



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

What Survived Production: Operating Game Backends at Million-Player Scale

Berkay Uckac - Staff Software Engineer, Server Lead @ Futureplay Games



Why This Story Matters to Small Teams



KubeCon



CloudNativeCon

Europe 2026

- Cloud-native can feel heavy for smaller teams
- We scaled without a big platform org
- We optimized for operability, not novelty
- What “Keep it lean, keep it minimal” means
- What survived, what failed, and what we removed



KubeCon



CloudNativeCon

Europe 2026

Part 1: The Story



The Comeback Story



KubeCon



CloudNativeCon

Europe 2026



- Merge Gardens originally launched in 2020
- Promising initial scale
- Plateau in metrics in following years

In 2023..

- .. enter the revamped Merge Gardens
- New story, new marketing, new graphics, new ads, new.. everything?
- +1000% more downloads, mega influx of new and returning players

LIVE OPERATIONS (LiveOps): ... refers to ongoing management, updates, and community-driven events implemented post-launch to keep players engaged, maintain game health, and drive consistent growth and revenue



GAME
Merge

ENVIRONMENT
dev

- Home
- Dashboard
- Live Ops
- Campaigns
- Templates
- Frameworks
- Segments
- Environments
- A/B Test Designer**
- Players
- Control Center

U Berkay Uçak

Test Configuration

Set up your A/B test name and configure the distribution buckets

Test Name
Billion Dollar Test

Total Distribution: 100%
Must equal 100% across all buckets

DISTRIBUTE EVENLY RESET ALL

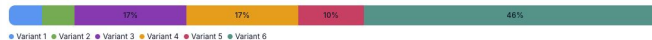
Perfect! Your A/B test distribution is ready.

Distribution Buckets

Configure your test variants and their traffic allocation

+ ADD BUCKET

- Variant 1 5 %
- Variant 2 5 %
- Variant 3 17 %
- Variant 4 17 %
- Variant 5 10 %
- Variant 6 46 %



