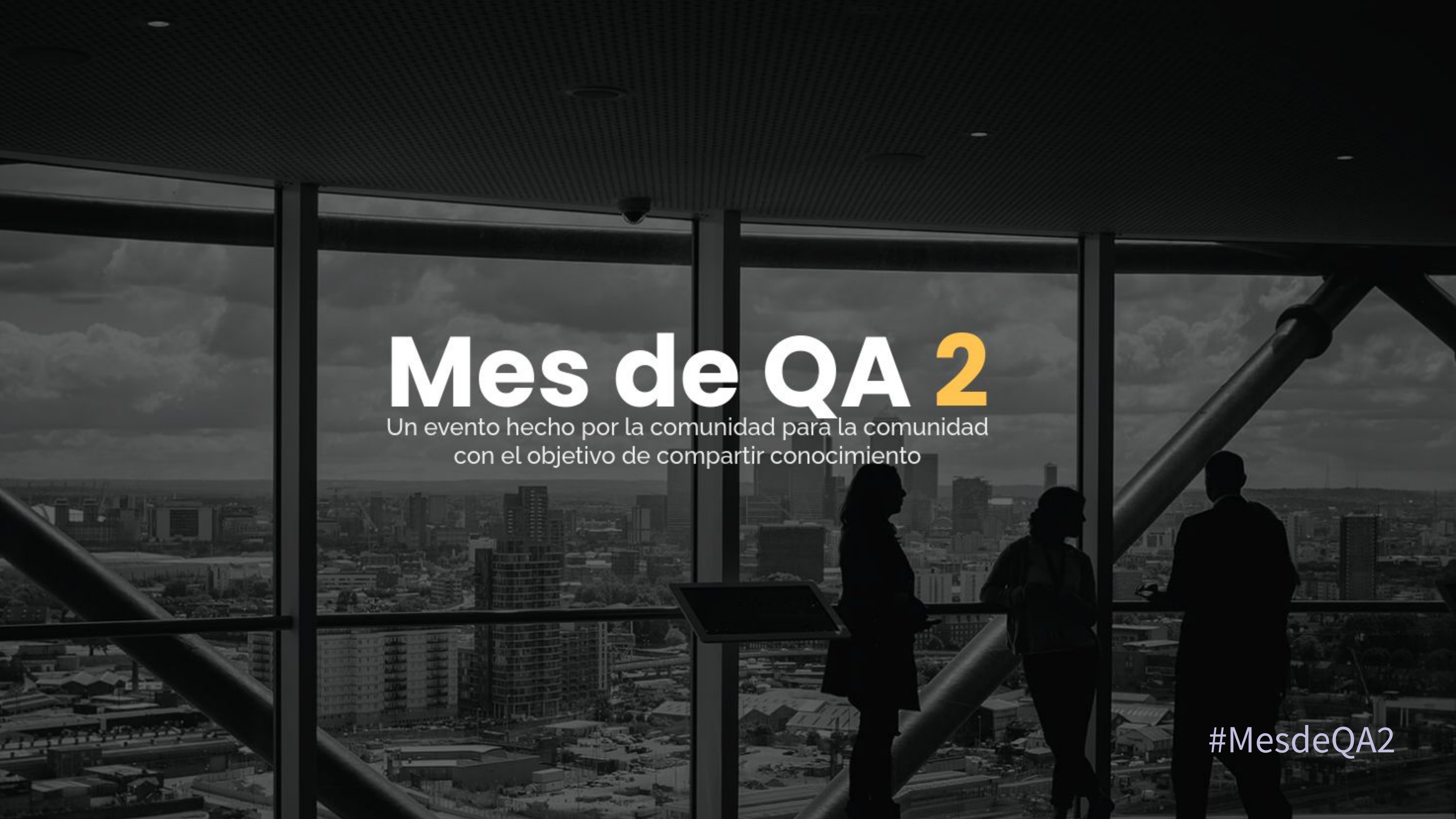

Mes de QA 2
Un evento hecho por la comunidad para la comunidad
con el objetivo de compartir conocimiento
#MesdeQA2

#MesdeQA2

Mes de QA 2

Integration Testing:

