Introduction to Chaos Engineering.

Canberra, Australia
February 5, 2018
@ GeoscienceAus

Tammy Butow
Principal SRE
Gremlin.com @Gremlininc
@tammybutow
Agenda.

2:00 - Welcome & Introduction to Chaos Engineering
2:40 - Q & A
2:55 - Thank you
Welcome

Hello I'm Tammy Butow, you can find me on twitter @tammybutow. I work at Gremlin, I'm an SRE.

I work remotely from Australia right now, our head office is in Silicon Valley.

Where else can you find me?
Twitter: twitter.com/tammybutow
Website: tammybutow.com
Where have I worked?

nab
DigitalOcean
Queensland University of Technology
WE ARE HACKERS, HUSTLERS AND HIPSTERS

Girl Geek Academy is a global movement encouraging women to learn technology, create startups and build more of the internet. Our face-to-face programs currently run in Australia and the USA.
Start-up brings girl geeks into technology sector

From left: Sarah Moran, Lisy Kane, April Staines, Tammy Butow and Amanda Watts, Girl Geek Academy founders. Photo: Josh Robenstone
Girl Geek Academy scored $1.3 million to encourage 1,000 women to establish tech startups

TONY YOO
JAN 31, 2017, 8:11 PM
Work Experiences:

- Infrastructure Engineering
- Building Tools
- Automation
- Incident Response
- Incident Management
- Observability & Monitoring
- Hardware Engineering
- Gamedays and Disaster Recovery Testing
- Team Leadership
- Security & Product Engineering

Work Locations:

- Sydney
- Brisbane
- Melbourne
- New York
- San Francisco
- … now remote!
Break things on purpose.

Downtime is expensive and damages customer trust. Gremlin’s Failure as a Service finds weaknesses in your system before they cause problems.

REQUEST A FREE TRIAL  JOIN OUR SLACK
About Gremlin

We’ve lived and breathed incidents, on-call, and Chaos Engineering for a decade. We’ve served as ‘Call Leaders’ at Amazon and Netflix, responsible for fixing global outages. We’ve employed Chaos Engineering to harden and prepare our services for internet scale. We’ve built this tooling before, and engineers loved it.

We hope you’ll love it too!
Kolton Andrus
CO-FOUNDER & CEO

Kolton is co-founder and CEO of Gremlin. Previously he was a Chaos Engineer at Netflix improving streaming reliability and operating the Edge services. He designed and built F.I.T., Netflix’s failure injection service. Prior he improved the performance and reliability of the Amazon Retail website. At both companies he has served as a ‘Call Leader’, managing the resolution of company-wide incidents. Kolton is passionate about building resilient systems, primarily as it lets him break things for fun and profit.
Matthew Fornaciari

CO-FOUNDER & CTO

Matt is co-founder and CTO of Gremlin Inc. He joined from Salesforce where he was a Senior Platform Engineer. Prior he improved the reliability and customer experience of the Amazon Retail website. He founded the 'Fatals' team to analyze and fix customer facing failures, reducing the number of Retail website errors by half in his first year. Matt loves building.

Twitter    LinkedIn
What is Chaos Engineering

A brief introduction to the practice of CE
What is Chaos Engineering

A brief introduction to the practice of CE

Chaos Engineering is an emerging discipline, but the underlying concepts are not. Failure is going to happen - Are you ready for it?

Put simply, Chaos Engineering is one approach to “breaking things on purpose” that teaches us new information about our systems through experimentation.

By triggering incidents intentionally in a controlled way, we gain confidence that our systems can deal with those failures before they occur in production. By practicing Chaos Engineering you’ll learn how to build systems and organizations that improve in the face of failure.
What is Chaos Engineering

A brief introduction to the practice of CE

The lesson we should learn and remember is that sooner or later, all complex systems will fail. It’s not a matter of if, it’s a matter of when. There will always be something that can — and will — go wrong.

Break Things on Purpose.

Building resilient systems requires experience with failure. Waiting for things to break in production is not an option. We should rather inject failures proactively in a controlled way to gain confidence that our production systems can withstand those failures.

By simulating potential errors in advance, we can verify that our systems behave as we expect — and to fix them if they don’t.
A Word of Caution

A brief introduction to the practice of CE

You should never conduct a chaos experiment in production if you already know that it will cause severe damage, possibly affecting customers — and with them, your reputation.

Always try to fix known problems first!

Chaos Engineering requires a base level of resilience.
The History of Chaos Engineering (in Australia!)

NAB deploys Chaos Monkey to kill servers 24/7

Engineers allowed full night's sleep.

