

Gremlin

Setting the Standard for Reliability

Today's call

1. Introductions & objectives
2. Reliability trends & challenges
3. How Gremlin helps
4. Demo
5. Discussion & next steps

Reliability is table stakes

...but it's harder than ever to get it right

Large enterprises are moving to DevOps



Running complex
cloud infrastructure



Automating the
software lifecycle



Trying to catch or
keep pace with cloud
native companies

But DevOps **velocity** is in
conflict with **reliability**

And the
cost of unreliability
has never been higher



Expensive Downtime



Pausing Software
Development



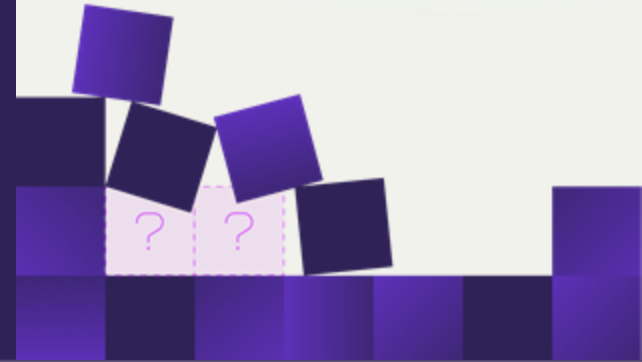
Manual Engineering
Processes

There is **no reliability solution** for modern development

SREs **can't scale** with demand.

Observability **isn't predictive**

Incident response **is too late.**



Reliability needs a strategy



Proactive



Standards-based



Automated



Built-in

Gremlin

How teams proactively
improve reliability at
every stage of maturity.

Shifting from observing to **improving**

Gremlin proactively improves reliability at every stage of maturity.

Experimenting



Custom Chaos Tests & Experiments

Robust, customizable chaos tests to safely replicate any incident scenario.

Standardizing



Standardized Reliability Tests

Pre-built test harness to cover the most common reliability risks.
Get started in minutes.

Scaling



Automated & Scaled Reliability Programs

Standardized scoring and measurement tools to identify and prioritize risks, and automate reliability programs.

Implementing Improvements & Expanding Scope

Demo

How Gremlin helps improve reliability at scale



Understand reliability risks

Identify hidden reliability risks at every level of your technology stack.



Manage to Standards

Discuss outcomes with teams. Identify changes, risks and potential improvements.





Automate and Integrate

Automatically run validations on a schedule to ensure reliability as systems change. Integrate with CI/CD tools to prevent or roll-back unreliable releases.



Trusted by global leaders






TRUIST   JPMORGAN CHASE & CO.  Ameritrade 

Chegg BMO  Harris Bank BARCLAYS DTCC 

Morgan Stanley   Charles SCHWAB  Citizens Financial Group, Inc.

 GRUBHUB NORDSTROM   KOHL'S

 The Washington Post COMCAST  YAHOO! JAPAN

PlayStation.  unity  Adobe  BLIZZARD  TOYOTA connected  Hewlett Packard Enterprise

 Cigna.  CENTENE Corporation Allstate  State Farm® S&P Glo

Gremlin Product Suite

Gremlin enables teams to both standardize reliability at scale and perform custom chaos engineering experiments.

Gremlin Reliability Management (RM) provides the essential elements to start or scale standardized reliability programs.

Gremlin Fault Injection (FI) is a toolset for teams performing custom chaos engineering experiments at the infrastructure layer.

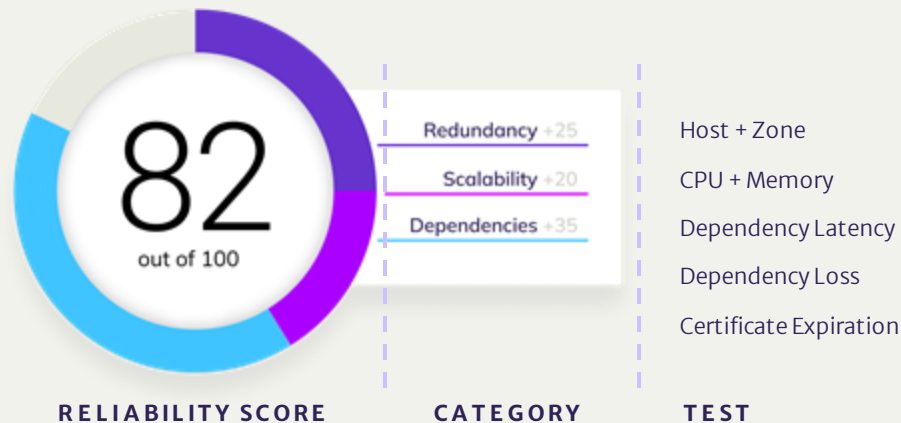
	Gremlin RM	Gremlin FI
	<i>Start or scale reliability programs</i>	<i>Perform custom chaos tests</i>
Reliability Scores & Dashboard	✓	
Reliability Tests	✓	
Dependency Discovery	✓	
Custom Fault Injections		✓
Custom Scenarios		✓
GameDay Manager		✓
API Integration	✓	✓
CI/CD Integrations	✓	✓
Annual Pricing	\$3k/Service	\$50k+

Thank you

The Gremlin Reliability Score

A standards-based, proactive approach to measure reliability.

