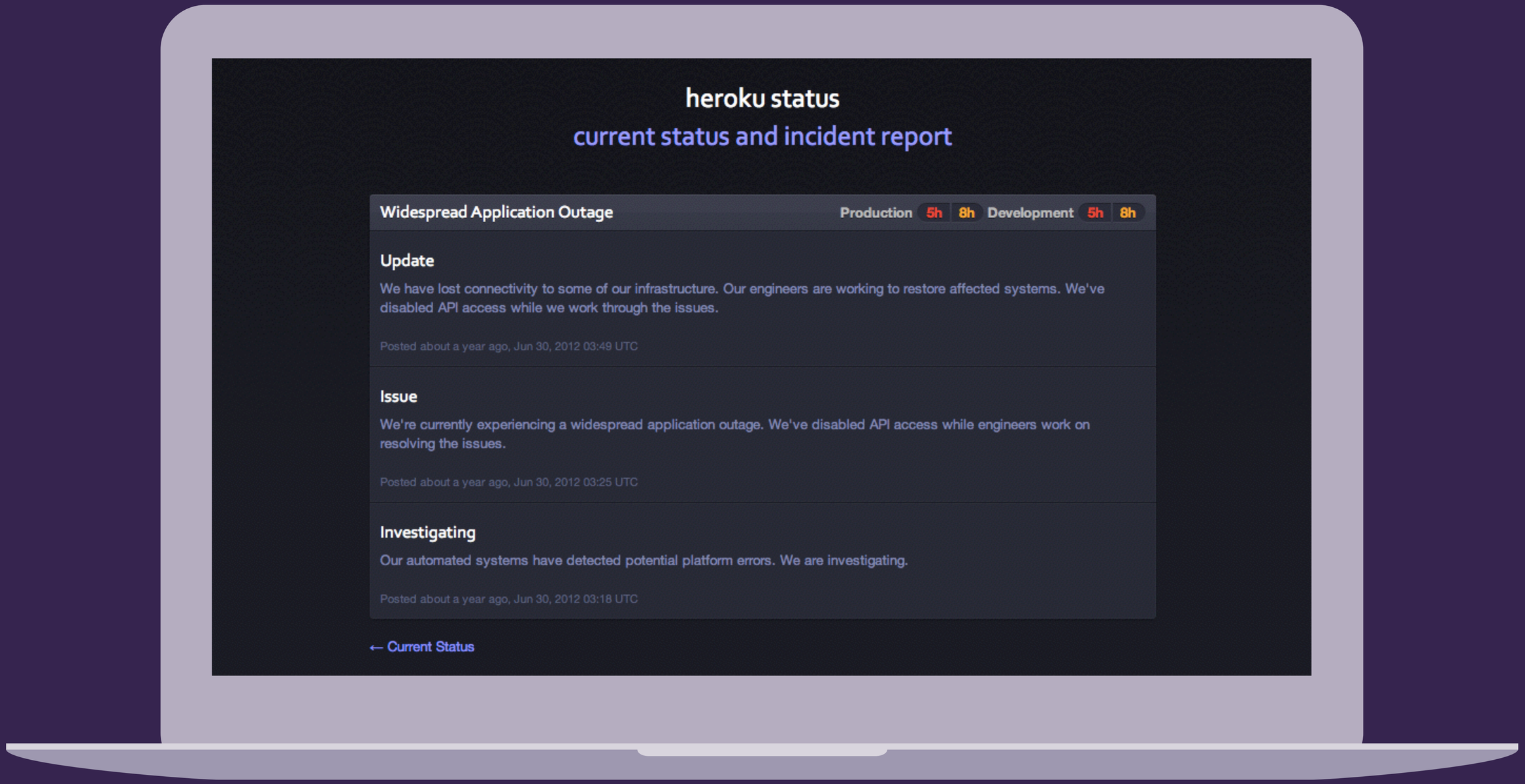
Heroku Operations

Noah Zoschke
noah@heroku.com


heroku status
current status and incident report
Widespread Application Outage
Production 5h 8h Development 5h 8h
Update
We have lost connectivity to some of our infrastructure. Our engineers are working to restore affected systems. We've disabled API access while we work through the issues.
Posted about a year ago, Jun 30, 2012 03:49 UTC
Issue
We're currently experiencing a widespread application outage. We've disabled API access while engineers work on resolving the issues.
Posted about a year ago, Jun 30, 2012 03:25 UTC
Investigating
Our automated systems have detected potential platform errors. We are investigating.
Posted about a year ago, Jun 30, 2012 03:18 UTC
← Current Status

