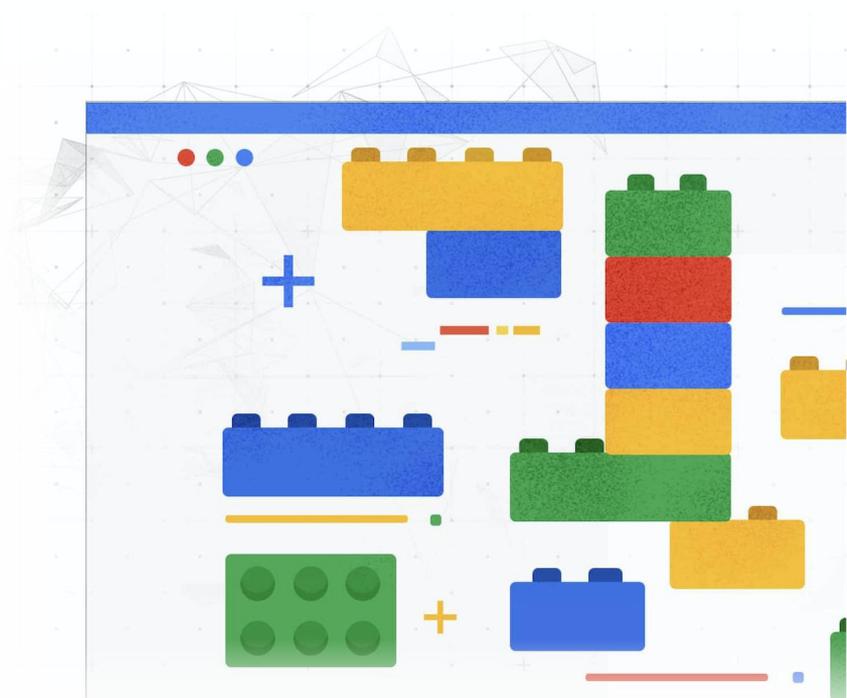


GKE で始める コンテナ開発

～ コンテナを使ったシステム開発をはじめる全ての人へ～

クラウドエース株式会社
ソフトウェアエンジニア
大磯 直人



自己紹介

大磯 直人(Naoto Oiso)

プロフィール

- クラウドエース株式会社
- ソフトウェアエンジニア
- Google Cloud Authorized Trainer

業務内容

- Google Cloud トレーニング
- Google Cloud の導入コンサルティング
- Google Kubernetes Engine を用いたシステム開発



クラウドエース紹介

クラウドエース カスタマーサービス

決済代行(日本円、銀行振込)+利用料割引+技術サポート+サイバー保険

SI・コンサルティング

導入初期の支援プラン「クラウドブースター」など。

Google Cloud 認定トレーニング

オンラインでも開講。

Google 製品の販売

Chromebook や Jamboard の販売。



本日のゴール

GKE を
なぜ(Why)
いつ(When)
どのように(How)
使うかを理解する

アジェンダ

1

GKE とは

2

GKE 活用プラクティス

3

GKE ユースケース紹介

GKE とは



GKE 前提知識

Google Kubernetes Engine(GKE)

