Go2Group
Rasmus Praestholm
Sr. DevOps Architect

Demo code and dev workspace at:

- https://github.com/Cervator/modern-jenkins
- https://github.com/Cervator/GitOpsUtilityJobs/tree/opsgenie
- https://github.com/Cervator/HoboHealthCheck

Go2Group

Target Audience:

- Anybody wanting to improve their handling of incidents
- Particularly complex enterprises with diverse teams

The problem

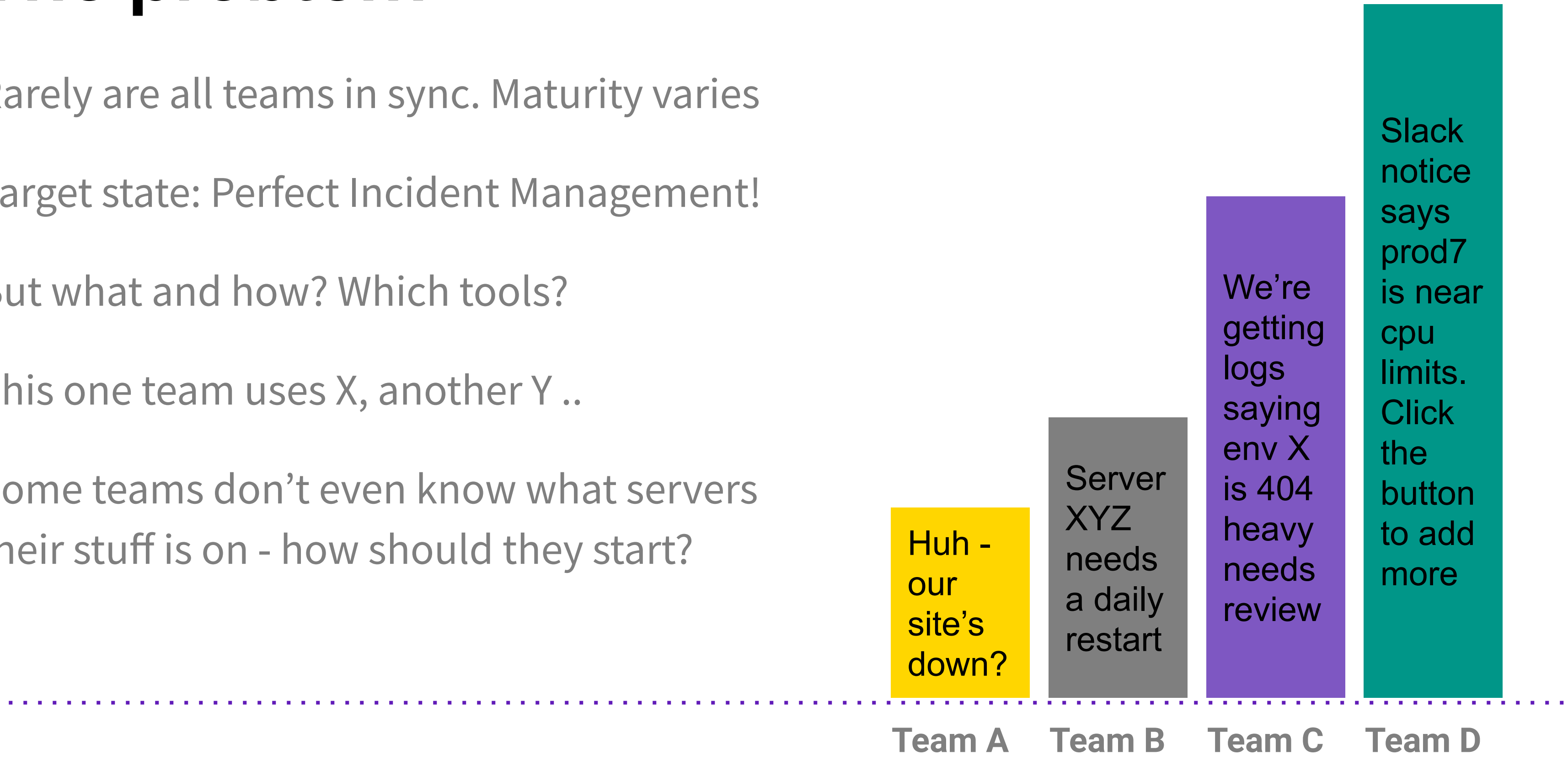
Rarely are all teams in sync. Maturity varies

Target state: Perfect Incident Management!

