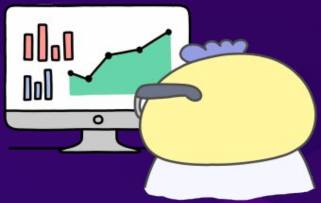
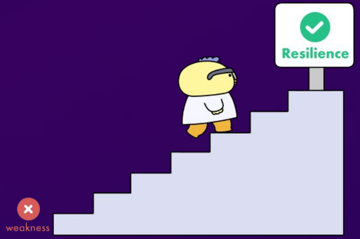
Chaos is a Feature, not a Bug!

Dave Nielsen, Sr. Director of Community & Advocacy, Harness
Open Source projects such as LitmusChaos & Drone CI





Terms & Topics Covered in this talk

- Reliability, Resilience
- Chaos Engineering, Fault Injection
- LitmusChaos
- GameDays/ChaosDays

Reliability: Putting the R in SRE

Creating large-scale, distributed applications have never been so easy

- Infrastructure is hosted in the cloud
- Programming language support is diverse
- Lots of open source components
- Lots of cloud services to build upon

What is Resilience?

Resiliency is the ability to withstand certain types of failure and yet remain functional

You cannot be resilient if the components that make up your service are not reliable.

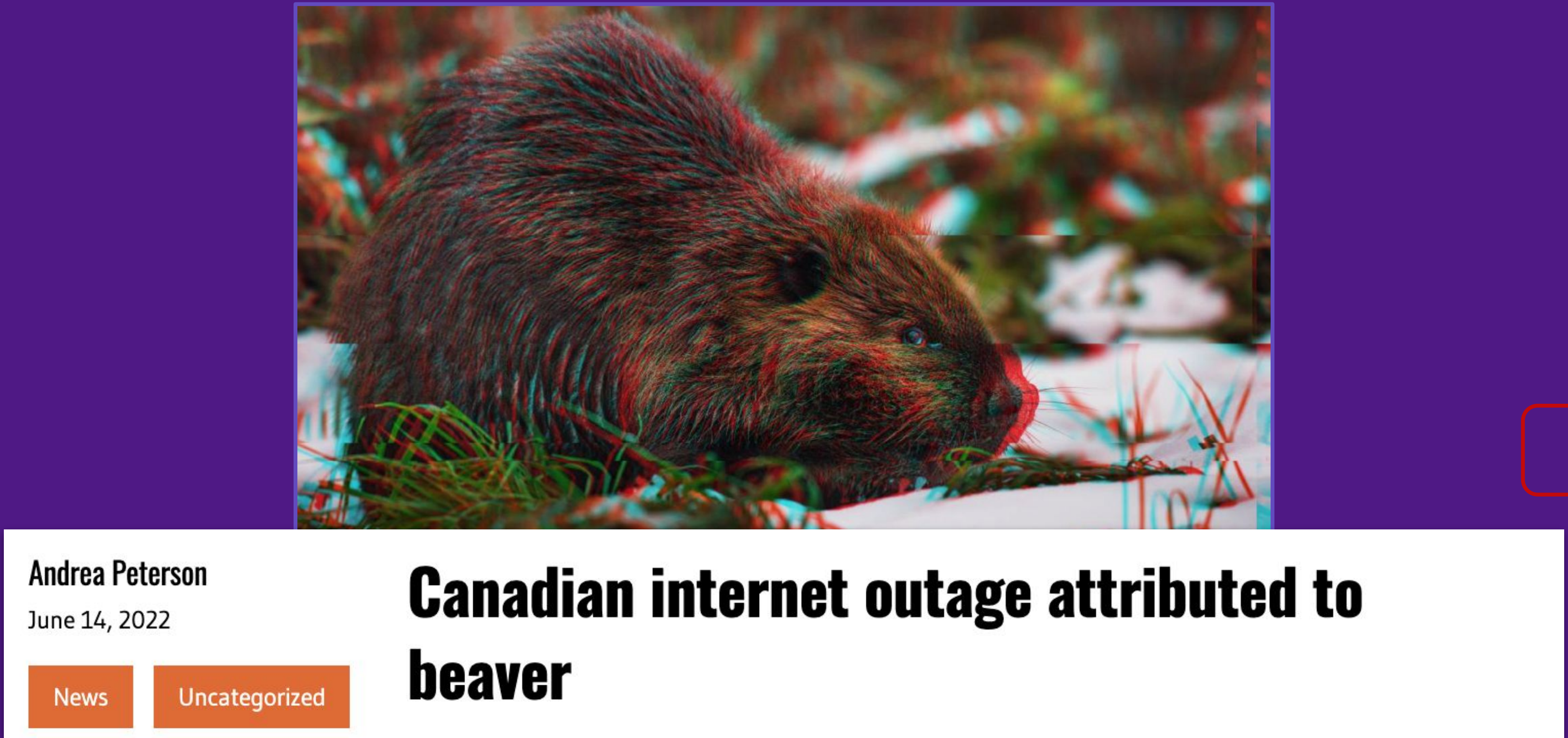
Dave Nielsen

Sr. Director of Community & Advocacy at Harness

@davenielsen
@harnessio

Harness = Modern Software Delivery Platform

- DroneCI
- LitmusChaos

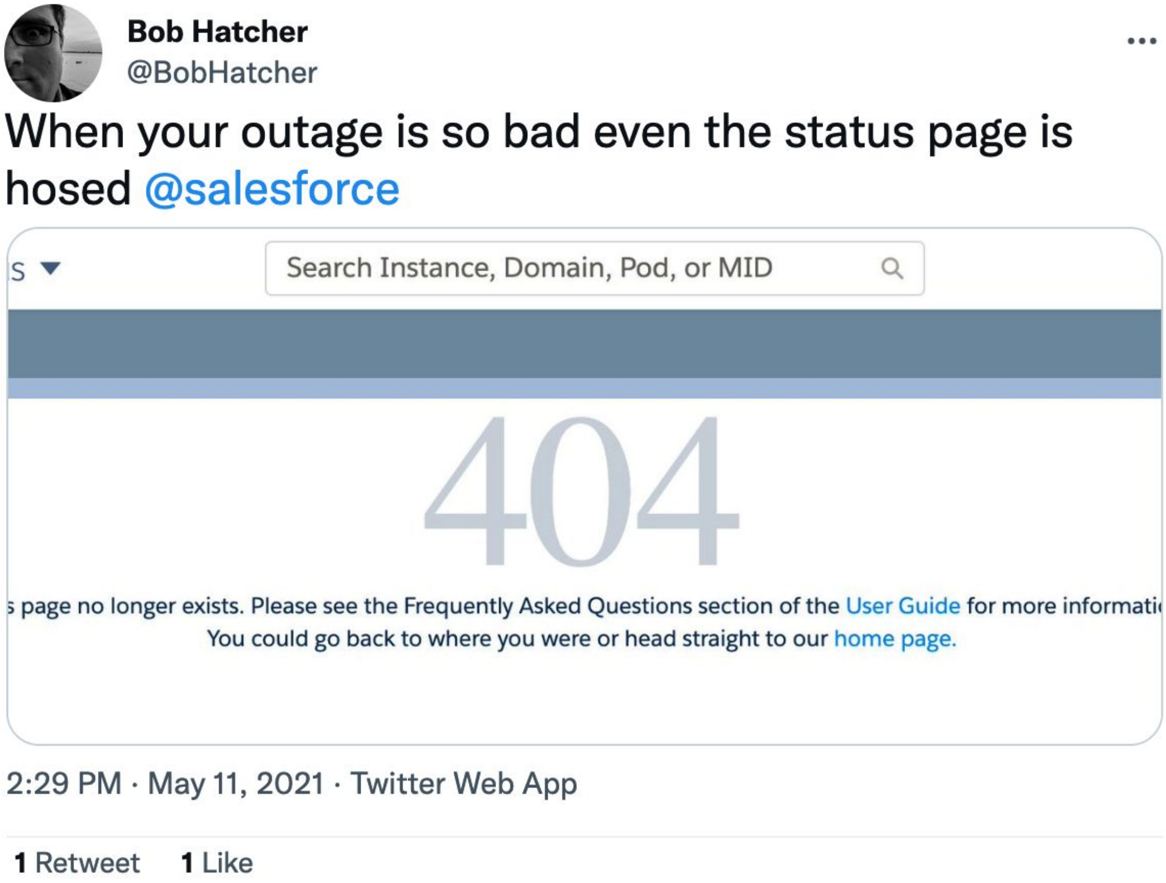


Andrea Peterson

June 14, 2022

News Uncategorized

Canadian internet outage attributed to beaver



Bob Hatcher
@BobHatcher

When your outage is so bad even the status page is hosed @salesforce

s page no longer exists. Please see the Frequently Asked Questions section of the User Guide for more informati
You could go back to where you were or head straight to our home page.

2:29 PM · May 11, 2021 · Twitter Web App

1 Retweet **1** Like

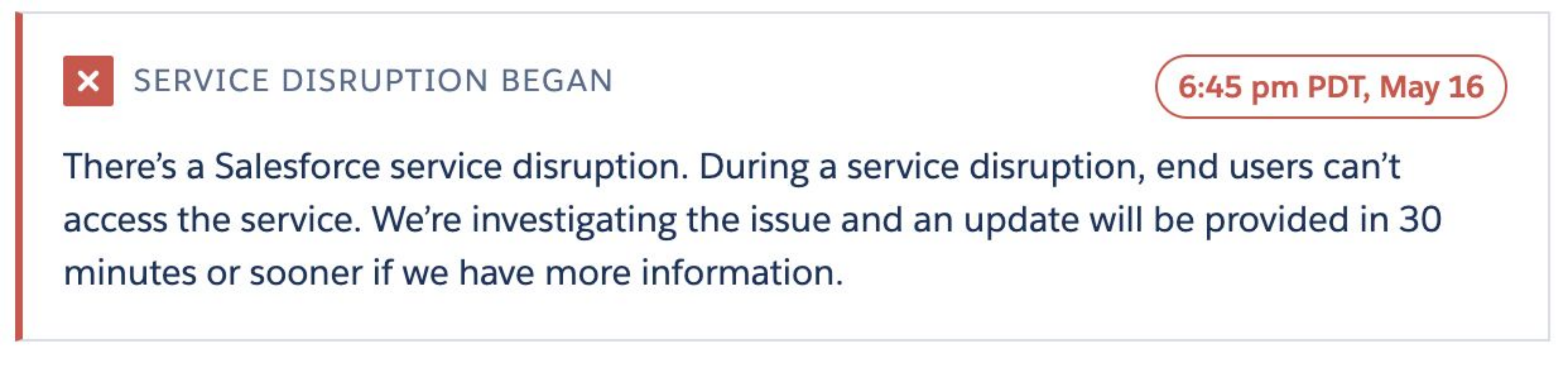
https://twitter.com/BobHatcher/status/1392230493844815873

salesforce

Faulty database script granted permissions to everyone.