Kubernetes

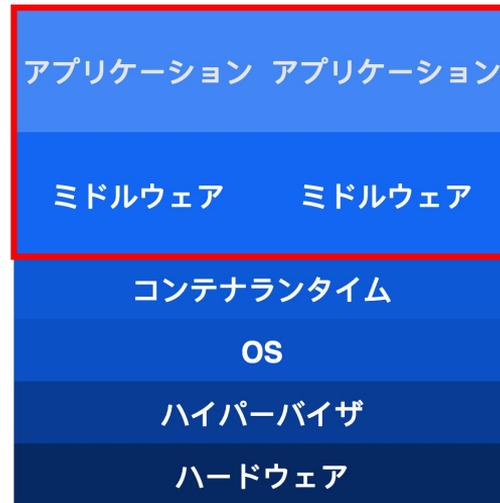
コンテナ

**前提となる知識が多く、抽象度の高いサービス
ステップを踏んで説明していく**

コンテナとは



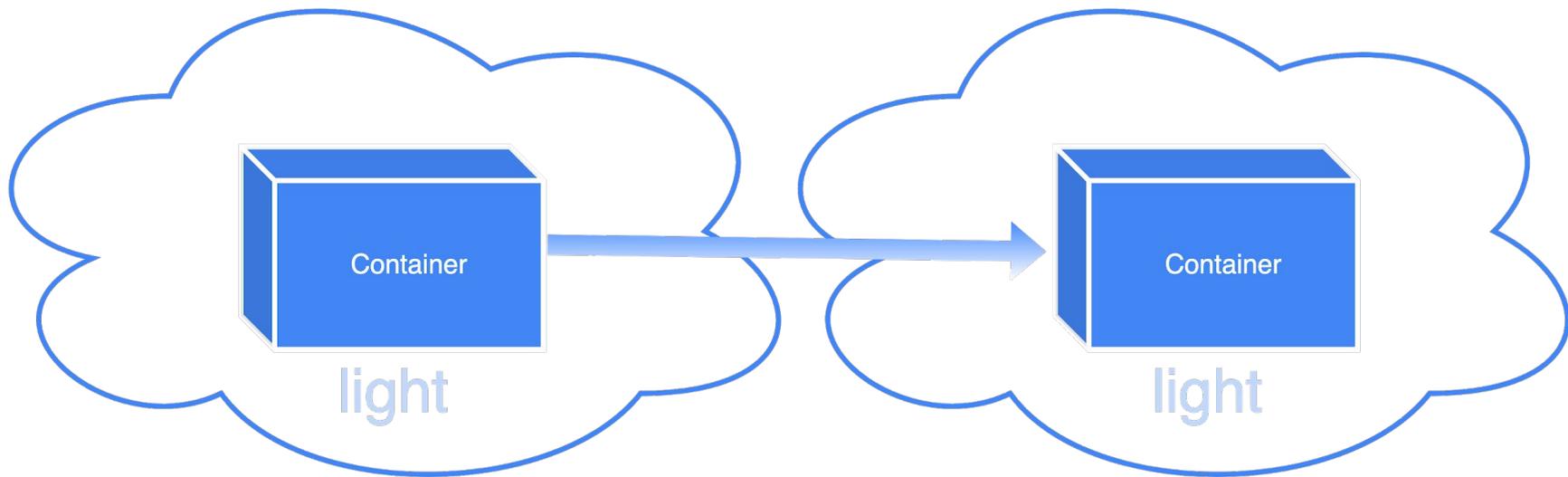
VM



コンテナ

リソース効率に優れた
軽量な仮想化技術

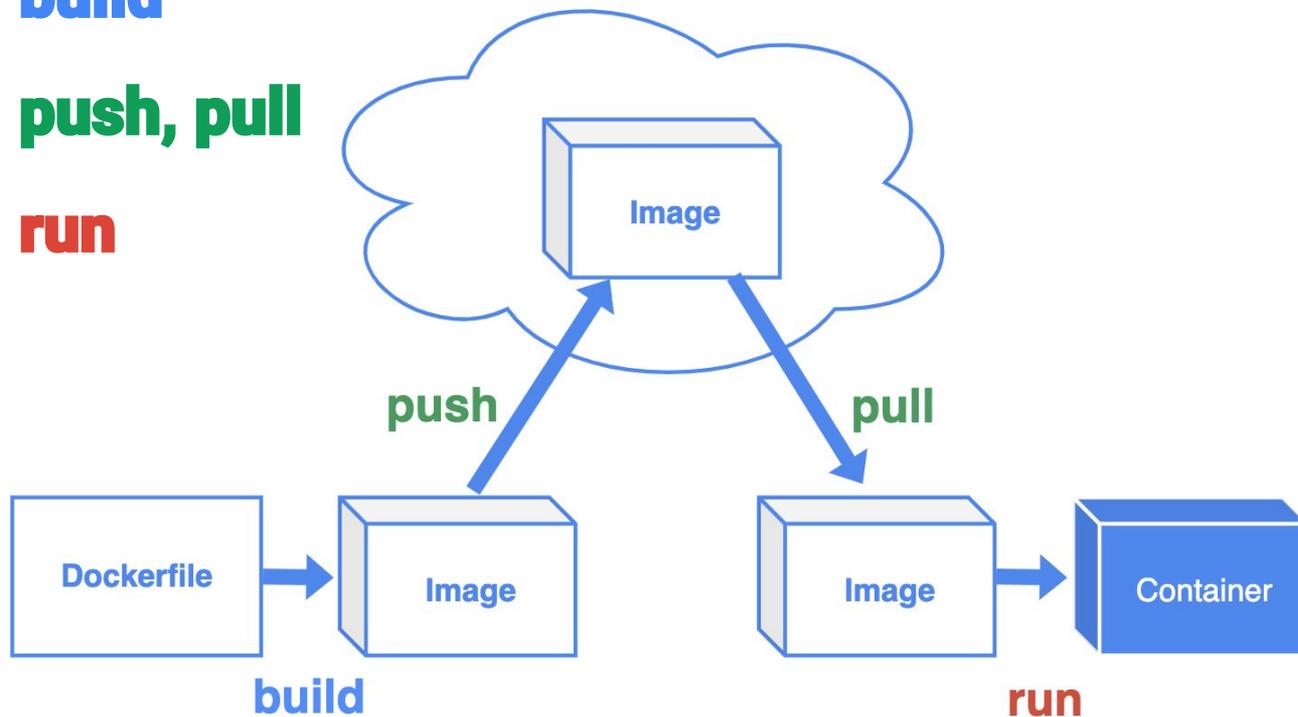
