

SEP 09 – 11, 2025

CONTAINER
days
CONFERENCE

DELIGHTFUL INTEGRATION TESTS IN GO

CONTAINERDAYS CONFERENCE 2025

Manuel de la Peña

@mdelapenya

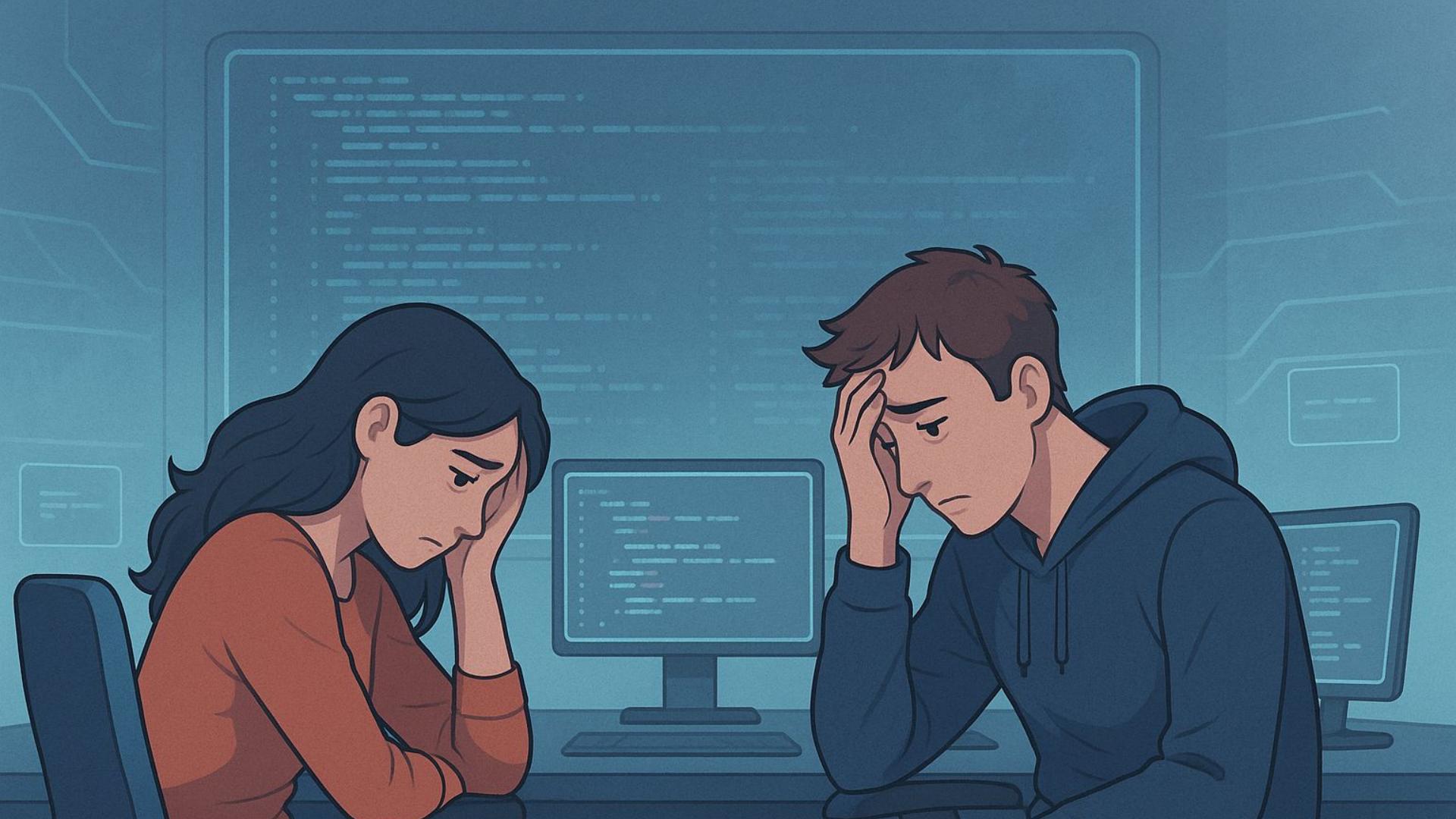




