Gaming on AWS

Develop, Operate and Analyze on Cloud

Building PUBG with Microservice

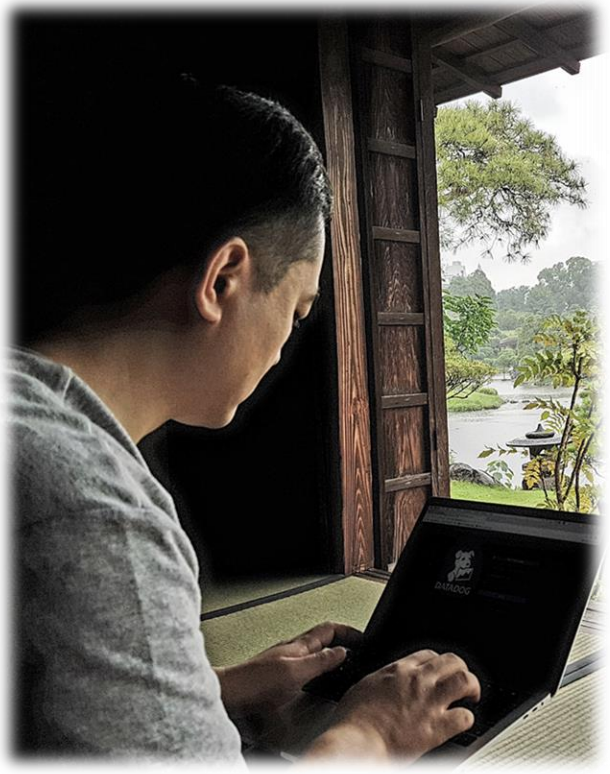
심재한

PUBG

소개

심재한 (pesim@bluehole.net)

- PUBG@bluehole platform software engineer
- TERA@bluehole platform software engineer
- Pmang@neowiz platform software engineer



