Let’s Design Ticketmaster!

Phil Lopreiato, Pat Cody

Why are we doing this?

- Eventually, you will probably have to do this in an interview
- Designing systems that scale is important
- If you don’t know how to monitor a system and prove it’s working, it’s probably broken

Warmup: How do we sell physical tickets?

- Capital One Arena is about to host a concert
- Imagine it’s like 30 years ago, and there’s no internet- how can we structure the ticket booth at the arena to sell physical tickets?

Some Questions to Get You Started

- How do people know when it’s their turn?
- How do we ensure fairness? (Ticketmaster’s specialty)
- How can we maximize the number of tickets sold?

Some Possible Answers

- Multiple ticket booths to improve throughput
- A queue
  - Follow-up question: should we use one queue for all the ticket booths? Or one queue for each? What happens if one person is really slow?
- Limit number of tickets per person

System Design: It’s a little like Lego!

- System design involves combining different “building blocks” together to build a thing
- With **distributed** system design, those components are running on different machines
- **Question:** What are some different components that make up a distributed system?

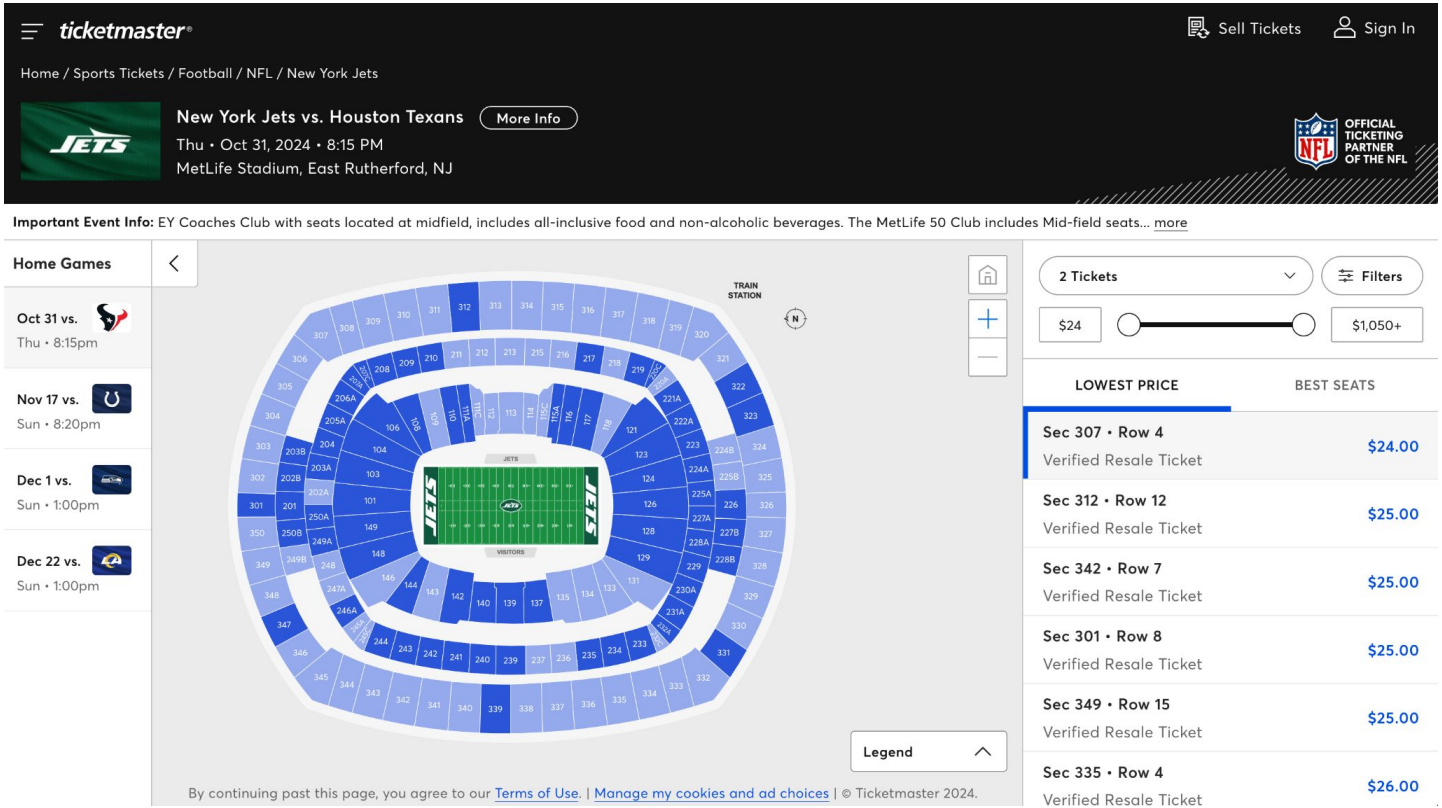
System Design: It’s a little like Lego!

