

**Reliability Matters**

**More Than Ever**

**“ATMs Spit Out Dollars  
as Customers Rush  
for Free Money”**





**Remember**

**your first time**

**on-call?**

**FAIL OVER  
CONF.**



# Tammy Butow

Principal SRE @ Gremlin

- ★ [twitter.com/tammybutow](https://twitter.com/tammybutow)
- ★ [linkedin.com/in/tammybutow](https://linkedin.com/in/tammybutow)
- ★ [tammybutow.com](https://tammybutow.com)
- ★ [twitch.tv/tammychaos](https://twitch.tv/tammychaos)
- ★ [tammy@gremlin.com](mailto:tammy@gremlin.com)



CHAOS ENGINEERING PLATFORM

Be prepared for the  
moments that matter

[gremlin.com/gremlin-free-software](https://gremlin.com/gremlin-free-software)

**How did you first work  
on reliability?**

# SOFTWARE ENGINEER



What my friends think I do



What my mom thinks I do



What society thinks I do



What my boss thinks I do



What I think I do



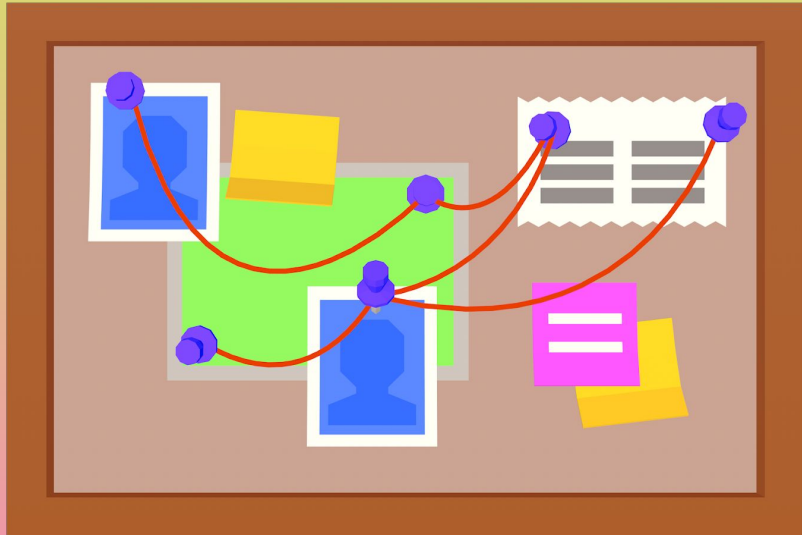
What I actually do



Being an SRE

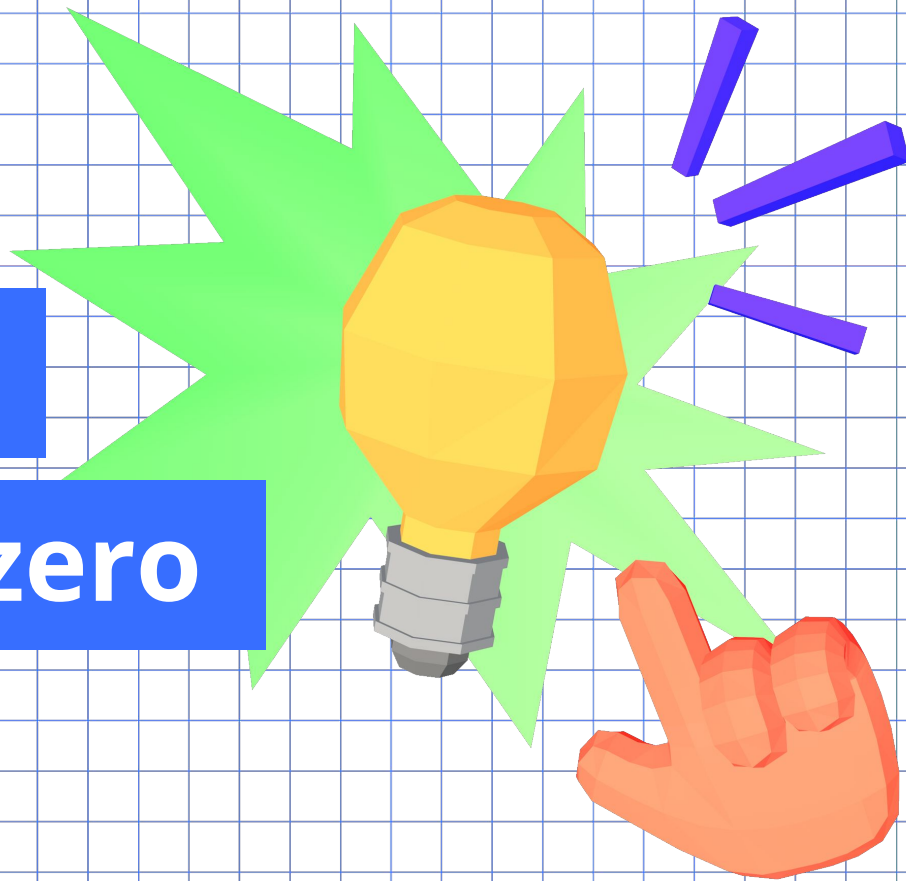
Is like being a

detective



**Reliability**

**is feature zero**



- ★ Create a map of people who are critical for reliability to improve
- ★ Tell everyone the critical issues we need to fix in the next 3 months
- ★ Prep **before** disaster recovery day **or** disaster strikes
- ★ Start small and gradually expand the blast radius

