





1, 2, 3... testing: Is
this thing on(line)?

Meet your new Microsoft Testing tools

Who Am I ▼ + New dashboard Edit dashboard Share Fullscreen Clone Delete



Quickstart tutorials

Windows Virtual Machines ▼
Provision Windows Server, SQL Server, SharePoint VMs

Linux Virtual Machines ▼
Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs

App Service ▼
Create Web Apps using .NET, Java, Node.js, Python, PHP

Functions ▼
Process events with a serverless code architecture

SQL Database ▼
Managed relational SQL Database as a Service

Brussels Edit

08:58

Monday, May 22, 2023



Marketplace



Service Health



Audit Logs

[View activity](#)

Audit Logs

[View activity](#)



Mike Martin

Technical Evangelist

#Azure

Ex-MVP, Advisor, User group



Social

- [@techmike2kx](#)
- mimar@microsoft.com

//what

- Flexible
- Passionate about tech
- Family
- Global reach
- Knowledge sharing

//also

- curious
- Straightforward
- Feedback eager!



New

Dashboard

Service Health

Subscriptions

What's new

Recent

Help + support

All resources

Resource groups

Resource Explorer

Activity log

Cost Management + Billing

Tags

Templates

Monitor

Metrics

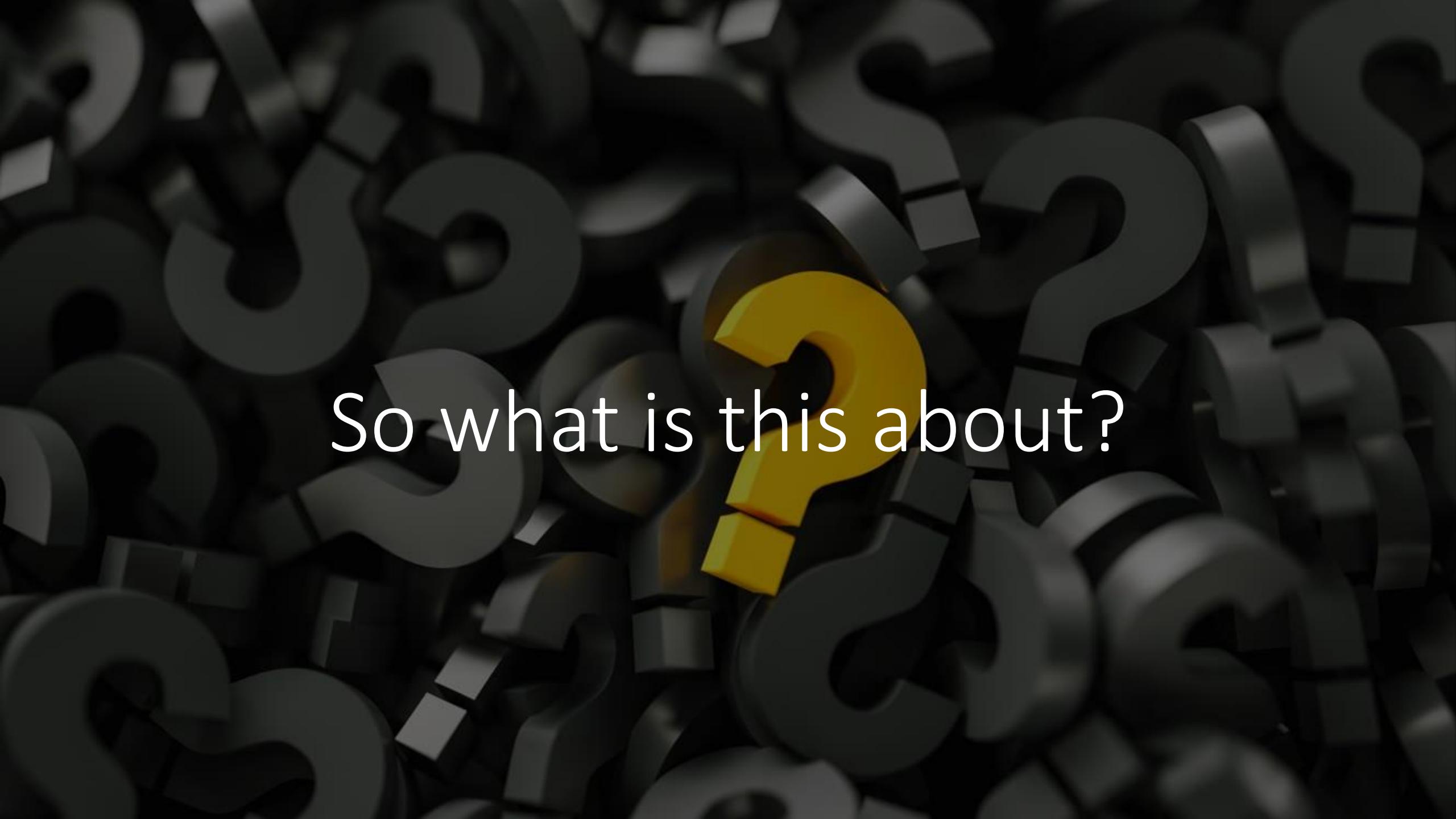
Security Center

Advisor

Users and groups

Portal settings

More services >

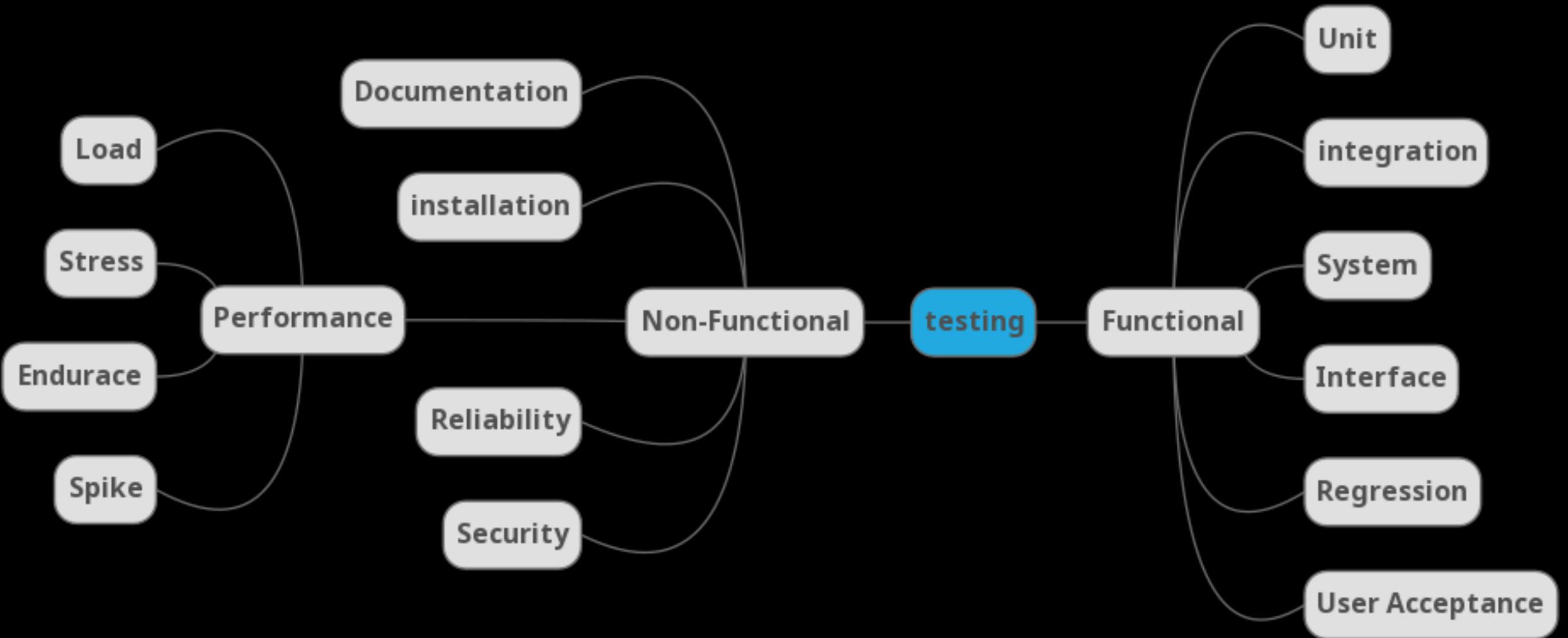
A large yellow 3D-style question mark is centered in the foreground, surrounded by numerous smaller, darker gray question marks of various sizes, creating a sense of depth and repetition.

So what is this about?

One word : FEEDBACK

Actually more like 4 : FINO

FAILURE IS NO OPTION



Life's a stage ... You have to Playwright ...!

