GDC

Rocket League: Scaling for Free to Play

Matthew Sanders
Lead Online Services Engineer / Psyonix

The Team & The Problem

- What does the Online Services team do?
- What is Free to Play (F2P) for Rocket League?
  - $20 → $0
  - Lower barrier to entry

The Journey

- Part 1: Planning and Preparation
  - Partnerships
  - Analysis
  - Load testing
- Part 2: Scaling Improvements
- Part 3: Launch Learnings
  - F2P launch experience
  - The new normal

Part 1: Planning and Preparation

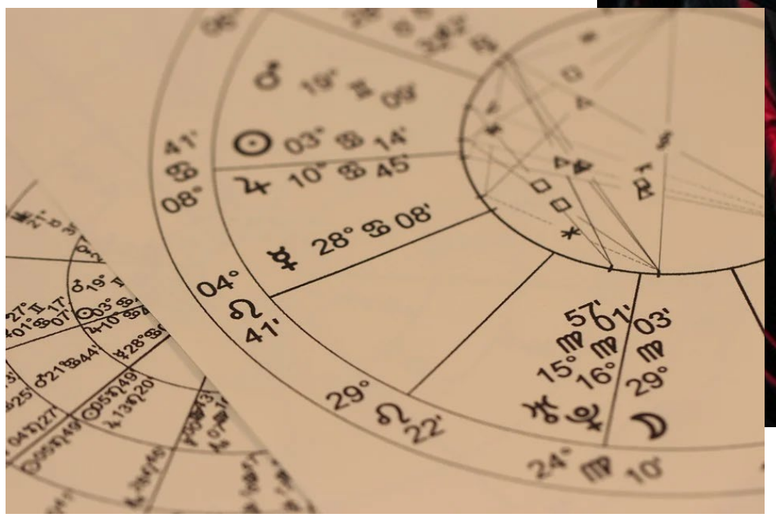
"We have top men working on it right now."
-Major Eaton

Planning and Preparation - Overview

- Analysis
  - Load projections
  - Architecture evaluation
- Partnerships
  - Support contracts
  - Outside advice
- Leveling up load testing

Analysis - Load projections

- How many new players will come?
  - It’s not divination
  - Find a comparable game release
- Let’s go with “3x to 5x”
- 5x became our load testing target



Analysis - Architecture Evaluation

- How does each component work?
  - Is the design scalable?
  - Backlog any identified risks
- Create a template
- Check your work
  - Peer review
  - Source control audit

Partnerships - Support Contracts

- Expert specialists
  - Cost comparison: Support versus Downtime?
- Google
  - Google Cloud Platform (GCP)
  - Customer Reliability Engineering (CRE)
  - Game Title Launch Assist (GTLA)

redislabs
HOME OF REDIS

Partnerships - Outside Advice

- Google’s CRE and GTLA
  - Premortem exercise
  - Resource planning and quotas
- Redis Labs
  - Extensive technical Q&A
  - Migration support

Load Testing

- Most important
- Most underestimated
  - Unique
  - Expensive
  - Hard to get right
    - Request patterns
    - Request volumes

Load Testing - Leveling Up!

- Our framework of choice:
  - Web sockets
  - Kubernetes
  - 1 locust = 1 client
- Developing a new codebase
  - Locust coordination
  - Bad inputs = bad simulation