Problem: Cloud Services

Lost EBS + EC2

API + Customer Databases

30% App Servers

Internal Ops Apps + Instances

AWS API Offline

Can’t get new capacity

PagerDuty Crushed

No alerts coming through

Heroku API Offline

Customers helpless

Solution: HA Architecture

Solution: HA Architecture

Solution: People & Operational Culture

A Personal Account...

Problem: Culture

Feature Culture → Too Much Software

No inventory of what's up or down

Surprising dependencies

Hacker Culture → Poorly Written Software

Feature rich, not fool proof

Lots of "beta" services with production workloads

Rockstar Culture → Individual Ownership

Lots of low bus factor

Implicit Culture → Unclear Expectations

Can I escalate?

How do we prioritize services and customers?

“Feature Culture” Side Effect: Legacy Services

Problem: Legacy Services

Two Routing Services

Router → Nginx/Varnish → Dyno (Bamboo)

Router → Dyno (Cedar)

Two Database Services

Shiny New Dedicated Databases (Heroku Postgres)

Years-old Legacy Shared Databases

Five Metrics Services, etc...

Solution: Sunsetting Culture

Treat Sunsetting as First Class Product and Engineering Work

Meticulously Catalog Running Services

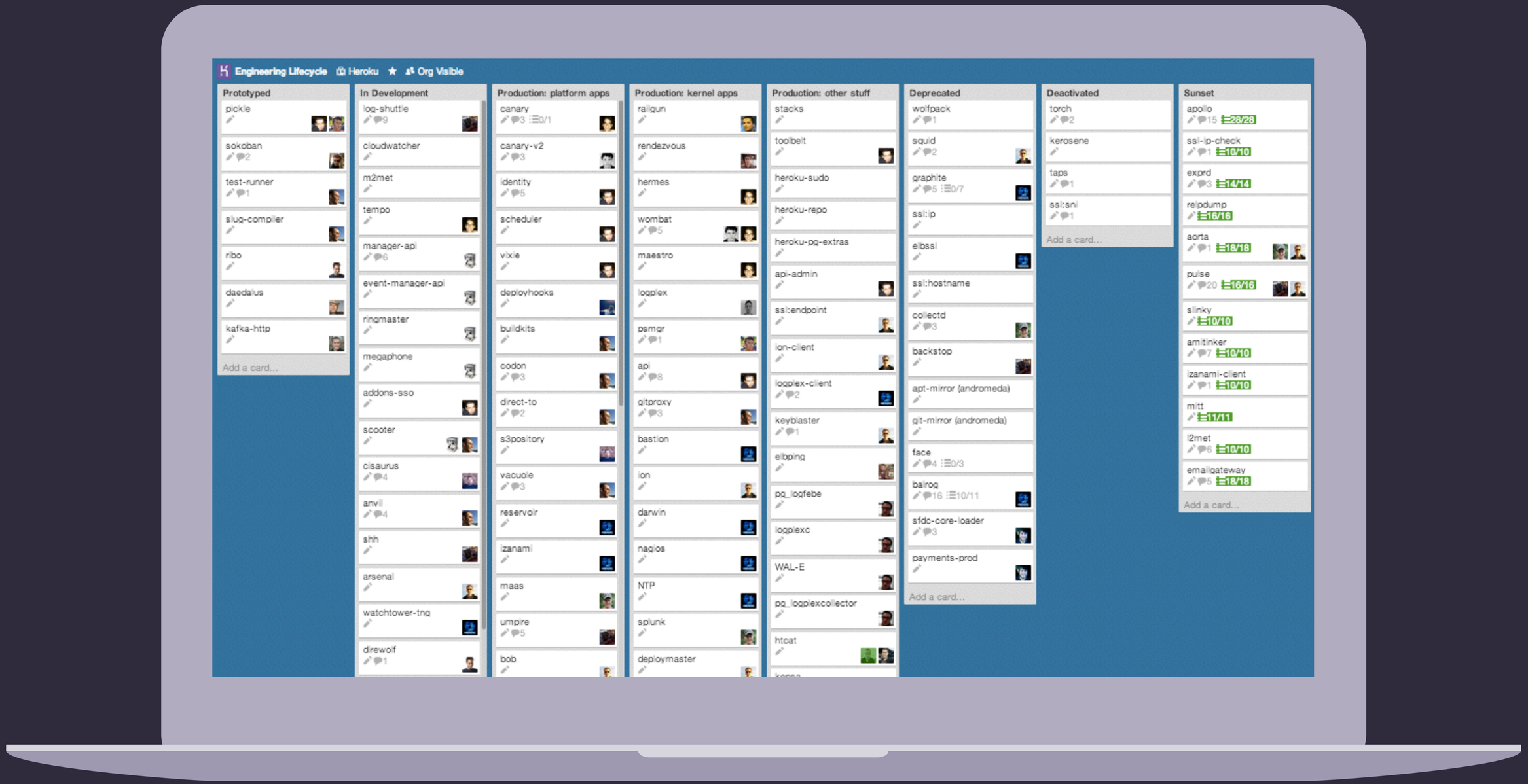
Celebrate Success When Shutting One Down

