して、10 人のユーザーがそれぞれ [SELECT * ..] を月に 10 回実行した場合、BigQuery の 1 か月あたりの処理データは 1 PB となり、料金は約 \$5,000 にまで跳ね上がります。妥当な上限を設けることによって、こうした不用意なクエリを防止できます。なお、クエリの実行を途中でキャンセルした場合でも、クエリを最後まで実行したのと同じだけの料金が発生する可能性があるのでご注意ください。

クエリごとに上限を設けるほか、ユーザーレベルやプロジェクト レベルで同じように制限を設けて費用をコントロールすることも可能です。

3. キャッシュを賢く利用する。

実装努力 ●●● 節約量 ●●●

一部の例外を除き、キャッシュを使ったほうがクエリのパフォーマンスが改善されます。加えて、キャッシュ テーブルから取得した結果については料金が発生しないというメリットもあります。キャッシュの設定はデフォルトでオンになっています。この設定は、GCP Console のクエリエディタで [詳細] > [クエリの設定] をクリックして、以下の画面で確認できます。また、キャッシュはユーザーごと、プロジェクトごとに維持される点にもご注意ください。

Destination table write preference

- ☐ Write if empty
- ☐ Append to table
- ☐ Overwrite table

Resource management

Job priority ?

- ☒ Interactive
- ☐ Batch

Cache preference ?

- ☒ Use cached results

This attempts to use results from a previous run of this query, as long as the referenced tables are unmodified. If cached results are returned, you will not be billed for any usage. Results are cached for approximately 24 hours.

Caching cannot be enabled when a destination table is selected. [Learn more](#)

キャッシュを使えば、クエリのパフォーマンスを改善できます。

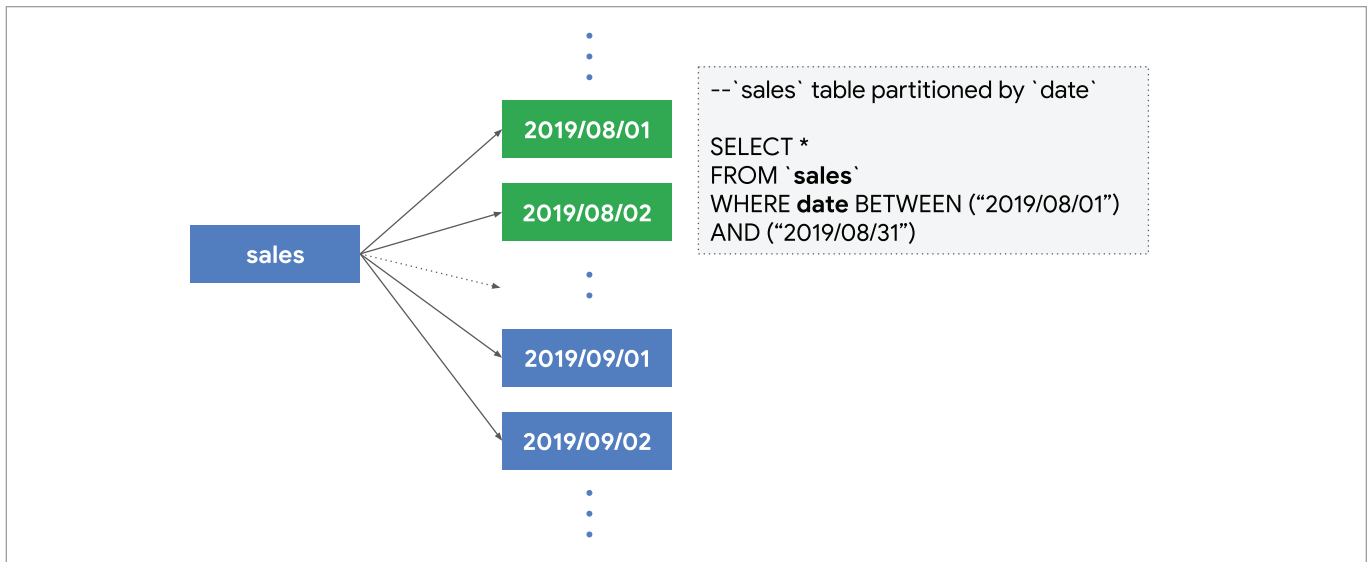
具体例を挙げてみましょう。数百人から数千人のユーザーがアクセスするデータポータルダッシュボードがあり、その背後で BigQuery が動作しているとします。この場合、複数のユーザー間でクエリのキャッシュを共有する工夫が必要であることがすぐにおわかりいただけるでしょう。