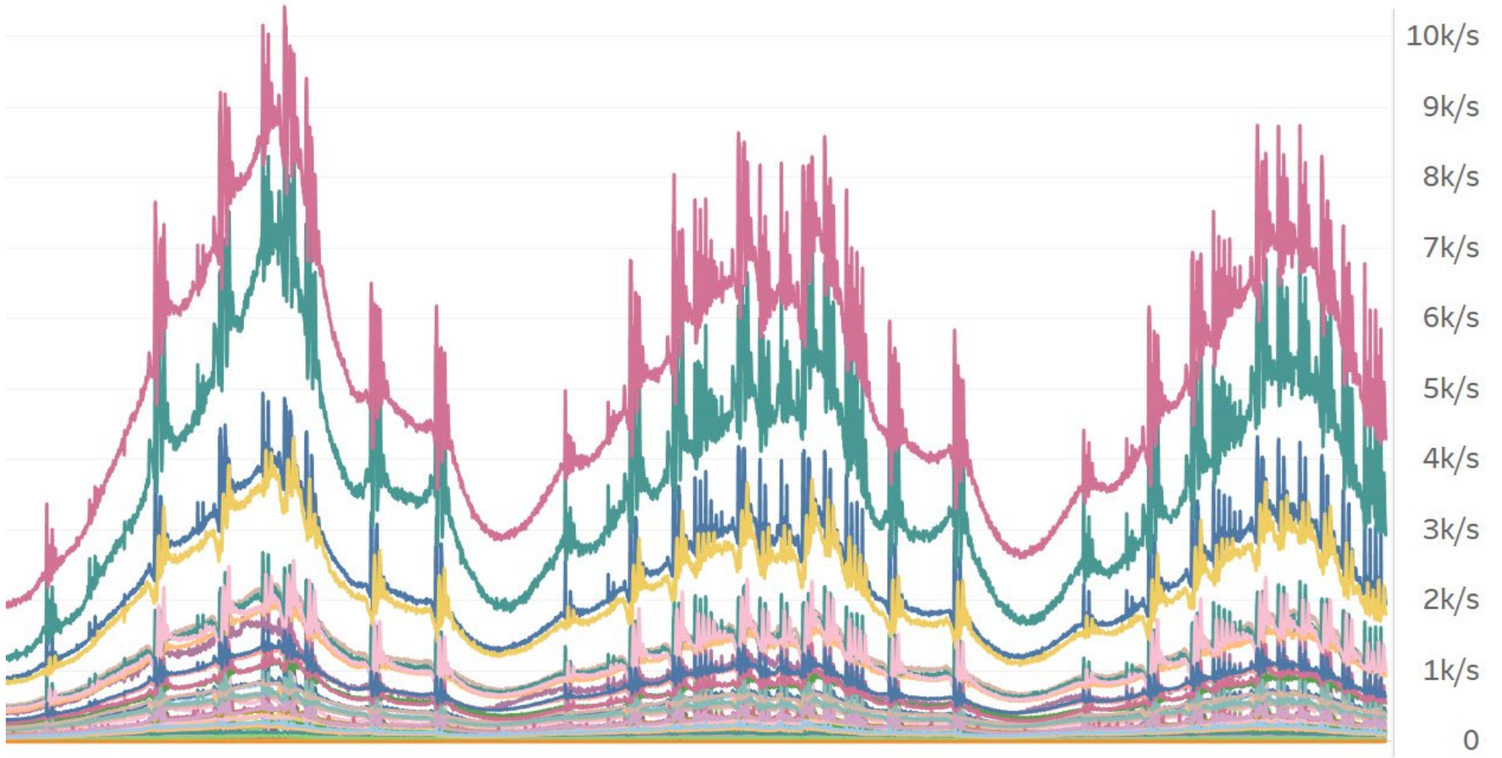
“Python… Very dangerous.
You go first!”
-Sallah

Load Testing - Leveling Up!

- Time consuming, especially at full scale
  - Linear scale up time
  - Run time
  - Iteration count

Load Testing - Gathering Results

- Data to gather:
  - HTTP response codes
  - Service logs
  - Resource usage
- Is it accurate?
- Led to dozens of sequential fixes



Load Testing - In Perpetuity

- Keep testing until the release
  - It is better to know, even if last minute
  - Mitigate instead!
- SDLC integration

