Google Cloud

Apigee X を利用した OAuth 実装と Terraform による IaC 管理のノウハウ

Agenda

自己紹介

谷口 友哉

クラウドエース株式会社
技術本部 システム開発部
SRE ディビジョン

大学卒業後、電子カルテを扱う医療系ソフトウェアベンダーで請求管理システムの開発を担当。数年後、社内研究室に配属となってからは幅広い技術領域にチャレンジし、特に深層学習のためのインフラ基盤の構築に携わる。パブリック クラウドの世界に進むため 2021 年にクラウドエース株式会社に入社。

加藤 大樹

クラウドエース株式会社
技術本部 システム開発部
SRE ディビジョン

大学院卒業後、海事関連のソフトウェアベンダーで船舶のモニタリングシステムの開発・運用・保守を担当。その後、マップ・ジオ系の知識を生かし，Google Maps Platform を活用したシステム開発等に従事し、2020 年に クラウドエース株式会社 に SRE エンジニアとして中途入社。入社後は、主にアプリケーションモダナイゼーション案件などを担当している。

バーゴス クリスチャン

クラウドエース株式会社
技術本部 システム開発部
Backend ディビジョン

CA のバックエンドディビジョンのバーゴスです。Google Cloud の経験は 4 年くらいです。取得している資格：DL、ACE、Data Eng.、Cloud Architect、Cloud Network Eng.、 DevOps Eng.、Cloud Security Eng.、Workspace Admin、ML Eng.、Cloud Dev です。これからも色々勉強したいと思います！よろしくお願いします！

Google Cloud

01

OAuth 2.0を使った Apigee セットアップ

どうして Apigee？

アナリティクス

- API トラフィックの時系列での傾向分析
- どの地域からのリクエストかの分析

セキュリティサポート

- API Key
  OAuth 2.0
  etc...

簡単なセットアップ

- 複雑なコーディングは不要
- 高速なデプロイ

お気に入りポイント

- API 管理を簡単に実現

“

API を限定公開したい時に認証/認可をどうやって実現するかを考えたことがありますか？ ”

バーゴスクリスチャン
クラウドエース株式会社・バックエンドエンジニア

Apigee を使った OAuth 2.0

実装は簡単です。設定とか複雑なコーディングが少ないので初心者でも実装ができます。

また、簡単な設定になっているので サービスに対して OAuth 2.0 によるセキュリティをかけたい場合も、ゼロから手動でコーディングをやらなくていいです。
その上、(レスポンスの一部を除いて) RFC 準拠になっているので信用もできます！

RFC 準拠ではないレスポンスの場合もある

OAuthV2

- “token_type”: “BearerToken”
- “expires_in”: “3600”（文字列）
- 期限切れのリフレッシュトークンのエラー

{"ErrorCode" : "invalid_request", "Error" :"Refresh Token expired"}

RFC 準拠

- “token_type”: “Bearer”
- “expires_in”: 3600 （数値）
- 期限切れのリフレッシュトークンのエラー

{"error" : "invalid_grant", "error_description" :"refresh token expired"}

デモ

- Apigee 上で OAuth 2.0 の実装前の動き
  - （認証がないでアクセス）
- Apigee 上で OAuth 2.0 の認証機能の実装
  - （ OAuth 2.0 認証をつける）
- Apigee 上で OAuth 2.0の実装後の動作
  - （認証が必要でのアクセス失敗）
- Apigee からアクセストークンを取得してアクセス
- Apigee 上のアナリティクス