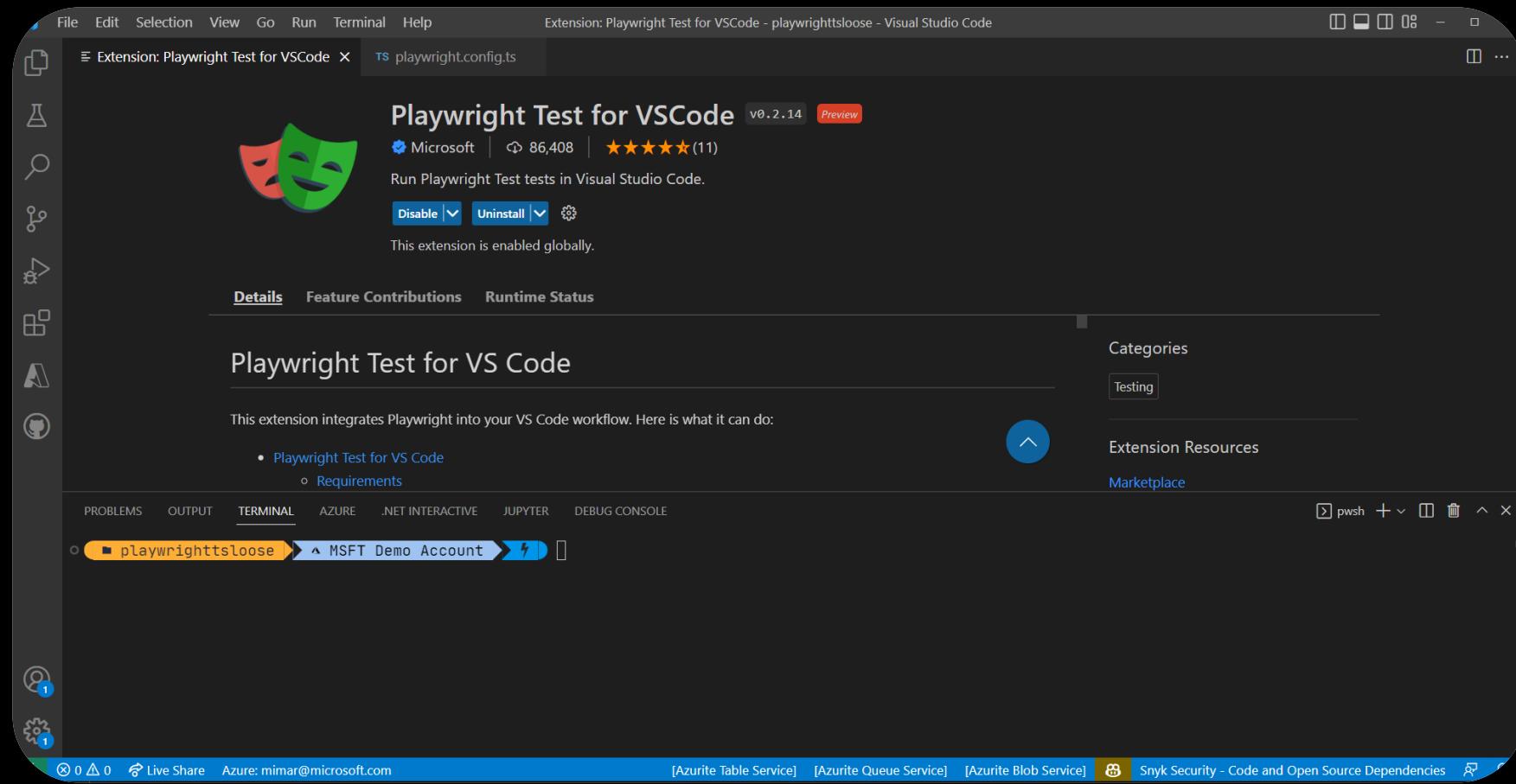


<https://playwright.dev>

What is Playwright?

- Automation library
- **automate Chromium, Firefox, and WebKit with a single API**
- evergreen, capable, reliable, and fast cross-browser web automation
- Using websockets and no browser hook-in
- So benefitting the devtools directly

Visual Studio Code Extension



trace

Playwright e2e.spec.ts:3 › test

The screenshot shows the Playwright Trace interface with a timeline from 0 to 13.5s. The timeline is divided into Actions (blue), Metadata (green), and a large black area representing the application's execution. A specific action, 'locator.click role=button[name='Reviews (1)']', is highlighted in blue on the timeline. Below the timeline, a table provides detailed information about the action:

Action	Before	After
browserContext.newPage	44ms	
page.goto https://msftt2kxjs.azurewe... 5.7s		
page.goto https://msftt2kxjs.azure... 1.9s	①	
locator.click role=button[name="Close"]	1.4s	
locator.click text=Apple Juice (1000ml)	601ms	
locator.click role=button[name="Re... 472ms		
locator.click text=One of my favorites!	296ms	
locator.click text=One of my favorites!	71ms	
locator.click text=One of my favorites!	64ms	
locator.click text=admin@juice-sh.op	50ms	
locator.click text=One of my favorites!	75ms	
locator.click role=region[name="Revie... 41ms		
locator.click role=button[name="Clos... 61ms		
locator.click role=button[name="La... 810ms		
locator.click text=Deutsch	478ms	
locator.click role=button[name="La... 245ms		
locator.click text=Nederlands	382ms	
locator.click role=button[name="dis... 152ms		

The screenshot also shows the OWASP Juice Shop website's product page. A modal for 'Apple Juice (1000ml)' is open, displaying its price (1.99€) and a 'Reviews (1)' button. The timeline highlights the click on this button.

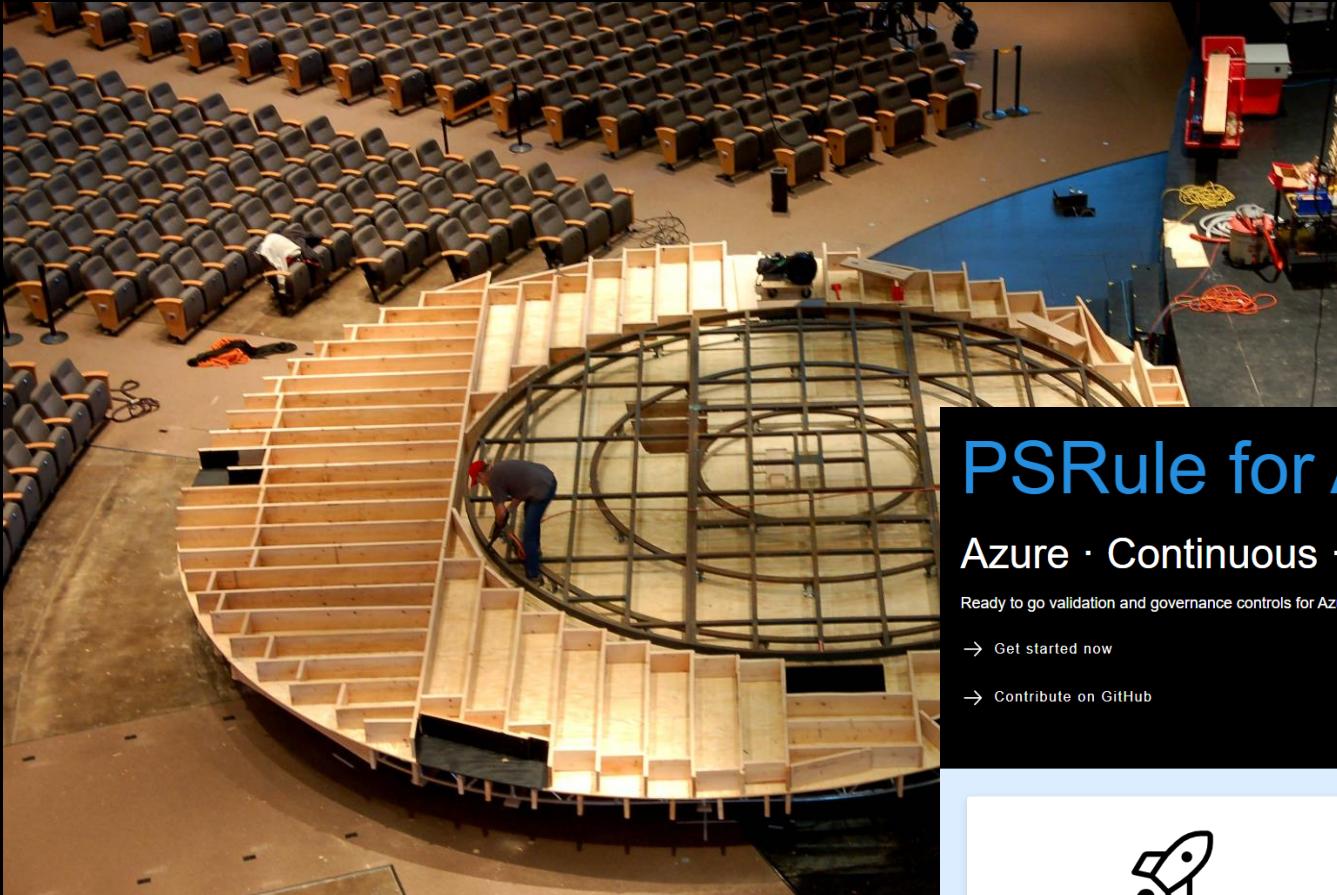
Call Console Network Source

locator.click
TIME wall time: 4/11/2022, 15:14:29 duration: 472ms
PARAMETERS selector: "role=button[name='Reviews (1)']"
strict: true
LOG waiting for selector "role=button[name='Re... selector resolved to visible <mat-expansion... attempting click action waiting for element to be visible, enabled a... element is visible, enabled and stable scrolling into view if needed done scrolling performing click action click action done waiting for scheduled navigations to finish navigations have finished

report

Q		All 6	Passed 6	Failed 0	Flaky 0	Skipped 0
⌄ e2e.spec.ts		1.7m				
✓ test	e2e.spec.ts:3	chromium	13.9s			
✓ test	e2e.spec.ts:3	firefox	14.5s			
✓ test	e2e.spec.ts:3	webkit	25.2s			
✓ test	e2e.spec.ts:3	Mobile Chrome	17.1s			
✓ test	e2e.spec.ts:3	Mobile Chrome	11.6s			
✓ test	e2e.spec.ts:3	Microsoft Edge	19.1s			

Setting the scene....



ARM-TTK

PSRule for Azure

Azure · Continuous · Consistency.

Ready to go validation and governance controls for Azure Infrastructure as Code (IaC).

- Get started now
- Contribute on GitHub



Ready to go

Leverage over 310 pre-built rules to test Azure resources.