Had to shut down access to its own services.

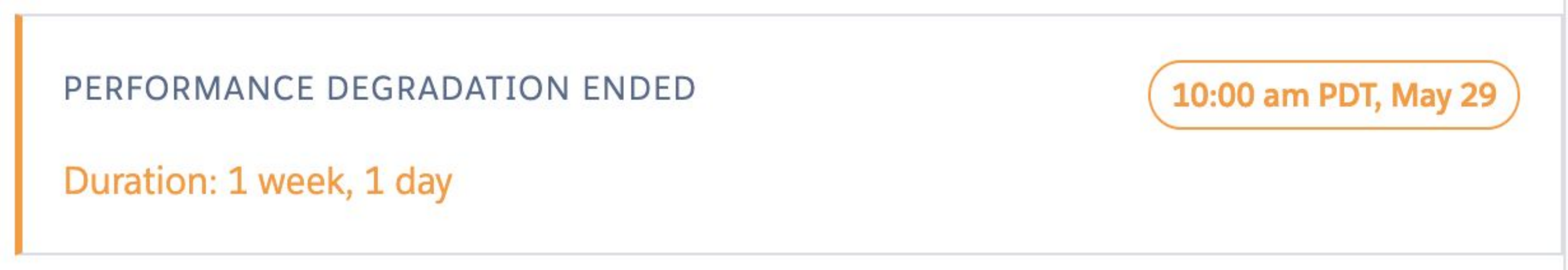
Duration: 1 week, 1 day!



SERVICE DISRUPTION BEGAN

6:45 pm PDT, May 16

There's a Salesforce service disruption. During a service disruption, end users can't access the service. We're investigating the issue and an update will be provided in 30 minutes or sooner if we have more information.



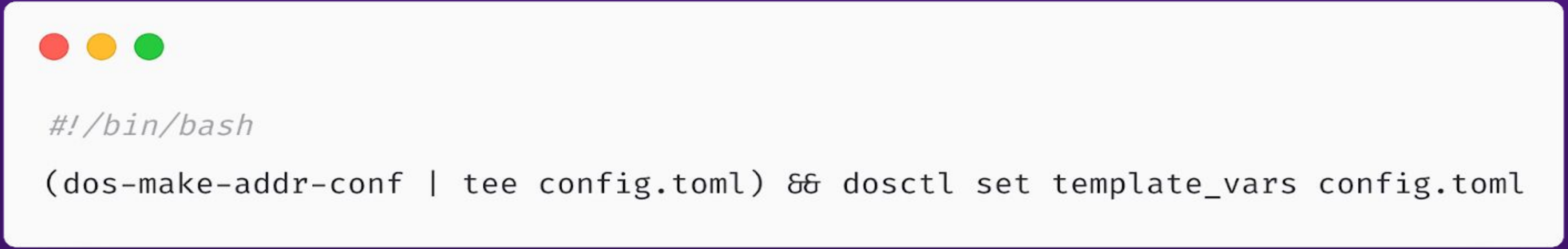
PERFORMANCE DEGRADATION ENDED

10:00 am PDT, May 29

Duration: 1 week, 1 day

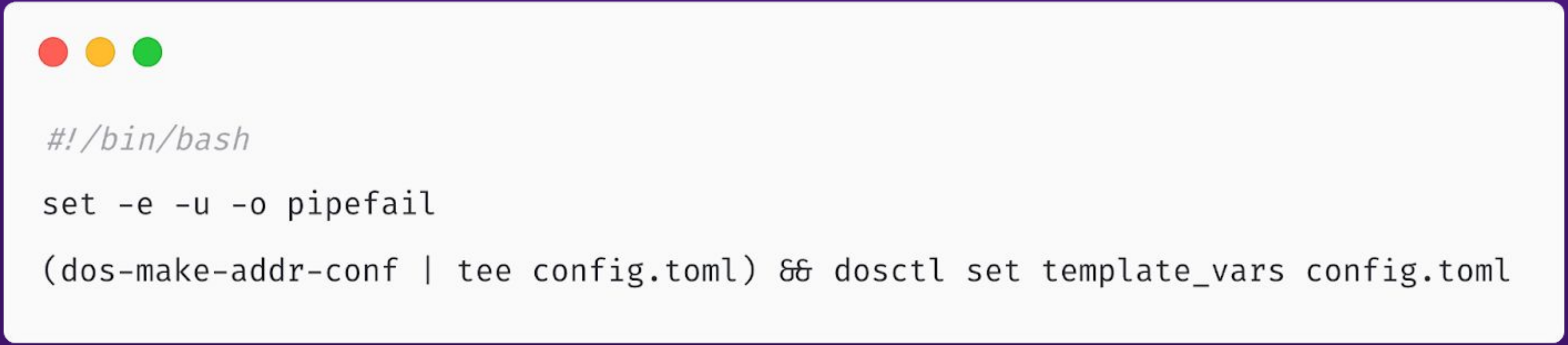
https://status.salesforce.com/incidents/3815