Part 2: Scaling Improvements

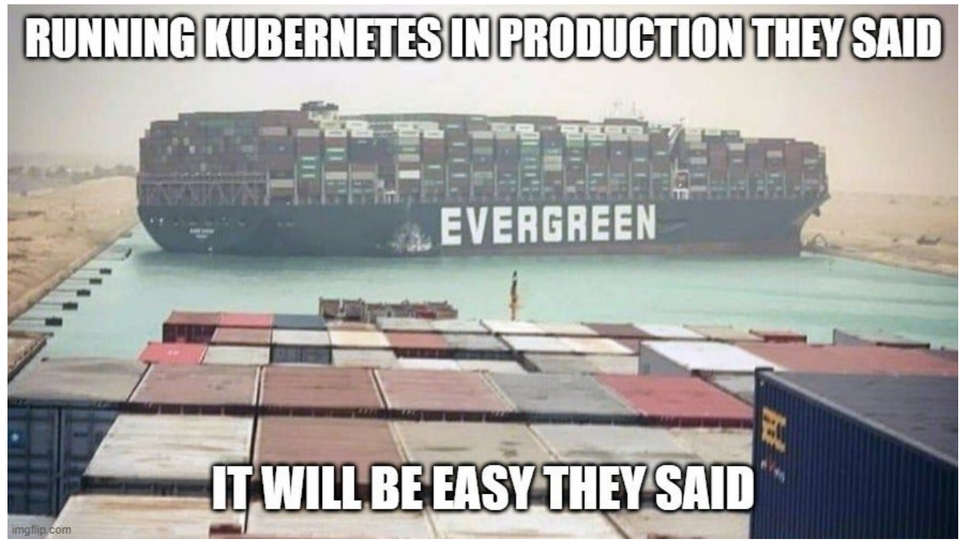
"I had bugs for lunch."
-Willie Scott

Scaling Improvements - Overview

- Core Services to Kubernetes
- Matchmaking overhaul
- Redis Enterprise migration
- MySQL improvements
- Rate Limiting
- Development Postmortem

