

# K8sクラスターでゲームサー バーをオーケストレーション

情報学群情報科学類2年 佐藤 良  
担当: 阿部 洋丈 先生

Ping : 20 FPS : 183

matched

status : Debug

1

1.0 / 100.0

10  
10

1

R

X



Q



# UnrealEngineの概要

- UnrealEngineはオンラインネットワーク対戦を前提とした作りのため、エンジン単体でクライアント実行バイナリと、サーバー用実行バイナリどちらも並行して作成が可能
- ある程度書き出される形式は決まっている

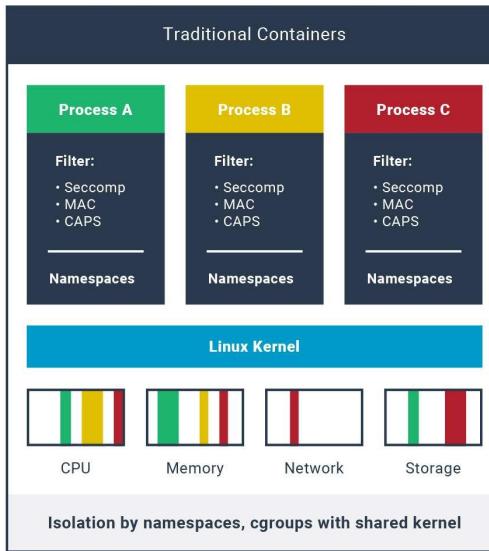
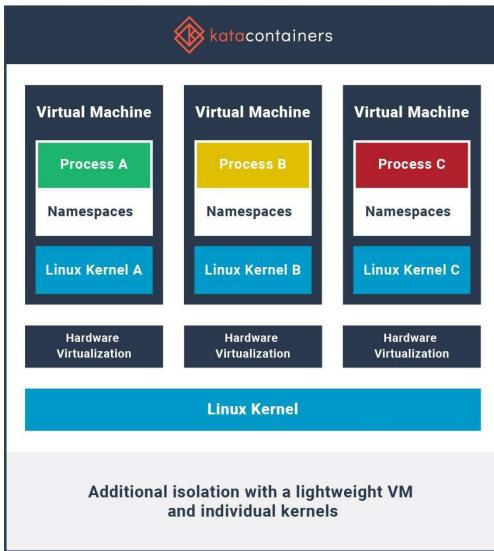


# 達成したいこと

- 安価なDedicated server as a Serviceを作りたい
  - できる限り既存のセッション/サービスは落とさたくない
  - 任意コード、バイナリの実行を許す
- セキュアな環境を作る必要がある



# Kata Containersの導入/概要



OCI コンテナ形式のランタイム。  
内部では基本的にQEMUを立ち上げている。  
VM並みの隔離をコンテナ環境で実行できる。  
ランタイムクラスでkataを指定するだけで変更可能。

```
kata > ! runtimesclass.yaml
1  apiVersion: node.k8s.io/v1
2  kind: RuntimeClass
3  metadata:
4    name: kata
5  handler: kata
```

