How our security requirements turned us into accidental chaos engineers
We don’t make mistakes, we have happy accidents
- Bob Ross
Hello!

I’m Paul Carleton

@Stripe

(we’re hiring!)
Topics / Spoilers

1. Old Instances are bad
2. Enter Lifespan Management
3. Some stories of how things broke
4. What we learned
1. Instance Age

What is it and why do I care?
Terminology

- **Instance**: Cloud Hosted VM (EC2)
- **Age**: Time since launch
Instance Age

# of hosts

Just launched

@paulcarletonjr
Instance Age

Weeks later

# of hosts

Young

Old

@paulcarletonjr
Instance Age

# of hosts

Young

Old

Months later

@paulcarletonjr
Instance Age

# of hosts

Terminate & Replace

Old
Instance Age

# of hosts

Young

Old

@paulcarletonjr
Instance Age

New hadoop cluster
Instance Age

- Young hadoop cluster
- Big migration

# of hosts

@paulcarletonjr
What’s wrong with old instances?
What’s wrong with old instances?

Replacement is like a fire extinguisher...
Instance Age

# of hosts

Breaking changes

Last replacement

@paulcarletonjr
What’s wrong with old instances?

Replacement is like a fire extinguisher...

... that might catch on fire
What’s wrong with old instances?

Replacement is like a fire extinguisher...

... that might catch on fire
What’s wrong with old instances?
What’s wrong with old instances?

Will replacing a
with a work?
What’s wrong with old instances?
CVE Patch Released

@paulcarletonjr
CVE Patch Released

# of hosts

Young

Old

@paulcarletonjr
CVE Patch Released

# of hosts

Young

Old

@paulcarletonjr
CVE Patch Released

# of hosts

Young

Old

@paulcarletonjr
CVE Patch Released

# of hosts

Young

Old

@paulcarletonjr
Old Instances are bad
2. Lifespan Management
Components

▷ ASG

▷ Terminator

▷ Lifespan Manager:
What is an auto-scaling group?
What is a terminator?

1. Terminate
2. Terminator
3. Shave yaks
4. Shutdown
5. Wait
Lifespan Manager

@paulcarletonjr
Terminate First vs. Launch First

ASG Size

Time

Steady State

@paulcarletonjr
Terminate First vs. Launch First

ASG Size

Steady State

Time
Now what?
Rollout Plan

Breaking the problem up with labels
Breaking it up with labels

Stateless
Safe to replace

Stateful Automated
Replaceable with some graceful state hand-off.

Requires Operator
Not safe to replace automatically. Want someone watching
Automated termination
What could go wrong?
A Year Long Journey

@paulcarletonjr
5 Chaotic Discoveries
3.1 How NOT to health check
The lifespan manager terminated all the LDAP servers.
We’re locked out of QA.
Don’t we check for this?
How did this happen?

What’s your health?
How did this happen?

What’s your health?

 LDAP
 Maintenance

@paulcarletonjr
How did this happen?

What’s your health?

- LDAP
- Maintenance

- Maintenance
How did this happen?

What’s your health?

Everything’s green! Let’s terminate!

LDAP
Maintenance

Maintenance
0 unhealthy != healthy
How NOT to health check

▷ Pick good defaults
▷ Use pre-shared knowledge to verify health
Explicit Expectations

What’s your LDAP health?

- LDAP
- Maintenance

- Maintenance
Explicit Expectations

What’s your LDAP health?

No LDAP? I better wait.
3.2
RIP Kubernetes Workers
The Kubernetes workers are going down... HARD!
Terminator Recap

1. Terminate
2. Terminator
3. Shave yaks
4. Shutdown

@paulcarletonjr
Terminator Recap

1. Terminate
2. AWS
3. Shut down
4. Wait

@paulcarletonjr
Terminate

Terminate
Terminate

Terminate
Terminate

Terminate
RIP Kubernetes Workers

- Track feature usage
- Make the chaos easy to turn off
3.3
Blackhole Scenario
Terminator Recap

1. Terminate

2. AWS

3. Wait

4. Shutdown

Terminator

Shave yaks
Heartbeat Options

- Delay termination
- Proceed with termination
- ... but no Cancel
Blackhole Scenario

1

Terminate
Blackhole Scenario

- Non-zero exit
- Timeouts
- Rate limits
The terminations will continue until morale improves!
Two Touch Termination

1. Terminate
2. AWS
3. Terminate
4. Shutdown

Terminator
Shave yaks

Already done!

@paulcarletonjr
The Blackhole Scenario

- Align incentives
- Systems vary, so adapt to match!
3.4 Self-Service Meltdown
A Year Long Journey
A Year Long Journey
Enhance!
Enhance!

@paulcarletonjr
If you would like to never think about a kernel upgrade again, consider Lifespan Management!
Part 1:
Turning it On
I want to enable lifespan management!
Great! Here are some docs!
Part 2: Who does what?
What happens during termination?
Let me tell you!
yadda yadda
yadda yadda
yadda yadda
yadda yadda
yadda yadda
yadda yadda
yadda yadda
yadda yadda
yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda yadda
What part of that is relevant to me?
... Great question!
Part 3: False Alarms
Did lifespan management just break my thing?
Let me check!
5 minutes later...
Nope!
Okay... what did break my thing?
Okay... what did break my thing?
How we changed
Part 1: Turning it On

@paulcarletonjr
Part 2: Who does what?
Part 3: False Alarm

@paulcarletonjr

go/whydead/$instance_id
Self-service Meltdown

▶ Make it easy to adopt safely
▶ Explicitly state the contract
▶ Make it easy to rule chaos out
3.5
Death by a thousand JIRA tickets
Something’s wrong, I can’t terminate anything

These warnings should be tickets!
Great!
Death by a thousand JIRA tickets

- File against ourselves first, then automate
- 1% case matters more with 10x terminations
- Measure Quantity and Reliability of tickets
4. Calling it Done

The End... for now
5. Summary and Closing
Takeaway

- What automation problems can you solve with a little chaos?
Takeaway

Do you know how old your instances are?
Thank you!
Credits

- Photo by rawpixel on Unsplash
- Photo by Jens Lelie on Unsplash