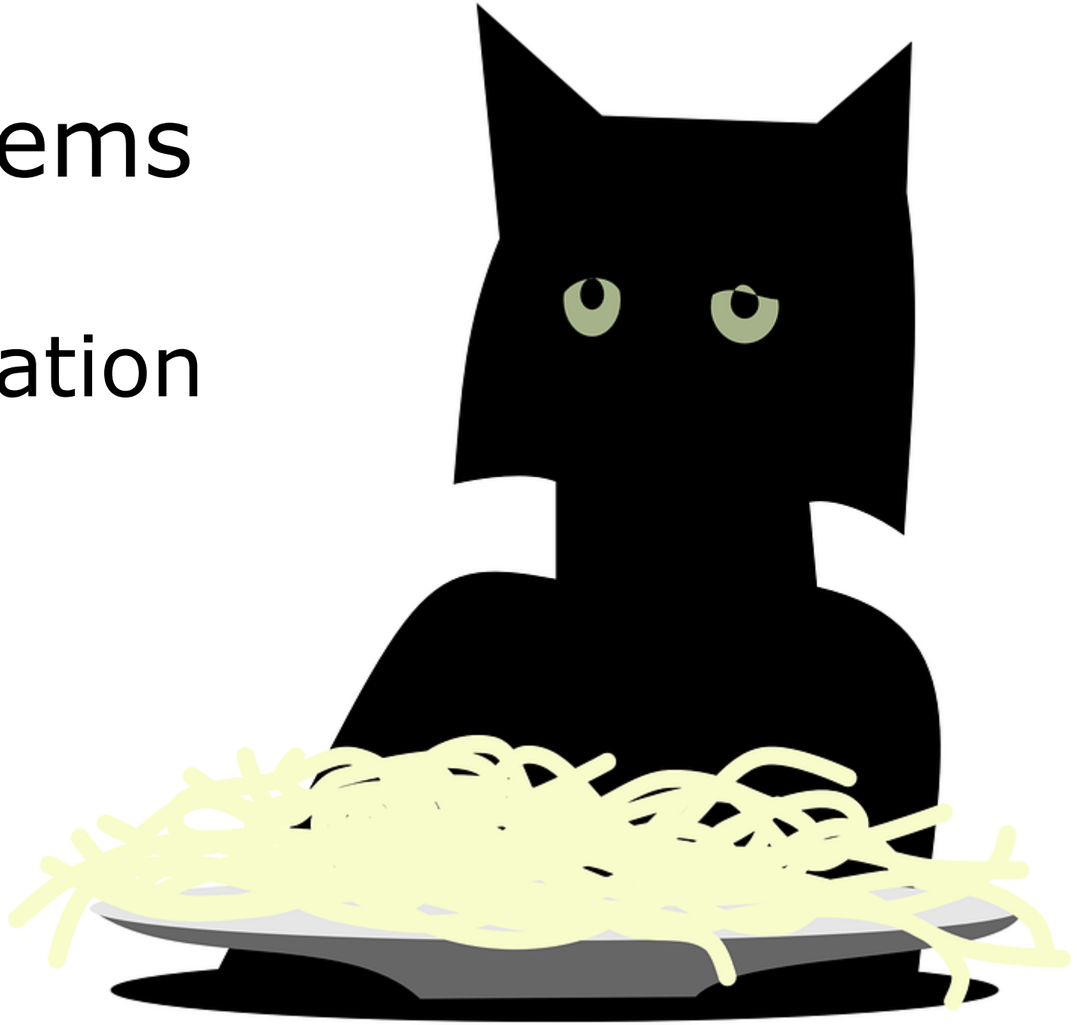
Core Services to Kubernetes

- Live migration from Google AppEngine (GAE) to Google Kubernetes Engine (GKE)
  - Control over the runtime and scaling
  - Consistency and cloud-agnostic
- Separate GCP projects
  - Clarity, simplicity, security
- Load testing: indispensable!



Matchmaking Overhaul

- Known performance problems
- Our Matchmaking service:
  - Single-threaded .NET application
  - Already “maxed out”
  - Mostly spaghetti by volume



Matchmaking Overhaul

- MapReduce seems like a good fit
- We found OpenMatch:
  - Designed for scaling
  - Orchestrated set of containers
  - Cutting edge (pre-1.0)
- Load testing… again
  - Tune it in GKE
  - Learn the dynamics

