Automation, Patching and Insights

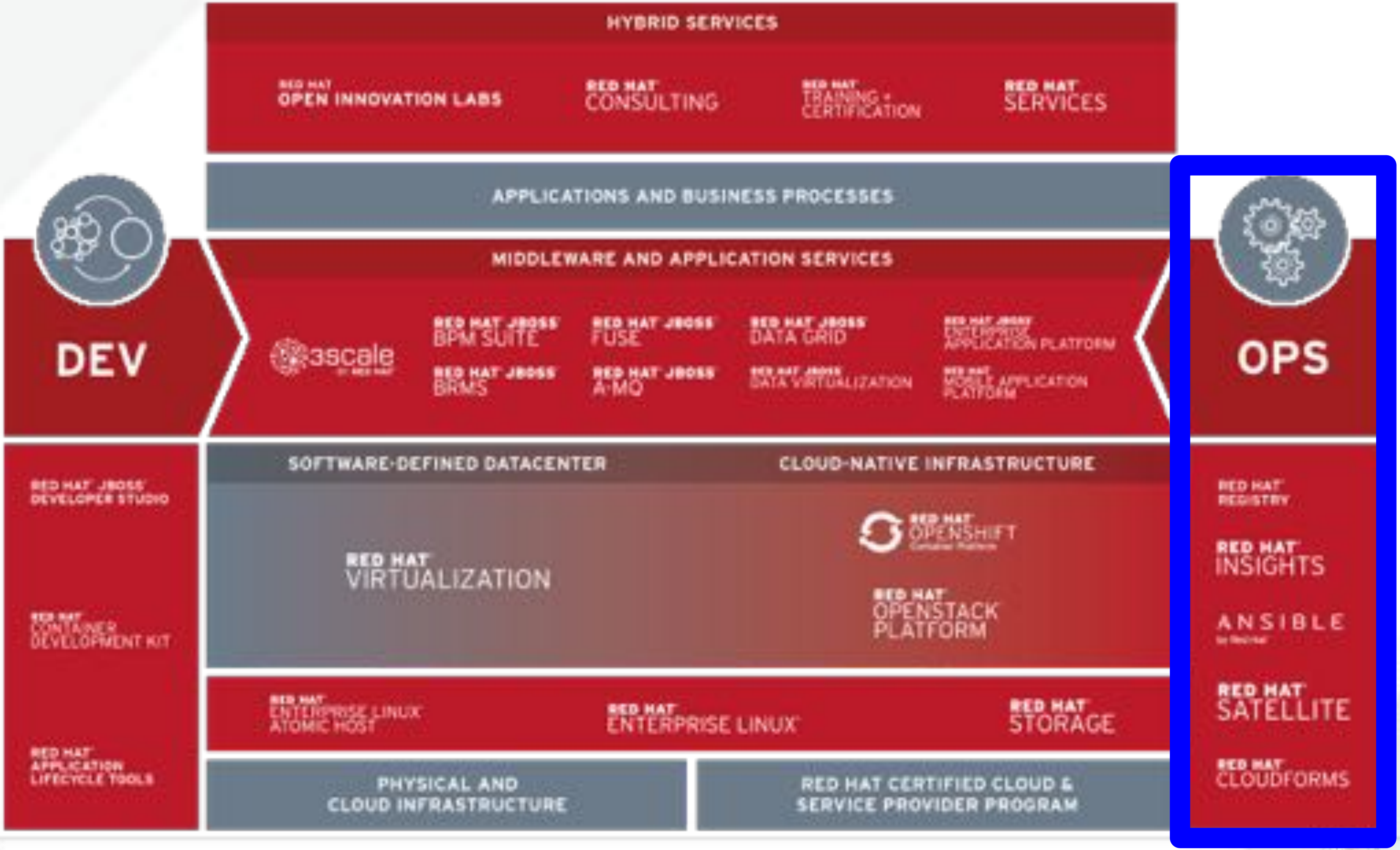
Why hope is not a Strategy

Chris Milsted
Principal Solution Architect, Financial Services
cmilsted@redhat.com

Agenda

- Disruption in the industry - technology
- Current state - Traditional Operations
- Application trends - what the developers are doing
- Broken Paradigms
- SRE - Why?
- The New Normal
- Predictive Analytics - how it can help
- Putting it all together
- Q&A

