Code to Production with GitHub Copilot

Antonio Lamorte
Consultant @icubedsrl
antonio.lamorte@lodestar.it

Matteo Tumiati
Solution Architect @icubedsrl
matteo.tumiati@lodestar.it

aspitalia.com

Lodestar
IT'S TIME TO INNOVATE

.NET Conference
Italia 2026

> A couple of years ago I would have been very excited about a new version of my favorite framework.
>
> But these days it's just: Who cares? I am not using .net (or any other framework for that matter), only AI is. It doesn't matter if it's .NET Framework 4 or .net 11. It's all the same because AI is working with it, not me. I haven't written a single line of code in months.
>
> I don't even care about unions in C# anymore, even though I was waiting for it for years. It's all completely meaningless now.
>
> Sadly nothing matters anymore in this new AI-dominated world, except what the newest LLM is.
> I hate this stupid AI world.

A day in the life...

Upgrade frameworks and manage dependencies.
Implement features and maintain existing systems.
Review security vulnerabilities and code quality.
Maintain CI/CD workflows and cloud infrastructure.
Investigate incidents and production regressions.

Writing code is only one part of the job
The real challenge isn't just coding faster. It's delivering reliable changes.

The “standard” software lifecycle

Requirements → Design → Code → Tests → Build → Deploy

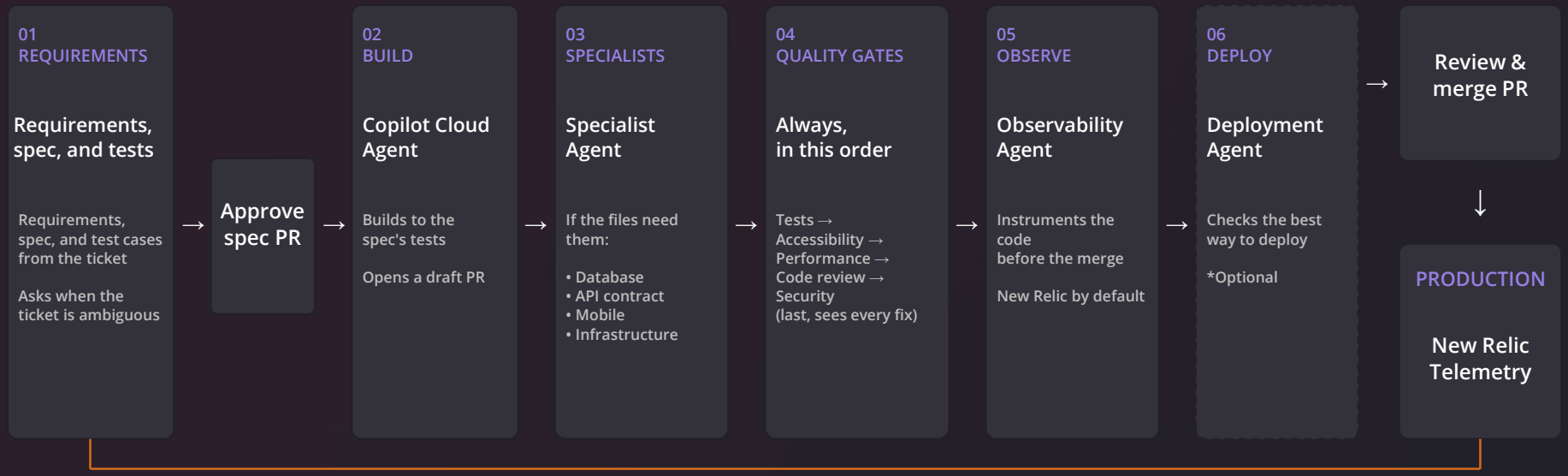
Things are… complicated ☺

	DEPLOY	RELEASE	POST RELEASE
• UNIT TESTS • FUNCTIONAL TESTS • COMPONENT TESTS • FUZZ TESTS • PROPERTY BASED TESTS • FORMAL VERIFICATION • STATIC ANALYSIS • MUTATION TESTS • COVERAGE TESTS • BENCHMARK TESTS • CONTRACT TESTS • LINT TESTS • ACCEPTANCE TESTS • REGRESSION TESTS • SMOKE TESTS • USABILITY TESTS • UI/UX TESTS • PENETRATION TESTS • THREAT MODELING • PERFORMANCE TESTS	• INTEGRATION TESTS • TAP COMPARE • LOAD TESTS • SHADOWING • CONFIG TESTS • SOAK TESTS	• CANARYING • MONITORING • TRAFFIC SHAPING • FEATURE FLAGGING • EXCEPTION TRACKING	• TEEING • PROFILING • LOGS/EVENTS • CHAOS TESTING • MONITORING • A/B TESTS • TRACING • DYNAMIC EXPLORATION • REAL USER MONITORING • AUDITING • ONCALL EXPERIENCE

