



---

## リソース管理

このモジュールではリソース管理について説明します。Google Cloud のリソースは課金対象であるため、リソースの管理はコスト管理につながります。リソースへのアクセスを制御する方法はいくつかあり、割り当てを使用すると消費量を制限できます。

ほとんどの場合、デフォルトの割り当ては要求に応じて増やすことができます。割り当てを設定すると、そのリソースが大量に消費することを目的としたものであるかを確認するチェックポイントまたは機会として利用できます。

# アジェンダ

---

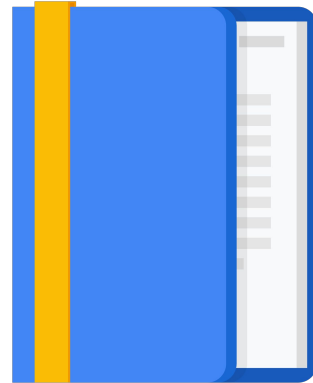
Resource Manager

割り当て

ラベル

課金

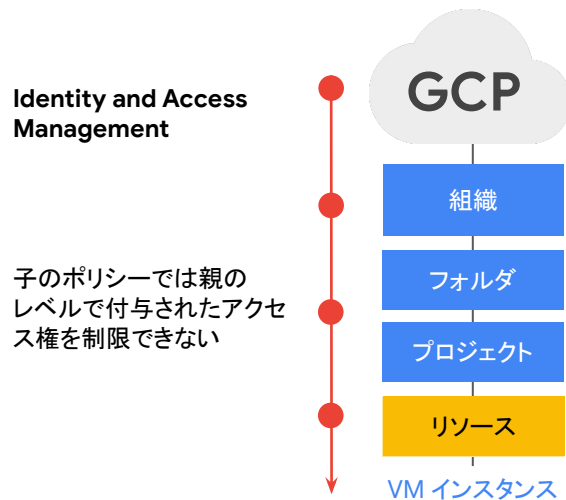
ラボ



このモジュールは、IAM モジュールで学んだ事柄に基づいて構築されています。最初に、Resource Manager の概要を示します。次に、割り当て、ラベル、名前について説明します。その後、予算とアラートの設定に役立つ課金について説明します。ラーニング エクスペリエンスの最後に、ラボで BigQuery を使用して課金データを調査します。

Resource Manager の概要から始めましょう。

## Resource Manager ではリソースを階層的に管理可能

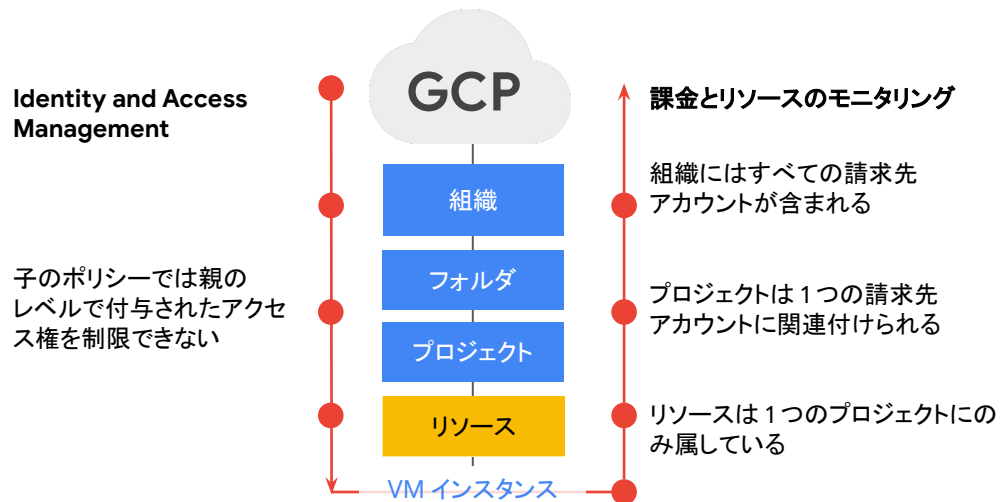


Resource Manager では、リソースをプロジェクト、フォルダ、組織ごとに階層的に管理できます。この点については、Cloud IAM モジュールで説明しているので馴染みがあるはずです。まずここでおさらいしましょう。

