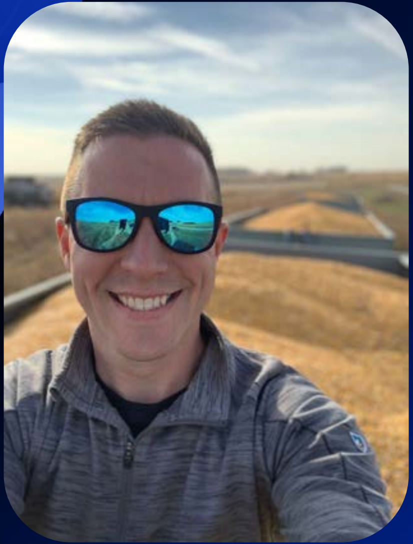
Revolutionizing Software Development

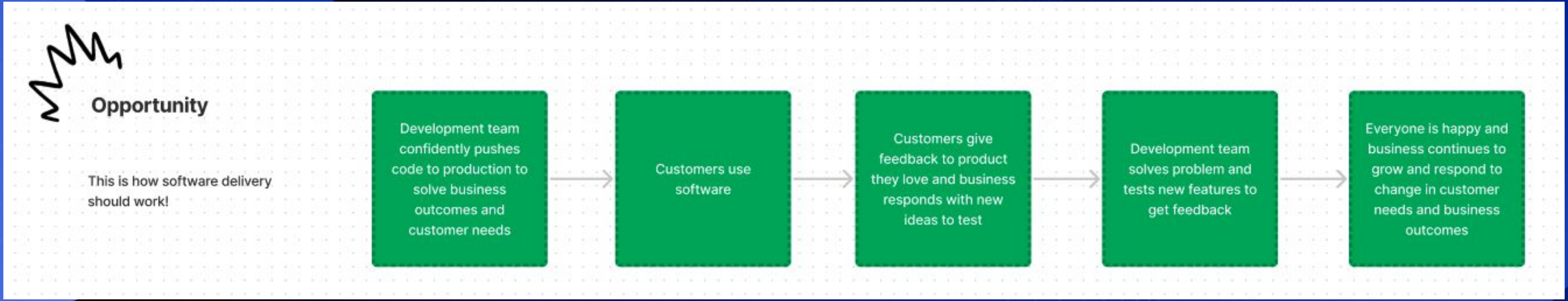
Integrating Chaos Engineering and Feature Flags for Enhanced Reliability and Agile Response



