Google Cloud

インフラストラクチャのコード化と Compute Engine の運用管理をアップデート

栃沢 直樹

グーグル・クラウド・ジャパン合同会社
パートナー エンジニア

スピーカー自己紹介

栃沢 直樹
グーグル・クラウド・ジャパン合同会社
パートナーエンジニア

パートナー エンジニア
Infrastructure Modernization 担当

VMware vExpert（ 2017 - ）

日本ネットワークセキュリティ協会
デジタルアイデンティティ WG サブスクライバ

このセッションのゴール

~~技術的な Deep Dive~~

- よりクラウドっぽくインフラを運用するための方法を知って、試してみようと思っていただく

ビジネスに必要なもの

More customer value

More quickly

Lower cost

Less risk

More infra choices

モダンなクラウド運用

Simplify:
パブリッククラウド、データセンター、エッジなど、
あらゆる場所へ展開

Accelerate:
開発速度と安全性 / コンプライアンスの
トレードオフを排除

Scale:
数百、数千のチームの組織規模と成長に
対応する将来性

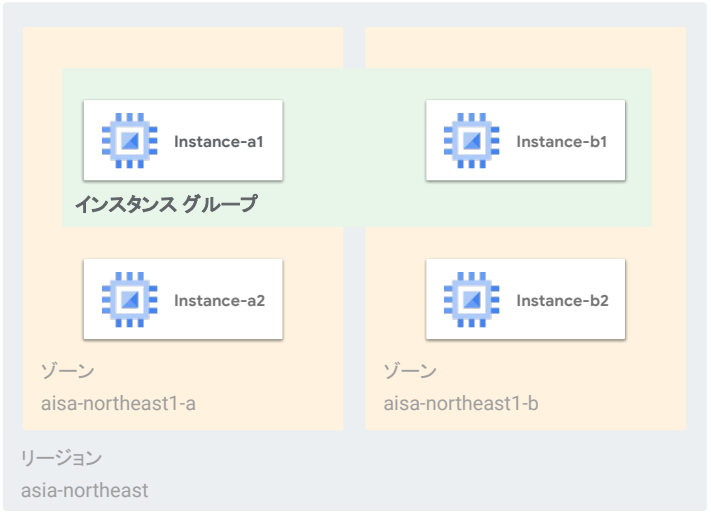
「仕組み」を上手に使って「モダンなクラウド運用」

システム構成の「統合管理」

Management

システム構成の「コード化」
または「データ化」

Deploy



Google Cloud

システム構成の「統合管理」

コンピューティング リソースの移行ステップ

Migrate for Compute Engine

Discover
既存システムの見える化

1

2

Estimate / Assess
コスト / 移行計画

Migrate
ワークロードの移行

3

4

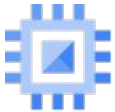
Operation
ワークロードの管理

STRATOZONE®

Cloud Monitoring

Cloud Logging

VM Manager : Google Compute Engine をより深く管理



複雑さを簡素化して
大規模な環境でも
Google Cloud の
マネージド サービス で
より管理しやすく

OS Config Agent

1  **OS Inventory Management**

2  **OS Configuration Management**

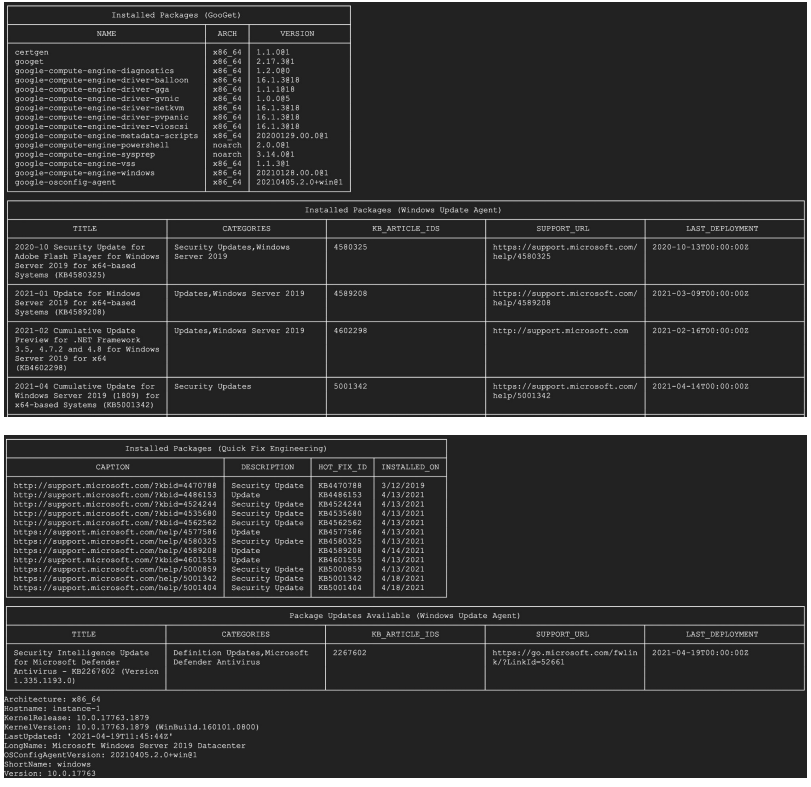
3  **OS Patch Management**

OS Inventory Management

次の情報を一元取得

- インスタンスの OS バージョン
- インスタンスにインストールされている
  パッケージ情報
- 各インスタンスで使用可能なパッケージ
  更新一覧
- インスタンスにインストールされていない
  パッケージや更新プログラム

部門ごとに管理されているインスタンスに
対してもすべてのアセット情報を部門を跨いで管理
することでガバナンスを強化



Installed Packages (GooGet)

NAME	ARCH	VERSION
certgen	x86_64	1.1.0@1
googet	x86_64	2.17.3@1
google-compute-engine-diagnostics	x86_64	1.2.0@0
google-compute-engine-driver-balloon	x86_64	16.1.3@18
google-compute-engine-driver-gga	x86_64	1.1.1@10
google-compute-engine-driver-gvnic	x86_64	1.0.0@5
google-compute-engine-driver-netkvm	x86_64	16.1.3@18
google-compute-engine-driver-pvpanic	x86_64	16.1.3@18
google-compute-engine-driver-vioscsi	x86_64	16.1.3@18
google-compute-engine-metadata-scripts	x86_64	20200129.00.0@1
google-compute-engine-powershell	noarch	2.0.0@1
google-compute-engine-sysprep	noarch	3.14.0@1