But what and how? Which tools?

This one team uses X, another Y ..

Some teams don't even know what servers their stuff is on - how should they start?

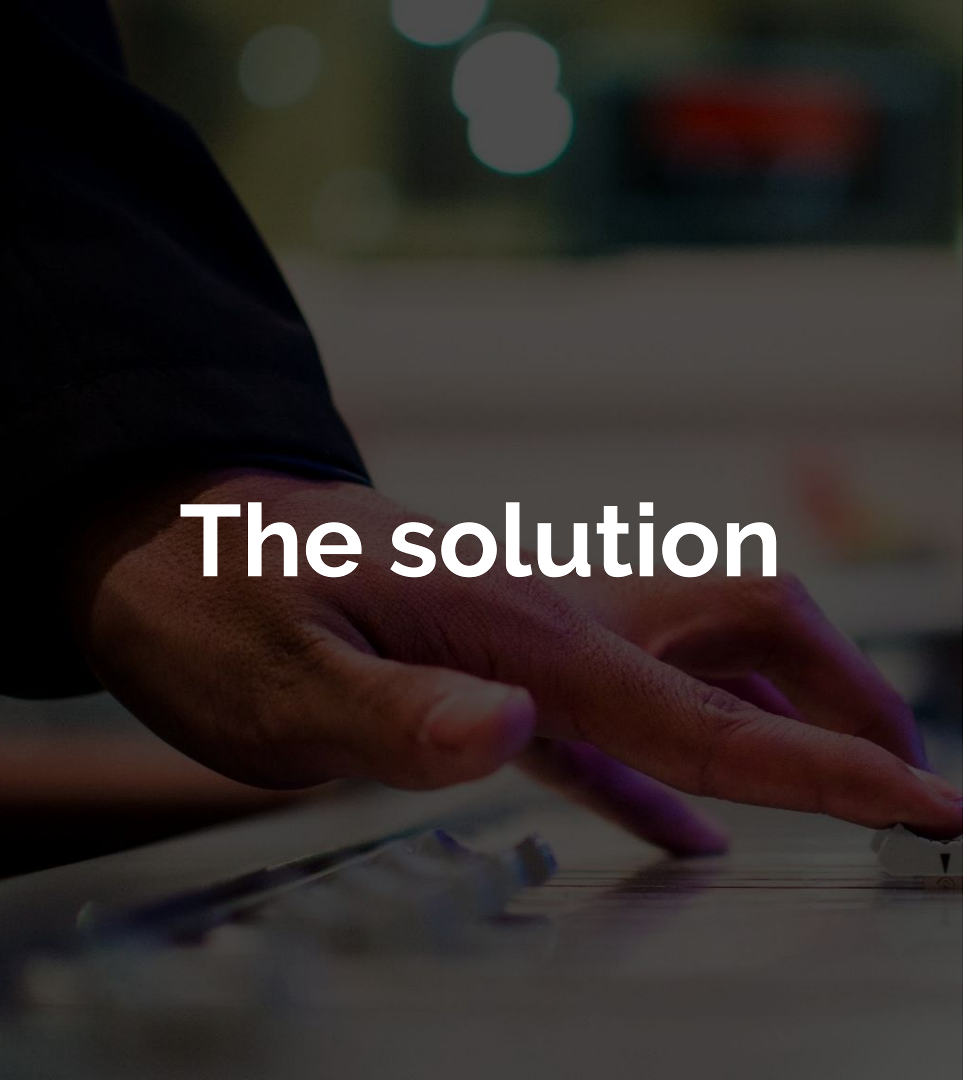


Team A: Huh - our site's down?