Matt Schillerstrom

Product Manager

harness



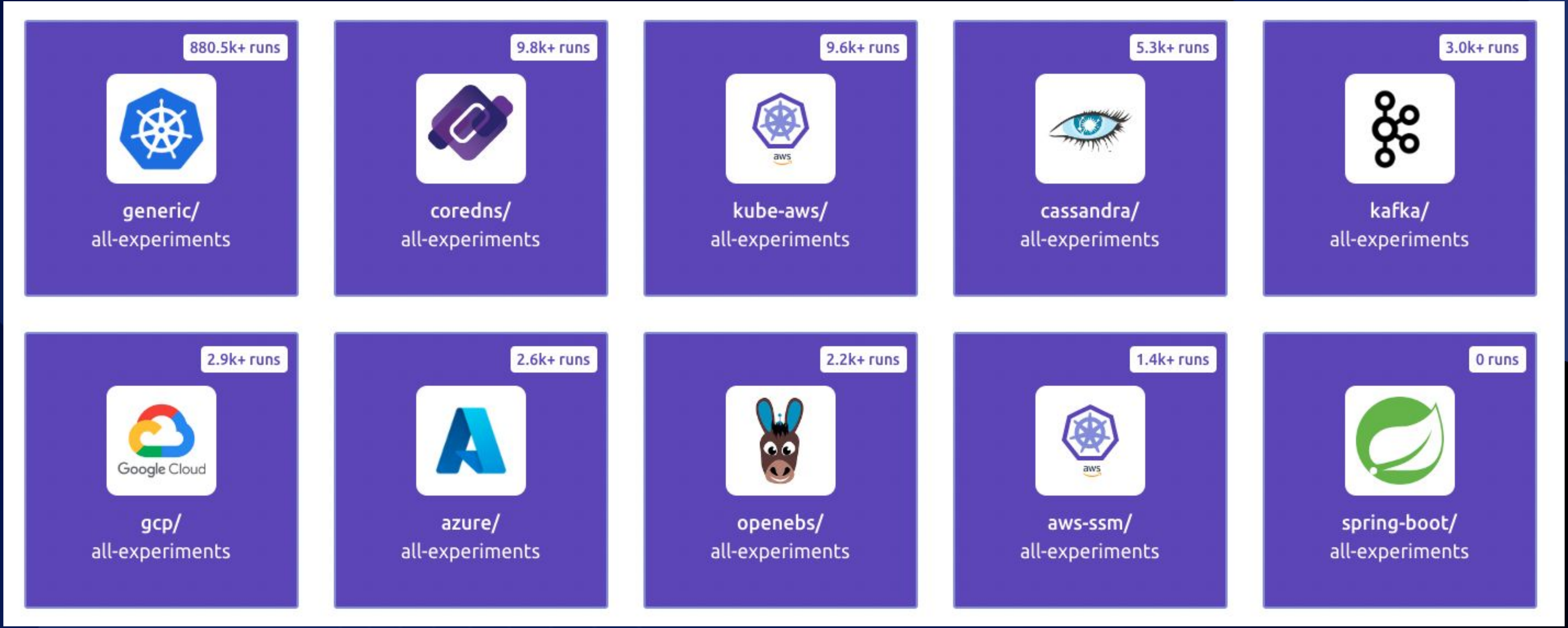
Opportunity

This is how software delivery should work!

Development team confidently pushes code to production to solve business outcomes and customer needs → Customers use software → Customers give feedback to product they love and business responds with new ideas to test → Development team solves problem and tests new features to get feedback → Everyone is happy and business continues to grow and respond to change in customer needs and business outcomes