DevOps

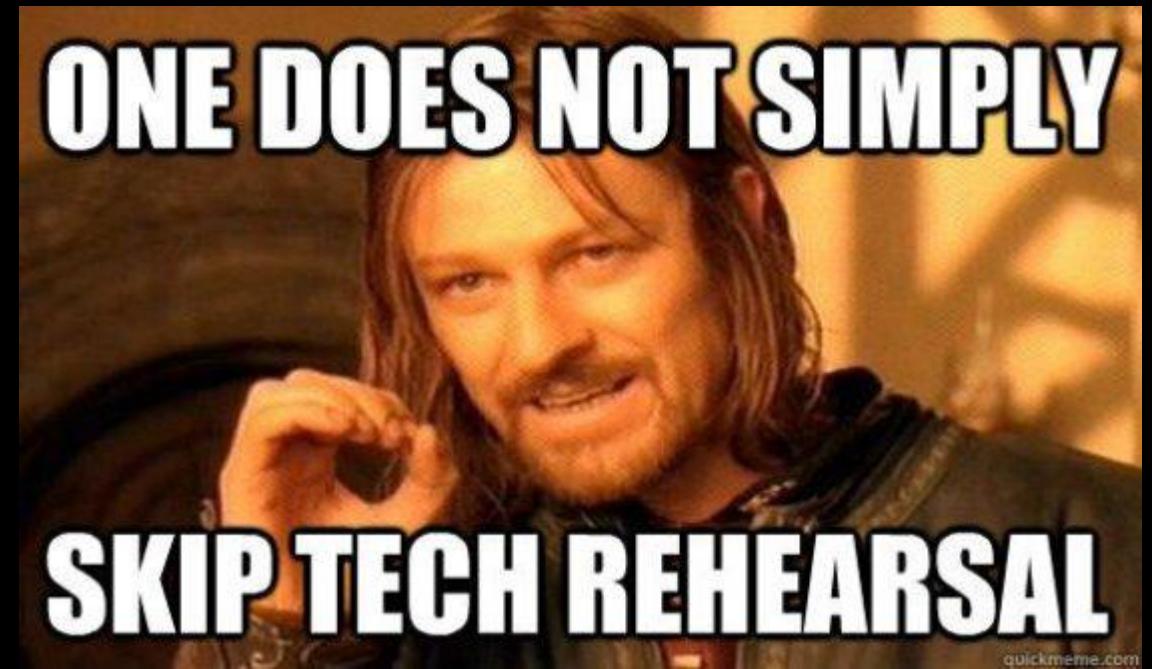
Validate resources and infrastructure code pre or post-deployment.



Cross-platform

Run on MacOS, Linux, and Windows.

Tryouts And general rehearsal



Azure Load Testing

Generate high-scale load with ease

Generate high-scale load without the need for complex infrastructure



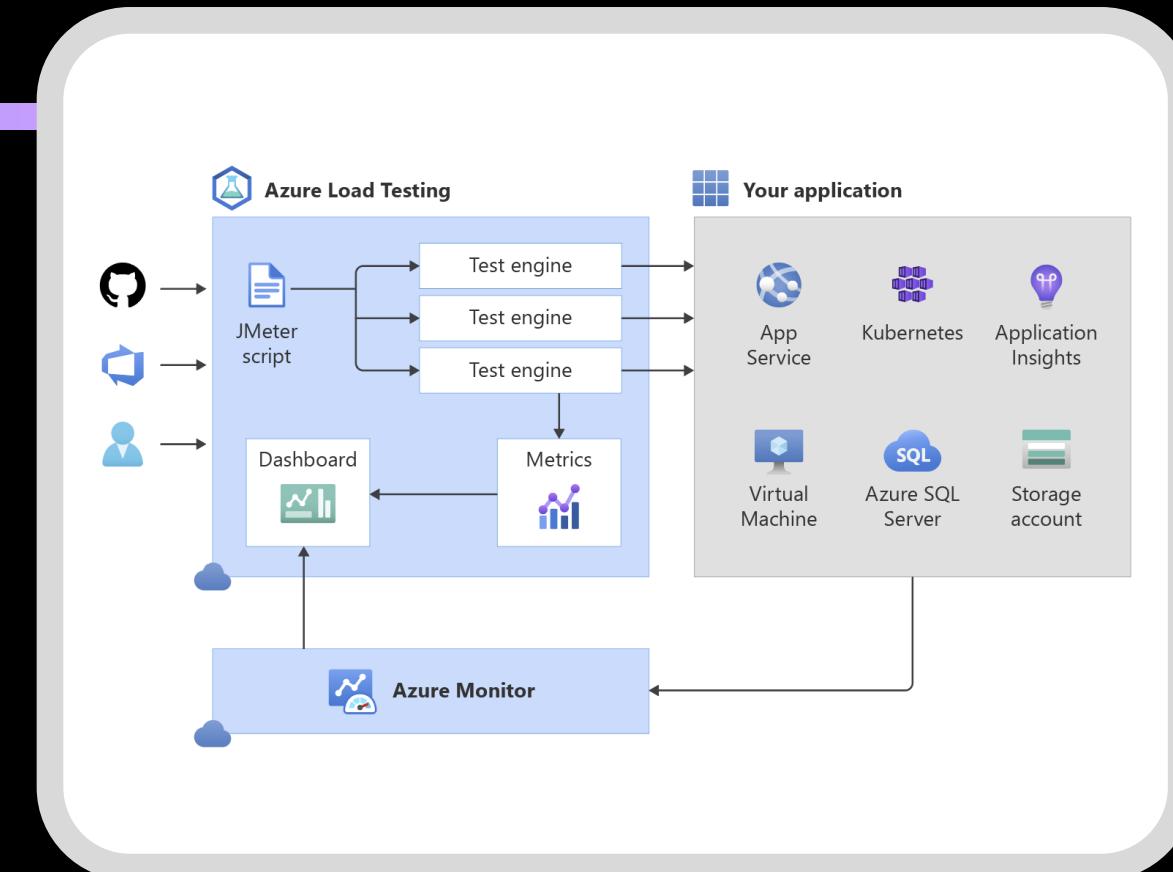
Run existing test scripts with high-fidelity JMeter support



Eliminate infrastructure needs with a fully managed service



Experience frictionless testing on Azure



Optimize performance at scale

Catch and identify app performance issues at scale



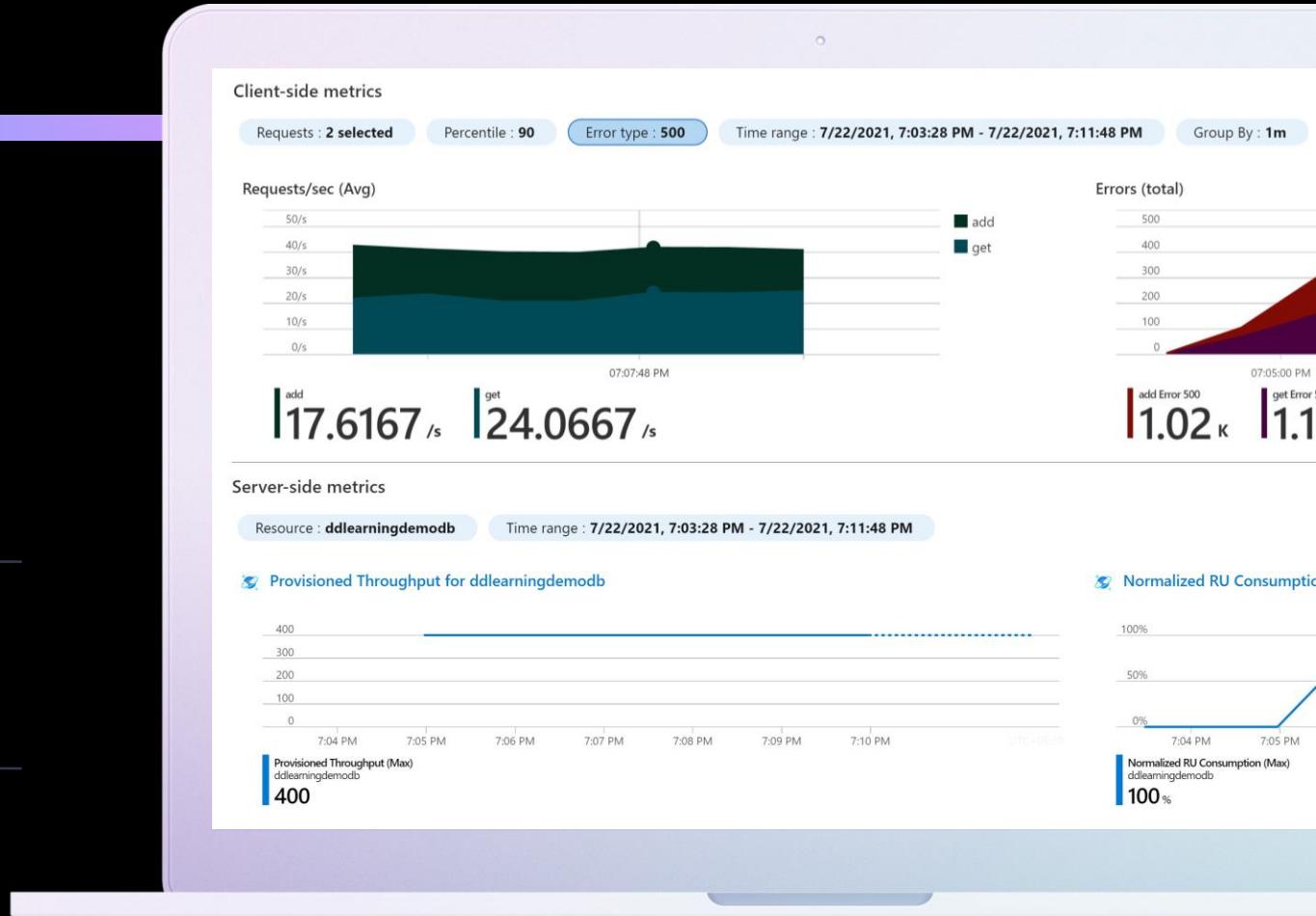
Generate deep, actionable insights and recommendations