Team B: Server XYZ needs a daily restart

Team C: We're getting logs saying env X is 404 heavy needs review

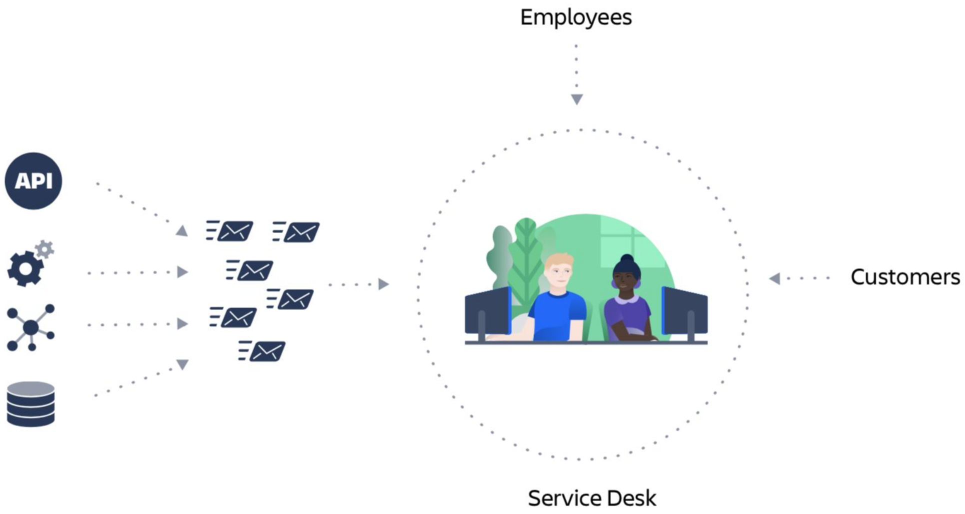
Team D: Slack notice says prod7 is near cpu limits. Click the button to add more



The solution

- **Basic**: Grab an incident management tool. Any tool. Use it!
- **Complex**: Grab a good tool, make it the default option in a toolchain
- **Assist** your teams in managing their own - more on that later

Background: Incident Management



- **As per OpsGenie anyway** - gotta pick somebody's definitions!
- **Show of hands**: who considers themselves well-versed in the principles of incident management?
- **Show of hands**: who actually has well-integrated tools in place, at least for some apps?

What are we trying to do?

Avoid Downtime!