Open Source Solutions

Litmus



880.5k+ runs
generic/
all-experiments

9.8k+ runs
coredns/
all-experiments

9.6k+ runs
kube-aws/
all-experiments

5.3k+ runs
cassandra/
all-experiments

3.0k+ runs
kafka/
all-experiments

2.9k+ runs
gcp/
all-experiments

2.6k+ runs
azure/
all-experiments

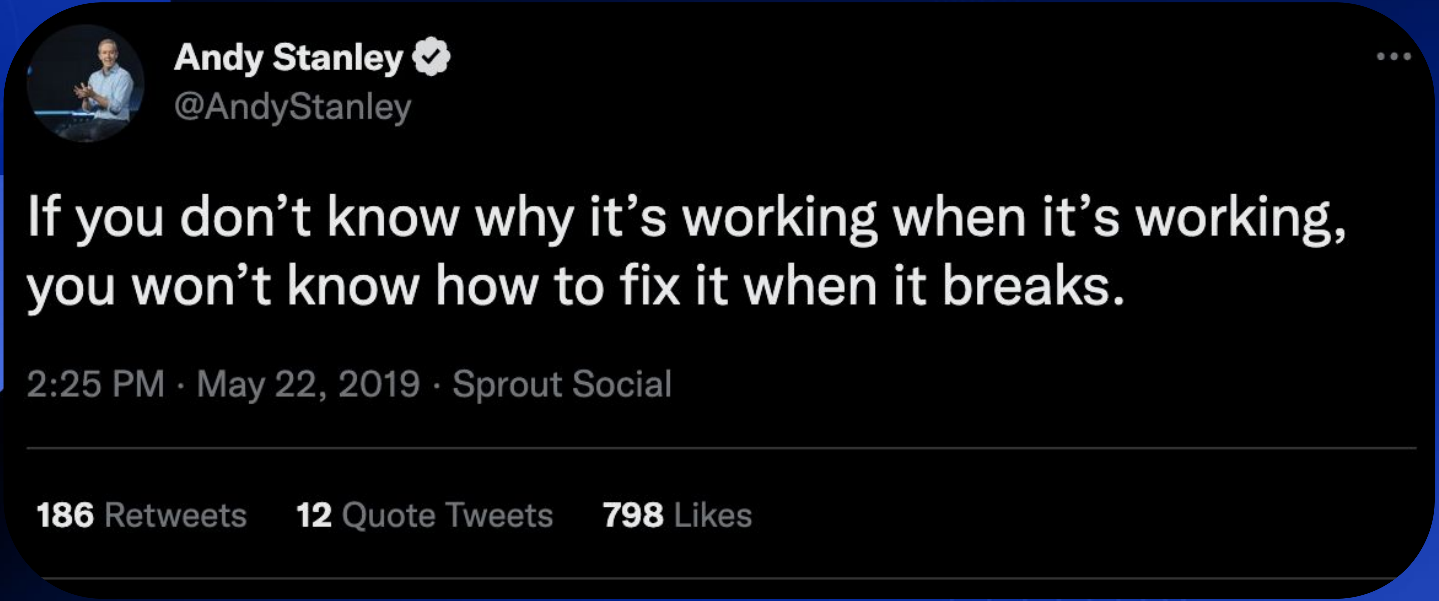
2.2k+ runs
openebs/
all-experiments

1.4k+ runs
aws-ssm/
all-experiments

0 runs
spring-boot/
all-experiments

OpenFeature

Standardizing Feature Flagging for Everyone



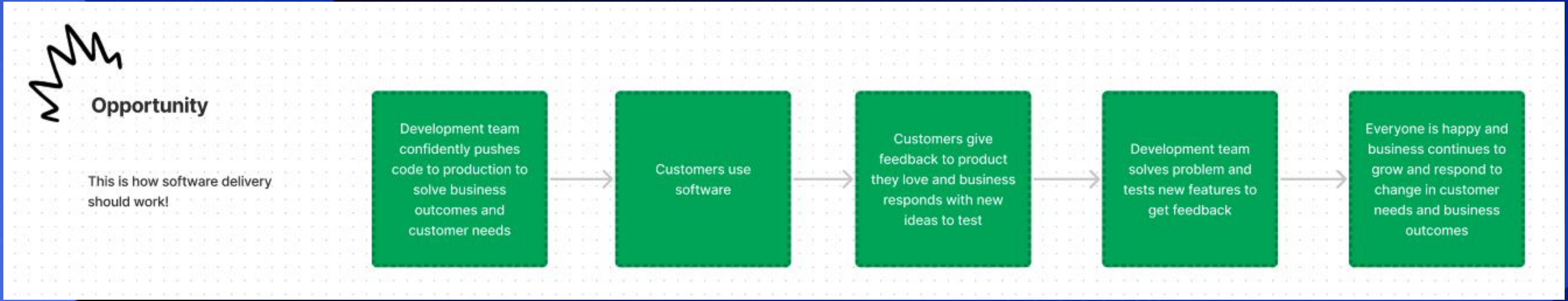
Andy Stanley
@AndyStanley

If you don’t know why it’s working when it’s working, you won’t know how to fix it when it breaks.

