Delightful Integration Tests

in Go applications

GoMAD

// Madrid Go UG

Meetup GoMAD

https://lu.ma/GoMAD

@mdelapenya everywhere

Manuel de la Peña

Software Engineer - OSS

- Docker, OSS team (2024)
- **In OSS since 2011**
  - **AtomicJar, OSS team (2022)**
    - **Core maintainer of Testcontainers for Go (2020)**
  - Elastic (2019)
  - WeDeploy/Liferay Cloud (2017)
  - Liferay (2011)
- Prev. Indra (2008)
- Prev. Civil Servant (2004)

AtomicJar is now part of Docker!

Why do we write tests?

Why do we write tests?

Fast feedback

Way to get experienced with code

Does my code works?

Test-based feedback

Pass the CI

Anything else?

Key learnings from this session

Evolution of the Testing Pyramid

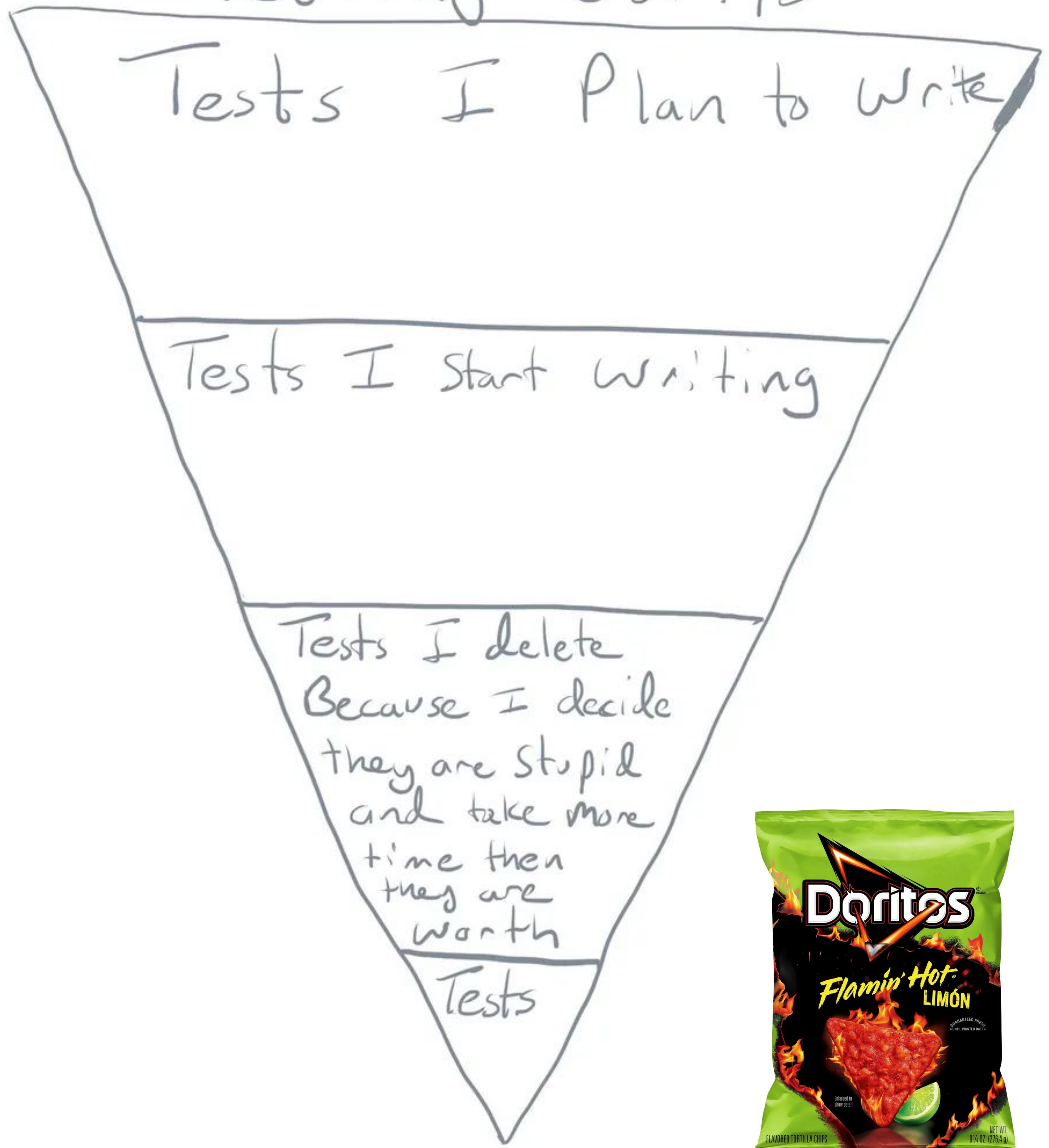
Evolution of how to set up test environments

