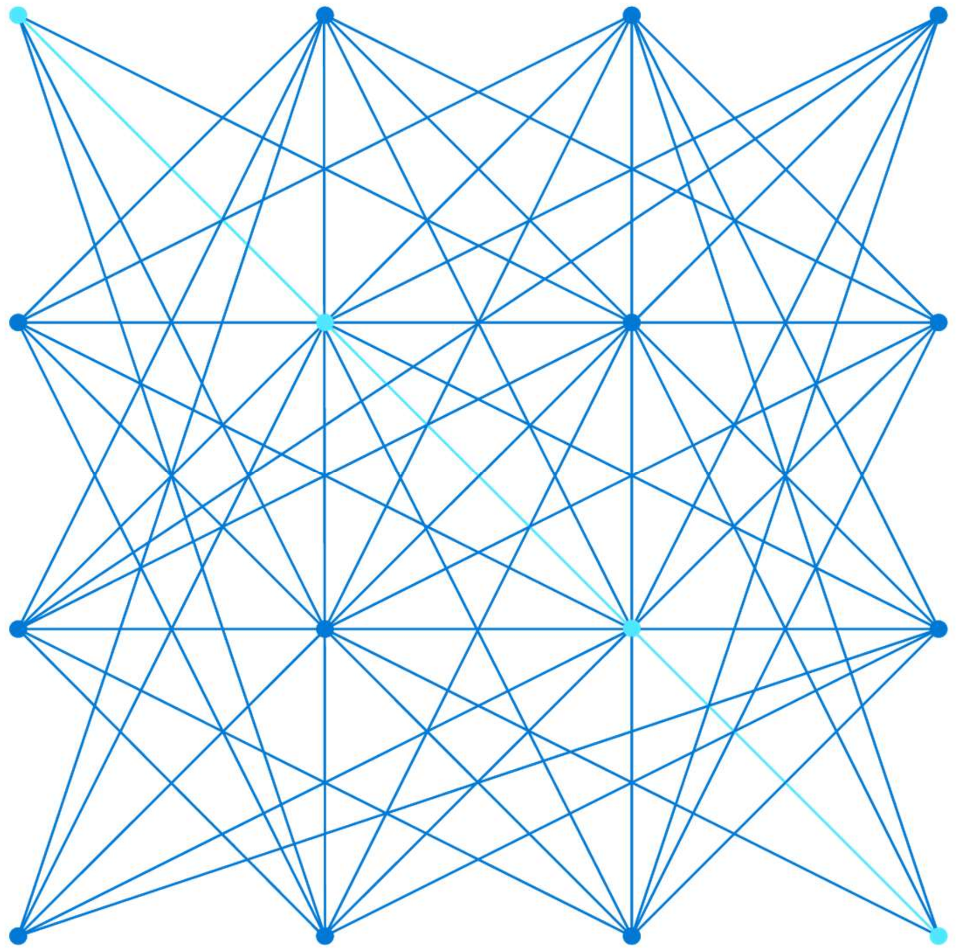
Microsoft Azure

Resilience through chaos engineering

Azure Chaos Studio



Cloud development

The cloud has revolutionized application development

Management, maintenance, and security of physical infrastructure is eliminated.
PaaS services abstract away compute and network layers, enable faster development of complex applications.

With this comes a lack of control

Outages have more impact and unknowns.
Disruptions to dependencies can have cascading effects.

Applications must be designed for failure

Resilience is a shared responsibility between cloud provider and application developer.
Microsoft provides the Well Architected Framework and application development guidance.

©Microsoft Corporation
Azure

Resilience: the capability of a system to handle and recover from disruption

Service outages impact availability

Availability impacts business: upsets customers + can lead to financial, life-and-death, or legal consequences

Outages steal developer time from working on improvements and new features

Quality practices need to be built-in to the entire service development and operation lifecycle

©Microsoft Corporation
Azure

Outages have major real-world consequences

2300 flights
cancelled during Southwest Airlines datacenter outage, costing $150M June 2016[1]

$10 refunds
after 30-minute YouTube TV outage during the 2018 World Cup, costing around $8M[2]

2.5 hours
Walmart lost over $9M in revenue during outage on Black Friday weekend 2018[3]