SCARY NUMBERS TIME

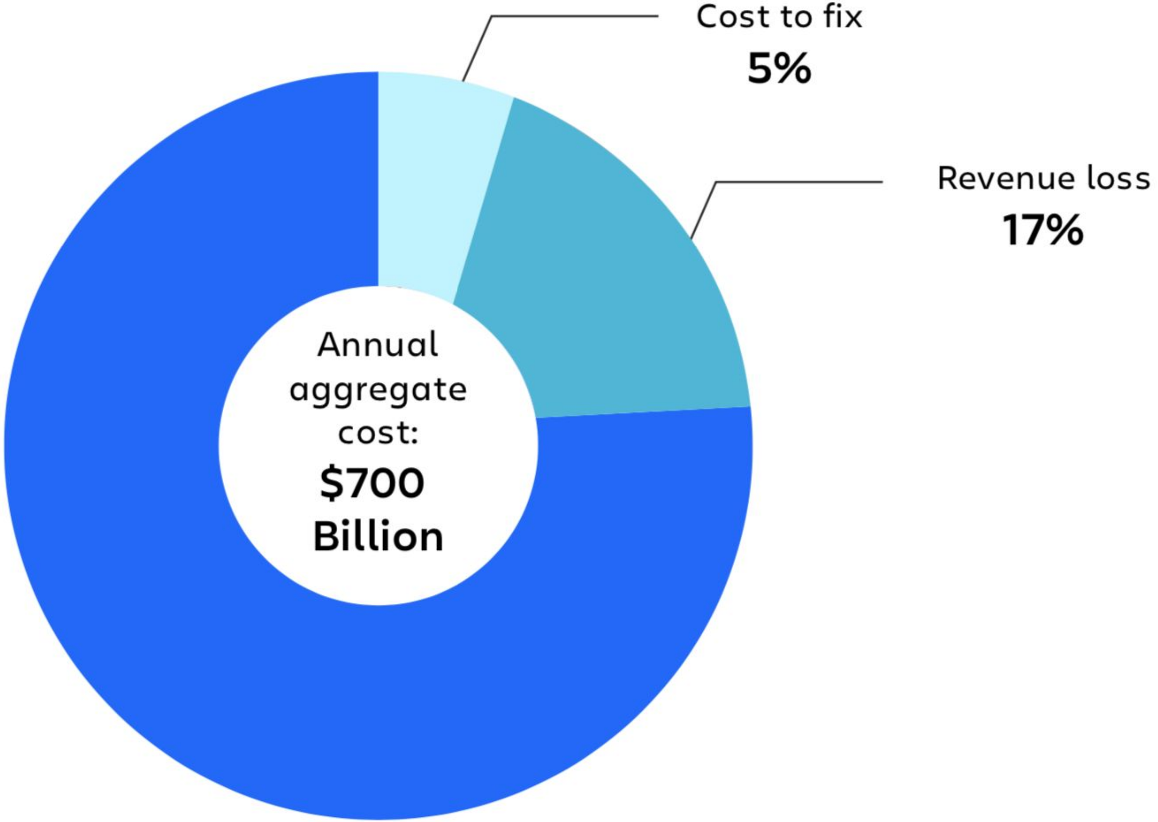
Average enterprise per-minute cost for primary system: $5600

Range from 140k-540k per hour

Numbers by Gartner

Numbers by IHS

- Estimated annual loss for North American orgs: Over $700 billion
  - 78% lost employee productivity
  - 17% lost revenue
  - Just 5% to fix the downtime
- Estimated mid-size company impact: 5 incidents for 27 hours of downtime a month, at a cool $1 million/year. $60 million for large



©IHS, IHS Infonetics The Cost of Server, Application, and Network Downtime: Annual North American Enterprise Survey and Calculator; 2016

Didn’t we apply ITIL and make IT Service Desks to help fix all this?

Well … that can encourage some structure, sure …

Old school Operations-centric support

- Tiered support (L1, L2 ..)
    - Cost optimization
    - Hero support
- Prescribed communication (email!)
    - Limited exposure
- Mean call time! KPIs!
- At least we have organized alerts
- File a ticket!

- Escalates / resolves slowly
    - New infrastructure dependencies
    - Whoops on vacation
- Forget Inbox Zero
    - Always On & Social Media
- What? My thing doesn’t work!
- To gain alert fatigue from
- ….

FILE A DASH TICKET

The 5 stages of Incident Management

Preparation

- What if?
    a. **Show of hands**: who has a plan or even scenarios for what might happen?
- Jump Bag - central basic info (who, what, where, how, access..)
- Runbooks - knowledge base of common scenarios and what to do
- Chaos Engineering (Monkeys!) - http://principlesofchaos.org