- Some possible answers:
  - A DB
    - What does it store? How does it scale?
  - Front-end
  - Load balancer
  - Cache
  - Queue
  - Periodic jobs
    - Cron!
  - Atomic operations
    - Check and set
    - DB locking
  - “Business logic”

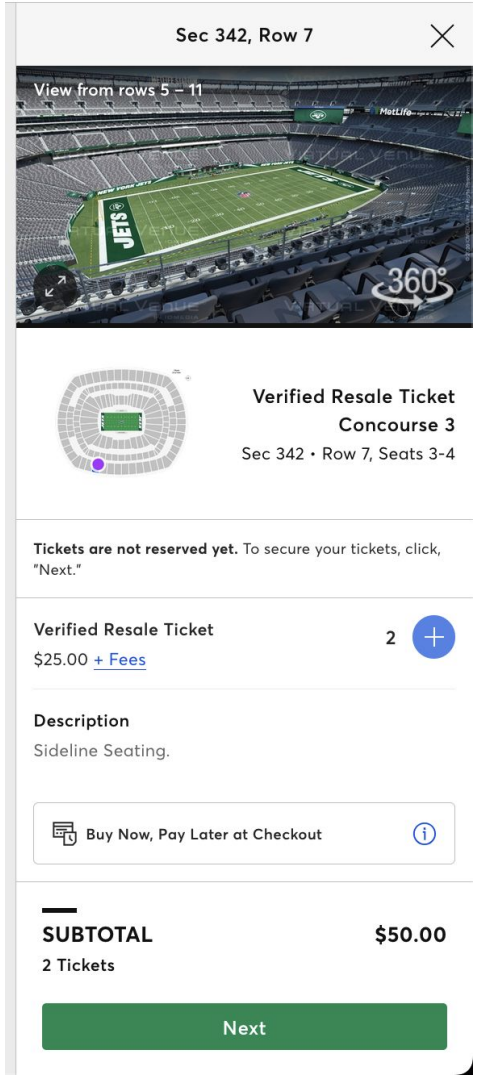
Problem Constraints

- For some reason, Phil is trying to go to a Jets game, how can the stadium sell him a ticket?
- We don’t have to design a payment system (we can assume that exists)
- Once you start reserving a ticket, you have 5 minutes to complete the transaction before the tickets are released to someone else
- We can start with general admission tickets (no seat reservations), and move on to reserved seating