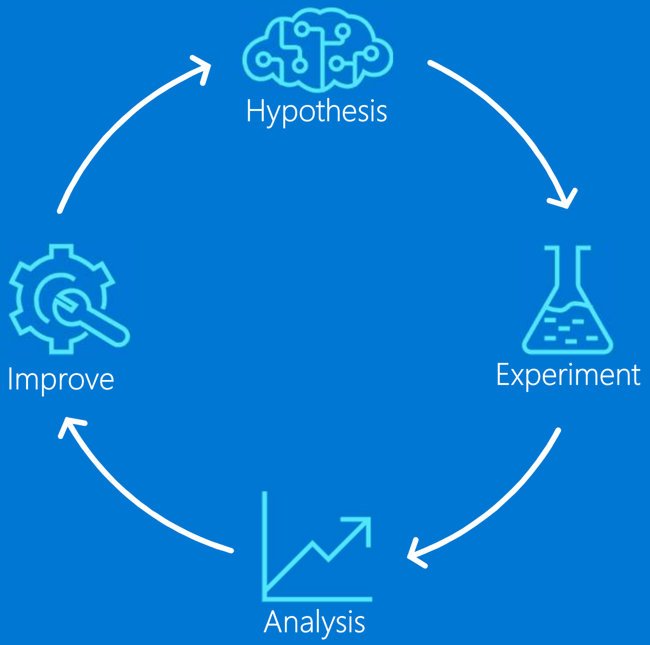
$100M
Amazon lost almost $100M in revenue in just 1 hour of downtime during Prime Day 2018[4]

1. Travel and Leisure, Southwest computer outage cancels flights, 2018 2. Business Insider, J. Crew website crashes on Black Friday, 2018 3. Variety, YouTube TV Offers one week credit after World Cup outage, 2018 4. Business Insider, Amazon Prime Day Website Issues, 2018

What is chaos engineering?

Improve resilience with systematic controlled chaos

- In the cloud, dependencies can be disrupted at any time
- Design for failure, architect for resilience
- Validate resilience through fault injection to confirm that your solution can withstand the disruptions it will encounter in production



Chaos use cases

Perform automated and UI-driven resilience validation

- Disrupt dependencies with ad hoc experiments in a dev/test environment to validate new code
- Gate code flow in CI/CD pipeline automation
- Perform incident fix and incident regression testing
- Host a drill event or game day
- Validate on call and livesite process
- Conduct Error Budget testing

Leverage the **scientific method** to formulate hypotheses around resiliency scenarios, choose a safe environment for experimentation, craft and execute a fault injection experiment, monitor the impact, analyze results and make improvements.

©Microsoft Corporation
Azure

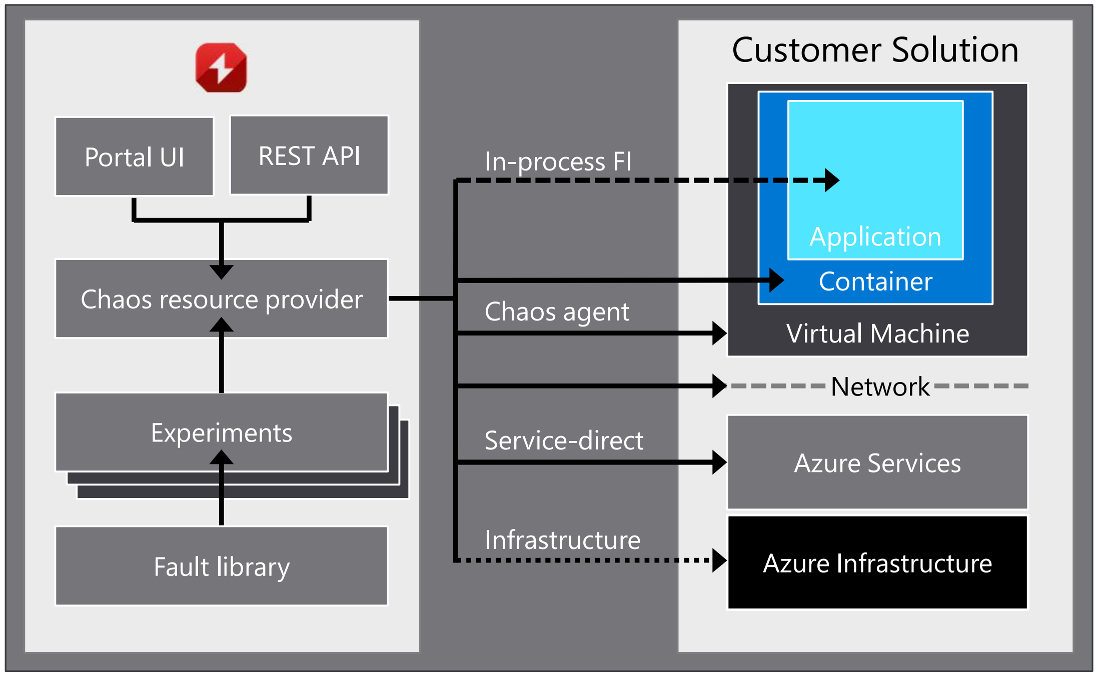
Azure Chaos Studio
Measure, understand, improve, and maintain product resilience

- Fully managed service to validate Azure application and service resilience.
- Deep Azure integration, including Azure Portal, Azure Resource Manager, and Azure Monitor.
- Portal UI and REST API and SDKs to execute manual and automated chaos experiments.
- Expandable library of common resource pressure and dependency disruption faults and actions.
- Advanced workflow orchestration for manual and automated fault injection experiments that simulate real-world scenarios with parallel and sequential action execution.
- Safeguards to minimize impact radius and to control experiment execution.