Red Hat Navigator



HYBRID SERVICES

RED HAT OPEN INNOVATION LABS

RED HAT CONSULTING

RED HAT TRAINING + CERTIFICATION

RED HAT SERVICES

APPLICATIONS AND BUSINESS PROCESSES

DEV

MIDDLEWARE AND APPLICATION SERVICES

3scale BY RED HAT

RED HAT JBOSS BPM SUITE

RED HAT JBOSS BRMS

RED HAT JBOSS FUSE

RED HAT JBOSS A-MQ

RED HAT JBOSS DATA GRID

RED HAT JBOSS DATA VIRTUALIZATION

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

RED HAT MOBILE APPLICATION PLATFORM

OPS

RED HAT JBOSS DEVELOPER STUDIO

RED HAT CONTAINER DEVELOPMENT KIT

RED HAT APPLICATION LIFECYCLE TOOLS

SOFTWARE-DEFINED DATACENTER

CLOUD-NATIVE INFRASTRUCTURE

RED HAT VIRTUALIZATION

RED HAT OPENSHIFT Container Platform

RED HAT OPENSTACK PLATFORM

RED HAT ENTERPRISE LINUX ATOMIC HOST

RED HAT ENTERPRISE LINUX

RED HAT STORAGE

PHYSICAL AND CLOUD INFRASTRUCTURE

RED HAT CERTIFIED CLOUD & SERVICE PROVIDER PROGRAM

RED HAT REGISTRY

RED HAT INSIGHTS

ANSIBLE by Red Hat

RED HAT SATELLITE

RED HAT CLOUDFORMS

Disruption in the industry

N/A

1 1

1 38

36 12

1009 2091

180 5 421

Global Utilization

CPU

75%

Memory

604.7

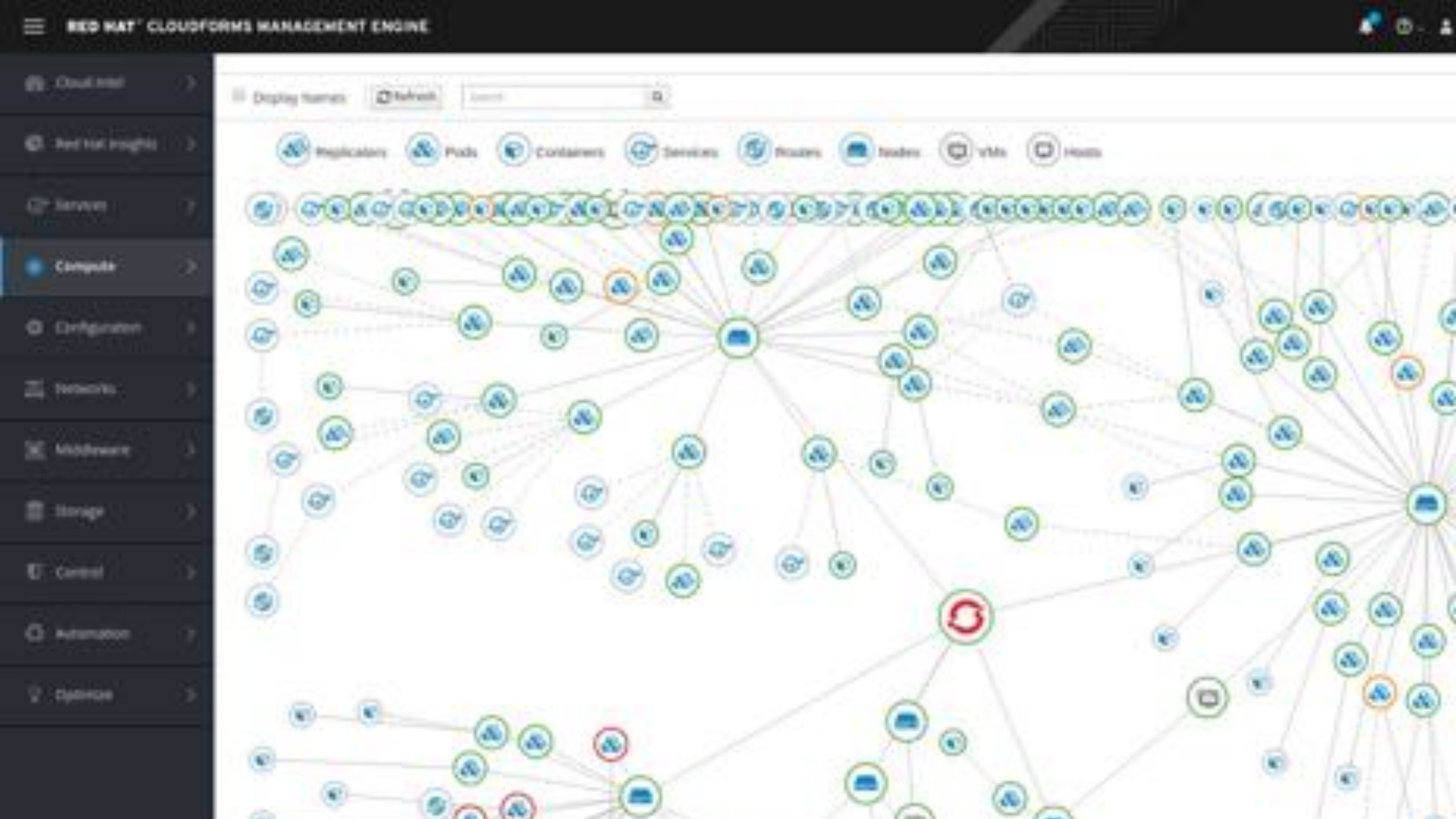
Storage

38.5

25 %

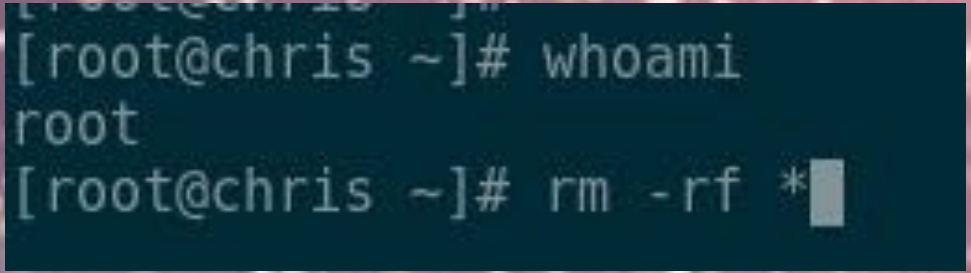
1951.4

18.5


RED HAT CLOUDFORMS MANAGEMENT ENGINE
Cloud Intel
Red Hat Insights
Services
Compute
Configuration
Networks
Middleware
Storage
Control
Automation
Optimize
Display Names
Refresh
Search
Replicators
Pods
Containers
Services
Routes
Nodes
VMs
Hosts

1 physical server	10 Virtual Machines	100 Containers
1 Monolithic application	1 application - 10 environments	1 app in 100 pieces
Health = binary	Health = binary	Health = complex

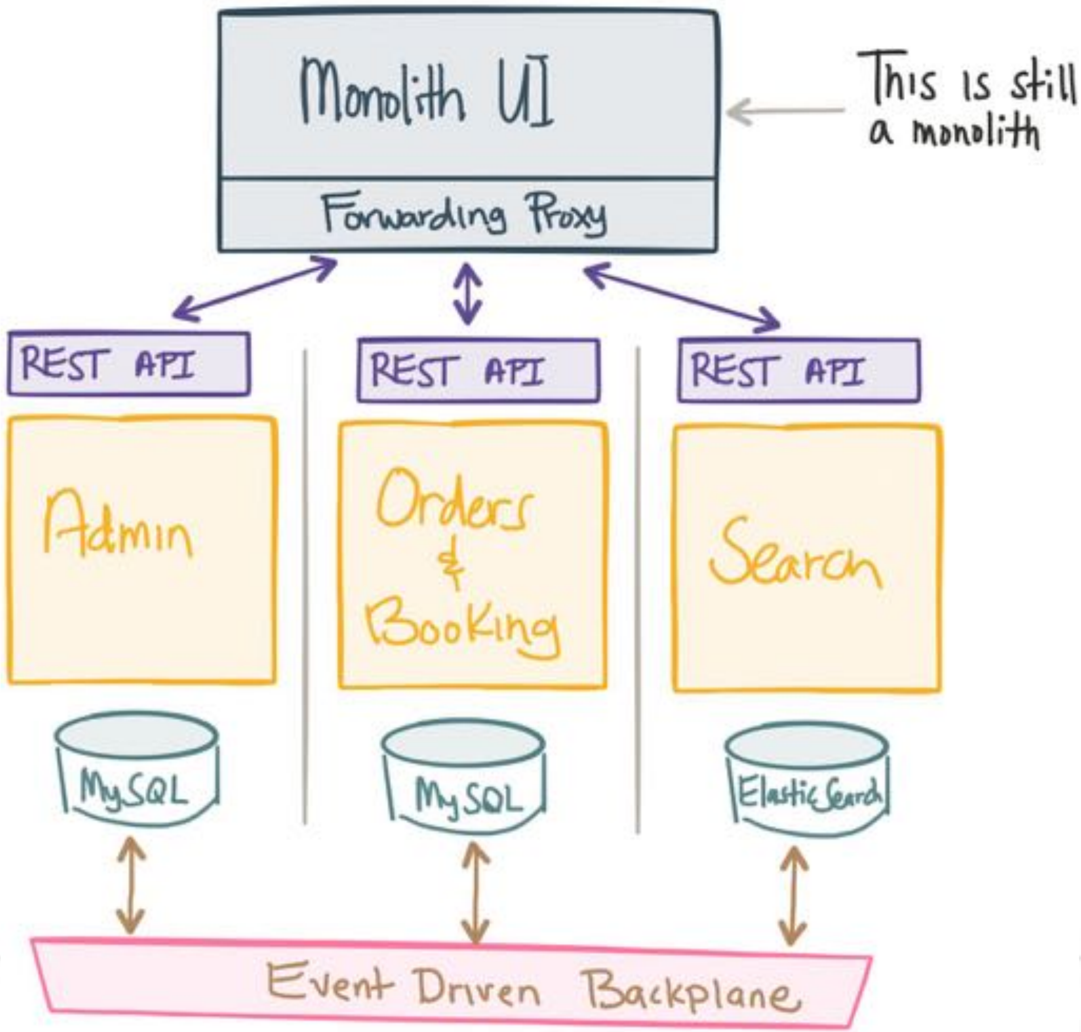
State of play today
Traditional Operations