Declare test environments as code

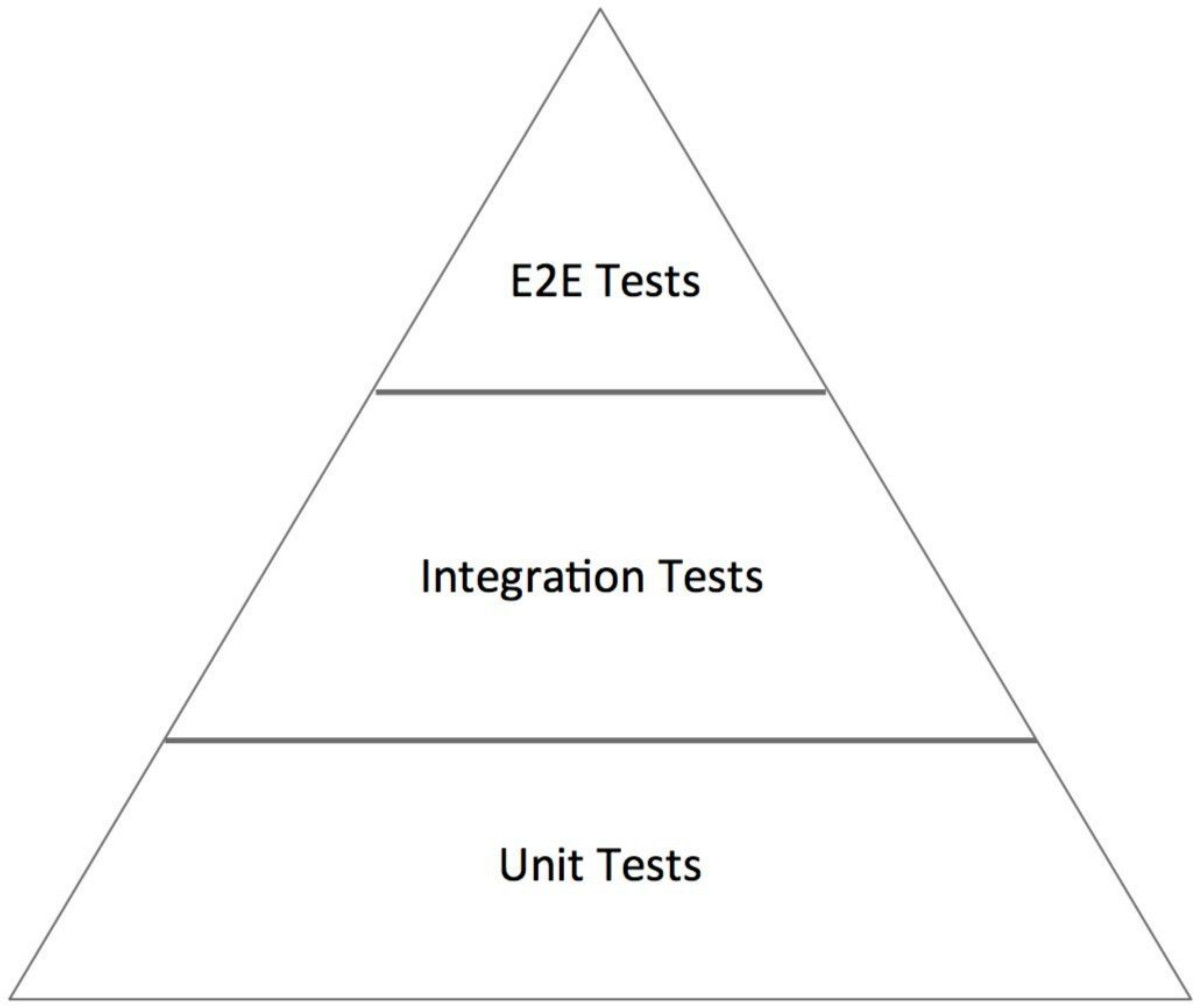
Testcontainers for Integration testing

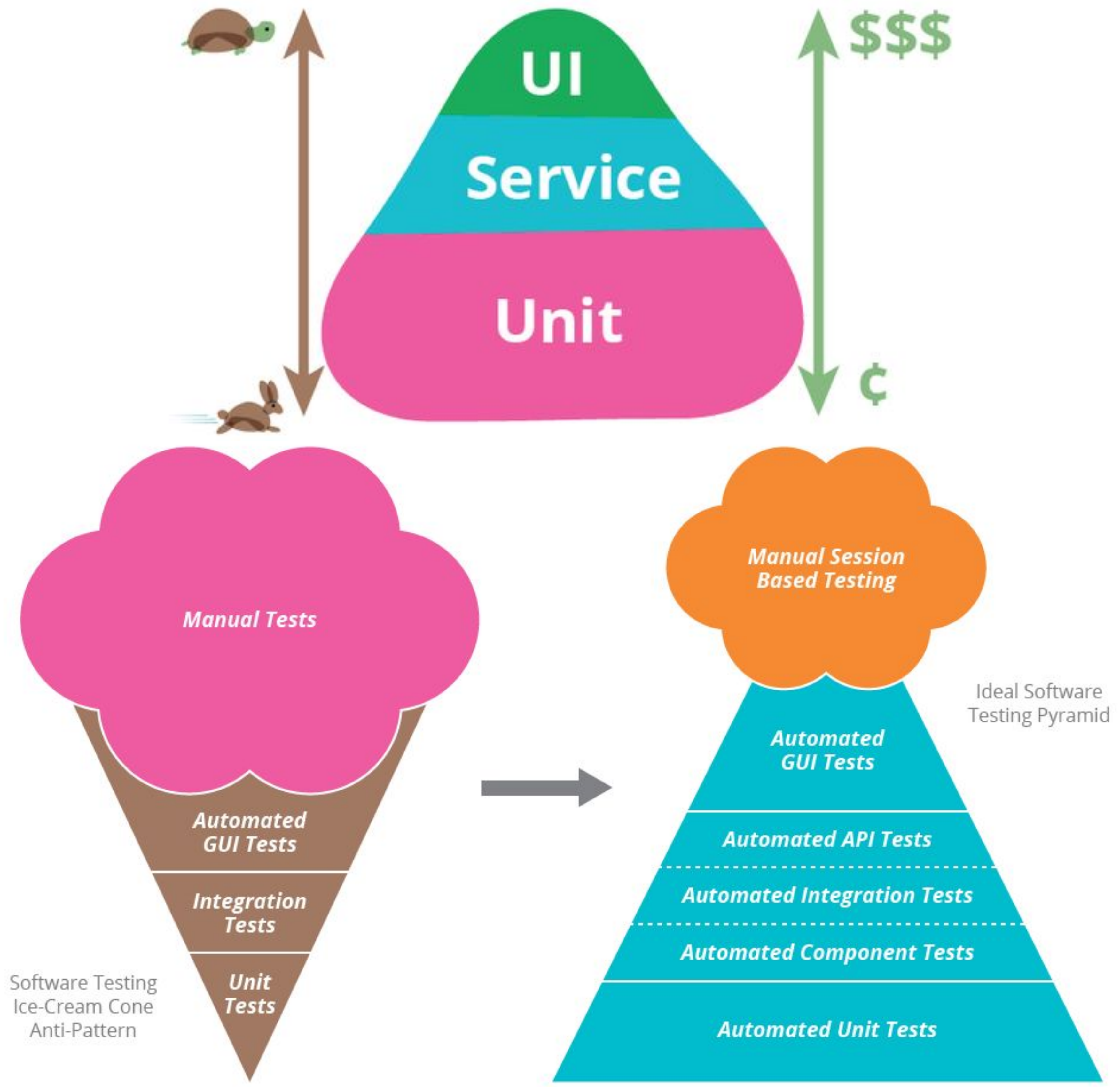
Integration tests are not hard to maintain

Testing Dorito

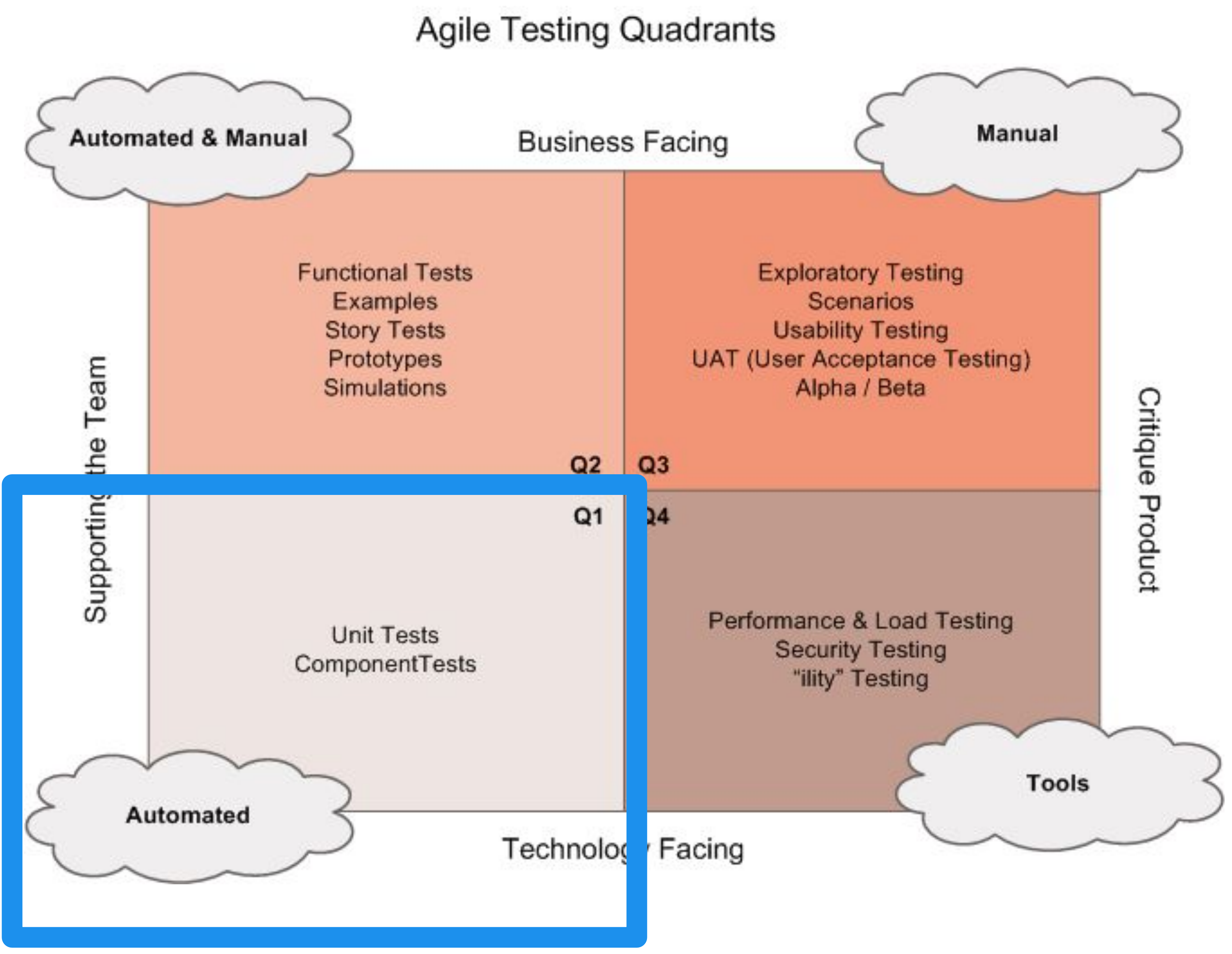

Tests I Plan to write
Tests I Start writing
Tests I delete Because I decide they are stupid and take more time then they are worth
Tests

Testing pyramid (Mike Cohn - 2003)




UI
Service
Unit
$$$
¢
Manual Tests
Automated GUI Tests
Integration Tests
Unit Tests
Software Testing Ice-Cream Cone Anti-Pattern
Manual Session Based Testing
Ideal Software Testing Pyramid
Automated GUI Tests
Automated API Tests
Automated Integration Tests
Automated Component Tests
Automated Unit Tests

Agile Testing quadrants (Lisa Crispin - 2009)

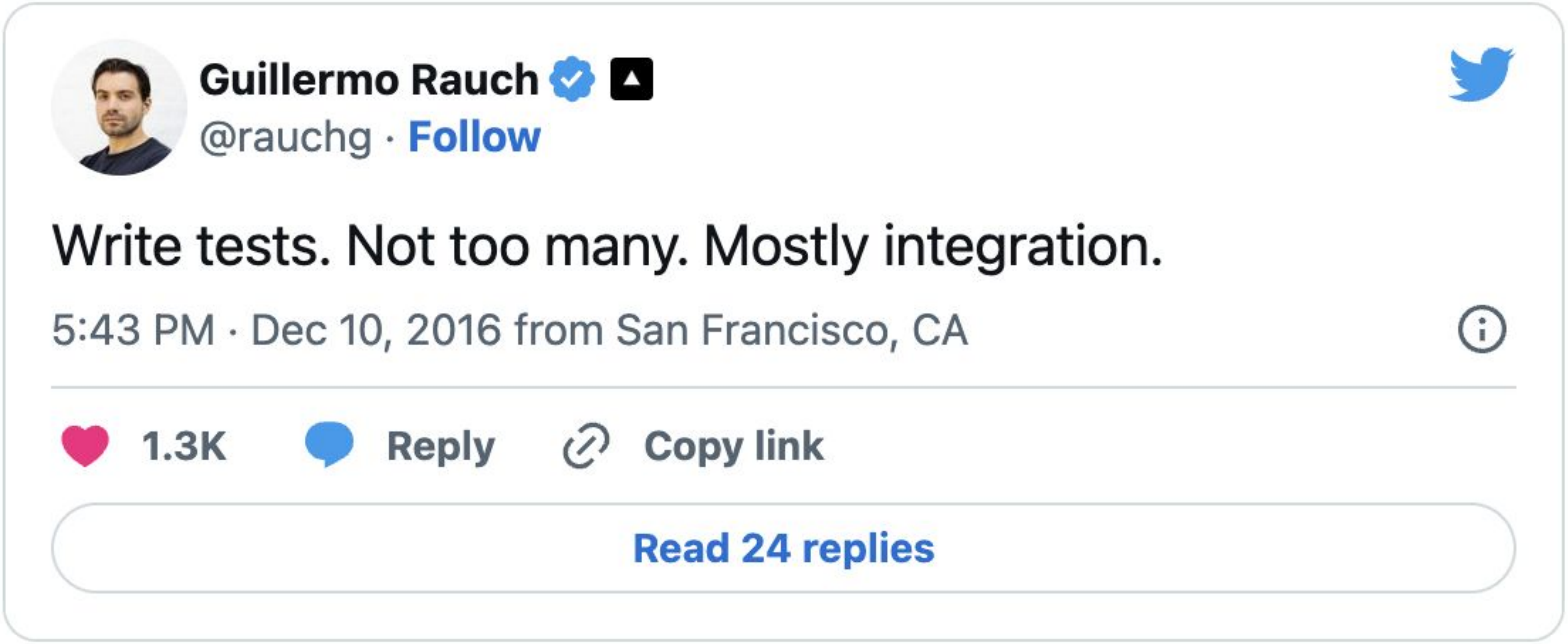


Technology-facing, Supporting the Team Tests (Q1): a major purpose is doing TDD. "These tests let the programmer **confidently** write code to deliver a story's features without worrying about making unintended changes to the system".