Gaming on AWS

Develop, Operate and Analyze on Cloud

1. Achievements
2. Story about platform developments
3. Microservice

Gaming on AWS

Develop, Operate and Analyze on Cloud

Achievements

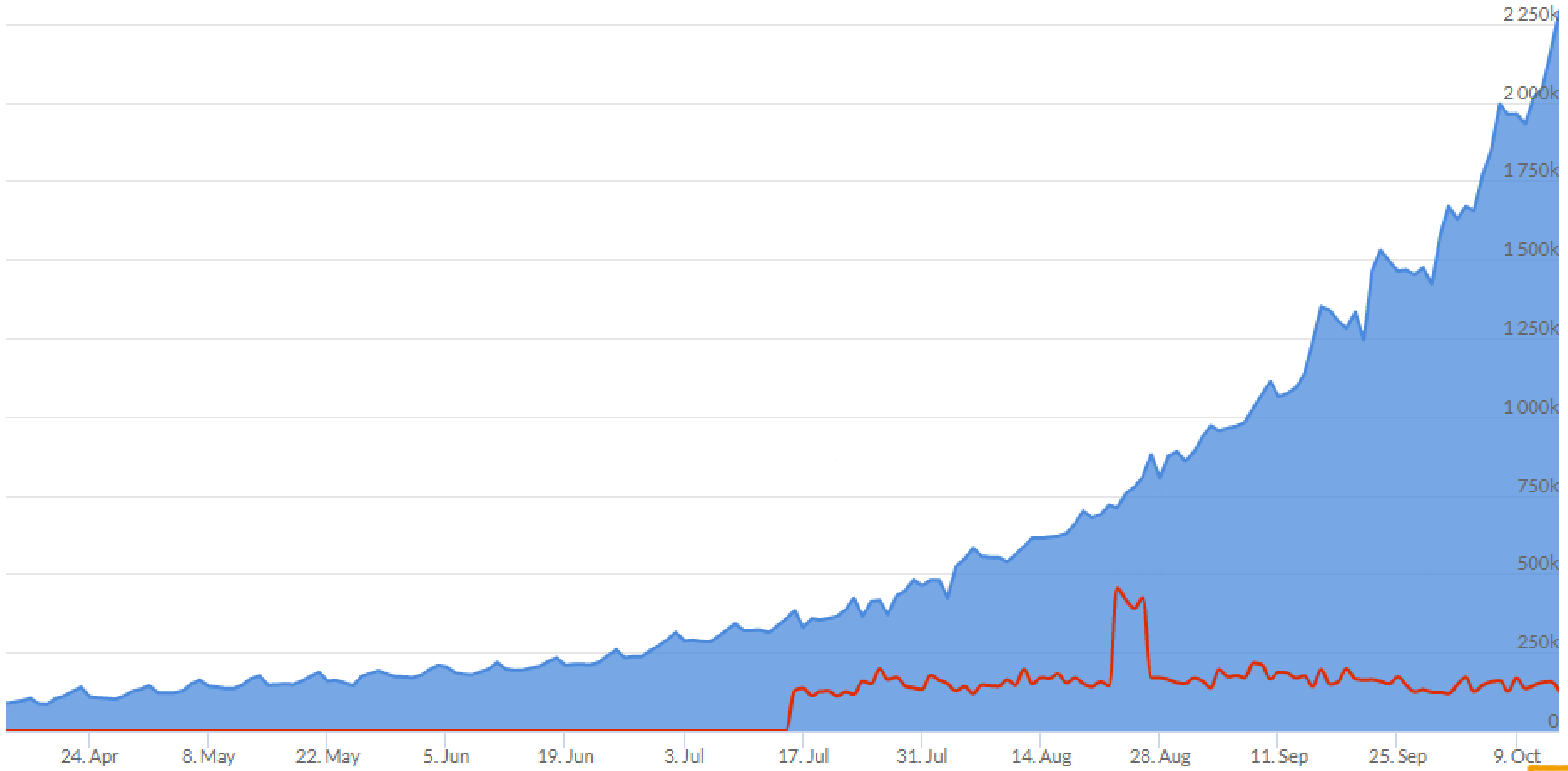
Sales Number

Gran Turismo 3 : A-Spec	**14.8M**
Call of duty : Modern Warfare	**17.2M**
Playerunknown's Battleground	**18M**
Mario Kart DS	**23.6M**
Call of duty : Modern warfare 2	**25M**

Daily MCU (from steamspy)

Gaming on AWS

Develop, Operate and Analyze on Cloud



Detailed events

4/20	100K
6/04	200K
7/30	Service failure just before half million

- According to the reasons of failure, it is determined that we couldn’t hold 600K users
- Began to think about the incensement of users seriously
- Started the “Project million”

08/27	870K

- Surpassed “DOTA 2” Ranked as the first at the Steam
- Started the “Project two million"

09/23	1.5M
10/11	2M

I Warned You

Notification Type *(string)* –

One of the following event notification types:

- autoscaling : EC2_INSTANCE_LAUNCH
- **autoscaling : EC2_INSTANCE_LAUNCH ERROR**
- autoscaling : EC2_INSTANCE_TERMINATE
- autoscaling : EC2_INSTANCE_TERMINATE_ERROR
- autoscaling : TEST_NOTIFICATION

Status Reason : Insufficient capacity. Launching EC2 instance failed.