Recipe: Lifecycle Board

Prototype → Development → Production → Deprecated → Deactivated → Sunset

Follow Checklists to Advance

Reflect Explicit Owners


Engineering Lifecycle
Heroku
Org Visible
Prototyped
pickle
sokoban
test-runner
slug-compiler
ribo
daedalus
kafka-http
Add a card...
In Development
log-shuttle
cloudwatcher
m2met
tempo
manager-api
event-manager-api
ringmaster
megaphone
addons-sso
scooter
cisaurus
anvil
shh
arsenal
watchtower-tng
direwolf
Production: platform apps
canary
canary-v2
identity
scheduler
vixie
deployhooks
buildkits
codon
direct-to
s3pository
vacuole
reservoir
izanami
maas
umpire
bob
Production: kernel apps
railgun
rendezvous
hermes
wombat
maestro
logplex
psmgr
api
gitproxy
bastion
ion
darwin
nagios
NTP
splunk
deploymaster
Production: other stuff
stacks
toolbelt
heroku-sudo
heroku-repo
heroku-pg-extras
api-admin
ssl:endpoint
ion-client
logplex-client
keyblaster
elbping
pg_logfebe
logplexc
WAL-E
pg_logplexcollector
htcat
Deprecated
wolfpack
squid
graphite
ssl:ip
elbssl
ssl:hostname
collectd
backstop
apt-mirror (andromeda)
git-mirror (andromeda)
face
bairog
sfdc-core-loader
payments-prod
Add a card...
Deactivated
torch
kerosene
taps
ssl:sni
Add a card...
Sunset
apollo
ssl-ip-check
expvd
relpdump
aorta
pulse
slinky
amitinker
izanami-client
mitt
l2met
emailgateway
Add a card...

“Hacker Culture” Side Effect: Inoperable Software

Recipe: Production Checklist

Code is visible on GitHub

Has operations docs with executable instructions for common tasks

Has a high-fidelity staging setup with production parity

Alerts a human if it is down

Uses structured logging

Enforces SSL access

Any credentials and their rotation procedures are added to “cred rolls” list

Send a launch email to engineering@ describing the new component

Move to Production on the Engineering Lifecycle board

Auto-scaled to maintain the needed number of instances

Set up to terminate unhealthy instances

“Implicit Culture” Side Effect: Platform and Pager Chaos

Problem: Implicit

Do I need to fix these warnings this week? Or put it off?

Can I escalate this alert? Should I?

Should I update the status site? Will someone else?