google-compute-engine-vss	x86_64	1.1.3@1
google-compute-engine-windows	x86_64	20210128.00.0@1
google-osconfig-agent	x86_64	20210405.2.0+win@1

Installed Packages (Windows Update Agent)

TITLE	CATEGORIES	KB_ARTICLE_IDS	SUPPORT_URL	LAST_DEPLOYMENT
2020-10 Security Update for Adobe Flash Player for Windows Server 2019 for x64-based Systems (KB4580325)	Security Updates,Windows Server 2019	4580325	https://support.microsoft.com/help/4580325	2020-10-13T00:00:00Z
2021-01 Update for Windows Server 2019 for x64-based Systems (KB4589208)	Updates,Windows Server 2019	4589208	https://support.microsoft.com/help/4589208	2021-03-09T00:00:00Z
2021-02 Cumulative Update Preview for .NET Framework 3.5, 4.7.2 and 4.8 for Windows Server 2019 for x64 (KB4602298)	Updates,Windows Server 2019	4602298	http://support.microsoft.com	2021-02-16T00:00:00Z
2021-04 Cumulative Update for Windows Server 2019 (1809) for x64-based Systems (KB5001342)	Security Updates	5001342	https://support.microsoft.com/help/5001342	2021-04-14T00:00:00Z

Installed Packages (Quick Fix Engineering)

CAPTION	DESCRIPTION	HOT_FIX_ID	INSTALLED_ON
http://support.microsoft.com/?kbid=4470788	Security Update	KB4470788	3/12/2019
http://support.microsoft.com/?kbid=4486153	Update	KB4486153	4/13/2021
http://support.microsoft.com/?kbid=4524244	Security Update	KB4524244	4/13/2021
http://support.microsoft.com/?kbid=4535680	Security Update	KB4535680	4/13/2021
http://support.microsoft.com/?kbid=4562562	Security Update	KB4562562	4/13/2021
https://support.microsoft.com/help/4577586	Update	KB4577586	4/13/2021
https://support.microsoft.com/help/4580325	Security Update	KB4580325	4/13/2021
https://support.microsoft.com/help/4589208	Update	KB4589208	4/14/2021
http://support.microsoft.com/?kbid=4601555	Update	KB4601555	4/13/2021
https://support.microsoft.com/help/5000859	Security Update	KB5000859	4/13/2021
https://support.microsoft.com/help/5001342	Security Update	KB5001342	4/18/2021
https://support.microsoft.com/help/5001404	Security Update	KB5001404	4/18/2021

Package Updates Available (Windows Update Agent)