ポリシーには一連のロールとメンバーが含まれています。また、ポリシーはリソースで設定されます。左側に示されているように、これらのリソースは親からポリシーを継承します。そのため、リソース ポリシーは親とそのリソースのポリシーを合わせたものです。

また、親ポリシーの方が制限が緩い場合、厳しい方のリソース ポリシーはオーバーライドされることに注意してください。

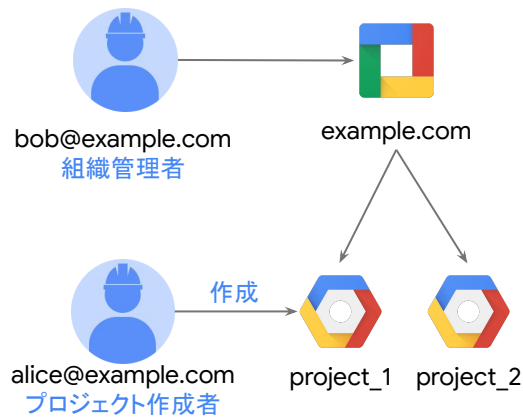
## Resource Manager ではリソースを階層的に管理可能



IAM ポリシーは上から順に継承されますが、右側に示されているように、課金は下から順に累積されます。リソースの消費は、使用率や時間の割合、アイテム数、機能の使用率などと同様に量で測定されます。リソースは 1 つのプロジェクトにのみ属しているため、プロジェクトではその全リソースの消費量が累積されます。

各プロジェクトは 1 つの請求先アカウントに関連付けられています。つまり、組織にはすべての請求先アカウントが含まれています。組織、プロジェクト、リソースなどについて見ていきましょう。

## 組織ノードは GCP リソースのルートノード



繰り返しになりますが、組織ノードは Google Cloud Platform のすべてのリソースのルートノードです。この図の例には、組織管理者ロールで組織ドメインを管理している Bob という人物がいます。Bob は Alice をプロジェクト作成者とし、個々のプロジェクトの権限とアクセス権を Alice に委任しています。

## プロジェクトではその全リソースの消費量が累積される

- リソースと割り当て使用量の追跡
  - 課金の有効化
  - 権限と認証情報の管理
  - サービスと API の有効化
- プロジェクトで使用される 3 つの識別属性:
  - プロジェクト名
  - プロジェクト番号
  - プロジェクト ID (アプリケーション ID と呼ばれる)

プロジェクトではその全リソースの消費量が累積されるので、プロジェクトを使用してリソースと割り当て使用量を追跡できます。具体的には、プロジェクトで課金の有効化、権限と認証情報の管理、サービスと API の有効化を行うことができます。

