A Tale of Two Oncalls

Building a Humane & Effective Oncall

Nick Lee
Where I come from

Started
Backend Engineering

Volunteered
for more oncall

Went Oncall

Moved to
Production Engineering
Defining Oncall
ring0
Protects the core customer experience
Vendor Integrations

Builds & supports interactions with third parties

https://topatoco.com/products/kcg-tfdog-plush
Two Things Matter
Two Things Matter

WORLD-WIDE DELIVERY IN 30 MINUTES OR LESS

OR YOUR NEXT ONE IS FREE

Attribution: Marco Verch https://creativecommons.org/licenses/by/2.0/
ring0

✅ - Effective
✅ - Humane
Vendor Integrations
⚠ - Effective
❌ - Humane
https://topatoco.com/products/kcg-fdog-plush
Three Nines of Uptime

44 minutes of downtime
Three Nines of Uptime

hours

of firefighting
Looking elsewhere for inspiration
Triage outages very aggressively
Constantly refine your alerting
The response to an alert should never be ambiguous.
Those best practices didn’t help
The search for the secret sauce begins

What did ring0 have that we didn’t?

Attribution:
https://www.youririshop.com/YR-sauce-Squeezy-485g
The Mitigation Toolbox
Stabilize

Diagnose

Treat

Attribution:
https://commons.wikimedia.org/wiki/File:Siemens_Magnetom_Aera_MRI_scanner.jpg
https://creativecommons.org/licenses/by-sa/4.0/
But aren’t tools expensive?
Quantify your oncall

01 Cost per minute of outage
02 Broken Nights per week
03 Incident Counts
A messy mitigation is better than no mitigation
<table>
<thead>
<tr>
<th>Destination</th>
<th>Platform/Voie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mannheim-Friedrich</td>
<td>11</td>
</tr>
<tr>
<td>Gernsheim</td>
<td>17</td>
</tr>
<tr>
<td>Köln Hbf</td>
<td>7</td>
</tr>
<tr>
<td>Berlin Hbf</td>
<td>9</td>
</tr>
<tr>
<td>Passau Hbf</td>
<td>6</td>
</tr>
<tr>
<td>Siegen</td>
<td>16</td>
</tr>
<tr>
<td>Saarbrücken Hbf</td>
<td>20</td>
</tr>
<tr>
<td>Fulda</td>
<td>8</td>
</tr>
<tr>
<td>Bruxelles-Midi</td>
<td>19</td>
</tr>
<tr>
<td>Hanau Hbf</td>
<td>5</td>
</tr>
</tbody>
</table>

Train is cancelled
Find ways of making your outage irrelevant to the user
What now?
Mitigation Runbook

Fifteen easy steps:

1. Wake up
2. Scream and run in circles
3. Stop all deployments
4. Tell everyone not to make any changes
5. Check that the other datacenters are healthy
6. Disable simulated traffic
7. Update the load balancers
8. Get a code review for that
9. Deploy the config change
10. Update your host files
11. Get a code review for that
12. Deploy the config change
13. Update your DNS entries
14. Realize this is too long
Make your tools an extension of human decision making
run lockdown/all
run failover fra1
Non-SREs who trust the tools:
Make your oncall feel as safe as possible
Decision Frameworks

- Can we mitigate this?
- Are lots of users affected?
- What are the side effects?
Opinionated Frameworks

Can we mitigate this?

Are lots of users affected?
Non-SREs who trust the tools:

Existing Team Members
Make the tool feel safe
Make every action deliberate

Enter 9000 to execute or n to abort: 1234
Invalid input. Enter 1713 to execute or n to abort: 0000
Enter 2777 to execute or n to abort: 2777

2019-09-03T12:56Z [I] (PIN_CHECK)
failover operator lock is unlocked
Log all mitigation actions
% toolbox run

Usage: toolbox run COMMAND [arg...]  
Run a mitigation plan  

Commands:

  undrain/time       Time-based revert of failover actions.  
  undrain/uuid       UUID based undrain for failover actions
Non-SREs who trust the tools:

Existing Team Members  New Team Members
Make the tool safe
Non-SREs who trust the tools:

Existing Team Members  New Team Members  Product Owners
Our Oncall Today

Things break, alerts fire

Take appropriate action

Monitor recovery

Undo the mitigating action
Our Old Oncall

Things break, alerts fire

Escalate to the vendor

Pull logs to help speed up response

Keep waiting...
Fixing oncall with mitigation tools

Nick Lee