コンテナのメリット



ポータブル × 軽量

コンテナのライフサイクル

- 構築: **build**
- 運搬: **push, pull**
- 実行: **run**

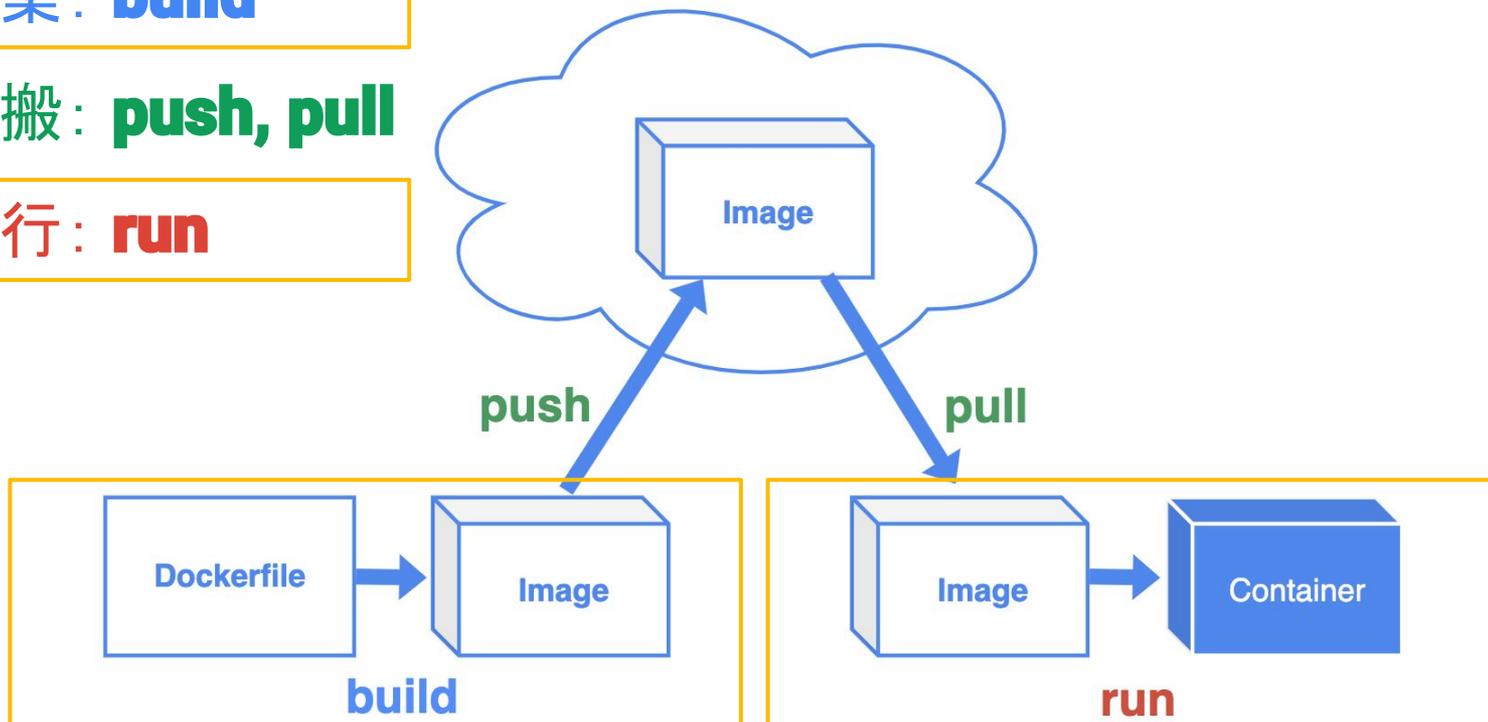


コンテナ操作デモ

- 構築: **build**

- 運搬: **push, pull**

- 実行: **run**