CLEAR ALL

CREATE TEST

N

ID

74cd4aed-1f17-4865-94ec-b8c4dedc5712

Name

Xmas BG Campaign

Description

Start at

01/09/2024 22:00

End at

01/11/2025 23:39

Recurrence

Does Not Repeat

Active Environments

berkay

Segments

Variants +

Name

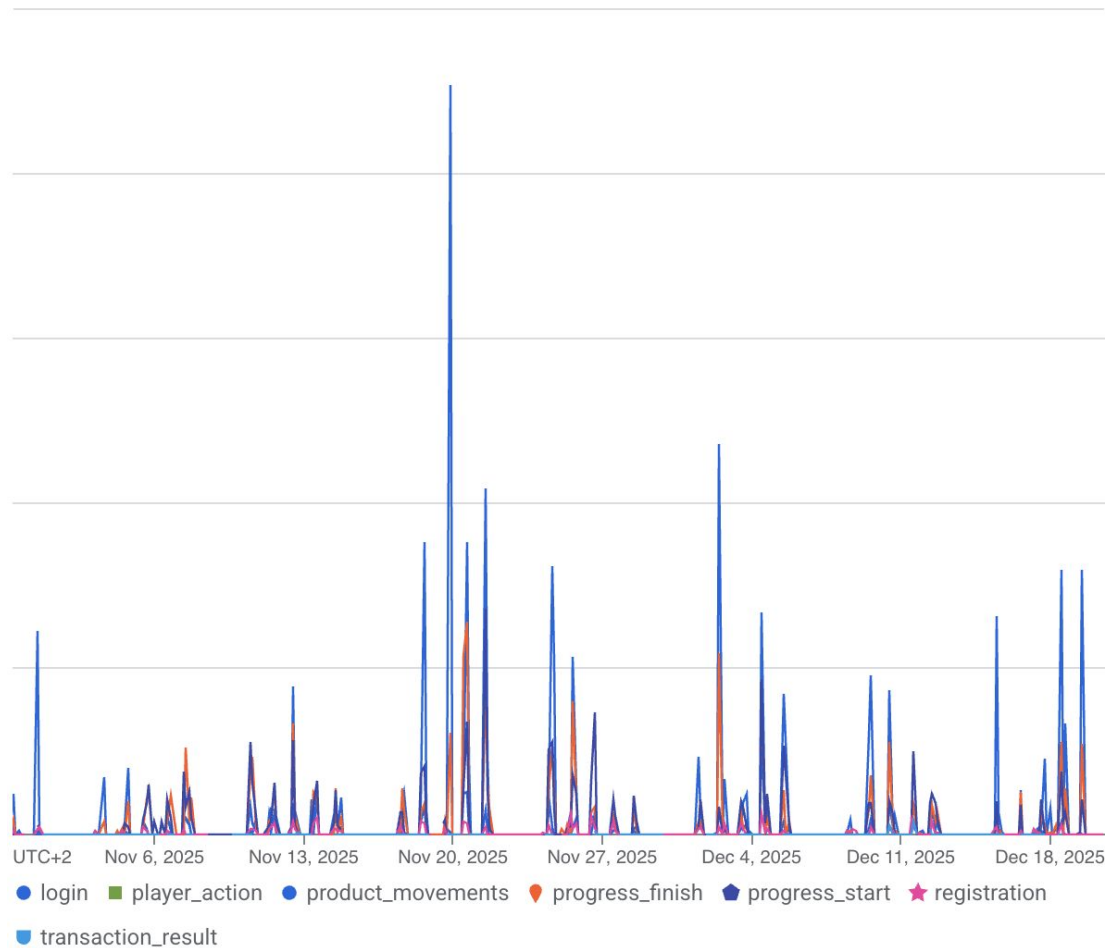
Xmas1BG

Description

Contents +

CHRISTMAS





Collected Points ⓘ



Rita's Fitness Scramble

Finished!

1 Garden #233 YOU 42

2 Player Name 40

3 Player Name 27

4 Player Name 12

5 Player Name 5

Claim

Congratulations!

Your Rank **1**

Reward



KubeCon



CloudNativeCon

Europe 2026

Part 2: What Survived?



The Challenge



KubeCon



CloudNativeCon

Europe 2026

- ... how about the tech?
- ... can we achieve it with a small team?
- ... can't be a challenge if it doesn't exist, right?

Not Quite... Greenfield projects come with their downsides:

- Cost and Timeline Overruns
- Uncertain ROI
- Over-Engineering
- Scope Creep
- Interfacing with Legacy Systems
- Team Experience
- Technology Obsolescence

Shopping List



KubeCon



CloudNativeCon

Europe 2026

- ☐ Highly available
- ☐ Maintainable with a small server team
- ☐ Scalable to react to changes in traffic
- ☐ Using tools that do the job in ways that are easy to understand and maintain
- ☐ Mostly stateless server architecture
- ☐ Vendor neutral tech
- ☐ Programming language w/ emphasis on performance and DX



Decision Journal as an Engineering Tool



KubeCon



CloudNativeCon

Europe 2026



- We logged every major decision we took with rationale and tradeoffs
- Every decision had an "ease-of-change" estimate (easy/medium/hard)
- We revisited decisions after certain milestones

Decision Journal

This page documents the topics and reasoning from researching and deciding on the backend tech and infra plan, described in more detail on other pages in this section.

The goal of this page is to have a log of the thought process to look back to when reconsidering any of the decisions in the future. As such, it can be good background information for anyone wanting to dig deeper into the thought process.