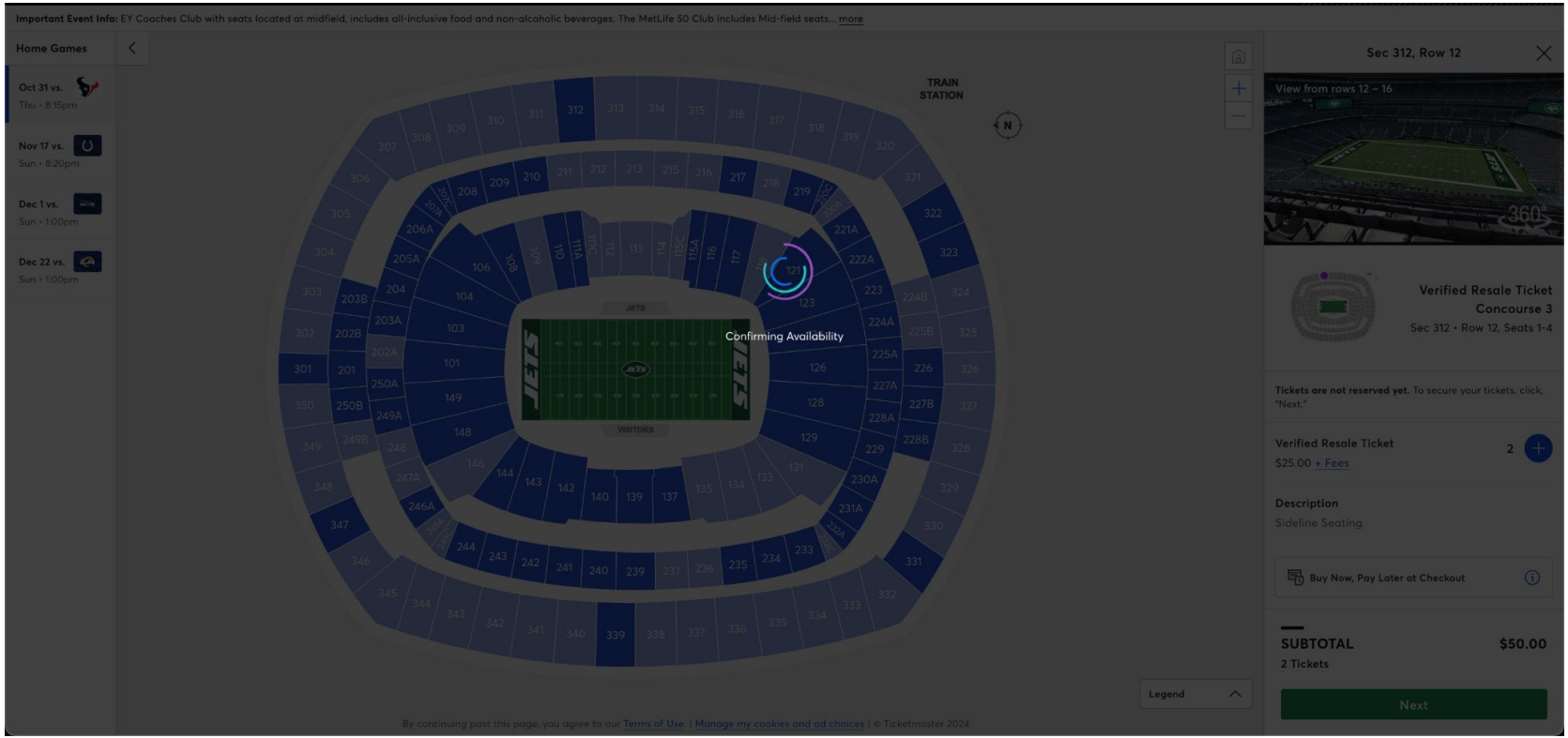
Ticket Selection



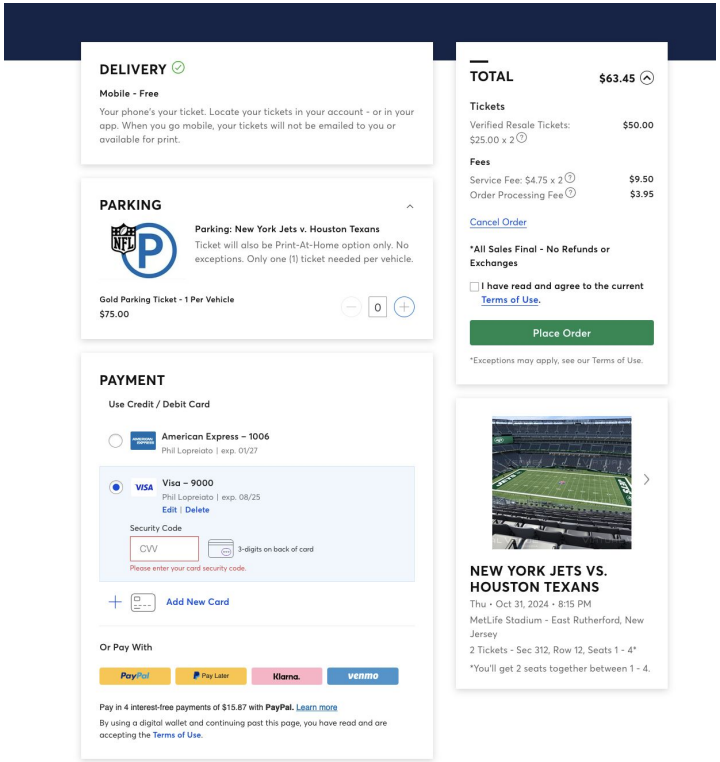
Needs Reservation



Confirming


Important Event Info: EY Coaches Club with seats located at midfield, includes all-inclusive food and non-alcoholic beverages. The MetLife 50 Club includes Mid-field seats... more
Home Games
Oct 31 vs.
Thu • 8:15pm
Nov 17 vs.
Sun • 8:20pm
Dec 1 vs.
Sun • 1:00pm
Dec 22 vs.
Sun • 1:00pm
TRAIN STATION
Confirming Availability
Legend
Sec 312, Row 12
View from rows 12 – 16
360°
Verified Resale Ticket
Concourse 3
Sec 312 • Row 12, Seats 1-4
Tickets are not reserved yet. To secure your tickets, click, "Next."
Verified Resale Ticket
$25.00 + Fees
2
Description
Sideline Seating.
Buy Now, Pay Later at Checkout
SUBTOTAL
$50.00
2 Tickets
Next
By continuing past this page, you agree to our Terms of Use. | Manage my cookies and ad choices | © Ticketmaster 2024.

Checkout


DELIVERY
Mobile - Free
Your phone's your ticket. Locate your tickets in your account - or in your app. When you go mobile, your tickets will not be emailed to you or available for print.
PARKING
Parking: New York Jets v. Houston Texans
Ticket will also be Print-At-Home option only. No exceptions. Only one (1) ticket needed per vehicle.
Gold Parking Ticket - 1 Per Vehicle
$75.00
0
PAYMENT
Use Credit / Debit Card
American Express – 1006
Phil Lopreiato | exp. 01/27
Visa – 9000
Phil Lopreiato | exp. 08/25
Edit | Delete
Security Code
