Chaos is a Feature, not a Bug!

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

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

Terms & Topics Covered in this talk

- Reliability, Resilience
- Chaos Engineering, Fault Injection
- LitmusChaos
- GameDays/ChaosDays
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

Reliability: Putting the R in SRE
The use of split screens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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
Canadian internet outage attributed to beaver
When your outage is so bad even the status page is hosed @salesforce

Faulty database script granted permissions to everyone.

Had to shut down access to its own services.
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.

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

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

Fixed version with pipefail enabled

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

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

What is Chaos Engineering?
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use them to make a statement.

Confidential / © Harness 2021

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.

Confidential / © Harness 2021
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.
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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!
The use of split screens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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 infra.
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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.”
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

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
- wecscale
- kublr
- lenskart.com
- halodoc
- keptn
- vmware
- ITOPi
- anuta networks
- wipro
- ZEBRIUM
- orange
- OpenEBS
- Pravega
- KUBESPHERE
- Kiwami
- 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 Total Experiments
- 684.2k+ Operator Installs
- 1.1M+ Total Experiment Runs
- 3.0k+ GitHub Stars

Periodic growth of Litmus

Confidential / © Harness Inc. 2021
What’s Exciting about LitmusChaos?

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

- ChaosHub
- GitOps
- Resilience Score Calculation
- Scheduling
- Custom Image Registry
- Monitoring and Observability (pull in from others)
- ChaosAgent
- User ChaosCenter Management & Teaming
- ChaosCenter (Web UI)
- Support for Cri-o, ContainerD and Docker
- More Control over Chaos Results
- New & Enhanced Experiments
- New & Enhanced Experiments
- Support for Cri-o, ContainerD and Docker
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
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!
The use of split screens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

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 ...

Chaos Engineering

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

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that's where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

HOW
to Chaos Engineer?
## Variety of experiments offered in *LitmusChaos 2.0*

<table>
<thead>
<tr>
<th>Why Resilience?</th>
<th>Achieving Resilience</th>
<th>Litmus 101</th>
<th>Litmus Experiments</th>
<th>Contributing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pod Chaos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod Failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container Kill</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod Autoscale</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Node Chaos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Drain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forced Eviction (Node Taints)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Restart/PowerOff</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Network Chaos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Latency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet Loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Corruption, Duplication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stress Chaos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod, Node CPU Hog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod, Node Memory Hog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod, Node Disk Stress</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pod Ephemeral Storage Fill</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cloud Services</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS EKS EC2 Termination</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS EBS Disk Detach</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCP GPD Disk Detach</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Application Chaos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kafka Leader Broker Failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cassandra Ring Disruption</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenEBS Control Plane / Volume Failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Experiment Statistics

- **generic/all-experiments**: 190.3k runs
- **openebs/all-experiments**: 2.2k runs
- **kafka/all-experiments**: 2.1k runs
- **cassandra/all-experiments**: 2.2k runs
- **kube-aws/all-experiments**: 159 runs
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”)
The use of splitscreens

Sometimes you need to draw comparison, or you need a graphic to sit away from the content, that’s where split screens come in. The template includes half and two-thirds split screens. Split screens are great when you want to call specific information out, such as in case studies, or to highlight a quote. Use can use them to make a statement.

How to Contribute
litmuschaos.io/community
Litmus channel in K8s Slack

Contact Info
dave.nielsen@harness.io
@davenielsen