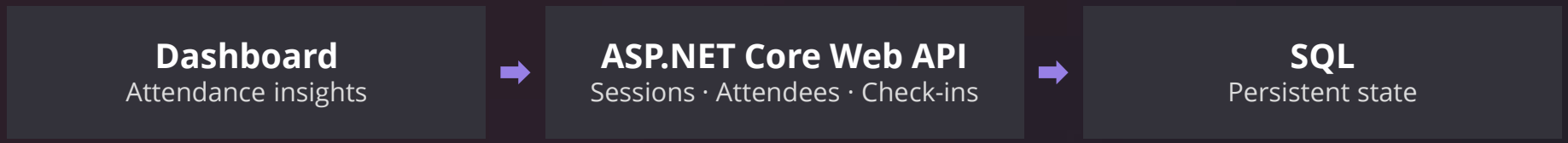
One orchestrator, specialist agents



From requirements to production...



Meet StagePass



What is does

- Session registration & capacity management
- QR code attendee verification
- Attendance statistics

Our delivery toolbox

- .NET 11
- xUnit
- GitHub
- GitHub Actions
- GitHub Copilot

Migration to .NET 11

Readiness checklist

- ✓ SDK and runtime verified
- ✓ Dependencies updated
- ✓ Breaking changes reviewed
- ✓ Regression tests passing
- ✓ Performance baseline recorded
- ✓ Rollback plan defined

```
<PropertyGroup>
  <TargetFramework>net11.0</TargetFramework>
  <Nullable>enable</Nullable>
  <ImplicitUsings>enable</ImplicitUsings>
</PropertyGroup>
```

Demo

.NET Conference
Italia 2026

What does “context” mean?

Application architecture

Business requirements

Security constraints

Monitoring runtime behaviors (logs, metrics, incidents)

SRE (pipeline failures, production changes, risks)

How do Agents work?

Using **Tools** to interact with resources

- **Capabilities** such as searching files, reading contents, editing files, running commands, or accessing external services
- Some tools are **built-in**, while others can be added

Using **SKILLS** to provide clear directives

- **Sets of detailed instructions** for a specific task saved into markdown files
- These instructions are loaded into the assistant context only when they’re needed

Skills

Skills turn engineering practices into reusable workflows.

Instructions tell agents how to behave. Skills teach them how to perform a task.

Example: checkin-testing

- Identify relevant domain invariants.
- Generate concurrent request scenarios.
- Verify duplicate handling.
- Execute the test suite.
- Report evidence and unresolved risks.

```
.github/
  skills/
    checkin-testing/
      SKILL.md
    dependency-remediation/
      SKILL.md
    release-readiness/
      SKILL.md
```

MCPs

01

Direct connection

Connects directly to resources (pipelines, clusters, monitoring systems) without manual data passing

02

Better answers

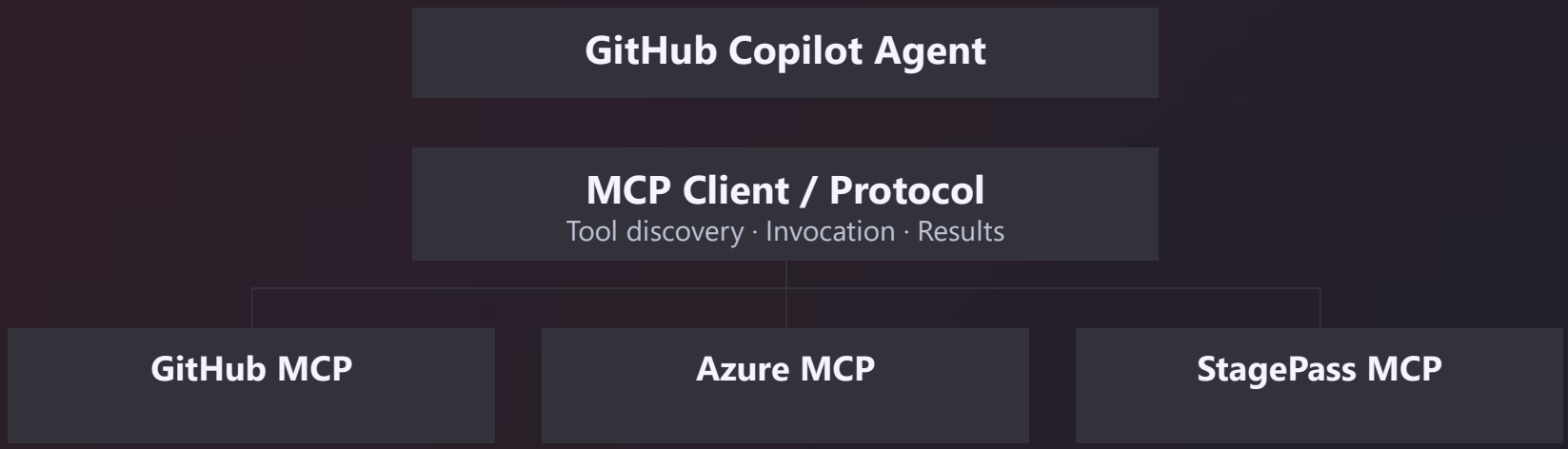
Makes AI responses more accurate and relevant in complex environments like DevOps