Original version

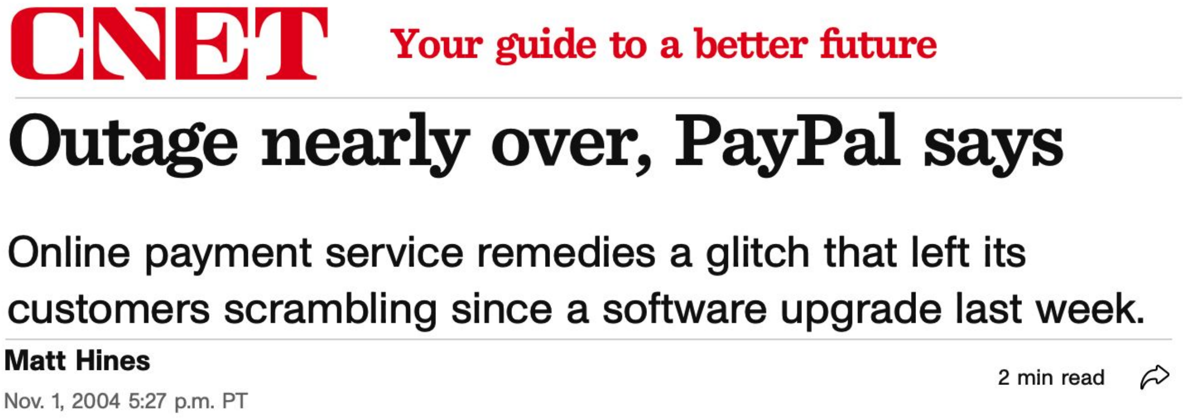


```
#!/bin/bash
(dos-make-addr-conf | tee config.toml) && dosctl set template_vars config.toml
```

Fixed version with pipefail enabled



```
#!/bin/bash
set -e -u -o pipefail
(dos-make-addr-conf | tee config.toml) && dosctl set template_vars config.toml
```



CNET Your guide to a better future

Outage nearly over, PayPal says

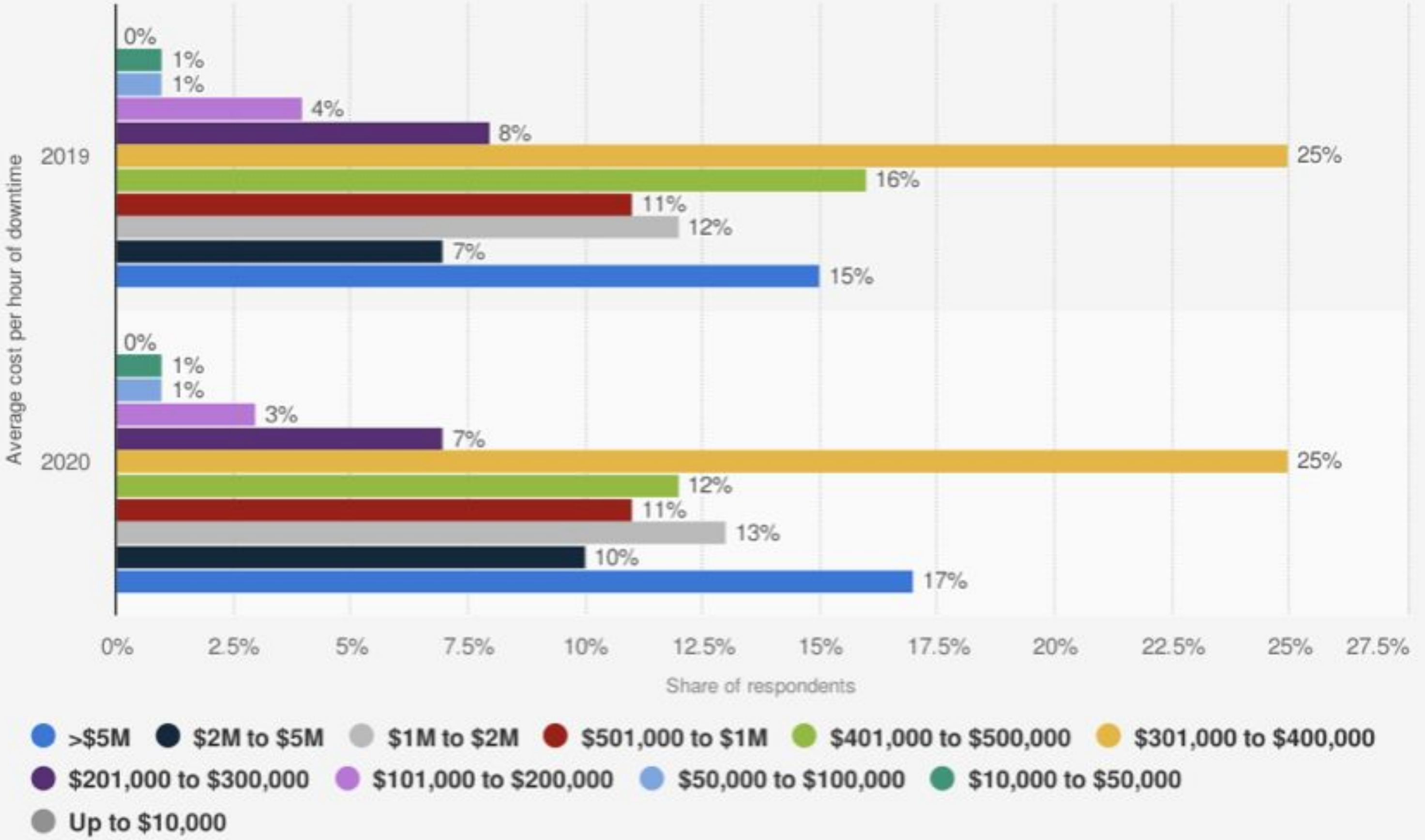
Online payment service remedies a glitch that left its customers scrambling since a software upgrade last week.