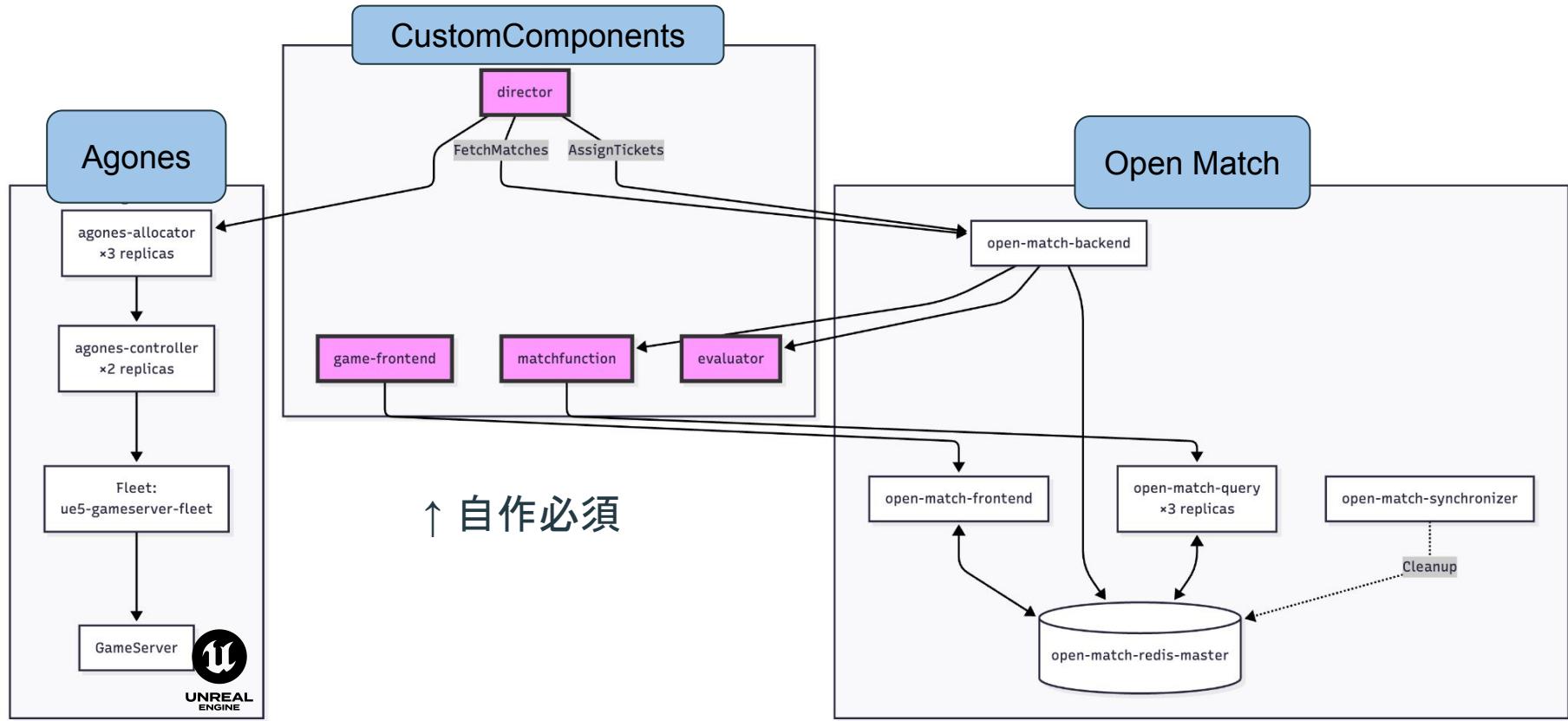
# インチキクラスタはリアルタイムゲームサーバーと相性が良い？

Dedicated game servers also need a direct connection to a running game server process' hosting IP and port, rather than relying on load balancers. These fast-paced games are extremely sensitive to latency, which a load balancer only adds more of. Also, because all the players connected to a single game server share the in-memory game simulation state at the same time, it's just easier to connect them to the same machine.

専用ゲームサーバーの場合、ロードバランサーに依存するのではなく、稼働中のゲームサーバープロセスが動作しているホストのIPアドレスとポートに直接接続する必要があります。このようなリアルタイム性が求められるゲームは遅延に対して非常に敏感であり、ロードバランサーはむしろ遅延を増大させる要因となります。さらに、単一のゲームサーバーに接続しているすべてのプレイヤーが同時にメモリ内のゲーム状態を共有するため、同じマシンに直接接続する方がはるかに効率的です。

<https://cloud.google.com/blog/products/containers-kubernetes/introducing-agones-open-source-multiplayer-dedicated-game-server-hosting-built-on-kubernetes>

