Google Cloud

Kubernetes でクラウドを管理！ Config Controller で ガードレールを構築する

グーグル・クラウド・ジャパン合同会社

アプリケーション モダナイゼーション スペシャリスト

内間 和季

Table of Contents

なぜガードレールが必要なのか？ 01

Config Controller 概要 02

Config Controller を活用したガードレールの整備 03

Config Controller デモ 04

まとめ 05

01

なぜガードレールが必要なのか？

クラウド活用における**ガードレール**の必要性

クラウド活用が進み環境の規模が大きく複雑になってくると、すべての環境で適切な**セキュリティを担保**することが難しくなっていきます

開発における自由度やスケーラビリティを落とさず、セキュリティリスクを低減するためには、マニュアルによる運用ではなく**自動化された仕組みが必要**です

責任共有モデルとセキュリティ

- インフラ面のセキュリティは Google の責任範囲
- サービスの構成 / 設定等に関するセキュリティは **利用者側の責任範囲**
- 安全にクラウド環境を運用するためには、不適切な設定がされないよう**ガードレール**を設け、また設定が常に正しい内容になっているか **継続的な　監査** を行う必要がある

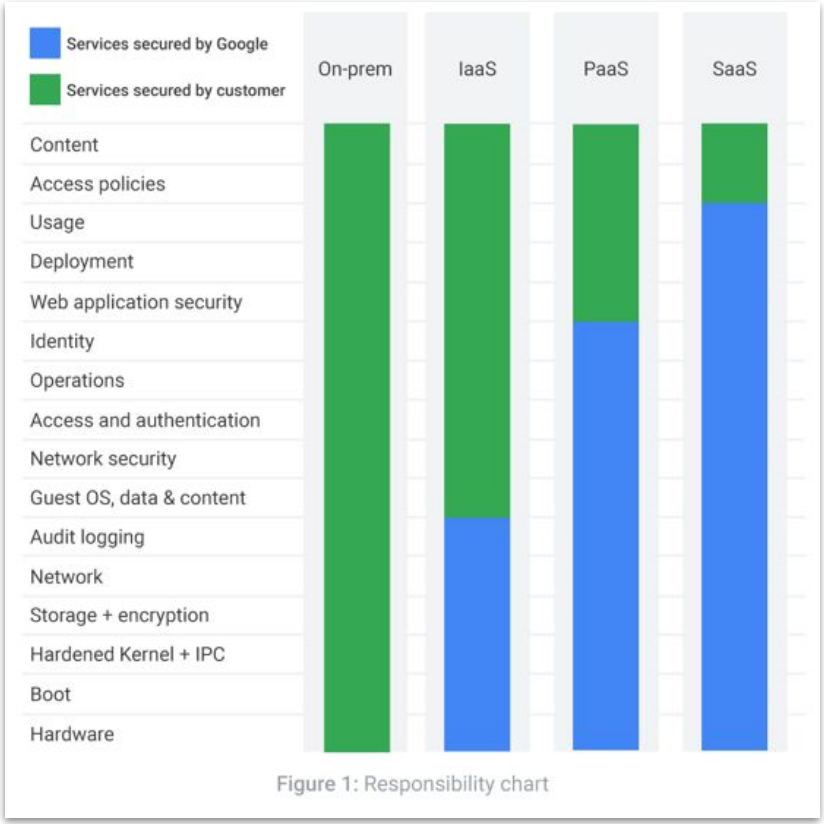