Matt Hines

Nov. 1, 2004 5:27 p.m. PT

2 min read

Average cost per hour of enterprise server downtime worldwide in 2019



Common Real World Examples

- Power failure - “80% of all IT load losses in data centers” over 25 yrs
- Resource exhaustion - Increase in load or database fills up hard drive
- Hardware failures - network adaptor, hard drive
- Network latency & partition
- Sudden increase (or decrease) in input, such as retry storms
- Downstream dependencies malfunction
- Functional bugs
- Race conditions
- State transmission errors (inconsistency of states between sender & receiver nodes)
- Unusual combinations of interservice communication
- “Dunning-Kruger” problem - A node believes it has the correct data

What is Chaos Engineering?

Chaos Engineering:

The practice of subjecting applications & services to real world stresses & failures in order to build & validate resilience to unreliable conditions and missing dependencies.

Fault Injection:

The act of introducing an error to a system. Different faults, such as network latency or loss of access to storage, can be used to target system components, causing scenarios that an application or service must be able

to handle or recover from.

History: 12+ Years of Chaos

- **2004 Amazon - Jessie Robbins - The Master of Disaster**
- **2010 Netflix - Greg Orzell - First implementation of Chaos Monkey**
- **2012 Netflix open sources Simian Army**
- **2014 Netflix coins term “Chaos Engineer”**
- **2020 Reliability pillar added to AWS Well-Architected Framework**
- **2020 LitmusChaos donated to CNCF by ChaosNative (Harness)**
- **2022 CNCF Takes LitmusChaos to the Incubation Level**

Not just Talk! Not just chapters!

O'REILLY®

Chaos Engineering

Building Confidence in System Behavior through Experiments

Compliments of NETFLIX

hal, Lor

viak, No

CHAOS ENGINEERING

Chaos Engineering

Site reliability through controlled disruption

kowski

O'REILLY®

Chaos Engineering

Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones & Ali Basiri

REPORT

O'REILLY®

Learning Chaos Engineering

Discovering and Overcoming System Weaknesses through Experimentation

Russ Miles

O'REILLY®

Chaos Engineering

System Resiliency in Practice

Casey Rosenthal & Nora Jones

THE SCIENCE OF DEVOPS

ACCELERATE

Building and Scaling High Performing Technology Organizations

Nicole Forsgren, PhD

Jez Humble and Gene Kim

AWS Well Architected Framework

- Operational Excellence Pillar
- Security Pillar
- Reliability Pillar
- Performance Efficiency Pillar
- Cost Optimization Pillar
- Sustainability Pillar

https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html

Why are we seeing this now?

A focus on Reliability. Now it's intentional, not a nice to have.

Reliability of sites during COVID became apparent.
Remember trying to find a place to get a covid test?

Just as K8s makes it easier for microservices, it also makes it easier to deploy entire copies of your infa.

Dev**Reliability**Ops?

We’re thinking about reliability from the beginning. Sort of like Security with DevSecOps. But DevReliabilityOps. However, really, it’s all just DevOps.

It’s important for our team culture & teamwork, not just customers

LitmusChaos

“

An Open Source Cloud-Native Chaos Engineering Framework with cross-cloud support. It is a CNCF Incubating project with growing adoption

CNCF Incubating project

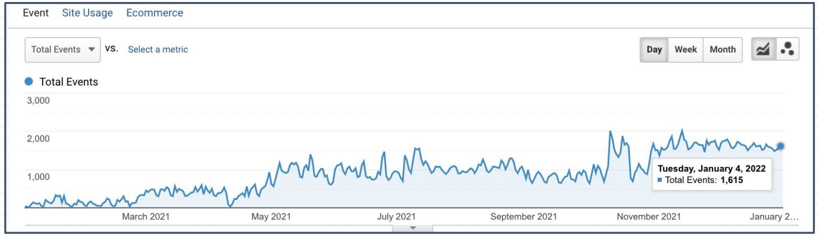
Project

LitmusChaos - a CNCF incubating project

Litmus is an open source platform for practicing chaos engineering in a cloud native way.

- Started in 2017; 4+ years of active development
- 500k+ Litmus installations; 30x usage growth in the last 3 quarters, 55+ chaos experiments, 100+ contributors
- Stable platform : 2.11 released
  50+ enterprises using 2.0
- **CNCF Incubating project**

30x growth in per-day installations of Litmus in the last 3 quarters; 1500 installations per day