Cloud Platform リソースとやり取りするには、リクエストごとにプロジェクト識別情報を提供する必要があります。

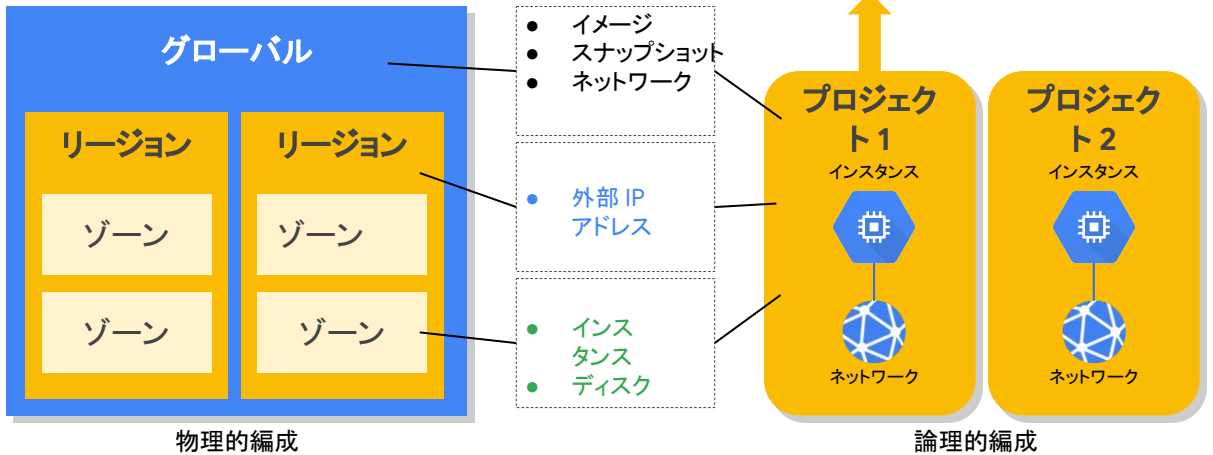
プロジェクトは次の要素で識別できます。

- プロジェクト名: 人が読める形式でプロジェクトを識別する方法ですが、Google API によって使用されることはありません。
- プロジェクト番号: サーバーによって自動的に生成され、プロジェクトに割り当てられます。
- プロジェクト ID: プロジェクト名から生成される一意の ID です。

これら 3 つの識別属性は GCP Console のダッシュボードで確認できます。また、Resource Manager API に問い合わせることもできます。

## リソース階層

リソースはグローバル、リージョン、ゾーンに分類される



最後に、リソース階層について説明します。物理的編成の観点からみると、リソースはグローバル、リージョン、ゾーンに分類されます。

例をいくつか見てみましょう。

- イメージ、スナップショット、ネットワークはグローバル リソースです。
- 外部 IP アドレスはリージョン リソースです。
- インスタンスとディスクはゾーンリソースです。

ただし、その種類に関係なく、各リソースは 1 つのプロジェクトに属しています。このため、各プロジェクトにおいて独自の課金とレポート作成を実行できます。

# アジェンダ

---

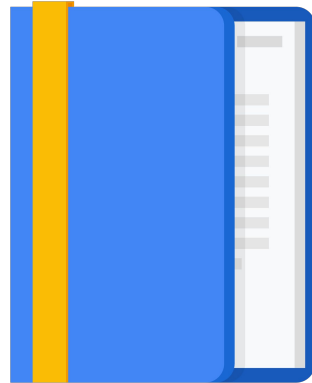
Resource Manager

割り当て

ラベル

課金

ラボ



プロジェクトではその全リソースの消費量が累積されることを理解したので、次に割り当てについて説明します。



## すべてのリソースにはプロジェクト割り当て または制限が適用される

- プロジェクトごとに作成できるリソースの数
  - 5 VPC ネットワーク/プロジェクト
- プロジェクトで API リクエストを実行できる速度(レート制限)
  - 5 つの管理アクション/秒 (Cloud Spanner)
- リージョンごとに作成できるリソースの数
  - 24 個の CPU/リージョン(プロジェクトごと)

**増加させる場合:** GCP Console の [割り当て] ページまたはサポート チケット

GCP のすべてのリソースには、プロジェクト割り当てまたは制限が適用されます。通常、これらは次に示した 3 つのカテゴリのいずれかに分類されます。

- プロジェクトごとに作成できるリソースの数。たとえば、プロジェクトにつき 5 つの VPC ネットワークのみを使用できます。
- プロジェクトで API リクエストを実行できる速度(レート制限)。たとえば、Cloud Spanner API を使用している場合、プロジェクトにつき 5 つの管理アクション/秒をデフォルトで実行できます。
- リージョン割り当てもあります。たとえば、デフォルトでは、リージョンごとに 24 個の CPU を使用できます。

これらの割り当てを前提とした場合、96 コアの VM のいずれかをスピンアップするにはどうすればよいでしょうか。

時間とともに GCP の使用量が多くなった場合、割り当て量を増やすことができます。使用量の大幅な増加が見込まれる場合は、事前に GCP Console の [割り当て] ページから割り当て量の調整をリクエストできます。また、このページには現在の割り当ても表示されます。

変更可能であるなら、なぜ割り当てが必要なのでしょう。

## プロジェクト割り当てを使用する理由

- エラーまたは悪意のある攻撃が発生した場合に過度な消費を防止
- 請求額の予定外の急増を防止
- サイズの検討と定期的な見直しを促す

プロジェクト割り当てを使用すると、エラーまたは悪意のある攻撃が発生した場合に過度な消費を防止できます。たとえば、gcloud コマンドラインで Compute Engine インスタンスを 10 個ではなく誤って 100 個作成した場合などです。