Gaming on AWS

Develop, Operate and Analyze on Cloud

Story about platform developments

PUBG aws

Story about platform developments

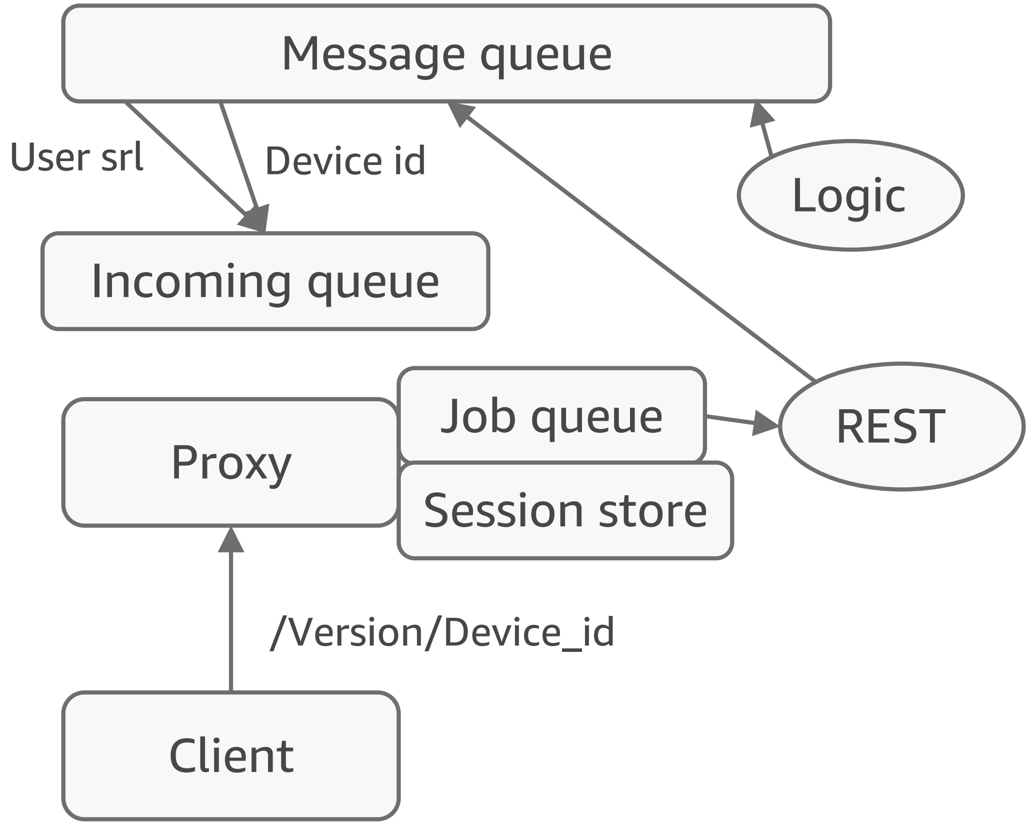
Directionality

- Worldwide service through STEAM
- Provide service with Public cloud (of course)
- Use microservice architecture
- Use websocket and message queue as Communication method
- Use service discovery

Platform for mobile game(2015)

Gaming on AWS

Develop, Operate and Analyze on Cloud



Story about platform developments

Development tool

- **C++**: Not suitable for web protocol handling
- **Java**: Popular but we have not experienced it
- **NodeJS**: Most trendy language for Microserivice
- C# dotnet core?

DevOps

- Everything is Code (In cloud). Codes can be changed
  - We have no system engineer

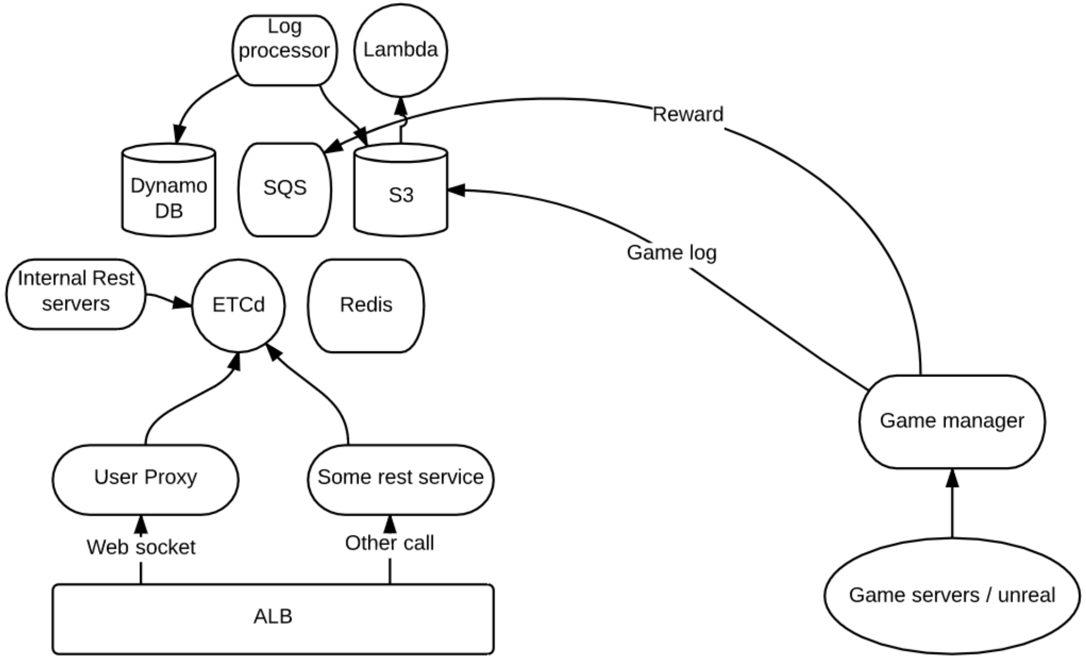
Develop period

1 year (as known as)

Visual studio 2015 update3 and dotnet core 1.0 released at June

AWS released ALB at October

Current architecture



Game shard / Main shard

Game server doesn't communicate with the database

If a game session is completed, a game log is uploaded to AWS S3.

The request of BP payment is sent by AWS SQS

The ranking service calculates the ranking with uploaded game log.

- ELO ranking can't be calculated before a round is completed

Service Discovery

Find a callable endpoint with logical service name

Popular method on AWS

- Using internal ELB and query by name

Problem found (with Prototype)

- Needs synchronization between Source code and infra
    - Every domain name requires unique name
    - To obtain a domain name, another query is required
- It is suffer to pay for Internal ELB
    - Internal traffic is free on the AWS

Message Queue

Major message passing in micro service is achieve by REST

- REST is sync call
- To process asynchronously, callback endpoint is required
  - Every context is passed by arguments or caller should keep it.
  - If caller keeps it, caller should not be scaled in.

Using message queue for every async message processing

Research on MQ (2015)

Rabbit MQ

- Rich reference
- Completed API
- de facto
- Performance

AMQP 1.0

- It is one-point-zero.