ユーザー間のキャッシュヒットが高確率で起こるようにするには、単一のサービス アカウントを使って BigQuery のクエリを実行するか、コミュニティ コネクタを使用します。詳しくは、こちらの Next '19 のデモをご覧ください。

4. テーブルを分割する。

実装努力 ●●● 節約量 ●●●

できるだけテーブルを分割すると、クエリ処理の費用を削減できると同時に、パフォーマンスも改善されます。現時点では、取り込み時間、日付、タイムスタンプの列の値に従って、テーブルを分割できます。たとえば、過去 12 か月分のデータを含む sales テーブルを以下のように日別のテーブルに分割してみましょう。



テーブルを分割すると、クエリ処理の費用を削減できます。

こうしておけば、8 月の **sales** データを分析するとき、テーブル全体ではなく、31 個のパーティションのデータのみが処理されることになり、費用を節約できます。

この方法のもう 1 つのメリットは、先ほど説明した長期保存ストレージへの移行が、個々のパーティションごとに考慮されるようになるという点です。上記の **sales** データの例で言うと、頻繁に読み込みや編集の対象となるデータは過去 2〜3 か月分です。従って、その他の過去 90 日間編集しなかったパーティションについては、ストレージ費用が自動的に削減されることになります。なお、パーティション分割テーブルのメリットを活かすには、クエリを実行するときに、パーティション列を指定してテーブルを絞り込むようにしてください。

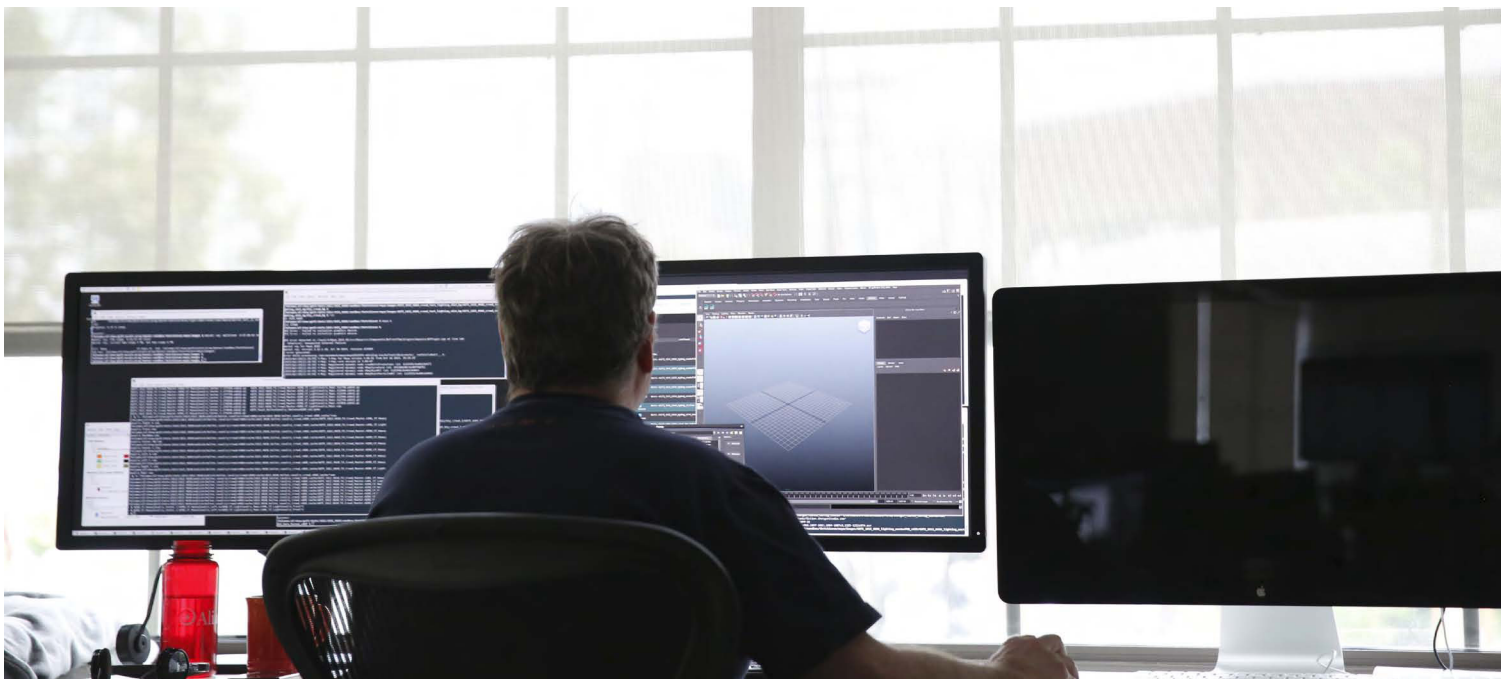
パーティション分割テーブルの作成や更新を行うときに [パーティション フィルタを要求] を有効にすると、WHERE 句を使ったパーティション列の指定をユーザーに強制できます (指定しないと、エラーが返されるようになります)。