Technical Questions

1. [Server Stack & Architecture](#)
2. [Communication](#)
3. [Database\(s\)](#)
4. [Programming Language](#)
5. [Cloud Provider](#)
6. [Monitoring and logs](#)
7. [Transitioning a live game to use a new backend](#)
8. [Stateless and Stateful Approach](#)
9. [Choosing and Maintaining a Message Broker](#)
10. [Realtime Analytics](#)

Team/Process Questions

1. [On-call process](#)

Architecture Principle That Survived



KubeCon



CloudNativeCon

Europe 2026

- Loosely coupled, mostly stateless services on Kubernetes
- "Microservice" means single responsibility, not single endpoint
- Services scale independently
- Containers preserve optionality, even if Kubernetes remains the default

☐ Name	Namespace	Pods	Replicas	Age	Status	
☐ fp-backend-analytics	backend-prod	2/2	2	56d	Running	
☐ fp-backend-auth	backend-prod	3/3	3	253d	Running	
☐ fp-backend-cache	backend-prod	1/1	1	253d	Running	
☐ fp-backend-cloudsave	backend-prod	5/5	5	253d	Running	
☐ fp-backend-cloudsaveprocessor	backend-prod	5/5	5	253d	Running	
☐ fp-backend-config	backend-prod	2/2	2	253d	Running	
☐ fp-backend-friends	backend-prod	1/1	1	108d	Running	
☐ fp-backend-gift	backend-prod	2/2	2	253d	Running	
☐ fp-backend-iap	backend-prod	2/2	2	253d	Running	
☐ fp-backend-migration	backend-prod	1/1	1	253d	Running	
☐ fp-backend-privacy	backend-prod	1/1	1	237d	Running	
☐ fp-backend-time	backend-prod	1/1	1	253d	Running	
☐ fp-backend-tournament	backend-prod	3/3	3	253d	Running	
☐ fp-backend-webshop	backend-prod	2/2	2	171d	Running	
☐ fp-backend-websocket	backend-prod	4/4	4	253d	Running	

Why Kubernetes Stayed



KubeCon



CloudNativeCon

Europe 2026



Ease of change: Medium/Hard (but unlikely to be needed)

Kubernetes is fast becoming a de-facto standard in architecting cloud native software these days. As such, it's unlikely that we'd want to shift away from it. However, code in containers can be run also without Kubernetes if there's a reason to transition away from it. Changing the size of services (combining or splitting parts of the system) can and should be done as needed, as a part of a continuous improvement / development practice.

- + Flexibility for evolving LiveOps services
- + High availability
- + Vast ecosystem and tooling
- + Operational familiarity within the team
- Initial time sink investment
- Infrastructure and operational costs
- Some GKE platform limitations

Communication Strategy: Use the Simplest Tool That Fits



KubeCon



CloudNativeCon

Europe 2026

- Client-server: HTTP REST API and Websockets
- Server-server: choose by workload shape
 - Synchronous HTTP when response is needed immediately
 - Event streams when decoupling and load distribution are needed

 In server-to-server communication, use the right tool for the right job (= simplest approach that does the job). E.g. event streams are used when it helps the system respond more quickly and distribute load more evenly. At the same time, in a synchronous request, HTTP is often the cleaner way.

Topics considered:

1. Being pragmatic and using tools that do the job in ways that are easy to understand and maintain is usually the best way forward
 - a. E.g. don't use queues when the result of the queued operation needs to be returned to the client in the request's response
 - b. GRPC or similar socket based solutions are good for an online (multiplayer) game (mode). For an asynchronous approach with low time pressures, HTTP works just fine.
2. One approach, however, is not the best for everything. For example, it would make sense to use an event stream based architecture for passing analytics events through the server, from the client to the data platform.

Stateful Additions: WebSockets + Message Broker



KubeCon



CloudNativeCon

Europe 2026

- Trigger: server-initiated client messaging needs
- Added WebSocket service for persistent channel lifecycle
 - Minimal implementation: few pods with 50MB memory and near zero CPU usage, hundreds of thousands of players and open sockets
- Added message broker (Kafka)
- Tradeoffs accepted: more resource pressure and Kafka operations

2. Could we have done this with other approaches than WebSockets?

- a. Yes we could've. We could've easily implement a "polling" system from the client, which would prompt a request to server every X or Y seconds or minutes. But given the benefits of having an "always-on" channel to client for future features and drawbacks of any given polling system, we've decided to stick with this approach for this release.