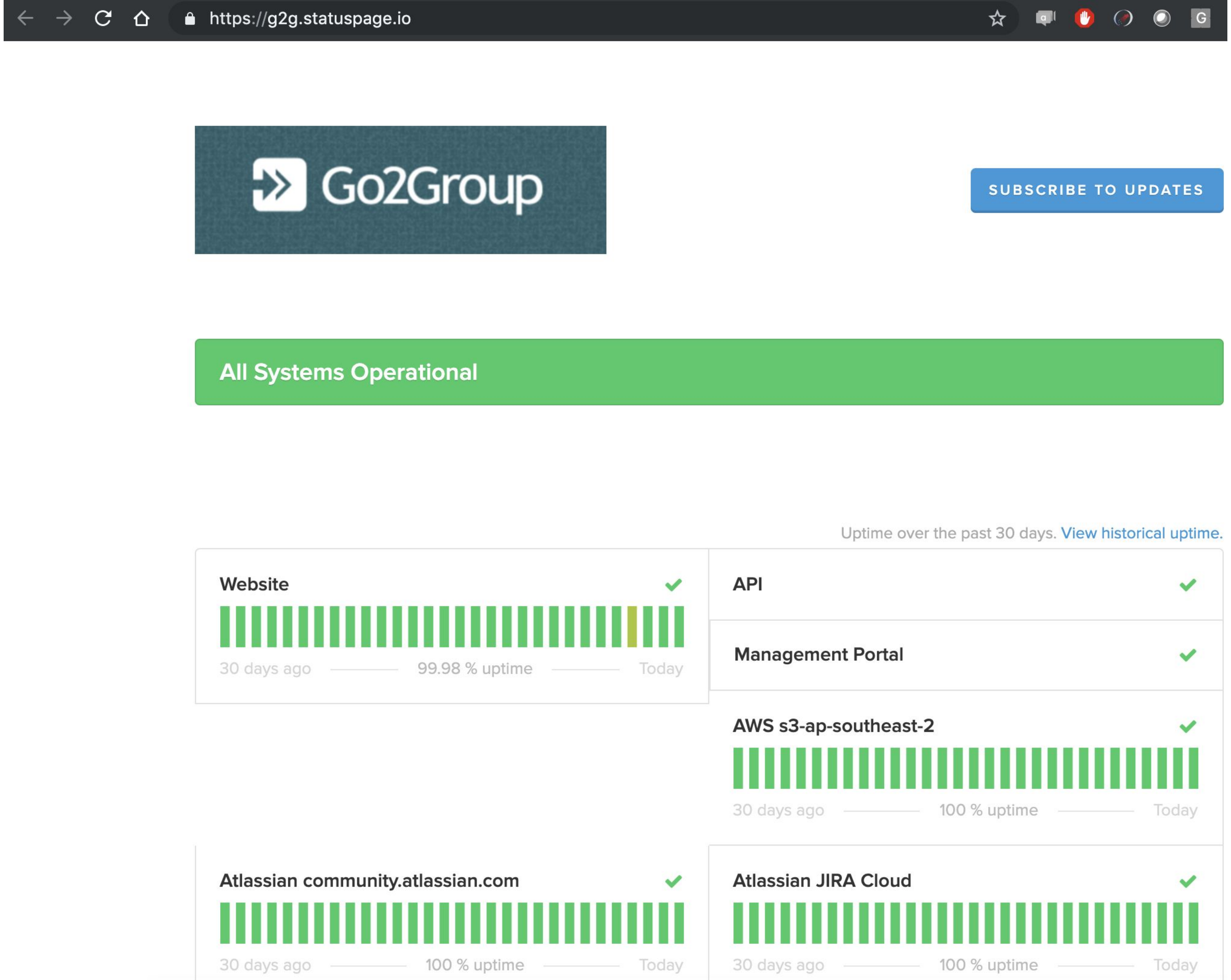
Detecting & Alerting

- Tools galore! More on that later …
- Go beyond an Operation Center (TOC/NOC) - automate it! Spread it!
- Quality instead of noise (anybody got dirty logs?)
- Rich alerts
- Who Watches the Watchmen? (Heartbeats!)