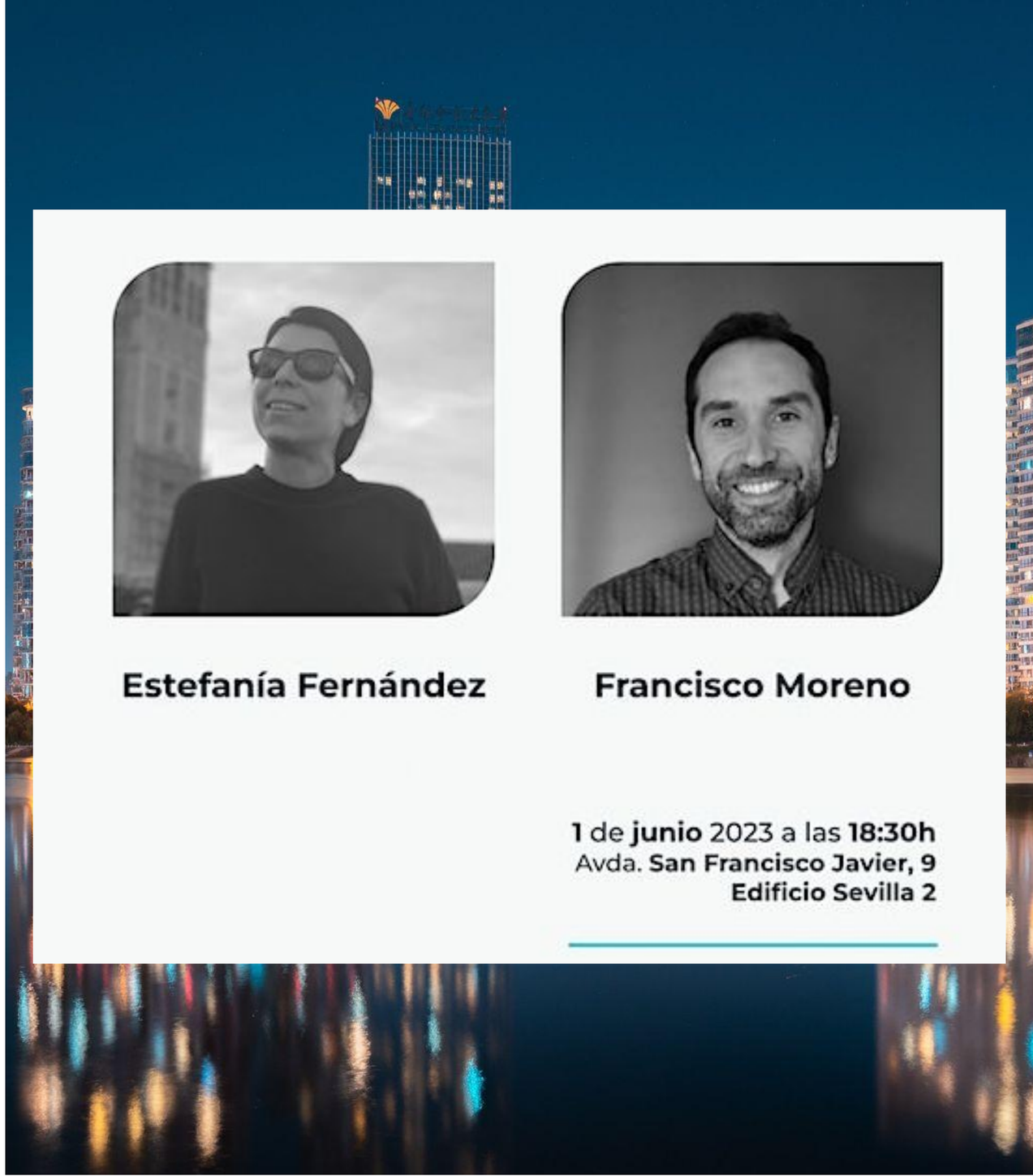
The way to Go

30 Mayo | 19:00 | Liferay Madrid

Manuel de la Peña
Ingeniero de software en AtomicJar

Próximo meetup 1 de Junio

Mes de QA



Integration Testing

The Way to Go

@mdelapenya everywhere

Manuel de la Peña

Software Engineer - OSS

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

ATOMICJAR

Who ~~loves~~ writes tests?

Why do we test?

Reasons why we test

Fast feedback

Way to get experience 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 we set up test environments