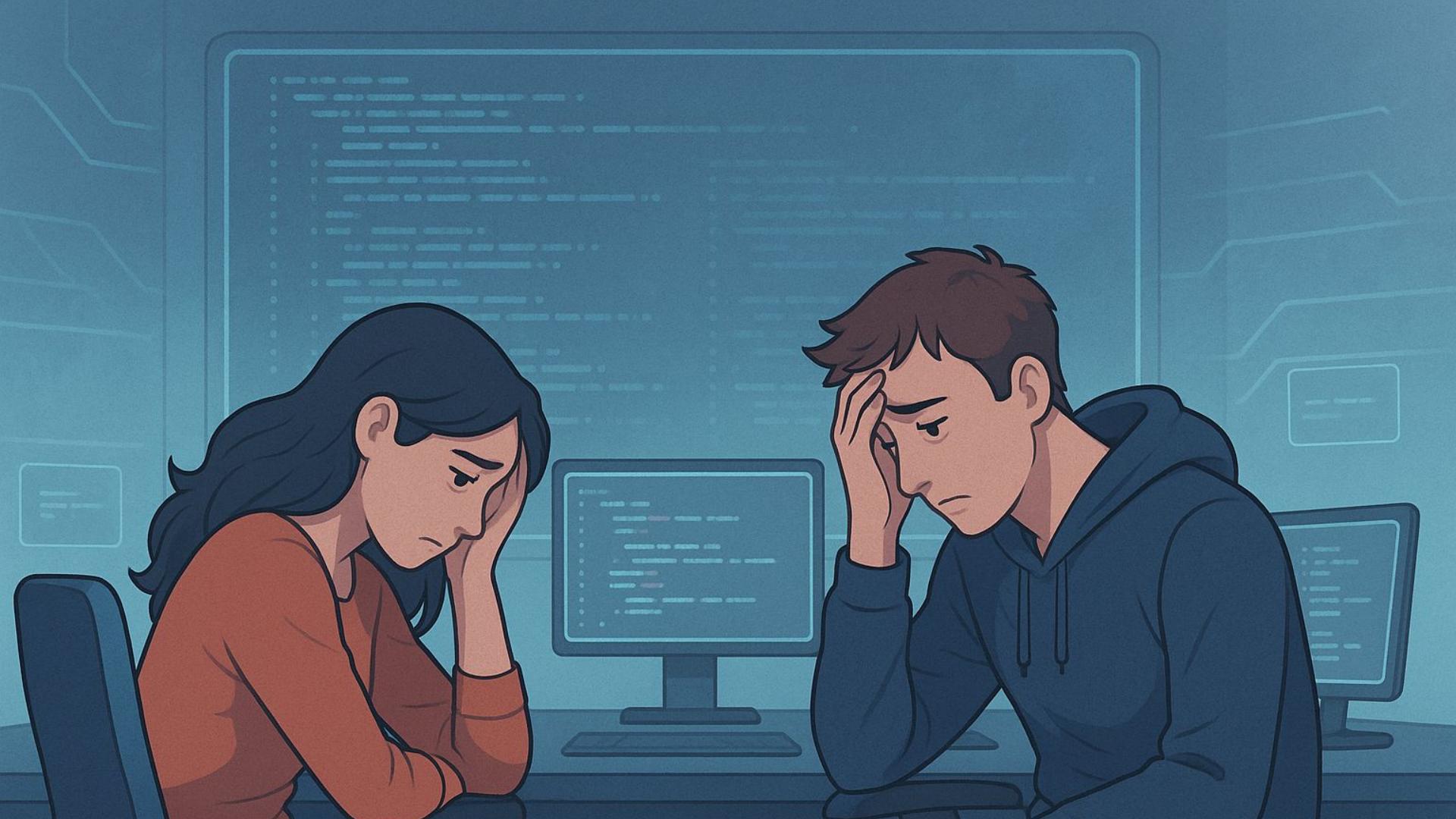












@mdelapenya

Manuel de la Peña

Staff Software Engineer @ Docker, Inc

Testcontainers Go core maintainer since 2020

Previously:

- AtomicJar, acquired by Docker
- Elastic: Eng. Productivity (Observability)
- Liferay: Core Engineer > QA Tech Lead (Cloud)
- Consultant
- Public Admin in Spain
- Hitting Keyboards since 1994



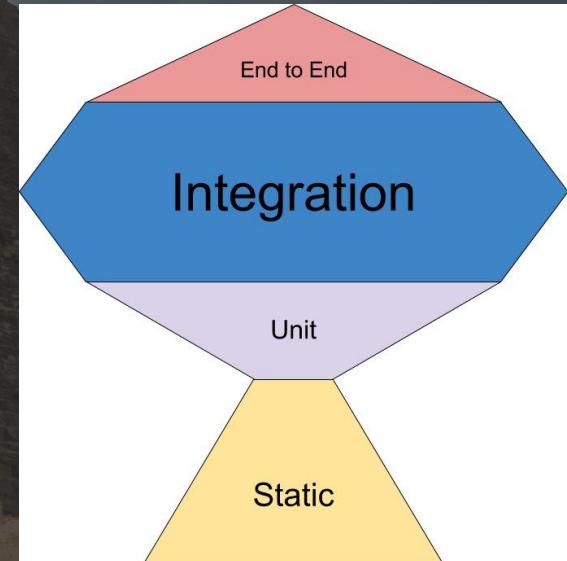
What to expect from this talk?

- ITs are fast
- Understand Developers' Issues with IT's
- Discover Testcontainers
- Pray to DEMO gods
- Taste **delightful** integration tests!



Integration tests are fast!

- Testing Pyramid is outdated
- Modern tooling enabled it
 - No need to mock



Issue #1: Reproducibility

- SQL Databases
- NoSQL Databases
- Vector Databases
- Cloud Providers
- Queues
- Authentication services
- You name it!



Issue #2: Running the tests

- GitHub
- Gitlab
- Jenkins
- ...
- Are you able to run the tests **locally?**



Issue #3: Config Management

- Versions
- Configurations
- Installed Packages
- Workers size and memory



Issue #4: Granularity

- Run one single dependency
- Run one single test
- Run tests for one single integration



Ideas??

- Avoid **mocks** when possible
- **Version** the dependencies in code
- Define dependencies **explicitly** where they are used
- Define a **local environment** that just works
- Create a “**Clone & Run**” experience
 - For new developers 
 - For the CI server 



Testcontainers for Go

- Open Source Go library ([MIT license](#))
- Developer-friendly **programmatic access** to Docker engines:
 - Create, Build, Start, Stop, Terminate
- Benefits:
 - Consistency
 - Test data isolation
 - Build speed



Consistency

- Thanks to containers!
- No need to install (Postgres|Redis|Kafka|Weaviate|...) as a process in your host
- Simplified onboarding experience
 - Better **readability!**
- The CI is another developer in the team



Data Isolation

- You choose!
 - A container per test function?
 - A container per suite?
 - A container per package?
- No more test data pollution...

REMEMBER: tests are code that need to be designed!



Build Speed

- Containers are lightweight
- Images use a caching layer system
- *NOTE:* certain technologies' Docker images are huge



Testcontainers for Go

- Random, free ports for all the exposed ports, enabling running multiple containers
- Wait Strategies to reduce flakiness:
 - ForListeningPort
 - ForLog
 - ForHTTP
 - ForFile
 - ForSQL
 - ForHealthcheck
 - ForAll



