Vax to K8s

Ticketmaster's Transformation to Cloud Native DevOps

ticketmaster®

HEATHER OSBORN

- Sr. Director Systems Engineering, Infrastructure
- Coming up on my 20th anniversary at Ticketmaster
- 25 years of being oncall drives the motivation to be better
    - More resiliency
    - More automation
    - Fewer calls
- Music and nightlife aficionado, and avid long distance runner

OUR JOURNEY

- Background
- Silos to DevOps
- Cloud Native Transformation
  - Why?
  - What?

Where do we go from here?

HISTORY

- Publicly Traded Company (LYV)
- $7.6B Revenue, $25B in GTV

- 1976 - Founded at Arizona State University
- 1996 - Ticketmaster.com launched
- 2010 - Live Nation and Ticketmaster join forces to power live experiences
- 2011 - Transformation journey begins...

ticketmaster®

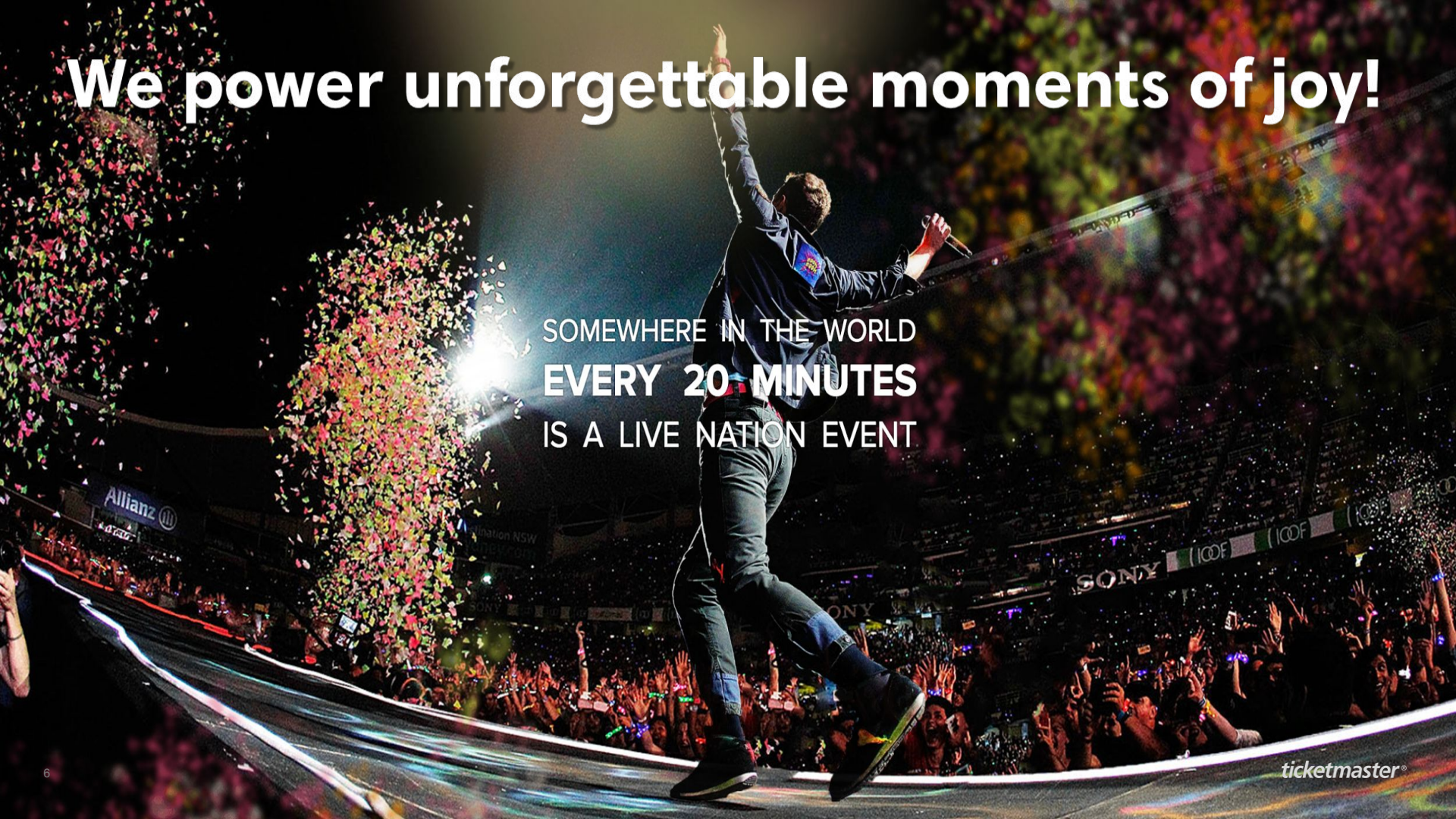
WHAT WE DO

Our platform powers:

- Over 26,000 Live Nation events to millions of fans worldwide
  - Sports, Concerts, Festivals, Theater, Arts, etc.
  - Over 53 Million fans in over 40 countries
- 484 Million Ticket transactions
- Own and Operate 167 Venues
- More than 1B unique visits to our web front
- 60% of our traffic is mobile (app + mobile web)

Ticketmaster is a Top 10 ecommerce site

We power unforgettable moments of joy!