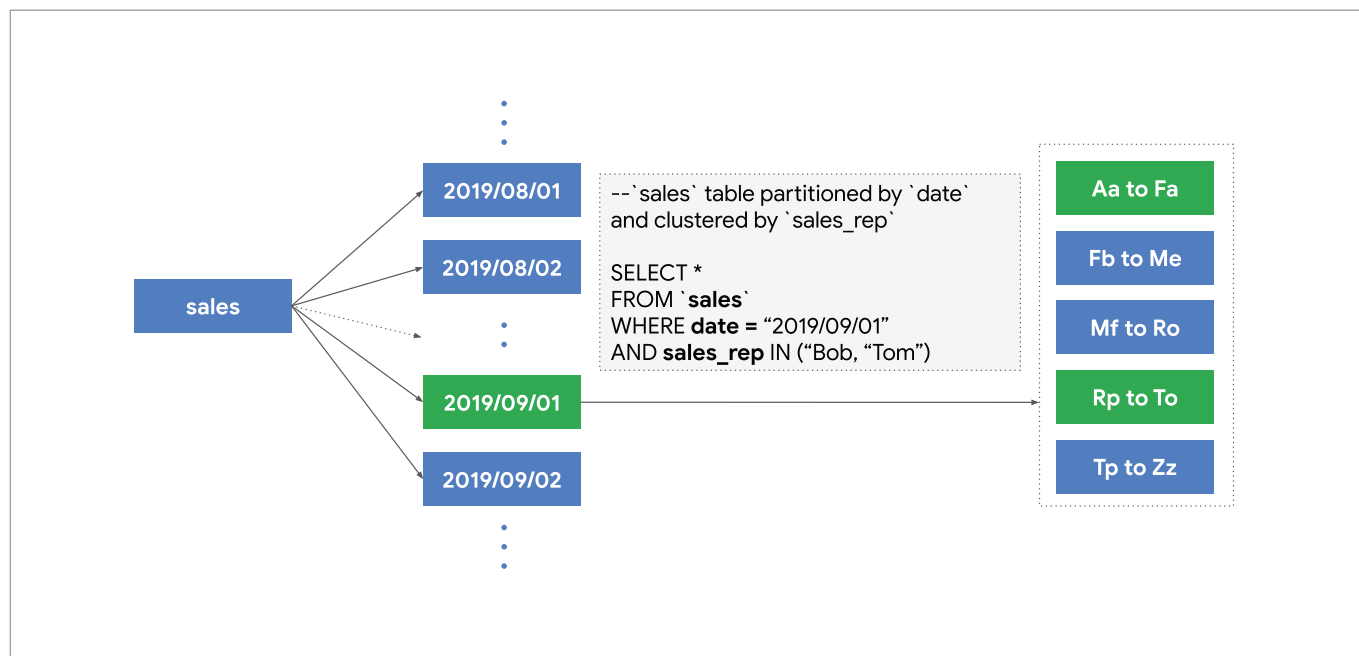
5. クラスタリングを使って、データをさらに縮小する。

実装努力 ●●● 節約量 ●●●

テーブルを分割した後、さらに**そのテーブルをクラスタリング**できるようになりました。クラスタリングとは、最大 4 つの列に値に基づいてデータを分類、整理する仕組みで、クラスタリングで指定された列の値に基づいてデータが並べられ、ブロックに分けられます。これらの列の値で絞り込むようなクエリを実行すると、ブロック プルーニングと呼ばれるプロセスによって、該当する値が含まれるブロックだけが自動的に判別されてスキャンされます。

たとえば、セールス部門の幹部社員が、特定の営業部員の指標だけをダッシュボードで確認できるようにしたいとしましょう。この場合は、絞り込みの条件となる `sales_rep` 列の値を順番に並べて、クラスタリングしておくとう便利です。以下の例では、1 つのパーティション (2019/09/01) 中の、Bob と Tom が含まれる 2 つのブロックのみがスキャンされ、このパーティションのその他のブロックは除かれています。