©Microsoft Corporation
Azure

Azure Chaos Studio
Measure, understand, improve, and maintain product resilience

- Fully managed service to validate Azure application and service resilience
- Deep Azure integration, including Azure Portal, Azure Resource Manager, Azure Monitor
- Portal UI and REST API and SDKs to execute manual and automated chaos experiments
- Expandable library of common resource pressure and dependency disruption faults and actions
- Advanced workflow orchestration for manual and automated fault injection experiments that simulate real-world scenarios with parallel and sequential action execution
- Safeguards to minimize impact radius and to control experiment execution



©Microsoft Corporation
Azure

Disruption types

Expanding library of faults and actions across the entire Azure stack

Chaos for Applications

In-process fault injection to perturb managed code function and API calls

- Azure SDKs (*preview*)
- Customer code (*future*)

Guest OS agent

Chaos agent for Windows and Linux VMs

- Resource pressure faults (CPU, memory, disk, network)
- Dependency disruption faults (process/service, network)

Service direct

Agentless and mock/proxy faults against Azure services

- Dependency disruptions
- Configuration changes
- Latency

Chaos for Infrastructure

Used internally by Microsoft teams to disrupt Azure infrastructure

- Chaos agent deploy to host
- Used by Microsoft to improve the Azure platform

©Microsoft Corporation
Azure

Chaos experiment

Orchestrated, real-world scenarios with fault actions applied to resource targets while under load

Elements

Application
: Application/service for resilience validation.

Target
: Resource target(s) enabled for use in chaos experiments.

Workload
: Synthetic workload or real customer traffic for representative customer usage.

Fault actions
: Orchestrated fault and other (delay, load) actions.

Observability
: Telemetry and thresholds to monitor application health.

Process

Formulate hypothesis
: What scenario is being validated and what are possible and expected outcomes?

Craft experiment
: Orchestrate workload execution with fault actions against subscription resource targets.

Execute and monitor
: Compare baseline observability metrics, monitoring telemetry, reporting and analysis.

Analyze and improve
: New code, code and configuration changes.

Steady state
: Rinse and repeat; add continuous production monitoring and validation.

©Microsoft Corporation
Azure

Chaos Studio use cases

Scenario Validation

- Availability Zone down
- DNS outage
- AAD outage
- Region down
- Region failover
- BC/DR and HA/DR
- Spike/crush load (Black Friday)
- Network outage
- Resource pressure and noisy neighbor
- Emergency certificate rotation
- Storage failover
- Autoscale validation
- Systematic dependency disruption
- Maintenance events
- and much more…

Fault and Action Library

Agent (Windows, Linux)	Service-direct
• CPU pressure	• VM shutdown or kill
• Physical memory pressure	• VMSS shutdown or kill
• Virtual memory pressure	• Classic Cloud Services shutdown
• Disk I/O pressure	• Cosmos DB failover
• Kill process	• Azure Cache for Redis reboot
• DNS failure (Windows)	• Network security group set rule
• Network latency (Windows)	• Key Vault deny access
• Network disconnect (Windows)	• Key Vault increment certificate version
• Network firewall disconnect (Windows)	• Key Vault disable certificate
• Stress-ng stress (Linux)	• Key Vault update certificate policy
• Stop service (Windows)	• Disable autoscale
• Change time (Windows)	• AKS network disconnect/packet loss/latency
	• AKS pod failure, container failure/kill
	• AKS CPU, memory pressure
	• AKS file I/O latency/failure
	• AKS DNS failure
	• AKS time change
	• AKS HTTP delay, modification

©Microsoft Corporation
Azure

Chaos Studio Roadmap

[Cu]

- ✓ Support for chaos experiments on customer Private Networks
- ✓ Dynamic Target selection for VMSS Availability Zones
- ✓ In-process Fault Injection pilot
- ✓ Deployment to more regions
- ✓ Faults
  - ✓ Azure Cache for Redis reboot
  - ✓ Classic Cloud Services shutdown
  - ✓ Key Vault Deny Access