BIG SCALE, BIG CHALLENGES

Onsales = Black Friday every day!

- Huge spikes / demand for tickets
- Global company = across time zones
- Limited inventory (Beyonce Tickets!)
- Multiple sales channels

TICKETMASTER TECH SCALE

(Every era of software, many not ready for containers and cloud)

- 21 Ticketing Systems and over 250 unique products
- 1,400+ people in Product & Tech org
- Custom Private Cloud with over 20,000 VMs across 7 global data centers
- Over 15,000+ network endpoints across the world (Venues, Arenas, Kiosks, etc)
- Over 60% VM growth in last year

SILOS!

- Operations was responsible for:
  - Software deployments
  - Monitoring
  - All alerts/escalations
  - All customer service inquiries
  - All business disruptions
  - The ire of angry clients when things didn’t go right!

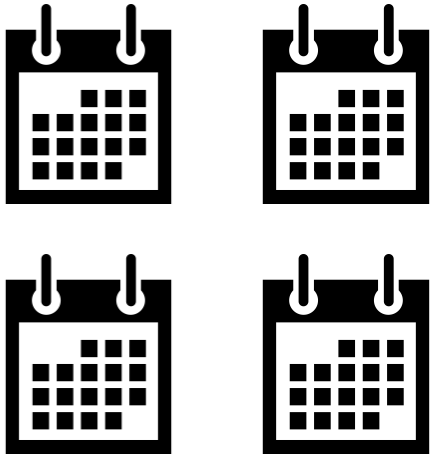
Consequently ops held everything VERY close to their chest.

SILOS!

- Development teams:
  - Responsible for time to market on features
  - Driven by product urgency
  - Accustomed to ‘throwing things over the wall’ with little guidance to ops on support

FRICTION

STEPS TO GET AN NEW APP DEPLOYED



1. Submit JIRA ticket to request new environments
2. Cloud team gets quotes and orders new hardware
3. Submit JIRA tickets to DC Ops to rack and cable new boxes
4. Follow up with Cloud Team would configure
5. Request ACLs via Network & InfoSec
6. Systems Engineering would have to build the RPM packages (depending on what software you needed)
7. To make the packages available, SysAdmins would deploy the packages to our home-grown configuration management system
8. Work with Application Support team to release software into the new environments
9. QA would test software
10. Submit JIRA ticket to get code released from QA to Prod

(if any issues are identified, repeat steps 8 & 9)

LET'S GET THIS DEVOPS PARTY STARTED!

- Give dev access to all of prod
  - Ops died a little inside

BUT!

- Dev was now responsible for doing their own releases and answering their own pages!
- Operations now had contact information for developers!
- Training for developers to handle operational tasks

ALMOST BUT NOT QUITE…

- Disparate monitoring
- No one could see the big picture
- Finger pointing among service teams
- No functional service catalog
- A bit of reluctance to take on unfamiliar work


CATASTROPHE

ONCE MORE, WITH FEELING!

- Development teams appoint fire chiefs
- Ownership of incidents, releases
- Operational teams embed with dev teams
- Tooling put in place to catalog our products and services (canonical source of info)
- Development teams now helped support onsales!

AUTONOMY!

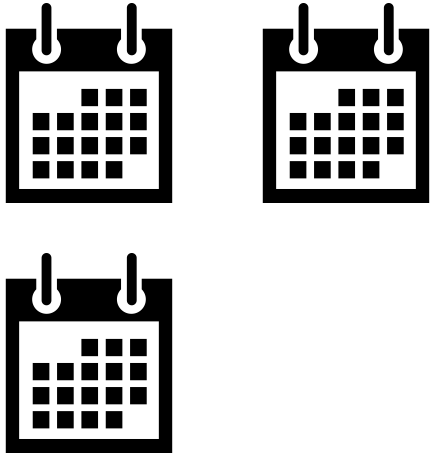
WHY TRANSFORM?

- Idle compute resources
  - Always scaled for that Black Friday but at what cost?