Recipe: PagerDuty Discipline

Everyone engineer is on-call

Every page is visible in HipChat

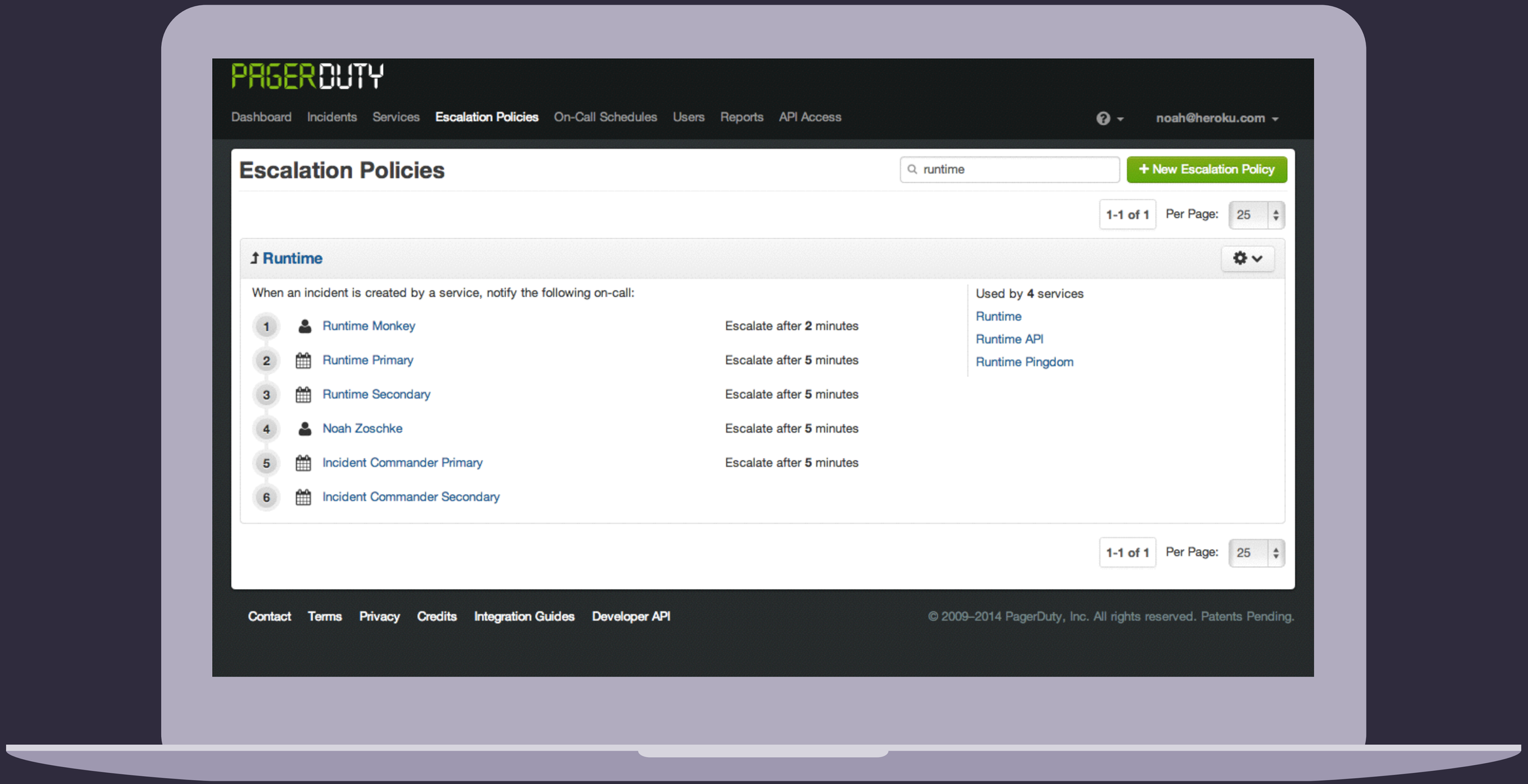
Monkey - Everyone should help ack pages in HipChat during work hours

Level 1 - Explicit expectation of first responder after hours

Level 2 - Explicit value that the team has each other's back

Engineering Manager - Explicit accountability for the whole team and its body of work service

Incident Commander - Experts trained in explicit procedures around updating the status site, opening up AWS tickets, paging extra engineers, etc.


