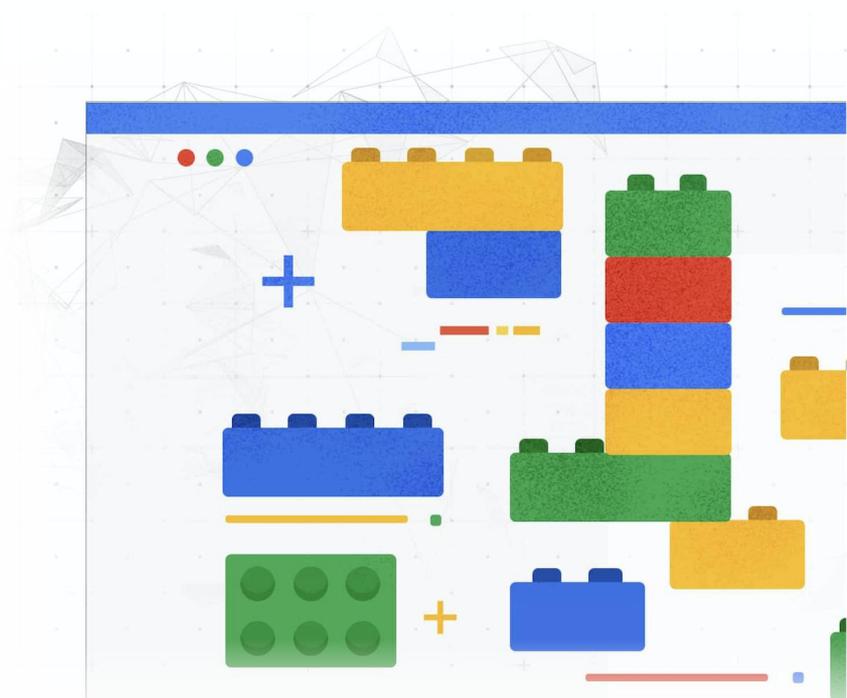


GKE Autopilot 入門

Google Cloud

Application Modernization specialist

篠原 一徳



GKE Autopilot 概要

GKE Autopilot は **GKE の新しいモード**です。

従来の GKE は GKE Standard と呼びます。

基本的な仕組みは従来の GKE と同じですが、以下の特徴があります。

- **Node も完全マネージド (Control Plane に加えて)**
- **Pod 単位の課金**
- **Pod 単位の SLA**

Node も完全マネージドな状態とは？

つまり、「**Node の運用をしなくてよい**」ということ。

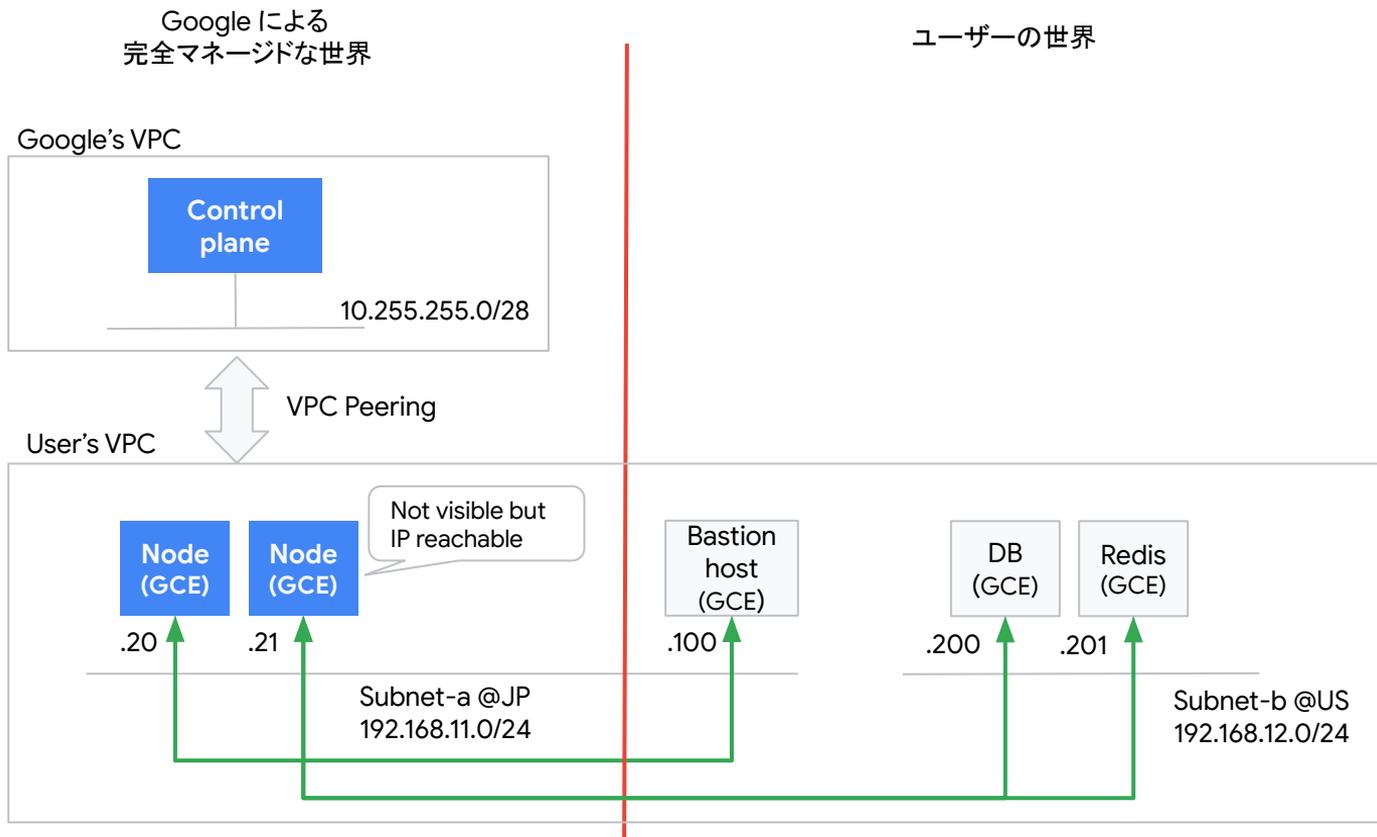
具体的には

- Node の作成、更新、削除は全て自動
- バージョンアップグレードも自動

ユーザー目線では

- **Kubernetes の「利用」に集中できる**

構成イメージ



Pod 単位の課金

作成された Pod のリソースに対してのみ**秒単位で課金**される。

システム系の Pod に対する課金はない。

Item	Price
GKE Autopilot vCPU Price (vCPU-hr)	\$0.0571
GKE Autopilot Pod Memory Price (GB-hr)	\$0.0063215
GKE Autopilot Ephemeral Storage Price (GB-hr)	\$0.0000704

確約利用割引(CUD) / 継続利用割引(SUD)

GKE Autopilot では CUD をサポート。

- 3年の確約で45% ディスカウント
- 1年の確約で20% ディスカウント

SUD は Node が E2 なのでサポート対象外。

<https://cloud.google.com/kubernetes-engine/cud?hl=ja>

Pod 単位の SLA

Autopilot の Pod には **99.9 % の SLO** (Monthly Uptime Percentage) が設定される。
対象となる条件は **2 つ以上の Zone に Pod が配置されていること**。

Pod とは別に **Control Plane には 99.95% の SLO** が設定されている。

<https://cloud.google.com/kubernetes-engine/sla>

Node 管理の仕組み

Node の作成、スケール、削除には従来のGKE が備えている、
Cluster autoscaler(CA) と **Node auto provisioning(NAP)** を使っている。