Figure 1: Responsibility chart

https://cloud.google.com/docs/security/incident-response

どのような機能が必要か？

予防

誤った設定が投入されるこ
とを**事前に防ぐ**

修正

誤って設定した内容を**正
しい状態へ修正**する

監査

構成ミスが存在しないか
継続的に監査する

Config Controller

予防

Policy Controller の制約による予防

修正

Kubernetes の **Reconciliation Loop** による自動修正

監査

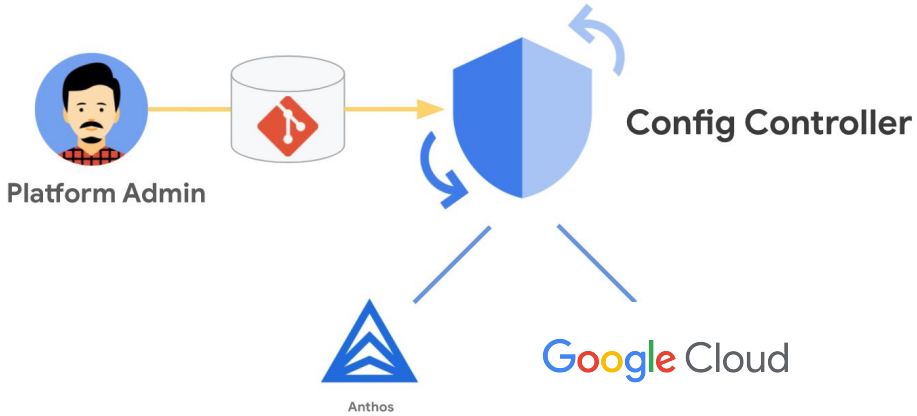
ルールに違反しているリソースを**監査ログ**として記録・検知

02

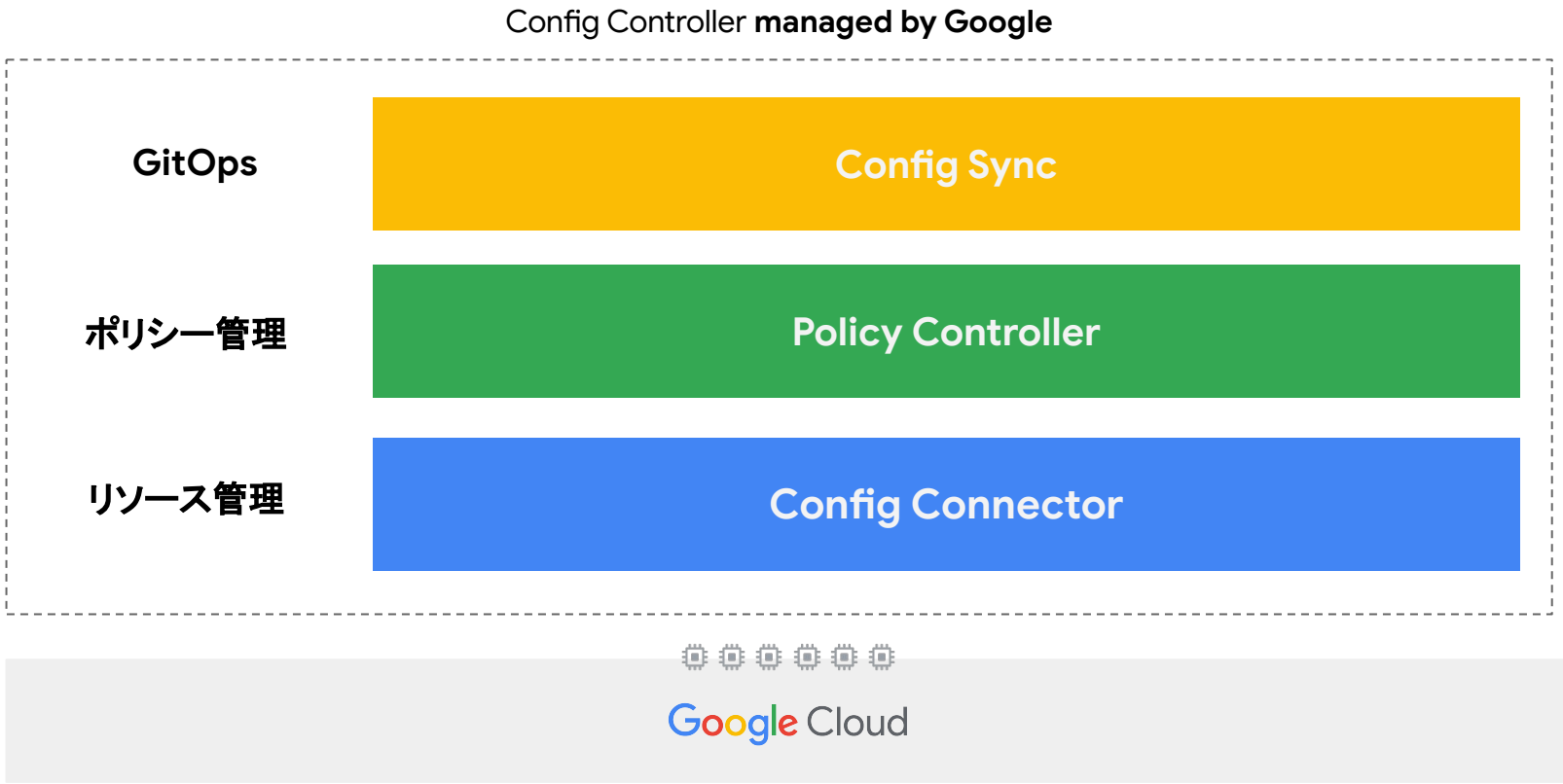
Config Controller 概要

Config Controller

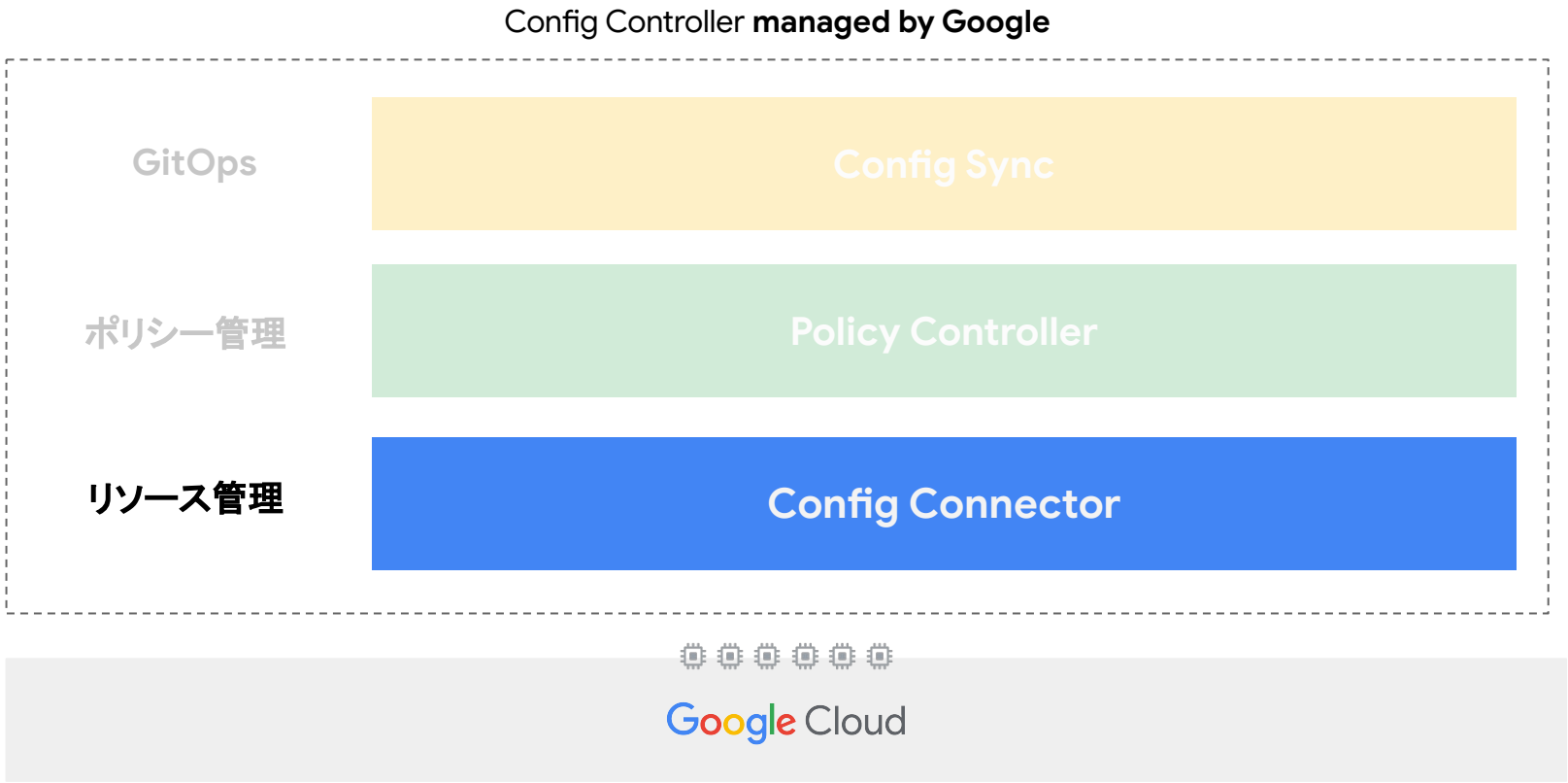
- Google Cloud リソースのプロビジョニングとオーケストレーションを行う **ホスト型サービス**
- Config Controller では、Kubernetes スタイルのシンプルな宣言型の構成を定義して使用可能
- Kubernetes のエコシステムやリソース管理の仕組みを **クラウドリソースの管理に適用**