03

Can take action

With the right permissions, it can actually operate in these environments

MCPs for our custom app



Create your own MCPs

Design principles

- Read-only by default
- Least-privilege access
- No personal data or secrets in tool responses
- Explicit authorization for sensitive operations

Tool	Purpose
get_session_capacity	Read configured room capacity
get_checkin_metrics	Read aggregated check-in metrics
get_session_health	Read application health information

.NET 11 includes an MCP server project template

```
dotnet new mcpserver -o StagePass.Mcp
```

Demo

.NET Conference
Italia 2026

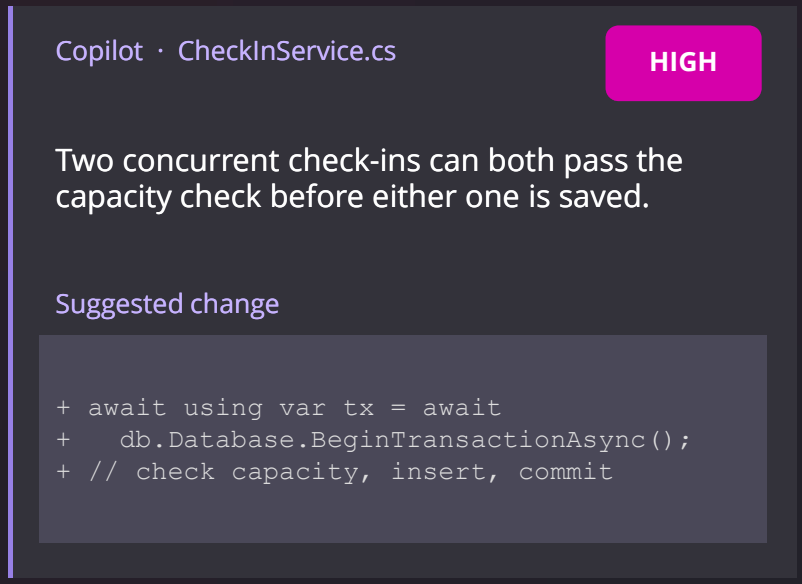
Context-Aware, Agentic Reviews

Reviews run on GitHub Actions with agentic capabilities

Full project context: analyses the whole repository, not only the diff

Review effort level: Lite, Balanced or Max

MCP servers and agent skills extend what it can check, but starts from instructions



Demo

.NET Conference
Italia 2026

Who likes sharding? ☺



Test sharding

There is no built-in --shard flag in dotnet test, but...

- List the tests with dotnet test --list-tests
- Sort them and keep only shard i of N
- Run that subset via dotnet test --filter
- Works with both VSTest and MTP runners

CI waits for the slowest shard, while xUnit still runs tests in parallel inside each one.

GitHub Actions · ci.yml

```
strategy:
  matrix:
    shard: [1, 2, 3]
steps:
  - run: >
      sharded-test
      --shard-index ${{ matrix.shard }}
      --total-shards 3
      tests/StagePass.Tests.csproj
```

Shard 1/3 | **Shard 2/3** | **Shard 3/3**

Introducing lever

You

```
/lever loop our CI is slow
```

Copilot

Loop Card, `ci-time-to-green`. No code changed yet.

OUTCOME	PR time to green, push → all required checks
INSTRUMENT	paired reruns on the same commit, read with gh run
BASELINE	`p50 31 min · p90 52 min (n=12, main)`
CRITICAL PATH	test-windows. Everything else is done 9 min earlier.
NOISE FLOOR	±2.4 min at p50 (6 unchanged runs)
KEEP BAR	~~any improvement~~ **≥ 1 min on the critical path**, flake rate not worse

Agents at scale. Scale ideas.

page file + --jobs=8 KEEP	drop git log --follow KEEP	E2E 13 shards × 4 KEEP	E2E 8 shards × 6 DROP	E2E 16 shards × 4 DROP
Dev Drive DROP	warm the compiler cache KEEP	split the lint job DROP	shallow checkout DROP	skip docs-only paths KEEP
E2E 8 shards × 8 DROP	retry only known flakes DROP	bigger runner DROP	prebuilt toolchain image DROP	typecheck first DROP
cache node_modules DROP	fewer debug symbols KEEP	parallel test binaries DROP	split the Windows job DROP	reorder job graph DROP