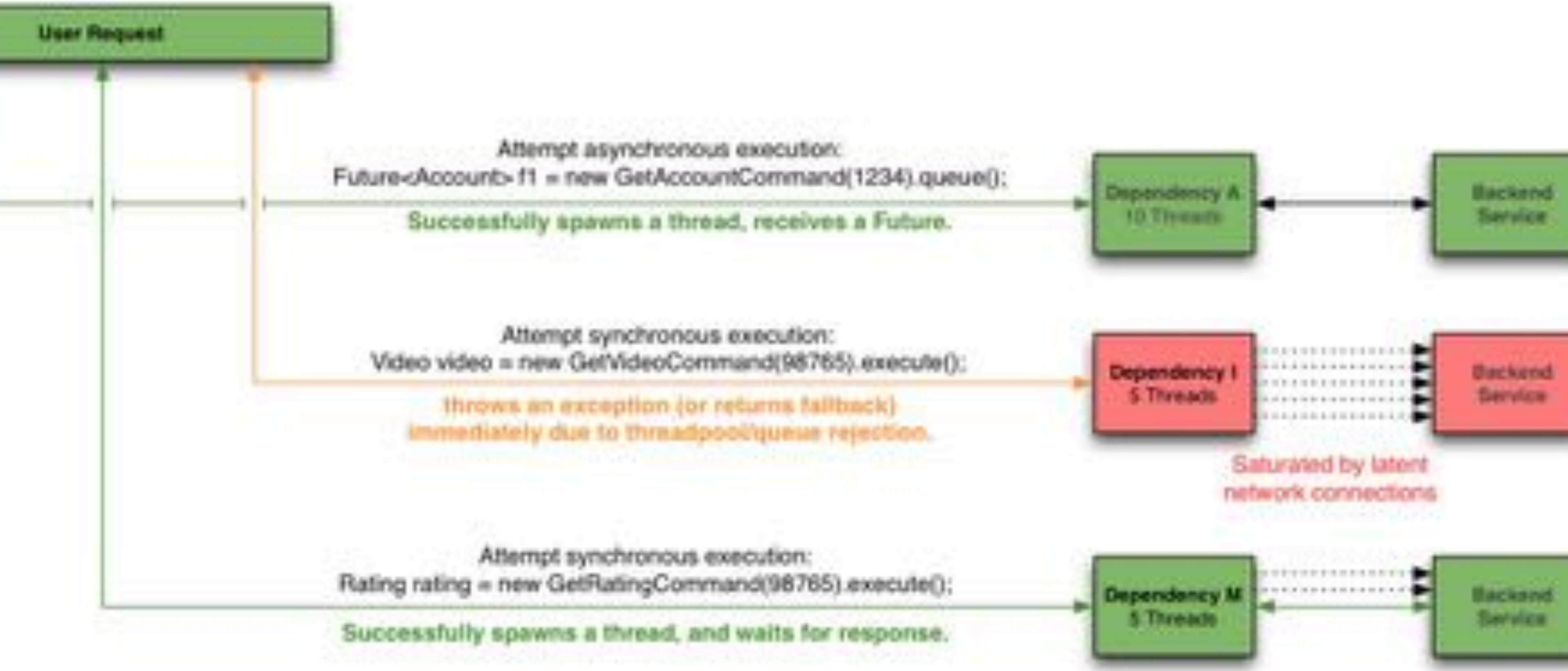


```
[root@chris ~]#
[root@chris ~]# whoami
root
[root@chris ~]# rm -rf *
```

Photo via Visual Hunt

Applications are changing
Micro-Services


Monolith UI
Forwarding Proxy
This is still a monolith
REST API
REST API
REST API
Admin
Orders & Booking
Search
MySQL
MySQL
ElasticSearch
Event Driven Backplane



Attempt asynchronous execution:
Future<Account> f1 = new GetAccountCommand(1234).queue();

Successfully spawns a thread, receives a Future.

Attempt synchronous execution:
Video video = new GetVideoCommand(98765).execute();

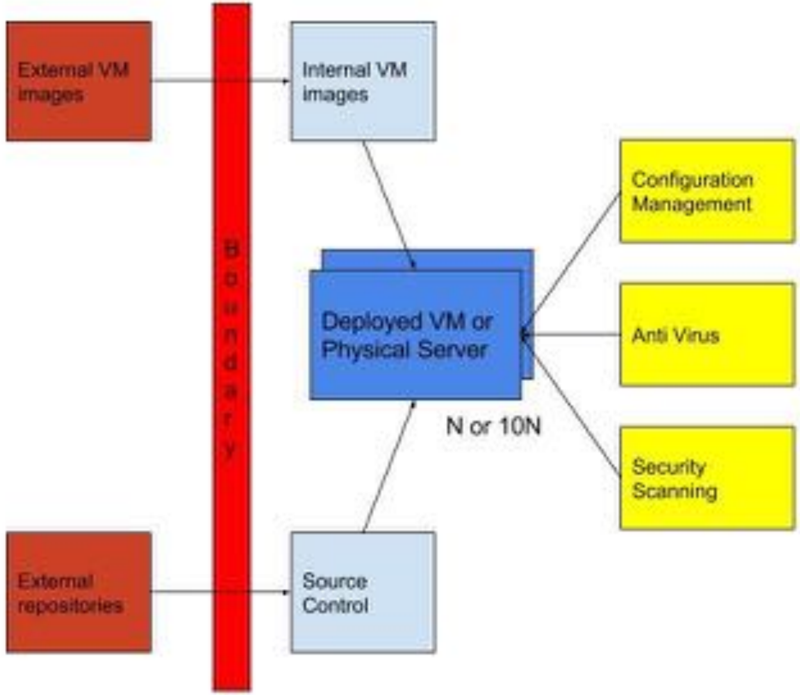
throws an exception (or returns fallback)
immediately due to threadpool/queue rejection.

Saturated by latent
network connections

Attempt synchronous execution:
Rating rating = new GetRatingCommand(98765).execute();

Successfully spawns a thread, and waits for response.

Broken Paradigms


External VM images
Internal VM images
Boundary
Configuration Management
Deployed VM or Physical Server
Anti Virus
N or 10N
Security Scanning
External repositories
Source Control

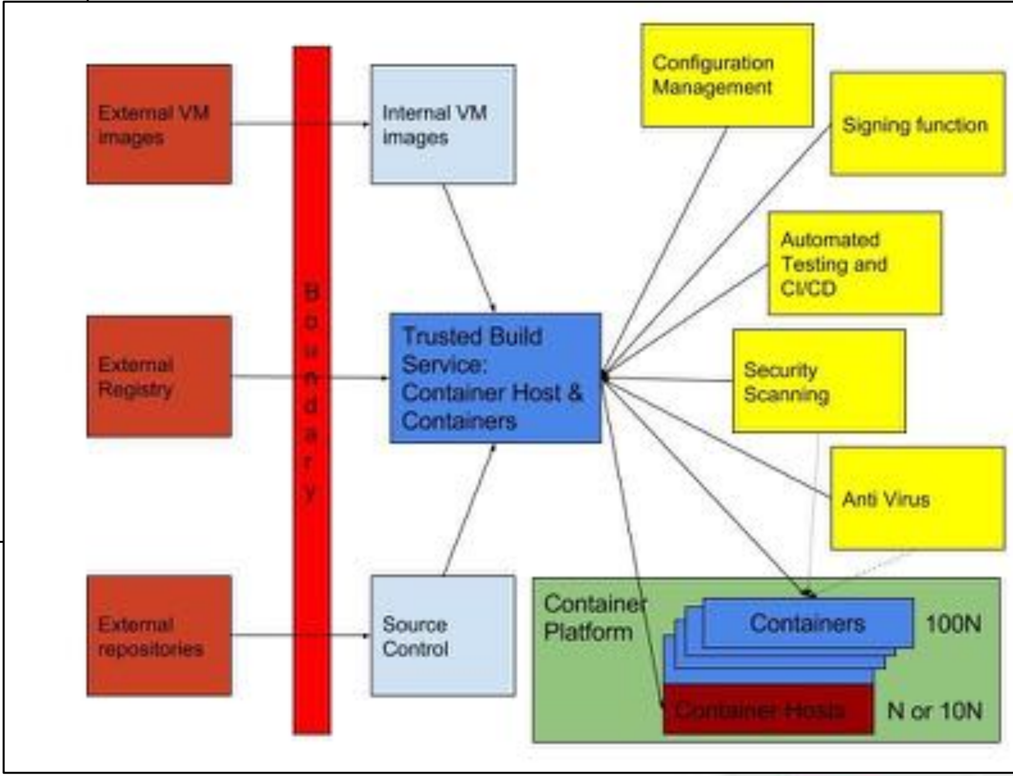

Configuration Management
Signing function
External VM images
Internal VM images
Automated Testing and CI/CD
Boundary
External Registry
Trusted Build Service: Container Host & Containers
Security Scanning
Anti Virus
External repositories
Source Control
Container Platform
Containers
100N
Container Hosts
N or 10N