Research on MQ (cont')

Redis

- Redis has already existed for cache purpose.
  - Number of server components can be reduced
- Easy but chappy. No Advanced routing
- There is no transaction, if handler crashes a message is gone also.

AWS SQS

- Bullet-proof system by Amazon
- Slow
- Easy but chappy. No advanced routing also

First Try

Gaming on **AWS**
Develop, Operate and Analyze on Cloud

Using Rabbit MQ on CBT stage

Problems

- Must be managed by ourselves
- Even with cluster configuration, it can only handle things within configured limits
- If it exceeds its limits, it explodes.

If a particular service is crashed, message queue crashes.
The message queue crashes, the entire service crashes.

AWS SQS

General perception

- Sluggish
- There is no handy wrapper

<u>Everything is true but…</u>

AWS SQS

Not as slow as you thought

- Message is passed through queue only once, so users can’t really notice it.

It can store as many messages as you send

- It can hold every messages until a particular service is restarted
- A service can handle stored messages after restart

The quantity of message pushed and popped doesn’t affect the response time

- Just make as many readers as you want

Message isn’t banished till you delete it

- When a handler throws an exception, another handler will handle it.

It is reliable even if you send messages through internet

- A game server at Singapore sends game logs to Virginia.
- One of the longest route on Earth.

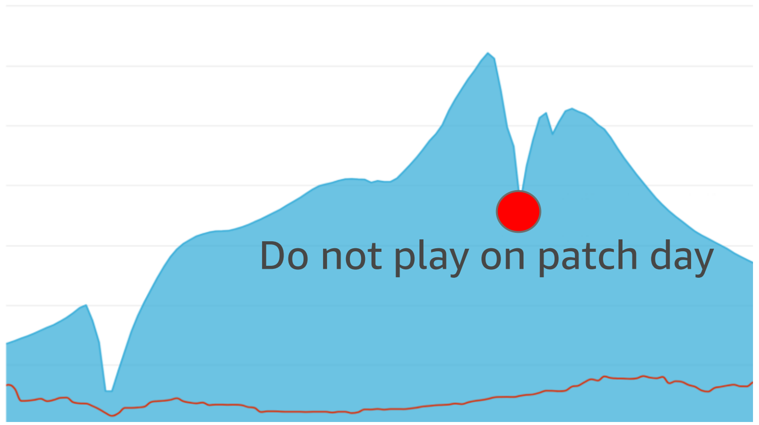
Microservice

Microservice

Gaming on AWS

Develop, Operate and Analyze on Cloud

No SPOF



Easily scalable

- Project million, two million

Back to Basic

Gaming on AWS

Develop, Operate and Analyze on Cloud

Microservice is not a silver bullet, but…

We can scale our operation independently, maintain unparalleled system availability, and introduce new services quickly without the need for massive reconfiguration.

Werner Vogels, Chief Technology Officer, Amazon Web Services

At the beginning

Gaming on AWS

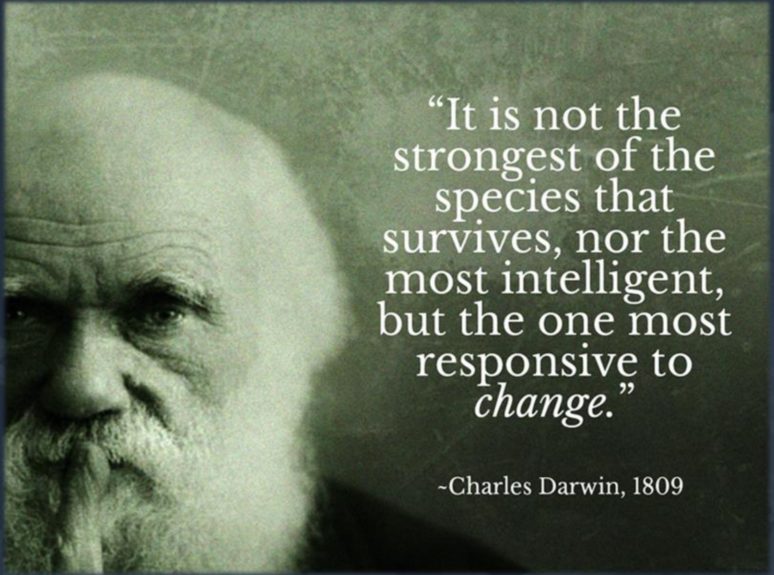
Develop, Operate and Analyze on Cloud

Main shard is consist of 8 services, Now it has 23 services

It was from “project million” when the diversion of the services had been occurred.

처음 설계한 구조는 단지 첫 버전일 뿐이다.

빠른 업그레이드가 이어지지 않는다면
반드시 도태될 것이다.


"It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to *change.*"
-Charles Darwin, 1809

Gaming on AWS

Develop, Operate and Analyze on Cloud

Thank you

We are hiring