The National Australia Bank has deployed the Netflix-developed 'Chaos Monkey' tool on a 24/7 basis to give its website development team some relief from needing to respond to server emergencies outside of work hours.

The application was developed by Netflix to constantly test the resiliency of its Amazon-based infrastructure, and randomly kill servers within its architecture to make sure it has the ability to compensate for the failure.

NAB migrated the public-facing areas of its nab.com.au website to the AWS public cloud in September last year.

Speaking at the Amazon Web Services Sydney summit today, the bank’s head of digital and online channel services, David Broeren, said the effort was aimed as much...
What is the state of Chaos Engineering right now?
The rise of chaos engineering
tcrn.ch/2EDbsvk by @DannyCrichton
How do you build reliable software? It is a question that has been at the top of my mind the past few weeks, as I seem to be increasingly confronted by software that just doesn’t work anymore. Bugs, crashes, errors, data leaks: they are so common in our every day lives that they can seem completely unremarkable.

The existing tools — unit tests, application performance monitoring, among many others — are useful to a degree, but they are clearly not the panacea to the problem. In response, there is a growing movement building around a new field known as “chaos engineering” that is designed to dramatically increase the quality and reliability of delivered services.

Last week, I had a conversation with one of the evangelists of the movement, Kolton Andrus. Andrus is the founder and CEO of a startup called Gremlin, which is building chaos engineering as a service. Formerly, he spent years working at Amazon and Netflix, where he implemented what have now been dubbed chaos engineering principles into those software teams.

The methodology of chaos engineering is simple in concept, but hard in execution. Software systems today are complex and tightly-coupled, meaning that the delivery of a webpage may actually rely on hundreds of database, file, image, and other requests in order to render. There has been a “combinatorial explosion” according to Andrus, particularly for engineering teams that have chosen a microservices architecture.
chaosengineering

310  # 24  3

Get Invite or sign in

%(http://www.gremlin.com)

Tech  Programming

Worldwide / English

Description
Chaos Engineering Community Slack, where we can discuss anything related to breaking stuff.

Channels
This team has 24 public channels in total.

# general
Topic: Channel to discuss all things Chaos Engineering
Which service teams should use Chaos Engineering?

Where should we focus first?

My top 3 recommendations for selecting services/systems:

1. Tier 0 / critical services - “what are your top 5 most critical systems?”
2. Services which serve critical functions, e.g. bushfire warning system
3. Services which store critical data, e.g. data storage/big data
What are the prerequisites for Chaos Engineering?

What do you need before you can get started?

My top 3 must-have recommendations for availability:

1. High Severity Incident (SEV) Management including SEV levels and definitions
2. Availability monitoring, including a high level health dashboard for WWW and API
3. Alerts and paging that call a human and wake them up for SEVs
Mini Bootcamp: Chaos Engineering + Docker

Be prepared for outages.
Mini Bootcamp Materials

A brief introduction to the practice of CE

We have the following:

1. A droplet from DigitalOcean (cloud infrastructure)
2. Docker (containers)
3. Weavenet Sock Shop (microservices app)
4. Gremlin (chaos engineering)
5. Datadog (monitoring)
Bootcamp Materials

A DigitalOcean Droplet

Cloud computing, designed for developers.

Create an Account

By signing up, you agree to the Terms of Service.
Bootcamp Materials

Docker and Docker Compose on your DigitalOcean Droplet

DOCKER PLATFORM ADDS KUBERNETES

Simplify and advance the management of Kubernetes for enterprise IT

LEARN MORE

SIGN UP FOR THE BETA
Bootcamp Materials

A demo Docker application, Cats vs Dogs voting app

Vote:
http://159.65.74.124:5000/

Results:
http://159.65.74.124:5001/

Processed by container ID
6844d7bdb076
Gremlin

Gremlin's Failure as a Service to find weaknesses in your system before they cause problems.

Break things on purpose.

Downtime is expensive and damages customer trust. Gremlin's Failure as a Service finds weaknesses in your system before they cause problems.

https://app.gremlin.com/dashboard
Datadog

Monitoring agent and dashboards for your application and containers

docker run -d --name dd-agent -v /var/run/docker.sock:/var/run/docker.sock:ro -v /proc:/host/proc:/ro -v /sys/fs/cgroup:/host/sys/fs/cgroup:ro -e API_KEY=faff9c88d8cdd357d76505f595f23797 -e SD_BACKEND=docker datadog/docker-dd-agent:latest
Let’s get started....