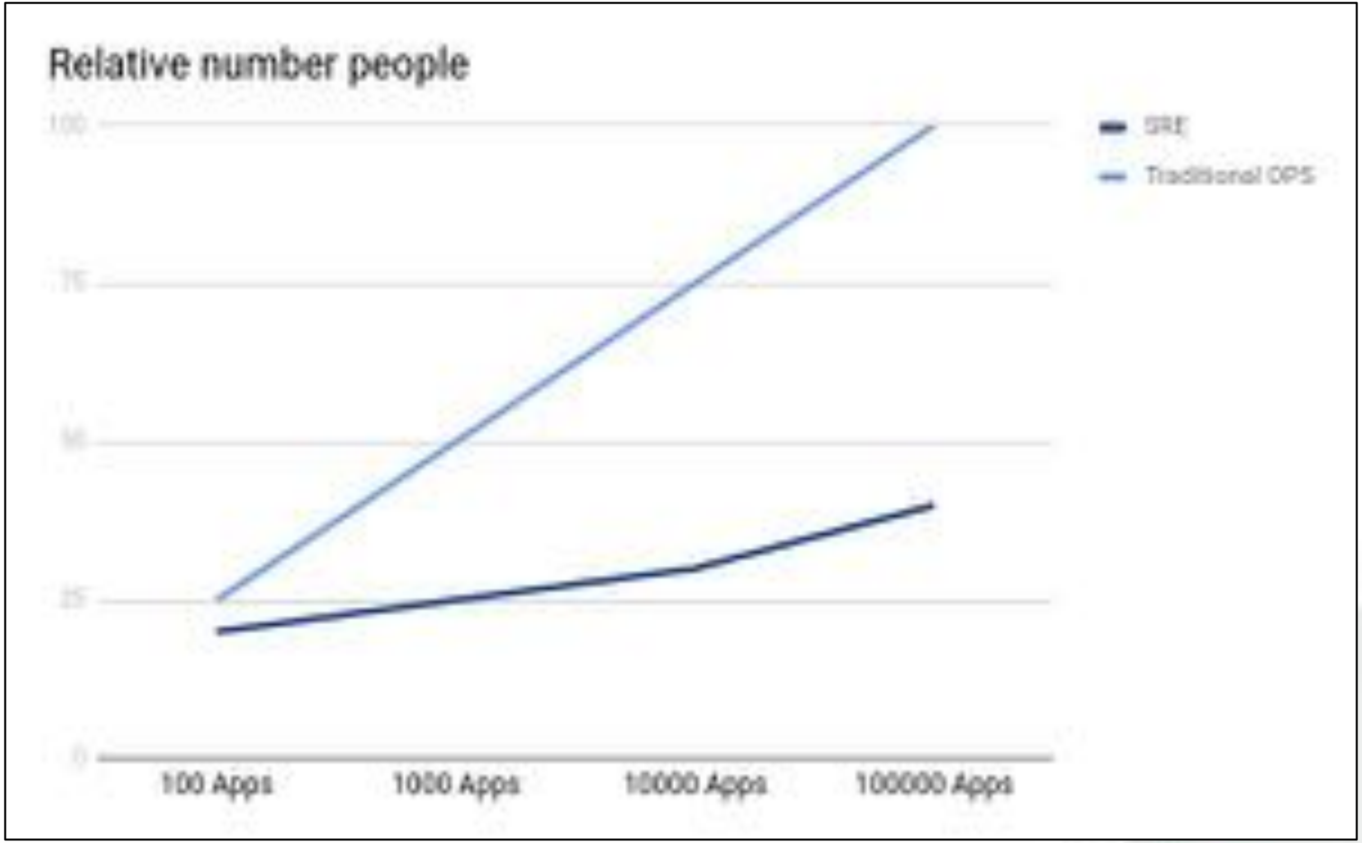
Photo credit: Michel Curi via Visualhunt / CC BY

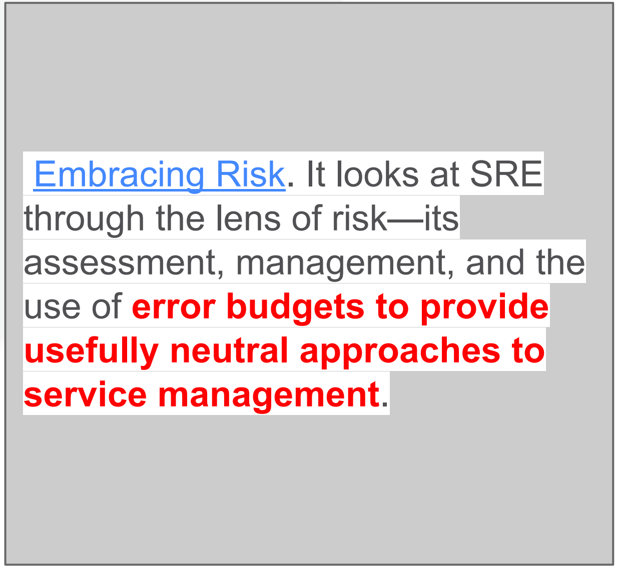
SRE - Why?