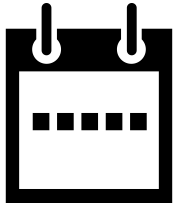
- Hardware upkeep costs
  - Infrastructure and security maintenance coordination nearly impossible
  - Shared resources at varying tech maturity and fault tolerance

SPEED TO DELIVERY: FEATURE RELEASES

PRE-DEVOPS



DEVOPS



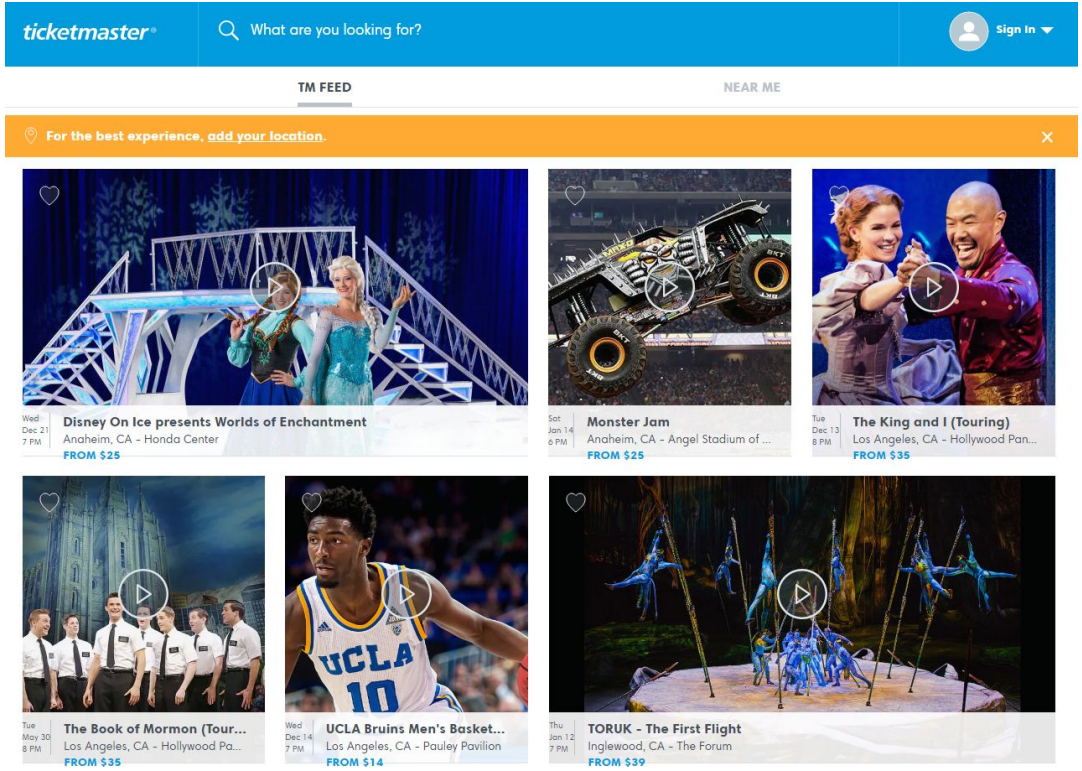
CLOUD NATIVE

WHAT DOES TRANSFORMATION MEAN TO US?

Modernize and mature tech stack

- Tech Maturity Model (https://github.com/Ticketmaster/techmaturity)
- Docker (build, ship, and run)
- Cloud Enablement (governance and guidelines)
- Leverage AWS (agility and dynamic scaling)
- Terraform(er) (desired state configuration)
- Kubernetes (bringing release times down to minutes)
- CoreOS partnership (let the makers make)

NEW TICKETMASTER WEB PLATFORM ON K8S



Before:

- Semi-manual stack creation, bespoke cloudformation + python boto scripts = 20+ mins to deploy
- Low Confidence

Now:

- K8S + Tectonic, fully automated = 60 second app updates
- High Confidence
- Unlocked Daily Delivery Culture

NEXT STEPS - HYBRID CLOUD

- K8s can add resilience while helping to assess and mature products into the next generation
- Leverage K8s to make the best use of resources - on-prem or public cloud
    - Public cloud isn't cheap - lift and shift is not cost effective
    - Some applications perform better on-prem - not moving our vax emulator into the public cloud anytime soon
    - Some legacy applications require significant redesign to move to public cloud - caching, NFS, failover, persistent storage
    - Allows us to define minimum viable standards for maintenance of an application

Conclusion

If a company whose technology and human infrastructure has grown up organically around a custom-written vax operating system can make the move to cloud native applications, so can you.

QUESTIONS?

heather@ticketmaster.com

@brokenheather

We're hiring!
jobs@ticketmaster.com

ticketmaster®