Config Controller の構成要素

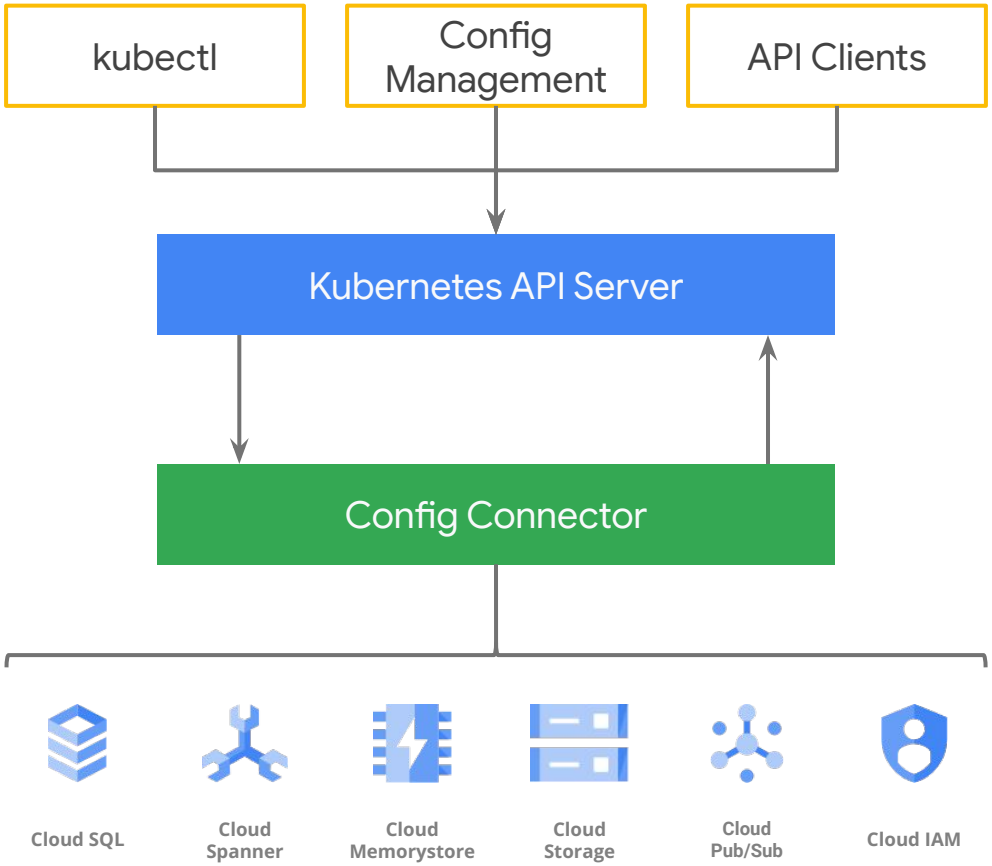


Config Controller - リソース管理



Config Connector

- Terraform / Deployment Manager などの Infrastructure as Code で実現するインフラ構築機能を **Kubernetes Resource Model (KRM)** を利用して実現
- リソースをプロビジョニングするための操作や手順を定義するのではなく、リソースの **理想の状態**を**データとして定義** する宣言型アプローチ (Configuration as Data) を　採用