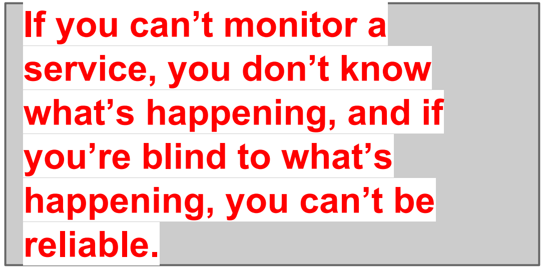
Site Reliability Engineering

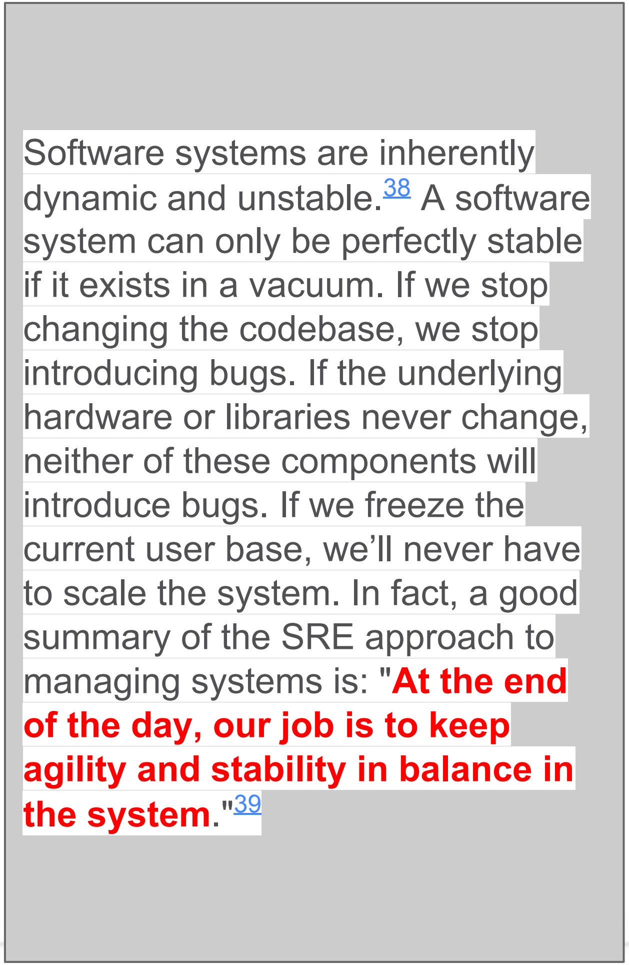
How Google Runs Production Systems

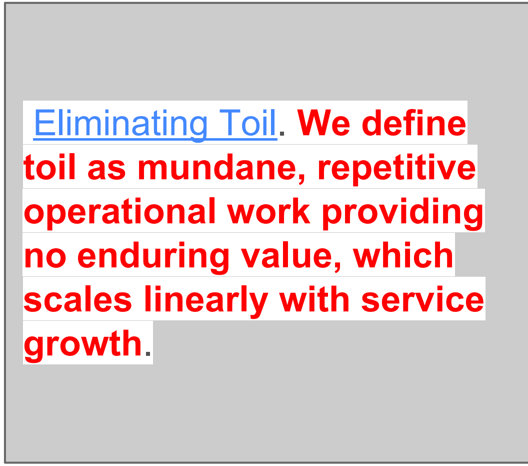
Without constant engineering, operations load increases and teams will need more people just to keep pace with the workload. Eventually, **a traditional ops-focused group scales linearly with service size**: if the products supported by the service succeed, the operational load will grow with traffic. That means hiring more people to do the same tasks over and over again.

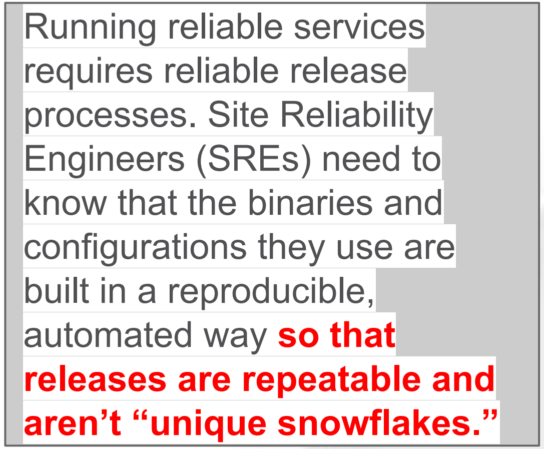



Embracing Risk. It looks at SRE through the lens of risk—its assessment, management, and the use of **error budgets to provide usefully neutral approaches to service management**.


If you can’t monitor a service, you don’t know what’s happening, and if you’re blind to what’s happening, you can’t be reliable.


Software systems are inherently dynamic and unstable.[38] A software system can only be perfectly stable if it exists in a vacuum. If we stop changing the codebase, we stop introducing bugs. If the underlying hardware or libraries never change, neither of these components will introduce bugs. If we freeze the current user base, we’ll never have to scale the system. In fact, a good summary of the SRE approach to managing systems is: "**At the end of the day, our job is to keep agility and stability in balance in the system**."[39]


Eliminating Toil. **We define toil as mundane, repetitive operational work providing no enduring value, which scales linearly with service growth**.

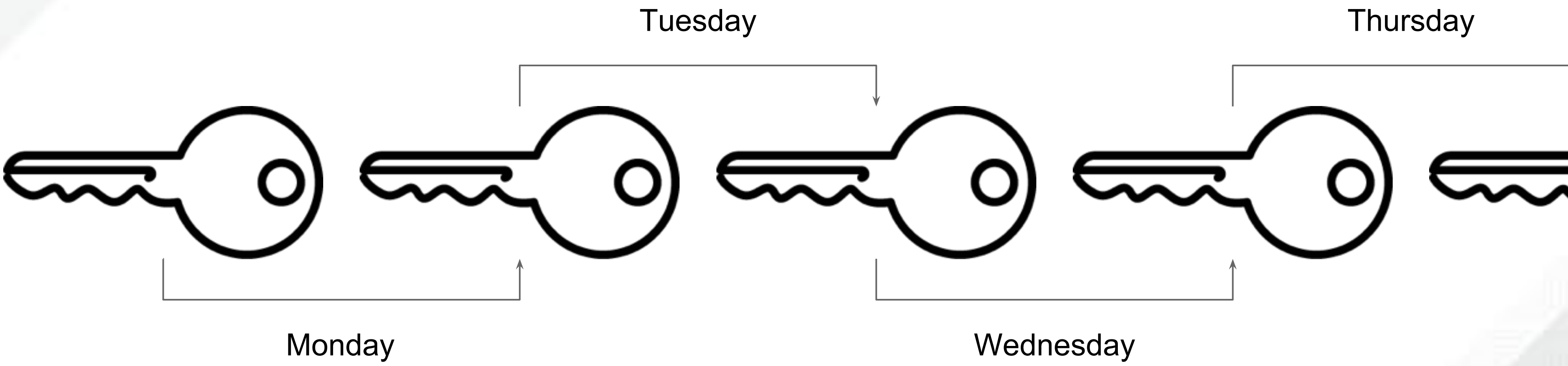

Running reliable services requires reliable release processes. Site Reliability Engineers (SREs) need to know that the binaries and configurations they use are built in a reproducible, automated way **so that releases are repeatable and aren’t “unique snowflakes.”**