Scores are created by running pre-defined reliability tests across common reliability risks.



Total **Reliability Score** ranges from 0 to 100

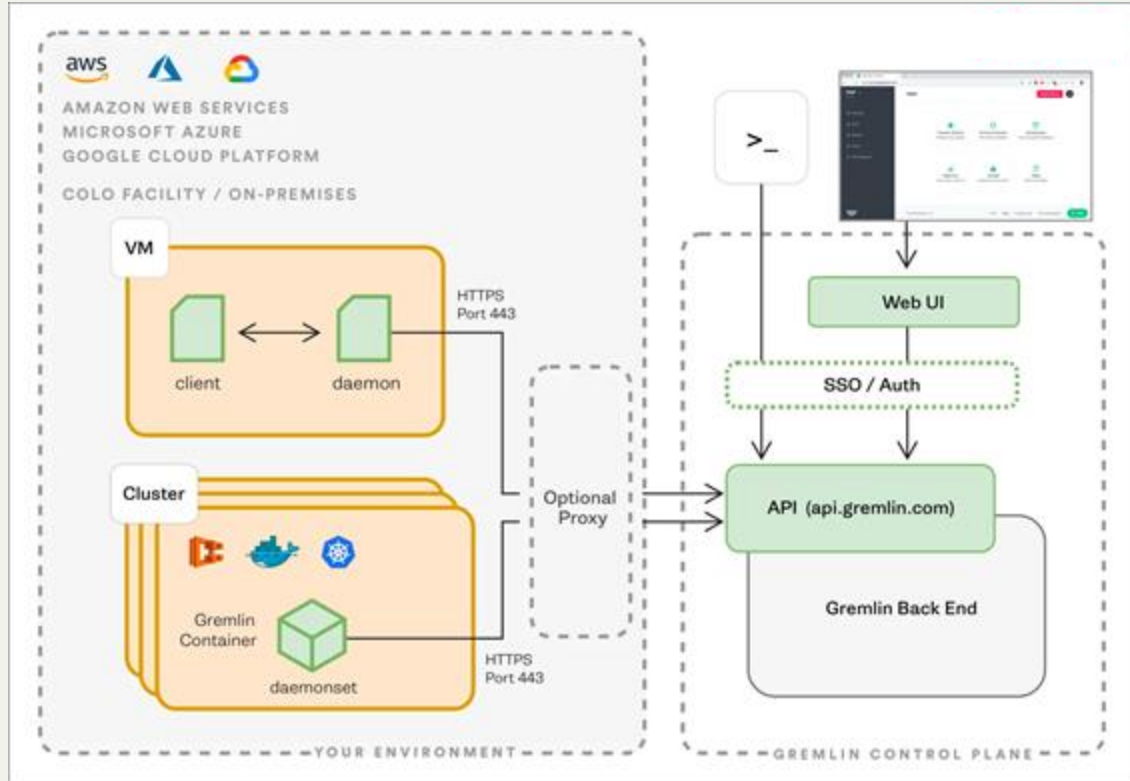
Each **Category** is equally weighted

Each **Test** is worth:

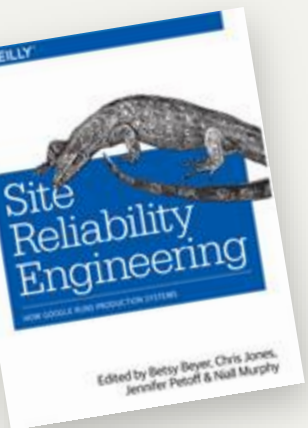
- No points for untested
- Half points for failed test
- Full points for passed test

Test score degrades by 25% if not tested in past seven days

Gremlin Architecture



Health Checks



Gremlin determines if your service passes or fails a reliability test by connecting to your critical service monitors and watching the alert thresholds you've set.

We recommend starting with the four *Golden Signals*: these are the essential indicators of service health for SRE teams.



Latency

The time it takes a system to respond to a request



Traffic

How much your service is in demand



Errors

The rate of requests that fail explicitly, implicitly, or by policy



Saturation

How much memory or CPU resources your system is utilizing

Integrates with



Service Identification

A service is the specific functionality provided by one or more systems within an environment, such as a checkout service or authentication service. Services have clear interfaces and can be independently deployed. For Gremlin, it's the unit of reliability measurement and improvement.

By making each service more reliable, you make the entire application more reliable.

In Gremlin, you define the service by identifying a process running on one or more hosts, containers, or Kubernetes resources.

Examples:

- Java application running simultaneously on two servers
- with a load balancer directing traffic between them
- Kubernetes deployment running two instances of the
- same container

Gremlin is Enterprise-ready



Simple

Intuitive interface and well-documented API



Safe

Safely halt and roll back any validation



Secure

SOC II Certified
RBAC, MFA, SSO



Cloud Native

Runs in all environments: AWS, Azure, GCP, bare metal



Guided

Recommended workflows and validations



Breadth

Supports Linux, Windows, Kubernetes, and more