これにより、処理するバイト数をさらに減らして、クエリ処理の費用を削減できます。





クラスタリングを有効にすると、費用を削減できます。

クラスタリングは、パーティション分割されたデータに対してのみ実行できます。テーブルをパーティショニングするには、データの取り込み情報を使用できます。または、架空の日付やタイムスタンプの列を導入し、その列の値に基づいて分割することも可能です。

クラスタリングについて詳しくは、[こちら](#)をご覧ください。また、パフォーマンスと効率を改善し、費用の削減を実現する新しいマテリアライズド ビューをお試しください。

BigQuery の費用最適化テクニック: ストレージ

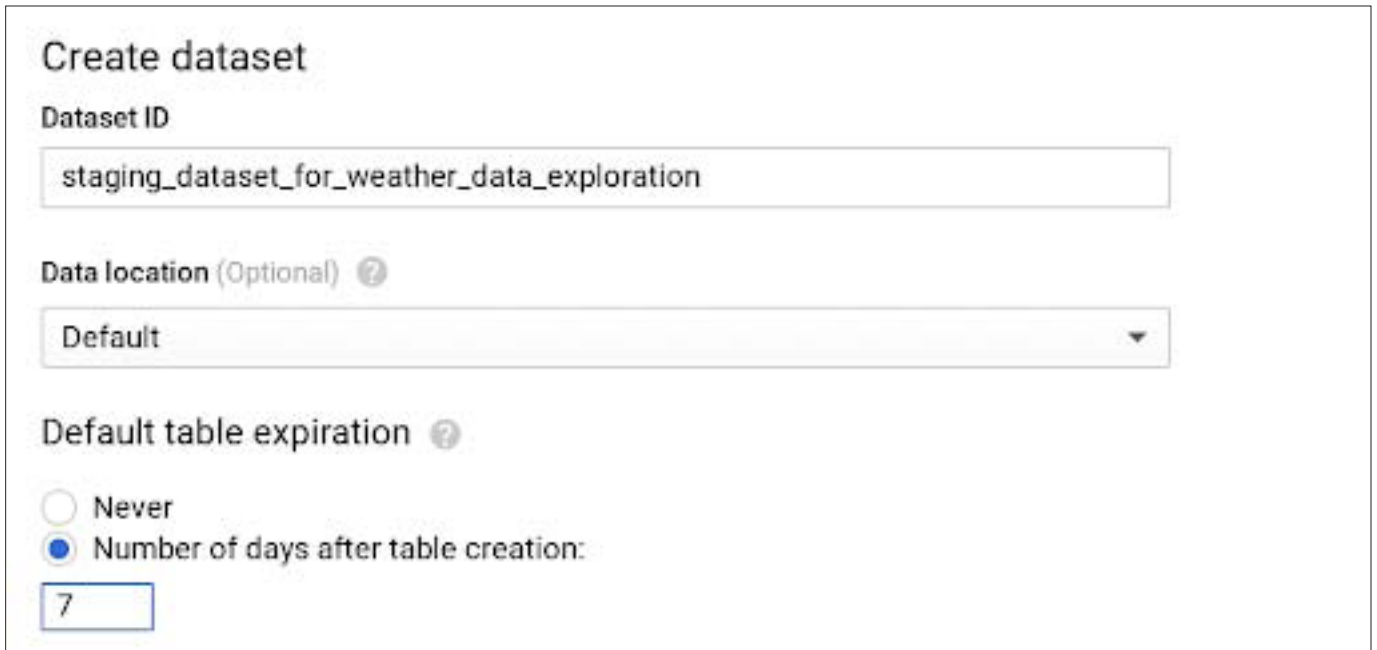
BigQuery にデータを読み込むと、テーブルに格納したデータ量と秒数に基づいて課金されます。BigQuery のストレージ費用を抑えるコツは以下のとおりです。