# AgonesとOpen Match



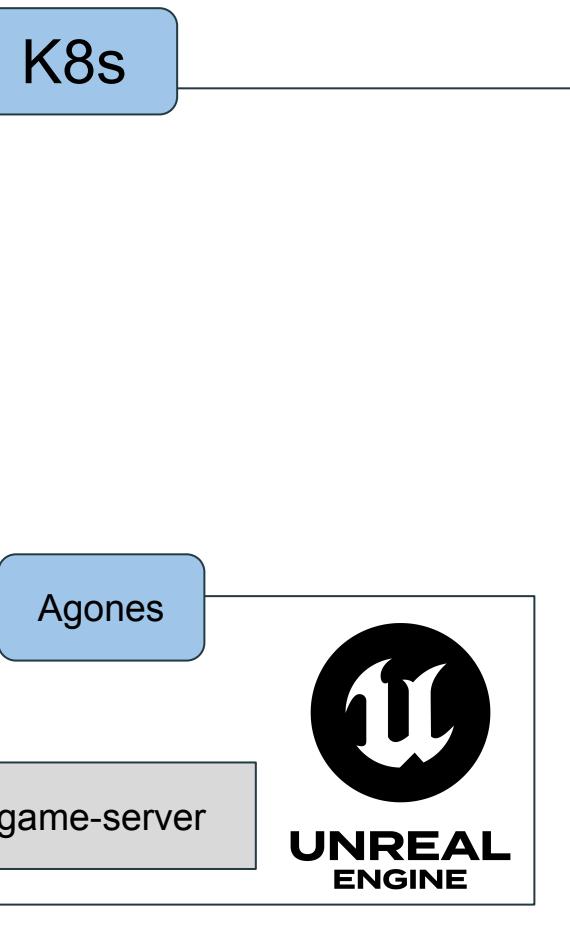
# 実際の接続フロー



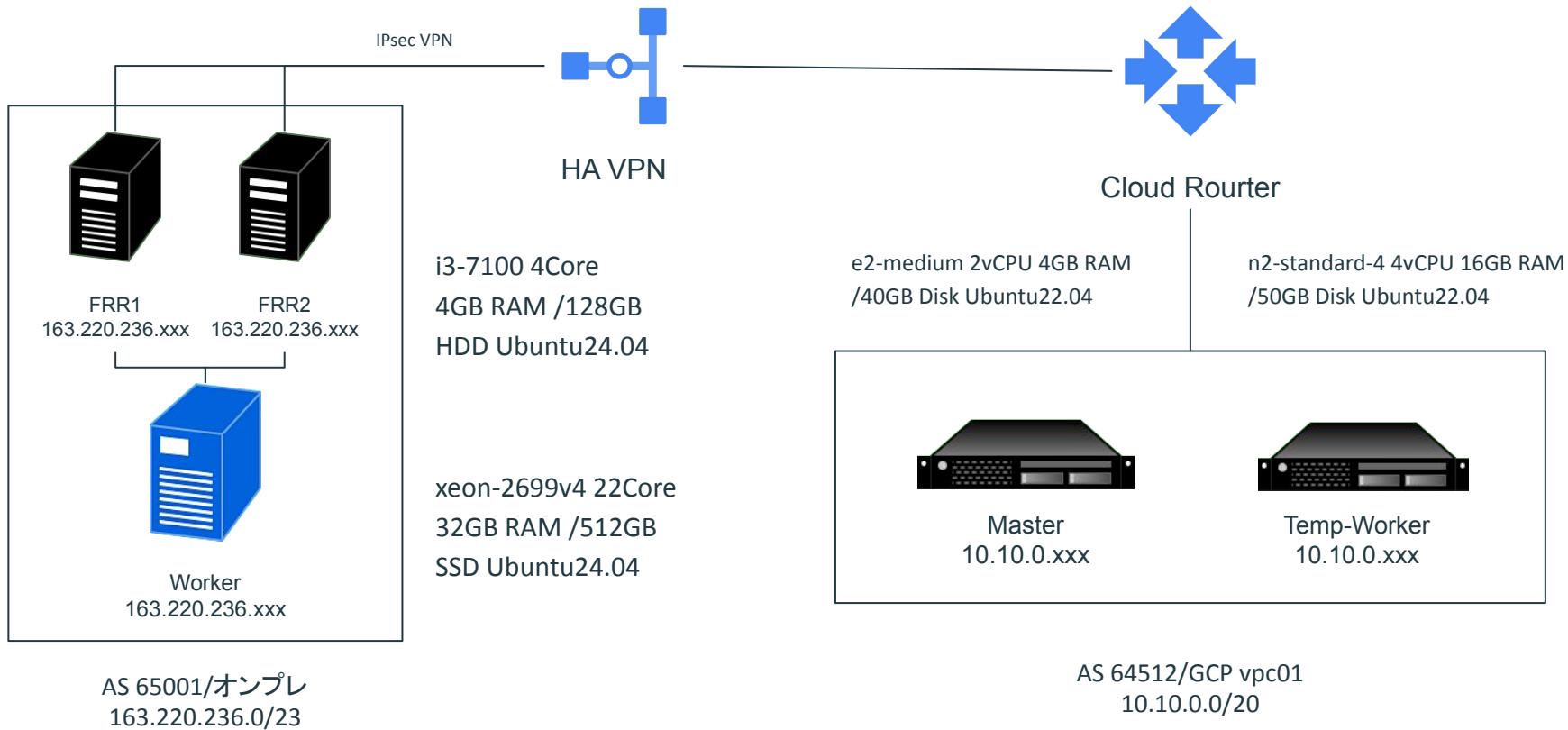
openlevel()  
w/アクセストークン

フロントへのトークン

サーバーへのアクセストークン



# オンプレ/クラウドハイブリッドK8s



# 闇のノードオペレーター

- オンプレを監視し、疎通が取れなくなった場合GCPにVMを作成する、闇のオペレーターを作成
- 復旧時は自動で一時VMをクリーンアップ

mizuamedesu/**hybrid-node-operator**



1  
Contributor

0  
Issues

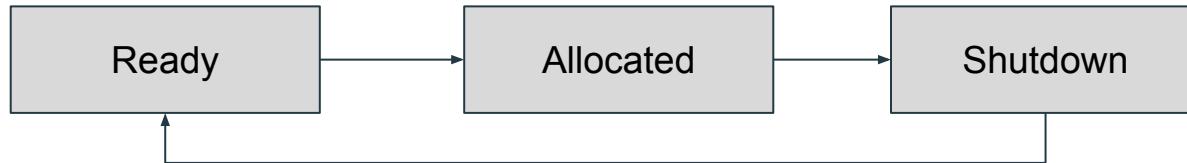
1  
Star

0  
Forks



<https://github.com/mizuamedesu/hybrid-node-operator>

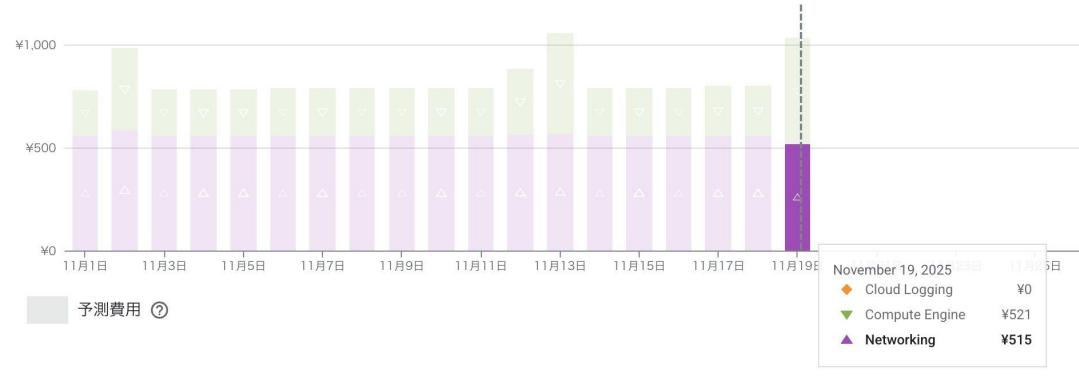
## Agonesの状態遷移



正確にはScheduled,Requested,Startingもある