"Programmer tests are normally part of the automated process that runs with every code check-in, giving the team **instant, continual feedback** about their internal quality".

"Database access usually consumes lots of time, so consider using **fake objects**, where possible, to replace the database, especially at the **unit** level".

Twitter (Guillermo Rauch - 2016)


Guillermo Rauch
@rauchg · Follow

Write tests. Not too many. Mostly integration.

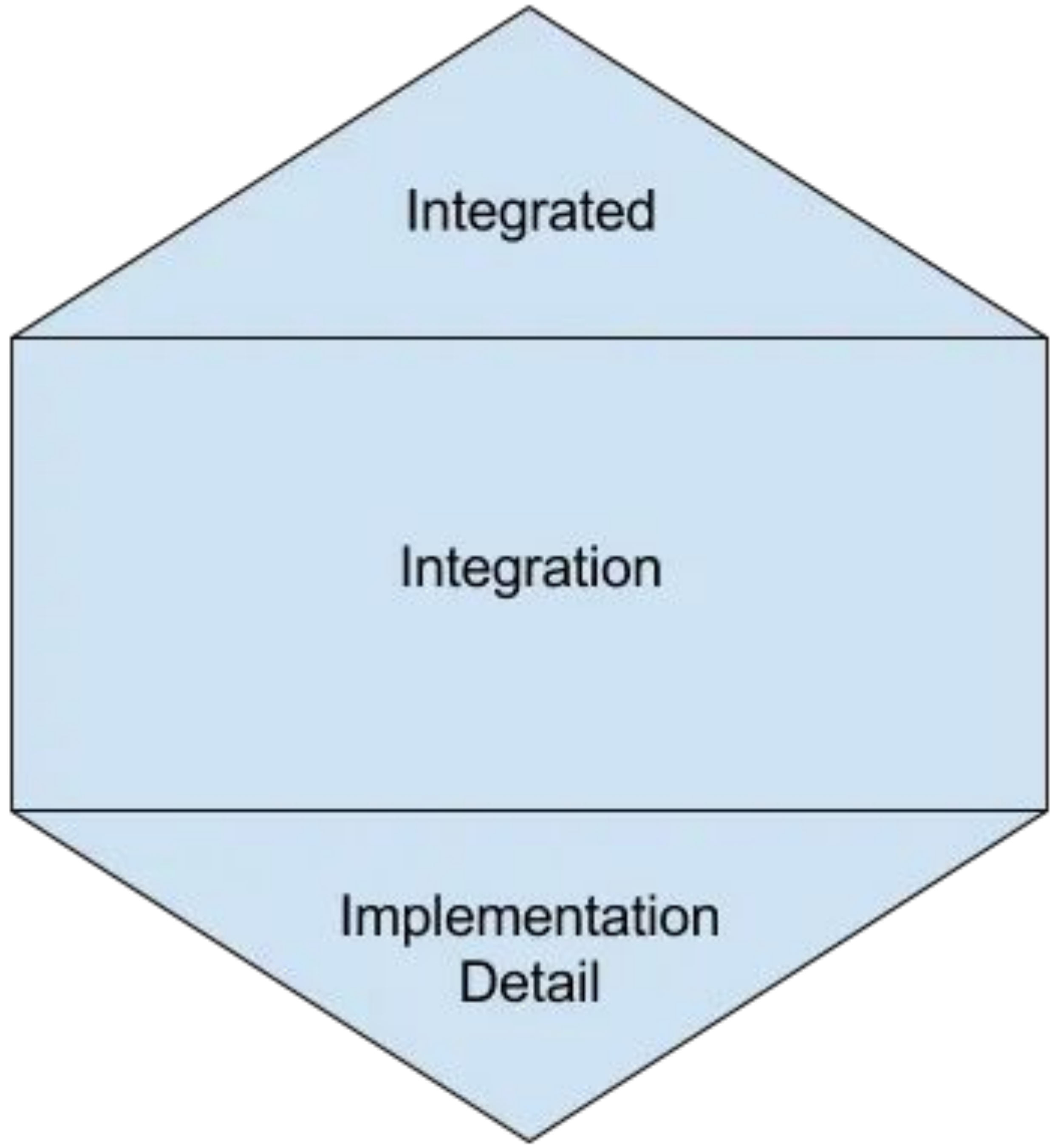
5:43 PM · Dec 10, 2016 from San Francisco, CA

1.3K Reply Copy link

Read 24 replies

https://twitter.com/rauchg/status/807626710350839808

Testing Honeycomb (by Spotify - 2018)



Integrated tests (*fragile!!*): a test that will pass or fail based on the correctness of another system.

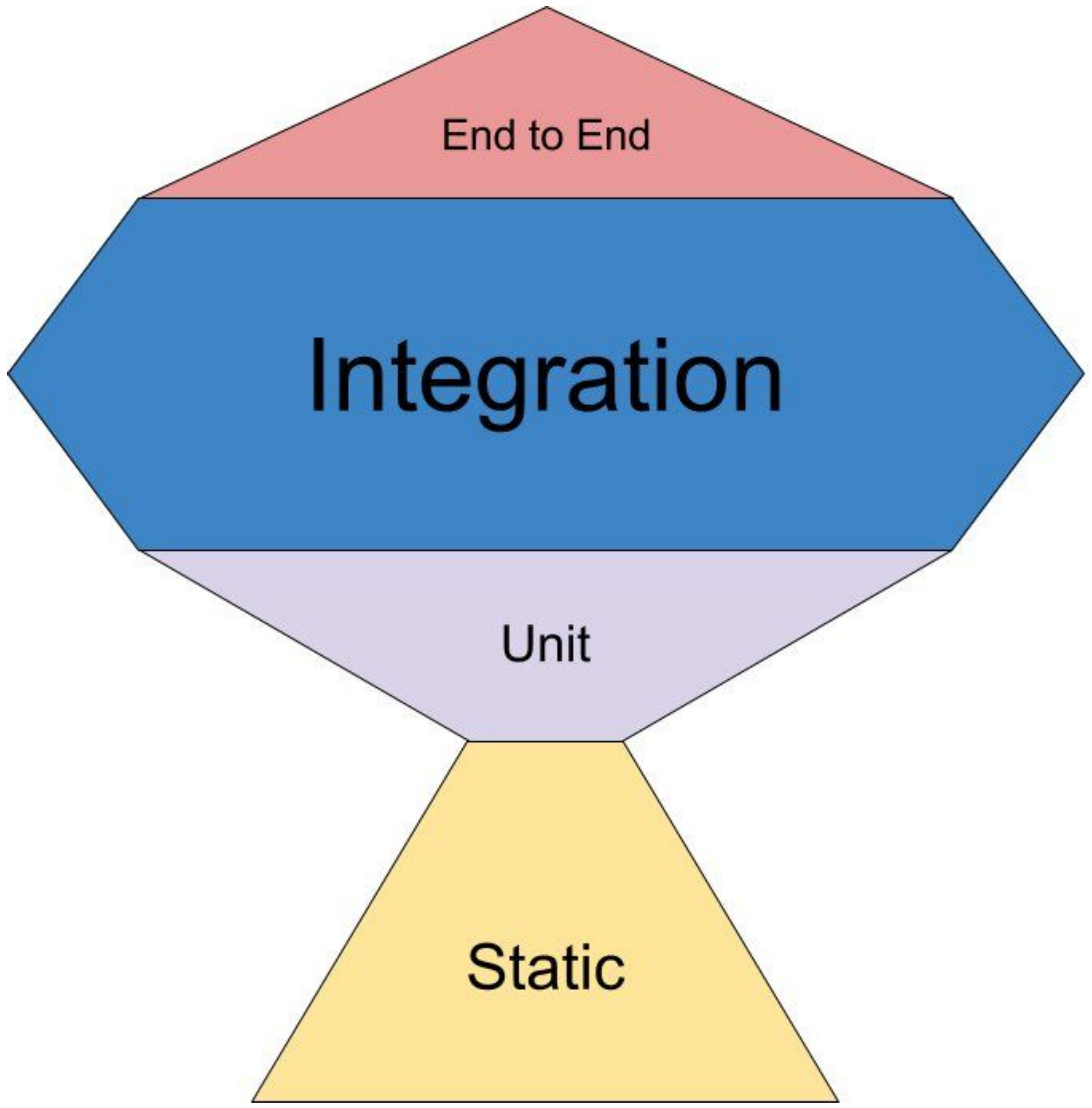
- Spin up services in a local testing environment
- Test against services in a shared environment

Aim for integration tests: verify the correctness of services in a more isolated fashion while focusing on the **interaction points** and making them very **explicit**.