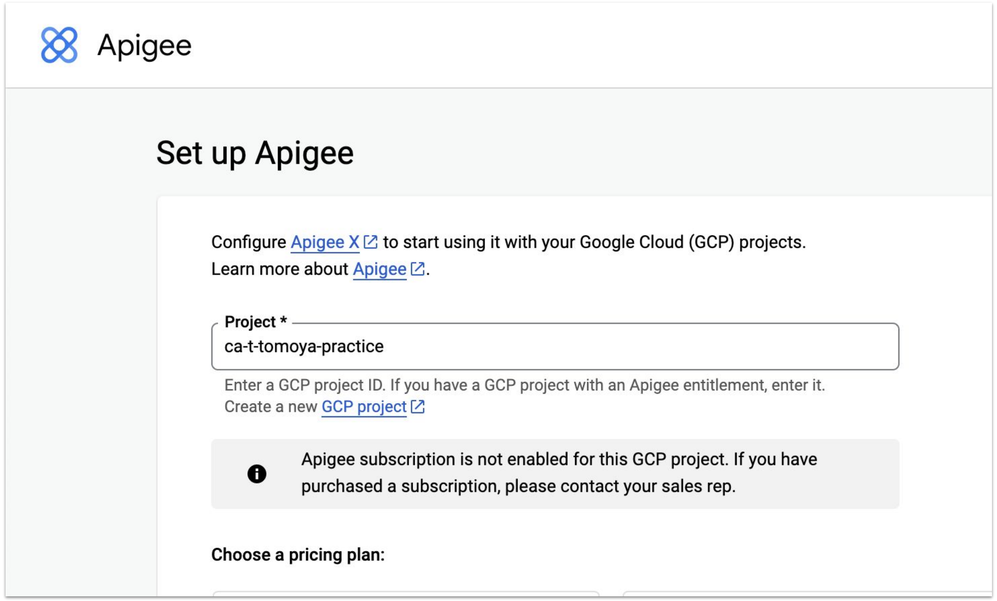
02

Terraform を用いた Apigee X 管理

Apigee X のインフラ構築

ウィザード

- Apigee プロビジョニング ウィザード



コマンドライン

- gcloud と Google API (REST API)

```
curl "https://apigee.googleapis.com/v1/organizations...
  -H "$AUTH" \
  -X POST \
  -H "Content-Type:application/json" \
  -d '{
    "name":"'"$PROJECT_ID"'",
    "analyticsRegion":"'"$ANALYTICS_REGION"'",
    "runtimeType":"CLOUD",
    "authorizedNetwork":"'"$NETWORK_NAME"'",
    "runtimeDatabaseEncryptionKeyName":"'"$RUNTIMED...
  }'
```