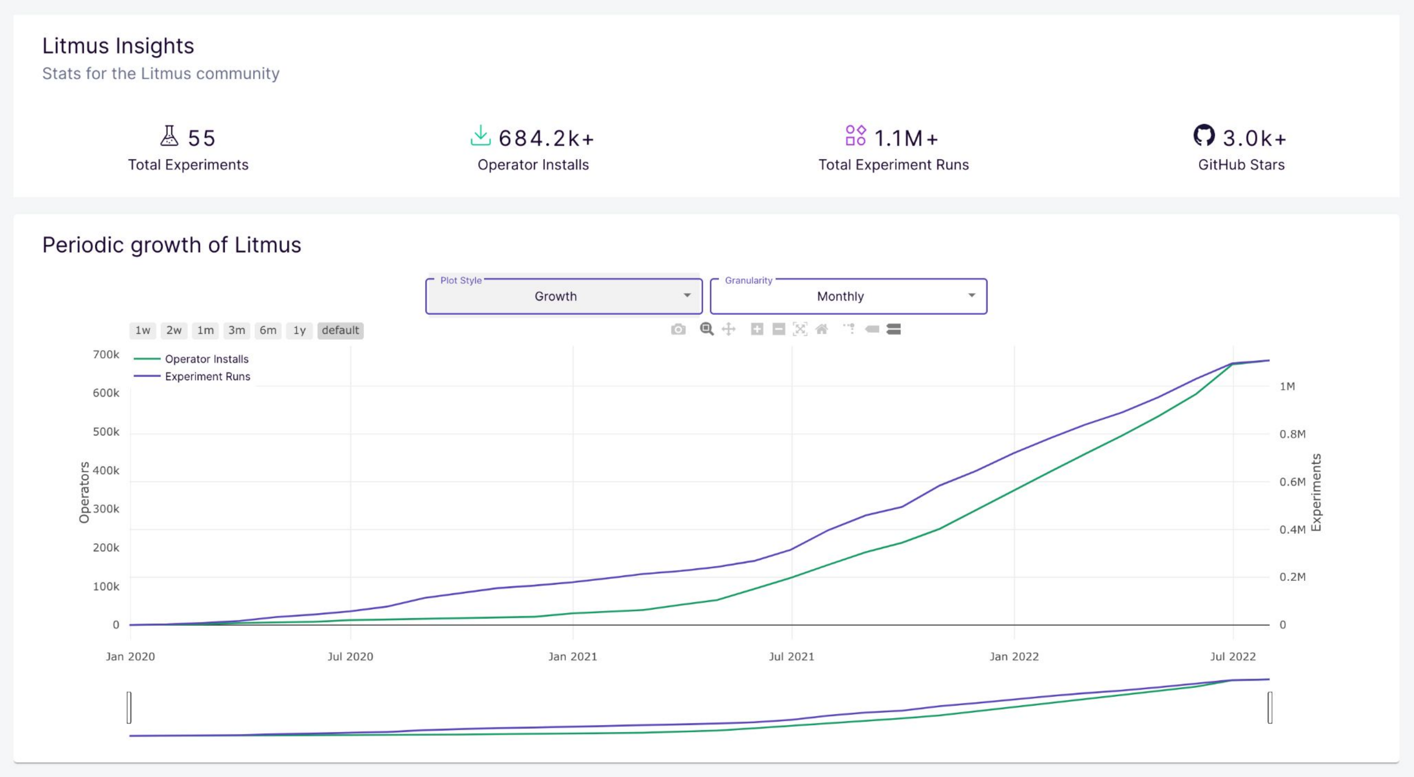


Litmus is **adopted** by

intuit, okteto, NetApp, Red Hat, wescale, kublr
lenskart.com, halodoc, keptn, vmware, KITOPI, anuta networks
wipro, ZEBRIUM, orange, OpenEBS, Pravega
KUBESPHERE, KLANIK, NEUDESIC, ifood, Pôle emploi.fr, Container Solutions

LitmusChaos Project Status

Why Resilience? | Achieving Resilience | Litmus 101 | Litmus Experiments | Contributing



Litmus Insights

Stats for the Litmus community

55	684.2k+	1.1M+	3.0k+
Total Experiments	Operator Installs	Total Experiment Runs	GitHub Stars