Let lever work for you

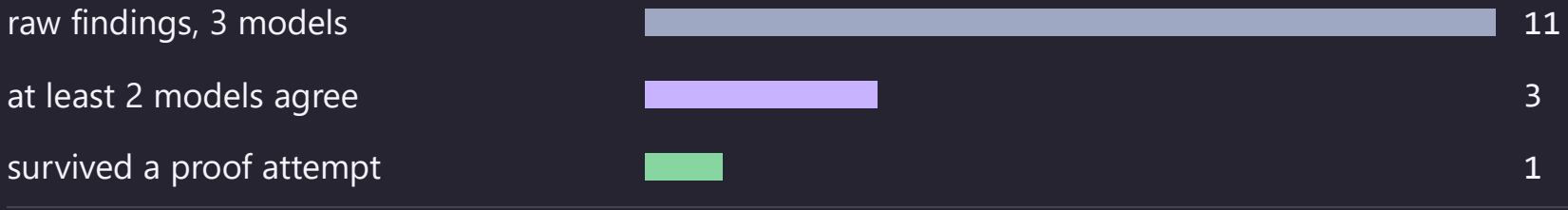
DON'T KNOW THE VERB? JUST SAY IT	`/lever our CI is slow and I'm tired of hitting rerun`	picks the verb for you
BUILD MAKE SOMETHING NEW	/lever **brief** add offline mode to the editor. ship it by tomorrow	12 words in, a runnable brief out
	/lever **refactor** src/core is 40k lines. cut it in half	three terminals, one job each
	/lever **shepherd** keep my stack green till it lands	rebase, fix, rerun, repeat
LOOP MAKE SOMETHING BETTER	/lever **loop** the editor feels janky when I scroll big files	how we'd know it worked
	/lever **arms** try the 6 ideas on the card	one agent per idea
	/lever **review** this PR	more than one model reads it
RUN MAKE IT HAPPEN WITHOUT YOU	/lever **automate** make the stack shepherd an automation for this repo	runs on a schedule
YOU TUNE LEVER TO HOW YOU WORK	/lever **mirror** last 30 days	how you actually prompt
	/lever **steer** push immediately, then verify	one correction, remembered
	/lever **autonomy** agents can merge docs PRs in this repo, nothing else	your dial, saved

Depth review with multiple models

You

```
/lever review  this PR
```

Copilot



Posted 1 inline comment: `cache key ignores the lockfile, so a dependency bump serves stale builds`. Repro attached.

Dropped 10. If your bot comments on every PR, people mute your bot.

Keep watching

You

```
/lever shepherd  keep my stack green till it lands
```

Copilot

wake 7

PR 1	green, approved	READY FOR YOU
PR 2	red: not a flake. Upload timeout too low for big files.	FIXED · PUSHED
PR 3	conflict with main	REBASED · PUSHED
reviews	3 threads: 2 fixed, 1 answered why not	RESOLVED

Next wake on a CI or review event. Nothing changes, nothing posted.