CVV
3-digits on back of card
Please enter your card security code.
Add New Card
Or Pay With
PayPal
Pay Later
Klarna.
venmo
Pay in 4 interest-free payments of $15.87 with PayPal. Learn more
By using a digital wallet and continuing past this page, you have read and are accepting the Terms of Use.
TOTAL
$63.45
Tickets
Verified Resale Tickets: $25.00 x 2
$50.00
Fees
Service Fee: $4.75 x 2
$9.50
Order Processing Fee
$3.95
Cancel Order
*All Sales Final - No Refunds or Exchanges
I have read and agree to the current Terms of Use.
Place Order
*Exceptions may apply, see our Terms of Use.
NEW YORK JETS VS. HOUSTON TEXANS
Thu • Oct 31, 2024 • 8:15 PM
MetLife Stadium - East Rutherford, New Jersey
2 Tickets - Sec 312, Row 12, Seats 1 - 4*
*You'll get 2 seats together between 1 - 4.

Let’s draw a system diagram!

Remember our problem constraints:

- We don’t have to design a payment system (we can assume that exists)
- Once you start reserving a ticket, you have 5 minutes to complete the transaction before the tickets are released to someone else
- We can start with general admission tickets (no seat reservations), and move on to reserved seating

Monitoring - 4 Golden Signals

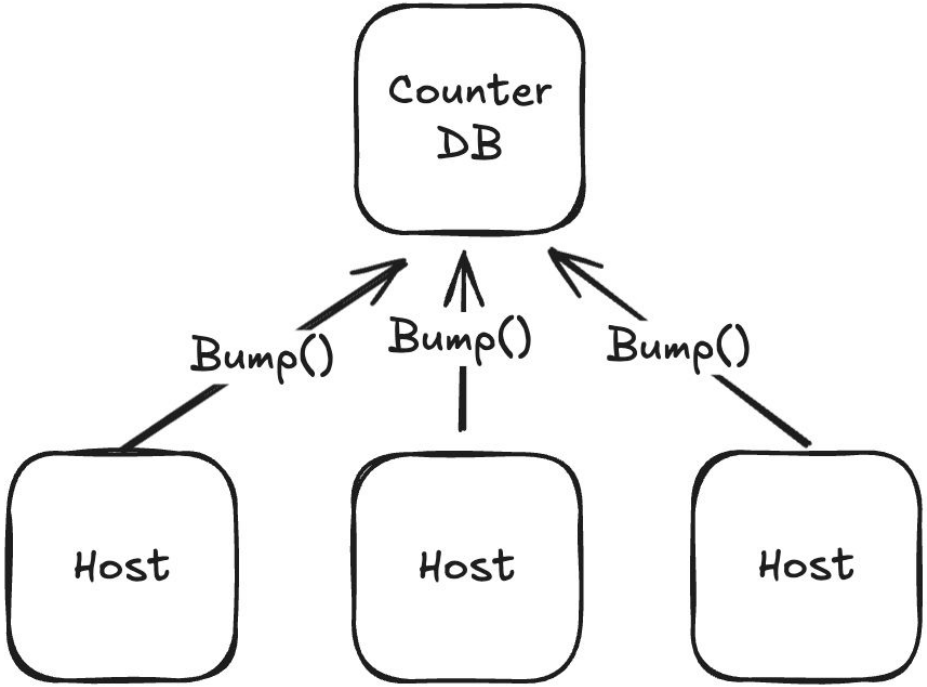
- Latency - how long does a user flow take to complete
- Traffic - are we actually serving requests to people?
- Errors - are requests failing?
- Saturation - are any of our limiting resources constrained?