Containment

- Triage - don’t accept as a new normal! That sound familiar to anybody?
  - Me too! Had a Jenkins get infected with crypto mining malware 2 days ago …
  - Don’t let it linger -

    https://matrix.org/blog/2019/05/08/post-mortem-and-remediations-for-apr-11-security-incident
- Stop the bleeding
- No solo heroes - collaborate! More tools
- Transparency and live updates. Even more tools! Like StatusPage.io



All Systems Operational

Uptime over the past 30 days. View historical uptime.

Website ✓
30 days ago — 99.98 % uptime — Today

API ✓

Management Portal ✓

AWS s3-ap-southeast-2 ✓
30 days ago — 100 % uptime — Today

Atlassian community.atlassian.com ✓
30 days ago — 100 % uptime — Today

Atlassian JIRA Cloud ✓
30 days ago — 100 % uptime — Today

Remediation

- What happened?
- What do?
- Timeline?
- ChatOps as a tool (bonus with deep integration like OpsGenie Actions)
    - (Example: can do more than just resolve an alert with a button in Slack)

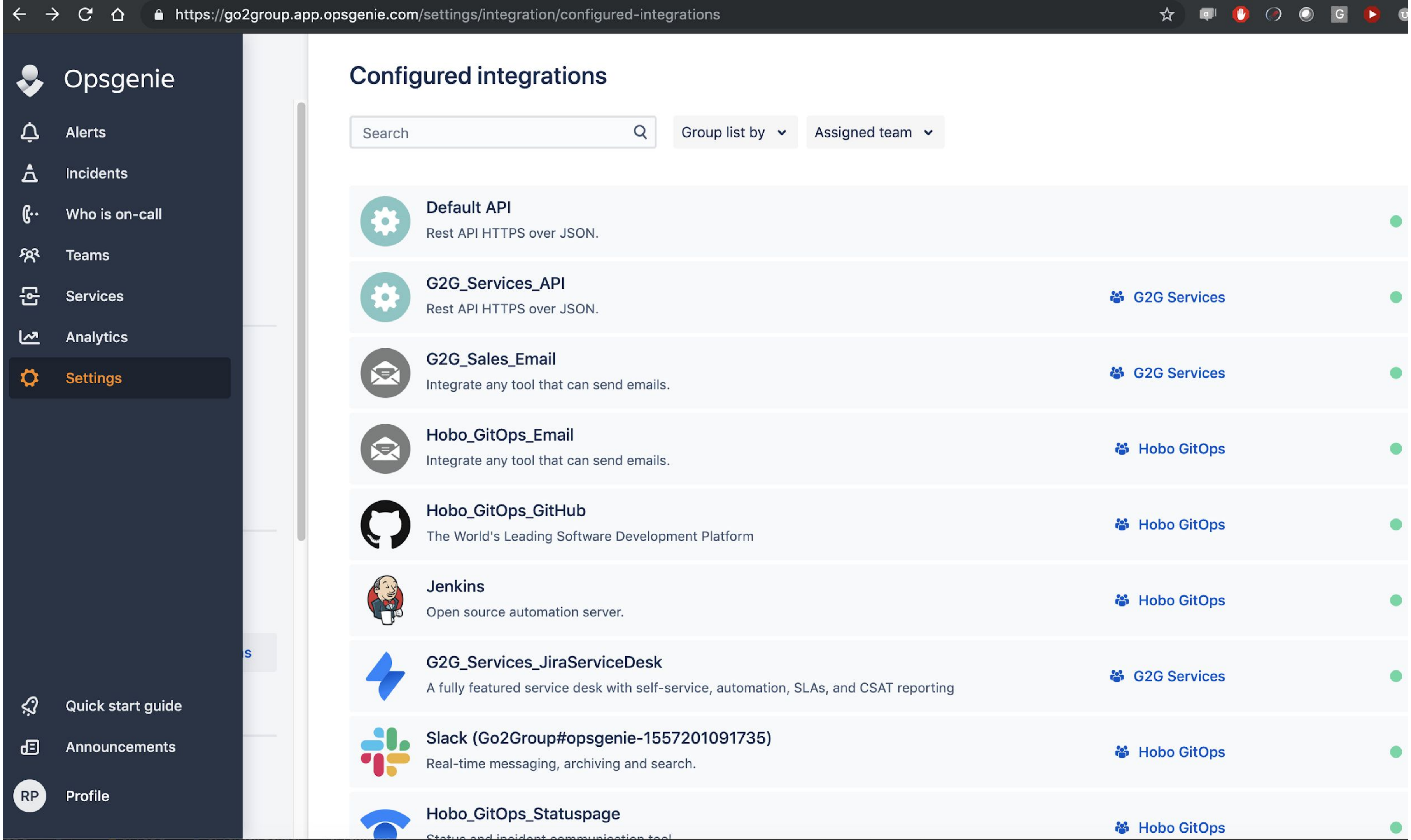
Analysis

- Post-mortem / RCA - why did it happen
- Learn from failure. Not just the big ones
- Don’t point fingers - just identify solutions

Okay, now what?

New tools!

- Tool middleware of sorts - feed into a central hub
- Let the hub help figure out the who what where how
- Integrate all the things!