TITLE	CATEGORIES	KB_ARTICLE_IDS	SUPPORT_URL	LAST_DEPLOYMENT
Security Intelligence Update for Microsoft Defender Antivirus - KB2267602 (Version 1.335.1193.0)	Definition Updates,Microsoft Defender Antivirus	2267602	https://go.microsoft.com/fwlink/?LinkId=52661	2021-04-19T00:00:00Z

```
Architecture: x86_64
Hostname: instance-1
KernelRelease: 10.0.17763.1879
KernelVersion: 10.0.17763.1879 (WinBuild.160101.0800)
LastUpdated: '2021-04-19T11:45:44Z'
LongName: Microsoft Windows Server 2019 Datacenter
OSConfigAgentVersion: 20210405.2.0+win@1
ShortName: windows
Version: 10.0.17763
```

OS Configuration Management

あらかじめ設定したポリシーに則り、インスタンスへのソフトウェアのインストール、削除、更新をマネージドサービスとして提供

- 定期的にポリシーに則っているかを確認
- ポリシーで定義された “あるべき姿”と差異がある場合には OS 標準のパッケージマネージャー (apt / yum install など) を利用して修正
- OS やラベル、リージョンなどによってゲストポリシーの適用対象となるインスタンスを制御

OS ポリシーを作成に活用できるサンプルコードも提供
https://cloud.google.com/compute/docs/os-configuration-management/working-with-os-policies#example-4_1

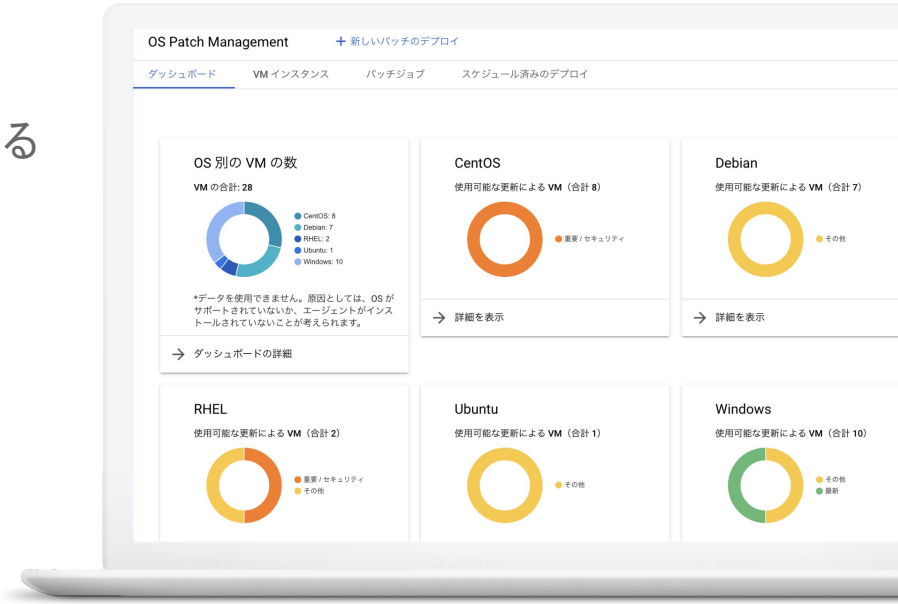
OS Patch Management / Patch Compliance Reporting

インスタンスにおけるOS の更新作業を簡略化し、
OS を脆弱性などの脅威から保護

更新を適用するOS をはじめ、次のパラメータを定めることで煩雑な更新作業が一括管理可能

- スケジュールや実行時間
  (例：60 分でタイムアウト)
- 一度に更新を適用するインスタンスの割合
- 更新前後に実行するスクリプト
- 更新後の再起動有無

更新の適用状況はOS ごとに一括して確認可能



VM Manager リファレンス

https://cloud.google.com/compute/docs/vm-manager

システム構成の
「コード化」または「データ化」

「コード化」「データ化」のメリット

構築工数の削減

再現性

設定ミスの防止

設定内容の可視化

継続的な活用によるバージョン管理

人為的な作業に頼らなくても

良い部分を仕組みとして実装

コード化　: Infrastructure as Code

データ化　: Configuration as Data

Infrastructure as Code

システムをそれぞれ手動で設定するのではなく、**「目指すべき構成」**を**「コード」**として定義する

コードで管理	ソースコードのように構成を扱う
デプロイの自動化	インフラストラクチャの目指すべき構成を自動化により再現
監査性	コードのバージョン管理とデプロイ時の状態の管理

Infrastructure as Code を実現するツール

Deployment Manager

Google Cloud のサービス・サポートあり

プロプライエタリ

ステートは Google 管理 (hosted)