2:25 PM · May 22, 2019 · Sprout Social

186 Retweets **12** Quote Tweets **798** Likes

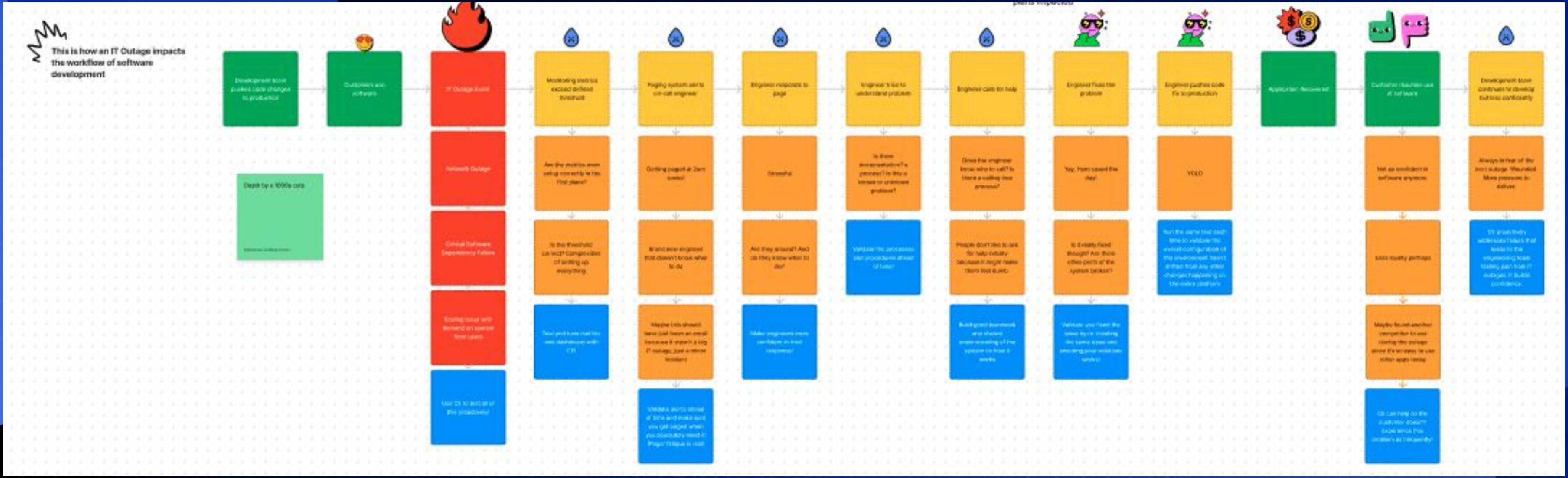
A recent exercise I took…

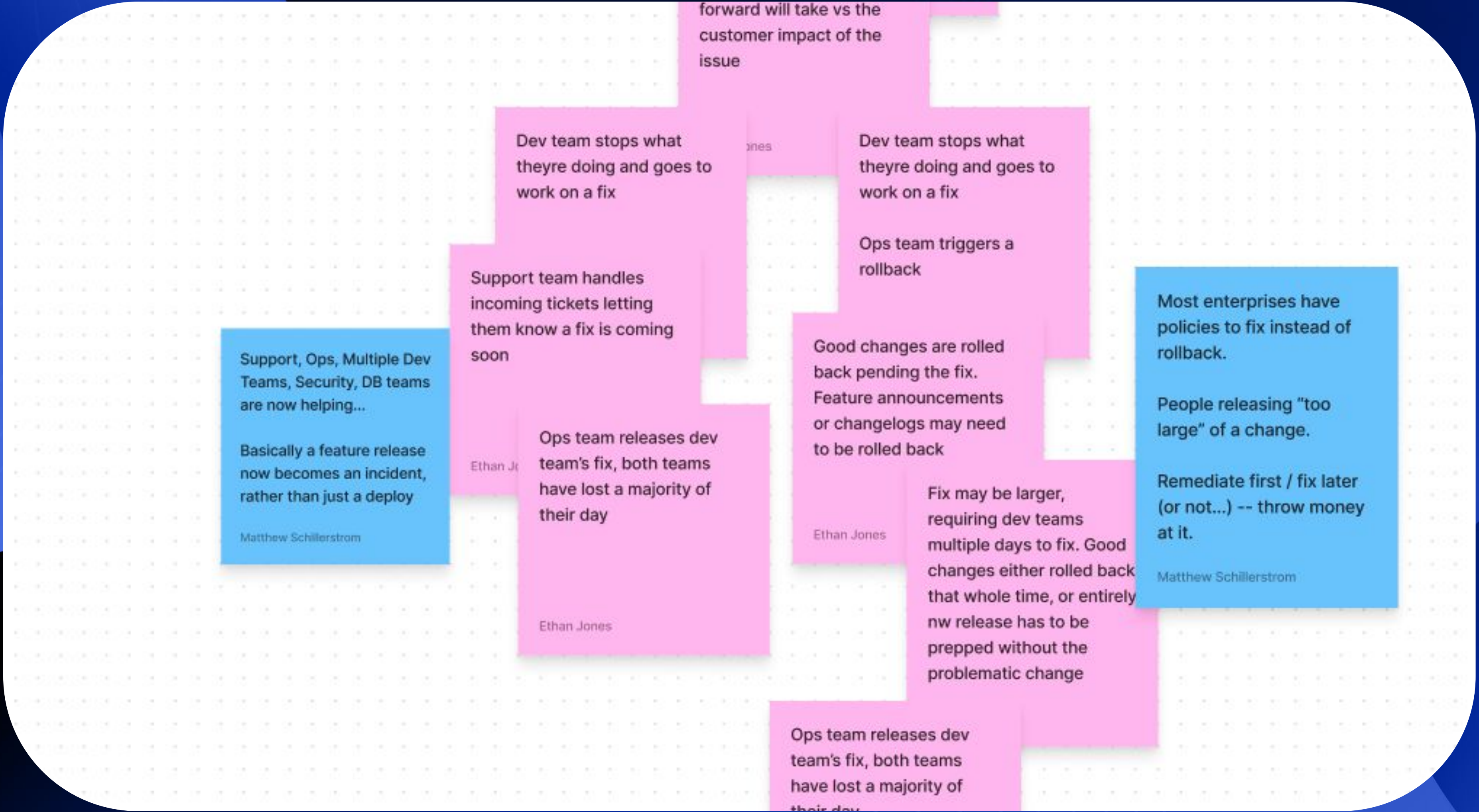


Opportunity

This is how software delivery should work!