PAGERDUTY
Dashboard
Incidents
Services
Escalation Policies
On-Call Schedules
Users
Reports
API Access
noah@heroku.com
Escalation Policies
runtime
+ New Escalation Policy
1-1 of 1
Per Page:
25
Runtime
When an incident is created by a service, notify the following on-call:
1 Runtime Monkey
Escalate after 2 minutes
2 Runtime Primary
Escalate after 5 minutes
3 Runtime Secondary
Escalate after 5 minutes
4 Noah Zoschke
Escalate after 5 minutes
5 Incident Commander Primary
Escalate after 5 minutes
6 Incident Commander Secondary
Used by 4 services
Runtime
Runtime API
Runtime Pingdom
1-1 of 1
Per Page:
25
Contact
Terms
Privacy
Credits
Integration Guides
Developer API
© 2009–2014 PagerDuty, Inc. All rights reserved. Patents Pending.

Recipe: Pager Metrics

Measure everything and review weekly

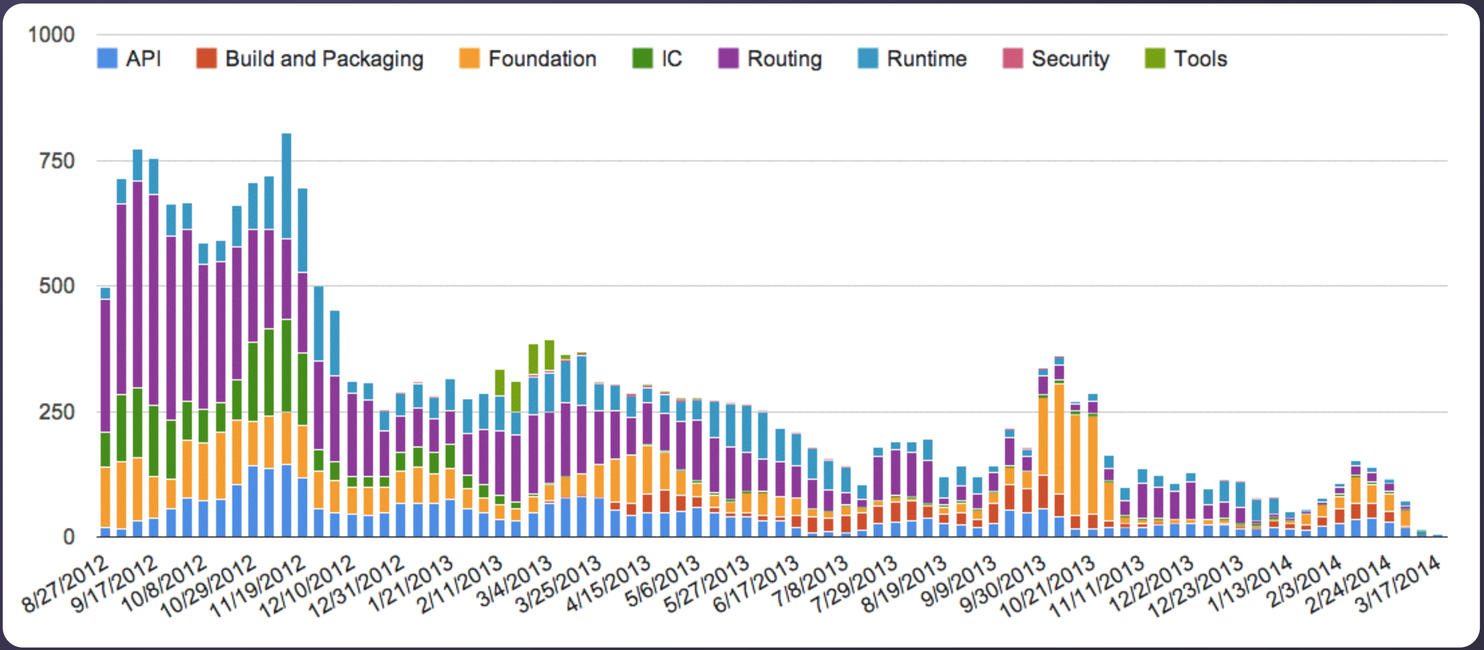
After hours pages are detrimental to engineering health and well being