- Refactor internals without touching any tests (increased maintainability)
- Replace backing services (e.g. DBs) without mocking
- Trade-off: from milliseconds to a few seconds

https://engineering.atspotify.com/2018/01/testing-of-microservices/

Test Trophy (Ken C. Dodds - 2019)

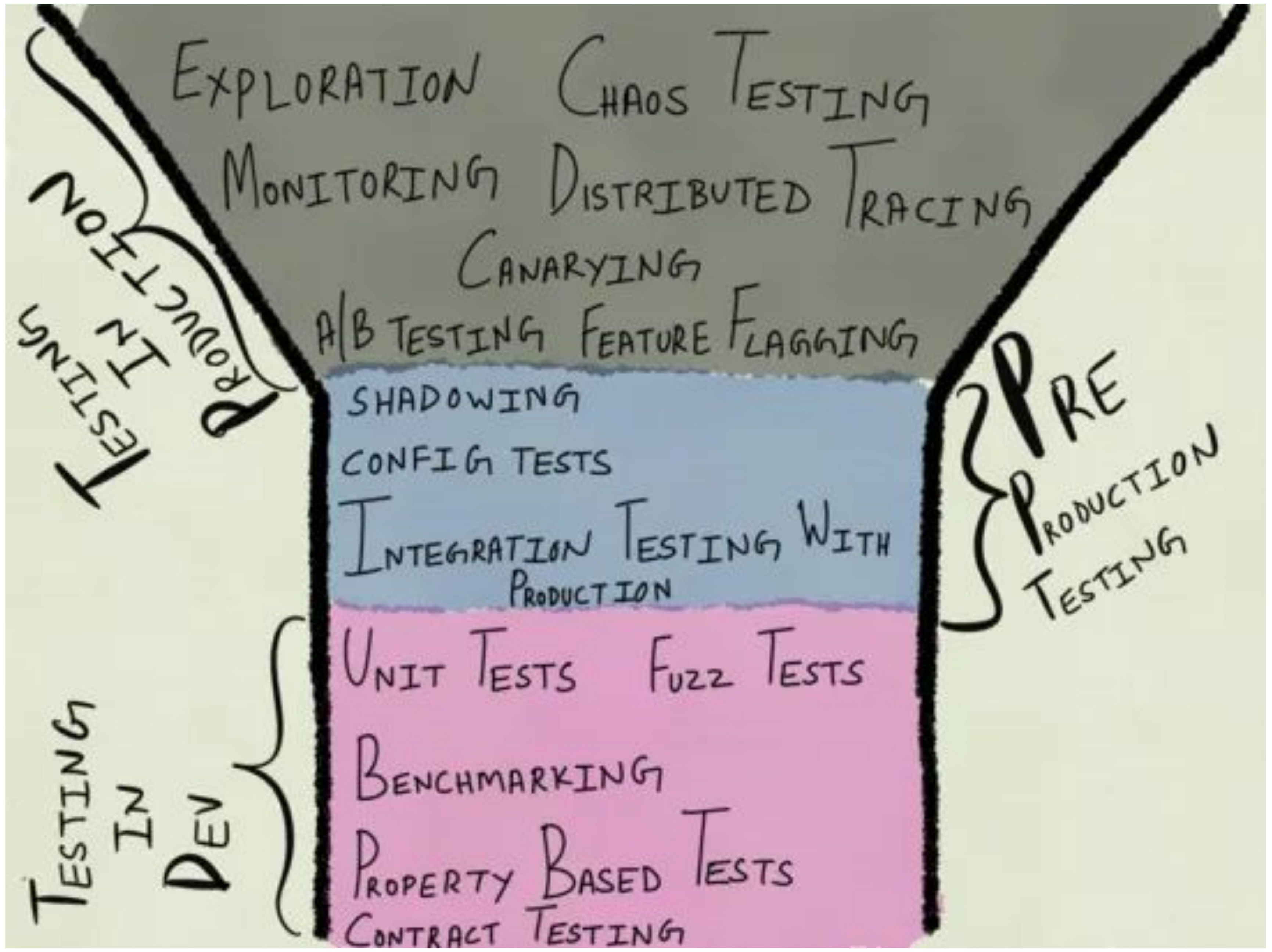


“The line between integration and unit is a little bit fuzzy. (...) the biggest thing you can do to write more integration tests is **to stop mocking so much stuff**. When you mock something *you're* ***removing all confidence in the integration*** *between what you're testing and what's being mocked*”.

“**The biggest challenge is knowing what to test** and how to test it in a way that gives **true confidence** rather than the false confidence of **testing implementation details**”.

https://kentcdodds.com/blog/write-tests

Testing funnel: Step Up rule (Cindy Sridharan - 2017)

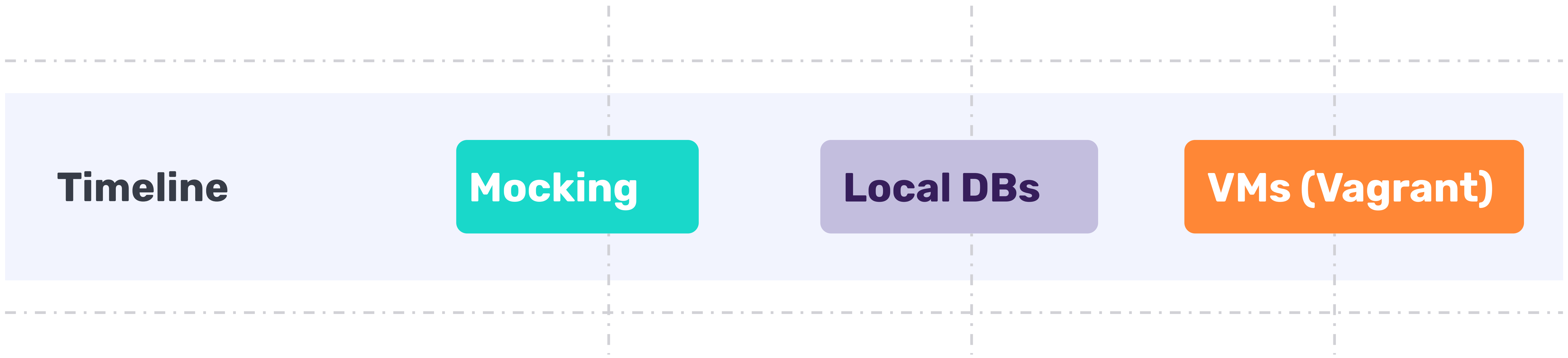


The "**Step Up Rule**": "to test at one layer above what's generally advocated for. Under this model, **unit tests would look more like integration tests** (by treating I/O as a part of the unit under test within *a bounded context*), **integration testing would look more like testing against real production**, and testing in production looks more like, well, **monitoring and exploration**".

Given how broad a spectrum testing is, there's really no One True Way of doing it right. Any approach is going to involve making compromises and tradeoffs.

https://copyconstruct.medium.com/testing-microservices-the-sane-way-9bb31d158c16

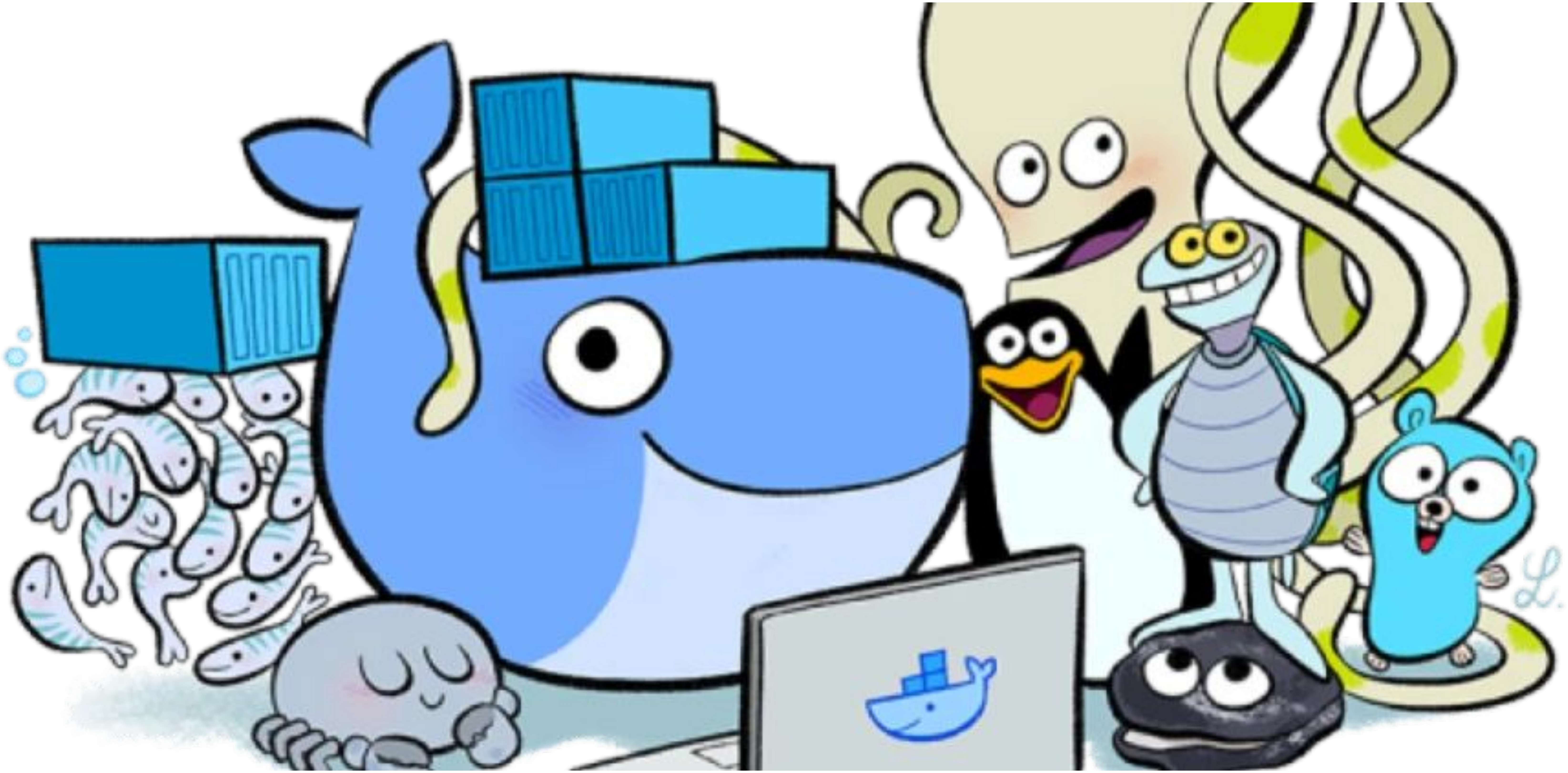
Integration Testing transformation over the years (i)