open match

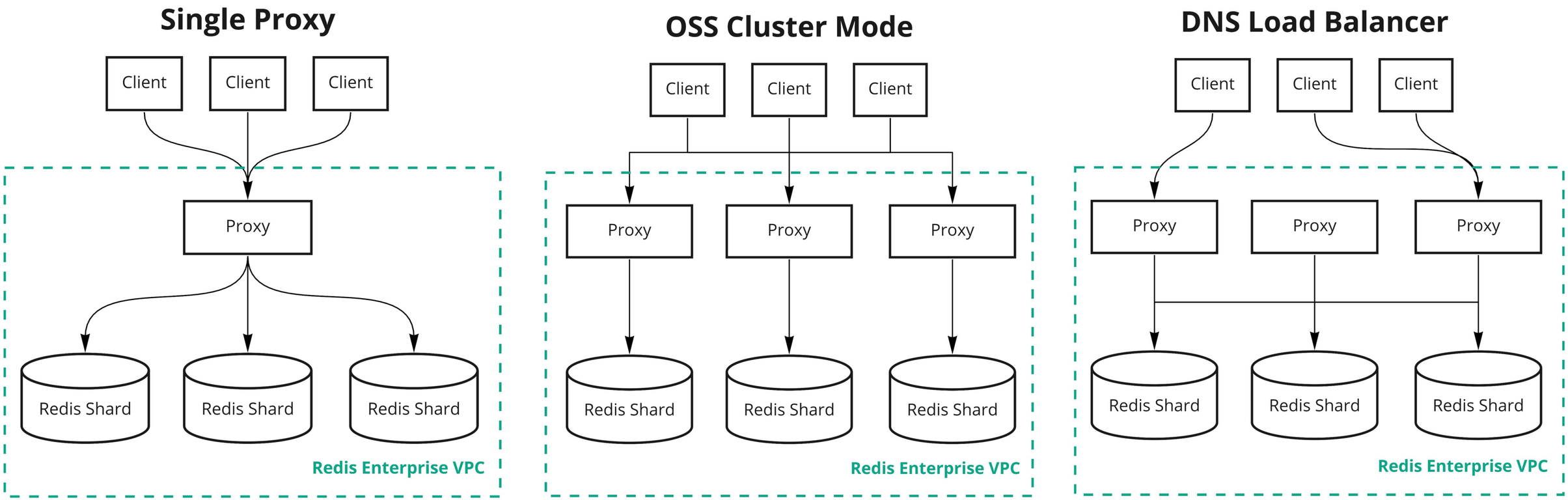
Redis Enterprise Migration

- Fully automated re-sharding
  - We absolutely used this!
  - Beware its performance cost

ENTERPRISE

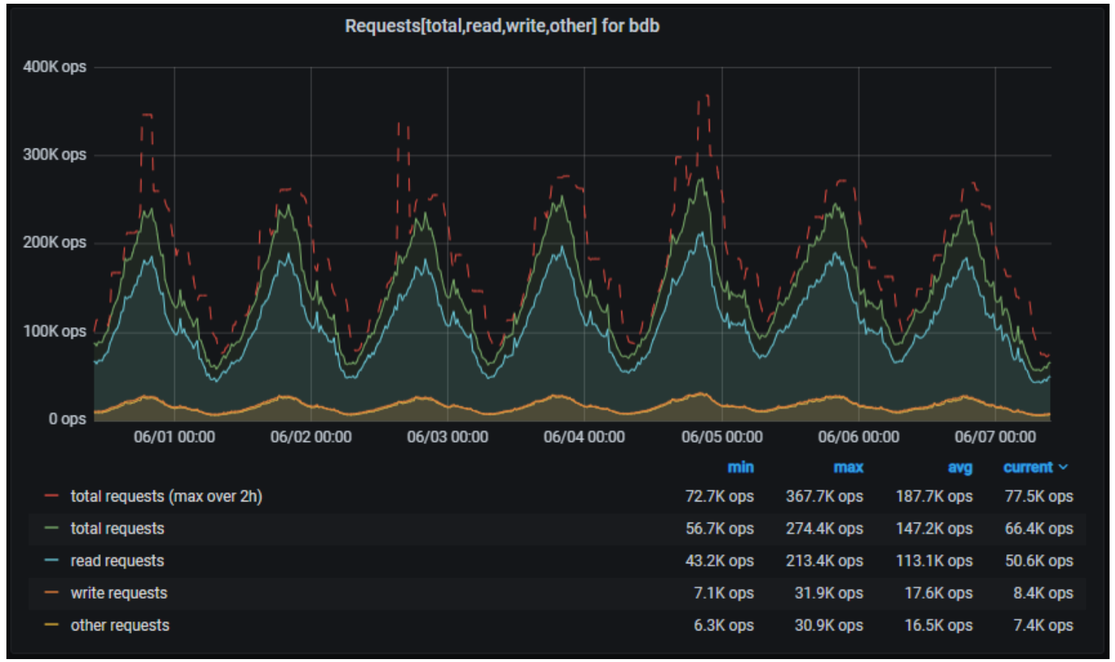
Redis Enterprise Migration

- Proxy configurations:



Redis Enterprise Migration

- Adjustments
  - Observability of commands
  - Monitoring and dashboards
- Other improvements
  - Pub/Sub optimizations
  - Temporary replicas
  - No vendor lock-in