また、割り当てを使用すると、請求額の予定外の急増を防止できます。割り当ては課金に関連しますが、課金の管理に非常に役立つ予算とアラートの設定方法については後から説明します。

最後に、割り当てを使用すると、サイズの検討と定期的な見直しが促されます。たとえば、96 コア インスタンスが本当に必要であるか、小規模で低コストなものに置き換えることができるか、などの点について検討します。

また、割り当ては、リソースが使用可能な場合に、そのリソースの種類に作成できるリソースの最大量であるという点にも注意しましょう。割り当てが適用されているからといって、そのリソースを常に利用できるわけではありません。たとえば、利用できるローカル SSD がないリージョンでは、ローカル SSD の割り当てが残っていてもローカル SSD を作成することはできません。

# アジェンダ

---

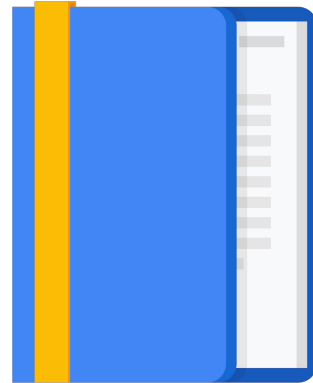
Resource Manager

割り当て

ラベル

課金

ラボ



プロジェクトとフォルダではリソースの分離レベルを指定できますが、さらに粒度が必要な場合にはどうすればよいでしょうか。ここで必要になるのがラベルです。

## ラベルは GCP リソースを整理するためのユーティリティです

- リソースに設定: VM、ディスク、スナップショット、イメージ
  - CCP Console、gcloud、または API
- ラベルの使用例:
  - 在庫
  - リソースのフィルタ
  - スクリプト
    - 費用の分析に使用
    - 一括オペレーションの実行

× ラベルの編集

選択されたインスタンス (1): instance-1

Key	Value
department	website-deve
engineering	development
owner	bobzalman
project	account-1569

+ ラベルを追加

保存 キャンセル

ラベルは GCP リソースを整理するためのユーティリティです。ラベルとは、VM、ディスク、スナップショット、イメージなどのリソースに設定可能な Key-Value ペアのことです。ラベルは、GCP Console、gcloud、または Resource Manager API を使用して作成および管理できます。また、各リソースには最大 64 個のラベルを設定できます。

たとえば、仮想マシンの環境を定義するラベルを作成できます。次に、各インスタンスのラベルを production または test のいずれかに定義します。このラベルを使用すると、すべての production リソースを検索して、在庫目的でリストできます。

また、ラベルをスクリプトで使用して、費用の分析や、複数のリソースでの一括オペレーションの実行が可能です。右側のスクリーンショットは、インスタンスで作成される 4 個のラベルの例を示しています。

## ラベルの用途

- チームまたはコストセンター  
team:marketing  
team:research
- コンポーネント  
component: redis  
component: frontend
- 環境またはステージ  
environment: prod  
environment: test
- オーナーまたは連絡先  
owner:gaurav  
contact:opm
- 状態  
state:inuse  
state:readyfordeletion

どのようなラベルを使用するかについて、例をいくつか見ていきましょう。

- チームまたはコストセンターに基づくラベルを追加し、別々のチームが所有しているインスタンスを区別することをおすすめします。このタイプのラベルは、費用計算または予算作成に使用できます。たとえば、team:marketing や team:research などです。
- また、ラベルを使用して、コンポーネントを区別することもできます。たとえば、component:redis や component:frontend などです。
- 環境やステージに基づくラベルを使用できます。
- また、ラベルを使用して、リソースのオーナーまたはメインの連絡先を定義することも検討してください。たとえば、owner:gaurav や contact:opm などです。
- または、リソースにラベルを追加して、その状態を定義します。たとえば、state:inuse や state:readyfordeletion などです。

## ラベルとタグの比較

- ラベルは GCP でリソースを整理するための方法
  - ディスク、イメージ、スナップショットなど
- Key-Value の形式でユーザーが定義した文字列
- 課金を通して反映される
- タグはインスタンスにのみ適用
- ユーザー定義の文字列
- タグは主にネットワーキングに使用される（ファイアウォール ルールの適用）