Declare test environments as part of test code

Using Testcontainers for Go for integration testing

Make it easier to maintain integration tests

ITs not hard anymore: not expensive to run/write

ATOMICJAR

Testing Dorito

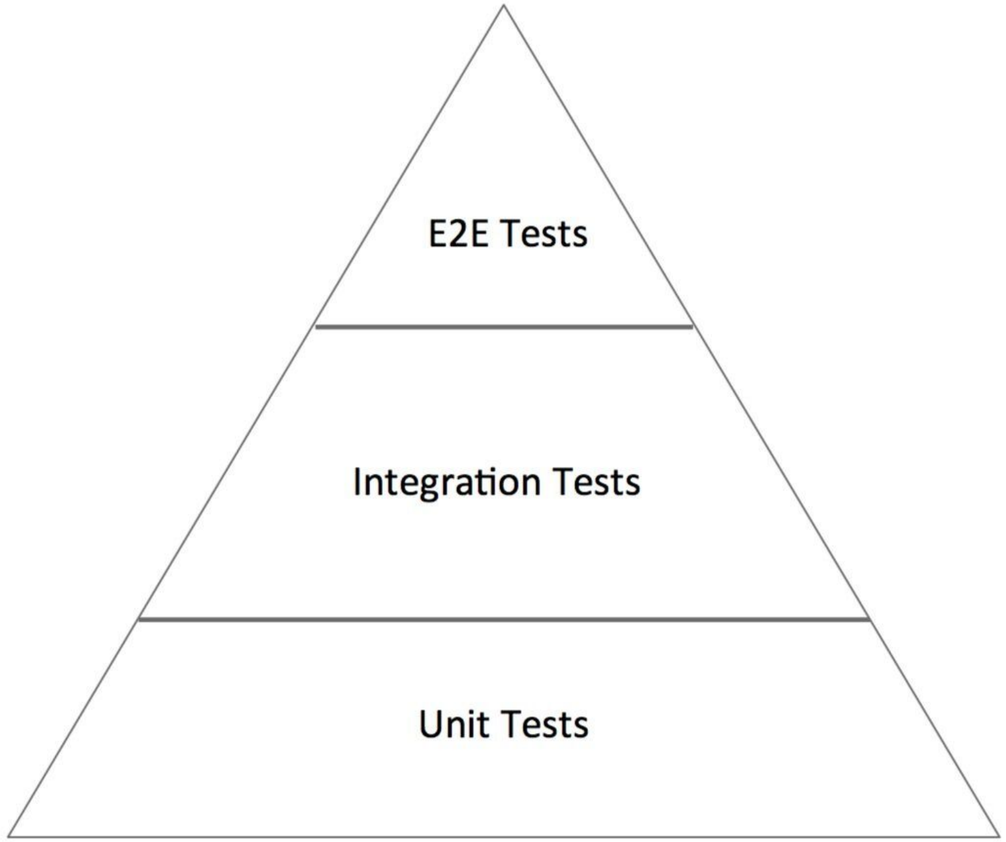
Tests I Plan to Write

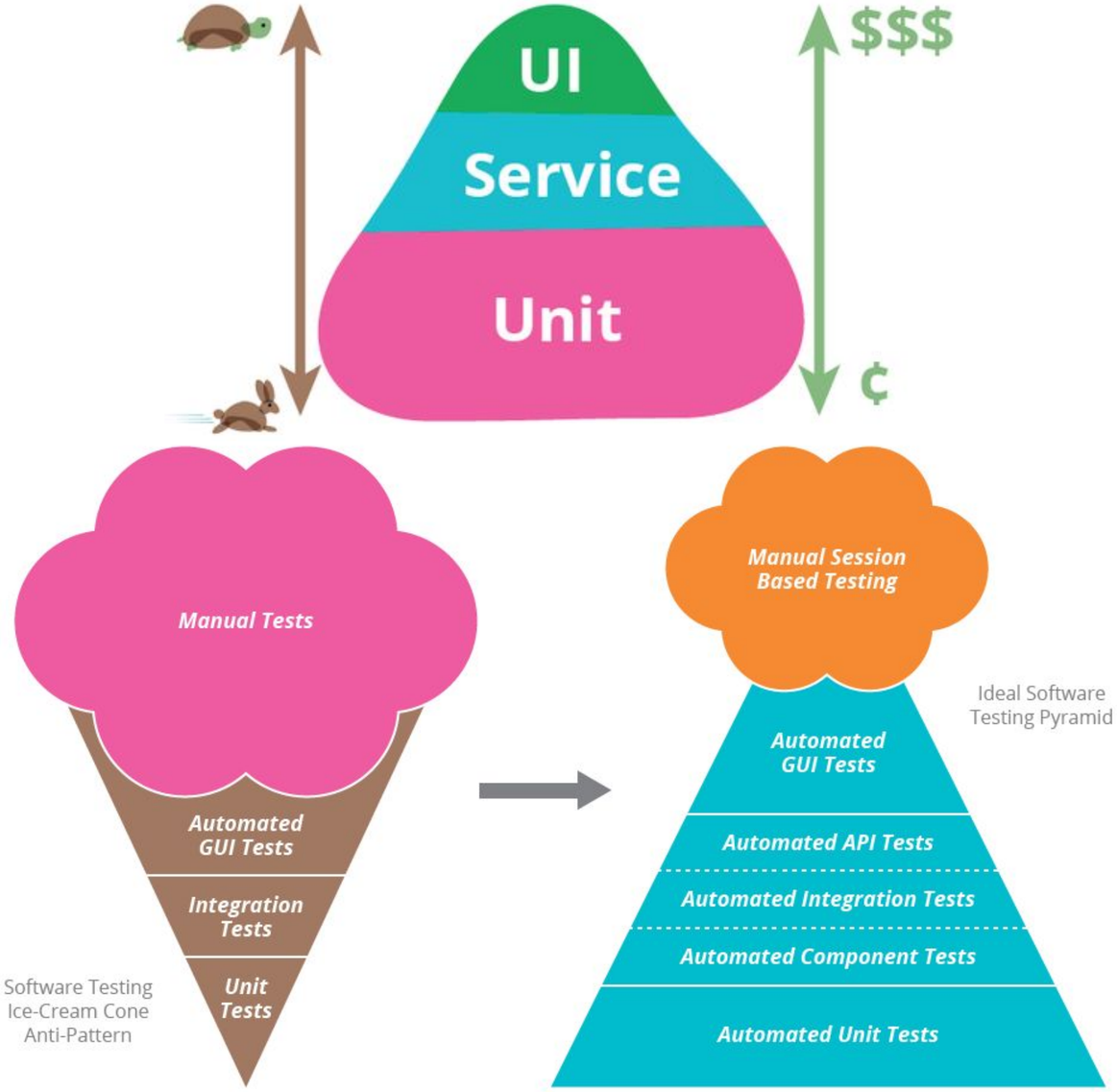
Tests I Start Writing

Tests I delete Because I decide they are stupid and take more time then they are worth

Tests

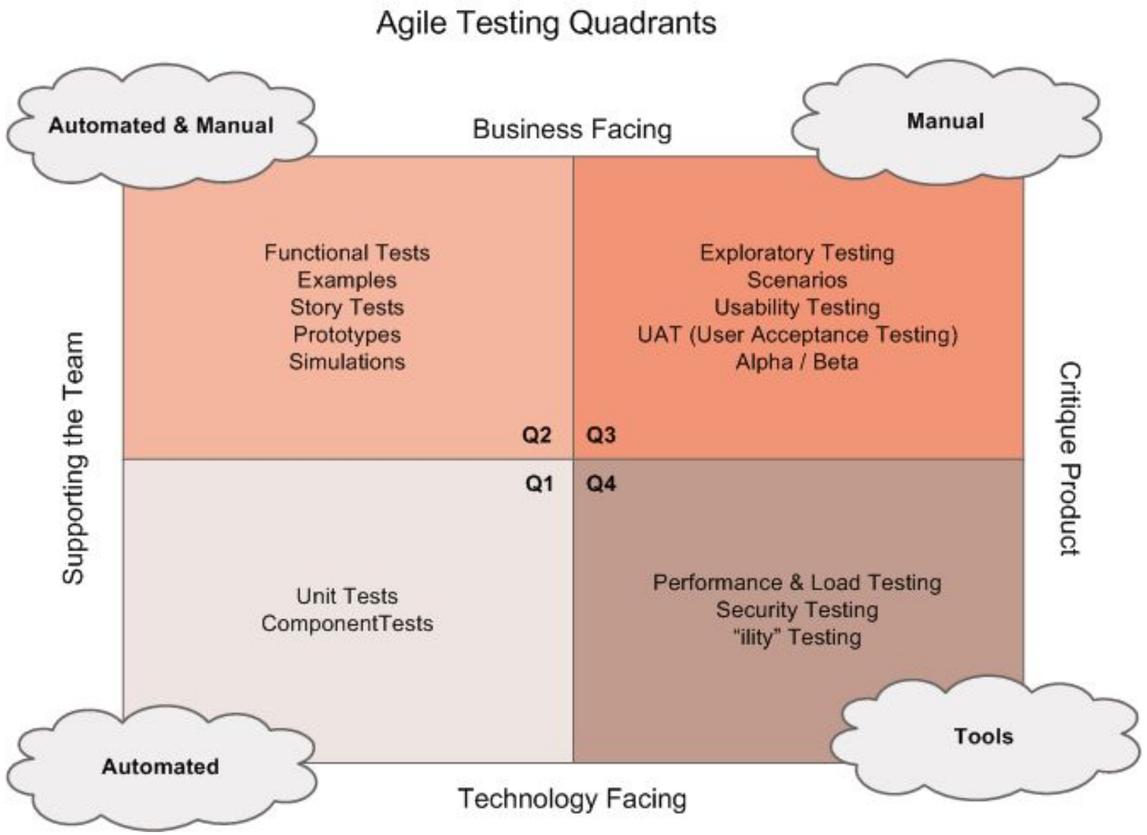
Testing pyramid (Mike Cohn - 2003)





ATOMICJAR

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".

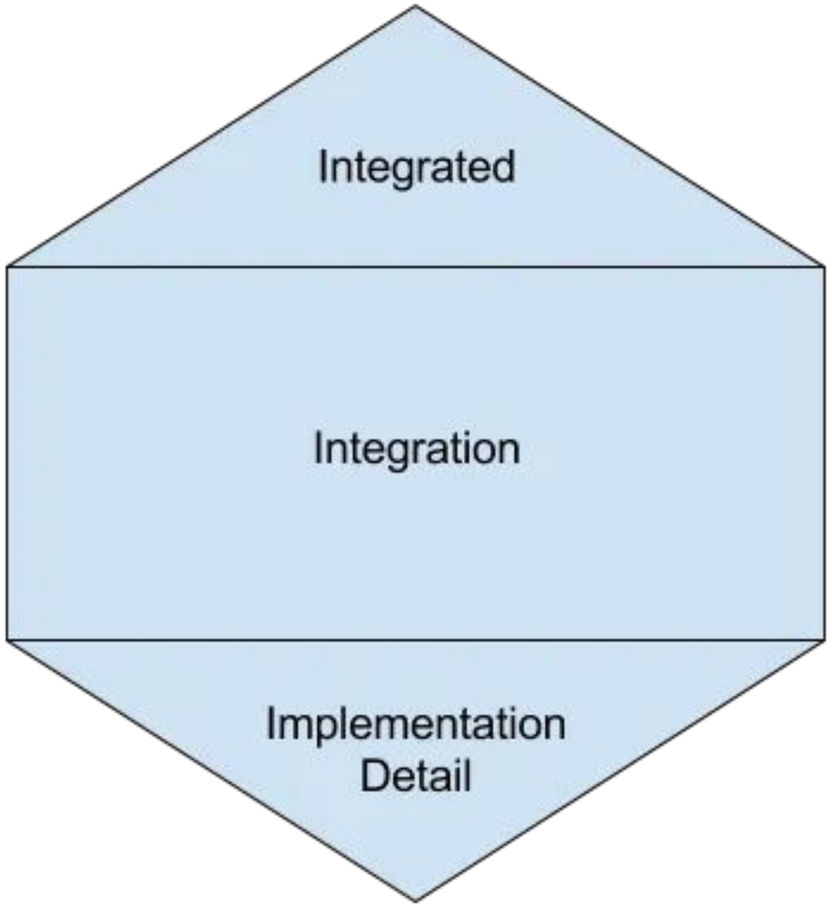
"Programmer tests are normally part of the automated process that runs with every code check-in, giving the team **instant, continual feedback** about their internal quality".