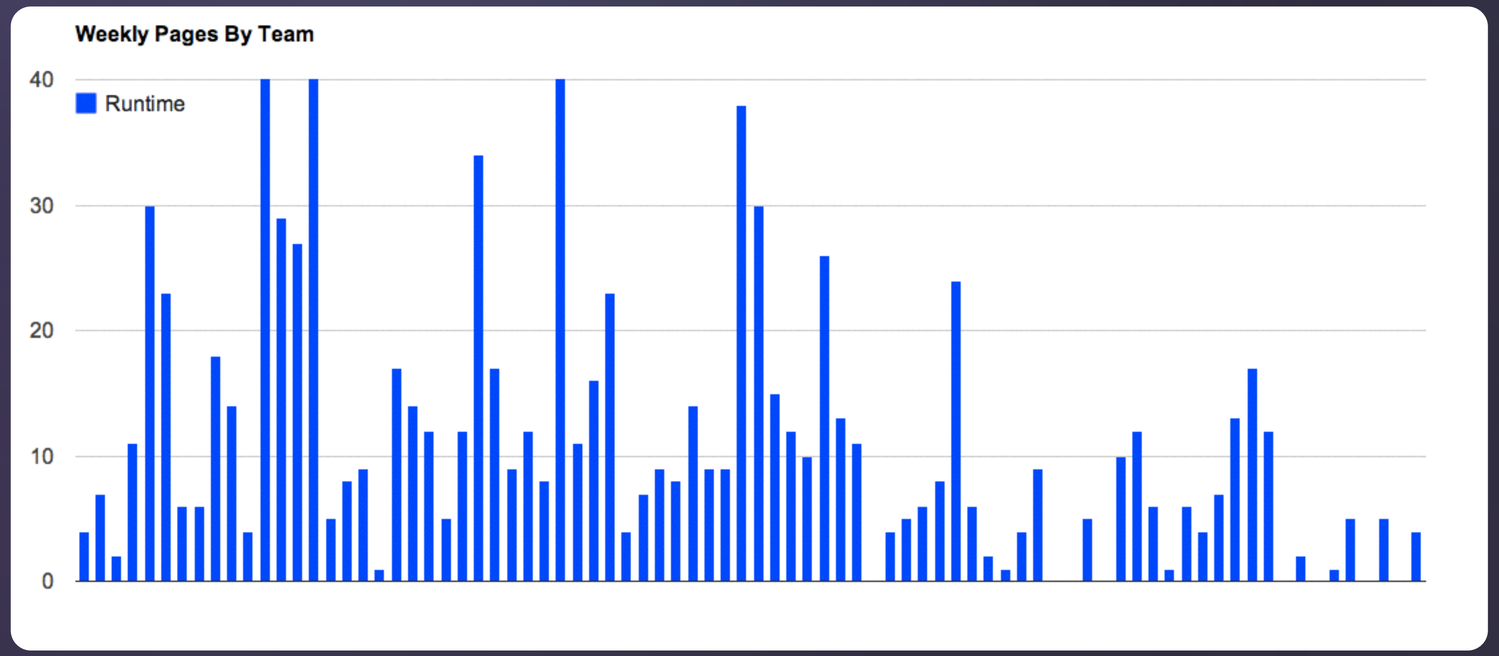
Engineers deserve weeks with no pages

Engineers have power to improve the operator experience

Engineering Managers are responsible for managing balance between operations and feature work

Service Reliability Engineering (SRE) team is accountable for overall pager burden program


1000
750
500
250
0
API
Build and Packaging
Foundation
IC
Routing
Runtime
Security
Tools
8/27/2012
9/17/2012
10/8/2012
10/29/2012
11/19/2012
12/10/2012
12/31/2012
1/21/2013
2/11/2013
3/4/2013
3/25/2013
4/15/2013
5/6/2013
5/27/2013
6/17/2013
7/8/2013
7/29/2013
8/19/2013
9/9/2013
9/30/2013
10/21/2013
11/11/2013
12/2/2013
12/23/2013
1/13/2014
2/3/2014
2/24/2014
3/17/2014


Weekly Pages By Team
Runtime
40
30
20
10
0