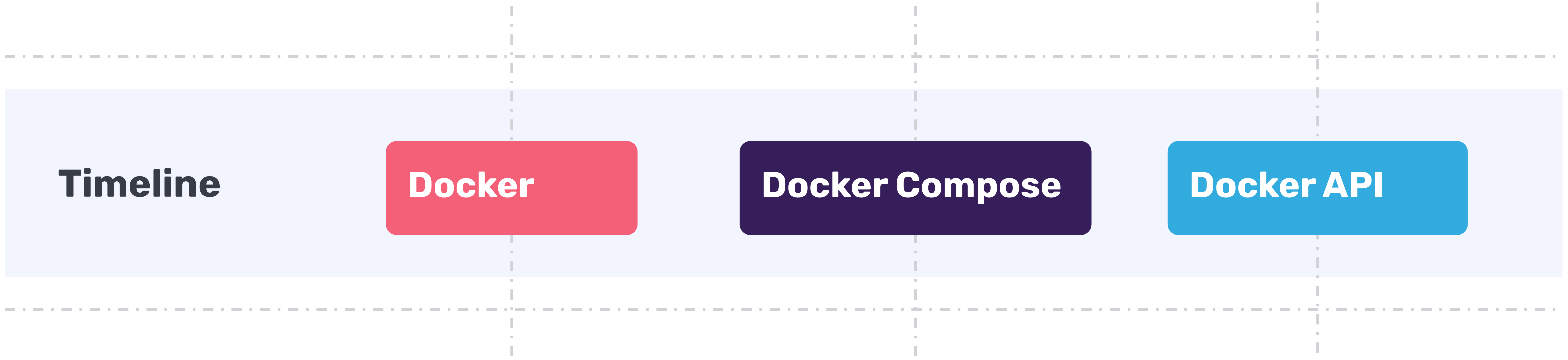
Stupid Example: Local DBs and UTF8, ISO-8859-1

DE LA PEÑA PEÑA → DE LA PENA PENA

Double painful of me!



Integration Testing transformation over the years (ii)



Why the Docker API?

Easy setup of dev environment

Uniform build and test environments

Self-contained and portable environments

No installation and setup of external software… *Well, you need a container runtime* 😁

Integration of the Docker API into your tests for using the same mechanism to setup environments, both local and the CI.

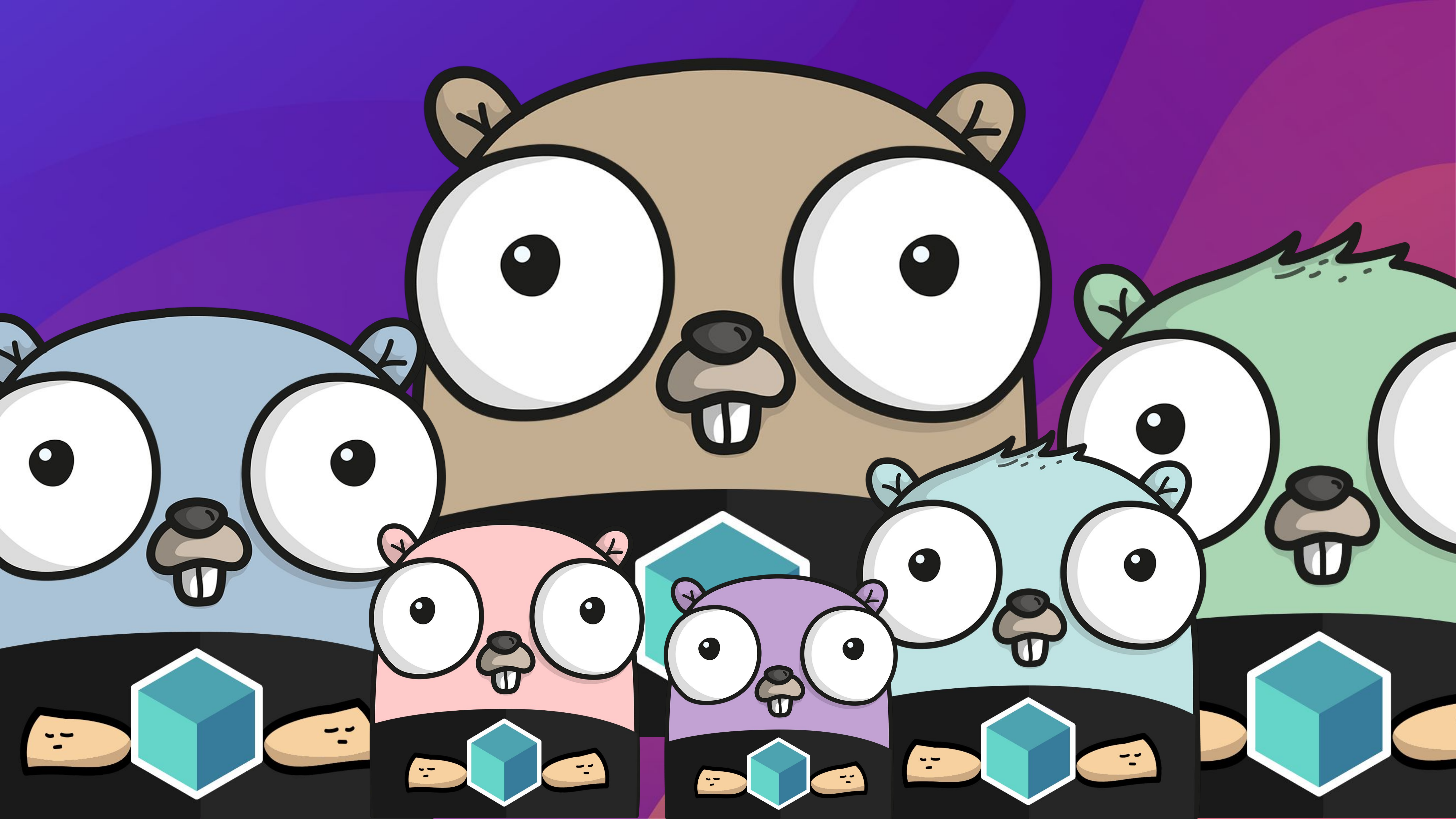
Also tests can define its dependencies in code, next to where they are used.

Testcontainers

Language implementations

- Testcontainers for Java*
- Testcontainers for Go*
- Testcontainers for .NET
- Testcontainers for Node
- Testcontainers for Python
- Testcontainers for Rust
- Testcontainers for Scala
- Testcontainers for Haskell
- Testcontainers for Ruby
- Testcontainers for Clojure
- Testcontainers for Elixir

** Means Docker sponsors the development of those implementations*



Testcontainers for Go

- OSS library - MIT license
- Directly consuming what Docker folks distribute!!
  - No Docker-java, Docker.DotNet, etc
- Run with “go test”
- Simple API to fully customize the Docker container
- 3659 ⭐ (YAVM)

https://github.com/testcontainers/testcontainers-go

Who is using Testcontainers for Go?

*YAVM: Yet Another Vanity Metric

Project	Github Stars (YAVM)	Purpose
tmc/langchaingo	4.7k ⭐	LangChain for Go, the easiest way to write LLM-based programs in Go
apache/beam	7.9k ⭐	Programming model for Batch and Streaming data processing
aquasecurity/trivy	23.7k ⭐	Find vulnerabilities, misconfigurations, secrets, SBOM in containers, Kubernetes, code repositories, clouds and more
confluent/confluent-kafka-go	4.7k ⭐	Confluent's Apache Kafka Golang client
ClickHouse/ClickHouse	37.7k ⭐	a free analytics DBMS for big data
weaviate/weaviate	11.5k ⭐	Open source vector database that stores both objects and vectors
influxdata/influxdb	29k ⭐	Scalable datastore for metrics, events, and real-time analytics
influxdata/telegraf	14.7k ⭐	The plugin-driven server agent for collecting & reporting metrics.
jitsucom/jitsu	4.1k ⭐	An open source high-performance data collection service
kumahq/kuma	3.7k ⭐	🐻 The multi-zone service mesh for containers, Kubernetes and VMs. Built with Envoy. CNCF Sandbox Project.
opentelemetry/opentelemetry-collector-contrib	3.1k ⭐	Contrib repository for the OpenTelemetry Collector
grafana/agent	1.6k ⭐	Grafana Agent is a vendor-neutral, batteries-included telemetry collector with configuration inspired by Terraform.

https://github.com/search?q=%22testcontainers-go+v%22+path%3Ago.mod+NOT+is%3Afork+NOT+org%3Atestcontainers+&type=code