Google Cloud 環境での利用

Terraform

CLI 実行

オープンソース

ステートは ローカル / GCS で自己管理

マルチクラウド、ハイブリッド クラウド での利用

Terraform with Google Cloud

オープンソース
※有償版あり

インフラ リソースの
プロビジョニング

マルチクラウド
ハイブリッド クラウド

コード化による
共用性とミスの削減

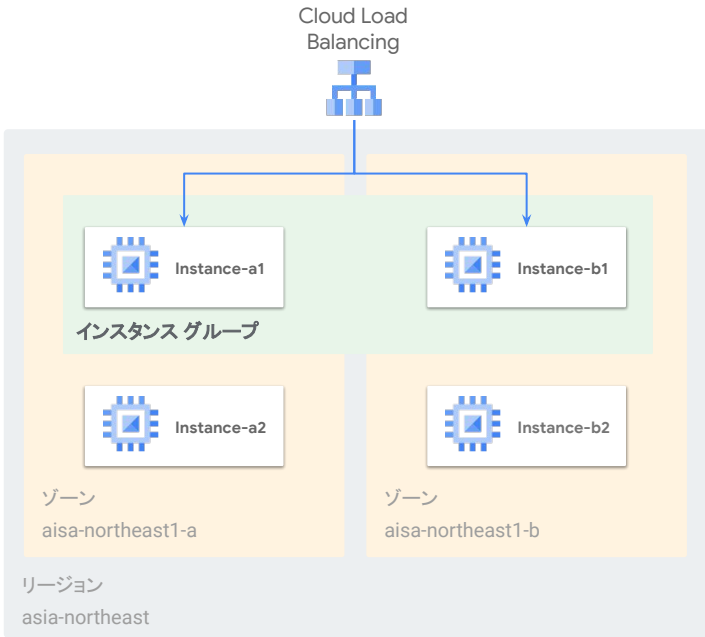
Modules

- main.tf
- variables.tf
- outputs.tf
- terraform.tfvars

```
module "gce-lb-http" {
  source          =
"github.com/GoogleCloudPlatform/terraform-google-lb-http"
  name            = "group-http-lb"
  target_tags     = ["${module.mig1.target_tags}",
"${module.mig2.target_tags}"]
  backends        = {
    "0" = [
      { group = "${module.mig1.instance_group}" },
      { group = "${module.mig2.instance_group}" }
    ],
  }
  backend_params    = [
    # ヘルスチェック パス, ポート名, ポート番号, タイムアウト（秒）
    "/,http,80,10"
  ]
}
```