"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)

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.

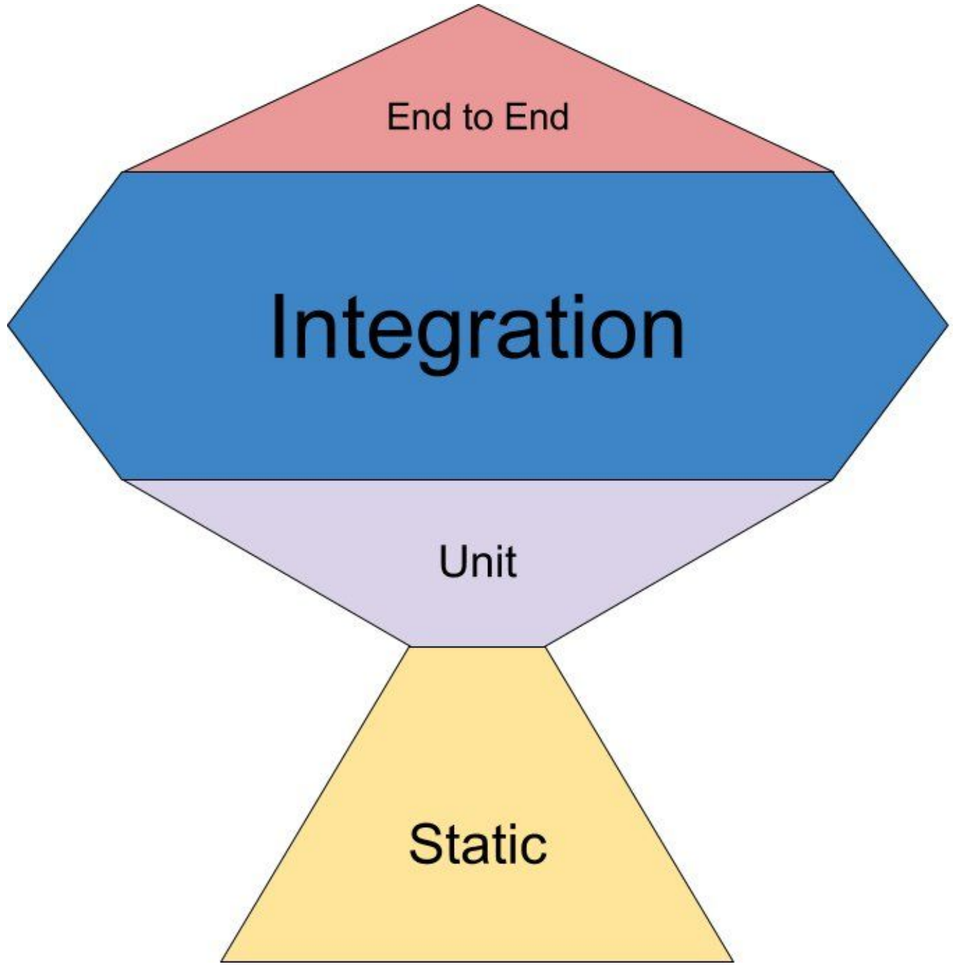
- Spin up services in a local testing environment
- Test against services in a shared environment

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)

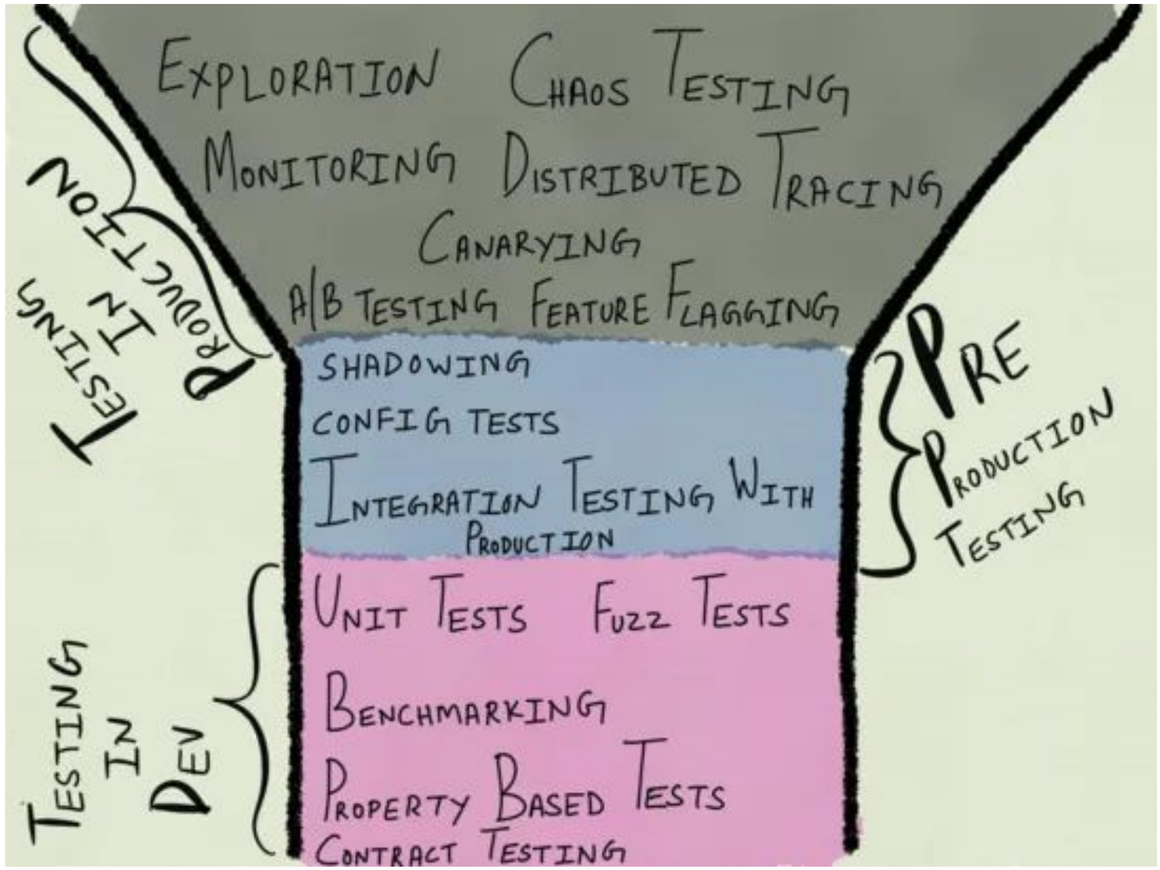


"The line between integration and unit is a little bit fuzzy. (...) the biggest thing you can do to write more integration tests is **to stop mocking so much stuff**. When you mock something *you're removing all confidence in the integration between what you're testing and what's being mocked*".

"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 (by 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.

ATOMICJAR

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

A possible definition??