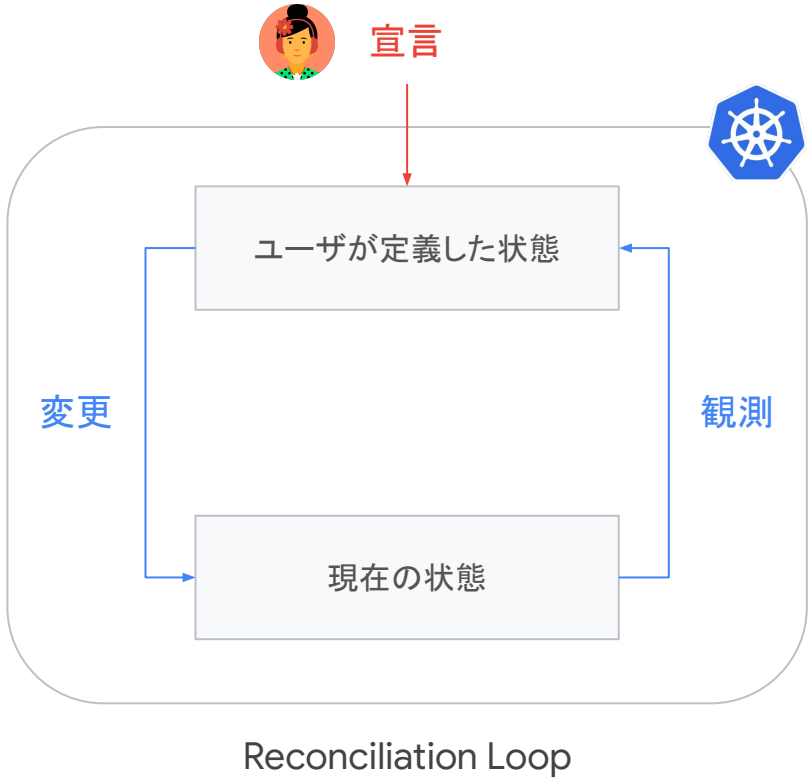
KRM によりクラウドリソースを管理することの利点

- Kubernetes の Reconciliation Loop という特徴を利用できる
  - 実際の状態を観測し、理想の状態（定義した内容）と比較
  - 理想の状態になるよう変更を繰り返す

  →継続的な適用（自動修正）を実現

- Kubernetes のエコシステムを活用できる
  - Kubernetes リソースに対する制約の設定
  - 構成管理・マニフェスト管理

  →開発と同じツールの利用、　　クラウドリソース管理の高度化



Config Connector によるリソースの作成 / 更新 / 削除

リソースの作成

(Pub/Sub インスタンスを作成する例)

```
apiVersion: pubsub.cnrm.cloud.google.com/v1beta1
kind: PubSubTopic
metadata:
  labels: LABEL_VALUE
  name: TOPIC_NAME
```

pubsub-topic.yaml

```
$ kubectl apply -f pubsub-topic.yaml
```

リソースの更新

(作成したリソースを更新)

```
apiVersion: pubsub.cnrm.cloud.google.com/v1beta1
kind: PubSubTopic
metadata:
  labels: NEW_LABEL_VALUE
  name: TOPIC_NAME
```

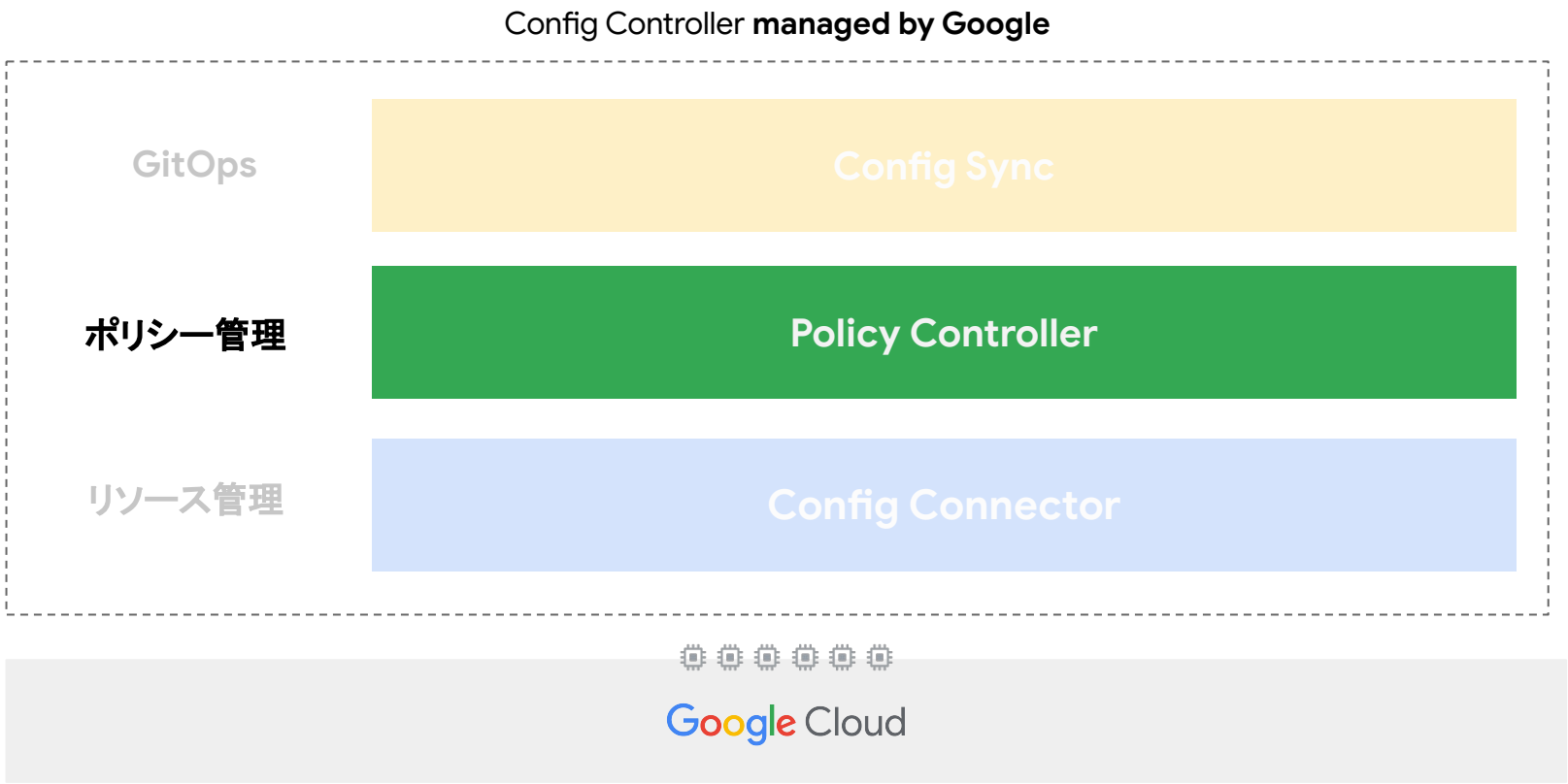
pubsub-topic.yaml

```
$ kubectl apply -f pubsub-topic.yaml
```