Understand how tests impact all parts of your app



Compare across load tests to understand change over time



Build load testing into DevOps workflows

Integrate load testing into existing CI/CD workflows



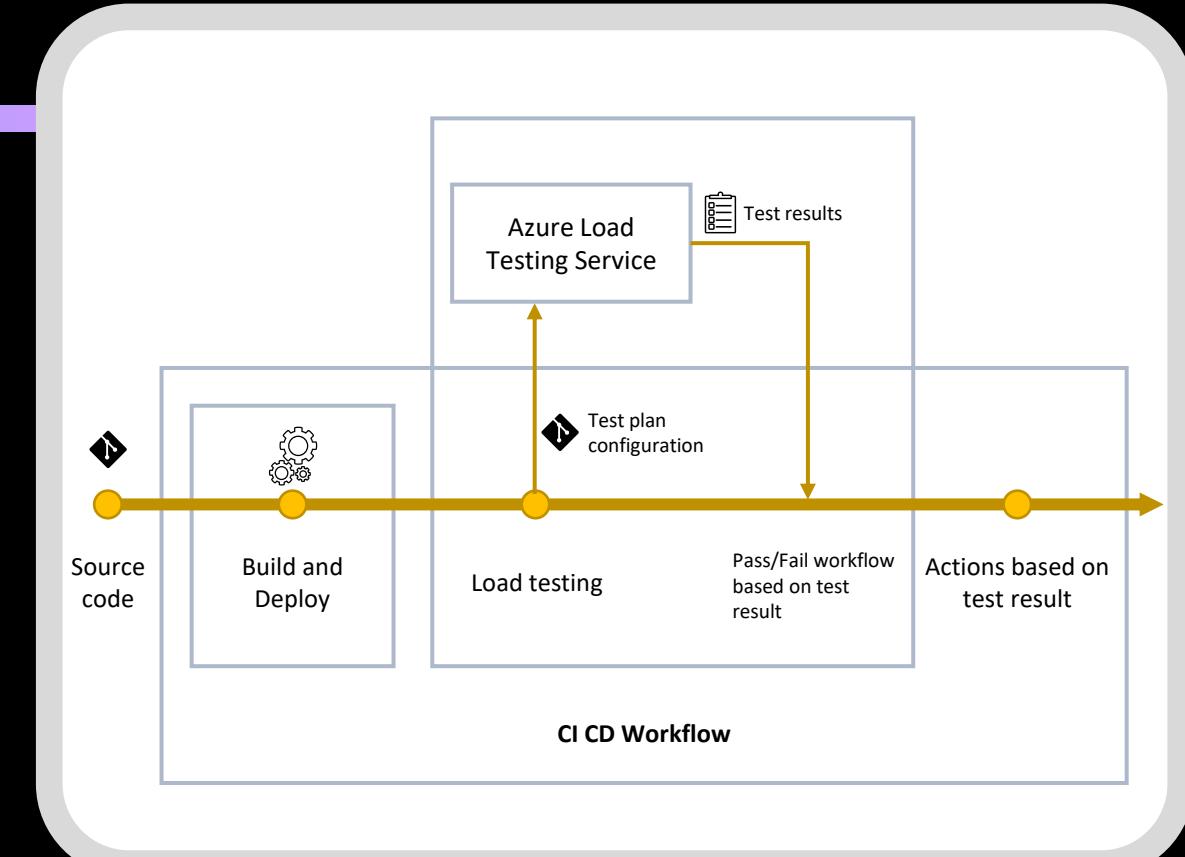
Shift load testing left to continually improve performance



Establish a performance baseline and set clear pass/fail criteria



Automatically export test results to GitHub with full context



Get started quickly with a fully managed service optimized for Azure



Streamline testing with Azure-specific networking patterns



Protect Azure workloads with automated throttling detection



Test efficiently with a service optimized for Azure

The screenshot shows the 'Create new test' wizard on a laptop screen. The title bar says 'Create new test' with a 'PREVIEW' link. Below it is a navigation bar with tabs: 'Basics' (which is selected and underlined), 'Test plan', 'Load', 'Monitoring', and 'Review + create'. A descriptive text block below the tabs says: 'Create and run a Jmeter test. Complete the configurations for each tab and then Review + create to create and run the test.' with a 'Learn more' link. The 'Test details' section contains fields for 'Test name' (set to 'Load Test Sample App API'), 'Test description' (set to 'Test for 10,000 VUs'), and 'Run test after creation' (with a checked checkbox). At the bottom of the wizard are three buttons: 'Review + create' (in blue), 'Previous' (disabled), and 'Next: Test plan'.

Break a leg!



Azure Chaos Studio



Azure Chaos Studio

Measure, understand, improve, and maintain product resilience

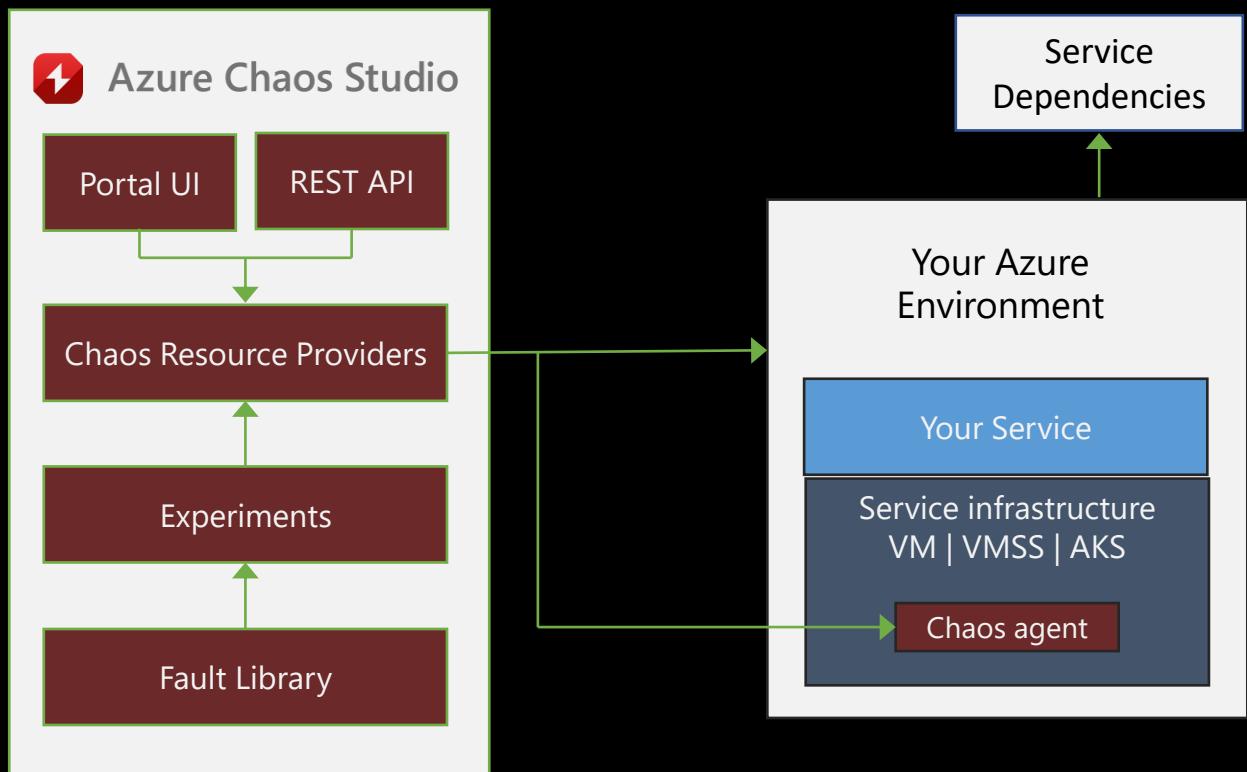
Fully managed service for validating Azure application and service resilience

Deep Azure integration, including Azure Portal, Azure Resource Manager, and Azure Monitor

Expandable library of common resource pressure and service dependency disruption faults

Advanced **workflow orchestration** of manual and automated fault injection experiments that **simulate real-world scenarios** with parallel and sequential actions

Safeguards to minimize impact radius and to stop and roll back experiments

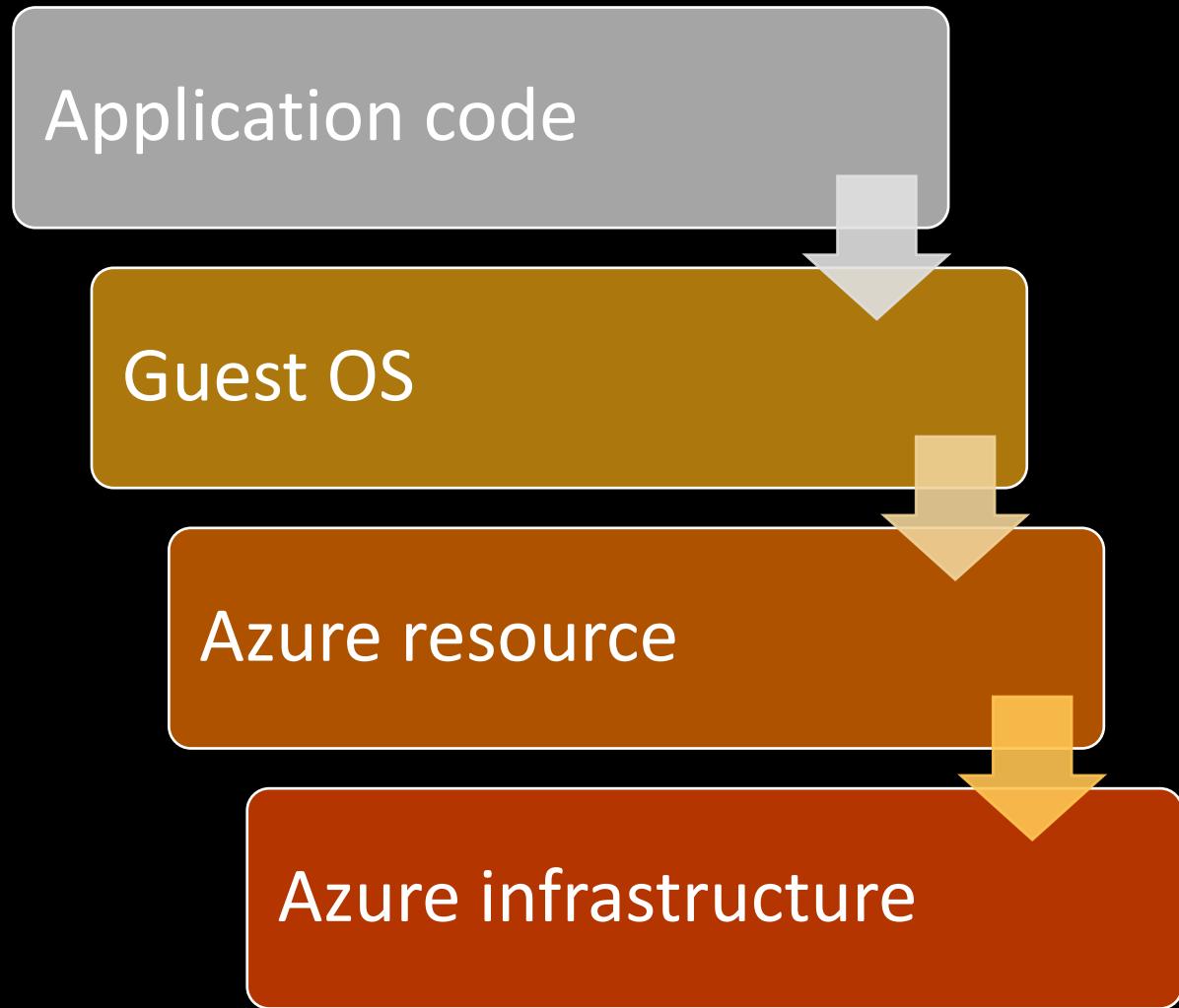


Fault Library

Expanding library of faults
across the entire Azure stack

- Current investments in [Guest OS](#) (agent-based) and [Azure resource](#) (service-direct) fault types.
- [Windows](#) and [Linux](#) agent support
- Destructive, pressure, latency, and configuration change faults