Development team confidently pushes code to production to solve business outcomes and customer needs → Customers use software → Customers give feedback to product they love and business responds with new ideas to test → Development team solves problem and tests new features to get feedback → Everyone is happy and business continues to grow and respond to change in customer needs and business outcomes


plans impacted
This is how an IT Outage impacts the workflow of software development
Development team pushes code changes to production
Customers use software
Death by a 1000s cuts
IT Outage Event
Network Outage
Critical Software Dependency Failure
Scaling issue with demand on system from users
Use CE to test all of this proactively!
Monitoring metrics exceed defined threshold
Are the metrics even setup correctly in the first place?
Is the threshold correct? Complexities of setting up everything
Test and tune metrics and dashboards with CE
Paging system alerts on-call engineer
Getting paged at 2am sucks!
Brand new engineer that doesn't know what to do
Maybe this should have just been an email because it wasn't a big IT outage, just a minor incident
Validate alerts ahead of time and make sure you get paged when you absolutely need it! Pager fatigue is real!
Engineer responds to page
Stressful
Are they around? And do they know what to do?
Make engineers more confident in their responses!
Engineer tries to understand problem
Is there documentation? a process? Is this a known or unknown problem?
Validate the processes and procedures ahead of time!
Engineer calls for help
Does the engineer know who to call? Is there a calling tree process?
People don't like to ask for help initially because it might make them feel dumb
Build good teamwork and shared understanding of the system on how it works
Engineer fixes the problem
Yay, Hero saved the day!
Is it really fixed though? Are there other parts of the system broken?
Validate you fixed the issue by re-creating the same issue and ensuring your solution works!
Engineer pushes code fix to production
YOLO
Run the same test each time to validate the overall configuration of the environment hasn't drifted from any other changes happening on the entire platform
Application Recovered
Customer resumes use of software
Not as confident in software anymore
Less loyalty perhaps
Maybe found another competitor to use during the outage since it's so easy to use other apps today
CE can help so the customer doesn't experience this problem as frequently!
Development team continues to develop but less confidently
Always in fear of the next outage. Wounded. More pressure to deliver
CE proactively addresses issues that leads to the engineering team feeling pain from IT outages. It builds confidence.



forward will take vs the customer impact of the issue

…ones

Dev team stops what theyre doing and goes to work on a fix

Dev team stops what theyre doing and goes to work on a fix

Ops team triggers a rollback

Support team handles incoming tickets letting them know a fix is coming soon

Ethan J…

Support, Ops, Multiple Dev Teams, Security, DB teams are now helping...

Basically a feature release now becomes an incident, rather than just a deploy

Matthew Schillerstrom

Ops team releases dev team's fix, both teams have lost a majority of their day

Ethan Jones

Good changes are rolled back pending the fix. Feature announcements or changelogs may need to be rolled back

Ethan Jones

Fix may be larger, requiring dev teams multiple days to fix. Good changes either rolled back that whole time, or entirely nw release has to be prepped without the problematic change

Most enterprises have policies to fix instead of rollback.

People releasing "too large" of a change.

Remediate first / fix later (or not...) -- throw money at it.

Matthew Schillerstrom

Ops team releases dev team's fix, both teams have lost a majority of their day

Lean into the Business Impact

Cloud Cost

What impacts the bottom line?

Churn

Incidents

Being too reliable?

Training

Speed

Risk

Onboarding

$$$ Cost

"In 2023, **cloud infrastructure spending increased by 23%**, highlighting the need for effective cost management strategies within DevOps and SRE practices. This trend underlines the importance of efficient resource utilization in cloud services."

https://canalys-prod-public.s3.eu-west-1.amazonaws.com/static/press_release/2023/1725098181Worldwide-Cloud-Market-Q4-2022.pdf

$$$ Increasing More from GenAI Services

Worldwide end-user spending on public cloud services is forecast to grow **20.4% to total $678.8 billion in 2024**, up from $563.6 billion in 2023, according to the latest forecast from Gartner, Inc. “Cloud has become essentially indispensable,” said Sid Nag, Vice President Analyst at Gartner.

https://www.gartner.com/en/newsroom/press-releases/11-13-2023-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-679-billion-in-20240#:~:text=Worldwide%20end%2Duser%20spending%20on,Vice%20President%20Analyst%20at%20Gartner.

Velocity

Continuous Delivery Isn’t Good Enough

The Problem – CI/CD Ends at the Production Deployment

Risk in Large Deployments

- 1 bad feature = full rollback
- Deploy and release are the same
- Can't control feature access

No Control in Production

- Production issues affect all users
- Can't resolve issues while in prod
- Lack ability to isolate changes

Diminishing Returns on CI/CD

- Slow, cumbersome deployments
- Lack of production governance
- More tech debt and rework