[Zn]

- Dynamic Target selection across more resource properties
- User assigned managed identity
- Identity Management improvements
- Chaos agent Private Network Support
- ADO Pipeline task for automation
- Observability integration and hypothesis attainment scoring
- In-Process Fault Injection preview
- Azure Load Test Integration
- Infrastructure multi-step, multi-branch
- Faults
  - Disable, Update, Increment Cert
  - Disable Autoscale
  - Network packet drop (Windows agent)
  - Host suspend/resume

[Ga]

- Drill Manager UX
- In-Process Fault Injection UX
- Observability stop conditions
- Resiliency Score
- Canary Drills powered by Chaos Studio
- More faults…
- More regions…

- Billing
- GA fall 2023

Get started at https://aka.ms/AzureChaosStudio

Get started

http://aka.ms/AzureChaosStudio

1. **Explore** – Become familiar with the Chaos Studio user interface. Set up a simple VM, enable agent and service-direct faults against it, create and execute an experiment to shutdown the VM or apply CPU pressure, and monitor the impact.
2. **Pilot** – Integrate chaos provisioning and agent deploy into a test environment and add chaos experiments to integration, stress, or other tests. Gauge impact through existing monitoring and impact to established baselines.
3. **Automate** – Provision chaos and agent deploy as part of test environment buildout or daily automation pipeline. Add automated chaos experiment execution to ADO Pipeline or CI/CD process in a pre-production environment to catch issues before they impact customers.
4. **Drill** – Perform a drill or game day event. Choose a scenario, establish a hypothesis or hypotheses, set aside a day, author and pre-test an experiment or experiments, choose participants and preview monitoring, perform the drill on the chosen day, monitor closely, analyze results, create repair items, rinse and repeat.

©Microsoft Corporation
Azure

FAQ

1. Q: Billing?
   A: Eventually. Planning pay-as-you-go model.
2. Q: Do you support faults against on-premise or cross-cloud resources? how about Azure Edge or Arc?
   A: No. Network faults can be used to block access to these, but at this time we do not support non-Azure and on-premise resources. This is in our roadmap.
3. Q: Does Chaos Studio support rollback at the end of an experiment or if things go wrong?
   A: No. At end of experiment, non-destructive fault behaviors end (CPU, memory, disk, network pressure) and systems return to pre-experiment state but if a VM or process is killed, Cosmos DB failed over, etc. nothing is done to restore state.
4. Q: Are you actually *causing* or *simulating* issues and outages?
   A: Both… it depends on both the fault and the environment. Faults can be combined to represent more complex scenarios.
5. Q: Can I create my own fault?
   A: Coming soonish: BYOF (internal/external) and inner source (internal)
6. Q: Does Chaos Studio create a workload or load against my application?
   A: No. Chaos Studio is primarily a platform for orchestrating fault injection. See ALT. Coming soon: ALT load action. Coming later: custom load action
7. Q: Do you have PaaS fault coverage for services XXX, YYY, and ZZZ?
   A: Not yet, but the NSG and network faults can impact many services