Kubernetes とは

クラスタ管理



状態管理

通信制御

コンテナアプリケーションの高度な実行、管理に必要な機能を
網羅的に有しているプラットフォーム

Kubernetes とは～クラスタ管理～

Cluster

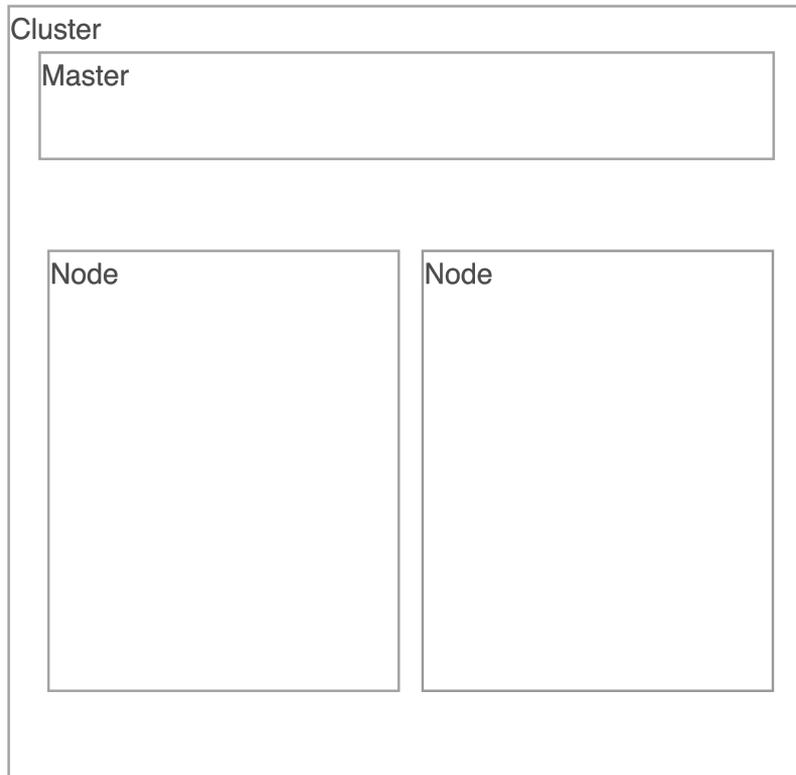
複数の VM または物理的なマシンを、一つの論理的なコンピュータとしてまとめ上げたもの