Reduced Velocity

Big deployments + rollbacks = more rework, less features

Increased Risk

No control in production means the deployment must be perfect

Poor Developer Experience

Tightly coupled deploy and release add stress and toil

Reliability and Resilience

Common Kubernetes Failure Modes

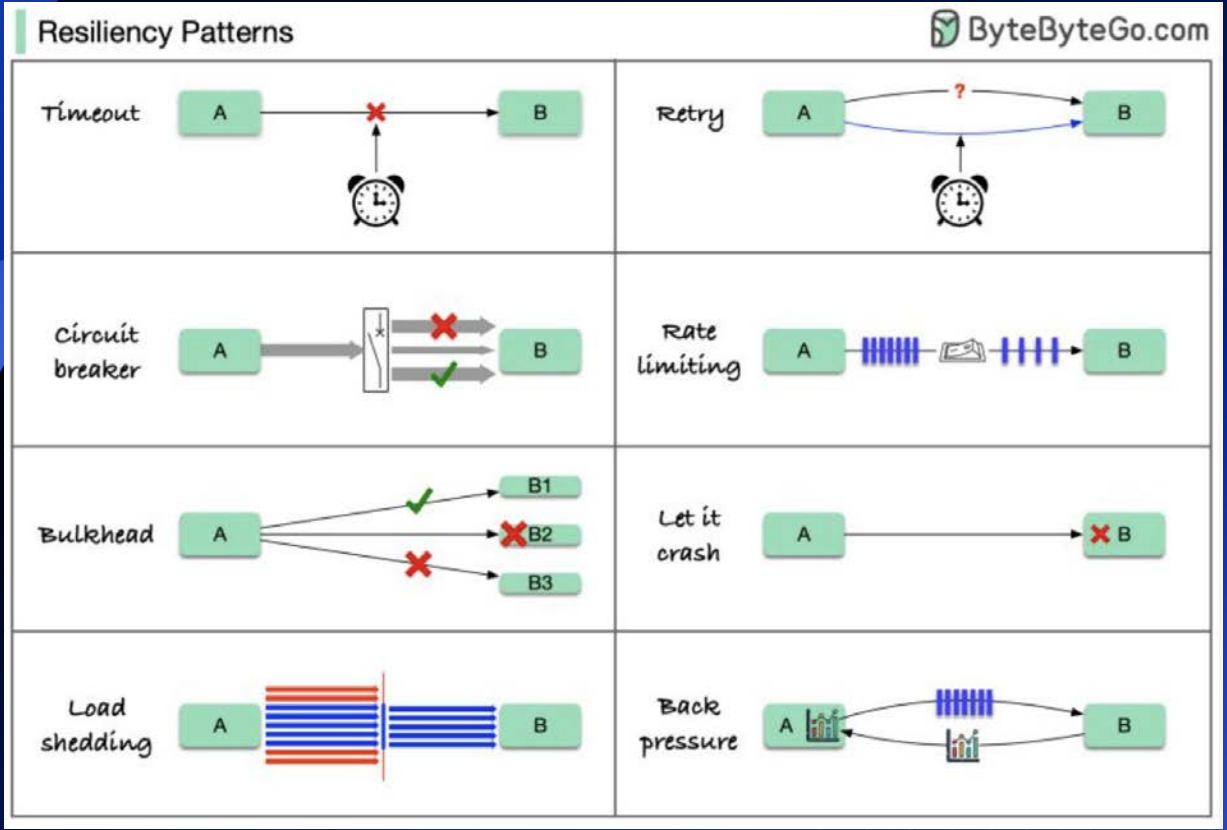
System Instability

Resource Exhaustion

Resource Contention

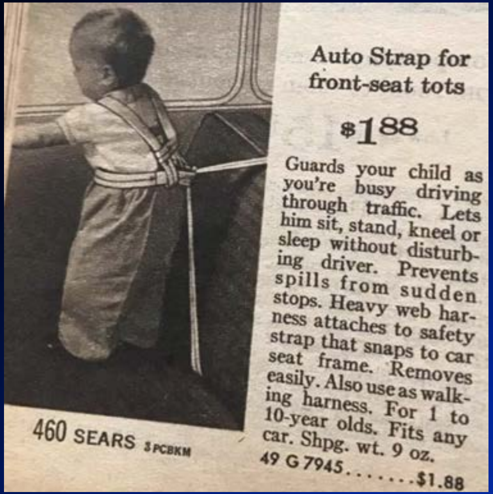
Configuration Errors

Scaling Issues


Resiliency Patterns
ByteByteGo.com
Timeout
A
B
Retry
?
A
B
Circuit breaker
A
B
Rate limiting
A
B
Bulkhead
A
B1
B2
B3
Let it crash
A
B
Load shedding
A
B
Back pressure
A
B

Technology and Standards Change

Child car seats have evolved. What seemed OK then is humorous now.