Interact with external system/dependencies

External processes or programs

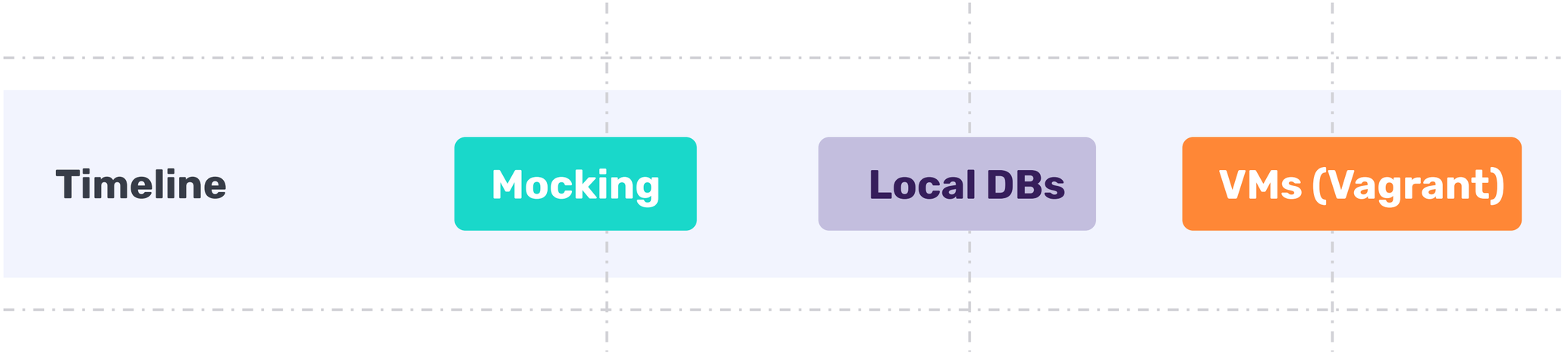
Outside the boundaries of my App

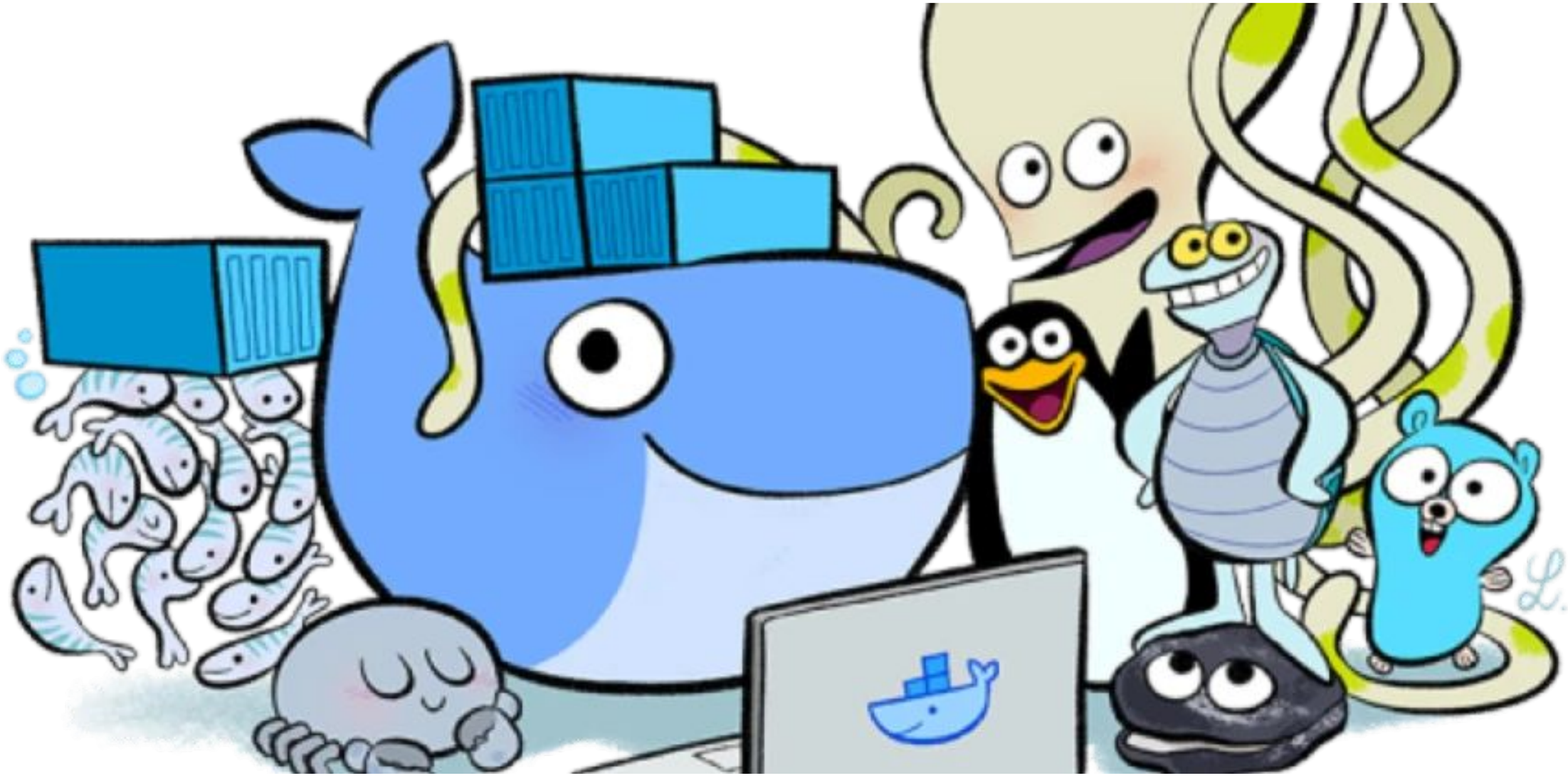
Integration Tests

E.g.: interactions with the network, the filesystem, databases, queues.

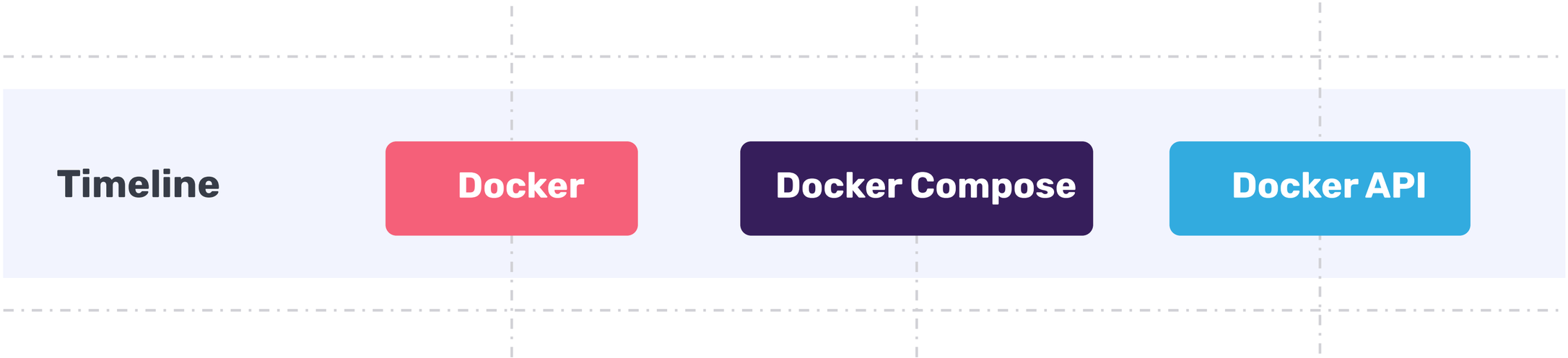
Each **mental model** would find their own definition.