Configured integrations

Default API Rest API HTTPS over JSON.		●
G2G_Services_API Rest API HTTPS over JSON.	G2G Services	●
G2G_Sales_Email Integrate any tool that can send emails.	G2G Services	●
Hobo_GitOps_Email Integrate any tool that can send emails.	Hobo GitOps	●
Hobo_GitOps_GitHub The World's Leading Software Development Platform	Hobo GitOps	●
Jenkins Open source automation server.	Hobo GitOps	●
G2G_Services_JiraServiceDesk A fully featured service desk with self-service, automation, SLAs, and CSAT reporting	G2G Services	●
Slack (Go2Group#opsgenie-1557201091735) Real-time messaging, archiving and search.	Hobo GitOps	●
Hobo_GitOps_Statuspage Status and incident communication tool	Hobo GitOps	●

But wait - what about DevOps?

I was promised there'd be DevOps! And maybe cookies.

- DevOps in Incident Management may be a stretch - audience ideas?
- Ultimate destination may become more of a No-Ops ?
  - https://www.appdynamics.com/blog/engineering/is-noops-the-end-of-devops-think-again/
  - http://perfcap.blogspot.com/2012/03/ops-devops-and-noops-at-netflix.html
- In the meantime - release the hobos!

YA KNOW I VASS NORMAL ONCE.
I CAME TO WORK VITH A TIE AND A SUIT
AND SOMETIMES PANFS.

What is GitOps and how do hobos relate?

We're using old school service desk and a dozen other isolated systems - halp!