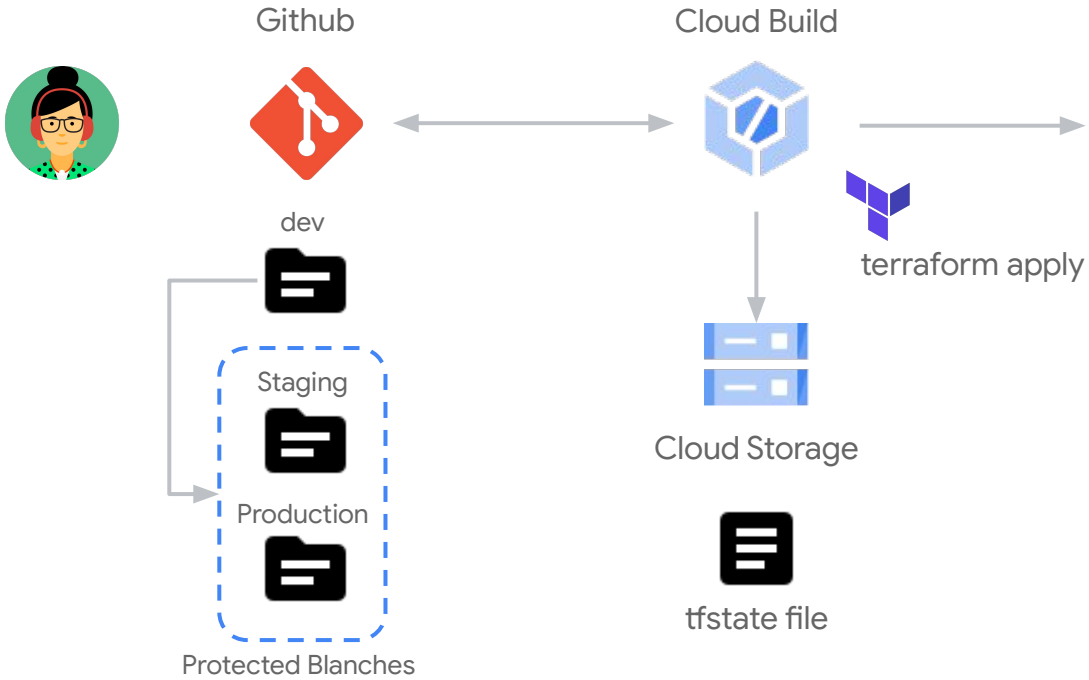
Plan → **Deploy**

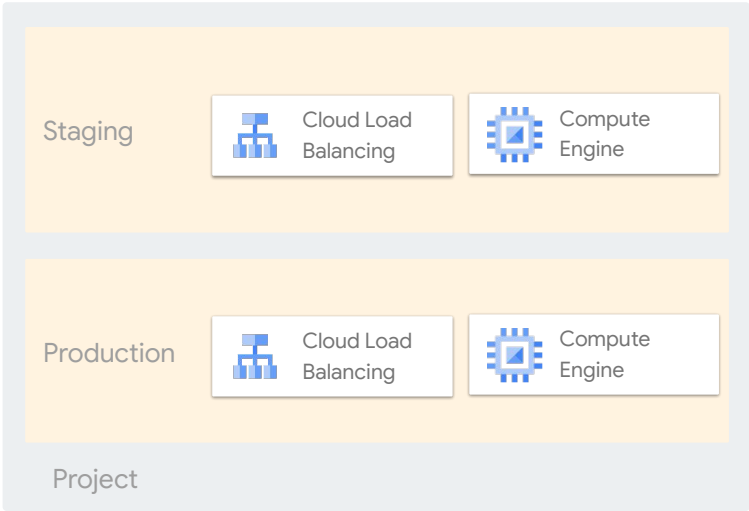
```
$ terraform plan

$ terraform apply
```



Cloud Build を活用したTerraform 環境

- **Pull Request によるコードレビュー、承認プロセスの確立**
- **オペレーションミス、組織としての対応を実現**





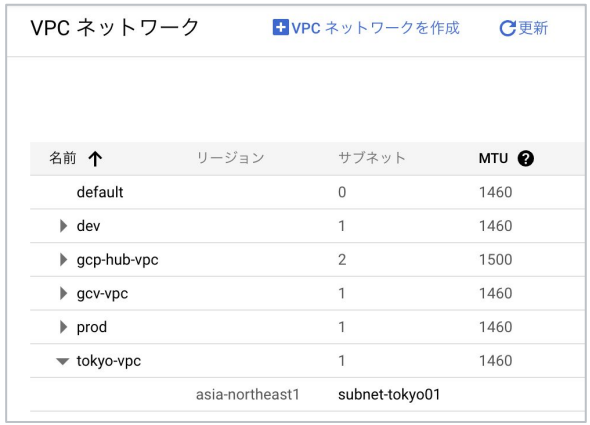
Google Cloud

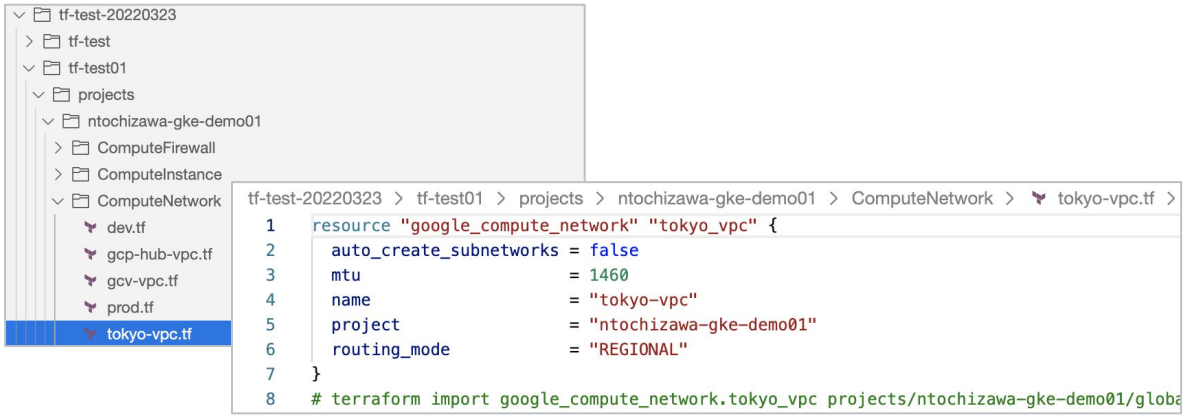
Public Preview

既存リソースから Terraform コードをエクスポート

Google Cloud CLI を利用して、デプロイされているGoogle Cloud リソースから Terraform 形式でエクスポート

gcloud beta resource-config bulk-export --resource-format=terraform --path=./ --project=ntochizawa-gke-demo01 --resource-types=ComputeNetwork,ComputeFirewall,ComputeInstance