Master

クラスタの望ましい状態を維持する責務を持つプロセスが集められた VM。

Node

1つの VM または物理マシン。コンテナランタイムや、コンテナの実行を管理するエージェントを有する。



Kubernetes とは～状態管理～

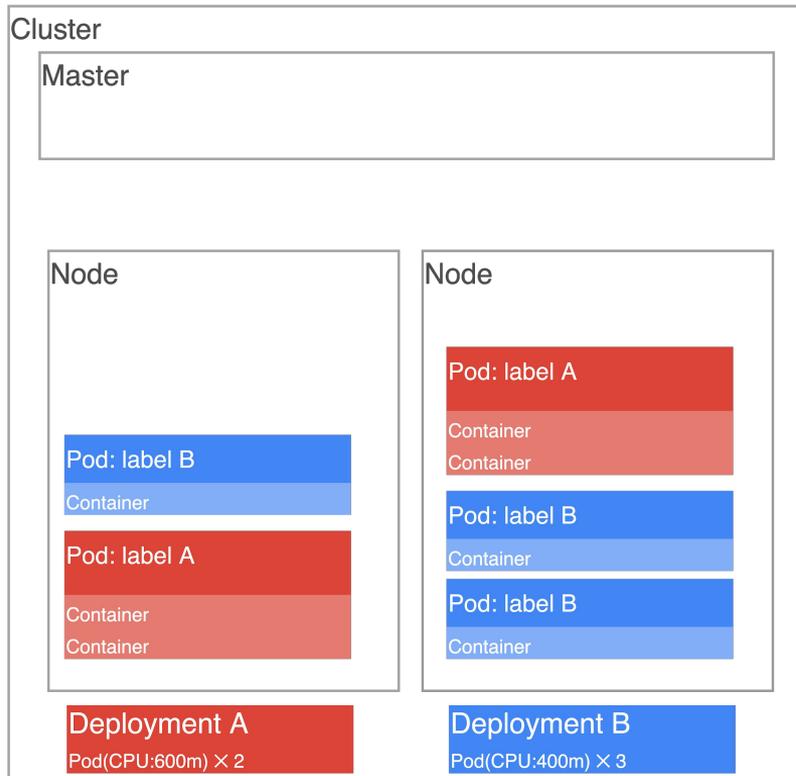
Pod

論理的なホストでアプリケーションの実行単位。

1つ以上のコンテナを、どれだけのマシンリソースを割り当ててデプロイするか定める。

Deployment

どの Pod をいくつ Node に配置するかを管理する。Deployment によって Pod が作成される。

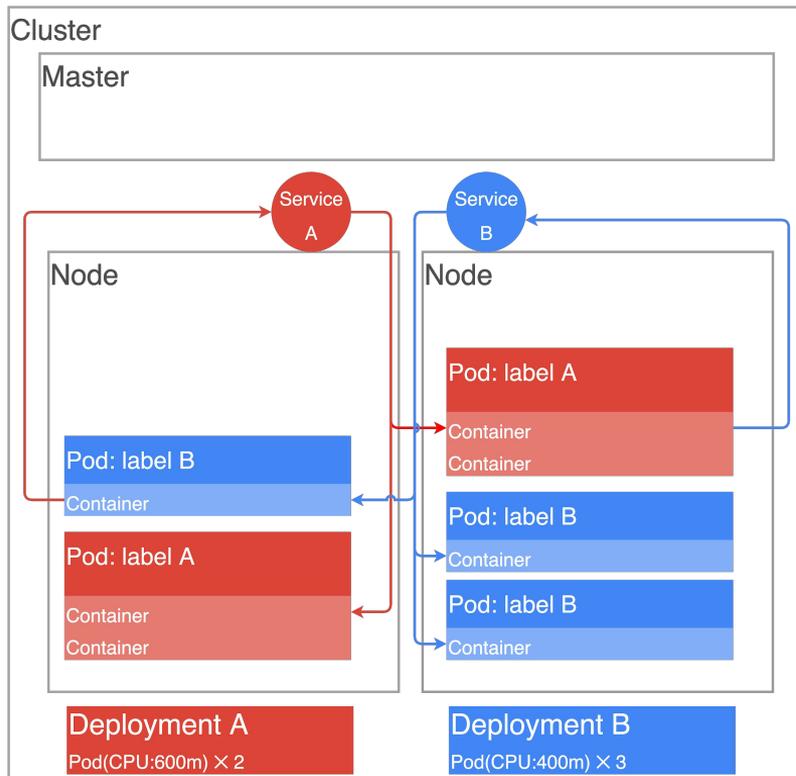


Kubernetes とは～通信制御～

Service

通信を制御する役割。Pod に到達する単一のエンドポイントを提供する。様々な Type が存在する。

- ClusterIP: クラスター内でコンテナに通信を流す
- NodePort: Node に到達した通信をコンテナに流す
- Loadbalancer: ロードバランサを使ってコンテナに通信を流す