1. 必要なデータだけをキープする。

実装努力 ●●● 節約量 ●●●

BigQuery では、データがデフォルトで **Capacitor と呼ばれるカラム形式** で暗号化、圧縮された状態で格納されます。データセットにテーブルのデフォルトの有効期間を構成し、長期間にわたって保存する必要のない一時的なステージング データを削除するようにしましょう。

たとえば、天気のスレーシング データセットは、ダウストリーム ジョブによるクエリやクリーニングを経て、本番環境のデータセットにプッシュされた時点で不要になります。このような場合は、テーブルのデフォルトの有効期限を 7 日間に設定できます。



Create dataset

Dataset ID

staging_dataset_for_weather_data_exploration

Data location (Optional) ?

Default

Default table expiration ?

☐ Never

☒ Number of days after table creation:

7

テーブルの有効期限は、リソースの節約に役立ちます。

なお、テーブルのデフォルトの有効期間を変更すると、その変更は新しく作成したテーブルに対してのみ有効となります。既存のテーブルに変更を加えるには、**DDL ステートメント**を使用してください。

BigQuery では、同じデータセット内でテーブルごとに異なる有効期限を設定することもできます。たとえば、先ほどのデータセットに new_york というテーブルがあり、他のテーブルよりも長く保存する必要があるとします。

次のページの図のように設定すると、new_york テーブルのデータは 6 か月間保持されますが、california テーブルの有効期限は特に設定していないので、デフォルトの 7 日間になります。

有効期限は、データセットやテーブルごとのほか、パーティションごとに設定することも可能です。デフォルトの動作については、**一般公開されているドキュメント**をご覧ください。

2.データの編集操作に注意する。

実装努力 ●●● 節約量 ●●●

テーブルやパーティションを 90 日間編集しなかった場合、そのデータのストレージ価格は自動的に約 50% 値下がりします。テーブルやパーティションが長期保存ストレージと見なされた場合でも、パフォーマンス、耐久性、可用性などが低下することはありません。

長期保存ストレージをできるだけ活用するため、テーブルのデータを編集する操作（データのストリーミング、コピー、読み込みや、DML オペレーション、DDL オペレーション）に気を付けましょう。こうした操作を行うと、長期保存ストレージからアクティブ ストレージに戻り、90 日間のカウントがリセットされます。これを防ぐには、新しいデータバッチを読み込む際、新しいテーブルやパーティションへの読み込みが可能かどうかを用途に応じて検討しましょう。

テーブルデータに対するクエリの実行など、**一部の操作**では、90 日間のカウントがリセットされず、長期保存ストレージのままになります。

一般的には BigQuery にデータを置いたままにするほうがメリットが大きいです。テーブル内のデータにアクセスするのが年に 1 度あるかないかという程度の場合は（法律や規制に従ってアーカイブを保存する場合など）、Cloud Storage バケットの Coldline クラスに**テーブルデータをエクスポート**することを検討しましょう。このクラスは、BigQuery の長期保存ストレージよりもさらにお得な価格設定になっています。