手動構築の課題

人の手で構築するため時間がかかる

- インフラの構築や各種設定を手動で行うため

作業ミスが発生した時の影響が大きい

- 規模が大きくなるにつれ、作業ミスによる影響が大きくなる

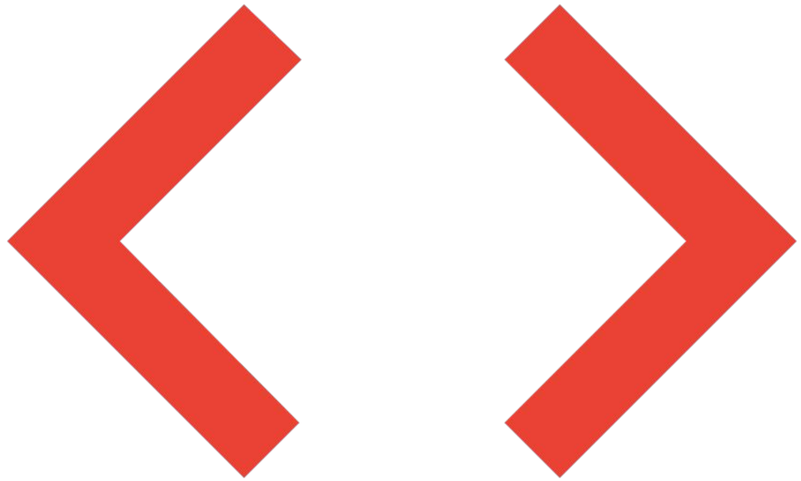
環境の変更管理がしにくい

- 複数環境を構築する場合による構成の管理が煩雑になる

IaC 管理のメリット デメリット

メリット

- インフラへの変更管理
- 検証環境、本番環境で同じコードで表現
- 変更に対するテストが容易



デメリット

- 学習コスト / 導入コスト
- 手動と比較して構築の手間が掛かる

Terraform を用いた Apigee X 管理

Apigee リソースの管理例

Apigee X を構成するリソース

Apigee X Organization（組織）

Apigee X の最上位のリソースで、Google Cloud プロジェクトと紐づいて作成される。
すべての API プロキシと関連リソースが含まれている。

ランタイム インスタンス

API プロジェクトと関連サービスが保存される仮想マシンリソースで、API 利用者から見た API エンドポイント。

Environment / Environment Group

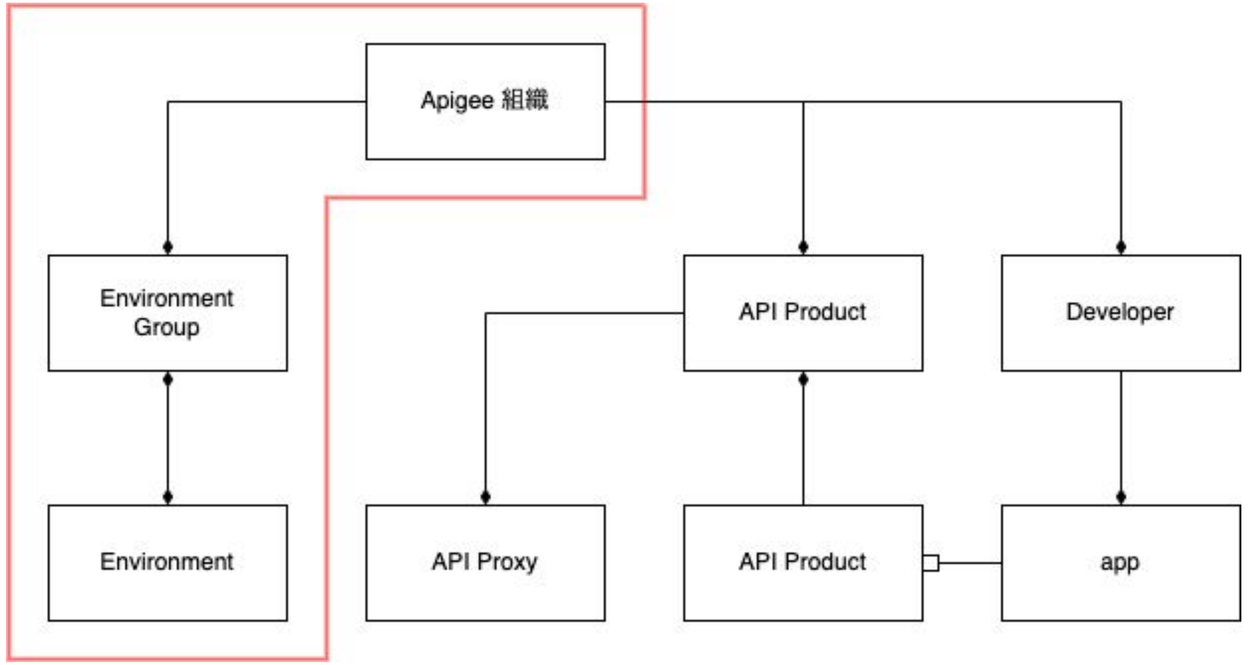
Apigee X 組織内の分離された環境を作るためのリソースで、API プロキシの作成とデプロイを行う。
単一または複数の環境をグループ化することが出来る。

NAT IP・IAM 許可ポリシー
API Proxy など

Apigee X がバックエンドに通信するときに使用される IP リソースを作成（予約）したり、Environment などのリソースに対する IAM 許可ポリシーを作成する。

Terraform で管理出来る Apigee X リソース

- Apigee 組織
- ランタイム インスタンス
- Environment Group
- Environment
- Environment の IAM ポリシー