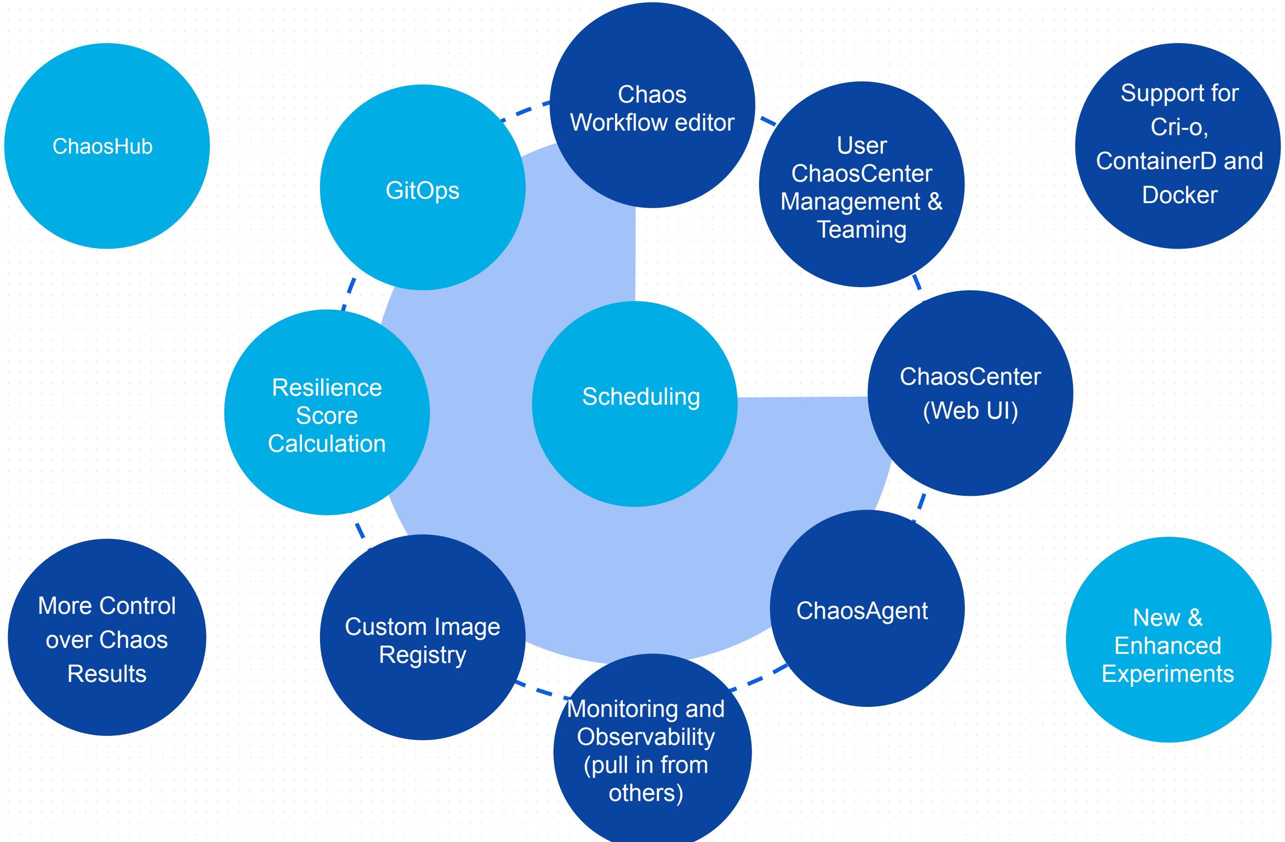
Periodic growth of Litmus

Plot Style: Growth

Granularity: Monthly

What's Exciting about *LitmusChaos?*

Why Resilience? › Achieving Resilience › Litmus 101 › Litmus Experiments › Contributing



ChaosHub

GitOps

Chaos Workflow editor

User ChaosCenter Management & Teaming

Support for Cri-o, ContainerD and Docker

Resilience Score Calculation

Scheduling

ChaosCenter (Web UI)

More Control over Chaos Results

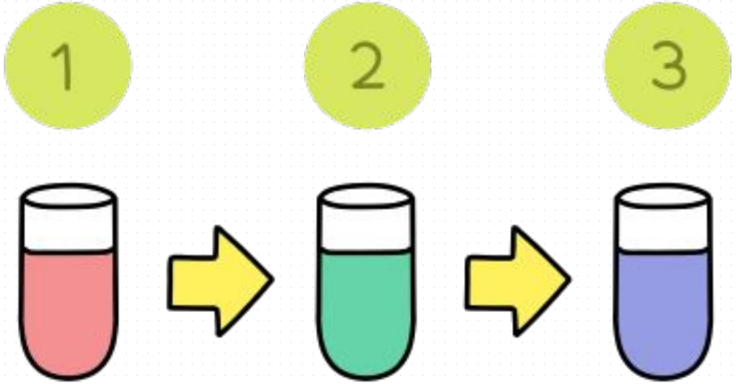
Custom Image Registry

ChaosAgent

New & Enhanced Experiments

Monitoring and Observability (pull in from others)

What are some **key Resilience practices**?



Key Practices

01

Chaos Engineering

02

GameDays

“

Build systems that are designed to be deployed easily, can **detect and tolerate failures**, and can have various components of the system updated independently.”

- ***Accelerate***

Dave Mangot
@davemangot

Protip: Every time someone explains to you that DevOps is just CI/CD, Platform engineering (a vendor said this, shocking!), etc. ask where gamedays fit in.

@realgenekim's 3rd Way "introducing faults to the system to increase resilience" is missing.

Welcome to partial DevOps. 😳

What is the best way to understand your systems?

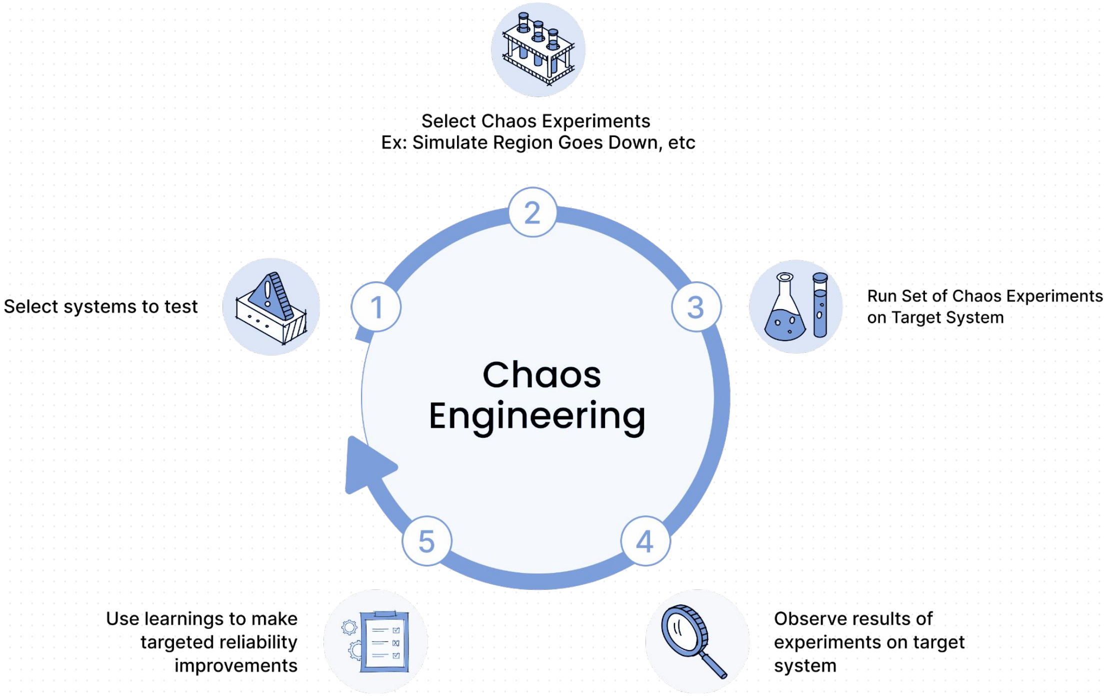
Science
AKA Chaos Engineering!