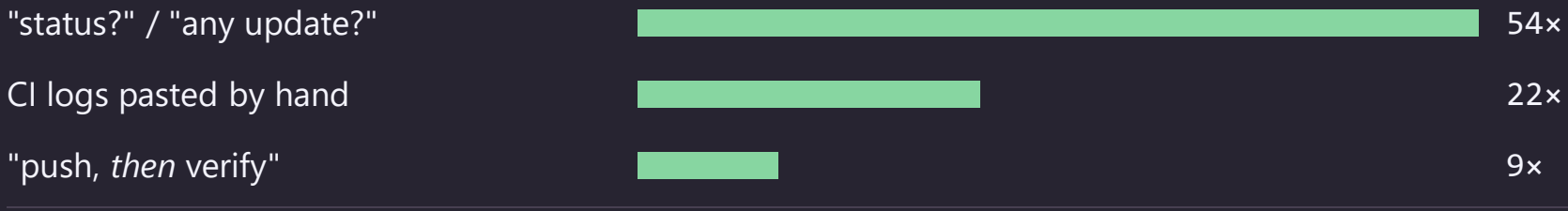
Session analyzer and steering

You

```
/lever mirror  last 30 days
```

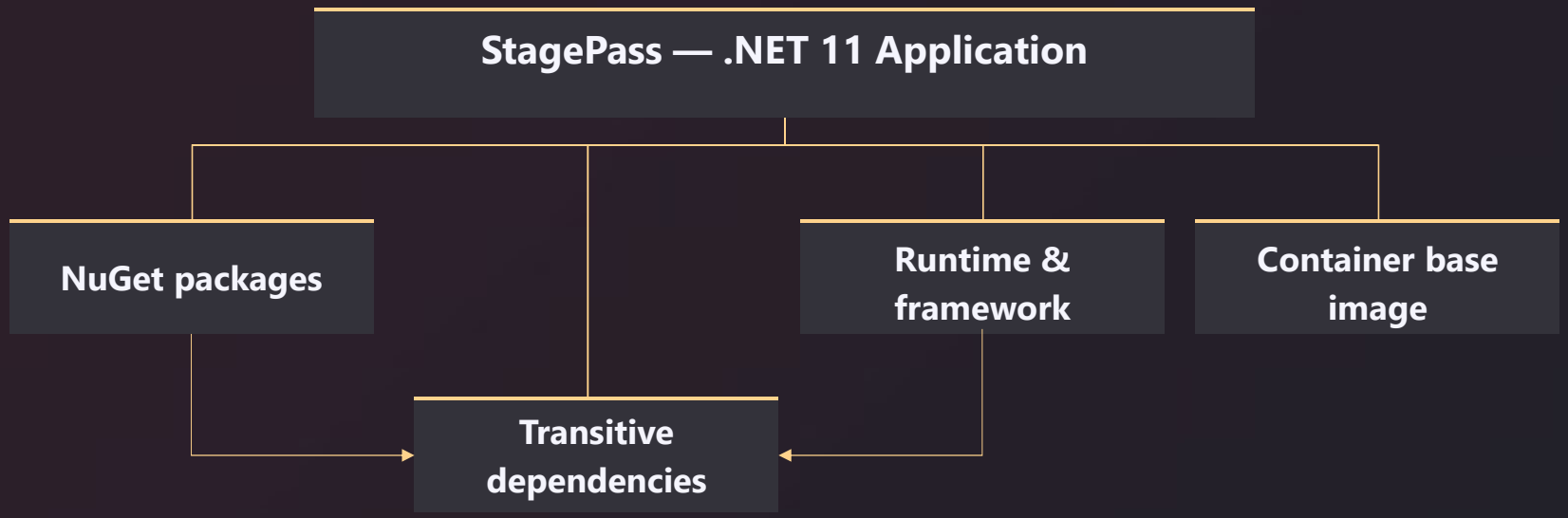
Copilot

412 prompts, 61 sessions. Extracted locally; only redacted samples went to the model.



1. You're the status page. Add `/lever status` and it reports when something changes.
2. You're the CI reader. Give it `gh run` and let it read logs itself.
3. Same correction 9 times. That's a rule: `/lever steer`.

Dependencies everywhere



NuGet Audit

Find known vulnerabilities in your dependency graph.

Security findings should answer four questions:

- Which package is affected?
- Which version is currently used?
- Is the dependency direct or transitive?
- Is a patched version available?

Known vulnerability ≠ automatically exploitable vulnerability. Investigate the advisory.

```
dotnet restore

dotnet package list \
  --vulnerable \
  --include-transitive
```

Dependabot

Dependency maintenance should be continuous.

Dependabot can help us:

- Monitor known vulnerable dependencies.
- Open security update pull requests.
- Schedule routine version updates.
- Keep package and GitHub Actions dependencies current.
- Review changes through normal repository workflows.

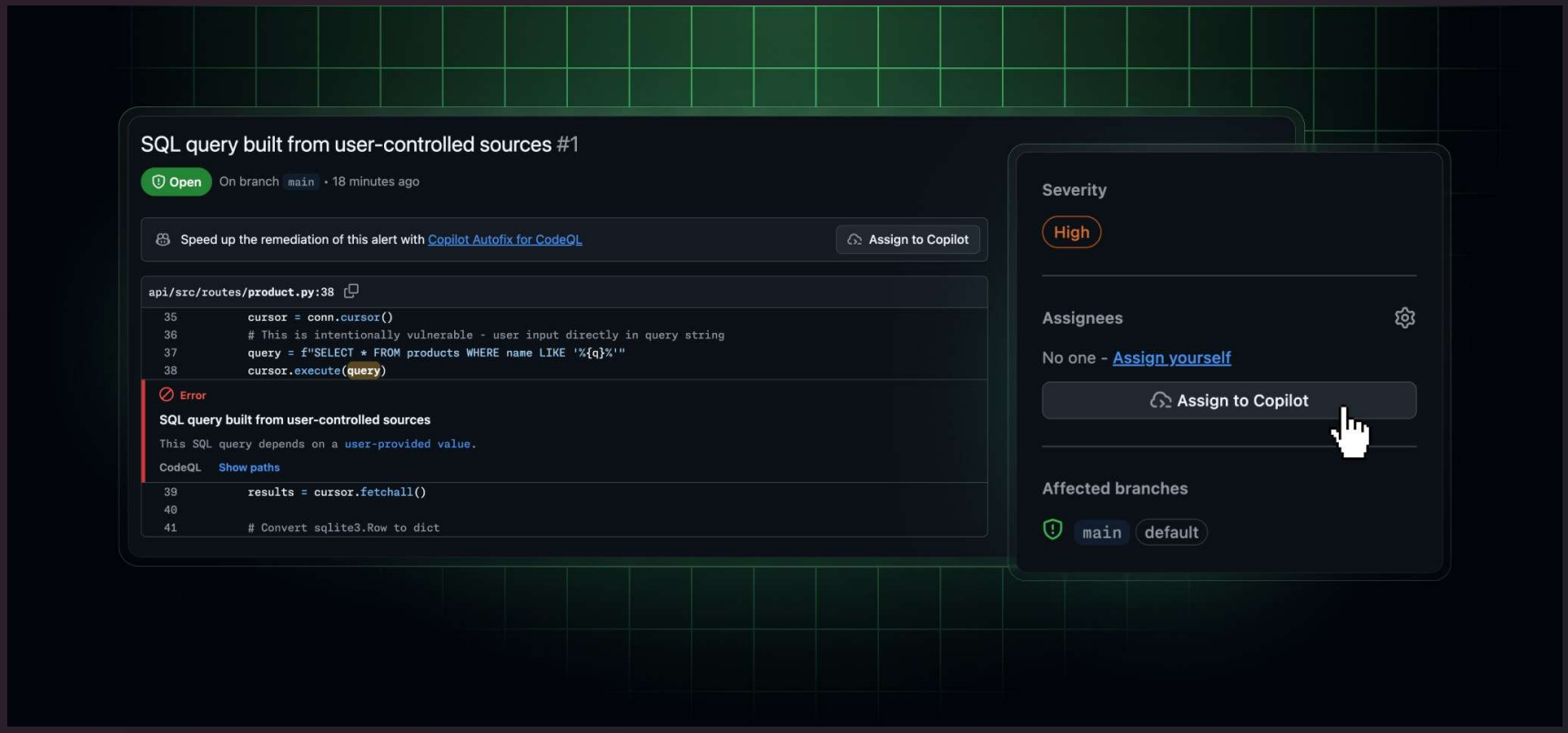
Small, frequent updates are easier to review than large, infrequent upgrades.