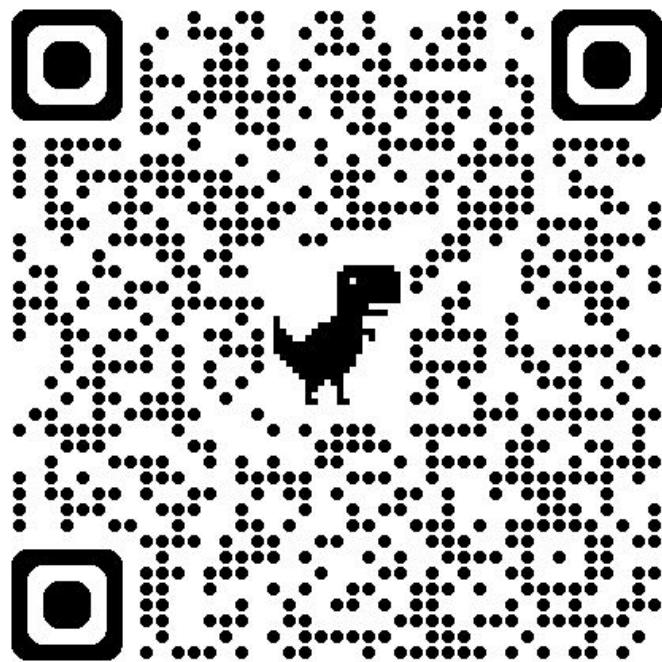
- すでにAllocatedされたpodが一時VMに立っている間は、VMの削除を実施しない。
- ReadyのPodは全削除
- 全てのVMのPodが消えるか、最大許容時間を超えるとVM自体の削除を実施

## 出来たこと/展望



- masterをクラウド、workerをオンプレで行うPoCに成功した
- オンプレのノードが死んだ際のフォールバックのPoCも成功  
→ 安価である程度安定性の高いDedicated server as a Serviceの基盤はできた
- kataコンテナでセキュア環境を作り上げた
- HA VPNが結局かなり高いので改善の余地はある  
大体月1.5万円→tailscaleに移行しています

Full Ver



# 謝辞

間瀬bb

[https://x.com/bb\\_mase](https://x.com/bb_mase)

Ultra-Coins

<https://ultra.coins.tsukuba.ac.jp/>

登 大遊(<https://x.com/dnobori>)先生をはじめとする、グローバルIPの貸し出しなどをして下さっているIPAの方々など