リソースの削除

```
$ kubectl delete -f pubsub-topic.yaml
```

Config Controller - ポリシー管理



Policy Controller

- Open Policy Agent (OPA) Gatekeeper がベース
- Constraints を用いてクラスタのコンプライアンスを強化
- Rego と呼ばれる Native Query Language を使ってポリシーを記述、テンプレートもあり

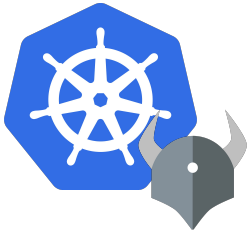
Policy Journey

Policy Blueprints
Google が作成した制約や、
カスタム制約がガードレールになる

サービス管理者

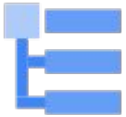
Authoring

デプロイ前に config の差分を確認



Admission

ポリシーに準拠していない変更を API でブロック



Audit

リアルタイムにコンプライアンス違反のアラート

プラットフォーム管理者

デフォルト ポリシー ライブラリ

- 45 + 制約テンプレート ライブラリが Policy Controller のデフォルトでインストール済
- ライブラリは Policy Controller チームにより継続的にメンテナンスされ、拡張される

ドメイン制限が設定された
ロールのみの利用許可

クライアント / サービス間での
相互 TLS 通信を強制

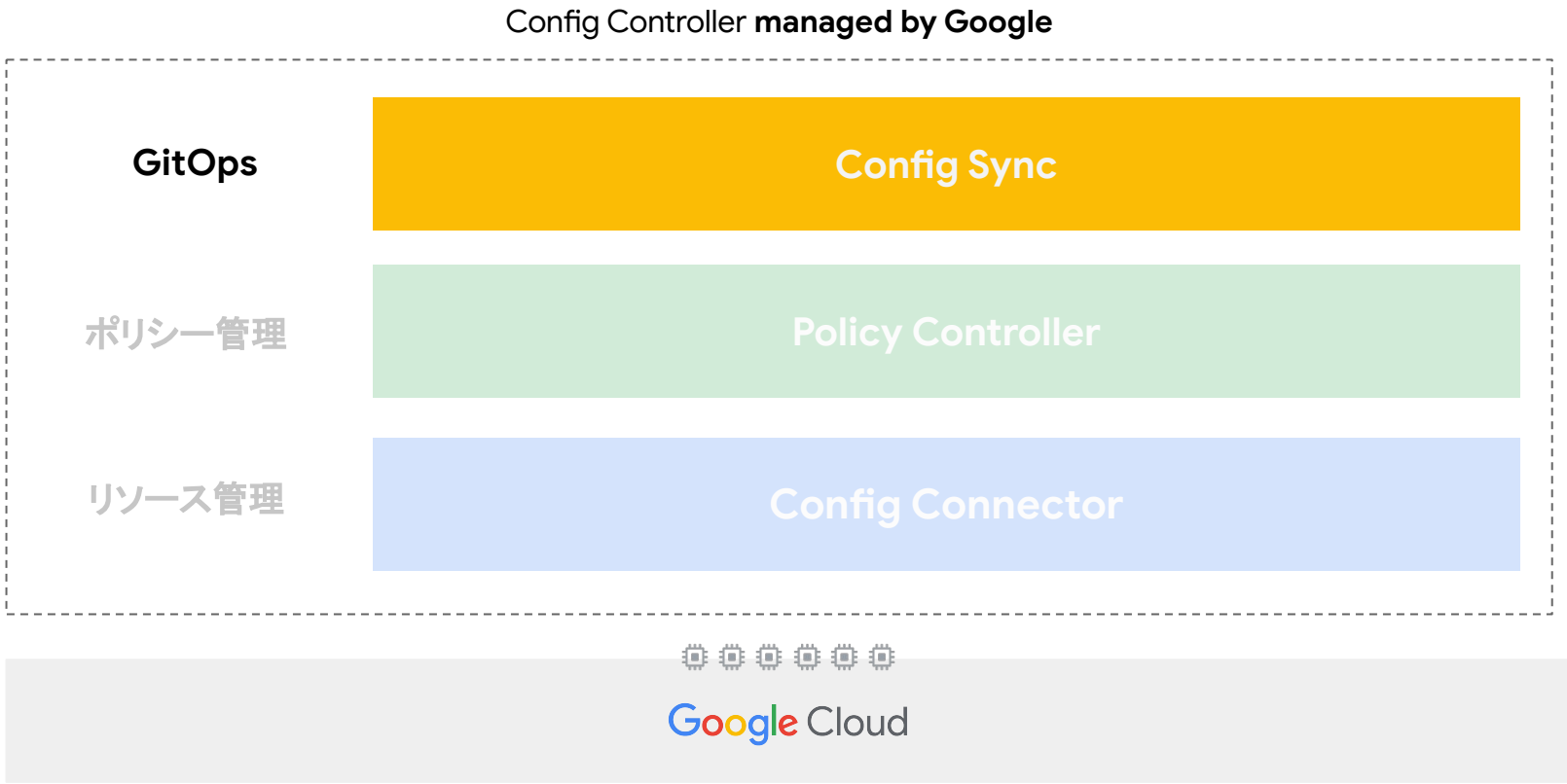
きめ細かいサービス認可管理
がなされていることを要求

コスト管理のために必要な
タグが必須とする制限

認証を不要とするサービスへの
アクセス許可を認めない

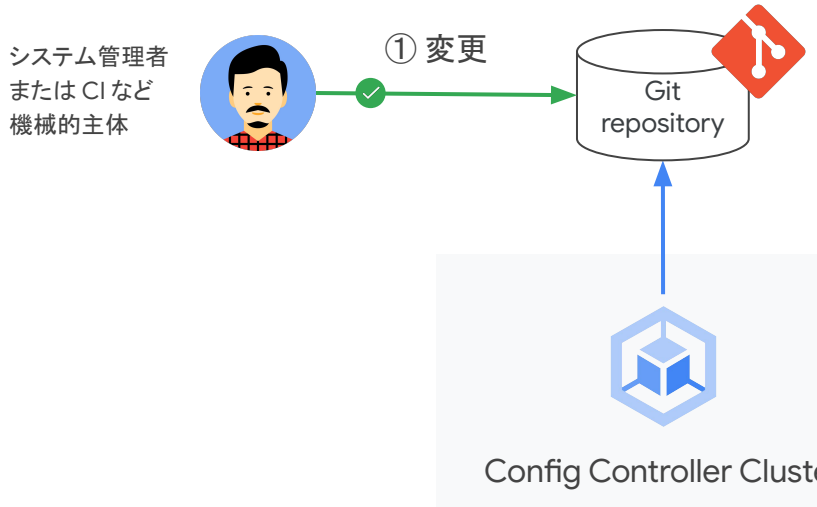
クラスタやサービスへの
アクセスログ有効化を必須に

Config Controller - GitOps



Config Sync によるクラスタ構成の一元管理・自動化

理想状態が **”維持される”**（ドリフトが起きない）仕組みを実現