©Microsoft Corporation
Azure

Demo

Azure Chaos Studio

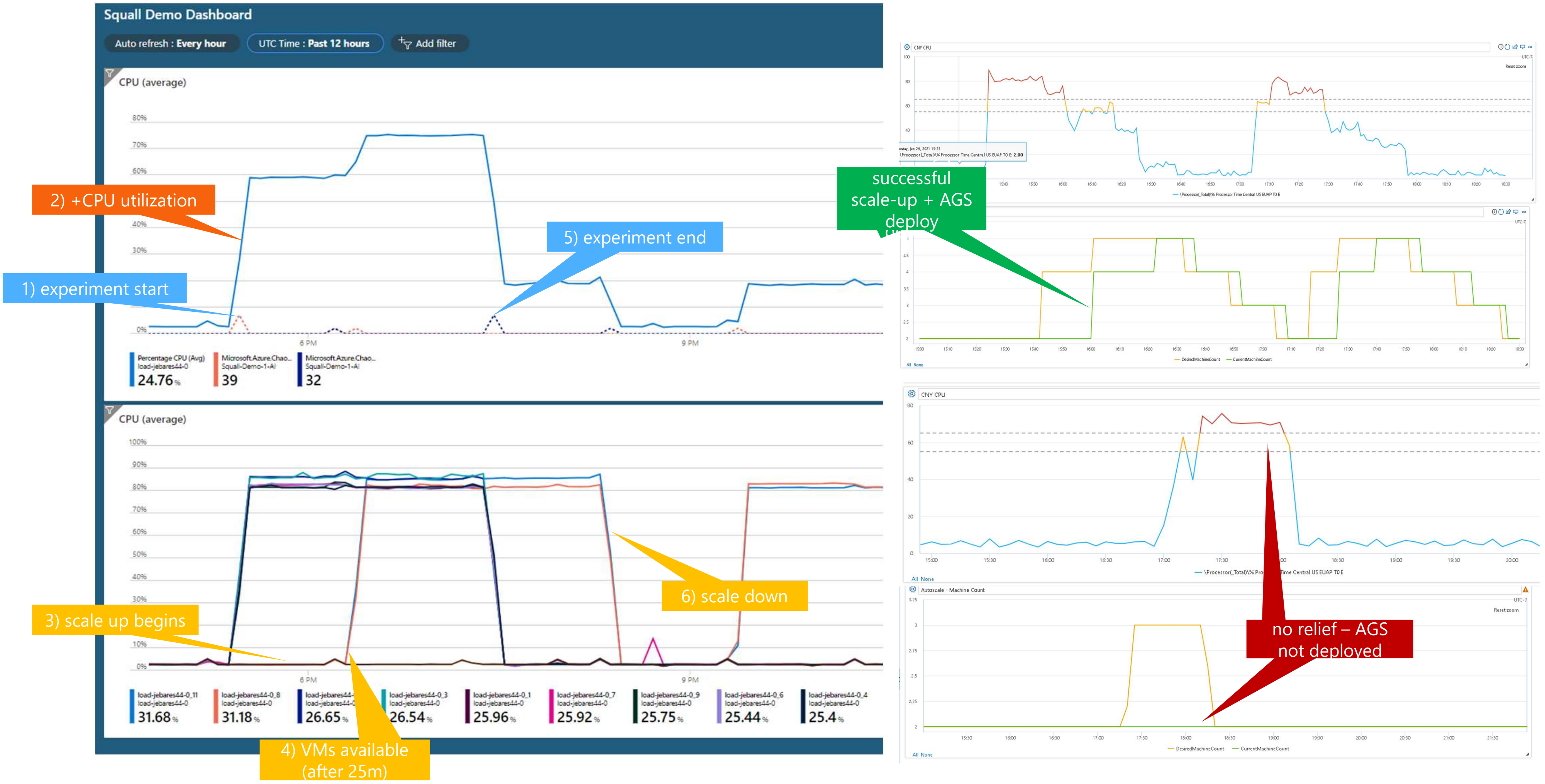
Microsoft Azure

Thank you.

© Copyright Microsoft Corporation. All rights reserved.

Case Study: autoscale validation

- Issue: machines provisioned, but AGS not deployed (autoscale worked, got billed, but service not deployed)

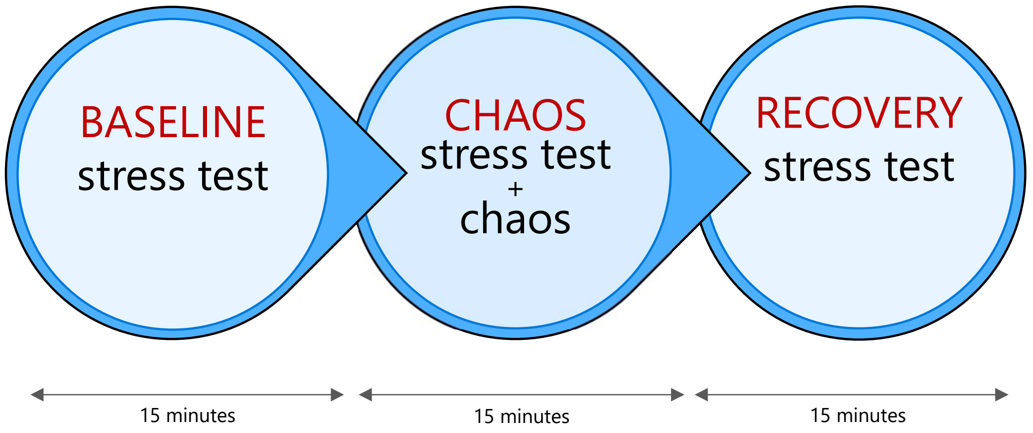


17

Case Study: nightly validation of new builds

Daily build deployed to VM test cluster.

Chaos added to stress test pass, stress extended to 3 phases with resource pressure faults (CPU, Memory, Disk IO, etc.) run across 10 VMs.



Multiple metric comparisons enabled

- **Baseline -** compare throughput and other test metrics against goals and historical trends
- **Chaos -** measure the impact of chaos to SLO and service health
- **Recovery -** validate time to return to steady state