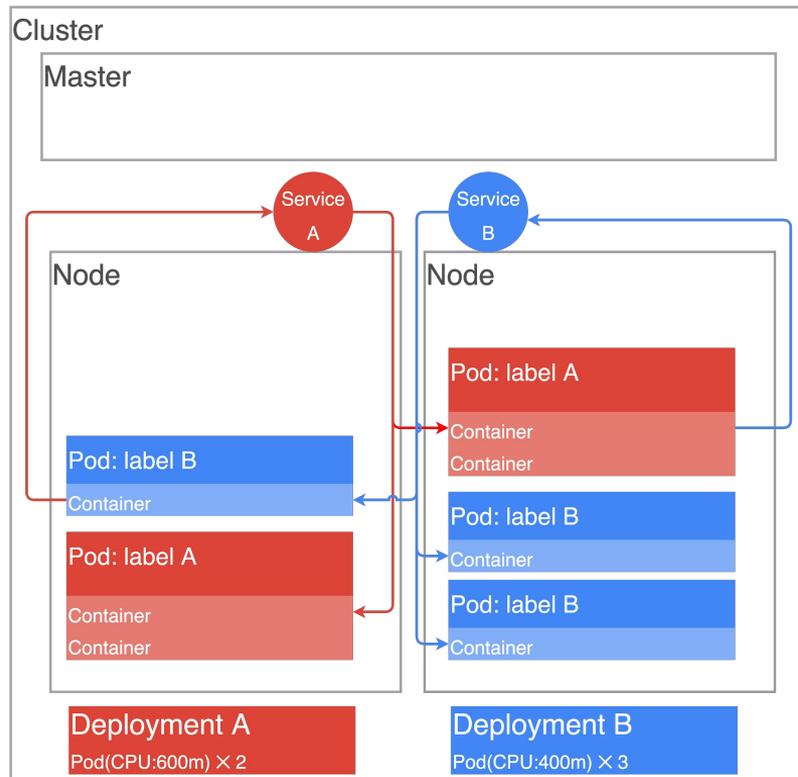


Kubernetes 操作デモ

クラスタ構築

アプリケーションのデプロイ

通信の疎通



GKE とは

**Kubernetes on
Google Cloud**

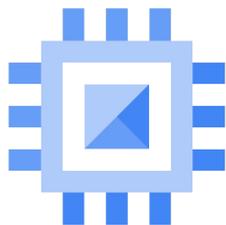


**Kubernetes with
Google Cloud Service**

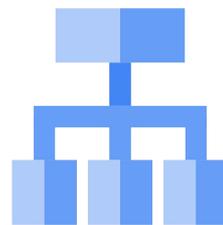
**Kubernetes by
Google Cloud**

Google Cloud 上で動く、Google Cloud のサービスと統合可能な、Google Cloudのマネージドサービスの Kubernetes

Kubernetes on Google Cloud



GCE (VM) のメリットを享受



Google Cloud Load

Balancing (GCLB) を使った

負荷分散

Kubernetes with Google Cloud Service



Cloud Logging

Cloud Monitoringと統合



IAM による権限管理

Kubernetes by Google Cloud



マスターの管理を **Google Cloud** が実施

自動 バージョンアップグレード

GKE 操作デモ

クラスタの構築

GKE 活用
プラクティス



GKE を使う時のポイント



- 自動スケールのチューニング
- Cloud Monitoring との統合



- 自動スケールインによる
サービスダウンリスク

Kubernetes Engineであることの

メリットを生かし リスクを抑える

可用性設計～自動スケール～

Node-1 vCPU2(=2,000m)



Pod A : CPU 1,000m

Pod B : CPU 600m

Pod の配置をする際に、**Pod** 毎に設定される

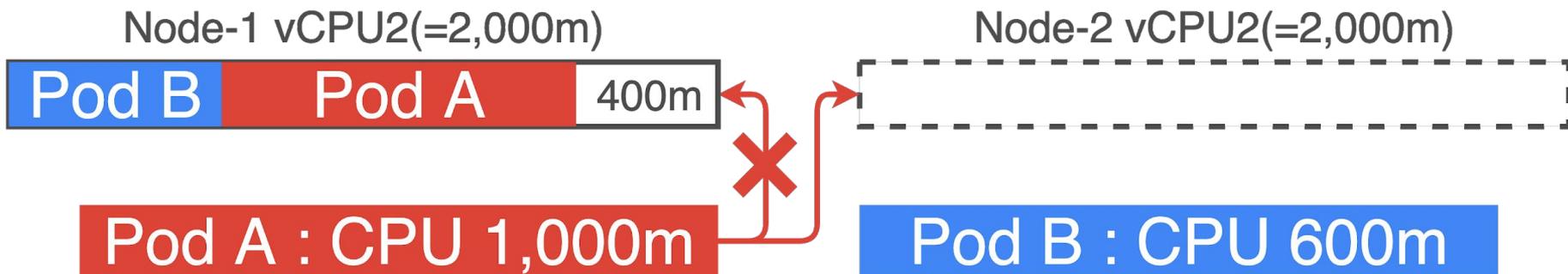
リソース **Request** 値に基づいて計算する



1vCPU = 1,000m

この VM には合計 2,000m、残り 400m 割り当て可能

可用性設計～自動スケール～



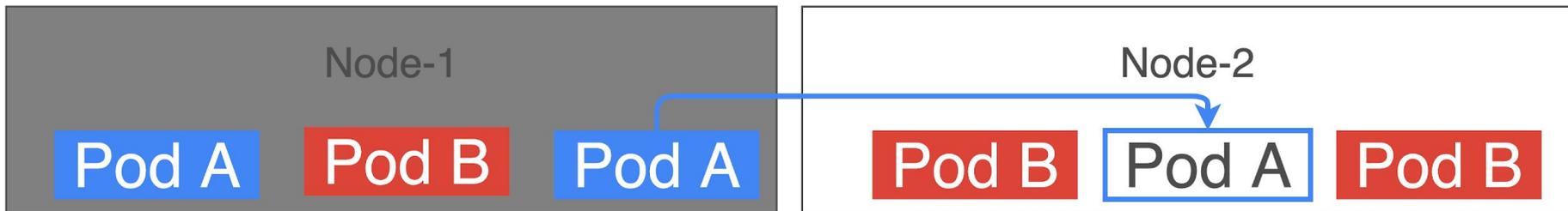
Pod のリソース request 値の合計が Node のリソース を上回り、