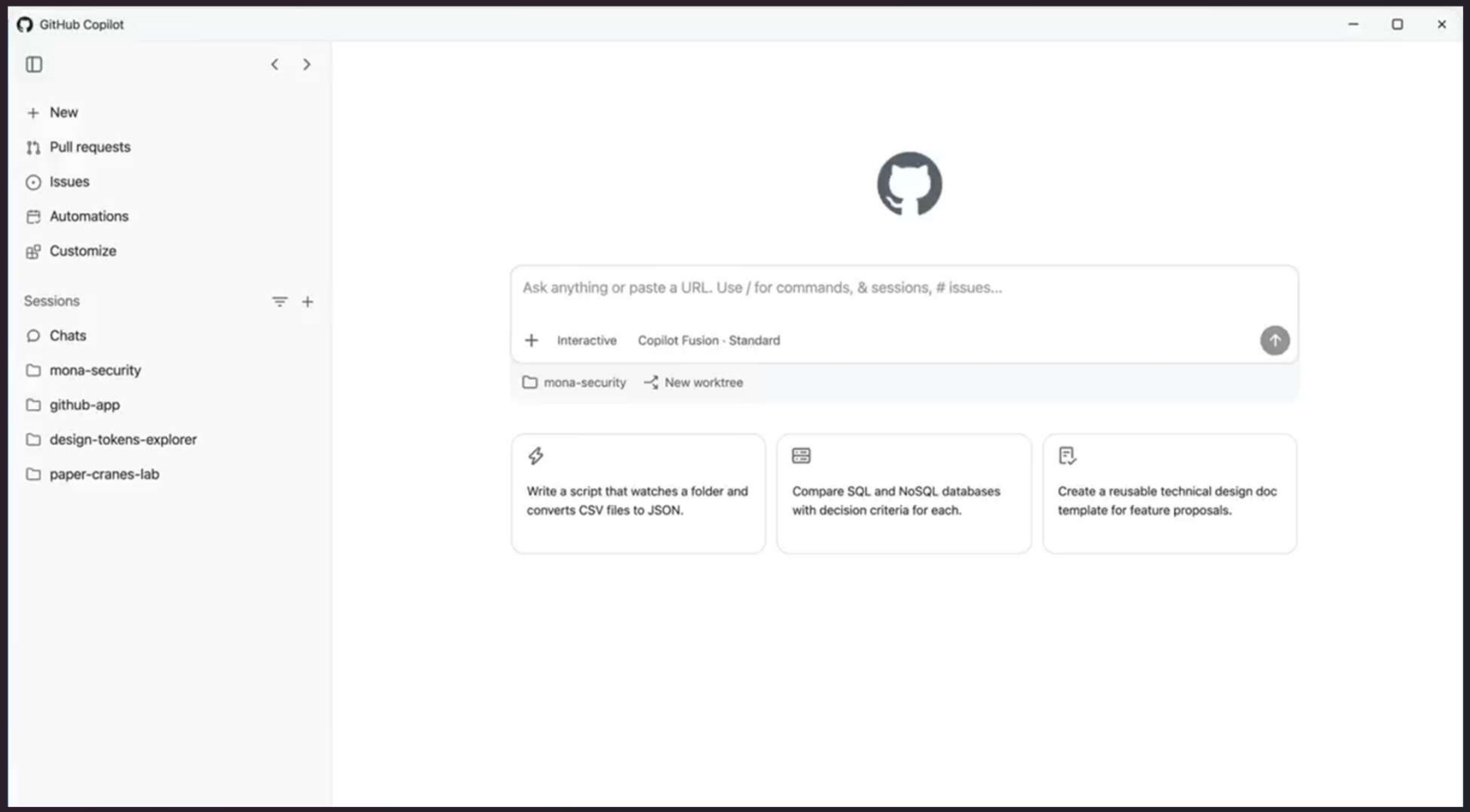
```
version: 2
updates:
  - package-ecosystem: "nuget"
    directory: "/"
    schedule:
      interval: "weekly"
```