Chaos Engineering is a feature, not a bug.
IOW, you plan to find out where our systems have weakness. Not wait to find out!

Chaos Engineering is: the discipline of experimenting on a distributed system in order to build confidence in the system's capacity to withstand turbulent conditions in production - Principles of Chaos Engineering

- Carefully applying failures with an explicit hypothesis
- Starting small and growing the blast radius
- Clearly communicating with all stakeholders
- A well defined process that requires regular practice

Again ...



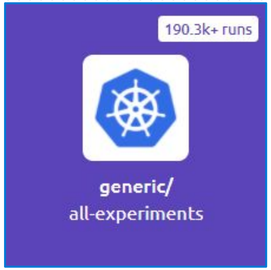
Select Chaos Experiments
Ex: Simulate Region Goes Down, etc
2
Select systems to test
1
3
Run Set of Chaos Experiments
on Target System
Chaos
Engineering
5
4
Use learnings to make
targeted reliability
improvements
Observe results of
experiments on target
system

HOW

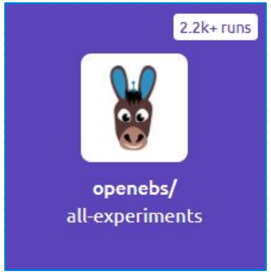
to Chaos Engineer?

Variety of experiments offered in *LitmusChaos 2.0*

Pod Chaos	Node Chaos	Network Chaos	Stress Chaos	Cloud Services	Application Chaos
Pod Failure Container Kill Pod Autoscale	Node Drain Forced Eviction (Node Taints) Node Restart/PowerOff	Network Latency Packet Loss Network Corruption, Duplication	Pod, Node CPU Hog Pod, Node Memory Hog Pod, Node Disk Stress Pod Ephemeral Storage Fill	AWS EKS EC2 Termination AWS EBS Disk Detach GCP GPD Disk Detach	Kafka Leader Broker Failure Cassandra Ring Disruption OpenEBS Control Plane / Volume Failure


190.3k+ runs

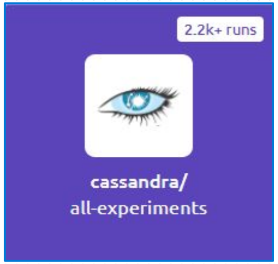
generic/
all-experiments


2.2k+ runs

openebs/
all-experiments

2.1k+ runs

kafka/
all-experiments


2.2k+ runs

cassandra/
all-experiments

159 runs

kube-aws/
all-experiments

How to *Contribute?*

We welcome contributions of all kinds

- Development of features, bug fixes, and other improvements.
- Documentation including reference material and examples.
- Bug and feature reports

You can choose from a list of sub-dependent repos to contribute to, a few highlighted repos that Litmus uses are:

- Chaos-charts
- Chaos-workflows
- Test-tools
- Litmus UI
- Litmus-go
- website-litmuschaos

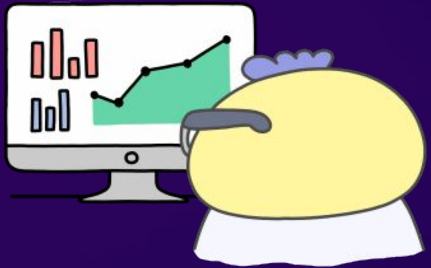
Key Practice #2:
GameDays

What is a GameDay?

A day dedicated to getting your team together, designing some experiments (tests), running *them in a controlled* chaos engineering environments (including prod) with the goal of proactively evaluating your systems to see if they are resilient to various kinds of failures.

What is an Example GameDay?

- Invite at least 1 team to participate
- Trigger a Pod Chaos Experiment
  - Pod Failure or Container Kill
  - Confirm that your application autoscales
- Trigger a Cloud Services
  - AWS EKS EC2 Termination
- Other examples:
  - Dependency Testing
  - Capacity Plan Testing



Karthik’s Suggestions (co-creator of LitmusChaos)

- Define your blast radius
- Define failures so there is minimal damage.
  - Like surgery. One variable at a time, one failure at a time
  - Also like the Scientific Method
- Simulate as many real-world problems as possible. Here are some:
  - Resources become slow
  - Noisy neighbor
  - Availability zones may go down
  - Instances may become unresponsive
  - Sudden shutdowns
  - Black Swan events

Karthik’s Suggestions (continued)

- Buildings going down. Regions going down. Ex: AWS says which cities they are in, but Google doesn’t say if in same city, or buildings that are near each other. So you need to be very careful and should deploy cross region.
- Test in prod b/c that’s where the most unknowns are
- Repeat experiments continuously - not 1 in 6 months b/c you don’t know what has changed in-between
- Automate test runs so you can automate as many experiments as possible, without human errors (see “Humans are cause of most outages”)

How to Contribute

litmuschaos.io/community

Litmus channel in K8s Slack

Contact Info

dave.nielsen@harness.io

@davenielsen