- **より簡単に Terraform コードを作成**
- **本番環境と設計当初の状態の差異を解消**

Terraform と gcloud CLI を使用した完璧なGoogle Cloud インフラストラクチャの構築
gcloud beta resource-config bulk-export

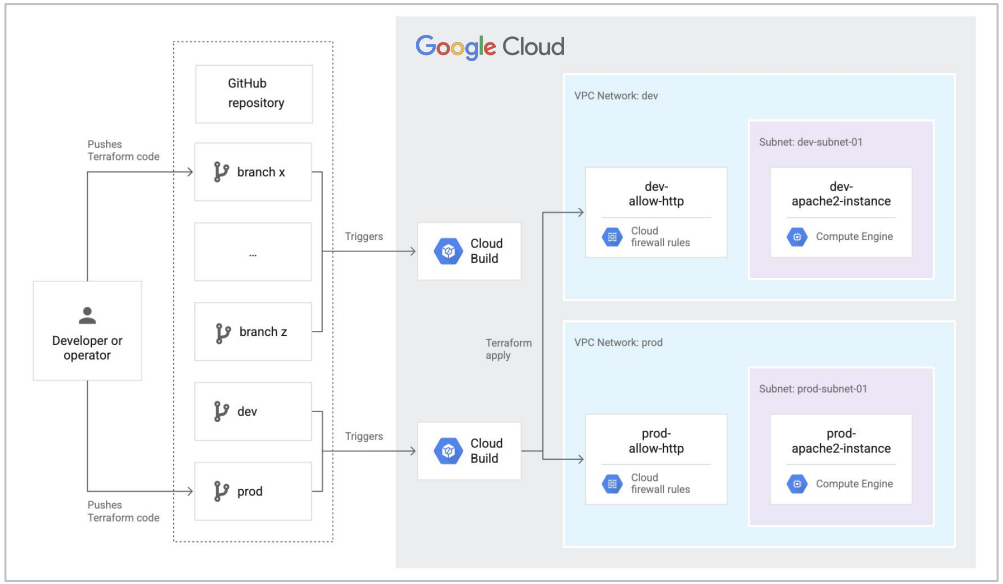
Google Cloud

Terraform with Google Cloud リファレンス

Cloud Build で Terraform デプロイのスケーリングとコンプライアンスを実現

https://cloud.google.com/blog/ja/products/devops-sre/terraform-gitops-with-google-cloud-build-and-storage

Terraform、Cloud Build、GitOps を使用してインフラストラクチャをコードとして管理する



https://cloud.google.com/architecture/managing-infrastructure-as-code

Configuration as Data

インフラやアプリの**「望ましい状態」**を**「データ」**として定義し、デプロイ、管理する
宣言型アプローチ

データで管理	「望ましい状態」を「データ」として定義
状態の監視	定義したデータと実際の環境との差異を観測
状態の復元	「望ましい状態」と実際の環境の差異があった場合に自律的に復元

- Google では宣言した**状態が 恒久的に 維持される仕組み**を併用するアプローチを推奨
- データなので継続的に検査＆検証しやすい

Google Cloud リソースを Configuration as Data で管理

インフラ ストラクチャの構築機能

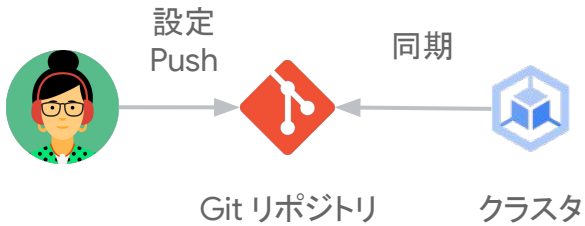
Kubernetes Resource Model（KRM）

- Kubernetes のデプロイの仕組みを利用するため、汎用的に利用ができる
  - Kubernetes 以外の **Google Cloud の各リソースを管理できる**
- コードの依存関係を極力意識せず、パラメータのみを設定
  - Kubernetes 初心者でも**リファレンス、Blueprint を活用**できる