The New Normal

So we made the developers do CI/CD…
Now it is our turn

Rotate - RePave - Repair

Let's Melt the Snowflakes


Tuesday
Thursday
Monday
Wednesday

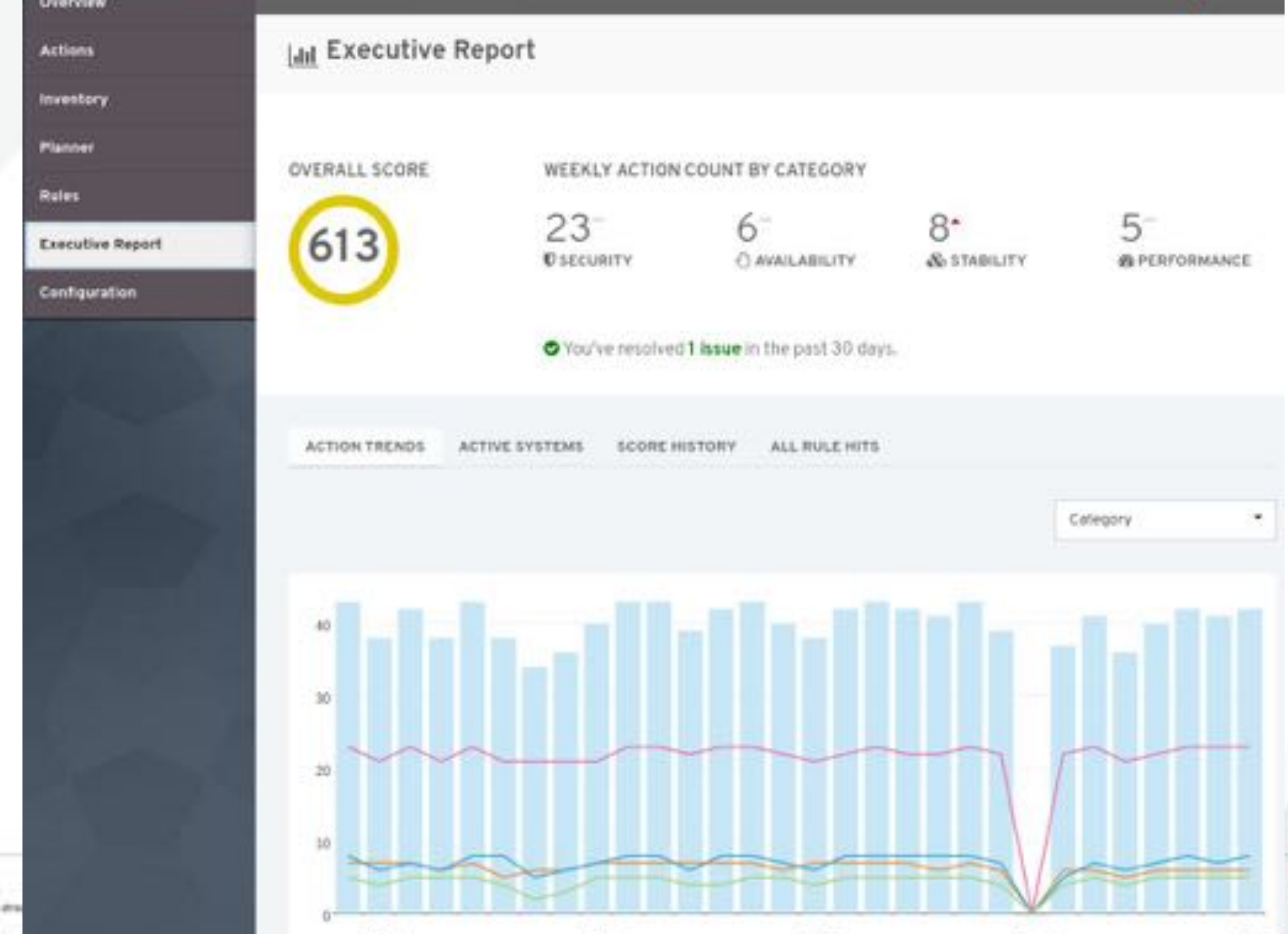
Red Hat CDN

Internet

RED HAT® ANSIBLE® Tower

Predictive Analytics

Robotic Process Automation



Executive Report

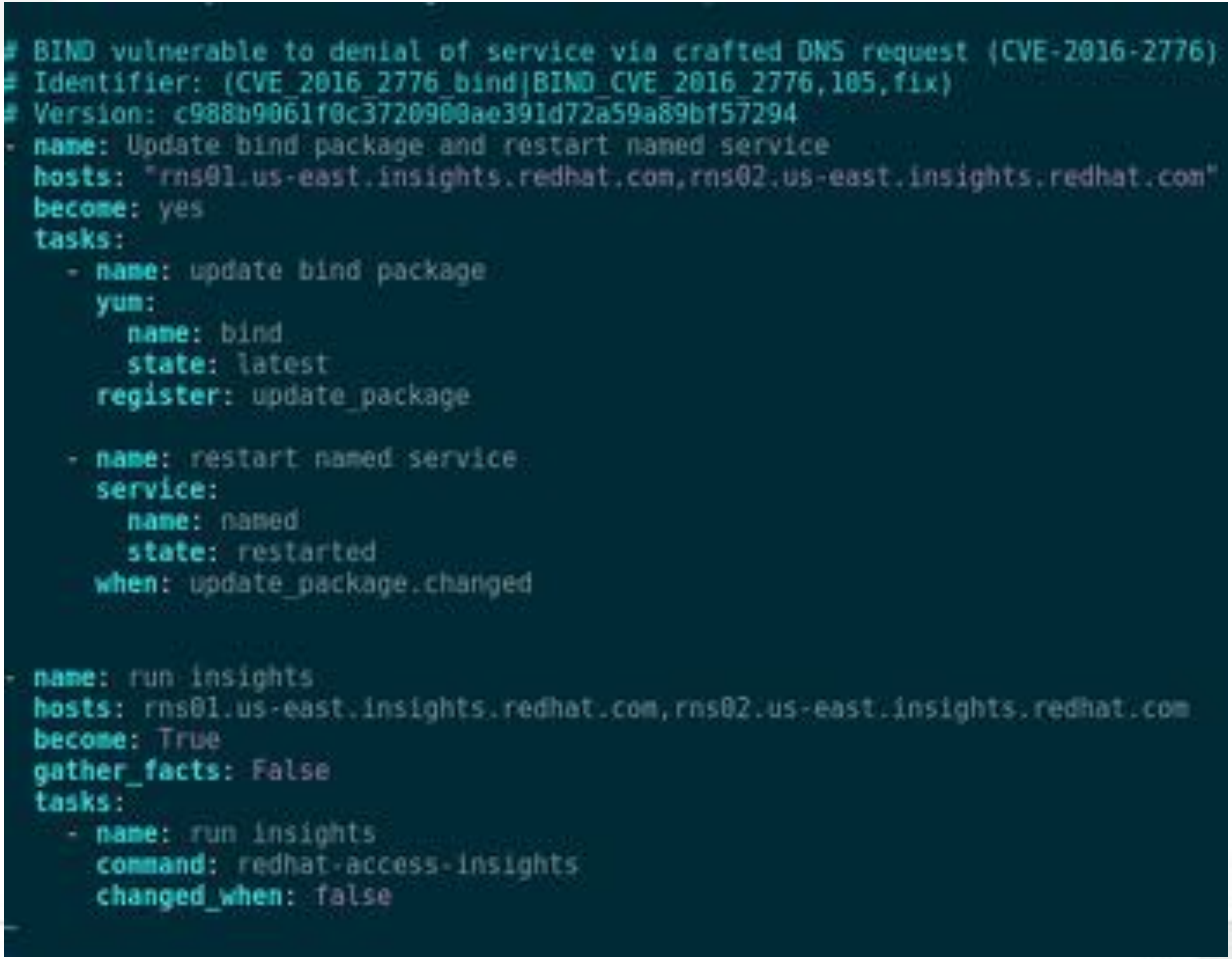
OVERALL SCORE

613

WEEKLY ACTION COUNT BY CATEGORY

23	6	8	5
SECURITY	AVAILABILITY	STABILITY	PERFORMANCE

✓ You've resolved **1 issue** in the past 30 days.



