

ゲーマーのためのサービス
作った

自己紹介

かきぴー

職業: SE(?) 最近わからなくなってきた

言語: Go, JavaScript, Python

VRC: [かきぴー Kakipy](#) (2023/11~)

X: [@kakipy755](#)

よくFUJIYAMAにいます。



作ろうと思ったきっかけ

友達複数人でゲームするとき、自分がホストでワールドを作ると、他の人はホストがゲームを起動している間しかゲームできないですよ？

かと言って、ゲームサーバーを契約するほど毎日何時間もやっているわけではないですよ？

必要なときだけ起動できるゲームサーバーがあると便利だと思いませんか？

作りました

ゲームサーバーを簡単に立ち上げれるサービス。こまめに電源のオン/オフができるので、ライトユーザー的には嬉しいサービス。

まだプロト

対応タイトルはCore Keeperだけ

サービス名: 未定

ローンチ日: 未定



真: 作ろうと思ったきっかけ

Web系の開発してみたいけど、現職だとまともにできなかったから、勉強でなにか作ろうと思った。

とりあえず、プロトでいいからタイトル一つ選んで作ってみた。

購入ページ

購入ページでは、
CPUコア数とメモリを入力
する

Core Keeper

購入情報

サーバー名 _____

ワールド名 _____

CPUコア数

1

メモリ

8

GB

購入

サーバー詳細ページ

電源のオン/オフ

CoreKeeperでサーバーに参加するときに必要なゲームIDを取得

Core Keeper

fee79286-9606-44f6-b77b-1a51bf78e113 Running

Game ID  

CPUコア数	メモリ	GB	更新
1	8		

 インポート

削除

Copyright ©2023 All rights reserved | xxx
version: d9ad40addfcbcae6c1c7fac7bb26b50a2e084dee

アクセス

ゲームIDを入力すればサーバーに参加できる。



アーキテクチャ

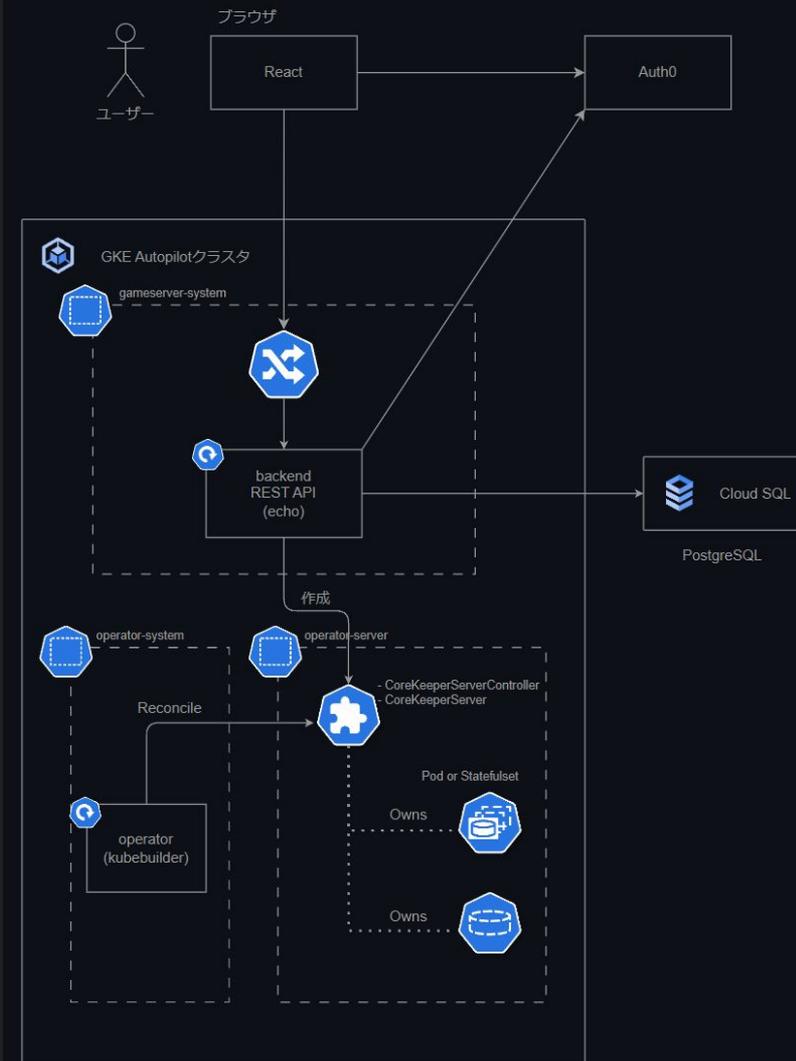
クラウド: GCP (GKE Autopilot)

フロント: React

バックエンド: echo (Go言語)

認証: Auth0

ゲームサーバーの制御: オペレーター
(CRD + カスタムコントローラー)



クラウド選定理由

GKE Autopilotしか要件を満たせなかった。

要件:

- コンテナでゲームサーバーをデプロイ可能(慣れと扱いやすさの観点)
- 必要なリソースだけ割当可能(スモールスタート可能)
- ボリュームをコンテナにアタッチ可能(セーブデータの格納先)
- ボリュームにはブロックストレージを使いたい(金ない)