定義したデータの管理機能

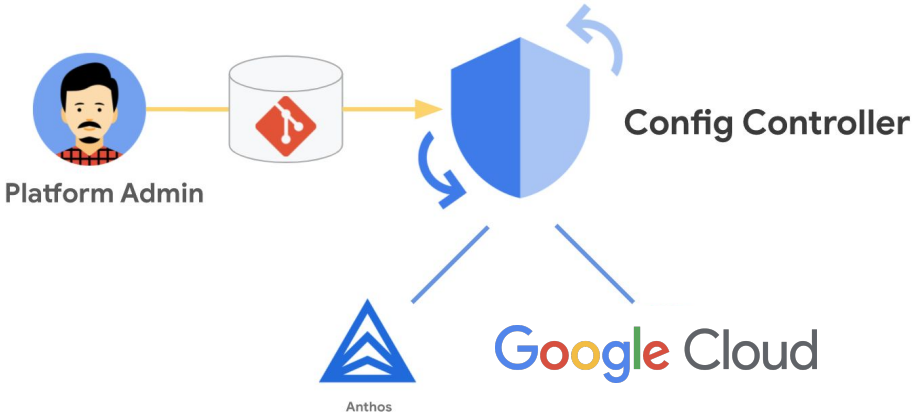
GitOps

- Git リポジトリの情報を信頼できる唯一の情報としてデプロイの再現性、完全性を担保
- バージョニング、複製についても適切に管理できる
- **CI / CD の既存のパイプライン**との親和性



Config Controller とは

- 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



- **Kubernetes スタイルのシンプルな**
  宣言型の構成を定義して使用
  - Kubernetes エコシステム、リソース管理の仕組みを
    クラウドリソース管理に適用

