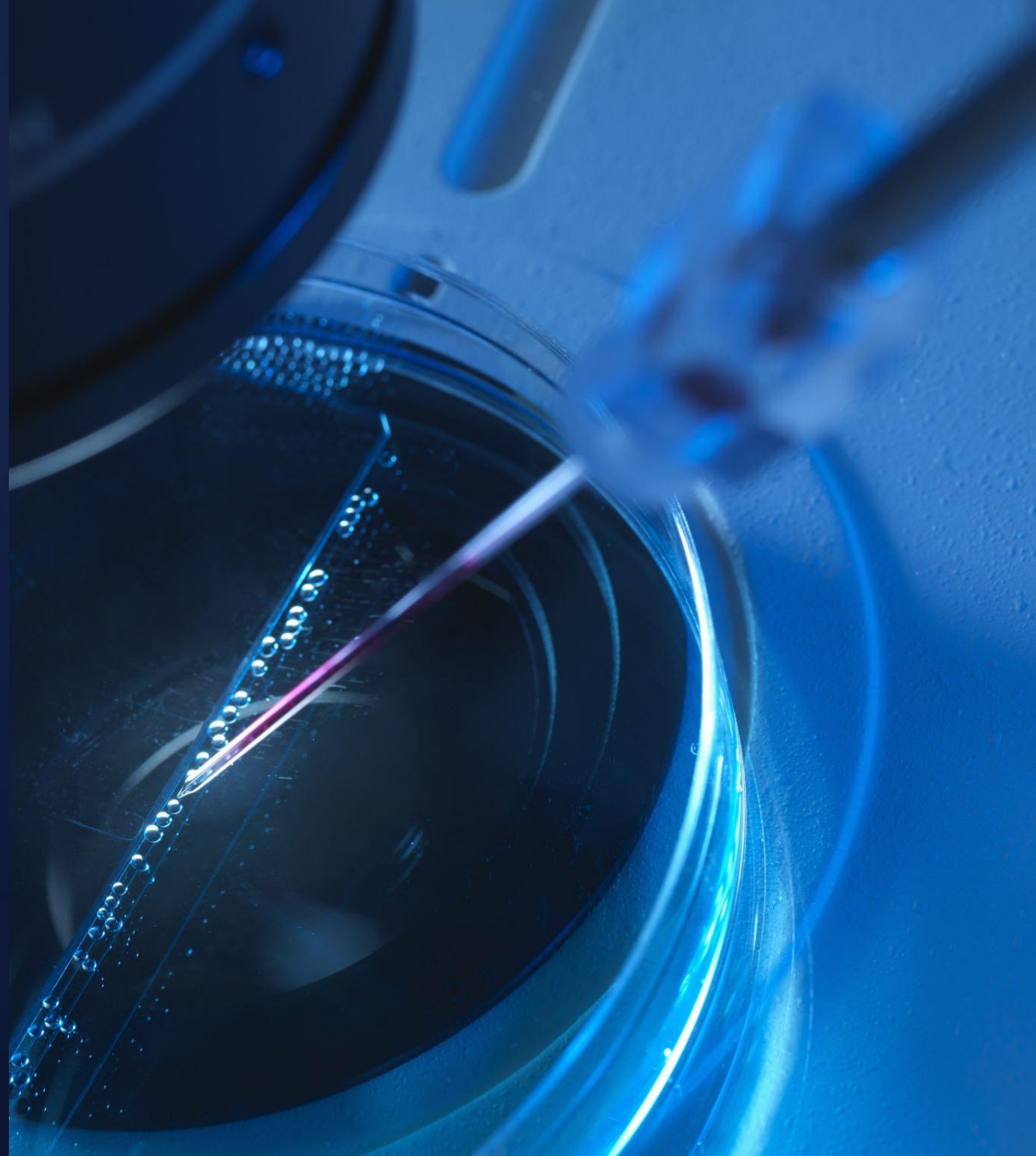
Microsoft

Azure Load Testing

N. Krishnakumar
Principal Engineer, Core AI, Microsoft

12/ 07 / 2025



Azure Load Testing

A fully managed load-testing service that helps developers and testers generate high-scale load and gain actionable insights to identify bottlenecks

Microsoft

Demo of Azure Load Testing

Microsoft

Overview

Building performant applications

Design principles

- Start with business goals
- Design to exceed the goals
- Achieve and sustain performance
- Fine tune to improve efficiency

Achieve & sustain

- Test early, test often, test everywhere
- Shift left your performance validation too!
- Continuous validation