Agent-based	Service-direct
<ul style="list-style-type: none">• CPU pressure• Physical memory pressure• Virtual memory pressure• Disk I/O pressure• Stress-ng stress• Stop service• Change time• Kill process• DNS failure• Network latency• Network disconnect• Network firewall disconnect	<ul style="list-style-type: none">• VM shutdown or kill• VMSS shutdown or kill• Cosmos DB failover• Azure Cache for Redis reboot• Network security group set rule• 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



In-process Fault Injection

- Perturb individual API calls
- Azure SDKs
- Customer code

Agent-based guest OS faults

- Windows and Linux VM agent support
- Resource pressure (CPU, memory, disk, network)
- Dependency disruption (process/service, network)

Agentless (Service-direct, Mocks/Proxy) faults

- Dependency disruption
- Configuration changes

Infrastructure Fault Injection

- Disrupt the host
- Used by Microsoft to improve the Azure platform



Experiment Metadata – Container for consisting of experiment metadata such as azure region where the test is to be deployed, and Identity to be used.

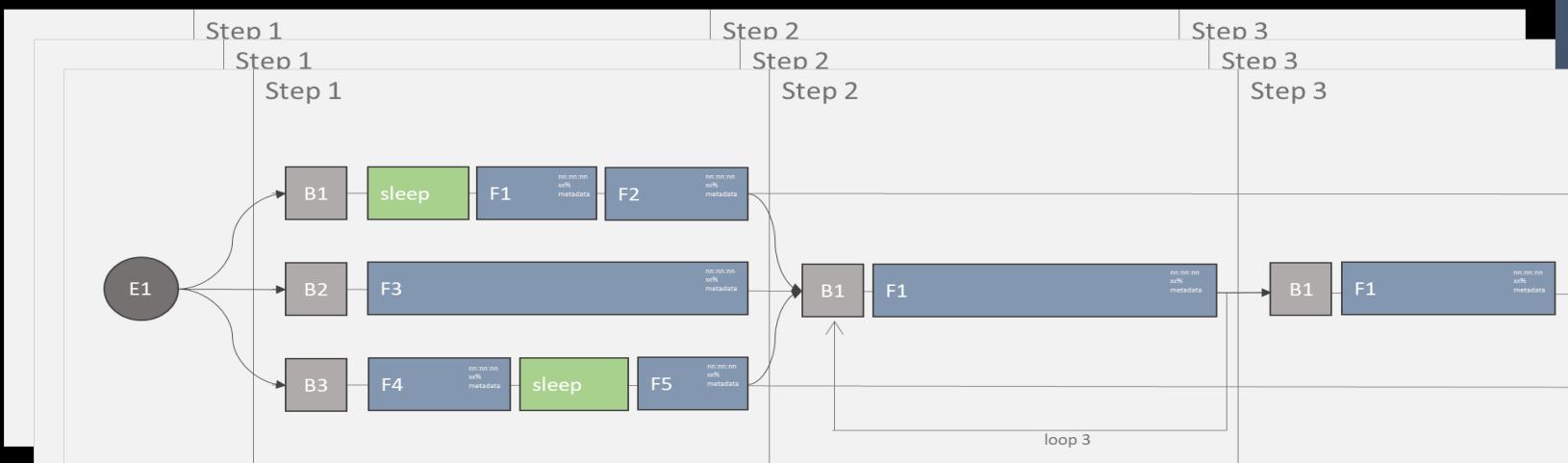
Step – Top level container consisting of Branches that will be run *sequentially*.

Branch – Second level container consisting of Actions that will be run in *parallel*.

Action – Fault that will be executed against a Selector.

Selector – A container of Targets.

Target – A Azure Resource that will.



{Experiment Metadata}

{Step Id}

{Branch Id}

{Action}

{Selector Id}

{Step Id}

{Branch Id}

{Action}

{Action}

{Selector Id}

{Selector Id}

{Target}

{Target}

{Selector Id}

{Target}

{Target}

Experiment Json Example

Step that will be preformed



Branch in this step preformed



Action or Fault to be preformed



Selector where these actions are to be preformed



Selector containing a set of targets



Target or Azure Resources that will be targeted



```
{  
    "location": "eastus2euap",  
    "identity": {  
        "type": "SystemAssigned"  
    },  
    "properties": {  
        "steps": [  
            {  
                "name": "Step1",  
                "branches": [  
                    {  
                        "name": "Branch1",  
                        "actions": [  
                            {  
                                "type": "continuous",  
                                "name": "urn:provider:agent-v2:Microsoft.Azure.Chaos.Fault.CPUPressureAllProcessors",  
                                "parameters": [  
                                    {  
                                        "key": "PressureLevel",  
                                        "value": "90"  
                                    }  
                                ],  
                                "duration": "PT5M",  
                                "selectorid": "Selector1"  
                            }  
                        ]  
                    }  
                ]  
            },  
            "selectors": [  
                {  
                    "id": "Selector1",  
                    "type": "List",  
                    "targets": [  
                        {  
                            "type": "ResourceId",  
                            "id": "/subscriptions/42707c56-5cd5-4343-82b5-  
8c63c64def11/resourceGroups/ChaosDemo/providers/Microsoft.Compute/virtualMachines/demoVM1/Microsoft.Chaos/chaosAgents/chaosAgent"  
                        },  
                        {  
                            "type": "ResourceId",  
                            "id": "/subscriptions/42707c56-5cd5-4343-82b5-  
8c63c64def11/resourceGroups/ChaosDemo/providers/Microsoft.Compute/virtualMachines/demoVM2/Microsoft.Chaos/chaosAgents/chaosAgent"  
                        }  
                    ]  
                }  
            ]  
        ]  
    }  
}
```

It's Showtime! Secure your seat!



What is PSRule for Azure?



Tests and docs for Azure Infrastructure as Code

Supports Azure Templates and Bicep



Azure Well-Architected Framework aligned

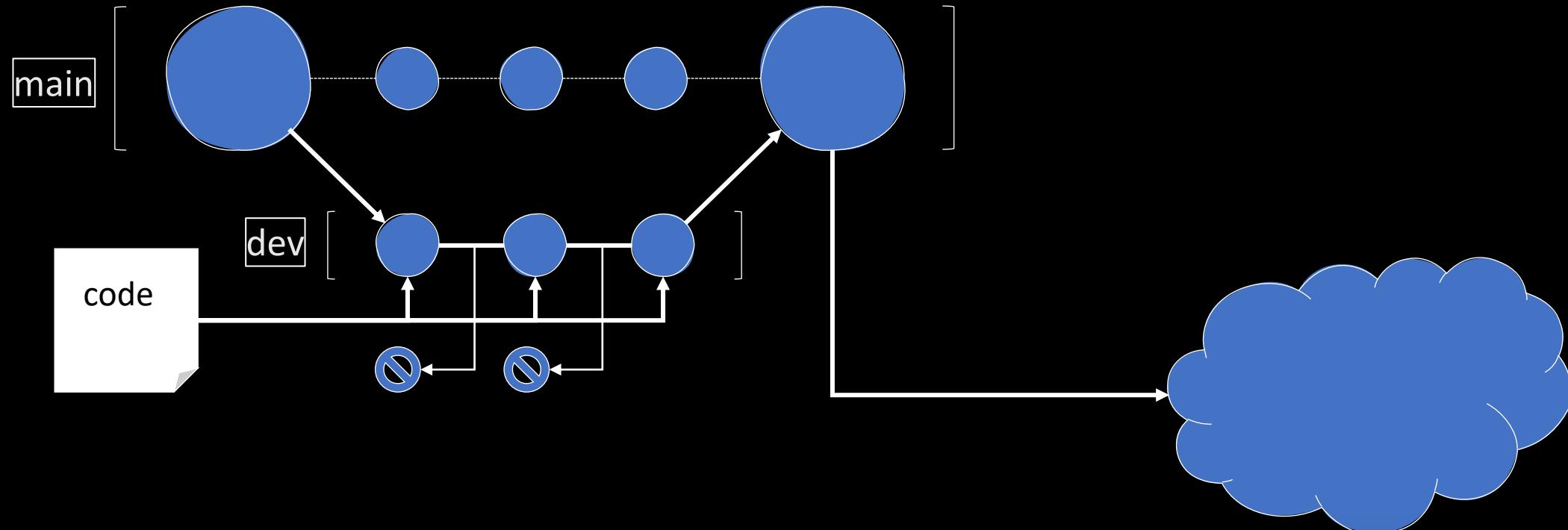
Start with 300+ pre-built tests



Plugs into continuous integration pipelines and DevOps mindset.

Supports GitHub Actions and Azure Pipelines

It makes it easier for you to do this



Write infrastructure code
and get feedback

Use the learnings to deploy
well architected solutions

Minimize blockers to
deploy to Azure

Real-world usage



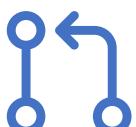
Increase success on Azure

Catch deployment and configuration issues early before release.



Meaningful guidance

Documentation provides recommendations, WAF references, and examples.



Integrate peer review

Provides feedback to enable continuous improvement and DevSecOps peer reviews.

analysis.yaml

on: push

✖ Analyze repository 35s

✖ Analyze repository
st001 failed Azure.Storage.SoftDelete. Enable blob soft delete on Storage Accounts.

✖ Analyze repository
st001 failed Azure.Storage.BlobAccessType. Storage Accounts use containers configured w...
[Show more](#)

✖ Analyze repository
One or more assertions failed.

GitHub Actions

Key points

Ready to go



Leverage over 300 pre-built rules to check Azure resources for Azure Well-Architected Framework alignment.