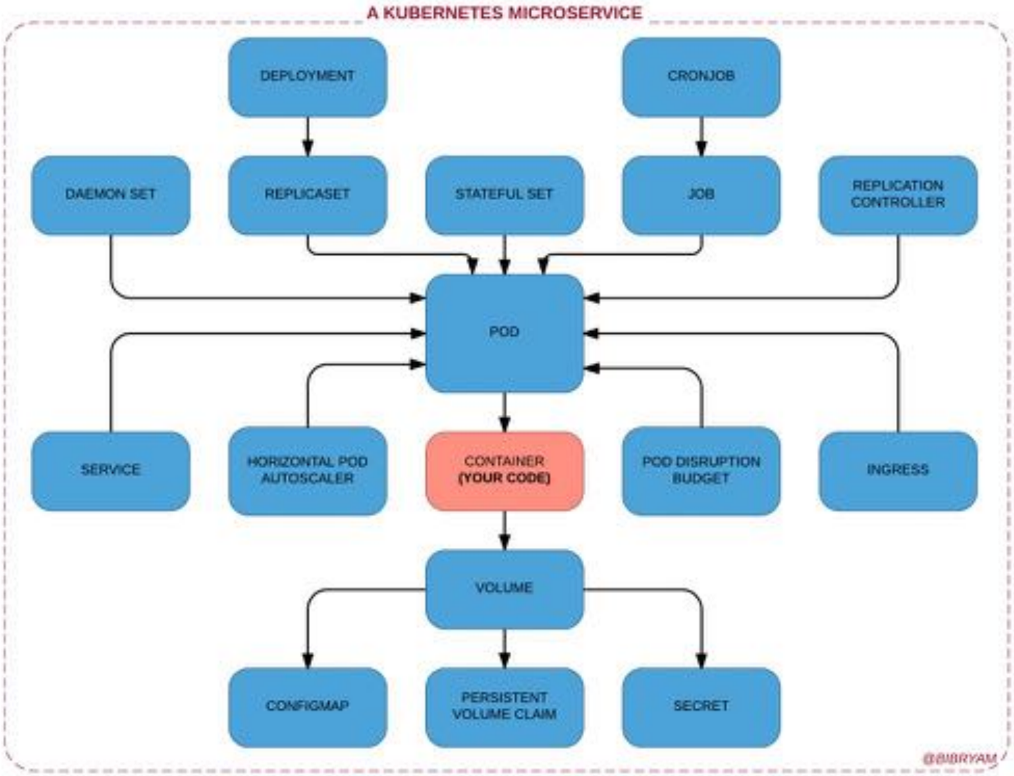
```
# BIND vulnerable to denial of service via crafted DNS request (CVE-2016-2776)
# Identifier: (CVE_2016_2776_bind|BIND_CVE_2016_2776,105,fix)
# Version: c988b9061f0c3720900ae391d72a59a89bf57294
- name: Update bind package and restart named service
  hosts: "rns01.us-east.insights.redhat.com,rns02.us-east.insights.redhat.com"
  become: yes
  tasks:
    - name: update bind package
      yum:
        name: bind
        state: latest
      register: update_package

    - name: restart named service
      service:
        name: named
        state: restarted
      when: update_package.changed


- name: run insights
  hosts: rns01.us-east.insights.redhat.com,rns02.us-east.insights.redhat.com
  become: True
  gather_facts: False
  tasks:
    - name: run insights
      command: redhat-access-insights
      changed_when: false
```

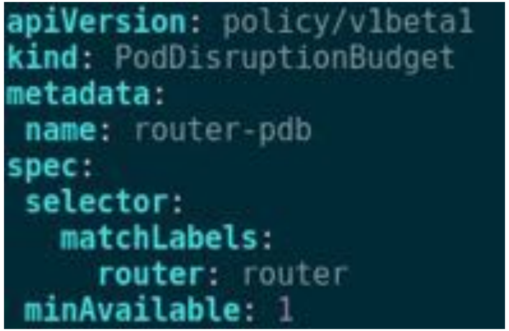
Putting it all together

Steps

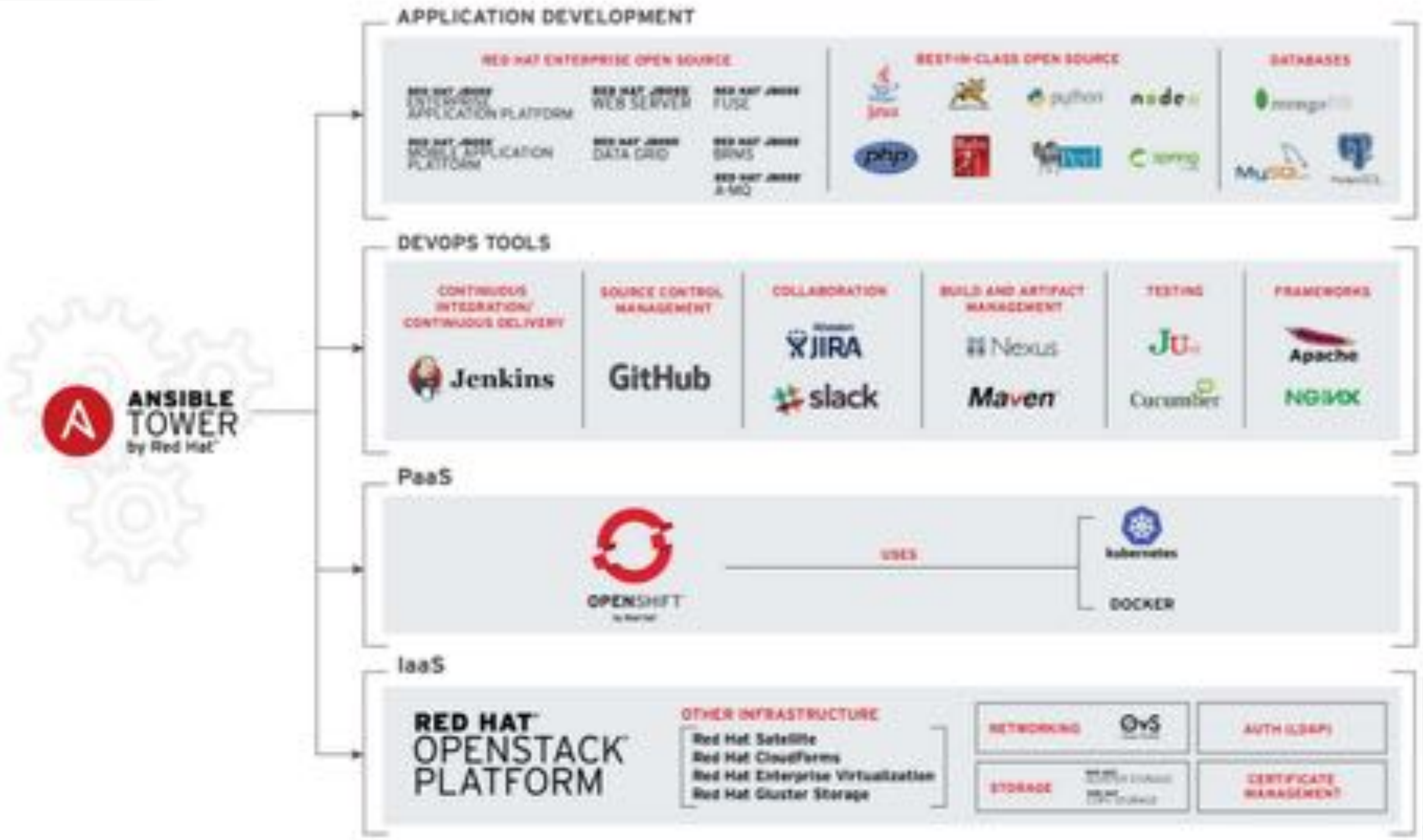
1. Do some realistic planning about your target availability for systems. 100% is not realistic. Consider your “Disruption Budget”
2. Work out your strategy; Suggestion would be Rotate, Repave and Repair
3. Select your tooling…. As this is at the Red Hat Forum it will be Ansible + Satellite + Insights!
4. Work out your CI/CD pipeline and what needs to be centralised.
5. Plan Metrics and Logging as the first thing to get working.
6. Automate everything. Remember Logs are for root cause analysis. Alerts and tickets should be what you use to run a system.
7. Leverage Predictive Analytics.