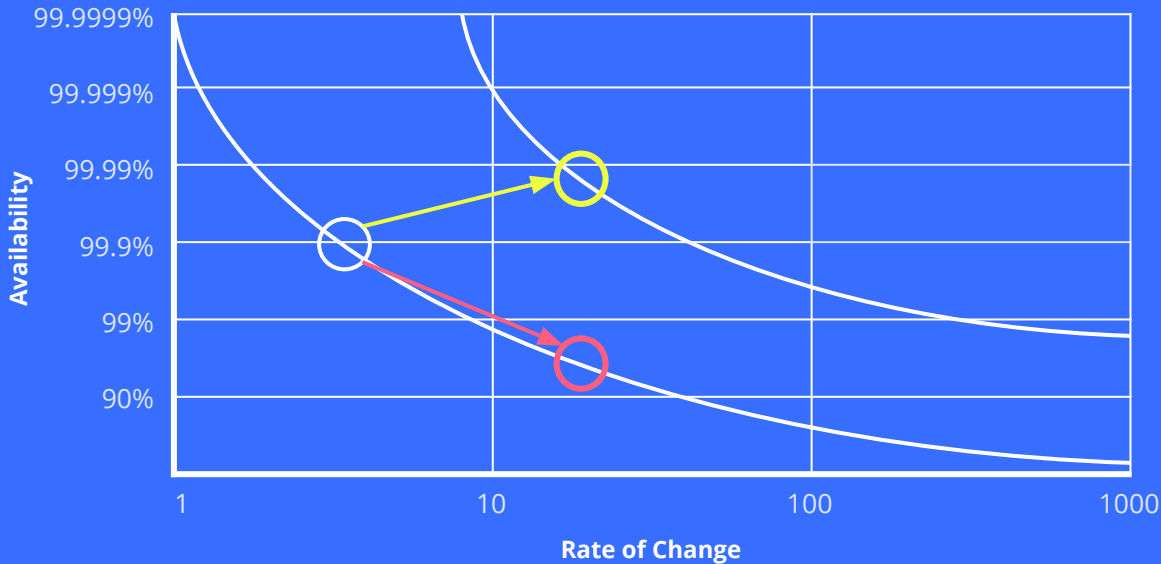
How to Start

Working Towards

Reliability

# The Journey to Chaos Engineering

# The Reliability Gap



**Availability  
vs.  
Rate of  
Change**

# A Brief History of Chaos Engineering

- 2010** ○ The Netflix Eng Tools team created Chaos Monkey.
- 2011** ○ The Simian Army was born.
- 2012** ○ Netflix shared the source code for Chaos Monkey on Github.
- 2014** ○ Bruce Wong (Netflix) created a new role: the Chaos Engineer.
- 2015** ○ Kolton creates FIT at Netflix
- 2017** ○ Kolton launches Gremlin



# My Chaos Engineering Journey



NAB was the first bank in the world to practice chaos engineering as defined by Netflix, we used Chaos Monkey from Netflix when it became open source. We were moving to AWS.



NAB (shutdown attacks)



DigitalOcean (fire drills)



Dropbox (process killer and shutdown attacks)



Gremlin (shutdown, cpu, memory, io, disk, process, blackhole, packet loss, latency....)

- ★ Remember that reliability is feature zero.
- ★ Identify your top 5 critical systems.
- ★ Gradually roll attacks out: shutdown, cpu, packet loss, latency...
- ★ Practice CE on your automation
- ★ Don't wait to see how your system handles failure, be proactive.