Terraform による Apigee X 管理

Apigee X Organizationと

ランタイム インスタンスの管理例

- google_apigee_organization
- google_apigee_instance

```
# Apigee X 組織リソース
resource "google_apigee_organization" "apigee_org" {
  Analytics_region   = "asia-northeast1"
  display_name       = "apigee-org"
  description        = "Terraform-provisioned Apigee Org."
  project_id         = "project-id"
  authorized_network = "VPC Network"
  runtime_database_encryption_key_name = "CMEK"

  depends_on = [
    google_service_networking_connection.apigee_vpc_connection
  ]
}

# Apigee X ランタイム インスタンス リソース
resource "google_apigee_instance" "apigee_instance" {
  name     = "apigee_instance"
  location = "asia-northeast1"
  org_id   = google_apigee_organization.apigee_org.id
  ip_range = "..."
}
```

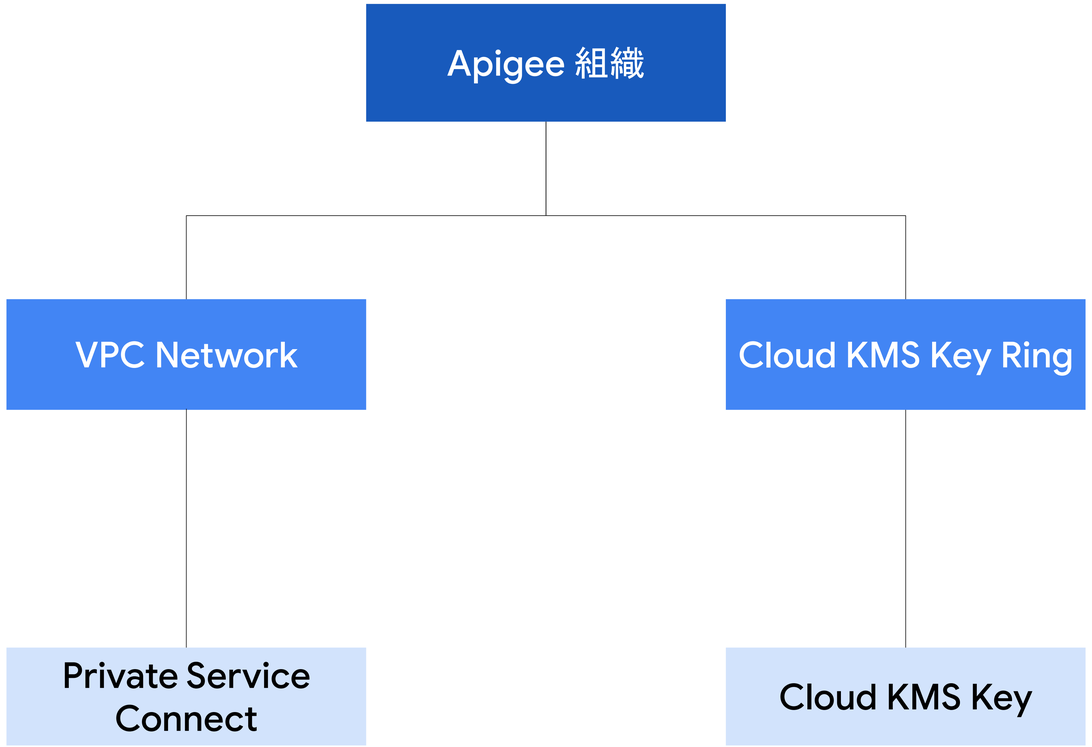
モジュールの構成

モジュールとは

お互いに関連する複数のリソースを 1 つの単位として管理するための仕組み

Apigee 組織 の関連リソース

- VPC Network
- Private Service Connect
- Cloud KMS Key Ring
- Cloud KMS Key



Terraform 管理時の注意点

Apigee リソースは、同時並行でリソースの作成などを行う事ができないため、Terraform で作成する場合は、並列処理を行わないように制御する必要がある

```
terraform apply -parallelism=1
```

まとめ

- Apigee X では、Auth0 の実装が簡単で速い
- Apigee X は、IaC を用いて管理することが可能
- Terraform での構築はマルチ処理ができないので注意が必要

Thank you.

Google Cloud