The Best Resource: https://sre.google/sre-book/monitoring-distributed-systems/

Monitoring Building Blocks

- Time series - maintain a counter of “hits” over time
- Structured logging - key/value pairs or database rows of “events”
- System logs - low level signals or print statements from code
- Distributed tracing - track interactions across components

Distributed Logging



Monitoring Our System In Production

- How do we apply the 4 Golden Signals?

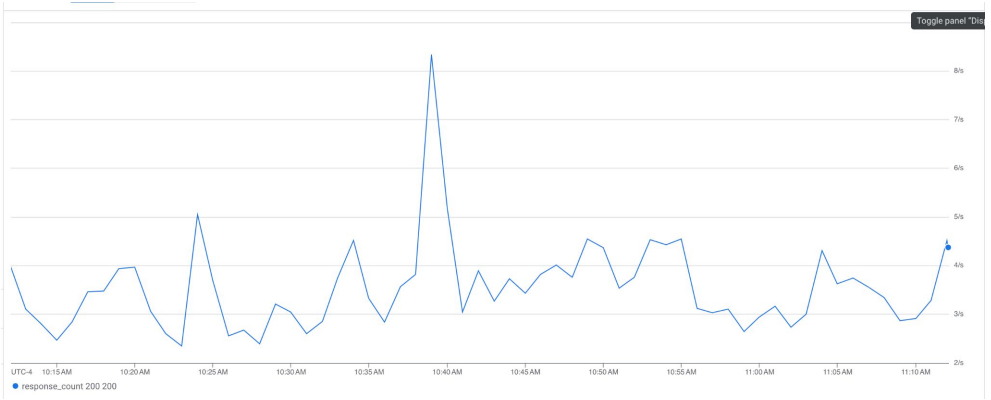
  - Latency: how long do the DB operations take
  - Traffic: are we completing operations
  - Errors: are any operations failing
    - Some errors are expected! (eg racing reservations for the same ticket) How do we classify those?
  - Saturation: CPU/memory usage of servers
    - What are the system bottlenecks?

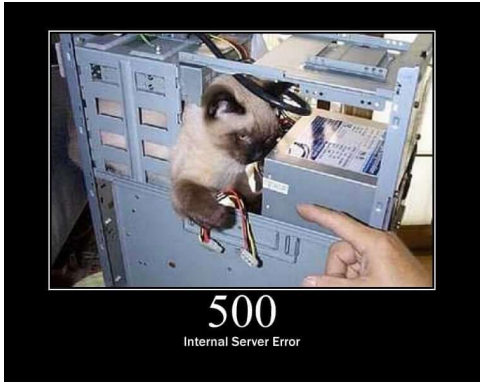
Debugging Issues

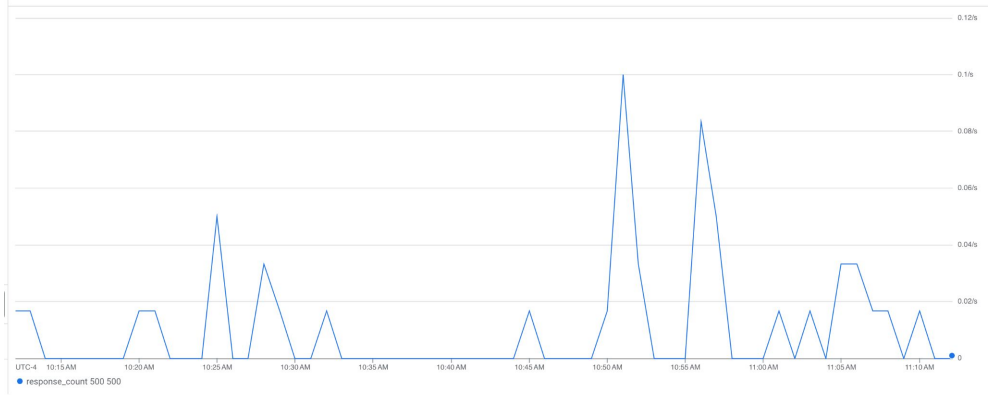
One of the Golden Signals has tripped … How do we fix it? What data do we need?