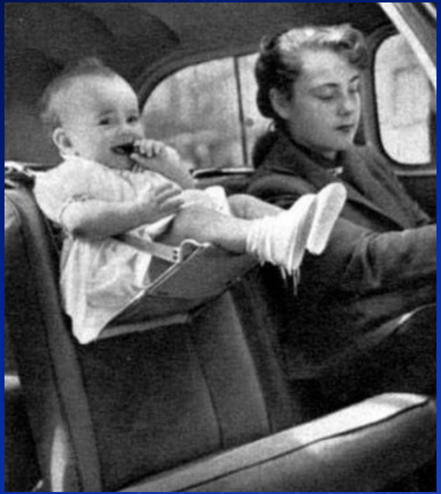
Auto Strap for front-seat tots

$1.88

Guards your child as you're busy driving through traffic. Lets him sit, stand, kneel or sleep without disturbing driver. Prevents spills from sudden stops. Heavy web harness attaches to safety strap that snaps to car seat frame. Removes easily. Also use as walking harness. For 1 to 10-year olds. Fits any car. Shpg. wt. 9 oz.

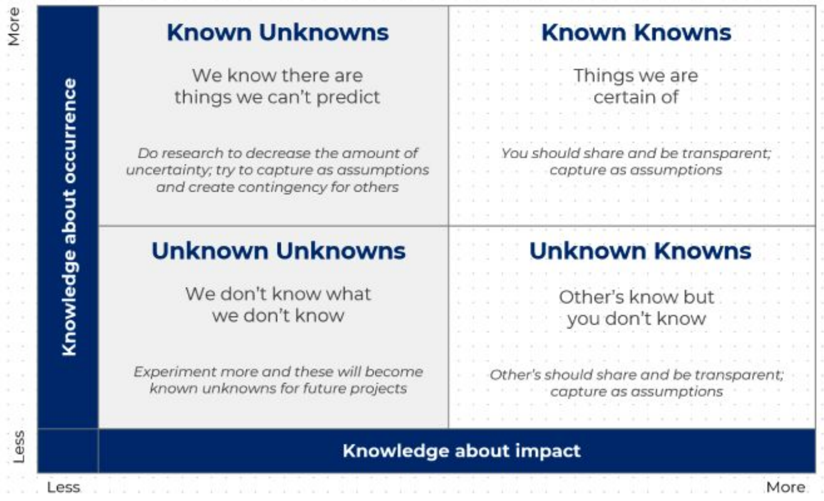
49 G 7945........$1.88

460 SEARS 3PCBKM



Failure Happens

Things break, systems degrade and fail



Knowledge about occurrence ↓ / Knowledge about impact →	Less	More
More	**Known Unknowns** We know there are things we can't predict *Do research to decrease the amount of uncertainty; try to capture as assumptions and create contingency for others*	**Known Knowns** Things we are certain of *You should share and be transparent; capture as assumptions*
Less	**Unknown Unknowns** We don't know what we don't know *Experiment more and these will become known unknowns for future projects*	**Unknown Knowns** Other's know but you don't know *Other's should share and be transparent; capture as assumptions*

Chaos Engineering can help you answer these questions:

What are my single points of failure?

Where does my system tip over?

What failures impact my customer experience?

What happens when a pod restarts?

What happens when my dependency has a slow response?

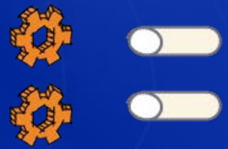
Does my system scale appropriately during peak traffic?

Did I configure my dashboard correctly?

Does my paging system work?

Integrating Chaos Engineering and Feature Flags…

Software Release Workflow



Step 1

Devs write code.
But, any changes are put behind a feature flag.

Value

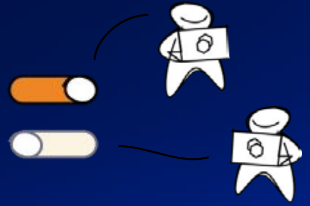
Deliver code on time
Test features for impact

Step 2

DevOps releases code to production. Test in production.

Value

Deploy on time every time
No change to process
Test safely



Step 3

Product Managers decide who gets the new feature, and who doesn't.

Value

Control release variations
Don't roll back - turn it off!

Step 4

Product and Dev decide what to iterate on. This is faster and more collaborative with FFs.