How to

Practice Chaos

Engineering



# Results

# And

# Milestones



Running 3x Chaos Engineering experiments a week



10x reduction in incidents in the three months

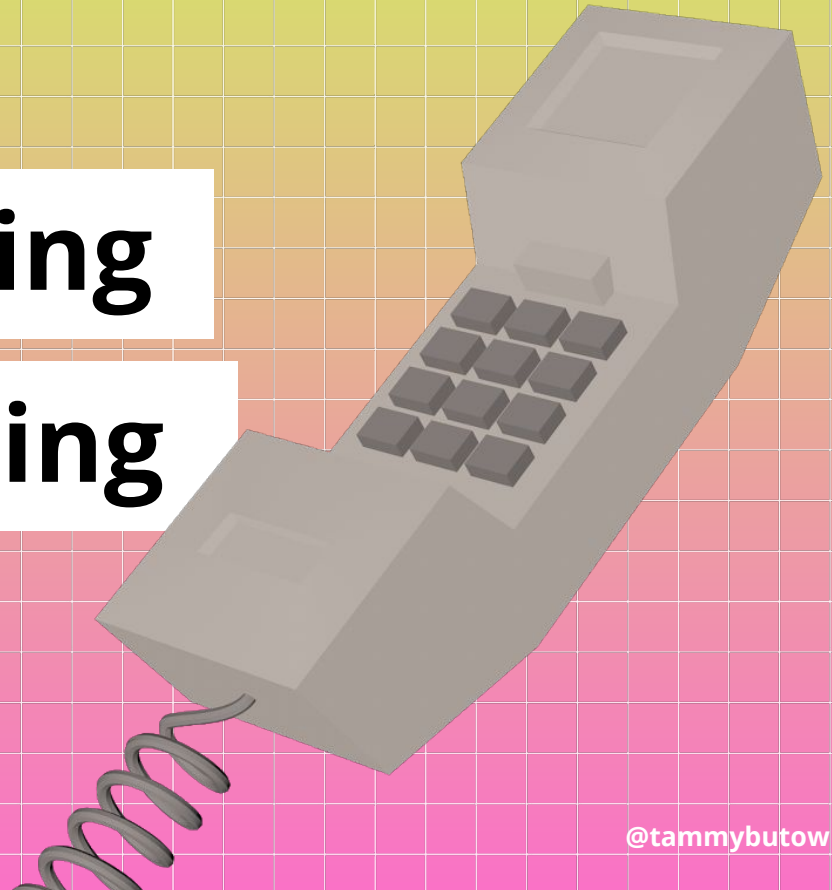


No SEV 0s managing DB for 500 mil customers with 500 PB data



Big wins using Chaos Engineering to improve database automation reliability with @M0sth8 (Slava Bakhmutov)

# Chaos Engineering for on-call training



Hipster Shop

Uh, oh!

Something has failed. Below are some details for debugging.

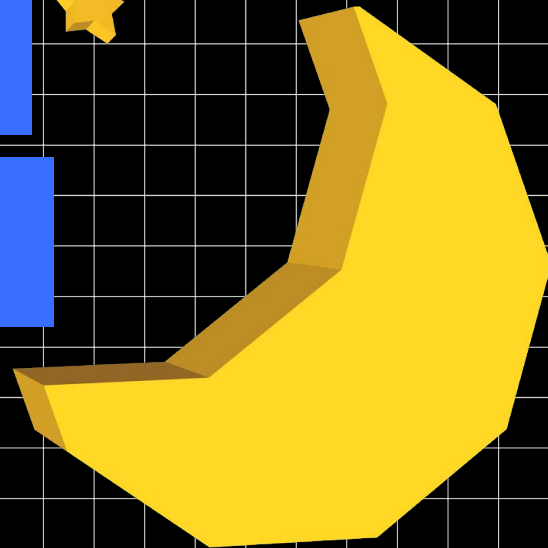
**HTTP Status:** 500 Internal Server Error

```
rpc error: code = Unavailable desc = all SubConns are in TransientFailure,
latest connection error: connection error: desc = "transport: Error while
dialing dial tcp 10.100.160.232:7070: i/o timeout"
could not retrieve cart
main.(*frontendServer).homeHandler
/go/src/github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/handlers.go:57
net/http.HandlerFunc.ServeHTTP
/usr/local/go/src/net/http/server.go:1995
github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/vendor/github.com/gorilla/mux.(*Router).ServeHTTP
/go/src/github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/vendor/github.com/gorilla/mux/mux.go:162
main.(*LogHandler).ServeHTTP
/go/src/github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/middleware.go:81
main.ensureSessionID.func1
/go/src/github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/middleware.go:103
net/http.HandlerFunc.ServeHTTP
/usr/local/go/src/net/http/server.go:1995
github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/vendor/go.opencensus.io/plugin/ochttp.(*Handler).ServeHTTP
/go/src/github.com/GoogleCloudPlatform/microservices-
demo/src/frontend/vendor/go.opencensus.io/plugin/ochttp/server.go:82
net/http.serverHandler.ServeHTTP
/usr/local/go/src/net/http/server.go:2774
net/http.(*conn).serve
/usr/local/go/src/net/http/server.go:1878
runtime.goexit
/usr/local/go/src/runtime/asm_amd64.s:1337
```



Engineers allowed

a full night's sleep.



# Thank You

**Find me on twitter @tammybutow**



**Jennifer Petoff**



**Julius Zerwick**



**Jeremy Daly**



**Matt Stratton**