Integration Testing transformation over the years (i)





Integration Testing transformation over the years (ii)



Why 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 Docker 😁

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

ATOMICJAR

Testcontainers

How we define Testcontainers

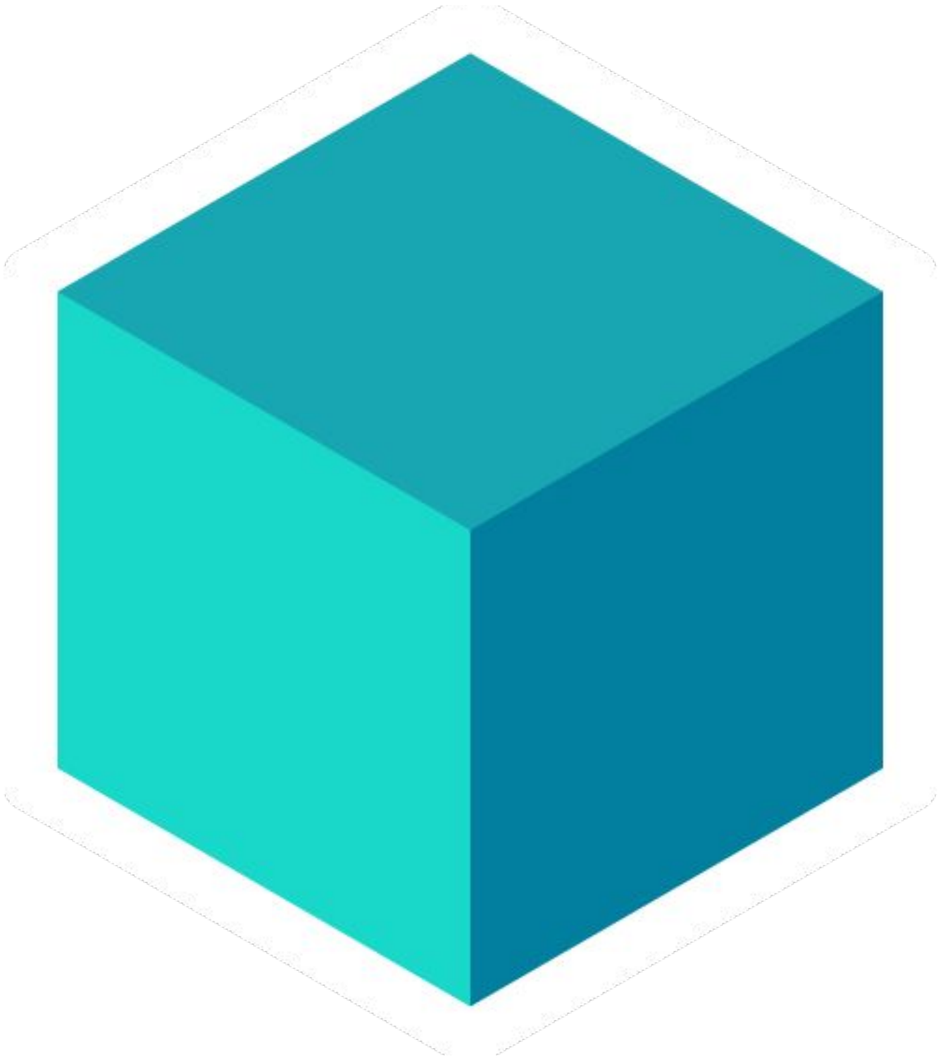
“Testcontainers allow developers to test and develop their code against the real dependencies they will use when the app goes live for use.”