Native tooling

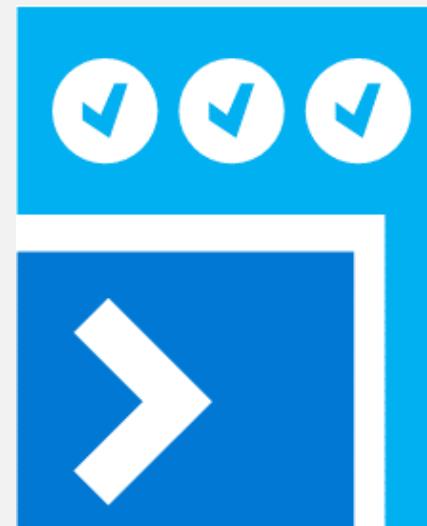


Checks are integrated with GitHub Actions, Azure Pipelines, and Visual Studio Code to provide recommendations as code is written.

Configure and customize

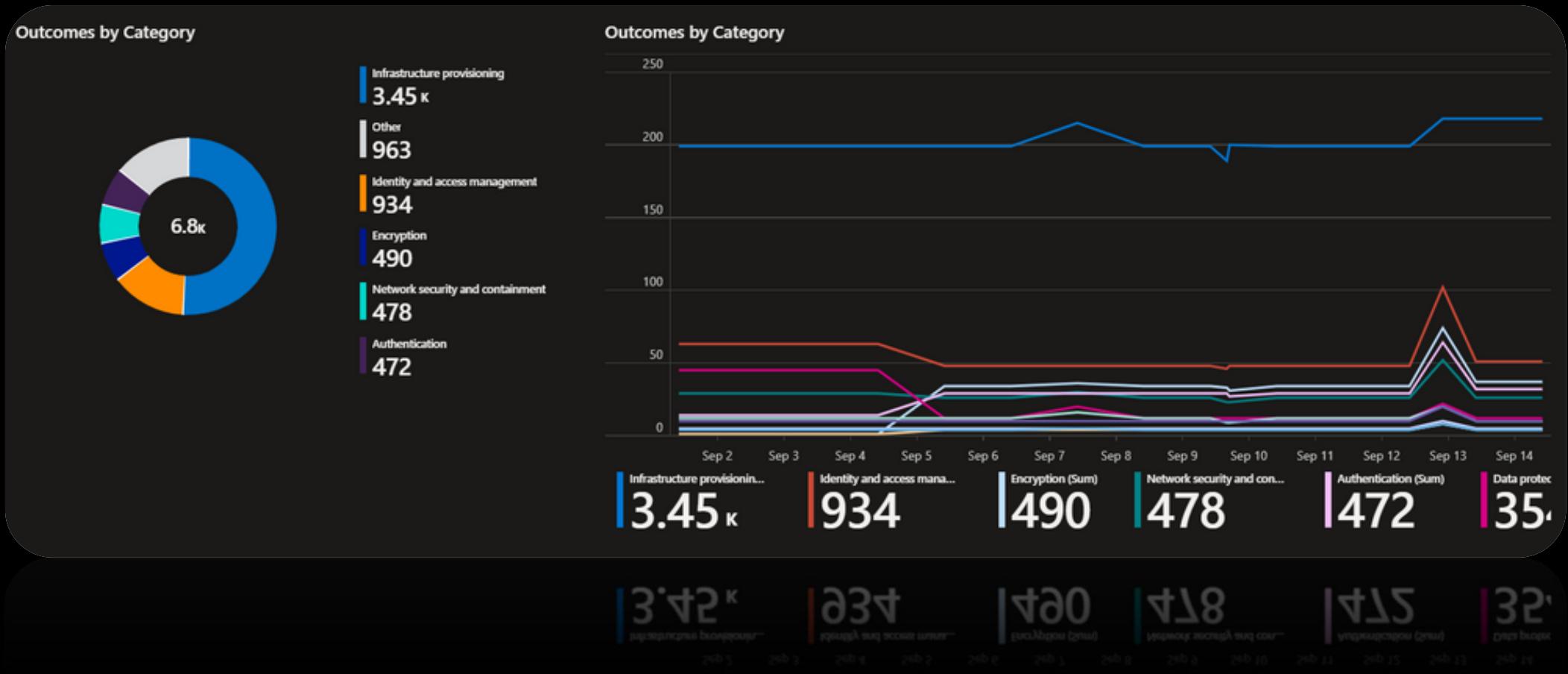


Custom rules and baselines can be created for customer specific use cases or application exceptions.



200+ releases shipped to date

Reporting in Workbooks and Log Analytics



GitHub secures your complete software lifecycle



Supply Chain



Shift security left with Advanced Security
code scanning and secret scanning



Development
Lifecycle

Platform for Security Governance

What is CodeQL?

- CodeQL is an industry-leading semantic code analysis engine
- It helps discover security vulnerabilities across a codebase through automation
- CodeQL lets us query the code as though it were data. These queries can be shared and reused by other Engineering orgs.
- Security vulnerabilities, bugs, and other errors are modeled as queries that can be executed against the code (extracted into a DB).

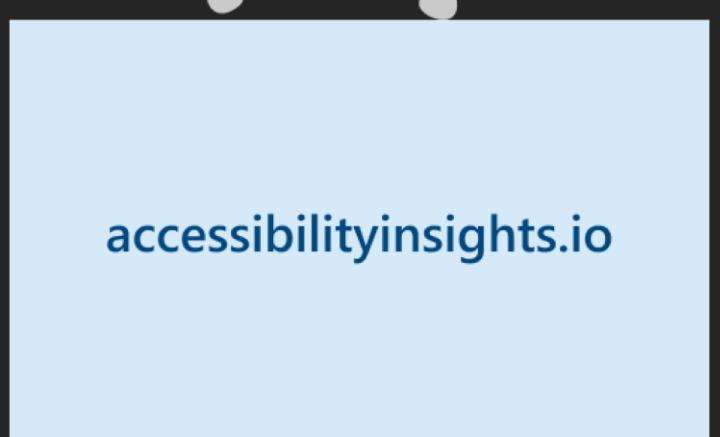
Devops CLI (aka Guardian vNext)

The image displays two screenshots illustrating the Microsoft Security DevOps ecosystem.

Github Marketplace Screenshot: Shows the GitHub Action named "security-devops-action" by Microsoft. It is described as a GitHub Action that integrates static analysis tools into the development cycle. The action is currently at version 1.4.0. A green button labeled "Use latest version" is visible. To the right, there is a "Verified creator" badge, which indicates that GitHub has verified the creator is Microsoft. Other details shown include "Stars" (8), "Contributors" (3), "Categories" (Security, Code Scanning Ready), and "Links" (microsoft/security-devops-action).

NuGet Package Screenshot: Shows the Microsoft.Security.DevOps.Cli NuGet package page. The package version is 0.118.2. It includes tabs for "Package Manager", ".NET CLI", "PackageReference" (which is selected), "Paket CLI", "Script & Interactive", and "Cake". Below the tabs, there is XML code for a package reference. A note at the bottom says: "For projects that support PackageReference, copy this XML node into the project file to reference the package." The "PackageReference" tab also contains a note: "For projects that support PackageReference, copy this XML node into the project file to reference the package."

Reserved seating



Accessibility Insights



Accessibility Insights

Existing today

Accessibility Insights for

Web

Accessibility Insights for Web

Watch 3-minute video introduction

Launch pad

- FastPass**
Run two tests to find the most common accessibility issues in less than 5 minutes.
- Assessment**
Walk through a guided process for assessing accessibility compliance.
- Ad hoc tools**
Get quick access to visualizations that help you identify accessibility issues.

Version 2.14.1 | Powered by [axe-core](#) 3.3.2

Windows

Accessibility Insights for Windows - Inspect

What to select: Element

Live Inspect

To begin inspecting a UI Automation element, hover over the element within the target application or set keyboard focus on the element. To inspect its properties, bring focus back to Accessibility Insights for Windows by pressing Shift + F9.
[Learn more about Inspect](#)

To run automated checks, hover over the element within the target application and select the Test icon . You can also focus on the element, then press Shift + F8.
[Learn more about automated checks](#)

Properties

Search

Property

Patterns

No available pattern...

Android

Accessibility Insights

Connect to your Android device

How do I connect to my device?

Connect a hardware device to your computer via USB, or run a virtual device in Android Studio or the Android Virtual Device Manager.

Detect my device

Close

Next

Existing today

Accessibility Insights for



CI/CD

Automated checks
integration in the build
pipeline



GitHub Actions

Automated checks
integrated in the PR
pipeline for GitHub



Axe-Windows

Automated accessibility
tests engine for Windows
Applications

What YOU can do:



Keep accessibility in mind in every step of the process



Download our tools at accessibilityinsights.io



Run the tools and address issues early in your product cycle



E2E

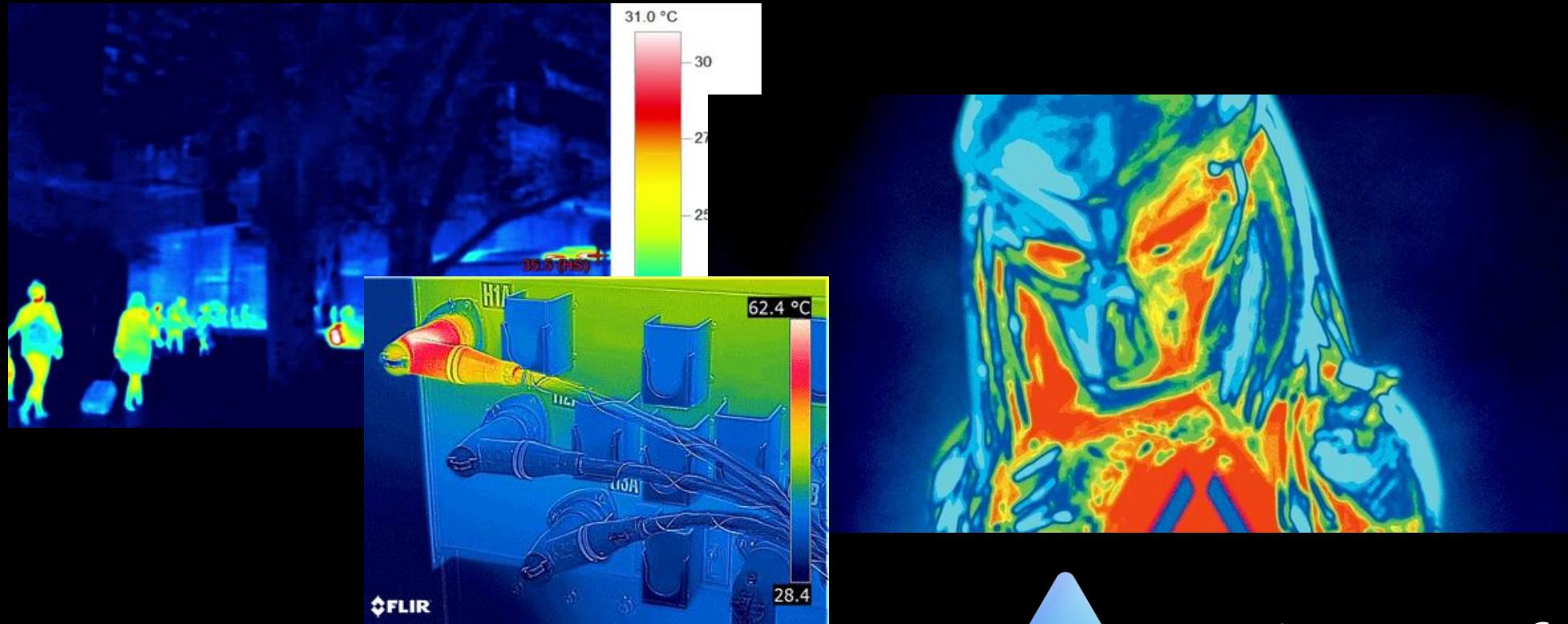


Contribute on GitHub



Spread the word and spread the love!

The Applause, the laughter, the feedback....



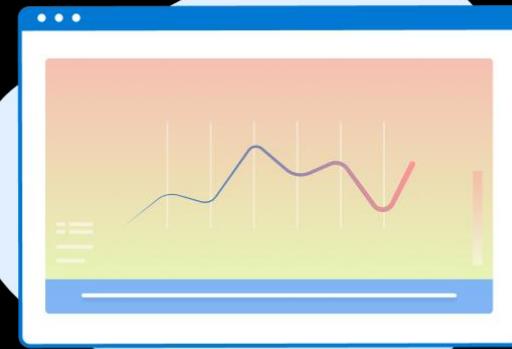
Microsoft Clarity

Why use Microsoft Clarity in testing?

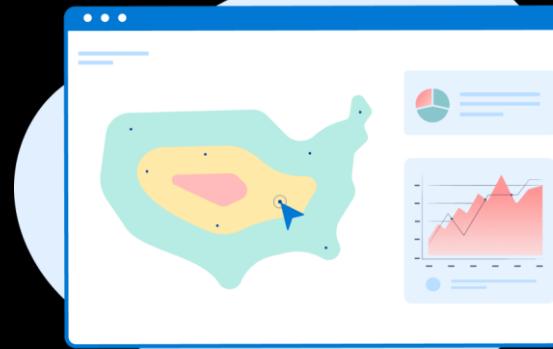
It's continuous feedback in usability.



Understand the user journey, such as the path to conversion, by replaying user sessions.



Understand engagement on your pages, including clicks, scrolls and movement through heat maps.



Gain insights into interesting user sessions to root out problems and analyze engagement by campaigns, channels and browsers.

Explore user session recordings, including path to conversion after ad clicks

Understand “why” when problems are discovered

Reason for metric movements by watching recordings to learn what part of your site works well for your users and where your users are struggling.

- Optimize path to conversion.
- Validate product changes.
- Identify target audience.
- Identify product opportunities.

The screenshot displays the Microsoft Clarity platform. On the left, a sidebar titled "Clarity Live Demo" shows a list of sessions from the last 3 days. Each session entry includes the entry point, exit point, duration, number of clicks, and device information. On the right, a large video player window shows a recorded user session. The video frame displays a heatmap overlay on a website page, with two people interacting with the site. The video controls at the bottom indicate the session is at 00:01 / 00:08, with a "Skip inactivity" checkbox checked. The overall interface is clean and modern, with a blue header and a white background.

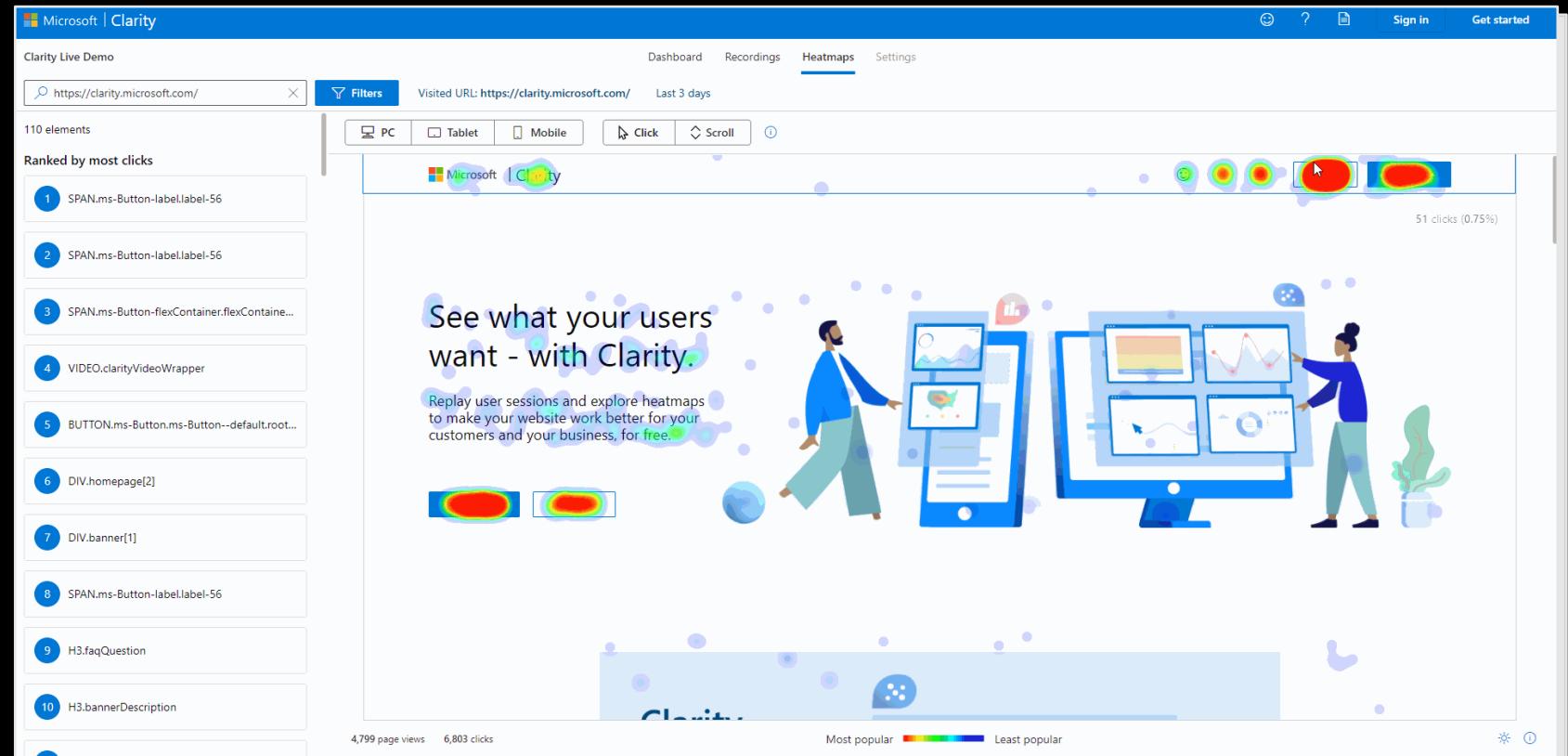
Session ID	Entry Point	Exit Point	Duration	Clicks	Device	Location		
1	Entry: clarity.microsoft.com	Exit: clarity.microsoft.com	12:38 PM	00:08	2 clicks	PC	MacOS	Ecuador
2	Entry: clarity.microsoft.com	Exit: clarity.microsoft.com	12:38 PM	00:06	1 click	PC	MacOS	United Kingdom
3	Entry: clarity.microsoft.com	Exit: clarity.microsoft.com	12:37 PM	00:15	3 clicks	PC	MacOS	Portugal
4	Entry: clarity.microsoft.com	Exit: clarity.microsoft.com	12:36 PM	00:17	0 clicks	Mobile	Android	Italy
5	Entry: clarity.microsoft.com	Exit: clarity.microsoft.com	12:36 PM	01:49	10 clicks	PC	Windows	United States

Explore heat maps to optimize ad landing pages

Visualize behavior in aggregate

Find out which part of your pages gets the most clicks and drives conversions to grow your key metrics across device types, browsers and interaction methods.

- Optimize placement of web elements.
- Detect drop-off areas.
- Show impact to regression and utilisation.
- Landing page testing.