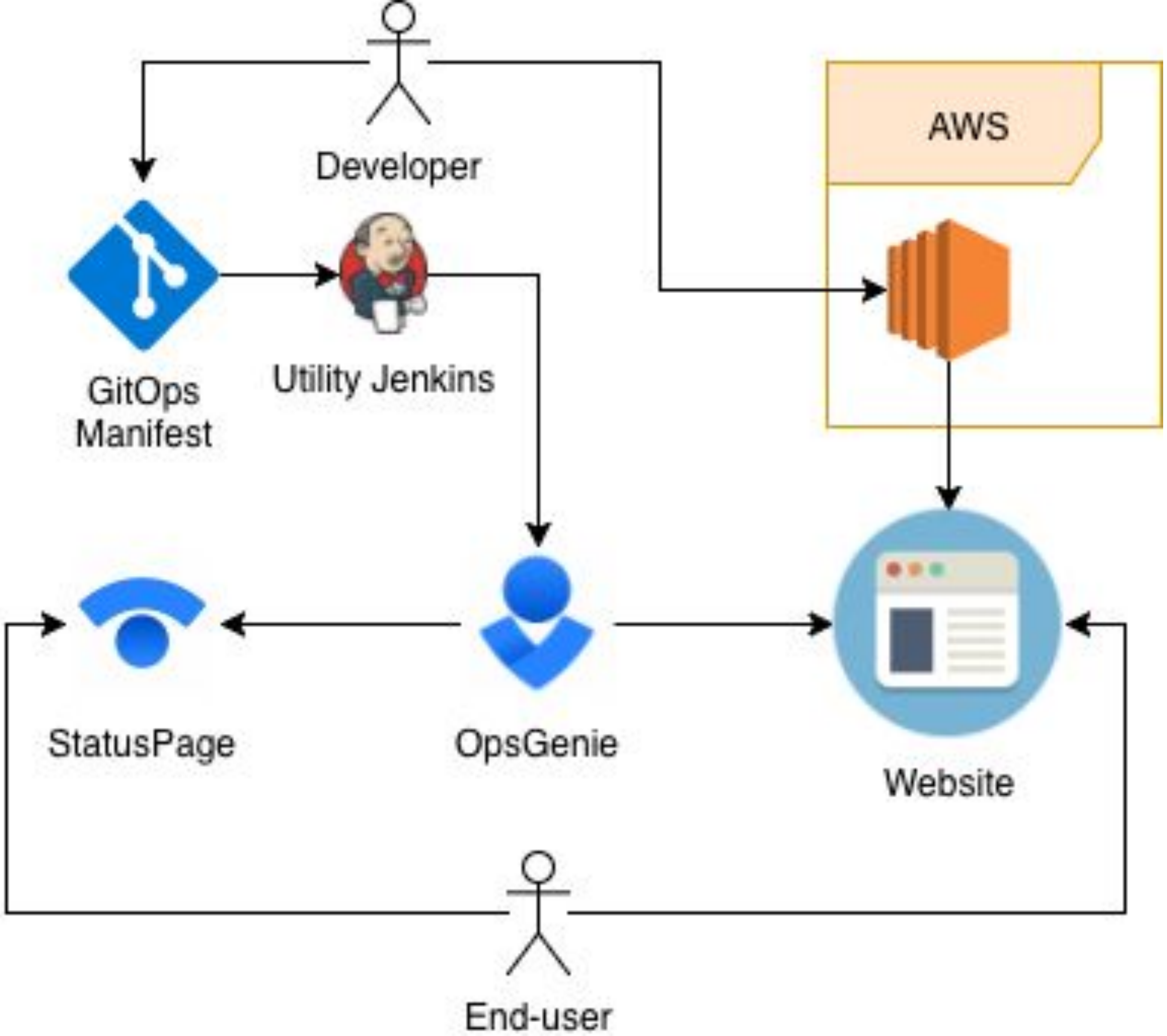
Well, for the teams that actually have any sort of structure ..

Your teams are all over the map. That's okay! Grab the best of each world, it is okay if the result looks a bit messy - to begin with

Jenkins!

Jenkins remains the king of flexibility
It may not always be glamorous work, but it works!

- Jenkins Job DSL - https://github.com/jenkinsci/job-dsl-plugin
- Layered seed jobs in different Git repositories - https://github.com/Cervator/GitOpsUtilityJobs.git
- Manifest repositories
  - Hobo GitOps manifests - for Jenkins - https://github.com/Cervator/GitOpsManifest
  - Environment manifests - for dev team apps
  - *Incident Management* hooks - new and shiny!


Developer
AWS
GitOps Manifest
Utility Jenkins
StatusPage
OpsGenie
Website
End-user

DEMO - Maybe?

How it works

Step 1

Set up some basic tools.
You could add minimal
Hobo GitOps structure

Step 2

Snare in pilot teams
(you’re in yourself!)

Step 3

Grow the circle. Build
shared libraries of stuff.
Make templates, good
practices, and share

The Future!

What’s next?

WELL MISTER HOBO, IT'S BEEN A LONG ROAD BUT, I THINK YOU'RE READY TO BECOME A PRODUCTIVE MEMBER OF SOCIETY.