つまり

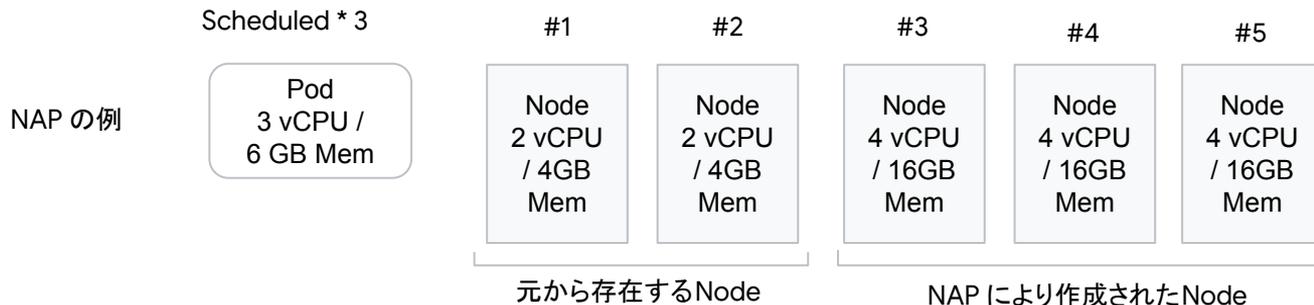
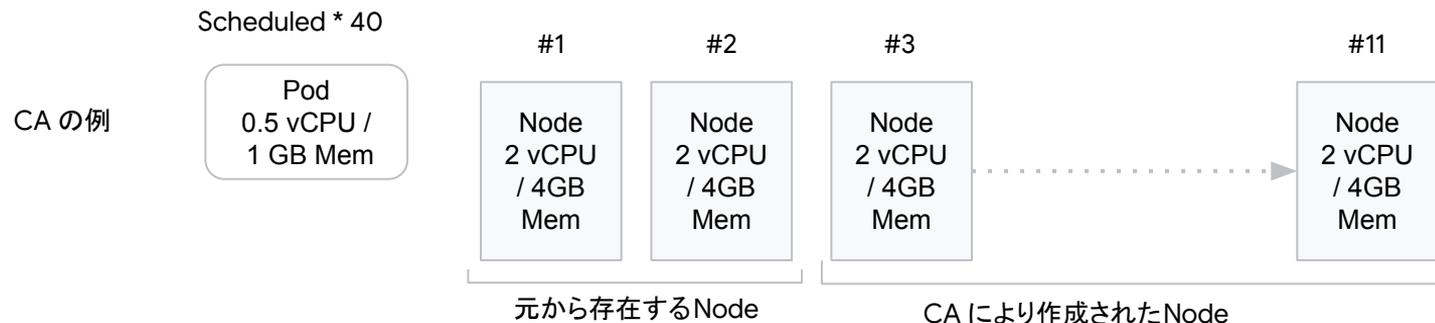
Node の作成、削除は Pod のスケジューリングに依存している

= 余剰の Node を予め用意しておいてスパイクに備えるということは基本出来ない。

([Workaround](#) はある。)

Node 管理のイメージ

Cluster autoscaler(CA) と Node auto provisioning(NAP) を使っている。



#Node の数は適当です。

Container のスペックの決まり事など

CPU は 0.25 vCPU 単位でインクリメント可能。中途半端な値を指定すると自動で是正される。

CPU と Memory は 1:1 ~ 1:6.5 の割合に収まっている必要がある。

Resource	Minimum	Default	Maximum
CPU	0.25 vCPU	0.5 vCPU	28 vCPU
Memory	512 MiB	2 GiB	80 GiB
Ephemeral Storage	10 MiB	1 GiB	10 GiB

https://cloud.google.com/kubernetes-engine/docs/concepts/autopilot-overview#allowable_resource_ranges

利便性の裏で、、主な制約

詳細は[公式ドキュメント](#)を参照のこと。

- Privileged pod は利用出来ない
- HostPort や HostNetwork は利用出来ない
- Node の sysctl / kubelet のパラメーターチューニングは出来ない
- Node に SSH アクセス出来ない
- Guaranteed QoS Class (No bursting)
- Mutating Admission webhook は使えない
- 全ての Pod は Seccomp profile が適用されている。
 - CAP_NET_RAW が drop される

ユースケース

k8s で動くワークロードは基本的に全て動かせるが、
GKE Autopilot 特徴を考えると以下のユースケースに特にハマる。

- バッチ
- 非同期のワーカー
- オートスケールの速さをそこまで求めない
- 大量のリクエストトラフィックを捌かない
- 開発、テスト環境用途
- ステートレス

まとめ

GKE Autopilot vs GKE Standard (機能面)

機能	GKE Autopilot	GKE Standard
Control Plane	完全マネージド	完全マネージド
Node	完全マネージド	マネージド
課金	Pod 単位 + Cluster 管理費用	Node 単位 + Cluster 管理費用
SLA	Pod 単位 + Control Plane のSLA	Node 単位 (GCE の SLA) + Control Plane の SLA
アップグレード	Control Plane: 自動 Node: 自動	Control Plane: 自動 Node: 自動 or 手動
Cluster type	リージョンクラスタ	リージョン or Multi zonal or Single zonal クラスタ
Network type	VPC native cluster	VPC native cluster or Route based cluster

もっと詳しいものが[公式ドキュメント](#)にあります。

まとめ

- GKE Autopilot の登場により Node の運用が不要になり、
より簡単に k8s が使えるように。
- GKE のベストプラクティスが詰まっています。
k8s をこれから使う方、運用を楽にしたい方にオススメ。
- GKE Autopilot は従来の GKE (Standard) をベースにより抽象度を上げたもの、
簡単に使える分、細かい部分で制約も。

Thank you