設定には逐次適用ではなく、GitOps + Reconciliation による制御を採用



② クラスタは対象リポジトリを
定期的に確認、
実環境と設定に差分が発生したら、
その変更を自らのクラスタに反映

03

Config Controller を活用した ガードレールの整備

Google Cloud リソースに対するガードレールの設定方法

Google Cloud リソースに対する制約（ガードレール）を作成する例：

1. **組織ポリシー**
   a. Google Cloud リソースに対する制約を設定
   b. 事前に定義されたポリシーから選択

2. **Config Controller (Policy Controller)**
   a. Config Controller で管理している Google Cloud リソースに対する制約を設定
   b. ポリシーを自由にカスタマイズ可能

組織ポリシー

Google Cloud リソースの制限をかけることができる機能
ポリシーは組織レベルだけではなく、フォルダやプロジェクト単位で設定することが可能

組織ポリシーの例:

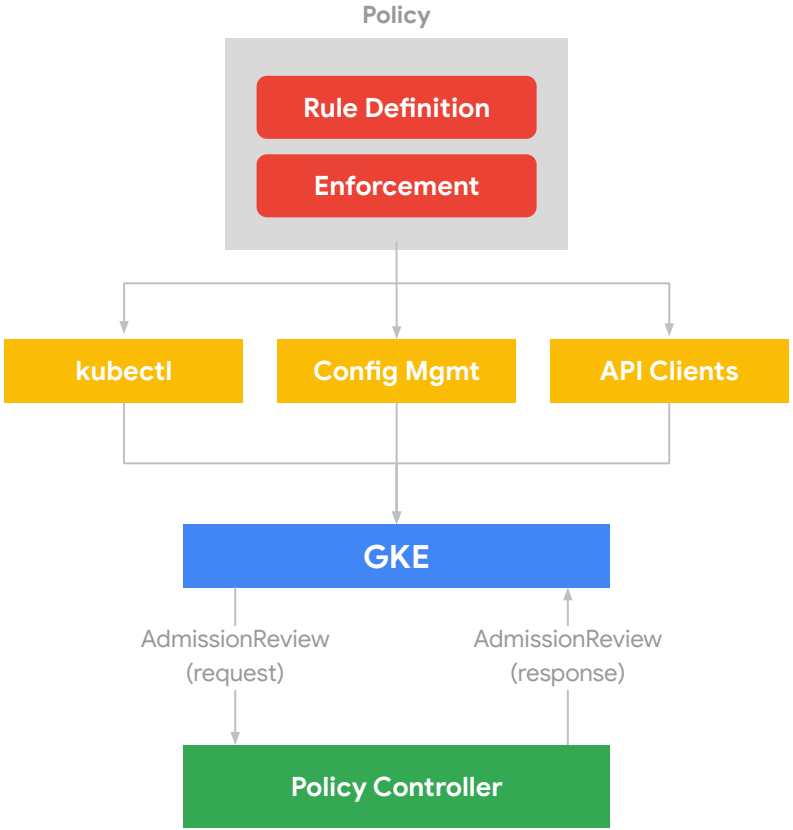
サービス	制約	概要
(ロケーションベースのサービス全般)	Google Cloud Platform - リソース ロケーションの制限	ロケーション ベースの Google Cloud リソースを作成できる一連のロケーションを定義
Cloud Storage	公開アクセスの防止を適用する	バケットの公開アクセスを禁止する
Cloud Storage	保持ポリシー期間(秒)	バケットに設定できる保持ポリシーの期間を定義
Compute Engine	VM インスタンス用に許可される外部 IP を定義する	外部 IP アドレスの使用が許可されている一連の Compute Engine VM インスタンスを定義
Cloud SQL	Cloud SQL インスタンスに対するパブリック IP のアクセスを制限する	パブリック IP アドレスの使用が許可されている Cloud SQL インスタンスを定義