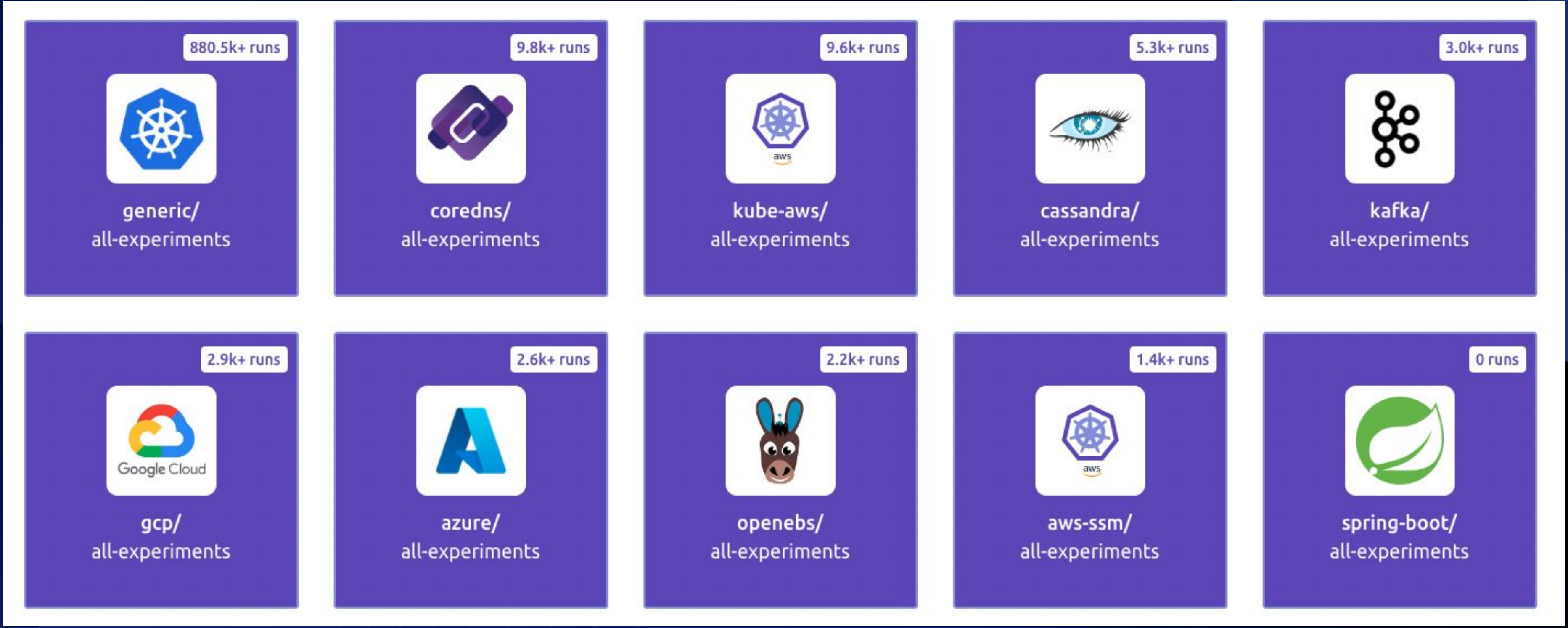
Value

Iterate on features faster
Higher feature quality

OpenFeature

Standardizing Feature Flagging for Everyone

Litmus



880.5k+ runs
generic/
all-experiments

9.8k+ runs
coredns/
all-experiments

9.6k+ runs
kube-aws/
all-experiments

5.3k+ runs
cassandra/
all-experiments

3.0k+ runs
kafka/
all-experiments

2.9k+ runs
gcp/
all-experiments

2.6k+ runs
azure/
all-experiments

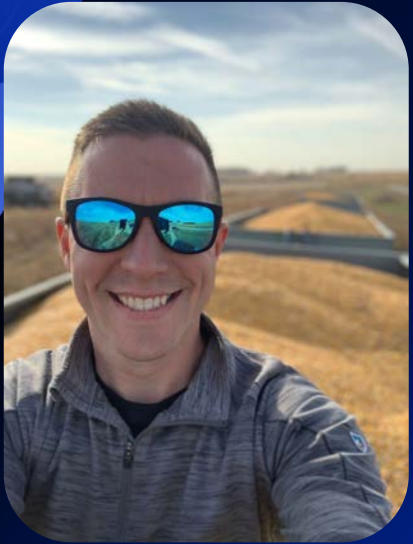
2.2k+ runs
openebs/
all-experiments

1.4k+ runs
aws-ssm/
all-experiments

0 runs
spring-boot/
all-experiments

Revolutionizing Software Development

Integrating Chaos Engineering and Feature Flags for Enhanced Reliability and Agile Response



Matt Schillerstrom

Product Manager

harness

https://www.linkedin.com/in/matt-schillerstrom-722b5559/