*Time for hands on Chaos Engineering*
Create an attack using Gremlin (UI or CLI)

https://app.gremlin.com/dashboard
Create an attack using Gremlin (UI or CLI)

```
$ gremlin attack help TYPE
attack: help: Invalid gremlin type.
```

Usage: gremlin attack TYPE [type-specific-options]

Type "gremlin help attack TYPE" for more details:

- blackhole  # An attack which drops all matching network traffic
- cpu        # An attack which consumes CPU resources
- io         # An attack which consumes IO resources
- latency    # An attack which adds latency to all matching network traffic
- memory     # An attack which consumes memory
- packet_loss # An attack which introduces packet loss to all matching network traffic
- shutdown   # An attack which forces the target to shutdown
- dns        # An attack which blocks access to DNS servers
- time_travel# An attack which changes the system time.
- disk       # An attack which consumes disk resources
- process_killer # An attack which kills the specified process

https://app.gremlin.com/dashboard
Create an attack using Gremlin (UI or CLI)

docker run -it \
>   --cap-add=NET_ADMIN \
>   -e GREMLIN_ORG_ID="${GREMLIN_ORG_ID}" \
>   -e GREMLIN_ORG_SECRET="${GREMLIN_ORG_SECRET}" \
>   -v /var/run/docker.sock:/var/run/docker.sock \
>   gremlin/gremlin attack-container 466bbb0e5246 cpu

https://app.gremlin.com/dashboard
atop

Monitoring from within the container you are attacking

docker run -d --name dd-agent -v /var/run/docker.sock:/var/run/docker.sock:ro -v /proc:/host/proc:/ro -v /sys/fs/cgroup:/host/sys/fs/cgroup:ro -e API_KEY=faff9c88d8cdd357d7650f595f23797 -e SD_BACKEND=docker datadog/docker-dd-agent:latest
Datadog

A sidecar will be created to perform the attack

<table>
<thead>
<tr>
<th>NAME</th>
<th>CPU %</th>
<th>RSS MEMORY</th>
<th>TX</th>
<th>RX</th>
<th>STATUS</th>
<th>START</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd-agent</td>
<td>0 %</td>
<td>130.39 MB</td>
<td>589 B</td>
<td>139 B</td>
<td>up</td>
<td>22 minutes ago</td>
</tr>
<tr>
<td>examplevotingapp_worker_1</td>
<td>1 %</td>
<td>83.83 MB</td>
<td>3.31 KB</td>
<td>2.07 KB</td>
<td>up</td>
<td>31 minutes ago</td>
</tr>
<tr>
<td>examplevotingapp_result_1</td>
<td>0 %</td>
<td>33.45 MB</td>
<td>194 B</td>
<td>157 B</td>
<td>up</td>
<td>31 minutes ago</td>
</tr>
<tr>
<td>examplevotingapp_vote_1</td>
<td>0 %</td>
<td>28.46 MB</td>
<td>0 B</td>
<td>0 B</td>
<td>up</td>
<td>31 minutes ago</td>
</tr>
<tr>
<td>gracious_tesla</td>
<td>0 %</td>
<td>11.18 MB</td>
<td>0 B</td>
<td>0 B</td>
<td>up</td>
<td>a minute ago</td>
</tr>
<tr>
<td>gremlin-466bbb0e5246</td>
<td>23 %</td>
<td>9.5 MB</td>
<td>3.14 KB</td>
<td>1.97 KB</td>
<td>up</td>
<td>a minute ago</td>
</tr>
<tr>
<td>redis</td>
<td>0 %</td>
<td>6.23 MB</td>
<td>675 B</td>
<td>1.49 KB</td>
<td>up</td>
<td>31 minutes ago</td>
</tr>
<tr>
<td>db</td>
<td>0 %</td>
<td>5.26 MB</td>
<td>1.57 KB</td>
<td>2.01 KB</td>
<td>up</td>
<td>31 minutes ago</td>
</tr>
</tbody>
</table>
How can you learn more about Chaos Engineering?

Useful resources and ways to learn

1. Chaos Engineering Community on Slack @ https://tinyurl.com/chaoseng
2. Follow Gremlin on Twitter @gremlininc
3. Technical Papers @ https://blog.gremlin.com/
4. Conferences (Qcon, Velocity and SREcon)
5. Follow Chaos Engineers on Twitter (@koltonandrus & @callmeforni)
Q & A

What's on your mind?
Thank You!

Thanks:

- James Kingsmill
- Everyone who attended today
- Geoscience Australia

Tammy Butow
Gremlin.com @Gremlininc
@tammybutow