Data Strategy: MongoDB First, Specialized Stores When Needed



KubeCon



CloudNativeCon

Europe 2026

MongoDB Atlas as main operational database

- Why it fit: player centric document model and good in-memory behavior
- Atlas benefits: managed scaling/resharding, lower ops burden



Ease of change: Hard (but unlikely to be needed)

Once enough data has been collected to a database, moving it to another will always require a well thought out migration plan. It's doable, but will take a lot of time and effort. With Atlas, there is the possibility of moving to a self-hosted MongoDB if needed, by doing a migration at the database level (e.g. by adding our in-house DB in the same replica set with the Atlas databases).

Also, while this decision is big, it leaves us the chance of using a different database for different sub services.

Principle remained: use another database where better suited

- Analytics scale: ~200-300M events/day estimate -> ClickHouse as the choice

Observability and On-Call Model That Worked



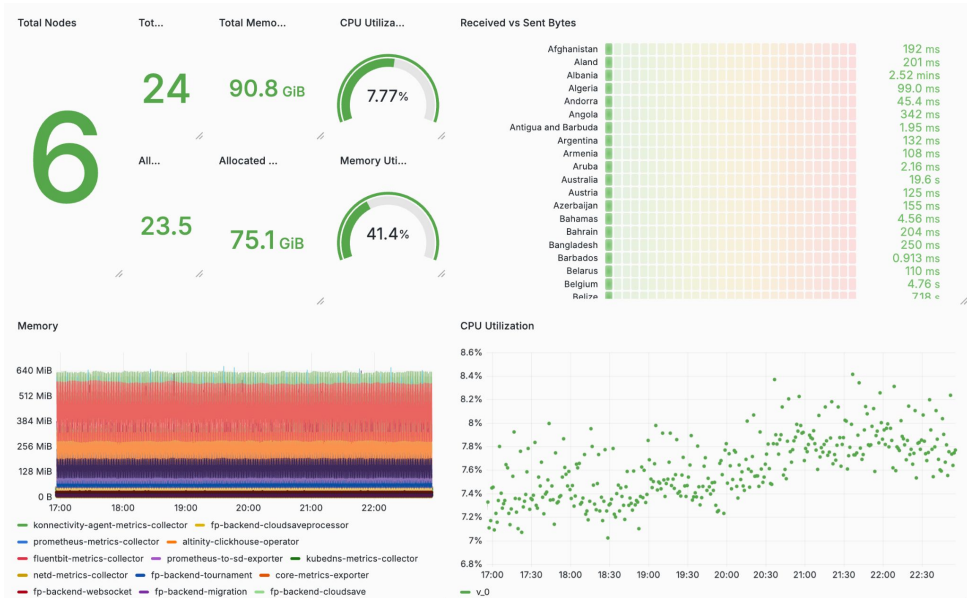
KubeCon



CloudNativeCon

Europe 2026

- App-level OpenTelemetry for vendor-agnostic instrumentation
- GCP Managed Prometheus + Grafana Cloud for metrics and alerting
- Cloud Logging for aggregation and search
- PagerDuty-integrated rotating on-call model