ブロックストレージってなんやねん

	ファイルストレージ	ブロックストレージ
目的	ファイル共有	データ保存
プロトコル	NFS, SMB	SCSI, NVMe
例	AWS EFS	AWS EBS, 市販のSSD, HDD
複数サーバーからアクセスすると	データが壊れない	データがぶっ壊れる
価格	ブロックの3倍くらい高い	安い
その他	最終的なデータの格納先はブロックストレージ	

フロント選定理由

Vueはちょっとしかわからない。

Reactくらいしかまともに書けない。

APIクライアント作るの面倒だったので、TypeScriptではなくJavaScript。Open API導入したらTypeScriptにするかも。

後悔: GCP全然知らなくて、CloudStorageにデプロイしたから、直リンクできない。デプロイ先おすすめあれば教えてください。

バックエンド選定理由

言語は慣れてるGoを選択。

Webフレームワークは、小規模向けにすすめとわれてるechoを選択。

後悔: 簡単なのでデプロイ先をK8sにしたが、ゼロスケールできない。あんまり知らないが、そのまま移せるならCloudRunに移行したい。

認証認可

あんまわからんし、適当にAuth0使ってみた。

GCP使うし、Firebaseとかでも良かった気がする。アドバイスほしいです。

ゲームサーバーの制御

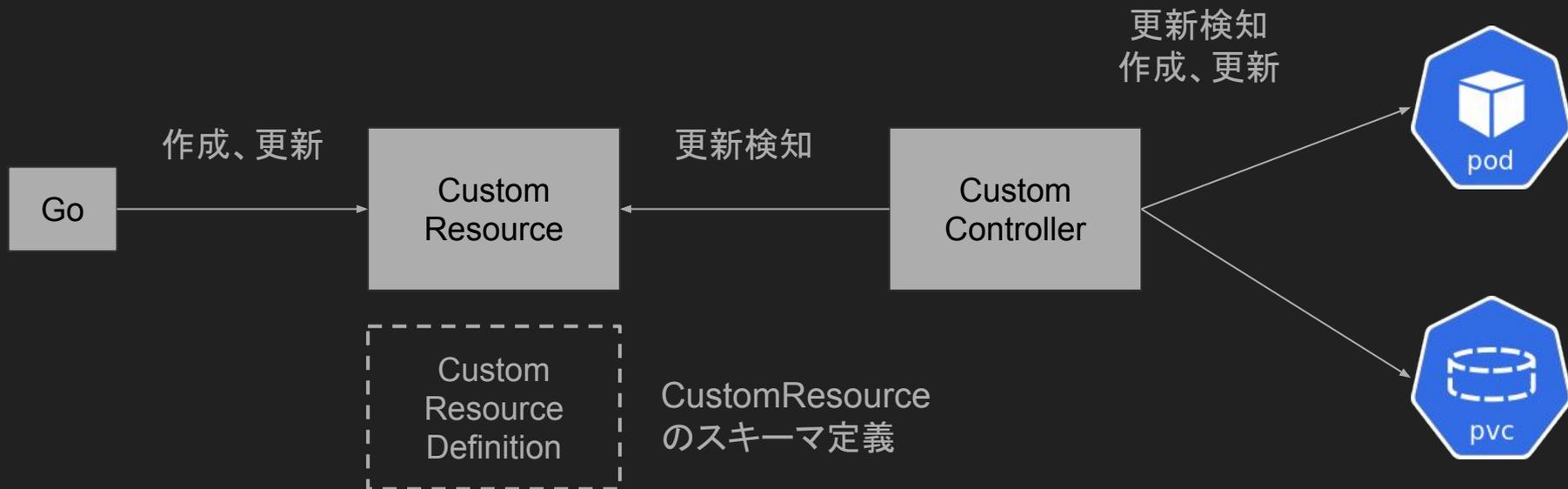
K8sのオペレーターでゲームサーバーのコンテナをデプロイ(kubebuilderを使って作成)

オペレーターはK8sの拡張機能でK8sのリソースをデプロイするのに便利なツールみたいなもの。管理対象のリソースが落ちたときデプロイし直すとかやれる。

ただ、クライアント生成方法に関する公式ドキュメントが見当たらないのがキツイ。

オペレーターの仕組み

CustomResourceや管理対象のリソースの変更を検知して、カスタムコントローラーのロジック(Reconcile)が動く。Reconcileの中で管理対象リソースの作成や変更を行う。



価格の話

例えばマイクラだと、既存サービスは4GB 1,500円/月くらい。

GKEのRegular Podだと原価だけで8,874円/月(12円/時)なので、とても戦えない。

Spot Podを使うと、原価だけで2,659円/月(3.7円/時)とギリ使えるレベルになる。

Spot Podでこまめにサーバーの電源をオン/オフしてくれるケースなら使い物になる。例えば、価格が10円/時としても150時間までならこのサービスのほうが安くつく。

本実装(気力あればやること)

機能追加:

- 課金周り
- 許可したユーザーが電源のオンオフできるようにする
- あと一つくらい(マイクラとか?)ゲーム対応する

技術負債解消:

- ゲーム追加を楽にする
- バックエンドをCloud Runに乗せる
- Open API
- TypeScript
- Next.js

おわり