Pod を schedule できなくなるとスケールアウト



Pod に適切な **request** 値の設定が必要

可用性設計～スケールイン対策～

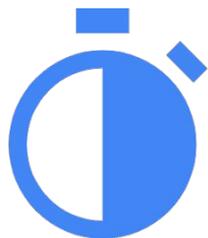


自動スケールによって、Node が削除されるとサービスダウンの可能性



Pod Disruption Budget: Pod 停止の許容可能数を制限し、**Pod** を別 **Node** に退避

監視設計～外形監視～



UptimeCheck

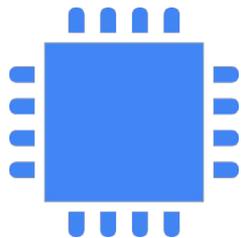
- レスポンス正常性
- レスポンスタイム



SLOベースのポリシー

- エラーバジェット
- バーンレート

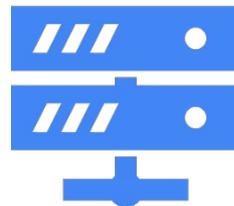
監視設計～内部監視～



マシンリソース監視

`gke-metrics-agent` (デフォルト)

マシンリソース指標を収集



Kubernetesリソース監視

`kube-state-metrics`(アドオン)

Kubernetesリソース指標を収集

GKE ユースケース紹介

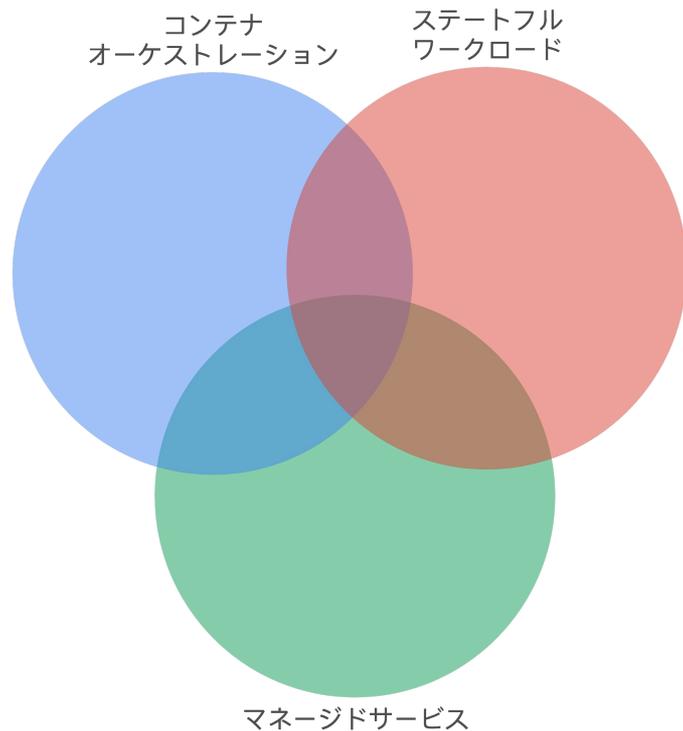