MySQL Improvements

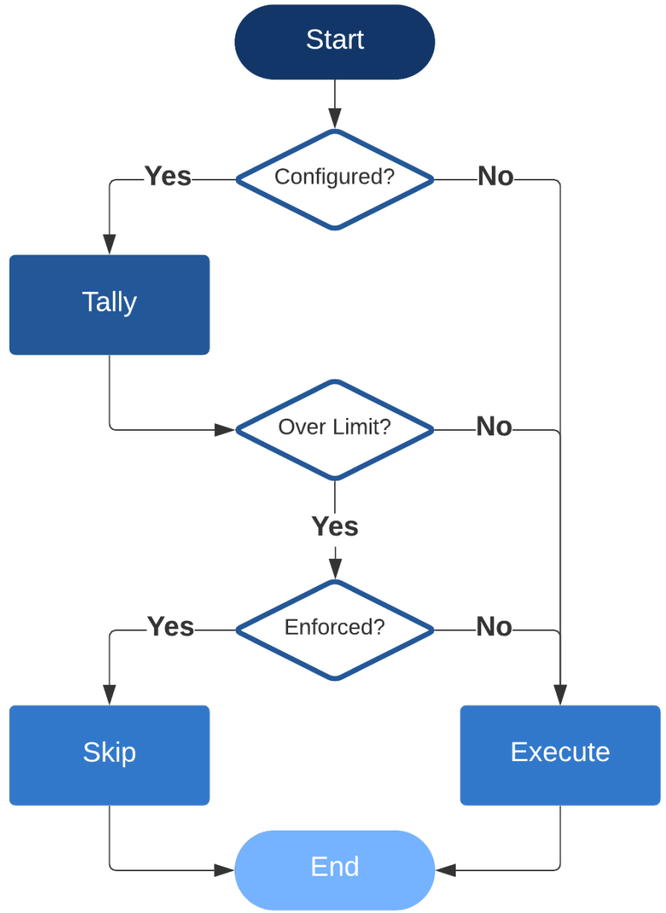
- Vertical scaling only, with replicas
- “Feature shards”
  - Sets of tables on different servers
  - Effort: Easy, usually
  - Does not solve vertical scaling problem
- Insufficient time to migrate horizontal scaling
  - Vitess was attractive, but not exactly drop-in
  - However, we reached our load test targets!

MySQL Improvements

- ProxySQL
  - Scale out as many as we need
  - Offloads connection churn
  - Instant failovers
  - Pros
    - Speaks MySQL
    - Dynamic query routing
  - Cons
    - Some bugs
    - Learning curve

Rate Limiting

- Started life as a Login Queue
- Rate limit session creation instead
- Custom implementation
  - Simple logic
  - Integrates with our infrastructure
- More versatile in the end
  - Limit any set of services
  - Global or per-player limits



Development Postmortem

- Where did the months go?
- Work-from-home transition
- High effort for major features!
- Don’t over-analyze
- Intermediate deadlines
  - Define
  - Respect
  - Coordinate

Part 3: Launch Learnings

"Let my armies be the ***engineers***, and the ***code***,
and the ***services in the cloud***…"
-Charlemagne

Launch Schedule

September 2020

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16 Game Update	17	18	19
20	21	22 Hotfix	23 F2P Launch	24	25	26
27	28	29	30			

Launch Boost!

- Look at that population!
- Sunday peak was over projected 5x increase but lower than our highest successful load test
- Wednesday launch was ideal

Launch Outages and Issues

1. Wednesday: Epic Accounts sign-in
2. Thursday: Redis Enterprise re-sharding
3. Friday: ProxySQL misconfiguration
4. No outages over the weekend!

Launch Feelings

- Tension and confidence
- New all-time peak every day
  - Accurate 5x projection!
- Zoom war room
  - Kept busy fixing smaller issues

The New Normal

- The population increase is still 3x
- Higher population makes services sensitive
  - Code AND configuration
  - Continue load testing!
- Renew best practices
  - Design of live migrations
  - Pushing out changes
  - Always have a rollback plan

Takeaways

"An army which carries the ***Load Tests*** before it…
is invincible."
-Marcus Brody

Takeaways

- Stay organized early on
  - Do seek support and advice
  - Don’t over-plan
- Load testing is hard and super important
- Coordinate big features
  - Stick to intermediate deadlines
- Implement versatile controls like rate limiting

Thank you for attending!