- Counters – what is the magnitude of the problem, which code branches are we taking
- Error logging – exception messages, stack traces
- Tracing – can we see how a user flow progressed / failed

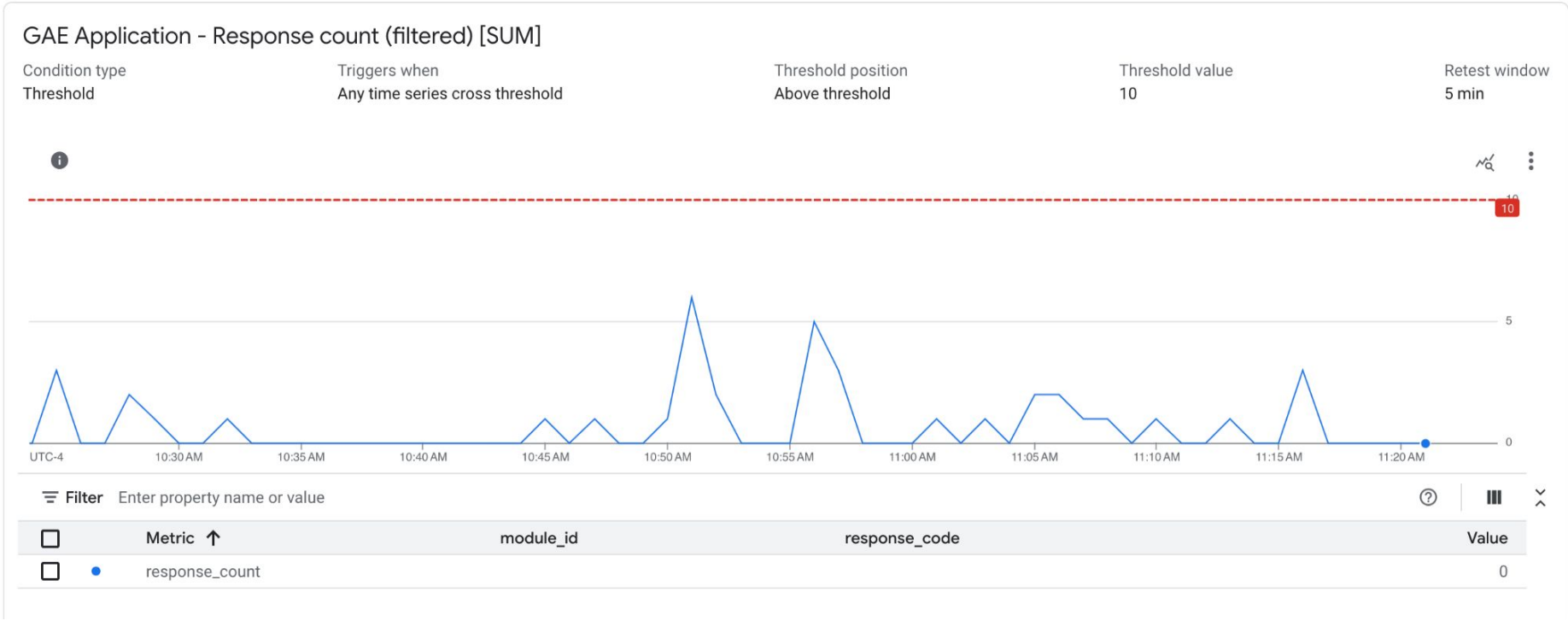
Example: HTTP Code Time Series



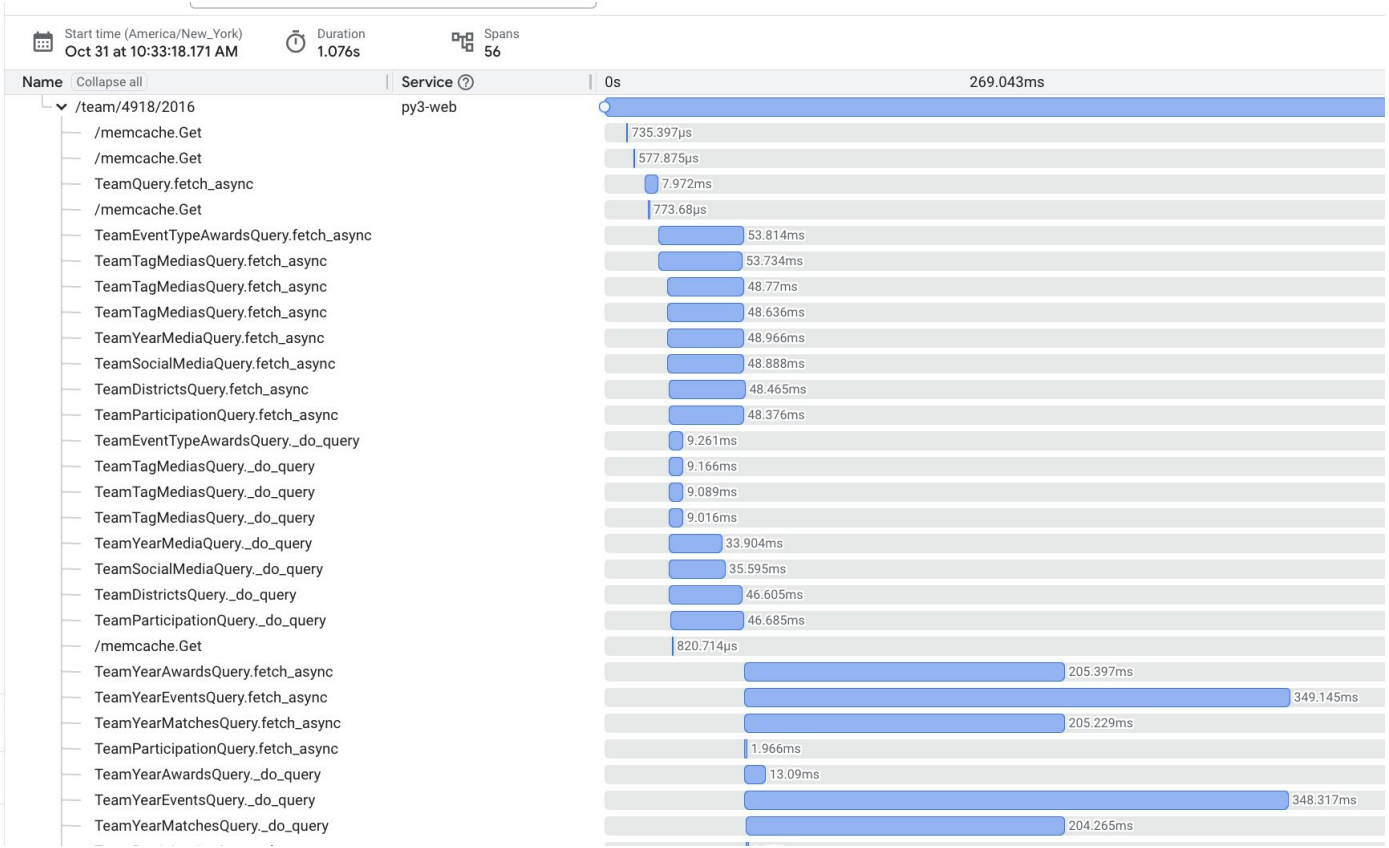




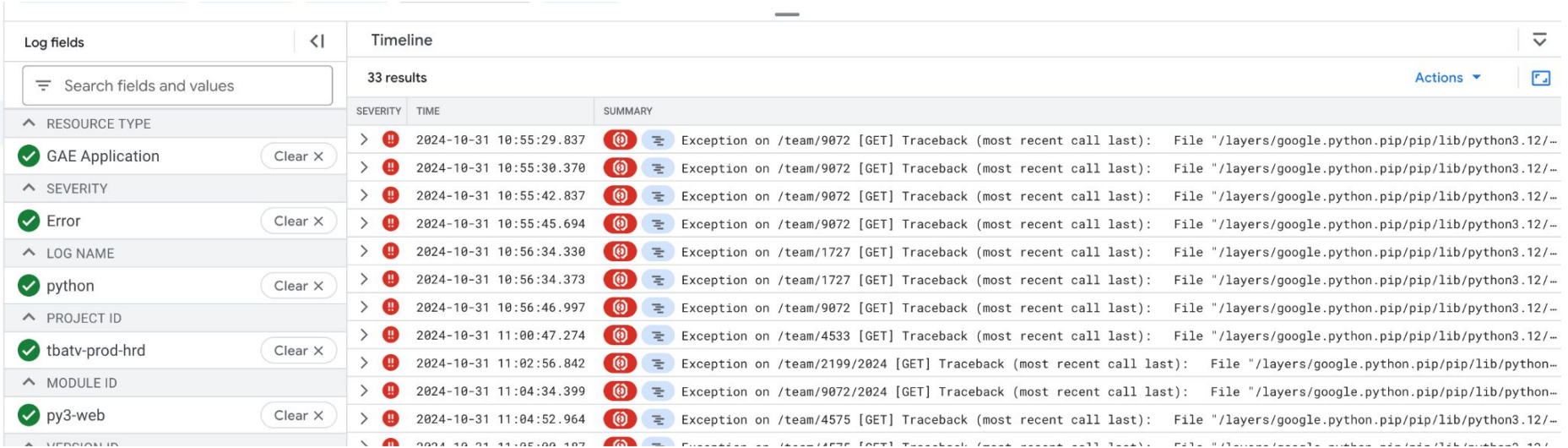
Example: Alerts



Example: Tracing



Example: Structured Logging



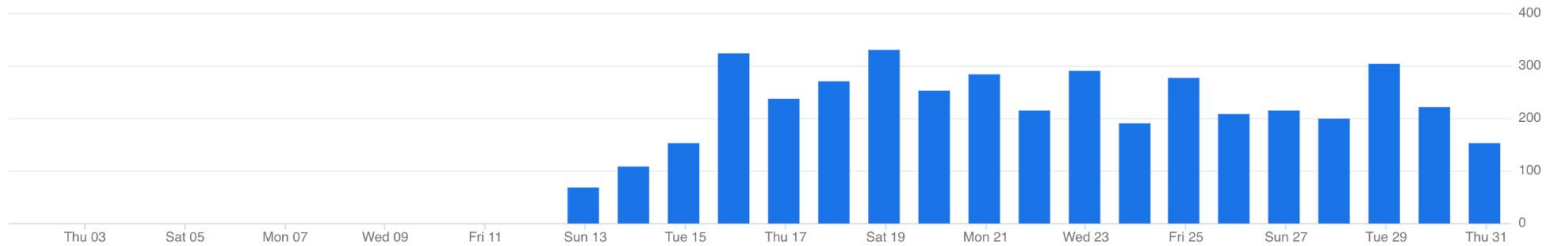
Example: Log Aggregation

ValueError: Bad CompLevel/set: qf 1

.get_double_elim_4_round (/workspace/backend/common/helpers/playoff_type_helper.py:201)

New | Troubleshooting suggestions

Resolution Status	Occurrences	Seen In	Response Code	Storage Location	First Seen	Last Seen