Performance Testing types

Load Testing

- Validate under expected load
- Throughput & response time
- Short to medium duration

Stress Testing

- Validate beyond peak limits
- Failure points & recovery
- Short bursts of high intensity

Soak Testing

- Long term stability
- Degradation over time
- Long duration

In CI/CD

Easily ensure app reliability with Azure Load Testing

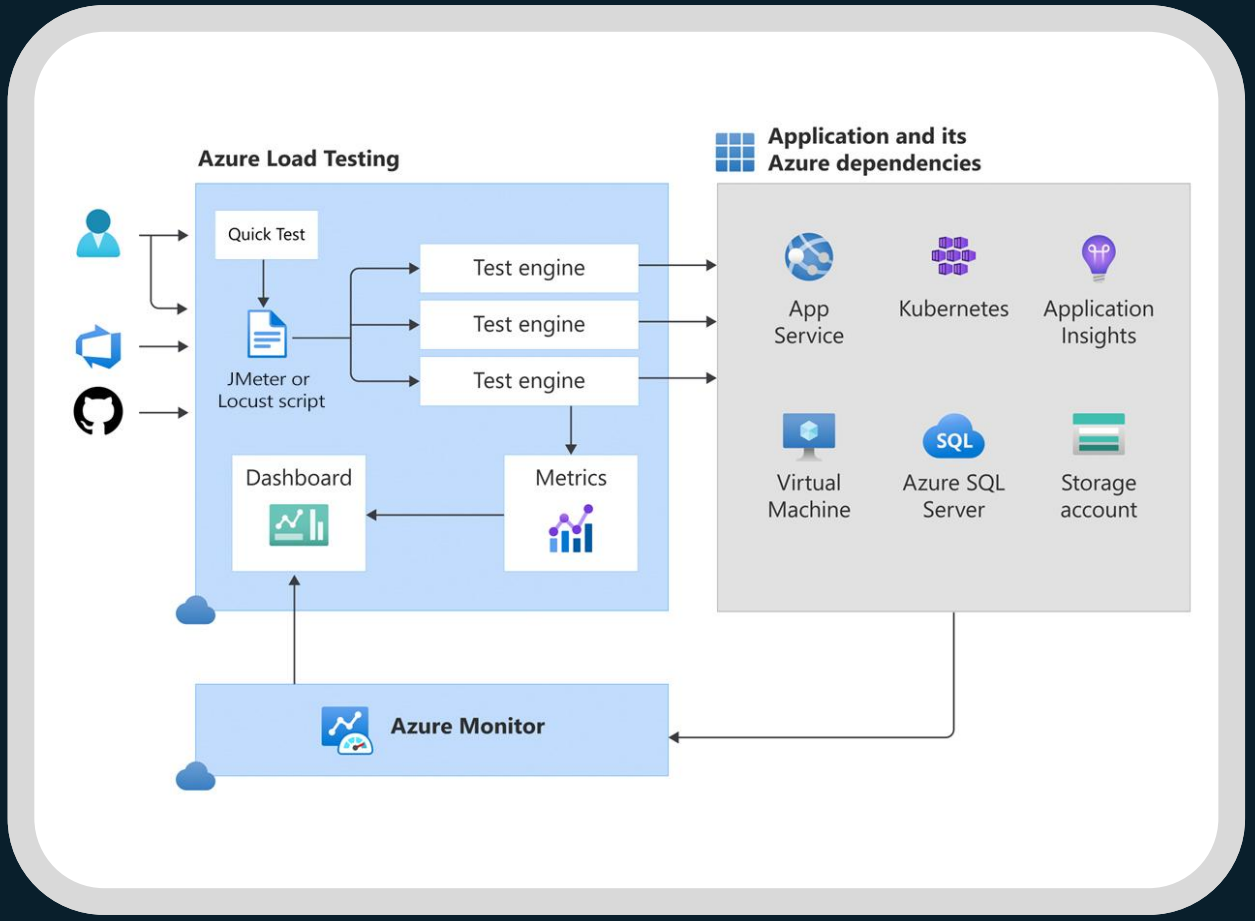
Generate high-scale load with ease

Optimize performance at scale

Build load testing into DevOps workflows

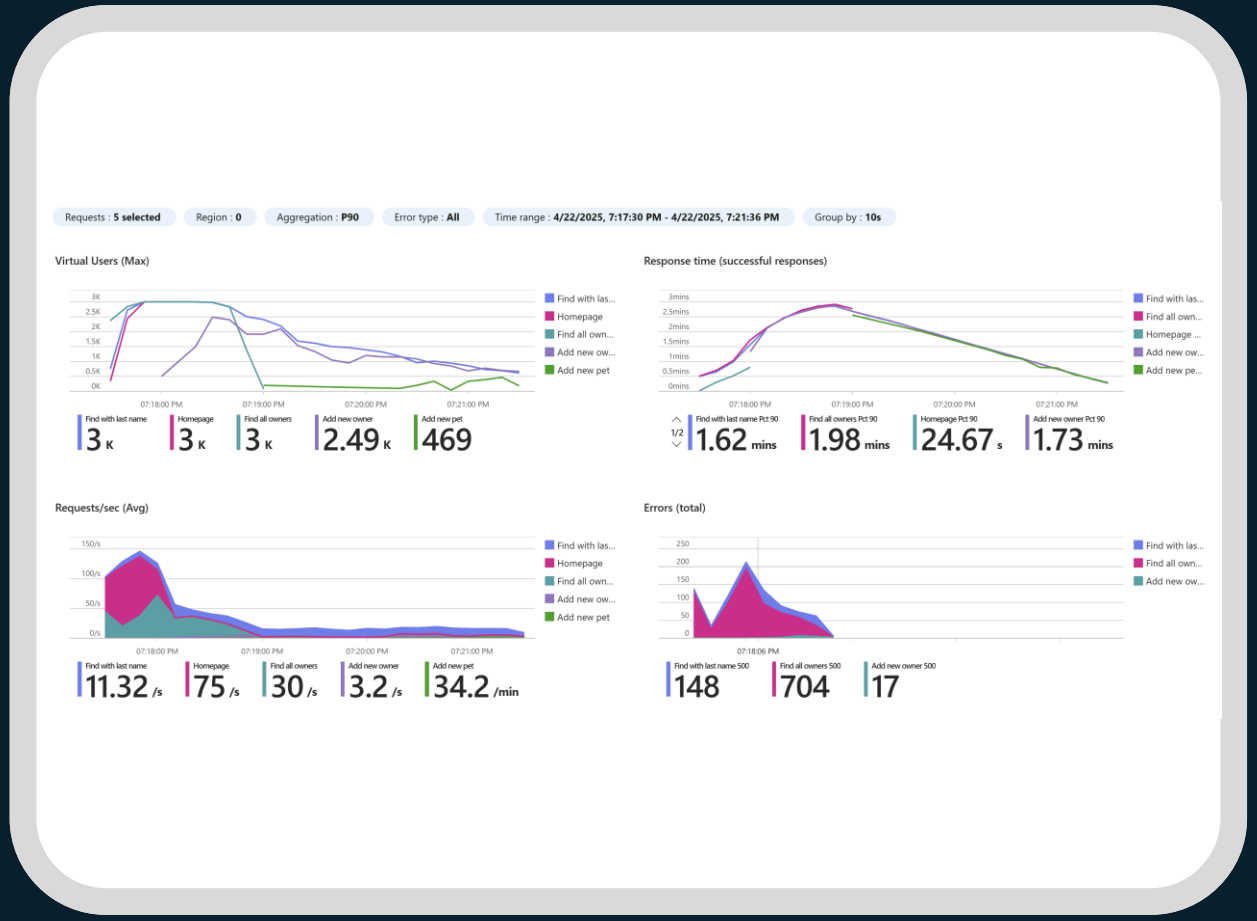
Generate high-scale load quickly and easily

- Create tests quickly without prior knowledge of load testing tools
- Run existing test scripts at scale with high-fidelity JMeter and Locust support
- Eliminate infrastructure needs with a fully managed service
- Experience frictionless testing on Azure



Identify bottlenecks with actionable insights

- Generate deep, actionable insights and recommendations
- Understand how tests impact all parts of your app
- Compare across load tests to understand change over time
- View trends in performance and compare with baseline



Microsoft

Design Principals

Design for Business Requirements

Drive customer success by defining targets for key user flows

Critical user flow:

Test run execution

- Executing the test runs reliably as configured for various load conditions is critical to ensure user success

What does that include?

- On-demand provisioning and configuration of virtual machines.
- Running the test script for the configured duration.
- Collecting metrics and logs from each engine; generating dashboards.
- Processing and archiving logs and results

Goal

- 99.9%

Design for Business Requirements

Identify dependencies and their impact on resilience

Dependence on Azure services

- Heavy dependence on Azure platform resources for compute, messaging, networking and storage requirements

Challenges at scale

- Consistent and performant on-demand/bursty test run infra provisioning
- Consistent and optimal metrics and logs collection and storage

Design for Business Requirements

Measures implemented to address reliability challenges

Measures implemented

- Implemented retries with exponential backoff.
- Established a warm pool of on-demand infrastructure resources for test runs.
- Optimized data collection through metrics aggregation and log chunking, among other methods.
- Monitored the health of dependencies.