ラベルとタグを混同しないことが重要です。

これまでに学んだように、ラベルは Key-Value の形式でユーザーが定義した文字列で、リソースの整理に使用されます。ラベルは、課金を通して反映させることができます。

一方、タグは、インスタンスのみに適用されるユーザー定義の文字列で、主にネットワーキング（ファイアウォール ルールの適用など）で使用されます。

ラベルの使用について詳しくは、この動画のリンク セクションをご覧ください  
[\[https://cloud.google.com/resource-manager/docs/using-labels\]](https://cloud.google.com/resource-manager/docs/using-labels)

# アジェンダ

---

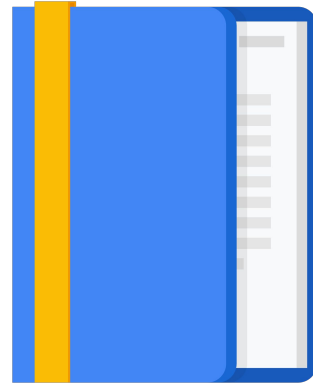
Resource Manager

割り当て

ラベル

課金

ラボ



プロジェクト下の全リソースの消費量は、1つの請求先アカウントに累積されるので、ここでは課金について見ていきます。

## 予算とメール通知アラート

### 1 範囲

名前 \*  
予算名

プロジェクト  
すべてのプロジェクト(2)

### 2 金額

予算タイプ  
指定額

費用の比較対象となる一定の金額です。

目標金額  
\$500

### 3 操作

予算の割合	金額	トリガー対象 ?
50 %	\$250	実値
90 %	\$450	実値
100 %	\$500	実値 予測

指定額

先月の利用額

プログラムによる予算: Cloud Pub/Sub → Cloud Functions

プロジェクトでの費用の計画と管理に役立てるため、予算を設定できます。予算を設定すると、その金額に対して利用額がどの程度増えているかを追跡できます。このスクリーンショットは、予算作成インターフェースを示しています。

- まず、予算名を設定して、その予算が適用されるプロジェクトを指定します。
- 次に、特定の金額の予算を設定したり、前月の利用額に合わせた予算を設定したりできます。
- 予算額を決定したら、予算のアラートを設定できます。これらのアラートによって、利用額が予算の一定の割合または指定額を超過した場合に、課金管理者にメールが送信されます。

この例では、利用額が予算額の 50%、90%、100% に達した場合にメールが送信されます。予算期間の終了時まで利用額が予算額の一定の割合を超過することが予想される場合にアラートを送信するかどうかを選択できます。

メールを受信するだけでなく、Cloud Pub/Sub 通知を使用して、この予算に関する利用額の更新情報をプログラムで受信できます。また、Pub/Sub トピックをリッスンする Cloud Functions 関数を作成して、費用管理を自動化することもできます。プログラムによる予算通知の例については、この動画のリンク セクションをご覧ください

[<https://cloud.google.com/billing/docs/how-to/notify>]。



## 予算アラートメールの例

### 請求アラート通知

Google をご利用のお客様

このメールは、Google Cloud Platform、Firebase、API をご利用のお客様にお送りしています。

この自動通知は、プロジェクト **arch-gce** が月額予算 **\$500.00** の **50%** を超過したことをお知らせするためのものです。

このメールは、このプロジェクトの予算に対してアラートを構成したお客様にお送りしています。このアラートを無効にするか、[予算の](#)しきい値を変更するには、[予算設定](#)を編集してください。

これは、メール通知の例です。このメールには、プロジェクト名、予算の超過している割合、予算額が含まれています。

## ラベルを使用した GCP の利用額の最適化



```
1 SELECT
2     TO_JSON_STRING(labels) as labels,
3     sum(cost) as cost
4 FROM `project.dataset.table`
5 GROUP BY labels;
```



RunQuery



クエリを保存



ビューを保存

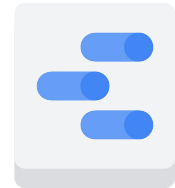
GCP の利用額を最適化するもう一つの方法は、ラベルを使用することです。たとえば、異なるリージョンに分散している VM インスタンスにラベルを設定できます。これらのインスタンスは、ほとんどのトラフィックを別の大陸に送信している可能性があります。その場合、高額のコストがかかります。このとき、一部のインスタンスを移動するか、Cloud CDNなどのキャッシュ サービスを使用して、コンテンツをユーザーの近くにキャッシュすることを検討するかもしれません。こうすることで、ネットワークのコストを削減できます。