- Config Controller の基盤として
  Google Kubernetes Engine (GKE) クラスタを構成

Config Controller を構成するコンポーネント

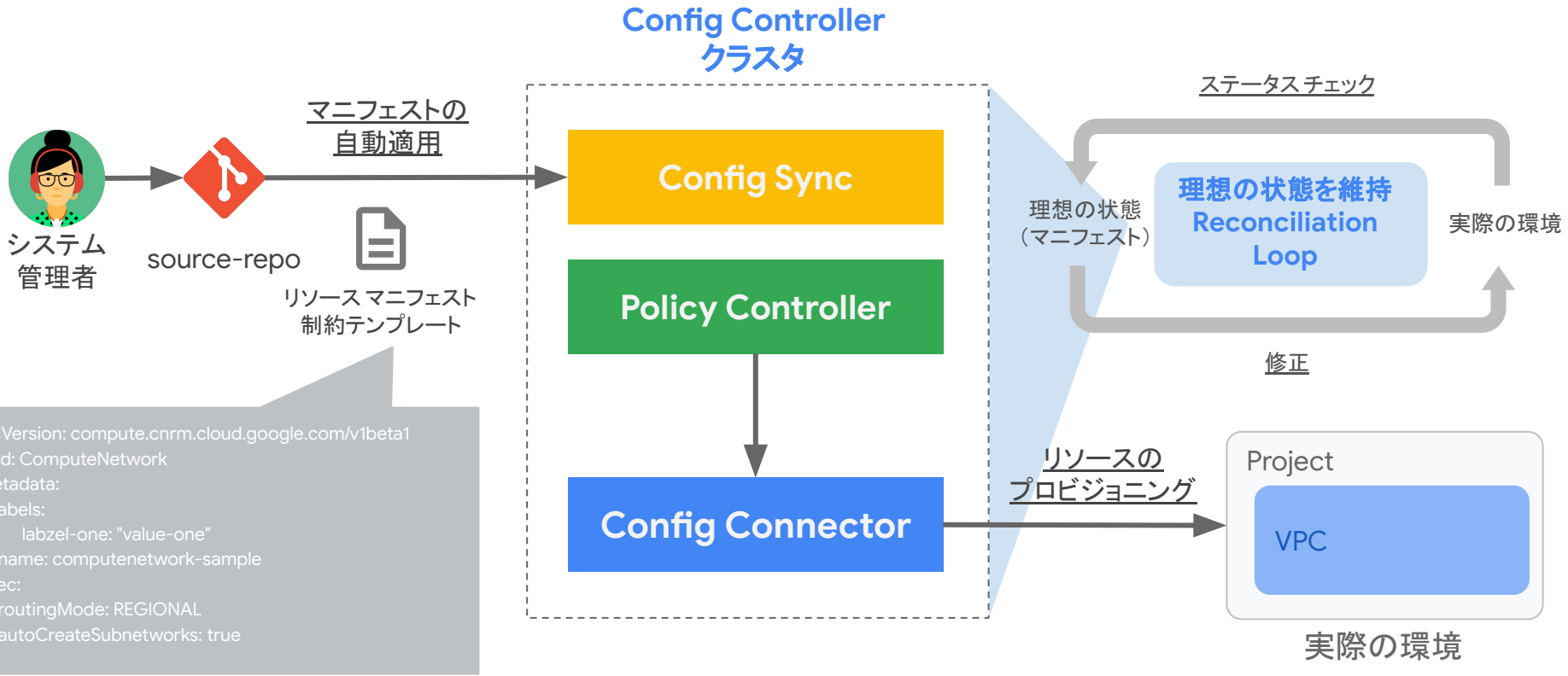
Config Controller managed by Google

GitOps	Config Sync	リポジトリの監視、変更を検出した際に（マニフェストの作成や更新等）Config controller クラスタにマニフェストを自動適用
ポリシー管理	Policy Controller	Google Cloud 既定の制約テンプレート、カスタムテンプレートによりセキュリティ ガードレールを設定
リソース管理	Config Connector	マニフェストの内容を「理想の状態」として実環境との差分をチェック 差分がある場合はマニフェストの内容に修正

Google Cloud

GKE に Config Controller クラスタを構成

Config Controller によるGoogle Cloud リソースの管理



Configuration as Data + GitOps のメリット

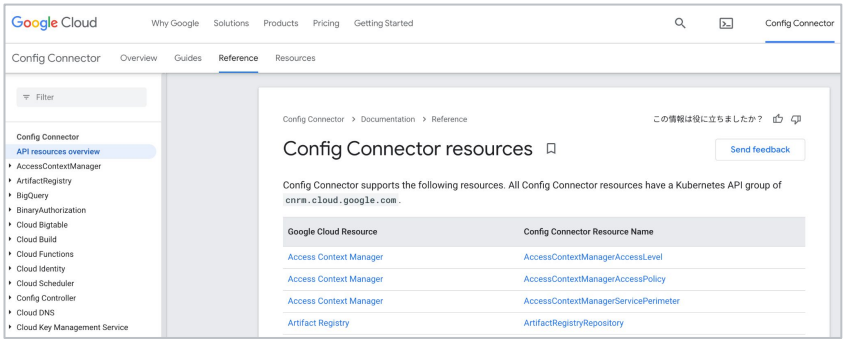
- **本番環境に安全にデプロイできるプロセスを確立できる**
  - コンプライアンス
  - バージョニング、コラボレーション
  - 環境変更前の**テストや適用自動化**によりリスク軽減

- **理想状態が維持され**、理想と実環境間で**差異は起きない**
  - **Reconciliation loop**

- 管理対象が大規模になろうと**運用負荷は一定**

Config Connector Resources リファレンス

Config Connector Resources リファレンス



https://cloud.google.com/config-connector/docs/reference/overview

制約テンプレート ライブラリ

https://cloud.google.com/anthos-config-management/docs/reference/constraint-template-library

制約テンプレートの作成 ： https://cloud.google.com/anthos-config-management/docs/how-to/write-a-constraint-template

Public Preview

Landing Zone

Google Cloud のベストプラクティスに基づいた環境を、**迅速にセットアップするためのブループリントを yaml で提供**する。カスタマイズも可能。

https://cloud.google.com/anthos-config-management/docs/tutorials/landing-zone

- Google Cloud の構築と移行の加速
  - 構成管理を自動化させ、ブループリントを活用することで **Google Cloud の構築や管理時間を短縮** できる
- 運用の一貫性
  - Google Cloud の構成管理を容易に自動化できる
  - **CaD（Configuration as Data）として yaml を git 管理**することでインタフェースを統一

Terraform with Google Cloud と Config Connector

	Terraform with Google Cloud Infrastructure as Code	**Config Connector** Configuration as Data
モデル	宣言型 デプロイする設定をコードとして定義	宣言型（KRMをベース） デプロイする状態をデータとして定義
ステータス管理	デプロイ時の状態を管理 tfstate ファイルでステータスを管理	「望ましい（理想）状態」を維持 （Reconciliation Loop）
言語	HCL（HashiCorp Configuration Language）	yaml
目的	サービス単位の デプロイの自動化	Google Cloud リソースの維持、管理
選択のポイント	**既に利用している経験を元に マルチクラウドでの統合管理**	**Kubernetes のスキルセット・仕組みを 生かしたリソース管理**

まとめ

まず試してみるところから始めてみましょう！

- Compute Engine インスタンスに対しても適切な Google Cloud が提供するマネージドサービスを活用してインベントリ、セキュリティ管理を実装できる

- Infrastructure as Code、Configuration as Data を取り組む上で必要なこと
  - インフラストラクチャの基本的なキャッチアップ
  - リファレンス、ブループリントを活用

- 徐々に大きくなるクラウド リソースの管理を見据えたインフラ管理を「チーム」で「適切に」実現できる仕組みを目指すきっかけに
  - テスト環境と本番環境の分離

Thank you.

Google Cloud