Eli Aleyner
AtomicJar Co-Founder

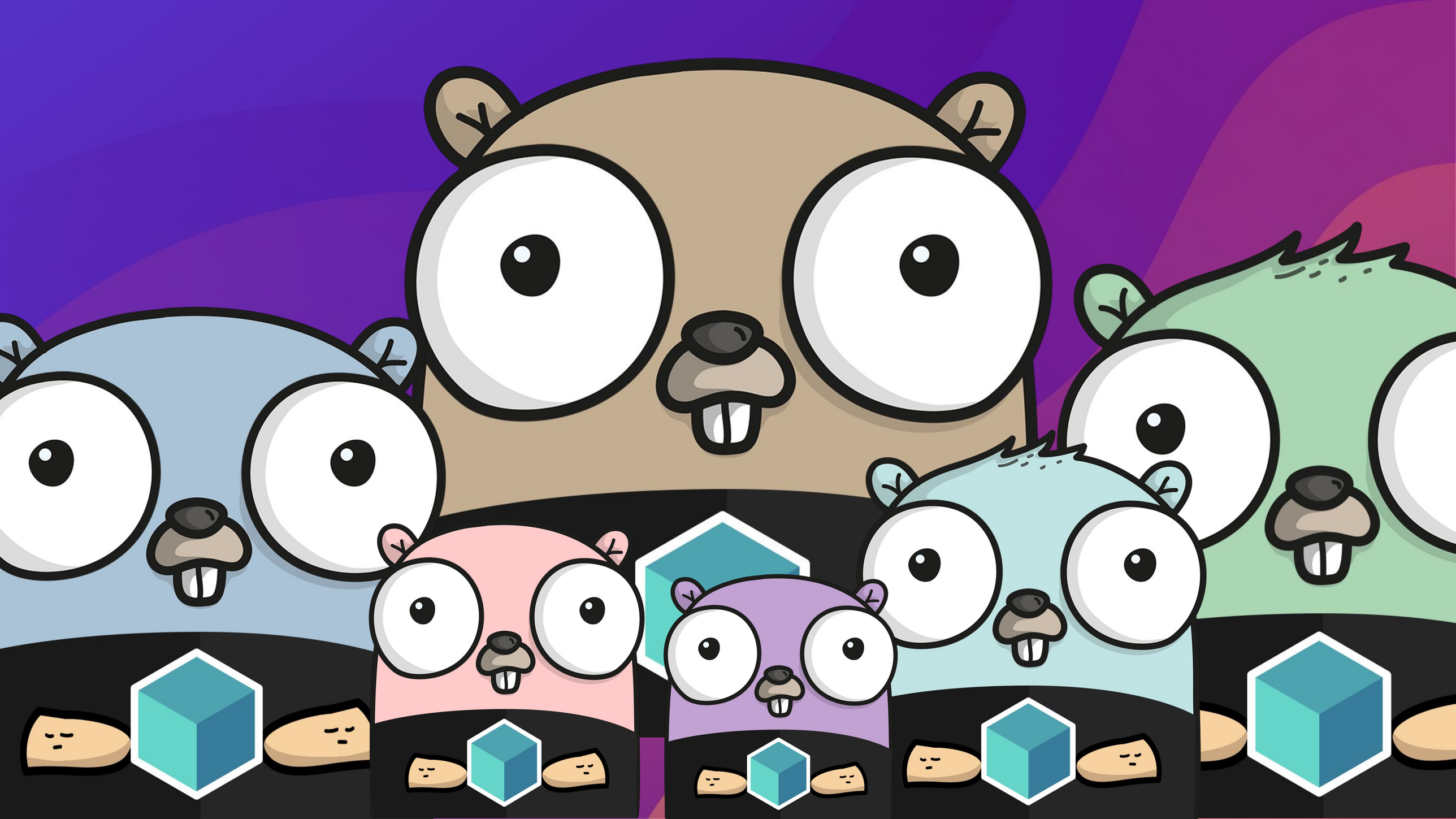
Language implementations

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

** Means AtomicJar, Inc sponsors the development of those implementations*



ATOMICJAR



Testcontainer 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

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

ATOMICJAR

Who is using Testcontainers for Go?

Project	Github Stars (YAVM)	Purpose
apache/beam	6.8k ⭐	Programming model for Batch and Streaming data processing
aquasecurity/trivy	17k ⭐	Find vulnerabilities, misconfigurations, secrets, SBOM in containers, Kubernetes, code repositories, clouds and more
confluent/confluent-kafka-go	3.9k ⭐	Confluent's Apache Kafka Golang client
ClickHouse/ClickHouse	28.2k ⭐	a free analytics DBMS for big data
elastic/apm-server	1.1k ⭐	Application Performance Metrics server for the Elastic Stack
influxdata/influxdb	25.3k ⭐	Scalable datastore for metrics, events, and real-time analytics
influxdata/telegraf	12.8k ⭐	The plugin-driven server agent for collecting & reporting metrics.
jitsucom/jitsu	3.3k ⭐	An open source high-performance data collection service
kumahq/kuma	3.1k ⭐	🐻 The multi-zone service mesh for containers, Kubernetes and VMs. Built with Envoy. CNCF Sandbox Project.
opentelemetry/opentelemetry-collector-contrib	1.7k ⭐	Contrib repository for the OpenTelemetry Collector

*YAVM: Yet Another Vanity Metric

https://github.com/search?q=%22testcontainers-go+v%22+path%3Ago.mod+NOT+is%3Afork&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 := redisC.Terminate(); err != nil {
        log.Fatal("Failed to terminate container")
    }
}
// test my stuff
```

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

Using Testcontainers for Go

Waiting for containers

Wait until the Redis log contains certain string.

There are many wait strategies

- For Exec
- For HostPort
- For HTTP
- For SQL query
- For Log entry
- For Health
- For multiple strategies

```
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 := redisC.Terminate(); err != nil {
        log.Fatal("Failed to terminate container")
    }
}
// test my stuff
```

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

Using Testcontainers for Go

Creating networks

Create networks and attach your containers to them.

```
newNetwork, err := testcontainers.GenericNetwork(ctx,
testcontainers.GenericNetworkRequest{
    NetworkRequest: testcontainers.NetworkRequest{
        Name:           "new-network",
        CheckDuplicate: true,
    },
})
if err != nil {
    log.Fatal("Failed when creating the network")
}
defer func() {
    if err := newNetwork.Remove(); err != nil {
        log.Fatal("Failed to remove network")
    }
}
// test my stuff
```

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

Using Testcontainers for Go

Building from Dockerfiles

Build and 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,
    },
    ExposedPorts: []string{"6379/tcp"},
    Env: map[string]string {
        "CUSTOM_VAR_1": "value1",
        "CUSTOM_VAR_2": "value2",
    },
},
// create container and test my stuff
```

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

Using Testcontainers for Go

Copying files or directories to a container

Sometimes it’s useful to populate the filesystem before the container it’s started.

```
req := testcontainers.ContainerRequest{
    Files:  []testcontainers.ContainerFile{
        HostFilePath:      filepath.Join(“path”, “to”, “local”, “file”),
        ContainerFilePath: “/etc/share/file”, // using Linux paths
        FileMode:          700,
    },
    Image:        “redis:latest”,
    ExposedPorts: []string{“6379/tcp”},
    Env:          map[string]string {
        “CUSTOM_VAR_1”: “value1”,
        “CUSTOM_VAR_2”: “value2”,
    },
},
// create container and test my stuff
// or copy a file when the container is already running
redisC.CopyFileToContainer(ctx, filepath.Join(“path”, “to”, “local”, “file”),
“/etc/share/file”, 700)
```

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

Using Testcontainers for Go

Leverage the container lifecycle

- PreCreate/PostCreate
- PreStart/PostStart
- PreStop/PostStop
- PreTerminate/PostTerminate

```
req := testcontainers.ContainerRequest{
    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
            },
        },
    },
    Image:              "redis:latest",
},
// 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

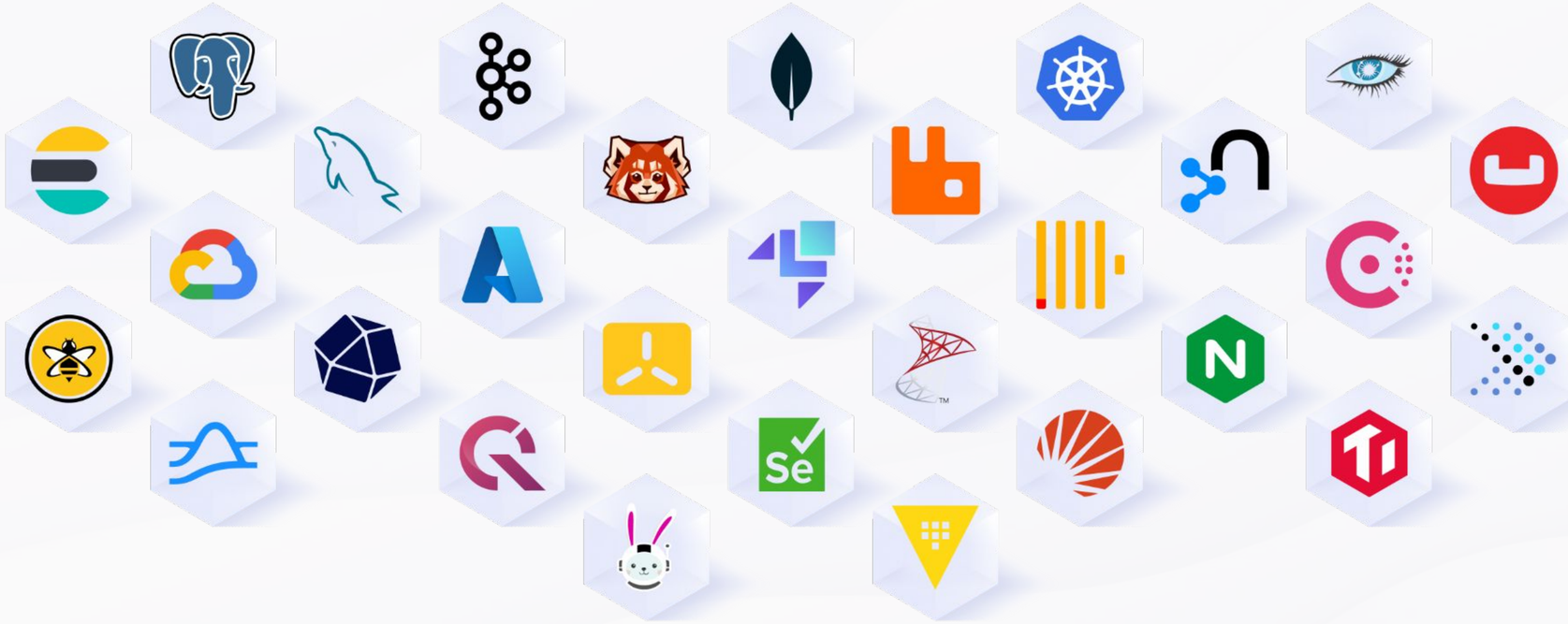
https://github.com/testcontainers/moby-ryuk