Policy Controller による制約

Policy Controller で作成した制約を Google Cloud リソースの構成ファイルに対して適用することでガードレールを作成

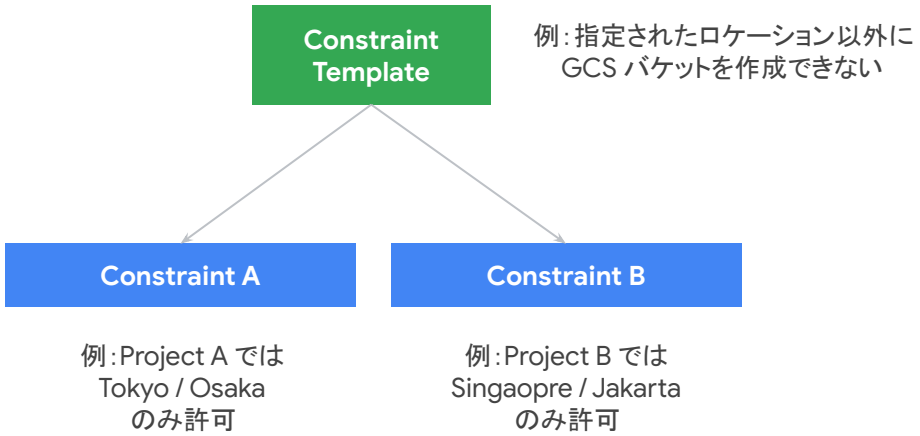
Rego という言語を利用し、組織ポリシーには存在しないような自由度の高い制約を作成することが可能

Config Controller (Config Connector) で管理しているリソースが対象



制約の構成要素

- Constraint Template (制約テンプレート)
  - 制約のスキーマとロジックを定義するリソース
  - ロジックは Rego で書かれる
- Constraint (制約)
  - 実際の制約を定義するリソース
  - 制約の対象となる API や Kind, Namespace (Project) 等を設定
  - 制約で利用するパラメータを設定



制約の構成要素

- Constraint Template (制約テンプレート)
  - 制約のスキーマとロジックを定義するリソース
  - ロジックは Rego で書かれる
- Constraint (制約)
  - 実際の制約を定義するリソース
  - 制約の対象となる API や Kind, Namespace (Project) 等を設定
  - 制約で利用するパラメータを設定

```
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: gcpstoragelocationconstraintv1
spec:
  crd:
    spec:
      names:
        kind: GCPStorageLocationConstraintV1
      validation:
        openAPIV3Schema:
          <スキーマの定義>
  targets:
  - target: admission.k8s.gatekeeper.sh
    rego: |
      <ロジックの記述>
```