Result

- Improved and sustained reliability above a remarkable 99.9%

Design for Operations

Measure, monitor and improve

Telemetry

- Create observable systems for correlating telemetry across components
- It aids in effective request trace analysis across diverse microservices.

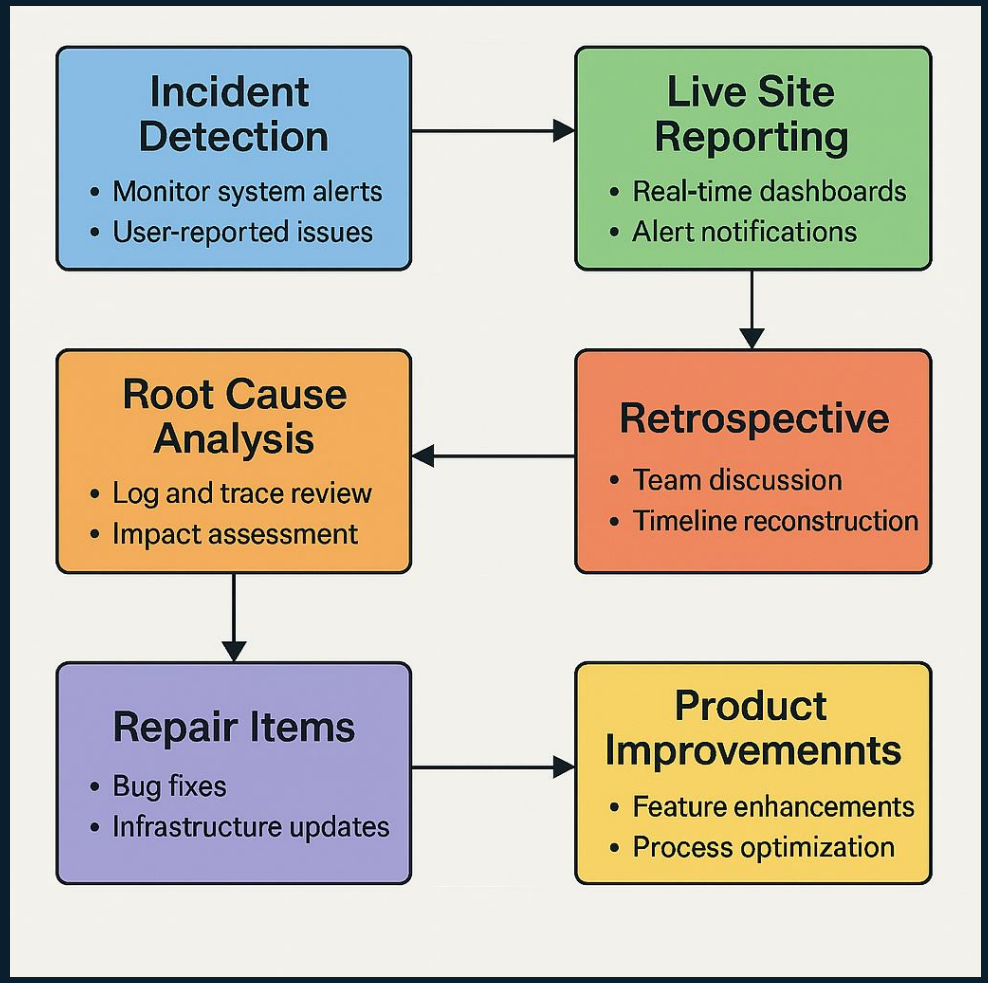
Monitoring & diagnostics

- Comprehensive health monitoring at service level
- Collect application metrics and logs via App Insights.
- Gather Azure resource infrastructure metrics and logs using Azure Monitor.
- Detect anomalies e.g. in key user flow metrics and usage metrics.

Design for Operations

Continuous learning

- Integrate with robust alerting, live site reporting and analysis tools.
- Complete the feedback loop with retrospective and repair items for the incidents



Reliability – Key take-aways

Achieving and maintaining reliability goals is a continuous effort

- Identify and measure critical user flows.
- Incorporate observability in design discussions from the beginning.
- Collect platform and applications metrics and logs to gain insights into the business and platform.
- Utilize anomaly detection for proactive issue identification and resolution
- Complete the feedback loop with incident reviews and repair items.