GKE のユースケース 1

複数コンテナ間で連携するシステム

特定のストレージと紐付くコンテナ

マネージドサービスである実行環境



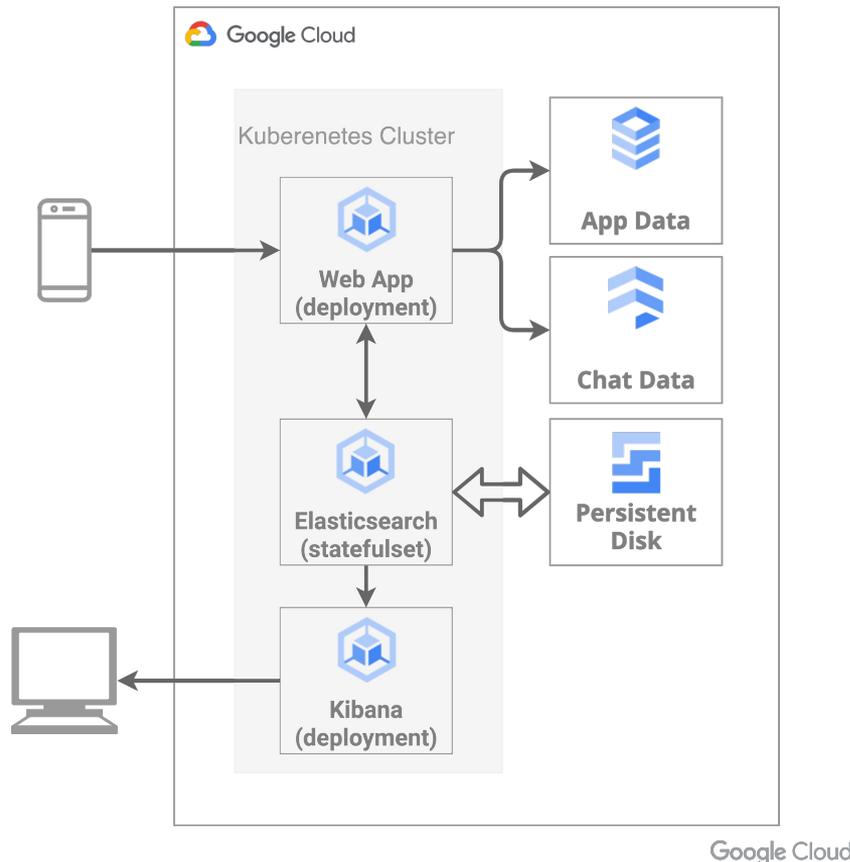
グループウェア SaaS 案件の ケース例

<ポイント>

- 自動スケール可能なインフラ
- ステートフルワークロードの運用

<設計>

- GKE と Firestore を採用し、事業規模拡大に対応
- Statefulset でステートフルワークロードの運用



GKE のユースケース 2



マイクロサービス

コンテナオーケストレーション
サービスメッシュとの統合が可能



ハイブリッド
マルチクラウド

オンプレや他のクラウドでも
共通実行基盤として利用可能

建設業者作業員管理システムのケース例

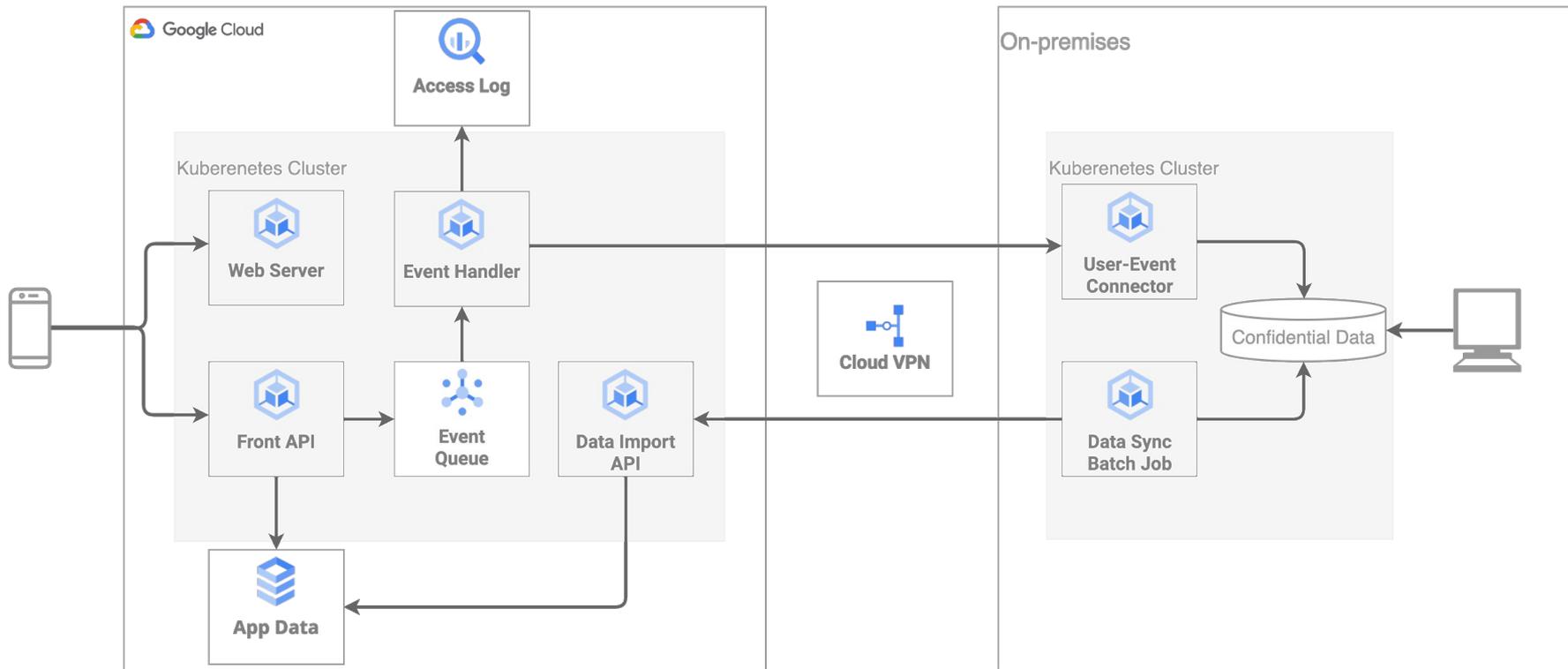
<背景>

- 作業員管理のシステム化
- オンプレにあるデータとの連携が必要

<ポイント>

- アプリケーション依存関係を整理
- オンプレとの安全な相互接続

建設業者作業員管理システムの事例



Google Cloud のお悩みはCAに

Google Cloud
導入実績

1,000社以上



Thank you