DevEx and Local Flows

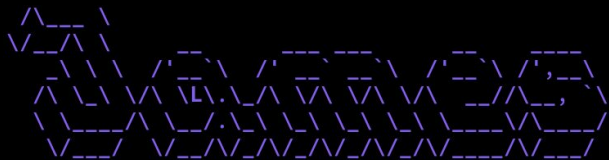


KubeCon



CloudNativeCon

Europe 2026



Initialize Configuration

Set up James CLI for your environment

> Tenant Management

Create, list, build, and deploy tenants

OpenAPI Operations

Generate SDKs from OpenAPI specs

up/down navigate enter select esc back q quit

james

home > tenant

> List Tenants

View all tenants and their status

Create Tenant

Create a new tenant in the cluster

Build Tenant

Build services for a tenant

Deploy Tenant

Deploy tenant applications

up/down navigate enter select esc back q quit

DevEx and Local Flows



KubeCon



CloudNativeCon

Europe 2026

```
james
home > tenant > list
```

Name	URL	Status
berkay	https://gateway.dev.merge.futureplay.fi/berkay	DOWN
iankko	https://gateway.dev.merge.futureplay.fi/iankko	DOWN

```
>_- build-to-tenant.sh
```

```
james
home > tenant > build

Tenant name
>

Service to build
> All services
auth
config
analytics
tournament
iap
time
websocket
migration
gift
cache
playground
cloudsave
webshop
admin
cloudsaveprocessor
privacy

Deploy via helmfile after build?
Yes No

Start build?
Yes No

enter next
```



KubeCon



CloudNativeCon

Europe 2026

Part 3: What failed (fast)?



How (not) to Redis



KubeCon



CloudNativeCon

Europe 2026

Campaign cache -> Application memory

Segmentation cache -> Application memory

Player session information -> Cache service

```
1 package domain
2
3 import (
4     "fmt"
5     "sync"
6 )
7
8 var cache sync.Map // 5 usages @ Berkay Uckac
9
10 func Get(key string) (string, error) { // 2 usages @ Berkay Uckac
11     value, ok := cache.Load(key)
12     if !ok { return "", fmt.Errorf("format: key not found") }
13
14     return string(value.([]byte)), nil
15 }
16
17 func Set(key string, value []byte) error { // 1 usage @ Berkay Uckac
18     cache.Store(key, value)
19
20     return nil
21 }
22
23 func Delete(key string) { // 1 usage @ Berkay Uckac +1
24     if _, ok := cache.Load(key); !ok { return }
25
26     cache.Delete(key)
27 }
28
29 }
```

Incident: Cloud Save Partial Outage (March 2025)



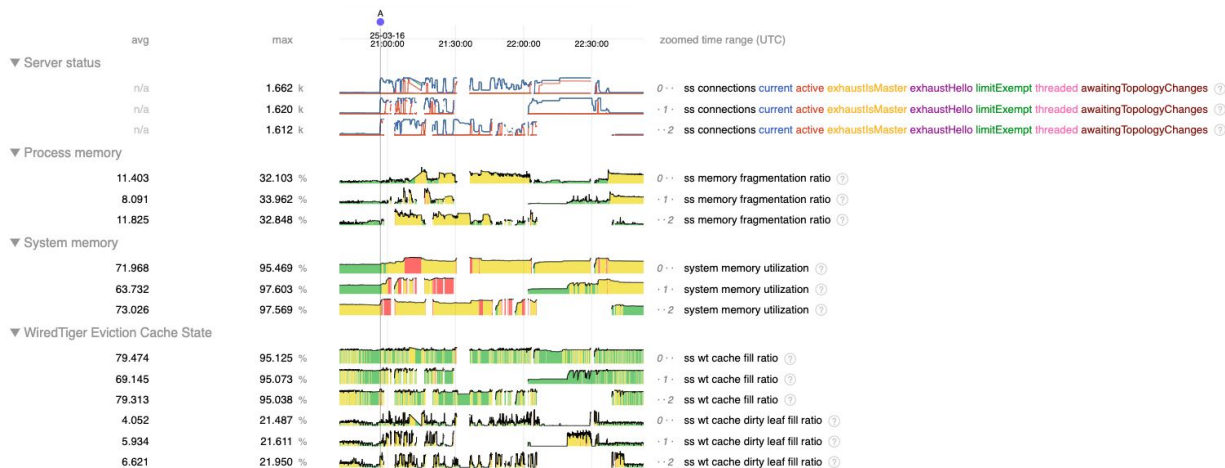
KubeCon



CloudNativeCon

Europe 2026

- Player impact: subset stuck at 49% loading (new sessions)
- Core failure chain:
 - sudden spike (~300% in seconds)
 - cloudsaver pod OOM/restarts
 - retry/reconnect burst to Mongo -> connection pool timeout
 - WiredTiger cache saturation and mongod OOM