Using Testcontainers for Go

Creating containers

Let’s start a Redis server!

```
redisC, err := testcontainers.GenericContainer(ctx,
testcontainers.GenericContainerRequest{
    ContainerRequest: testcontainers.ContainerRequest{
        Image:           “redis:latest”,
        ExposedPorts: []string{“6379/tcp”},
    },
    Started: true,
})
if err != nil {
    log.Fatal(“container failed to start”)
}
defer func() {
    if err := testcontainers.Terminate(redisC); err != nil {
        log.Fatal(“failed to terminate container”)
    }
}
// test my stuff
```

https://golang.testcontainers.org/quickstart/

Using Testcontainers for Go

Creating networks

Create networks and attach your containers to them.

```
newNetwork, err := network.New(ctx,
    network.WithCheckDuplicate(),
    network.WithAttachable(),
    network.WithLabels(map[string]string{"key": "value"}),
)
if err != nil {
    log.Fatal("failed when creating the network")
}
defer func() {
    if err := newNetwork.Remove(); err != nil {
        log.Fatal("failed to remove network")
    }
}

networkName := newNetwork.Name
// test my stuff
```

https://golang.testcontainers.org/features/creating_networks/

Using Testcontainers for Go

Building from Dockerfiles

Build an image and run a container for it.

```
req := testcontainers.ContainerRequest{
    FromDockerfile:  testcontainers.FromDockerfile{
        Context:     filepath.Join(“path”, “to”, “build”, “context”),
        Dockerfile: “MyDockerfile.dockerfile”,
        BuildArgs:  map[string]*string {
            “FOO”: “BAR”,
        },
        PrintBuildLog: true,
        KeepImage: true,
        BuildOptionsModifier: func(opts *types.ImageBuildOptions) {
            opts.Target = “target2”
        }
    },
    Env: map[string]string {
        “CUSTOM_VAR_1”: “value1”,
    },
},
// create container and test my stuff
```

https://golang.testcontainers.org/features/build_from_dockerfile/

Using Testcontainers for Go

Copying to a container

Sometimes it’s useful to populate the filesystem before the container it’s started: loading a SQL script, copying a config file, etc.

```
req := testcontainers.ContainerRequest{
    Image:   “redis:latest”,
    Files:  []testcontainers.ContainerFile{
        {
            // HostFilePath:       filepath.Join(“path”, “to”, “local”, “file”),
            Reader:               myFileReader,
            ContainerFilePath: “/etc/share/file”, // using Linux paths
            FileMode:             700,
        },
    },
    Env:       map[string]string {
        “CUSTOM_VAR_1”: “value1”,
    },
},
// create container and test my stuff

// or copy a file when the container is already running
redisC.CopyToContainer(ctx, bytes, “/etc/share/file”, 700)
```

https://golang.testcontainers.org/features/files_and_mounts/

Using Testcontainers for Go

Waiting for containers

- For Exec
- For File
- For HostPort
- For HTTP
- For SQL query
- For Log entry
- For Health
- For All (multiple strategies)

E.g. Wait until the Redis log contains certain string.

```
redisC, err := testcontainers.GenericContainer(ctx,
testcontainers.GenericContainerRequest{
    ContainerRequest: testcontainers.ContainerRequest{
        Image:          "redis:latest",
        ExposedPorts: []string{"6379/tcp"},
        WaitingFor:     wait.ForLog("Ready to accept connections"),
    },
    Started: true,
})
if err != nil {
    log.Fatal("container failed to start")
}
defer func() {
    if err := testcontainers.Terminate(redisC); err != nil {
        log.Fatal("failed to terminate container")
    }
}
// test my stuff
```

https://golang.testcontainers.org/features/wait/introduction/

Using Testcontainers for Go

Leverage the container lifecycle

- PreBuilds/PostBuilds
- PreCreates/PostCreates
- PreStarts/PostStarts
- PostReadies (*after wait strategies*)
- PreStops/PostStops
- PreTerminates/PostTerminates

```
req := testcontainers.ContainerRequest{
    Image:          “redis:latest”,
    LifecycleHooks:  []testcontainers.ContainerLifecycleHooks{
        PreCreates: []ContainerRequestHook{
            func(ctx context.Context, req ContainerRequest) error {
                logger.Printf("🐳 Creating container for image %s", req.Image)
                return nil
            },
        },
        PreStarts: []ContainerHook{
            func(ctx context.Context, c Container) error {
                logger.Printf("🐳 Starting container: %s", c.GetContainerID())
                return nil
            },
        },
    },
},
// create container and test my stuff
```

https://golang.testcontainers.org/features/creating_container/#lifecycle-hooks

Using Testcontainers for Go

Garbage collector

Sidecar container that removes:

- Containers
- Images
- Networks
- Volumes

https://github.com/testcontainers/moby-ryuk

```
$> cat ${HOME}/.testcontainers.properties
ryuk.disabled=false
ryuk.container.privileged=true
```

https://golang.testcontainers.org/features/garbage_collector/

TEST DEPENDENCIES AS CODE

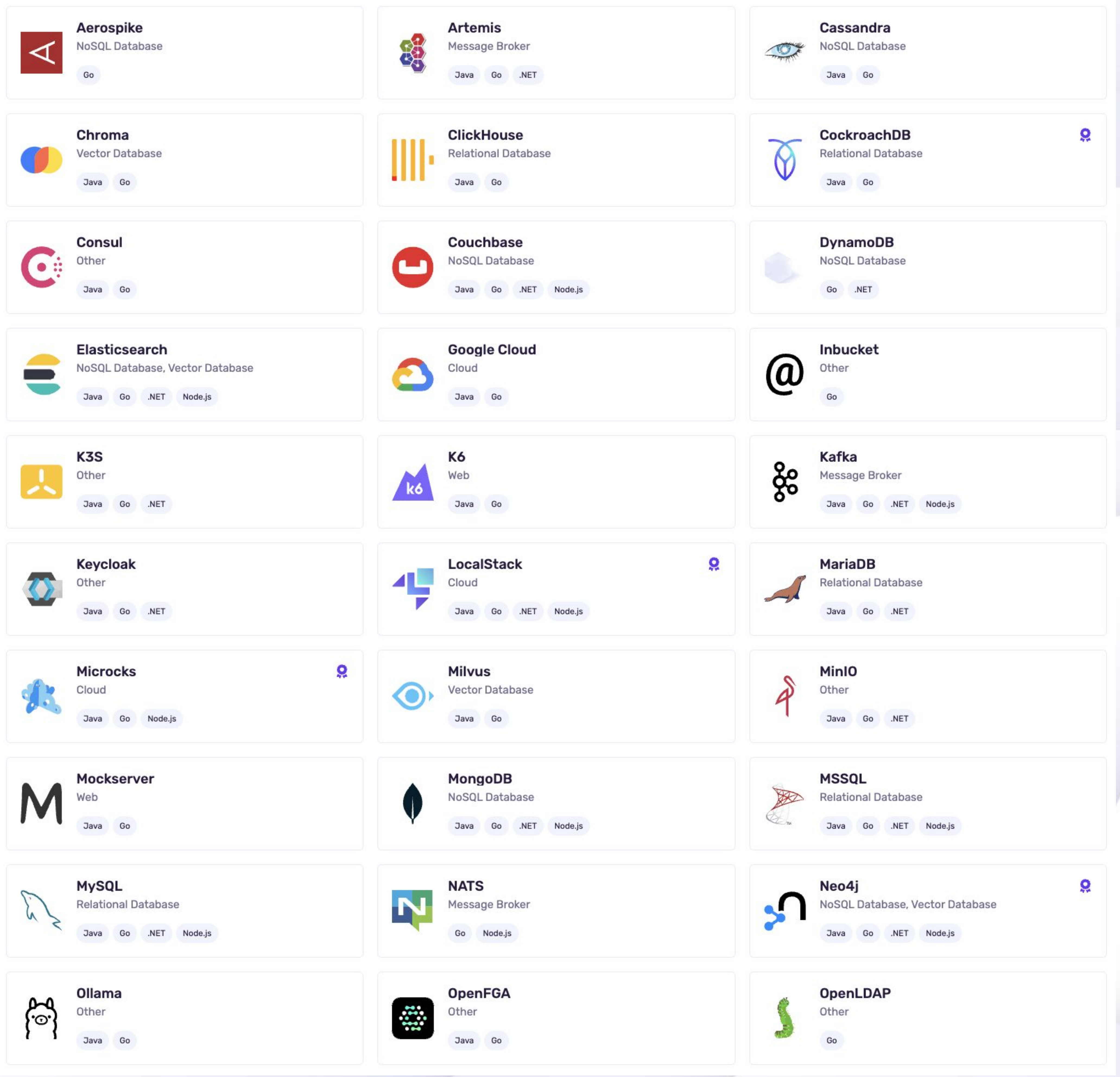
Test against any technology that runs in a Docker container!

Using Testcontainers for Go

Modules!

A wrapper on top of the GenericContainer + some sugar

- Chroma, Milvus, pgVector, Qdrant, Weaviate
- Ollama
- K3s
- GCloud, Localstack, Microcks, Wiremock
- CockroachDB, Couchbase, SurrealDB
- MariaDB, MSSQL, MySQL, Postgres
- ClickHouse, Elasticsearch, MongoDB, Neo4j, OpenSearch, Redis
- Kafka, Pulsar, RabbitMQ, Redpanda
- MinIO, NATS
- Keycloak, OpenFGA, Vault
- …and many more coming soon!