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: GCPStorageLocationConstraintV1
metadata:
  name: tokyo-and-osaka-only
spec:
  enforcementAction: deny
  match:
    kinds:
    - apiGroups:
      - storage.cnrm.cloud.google.com
      kinds:
      - StorageBucket
    namespaces:
    - ns-japan
  parameters:
    <パラメータの定義>
```

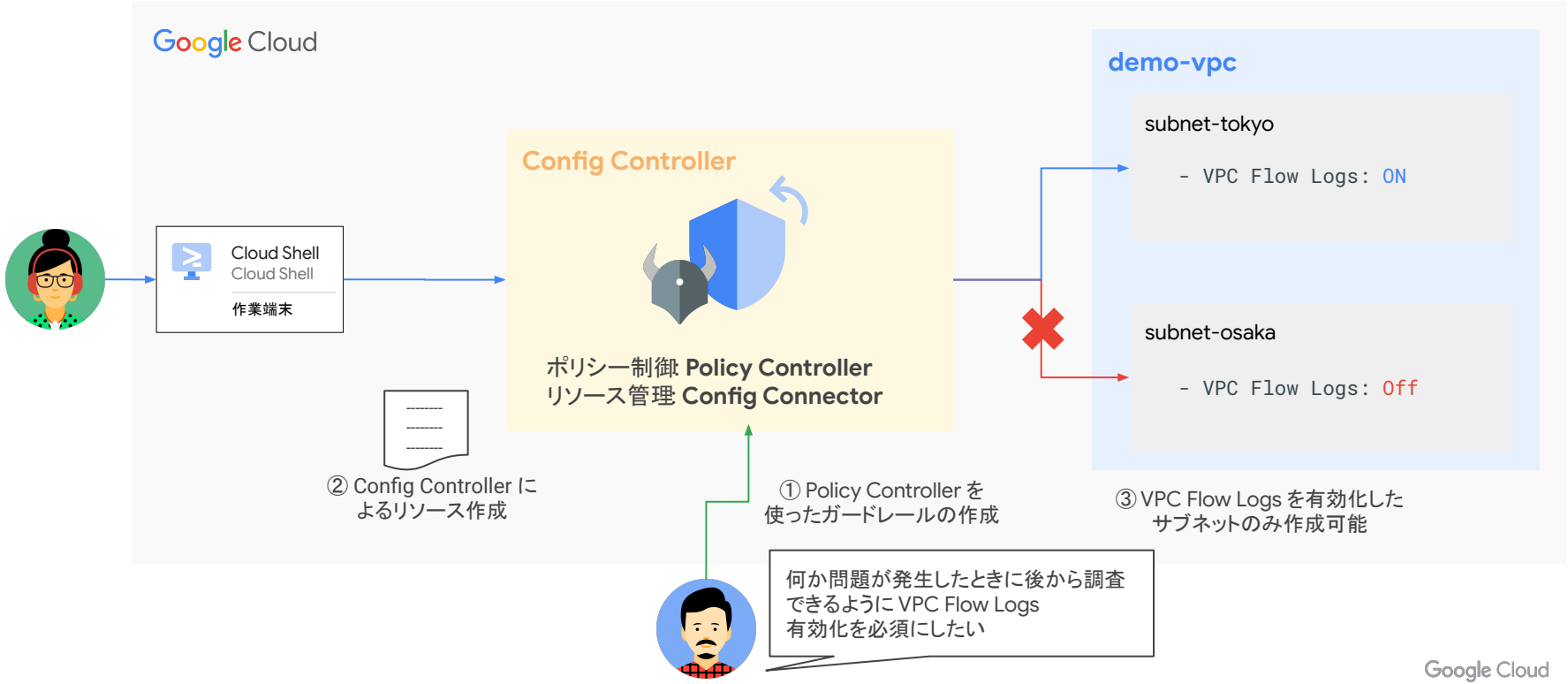
組織ポリシー と Config Controller の使い分けについて

- まずは組織ポリシーの適用を検討し、組織ポリシーではカバーできない要件がある場合に Config Controller など他の手段を検討
- 組織ポリシー自体を Config Controller で管理することで、誤って組織ポリシー自体を無効化する等、オペレーションミスにより発生するセキュリティリスクを低減することも可能に
- 監査目的の場合、Cloud Asset Inventory API や config-connector bulk-export を使って定期的にリソースを Import することで Config Controller で管理されていないリソースの制約違反を検知するこ とも可能

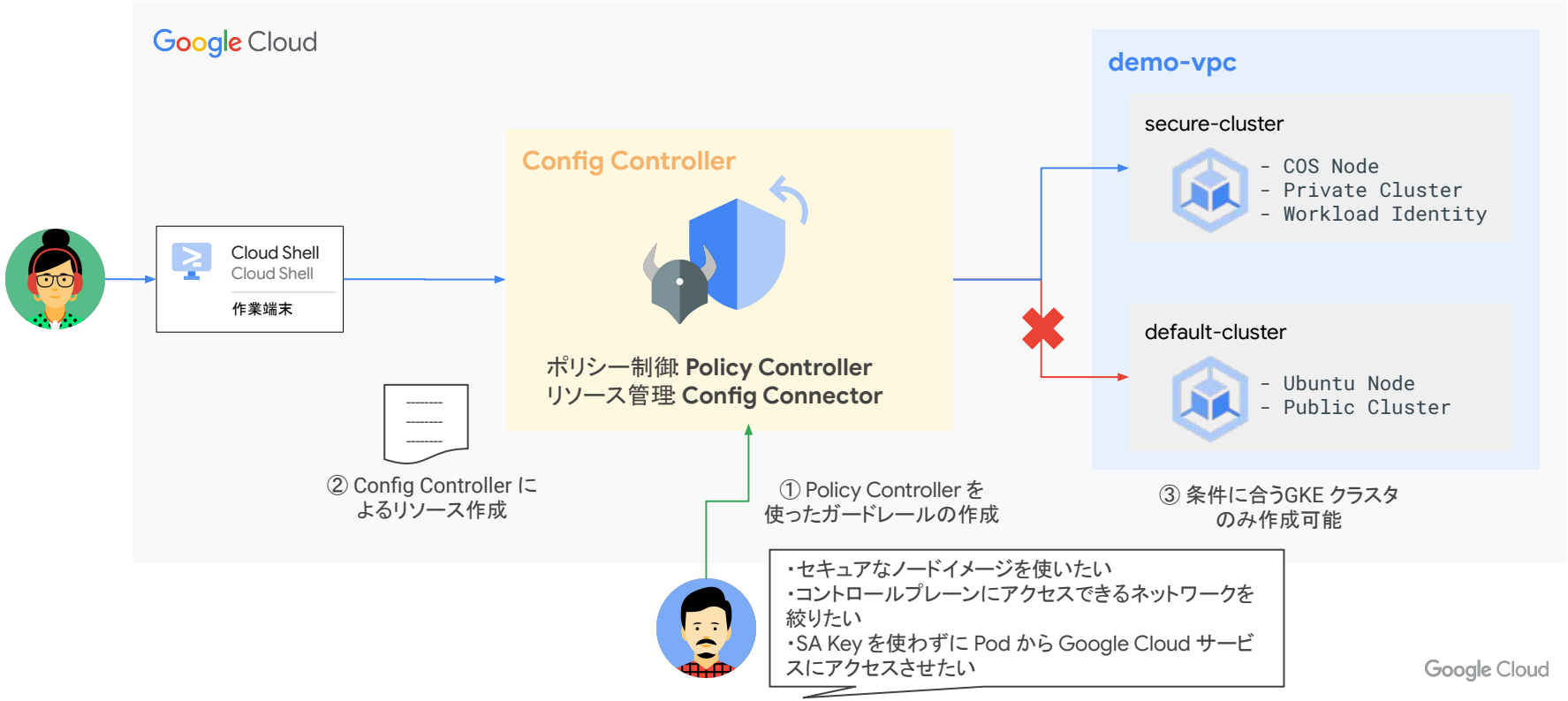
04

Config Controller デモ

デモの構成① VPC Flow Logs を有効化したサブネットのみ作成可能



デモの構成② セキュアな GKE クラスタのみ作成可能



05

まとめ

まとめ

- 開発における自由度やスケーラビリティを落とさず、セキュリティリスクを低減するためには、**ガードレール**のような仕組みが必要になる

- Config Controller を活用することで、Google Cloud 環境のガードレールを構築可能
  - **予防**... Policy Controller の制約
  - **修正**... K8s の Reconciliation Loop
  - **監査**... 監査ログによる記録・検知

Thank you.

Google Cloud