Testcontainers for Go

CONTRACT

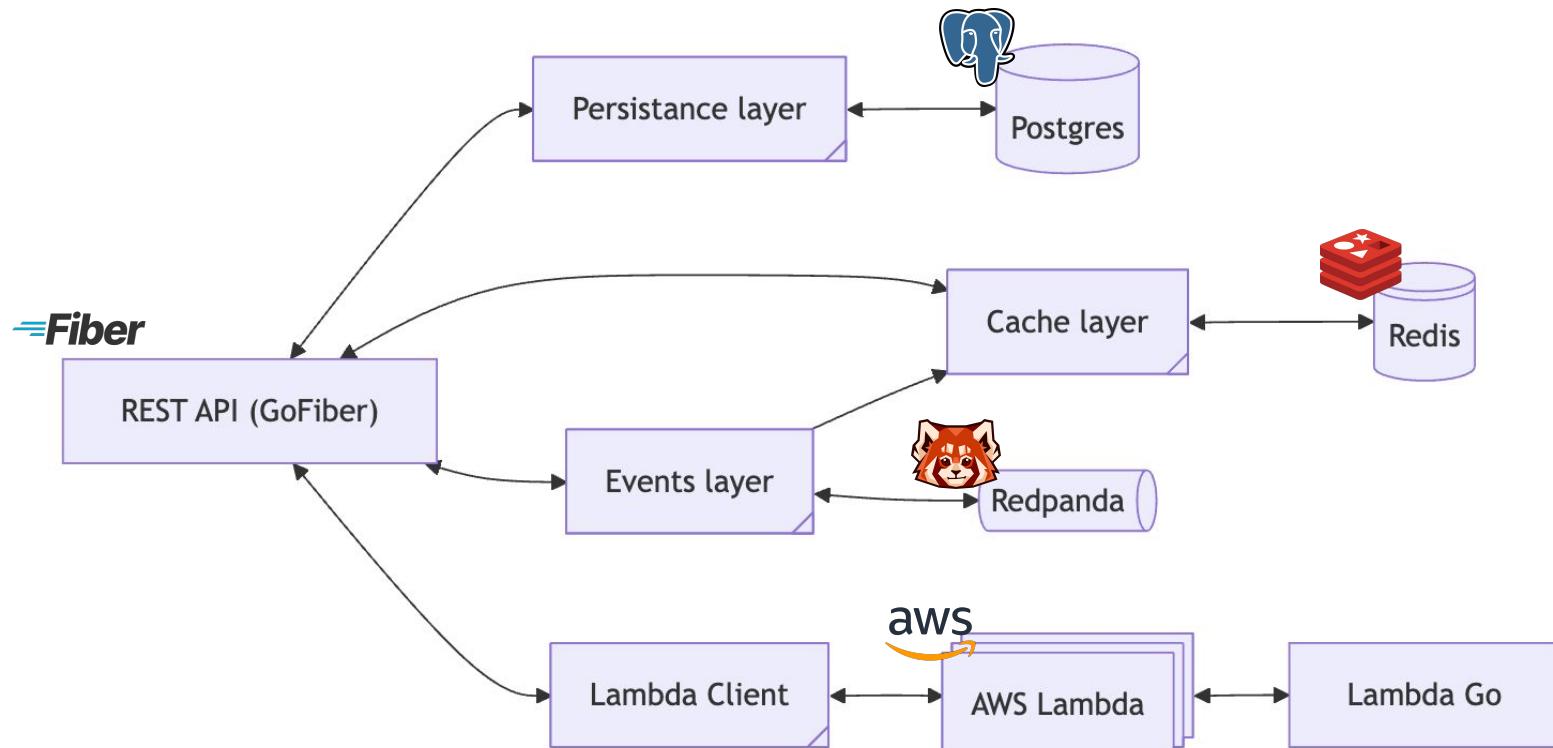
- Copy Files APIs: from and to created containers
- Custom lifecycle: execute your own code before and after hooks
- Compose support
- Garbage collection of created resources: containers, images, networks and volumes.



**Talk is
cheap...
...show me the
code**



Example App



What about Dev-Mode????

<https://www.docker.com/blog/local-development-of-go-applications-with-testcontainers/>

<https://www.docker.com/blog/go-local-dev-fiber-v3-testcontainers/>



[Testcontainers for Go Workshop](#)



[Testcontainers for Go Modules](#)



[Testcontainers for Go Docs site](#)



SEP 09 – 11, 2025

CONTAINER
days
CONFERENCE

THANK YOU!

Find me **@mdelapenya** everywhere