Finding Security Issues

Dependency scanning	Code scanning
Known vulnerable packages	Potential issues in source code
Advisory-based findings	Static analysis findings
Dependency upgrade paths	Code-level remediation
Software composition risks	Application implementation risks

Copilot-Assisted Vulnerability Remediation




GitHub Copilot
New
Pull requests
Issues
Automations
Customize
Sessions
Chats
mona-security
github-app
design-tokens-explorer
paper-cranes-lab
Ask anything or paste a URL. Use / for commands, & sessions, # issues...
Interactive
Copilot Fusion · Standard
mona-security
New worktree
Write a script that watches a folder and converts CSV files to JSON.
Compare SQL and NoSQL databases with decision criteria for each.
Create a reusable technical design doc template for feature proposals.

SBOM

An SBOM (Software Bill of Materials) describes the components included in a software product.

Typical information

- Package and component identity.
- Version and dependency relationships.
- Package identifiers and licenses.
- Direct and transitive dependencies.
- Machine-readable formats such as SPDX or CycloneDX.

Repository SBOM	Artifact SBOM
Source dependency graph	Components in the built deliverable
Useful for dependency visibility	Useful for release traceability

Know what you're building. Know what you're shipping.

Image scanning

Required actions

- Publish the image to Azure Container Registry.
- Scan the image for known vulnerabilities.
- Generate an artifact-level SBOM.
- Record provenance and build metadata.
- Promote the same image digest across environments.

```
Repository: stagepass-api
Version:    1.0.0
Commit:     <git-sha>
Digest:     sha256:<image-digest>
```

Build once. Verify. Promote. Deploy. Repeat.

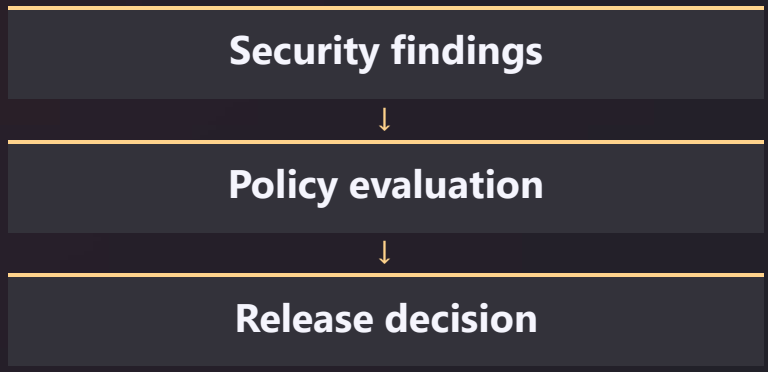
Security gates

Security becomes part of the delivery decision.

Define a Release Policy

- Required build and test checks must pass.
- Dependency and code scanning results must be assessed.
- Critical or high-risk findings require remediation or approved exceptions.
- SBOM and provenance evidence must be available.
- Production deployment requires authorized approval.

An agent may recommend. The release policy decides.