What We Changed After the Incident



KubeCon



CloudNativeCon

Europe 2026

Controlled traffic better: stronger rate limiting, safer client retries

Made dependencies more resilient: Mongo driver overhaul, circuit breakers

Simplified the critical path: backup writes moved off the save path

Measured impact better: improved SLO formulas for partial incidents

Incident: Config Service Config (February 2026)

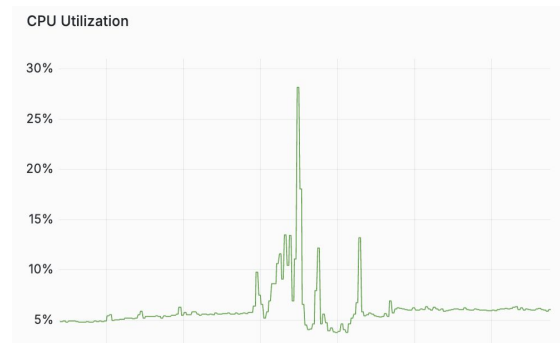
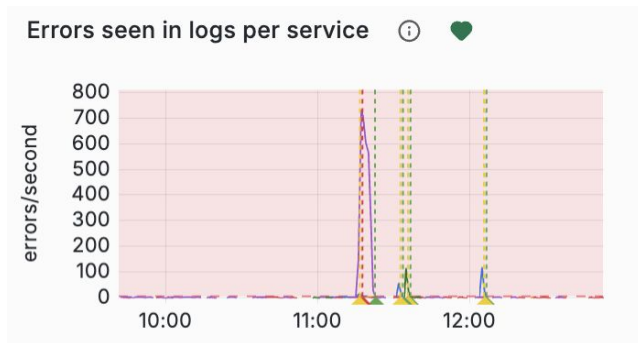
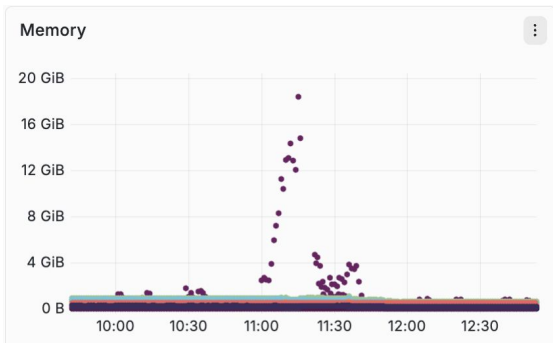


KubeCon



CloudNativeCon

Europe 2026



deploy new service resource values -> don't adjust HPA -> config pod caches trigger HPA -> spawn more config service pods -> more cache requests -> more traffic -> more traffic = brings down admin -> admin down == config down -> pods not ready -> more pods -> more cache requests -> more traffic = admin even more down... repeat

all this == !profit

... and more

Bad assumptions before launch

- Underestimating burst behavior and retry storms
- Underestimating happy client-server paths

Scaling mistakes

- Expecting default HPA to solve complex burst patterns alone
(<https://github.com/berkayuckac/k8sschedul8r>)

Complexity we steered away from

- Service Mesh
- VPA



KubeCon



CloudNativeCon

Europe 2026

Part 4: What small teams should copy

- Choose primitives you can operate confidently
- Document decisions with explicit change difficulty
- Keep sync paths simple; add async only where it buys extra
- Build kill switches and phased rollout before launch

Simplicity is the uptime strategy

Scale and Reliability Outcomes



KubeCon



CloudNativeCon

Europe 2026

Player Count: **~10M**

Campaigns, AB tests, Collected Items & Tournament Points : *too much*

1Y SLO: 99.99%

- 2023: **100%***
- 2024: **99.99896%** (~5m downtime)
- 2025: **99.9992083%** (~4m 9.8s downtime)

What next?

- Hybrid workloads
- Redis?
- ArgoCD



KubeCon



CloudNativeCon

Europe 2026

Thanks! Q&A

www.berkay.fi
hey@berkay.fi