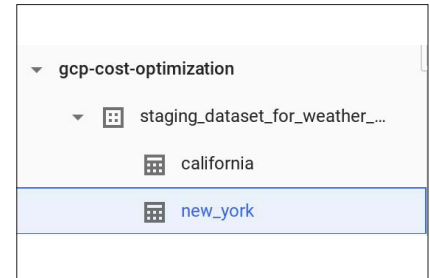


Table info

Table expiration

- ☐ Never
☒ Specify date

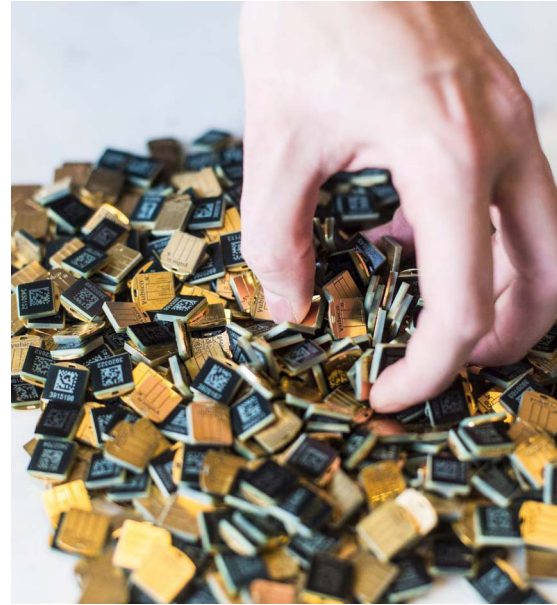
2/19/20, 11:30 PM EST ▼

3.データの重複コピーを避ける。

実装努力 ●●● 節約量 ●●●

BigQuery では[連携型のデータアクセス モデル](#)を採用しており、Cloud Bigtable、Cloud Storage、Google ドライブ、Cloud SQL などの外部データソースに対して直接クエリを実行することが可能です。この仕組みを利用すれば、データの重複コピーを避け、ストレージ費用を抑えることができます。さらに、外部ソースから直接データを読み取れるほか、少量のデータが頻繁に変更されるような場合には、変更のたびに BigQuery に読み込まなくても直接アクセスできるので便利です。

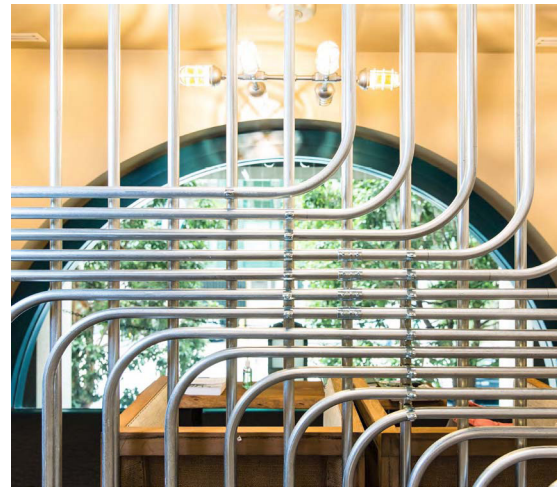
このテクニックは、目的に応じてメリットが得られる場合にのみ使用するようにしましょう。一般的には、[外部ソースに対するクエリ](#)は、BigQuery 上の同一データに対するクエリに比べてパフォーマンスが劣ります。BigQuery のデータはカラム形式で格納されているため、高いパフォーマンスを得られるのが特長です。



4.データをストリーミング挿入で読み込んでいるかどうかを確認する。

実装努力 ●●● 節約量 ●●●

BigQuery へのデータの読み込みには、一括読み込みジョブとして読み込む方法と、[ストリーミング挿入](#)を使用してリアルタイム ストリーミングで読み込む方法の 2 つがあります。BigQuery の費用を最適化する場合、まず最初にすべきことは請求書を確認し、ストリーミング挿入の料金が発生しているかどうかを確認することです。ストリーミング挿入の料金が発生している場合は、次の点を確認してみてください。「BigQuery のデータは早急に（数時間ではなく、数秒で）必要か?」、「BigQuery で用意したデータを、リアルタイムで使用するケースがあるか?」これらの質問のいずれかに対する答えが「いいえ」である場合は、無料の一括読み込みに切り替えることをおすすめします。