Open Link to issue	4,297	py3-web:1	0	global	Aug 21, 2024 Oct 12, 2024	23 minutes ago 23 minutes ago



Sample stack trace

Parsed | Raw

```
ValueError: Bad CompLevel/set: qf 1
    at .get_double_elim_4_round ( /workspace/backend/common/helpers/playoff_type_helper.py:201 )
    at .organized_double_elim_4_matches ( /workspace/backend/common/helpers/match_helper.py:133 )
    at ._build_playoff_info_double_elim ( /workspace/backend/common/helpers/event_team_status_helper.py:574 )
    at ._build_playoff_info ( /workspace/backend/common/helpers/event_team_status_helper.py:444 )
    at .render_team_details ( /workspace/backend/web/renderers/team_renderer.py:193 )
    at .team_detail ( /workspace/backend/web/handlers/team.py:42 )
    at ._call_fn ( /layers/google.python.pip/pip/lib/python3.12/site-packages/flask_caching/__init__.py:185 )
```

Tools & Resources

Open Source Tools

- Prometheus
- Grafana

Papers to Read

- Gorilla (counters): https://www.vldb.org/pvldb/vol8/p1816-teller.pdf
- Scribe (event logs): https://engineering.fb.com/2019/10/07/core-infra/scribe/
- Scuba (structured logging): https://research.fb.com/wp-content/uploads/2016/11/scuba-diving-into-data-at-facebook.pdf
- Canopy (tracing): https://research.facebook.com/publications/canopy-end-to-end-performance-tracing-at-scale/

Books to Read

- The SRE Book: https://sre.google/sre-book/table-of-contents/