Aerospike NoSQL Database — Go	**Artemis** Message Broker — Java, Go, .NET	**Cassandra** NoSQL Database — Java, Go
Chroma Vector Database — Java, Go	**ClickHouse** Relational Database — Java, Go	**CockroachDB** Relational Database — Java, Go
Consul Other — Java, Go	**Couchbase** NoSQL Database — Java, Go, .NET, Node.js	**DynamoDB** NoSQL Database — Go, .NET
Elasticsearch NoSQL Database, Vector Database — Java, Go, .NET, Node.js	**Google Cloud** Cloud — Java, Go	**Inbucket** Other — Go
K3S Other — Java, Go, .NET	**K6** Web — Java, Go	**Kafka** Message Broker — Java, Go, .NET, Node.js
Keycloak Other — Java, Go, .NET	**LocalStack** Cloud — Java, Go, .NET, Node.js	**MariaDB** Relational Database — Java, Go, .NET
Microcks Cloud — Java, Go, Node.js	**Milvus** Vector Database — Java, Go	**MinIO** Other — Java, Go, .NET
Mockserver Web — Java, Go	**MongoDB** NoSQL Database — Java, Go, .NET, Node.js	**MSSQL** Relational Database — Java, Go, .NET, Node.js
MySQL Relational Database — Java, Go, .NET, Node.js	**NATS** Message Broker — Go, Node.js	**Neo4j** NoSQL Database, Vector Database — Java, Go, .NET, Node.js
Ollama Other — Java, Go	**OpenFGA** Other — Java, Go	**OpenLDAP** Other — Go

https://testcontainers.com/modules/?language=go
https://golang.testcontainers.org/modules/

Using modules

Ex #1 Postgres

```
Run(
    ctx, “image”, opts
…Customizers,
)
```

```
package main_test
import (
	…
	“github.com/testcontainers/testcontainers-go”
	“github.com/testcontainers/testcontainers-go/modules/postgres”
)
func TestPostgres(t *testing.T) {
	ctx := context.Background()
	container, err := postgres.Run(ctx,
		“postgres:14”,
		postgres.WithDatabase(“my-database”),
		postgres.WithUsername(“gopher”),
		postgres.WithPassword(“p4ssw0rd!”),
		testcontainers.WithWaitStrategy(wait.ForLog(“database system is ready to accept
	connections”).WithOccurrence(2),
	)
	if err != nil {
		t.Fatal(err)
	}
}
```

https://golang.testcontainers.org/modules/postgres/

Using modules

Ex #2.1 Ollama

Run(
    ctx, “image”, opts
…Customizers,
)

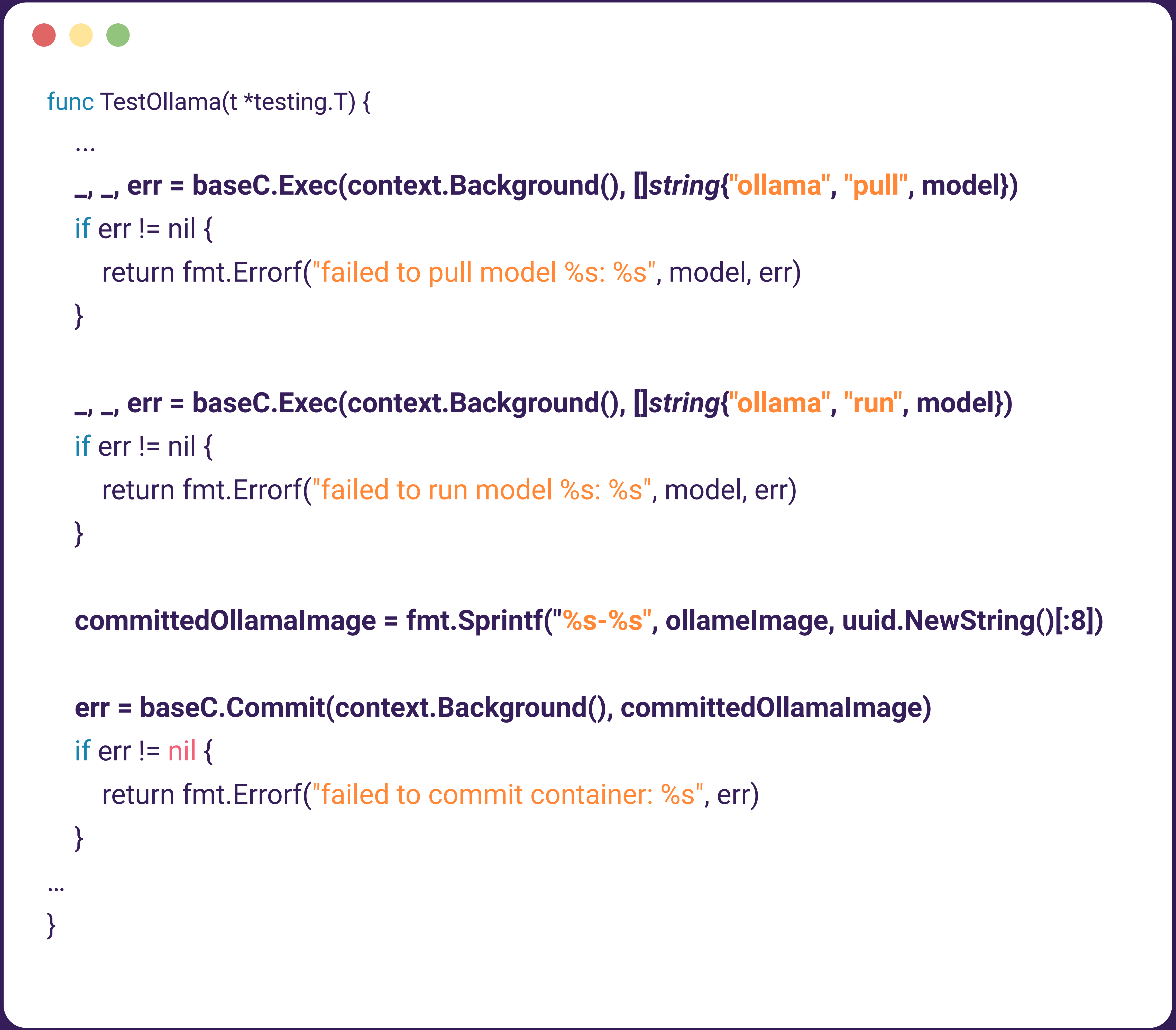
```go
package main_test
import (
        …
        "github.com/testcontainers/testcontainers-go"
        tcollama "github.com/testcontainers/testcontainers-go/modules/ollama"
        "github.com/tmc/langchaingo/llms"
        "github.com/tmc/langchaingo/llms/ollama"
)
const (
  model       string = "llama2"
  ollamaImage string = "ollama/ollama:0.1.25"
)
var committedOllamaImage string
func TestOllama(t *testing.T) {
   baseC, err := tcollama.Run(context.Background(), ollamaImage)
   if err != nil {return err}
   defer func() {
      err := baseC.Terminate(context.Background())
      if err != nil {
         // Do not fail the test if we cannot terminate the container,
         // as the committed image will be used in the following subtests.
         // Ryuk will take care of cleaning up the container eventually.
         fmt.Printf("failed to terminate container: %s", err)
      }
   }()
…
}
```

https://golang.testcontainers.org/modules/ollama/

Using modules

Ex #2.2 Ollama

Pull & Run model
Commit(ctx, targetImage)



```go
func TestOllama(t *testing.T) {
   ...
   _, _, err = baseC.Exec(context.Background(), []string{"ollama", "pull", model})
   if err != nil {
      return fmt.Errorf("failed to pull model %s: %s", model, err)
   }

   _, _, err = baseC.Exec(context.Background(), []string{"ollama", "run", model})
   if err != nil {
      return fmt.Errorf("failed to run model %s: %s", model, err)
   }

   committedOllamaImage = fmt.Sprintf("%s-%s", ollameImage, uuid.NewString()[:8])

   err = baseC.Commit(context.Background(), committedOllamaImage)
   if err != nil {
      return fmt.Errorf("failed to commit container: %s", err)
   }
...
}
```

https://golang.testcontainers.org/modules/ollama/

Using modules

Ex #2.3 Ollama

Leveraging tmc/langchaingo to interact with the Ollama model.

```go
func TestOllama(t *testing.T) {
   ...
   c, err := tcollama.Run(context.Background(), committedOllamaImage)
   if err != nil {return err}

   url, err := c.ConnectionString(context.Background())
   if err != nil {
      tt.Fatalf("failed to get connection string: %s", err)
   }

   llm, err := ollama.New(ollama.WithModel(model),
ollama.WithServerURL(url))
   if err != nil {
      tt.Fatalf("failed to create LLM: %s", err)
   }
   ...
}
```

https://golang.testcontainers.org/modules/ollama/

Using modules

Ex #2.4 Ollama

Leveraging tmc/langchaingo to interact with the Ollama model.

```go
func TestOllama(t *testing.T) {
    ...
    completion, err := llm.Call(
        context.Background(),
        “How can Testcontainers help with testing?”,
        llms.WithSeed(42),
        llms.WithTemperature(0.0),
    )
    if err != nil {tt.Error(err)}
    lwCompletion := strings.ToLower(completion)
    expectedWords := []string{"easy", "isolation", "consistency"},
    for _, word := range expectedWords {
        // compare in lowercase to avoid case sensitivity
        if !strings.Contains(lwCompletion, strings.ToLower(word)) {
            tt.Errorf("expected completion to contain '%s'", word)
        }
    }
}
```

https://golang.testcontainers.org/modules/ollama/

Using modules

Ex #3 Localstack

```
Run(
    ctx, “image”, opts
…Customizers,
)
```

```
package main_test
import (
        …
        “github.com/testcontainers/testcontainers-go”
        “github.com/testcontainers/testcontainers-go/modules/localstack”
)
func TestLocalstack(t *testing.T) {
        ctx := context.Background()
        container, err := localstack.Run(ctx,
                “localstack:2.2.0”,
                testcontainers.WithEnv(map[string]string{
                    “SERVICES”: “lambda”,
                }),
        )
        if err != nil {
                t.Fatal(err)
        }
}
```

https://pkg.go.dev/github.com/testcontainers/testcontainers-go/modules/localstack#example-RunContainer-UsingLambdas