Demo

.NET Conference
Italia 2026

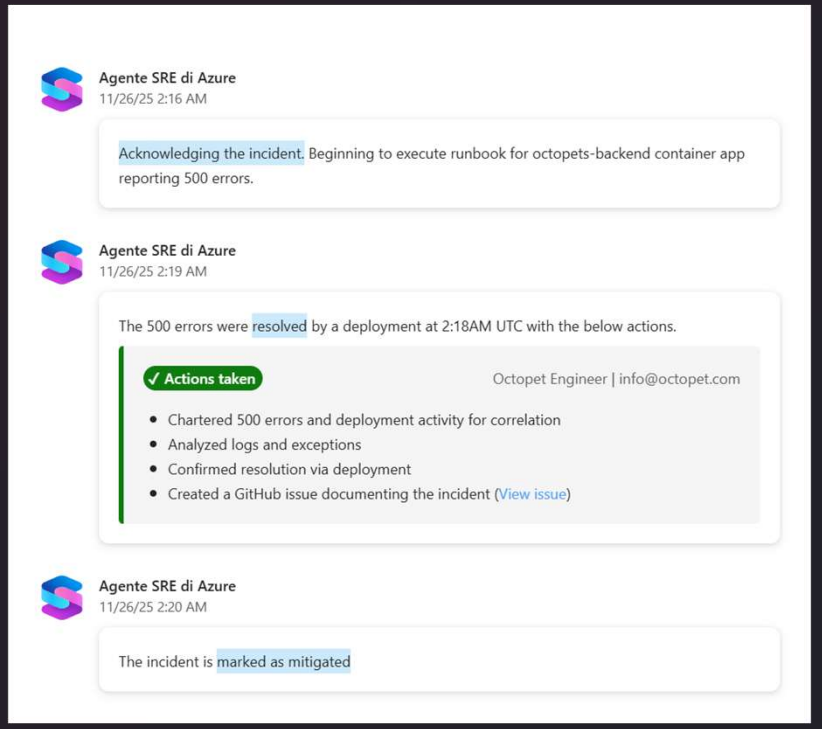
SRE Agent

Azure SRE Agent automates operational work and reduces toil, so developers and operators can focus on high-value tasks.

SRE Agent gives you an AI-driven platform to connect observability tools, incident platforms, and source code repositories, and automate the workflow end-to-end.

It remembers every investigation, learns your team's patterns, and runs background analysis even when nobody is asking questions. The agent gets smarter with every interaction.

Prioritize MTTR over actual fixes



SRE Agent coverage

Compute services: Virtual machines, App Service, Container Apps, Azure Kubernetes Service (AKS), Azure Functions, and more

Storage services: Blob storage, file shares, managed disks, and storage accounts

Networking services: Virtual networks, load balancers, application gateways, and network security groups

Database services: Azure SQL Database, Cosmos DB, PostgreSQL, MySQL, and Redis

Monitoring and management: Azure Monitor, Log Analytics, Application Insights, and Resource Manager

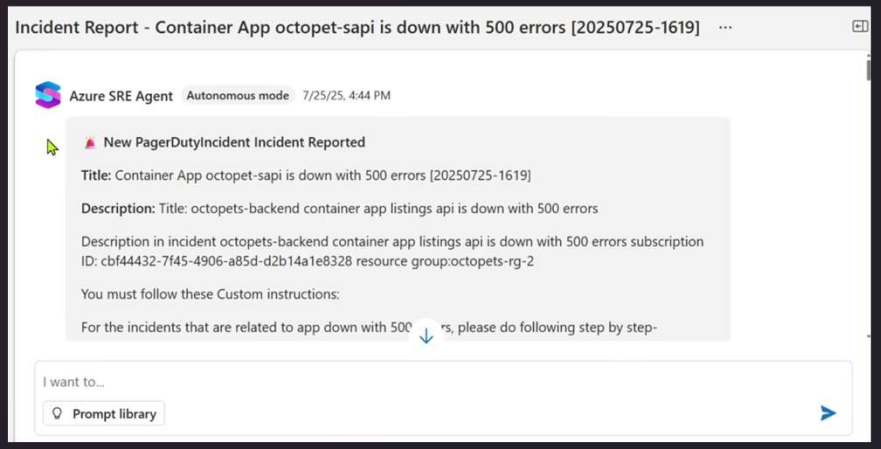
How does this work?

Built-in Azure knowledge: Preconfigured understanding of Azure services with optimized operational patterns

Custom runbooks: Execute Azure CLI commands, and REST API calls for any Azure service

Subagent extensibility: Build specialized agents for specific services like VMs, databases, or networking components

External integrations: Connect to monitoring, incident management, and source control systems




NOBODY CARES

Next steps

Il 28 gennaio 2027 arriva GitHub Enterprise & AI Day! Una giornata dedicata a GitHub, Copilot e alle nuove frontiere dello sviluppo software con l'intelligenza artificiale, tra sessioni tecniche e demo pratiche.



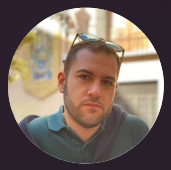
Thank you!



Antonio Lamorte

Consultant - Cloud Native Apps @ iCubed srl

I focus on implementing and maintaining DevOps solutions tailored for PLC software and .NET applications. My work integrates tools such as Kubernetes, Azure Pipelines, GitHub Actions, and cloud technologies within the Azure ecosystem to streamline processes and enhance operational efficiency.



Matteo Tumiati

Solution Architect for DevOps and Cloud Native Apps @ iCubed srl

Microsoft MVP

I'm the lead of the DevOps and Cloud Native Apps team and I'm responsible for defining technical DevOps and SRE product/architecture strategy at customers. I own GitHub and GitHub Copilot initiative at Lodestar with 750+ seats.

Slide e materiale su
https://aspit.co/netconf-26

aspitalia.com

Lodestar
IT'S TIME TO INNOVATE

.NET Conference Italia 2026