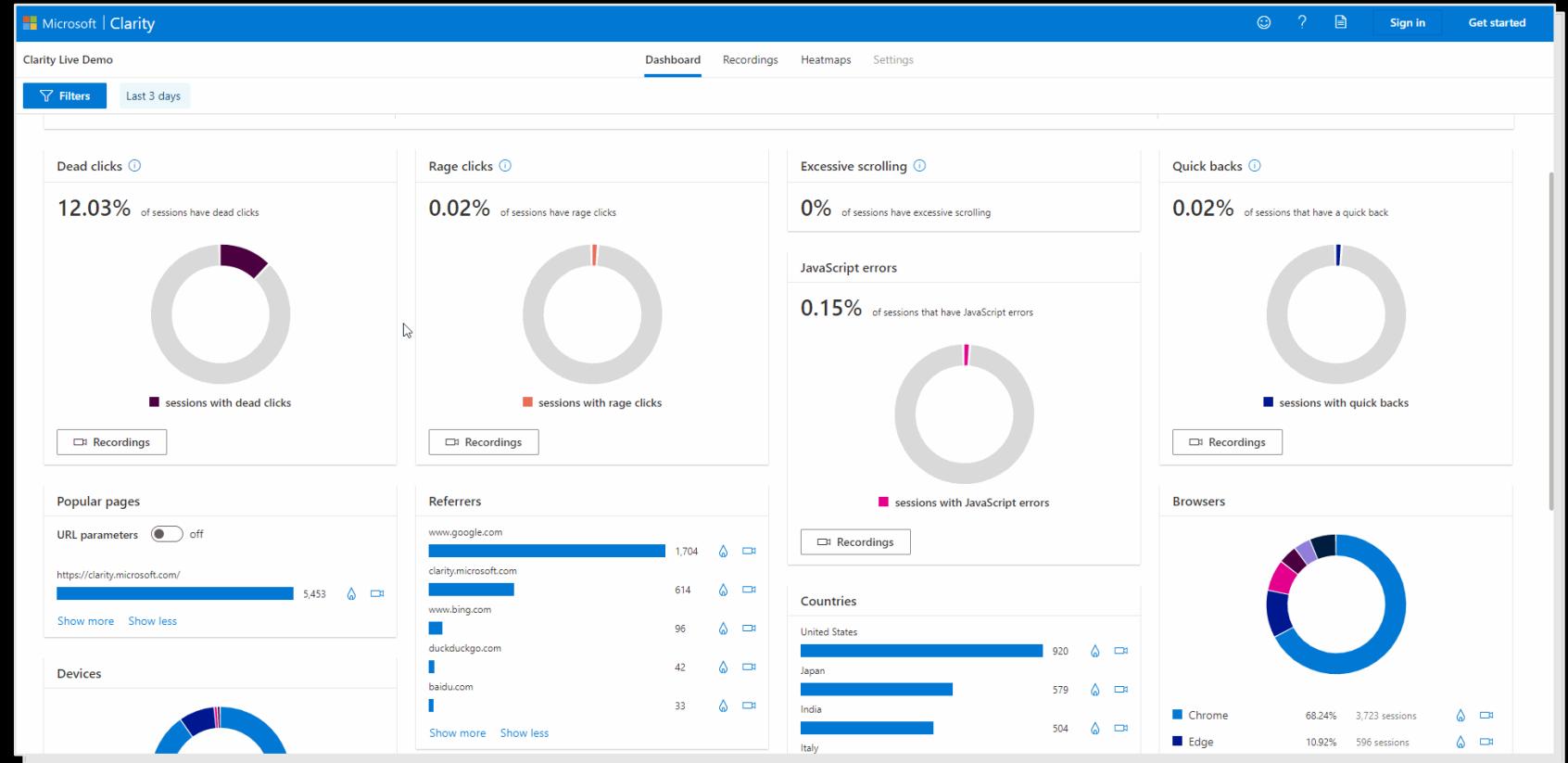


Explore dashboards and metrics to optimize user experience

Discover opportunities and eliminate problems

Make informed and data-driven decisions from the dashboard insights and drill down to specific sessions to streamline your website and remove annoying bugs.

- Quick-back clicks.
- Rage clicks.
- Dead clicks.
- Excessive scrolling.
- JavaScript errors.



Additional Clarity features

- \$ **Free service** – Use Clarity as much as you want and on as many sites as you want without being charged.
- ⚙️ **Simple setup** – Guides for major third-party platforms (ex. Squarespace, WordPress, Shopify, Wix, etc.).
- 🚦 **No sampling** – Analyze 100% of your website traffic.
- ⌚ **Nearly real time** – Get your insights in nearly real time without having to wait.
- ⚡ **Lightweight** – Clarity doesn't get in the way of your site's performance.
- 🌐 **Compatible** – Works with major web analytic tools you may already be using (ex. Google Analytics, Google Tag Manager, Adobe Tag Manager, etc.)
- 👤 **Masking support** – Protect your user privacy and mask sensitive data directly onto the client.
- 👥 **Built for scale** – Clarity can handle terabytes of data with millions of users, and with no website traffic limits.
- FilterWhere **Filters** – Supports custom filters and tags to present the data the way you want it presented.
- 🔒 **GDPR and CCPA compliant** – Compliant with General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).

So ... is that it?

Well actually



Microsoft Test Management

The screenshot displays the Microsoft Test Management interface within the Azure DevOps environment. The main area shows a summary of a recent test run, "Run 18 - E2E (Manual)", with details like completion status, owner, and run type. To the right, five circular charts provide quick insights into test outcomes, priority, configuration, failure type, and resolution. A large callout box at the bottom left illustrates the interconnected components of the testing process: Test Plan, Test Suite, Test Case, Shared Steps, and Shared Parameters, along with their relationships to Requirements, User Stories, Product Backlog Items, Bugs, and Test Results.

Test Run Summary:

- Run 18 - E2E (Manual)
- Run type: Manual
- Owner: Mike Martin
- Tested build: 20221109.1
- Release: not available
- Release Stage: not available
- Build platform: not available
- Build flavor: not available
- Test settings: Run time settings
- MTM lab environment: not available

Comments:

- No comments
- Error message

Outcome: 1 (1)

Outcome by priority: 2 (0, 1, 2)

Outcome by configuration: 17 (0, 1, 2)

Outcome by failure type: None (0, 1, 2)

Outcome by resolution: None (0, 1, 2)

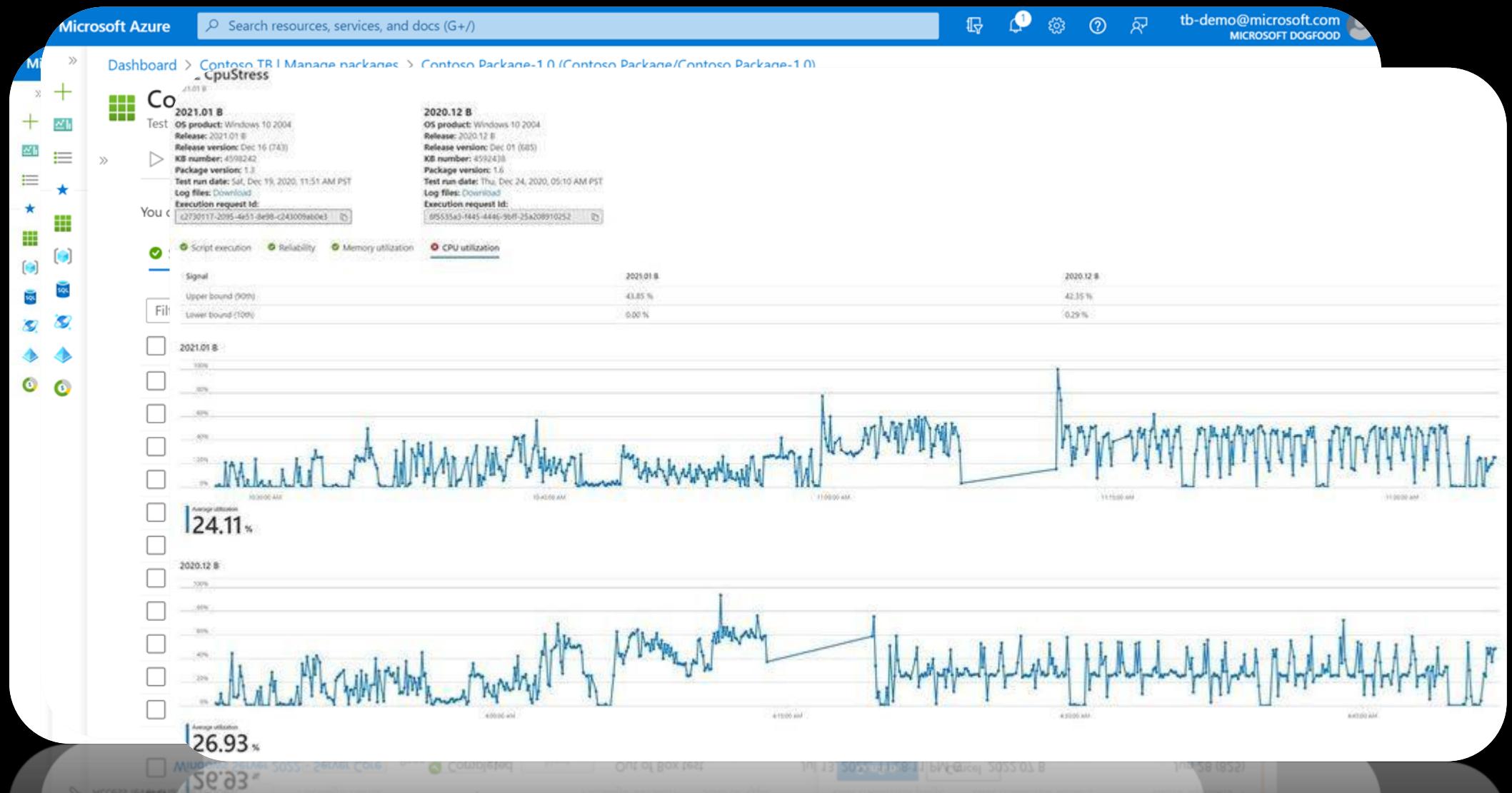
Process Flow Diagram:

```
graph TD; subgraph Requirements [Requirement (CMMI)]; RS[User Story]; end; subgraph Agile [User Story (Agile)]; RS; end; subgraph Scrum [Product Backlog Item (Scrum)]; PBI[Product Backlog Item]; end; subgraph TestPlan [Test Plan]; TS[Test Suite]; end; subgraph TestCase [Test Case]; TC[Test Case]; end; subgraph SharedSteps [Shared Steps]; SS[Shared Steps]; end; subgraph SharedParams [Shared Parameters]; SP[Shared Parameters]; end; subgraph Bugs [Bug]; B[Bug]; end; Requirements --> Agile; Agile --> Scrum; Scrum <--> Bug; Requirements --> TestPlan; TestPlan <--> TestSuite; TestSuite <--> TestCase; TestCase <--> SharedSteps; TestCase <--> SharedParams; SharedSteps <--> SharedParams;
```

Relationships:

- Requirement (CMMI) → User Story (Agile)
- User Story (Agile) → Product Backlog Item (Scrum)
- Product Backlog Item (Scrum) ↔ Bug
- Requirement (CMMI) → Test Plan
- Test Plan ↔ Test Suite
- Test Suite ↔ Test Case
- Test Case ↔ Shared Steps
- Test Case ↔ Shared Parameters

Microsoft Test Base



That's all Folks!