Compare to OpenShift





```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
 name: router-pdb
spec:
 selector:
   matchLabels:
     router: router
 minAvailable: 1
```



ANSIBLE TOWER by Red Hat

APPLICATION DEVELOPMENT

RED HAT ENTERPRISE OPEN SOURCE

- RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM
- RED HAT JBOSS WEB SERVER
- RED HAT JBOSS FUSE
- RED HAT JBOSS MOBILE APPLICATION PLATFORM
- RED HAT JBOSS DATA GRID
- RED HAT JBOSS BRMS
- RED HAT JBOSS A-MQ

BEST-IN-CLASS OPEN SOURCE

DATABASES

DEVOPS TOOLS

CONTINUOUS INTEGRATION/ CONTINUOUS DELIVERY	SOURCE CONTROL MANAGEMENT	COLLABORATION	BUILD AND ARTIFACT MANAGEMENT	TESTING	FRAMEWORKS
Jenkins	GitHub	JIRA slack	Nexus Maven	JUnit Cucumber	Apache NGINX

PaaS

OPENSHIFT by Red Hat — USES — kubernetes, DOCKER

IaaS

RED HAT OPENSTACK PLATFORM

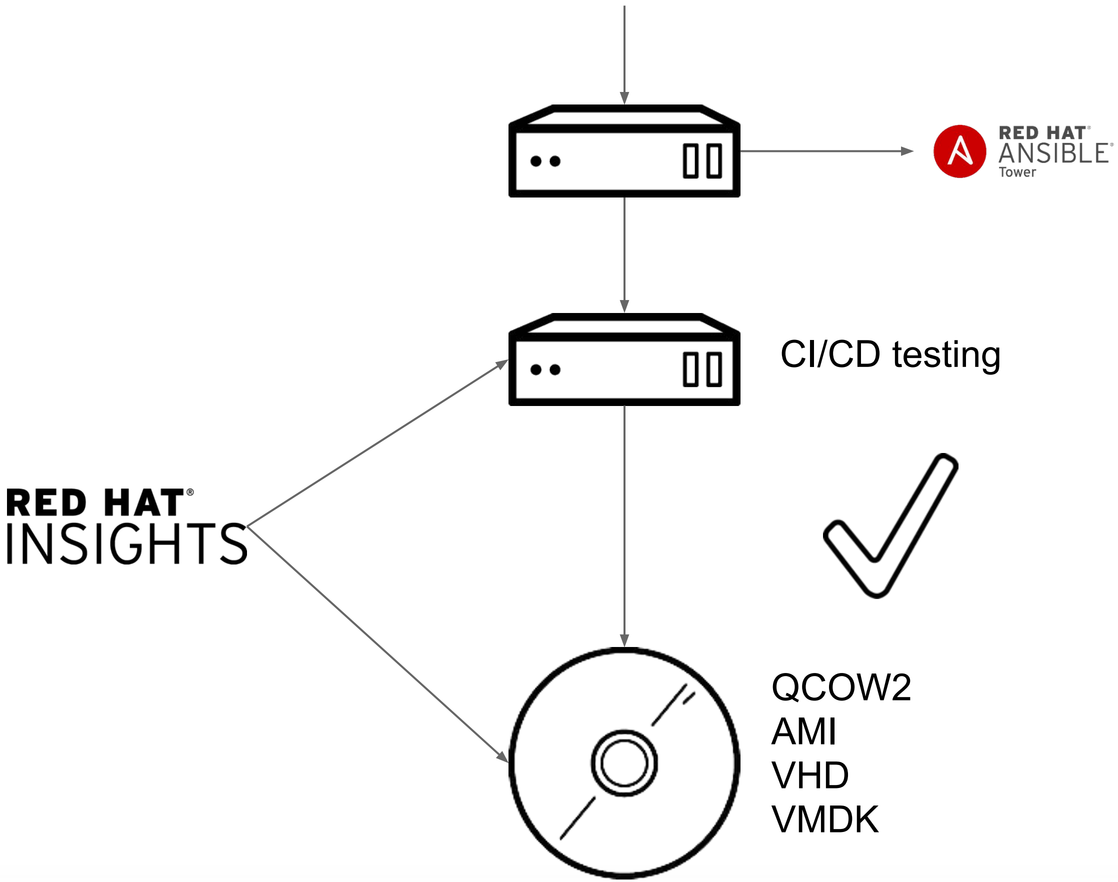
OTHER INFRASTRUCTURE

- Red Hat Satellite
- Red Hat CloudForms
- Red Hat Enterprise Virtualization
- Red Hat Gluster Storage

NETWORKING OvS	AUTH (LDAP)
STORAGE RED HAT GLUSTER STORAGE RED HAT CEPH STORAGE	CERTIFICATE MANAGEMENT

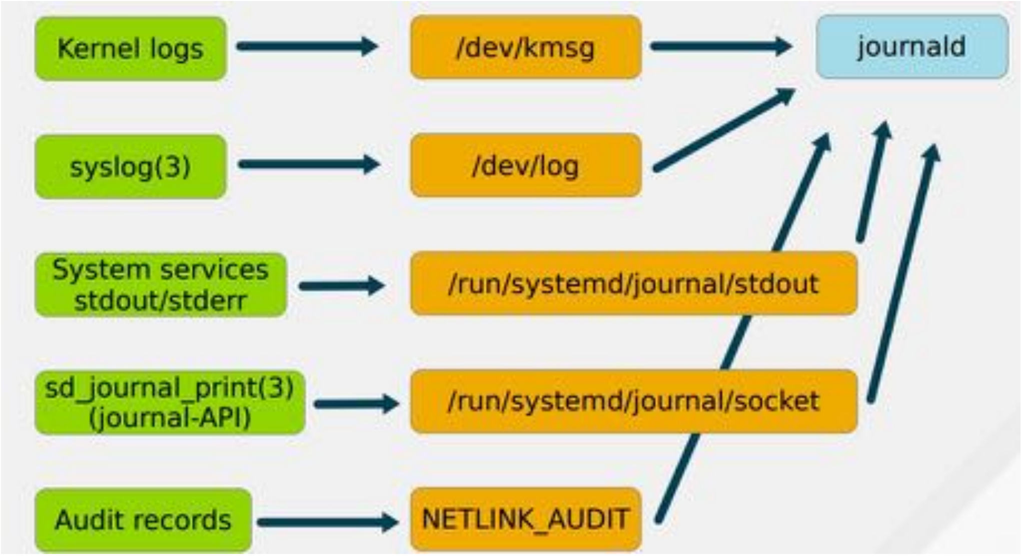
RED HAT® INSIGHTS

RED HAT® ANSIBLE® Tower



CI/CD testing

QCOW2
AMI
VHD
VMDK



Kernel logs → /dev/kmsg → journald

syslog(3) → /dev/log → journald

System services stdout/stderr → /run/systemd/journal/stdout → journald

sd_journal_print(3) (journal-API) → /run/systemd/journal/socket → journald

Audit records → NETLINK_AUDIT → journald

Here is the critical bit....

Remember - nobody should be looking at raw logs and raw metrics

- We also need a threshold and alerting engine - Metrics as a Service
- We also need a stream processing enging - Logging as a Service

Automate Everything

Photo via Visual Hunt

https://developers.redhat.com/blog/2016/10/24/how-we-automate-everything-at-red-hat-open-innovation-labs/

http://www.opensourcerers.org/learning-by-prototype-bringing-ansible-tower-openshift-cloudforms-and-insights-together/

https://access.redhat.com/articles/3119481

https://developers.redhat.com/blog/2016/07/07/carving-the-java-ee-monolith-into-microservices-prefer-verticals-not-layers/

https://builttoadapt.io/the-three-r-s-of-enterprise-security-rotate-repave-and-repair-f64f6d6ba29d

http://www.ofbizian.com/2017/05/bet-on-cloud-native-ecosystem.html

Q&A

RED HAT

FORUM

Europe, Middle East & Africa