5. BigQuery のバックアップおよび DR プロセスを理解する。

実装努力 ●●● 節約量 ●●●●

BigQuery では、テーブルの変更履歴が 7 日間保持されます。ポイントインタイム スナップショットに対してクエリを実行できるので、バックアップから復元しなくても、データを特定の時点まで戻すことが可能です。なお、テーブルを削除すると、その履歴は 2 日後にフラッシュされます。

1 時間前のテーブルのスナップショットから行数を求めるクエリの例を以下に示します。

```
Select COUNT(*) FROM [Project _ ID:Dataset.Table@-3600000]
```

詳しくは、ドキュメントに記載されている例を参照してください。

ビジネスに不可欠なデータは、[データの障害復旧シナリオ](#)のガイドに沿ってバックアップするようにしましょう。特に BigQuery を [リージョン ロケーション](#) で使用している場合は、データを必ずバックアップしてください。

努力を称える

適正に使用することで、BigQuery は非常にリーズナブルな価格で最新のデータ ウェアハウジングのニーズをすべて満たすことができます。これらの費用削減テクニックをいったん取り入れれば、BigQuery の料金が目に見えて下がっていくのを実感できるはずです（最初から実践していたのなら、話は別ですが）。いずれにしても、努力を称えましょう。必ずや成果を得られるものと確信しております。

実現された費用の最適化

これらの原則に従うことで、短期的には費用を節約し、容量を開放できますが、長期的な費用対効果としては、より多くの利益がもたらされます。お話をしたお客様からは、データ費用の大幅な削減とインフラストラクチャの節約により、まったく新しい機能を追加して、機械学習などの革新的なソリューションを検討できるようになった、と聞いています。また、これらのお客様においては、IT 環境全体のパフォーマンスの向上とシンプルさも実現されています。必要に応じたスケーリングもはるかに簡単に行えます。

たとえば、ある調査では、Google Cloud の Spanner データベースを使用することにより、オンプレミス データベースの総所有コスト (TCO) を 78% も削減できることがわかりました。[Optiva はレガシー Oracle データベースを Google Cloud へ移行](#)することにより費用効率を高め、より優れたスケーリングを実現しました。

また、さまざまな業界のお客様が、クラウドを使用して多くのことを実行でき、他のクラウド プロバイダーやレガシー テクノロジーと比較して、チームの生産性を向上させることができています。これは、顧客体験の向上、新たなプロダクト イノベーション、新しいビジネス イニシアチブの追加につながる可能性があります。機械学習プラットフォームのプロバイダーである [MD Insider](#) は頻繁にネットワーク パフォーマンスの低下を経験し、その結果、失敗率と費用が増大しました。そこで、データサービスに Google Cloud を使用すると、パフォーマンスが 5 倍向上し、製品化までの時間が最大 30% 短縮されました。医師のパフォーマンス サイトの開発者においては、今や生産性が大幅に向上され、以前はインフラストラクチャの管理に費やされていた時間が、製品開発に使用されています。

Compute Engine、Cloud Bigtable、その他のテクノロジーを使用して、[Raycatch](#) は AI が主導する再生可能エネルギー技術のインフラストラクチャ費用を 80% 削減しました。同社は、システムに柔軟性と安定性を追加し、以前のクラウドから容量を増やしたため、開発者はより速く作業ができるようになりました。Raycatch は、1 つの分析タスクを監視するために必要な作業時間を 1/60 に削減できたため、IT チームはより価値のある最適化作業に時間を使えるようになりました。

クラウドを最大限に活用するために役立つ多くの費用管理戦略とベスト プラクティスがあります。それぞれのチーム、組織に最適なものを使用してください。ほんの少しの努力で、クラウドはすぐに役立つ効率性と将来の大きな ROI の可能性を提供します。

費用の最適化の詳細

短期的および長期的な成長を達成するためには、IT 支出を削減、最適化する方法を見つけることが重要です。

[Google Cloud エキスパートに相談しましょう。](#)





Google Cloud

cloud.google.com