Using modules

Ex #4 k3s + k6

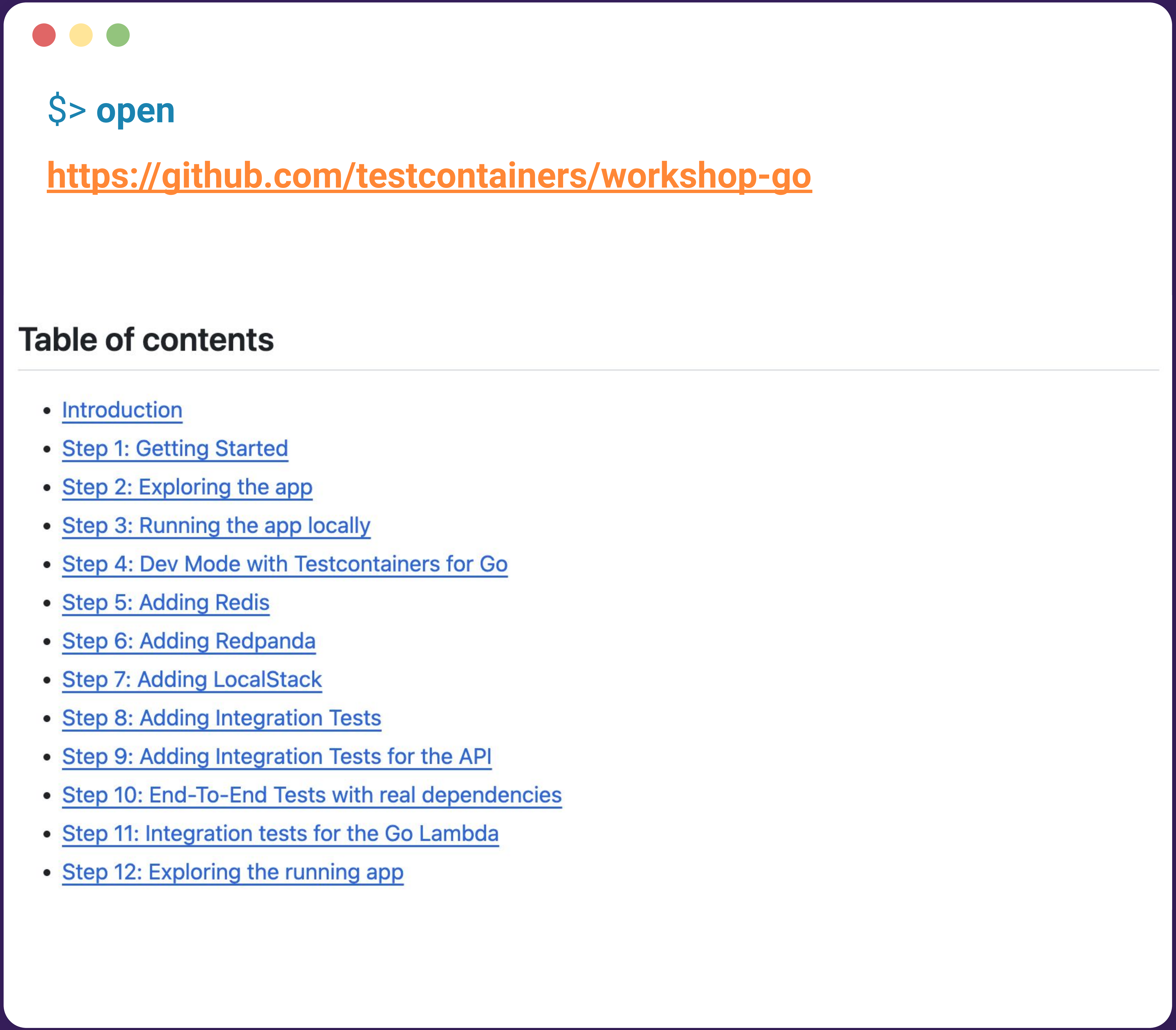
```
Run(
    ctx, "image", opts
...Customizers,
)
```

```
$> open
https://github.com/grafana/k6-testcontainers-demo/blob/main/
demo_test.go
```

Using Testcontainers for Go

Hands on!!

In the case you want to learn more about Testcontainers for Go at your own pace.



```
$> open
```

https://github.com/testcontainers/workshop-go

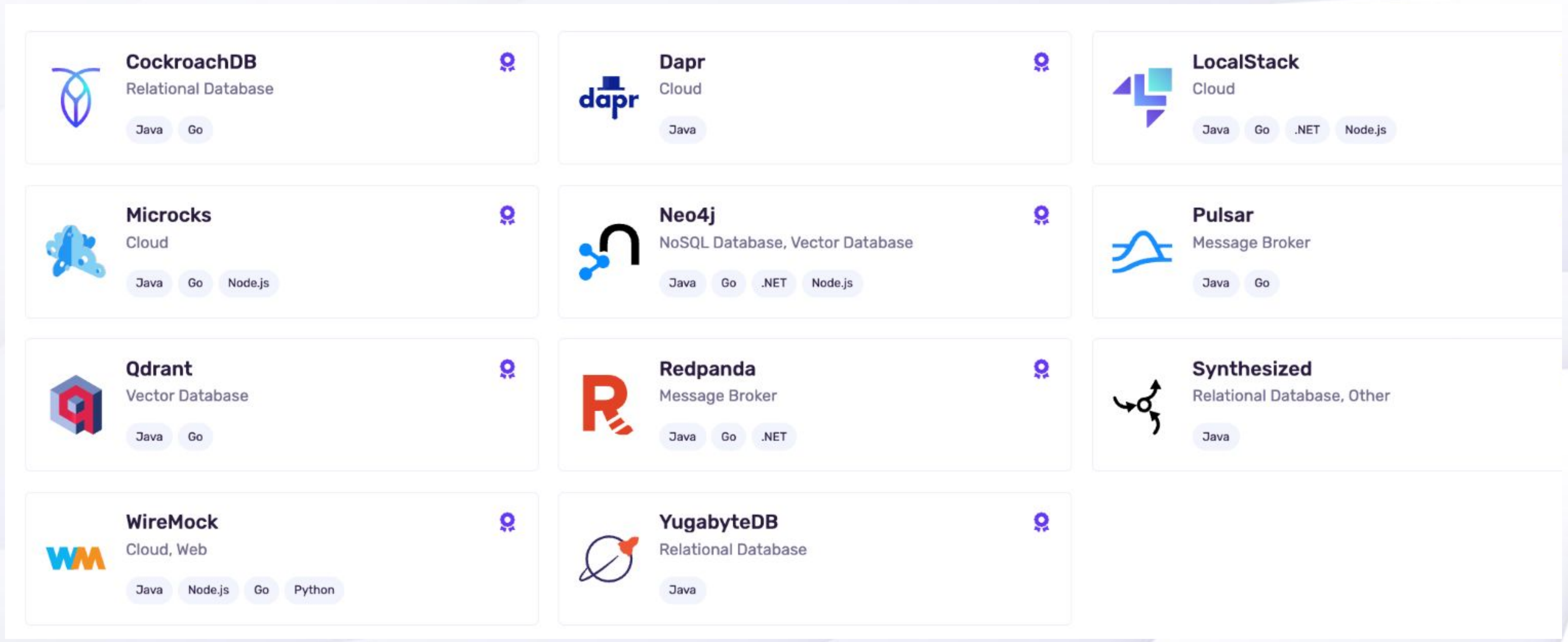
Table of contents

Using Testcontainers for Go

Official Modules

Major vendors are backing the development of the modules for all the languages.

Testcontainers is seen as a super valuable tool for CI.



CockroachDB
Relational Database
Java Go

Dapr
Cloud
Java

LocalStack
Cloud
Java Go .NET Node.js

Microcks
Cloud
Java Go Node.js

Neo4j
NoSQL Database, Vector Database
Java Go .NET Node.js

Pulsar
Message Broker
Java Go

Qdrant
Vector Database
Java Go

Redpanda
Message Broker
Java Go .NET

Synthesized
Relational Database, Other
Java

WireMock
Cloud, Web
Java Node.js Go Python

YugabyteDB
Relational Database
Java

https://thenewstack.io/how-testcontainers-is-demonstrating-value-as-a-key-ci-tool/

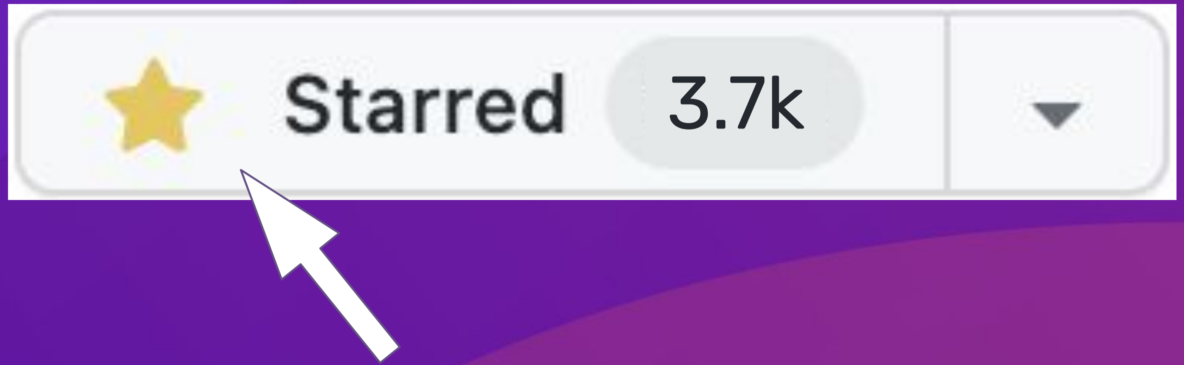
Self-paced resources

https://golang.testcontainers.org

https://testcontainers.com/guides/getting-started-with-testcontainers-for-go/

https://github.com/testcontainers/workshop-go

You liked it??



Thanks!

https://slack.testcontainers.com

@mdelapenya everywhere