すべてのリソースにラベルを設定し、料金データを BigQuery にエクスポートして利用額を分析することをおすすめします。BigQuery は、SQL や素早い応答時間を兼ね備えた、Google が提供するスケラブルでフルマネージドのエンタープライズ向けデータ ウェアハウスです。

このスクリーンショットに示されているように、クエリは簡単に作成できます。クエリの実行方法については、後のラボで説明します。

# データポータルを使った Google Cloud 利用額の可視化

## 課金ダッシュボード



Google データ  
ポータル

データポータルを使用して、一定期間の利用額を可視化することもできます。データポータルでは、データをダッシュボードやレポートの形式で有益な情報に変換して表示できます。こうした情報は見やすく簡単に共有できるうえ、全面的なカスタマイズも可能です。たとえば、ラベルを使用して課金レポートを詳細に分析できます。

# デモ

---

## 請求管理

Philipp Maier



次のラボでは、エクスポートされた課金データを確認します。課金データのエクスポート方法を紹介し、課金管理者が実行するほかの一般的な操作を実演します。これらの操作はセキュリティ制限により Qwiklabs 環境で実行できないため、デモとして紹介します。

[Demo]

GCP ではこのように課金を簡単に管理できます。課金管理者は、アカウントを設定してレポートを実行できます。これらはごく普通のタスクです。利用可能なオプションに慣れ、これらのタスクの実行方法を確認することで、混乱する可能性を低減できます。たとえば、レポートは JSON または CSV 形式で生成できることを学びました。

次のラボで学習するように、次は課金のエクスポート後にデータのより洗練された処理やフィルタリングを行います。

# ラボ

---

## BigQuery を使った課金 データの調査



BigQuery を使用して課金データを調べていきましょう。

このラボでは、BigQuery にログインしてデータセットを作成します。このデータセットでは、Cloud Storage バケットに保存されている課金データをインポートして、テーブルを作成します。次に、インポートされたデータで単純なクエリを実行した後、大規模なデータセットでより複雑なクエリを実行します。

# ラボの復習

---

## BigQuery を使った課金 データの調査

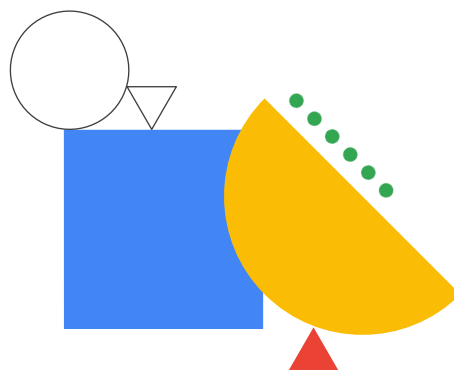


このラボでは、CSV ファイルとしてエクスポートされた課金データを BigQuery にインポートし、最初にそのデータにシンプルなクエリを実行しました。

その後、22,000 件を超える課金情報レコードがある共有データセットにアクセスし、このデータに対して各種のクエリを実行しました。これにより、BigQuery を使用してリソースの課金の消費に対する分析情報を取得する方法を学習しました。

日常的に BigQuery を使用する場合は、独自のクエリの開発を開始して、アプリケーション内でリソースが消費される場所を特定します。また、リソース消費の経時的な変化をモニタリングできます。このような分析はキャパシティ プランニングに対する入力であり、アプリケーションのスケールアップによる成長への対応や、スケールダウンによる効率性の向上の方法の判断に役立ちます。

## 復習: リソース管理



このモジュールでは、Cloud Resource Manager について学び、割り当て、ラベル、課金について確認しました。その後、ラボで BigQuery を使用して課金データを確認しました。

レポートは、リソース管理の重要な要素です。レポートを生成して消費量を追跡し、アカウントビリティを確立できます。Google Cloud の基本原則の一つに透明性があります。つまり、このモジュールで学習したように、消費データへのアクセスと処理を簡単に行うことができます。