```
$> 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!

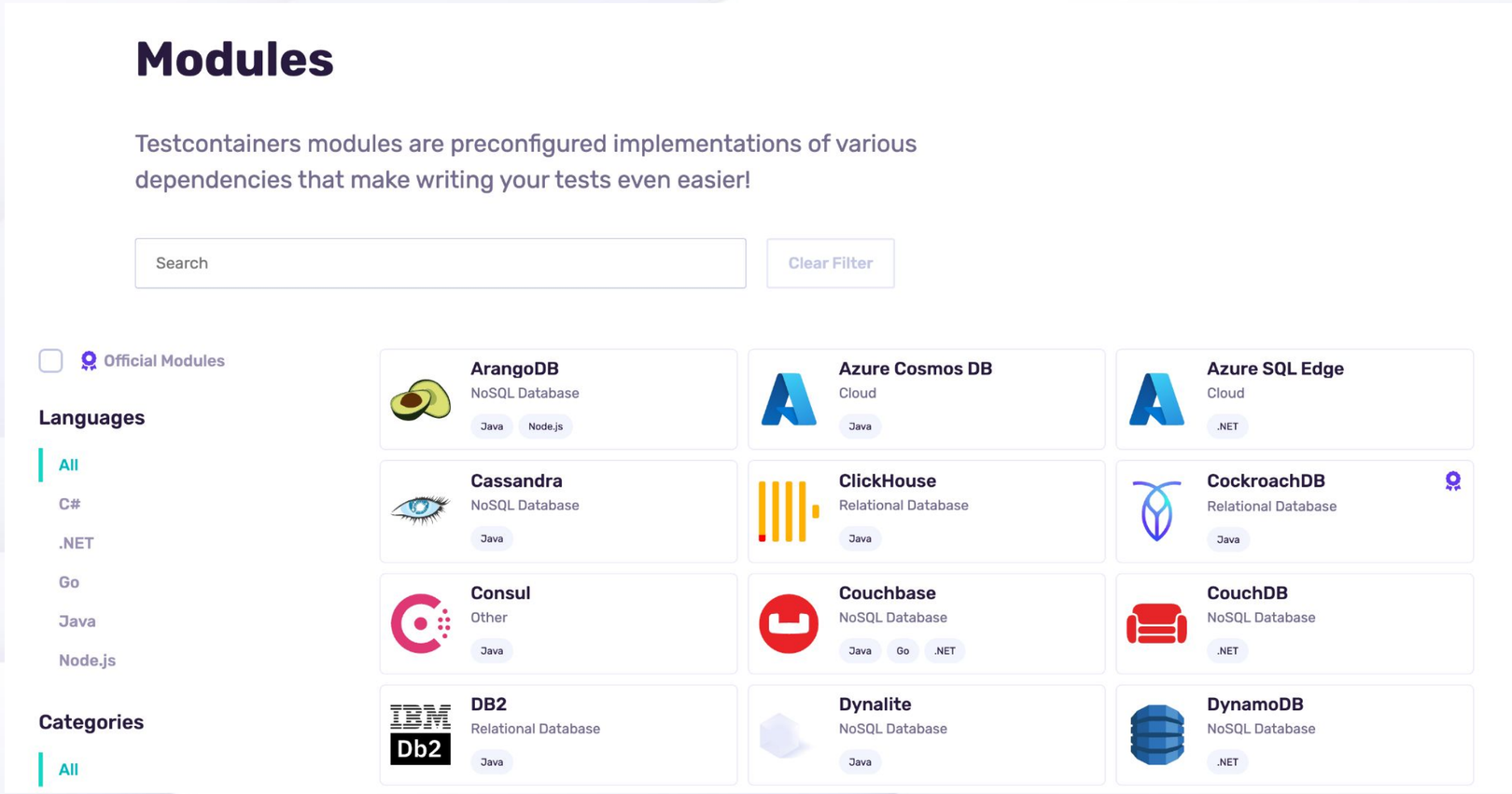


Using Testcontainers for Go

Modules!

A wrapper on top of the GenericContainer with some sugar and custom capabilities.

- Apache Pulsar
- Couchbase
- Localstack
- MySQL
- Postgres
- Neo4j
- Redis
- Vault
- …and many more coming soon!



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

Using Go modules

Ex #1 Postgres

```
RunContainer(
    ctx, opts ...Customizers,
)
```

```go
package main_test
import (
    ...
    "github.com/testcontainers/testcontainers-go/modules/postgres"
)
func TestPostgres(t *testing.T) {
    ctx := context.Background()
    container, err := postgres.RunContainer(ctx,
        testcontainers.WithImage("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 Go modules

Ex #2 Neo4j

```
RunContainer(
    ctx, opts ...Customizers,
)
```

```
package main_test
import (
        ...
        “github.com/testcontainers/testcontainers-go/modules/neo4j”
)
func TestNeo4j(t *testing.T) {
        ctx := context.Background()
        container, err := neo4j.RunContainer(ctx,
                testcontainers.WithImage(“neo4j:4.4”),
                neo4j.WithAdminPassword(“p4ssw0rd!”),
                neo4j.WithNeo4jSettings(map[string][string {“key.a”, “valueA”}),
                neo4j.WithNeo4jSetting(“key.b”, “valueB”),
                neo4j.WithLabsPlugins(neo4j.Apoc),
        )
        if err != nil {
                t.Fatal(err)
        }
}
```

https://golang.testcontainers.org/modules/neo4j/

Using Go modules

Ex #3 Localstack

```
RunContainer(
    ctx, opts ...Customizers,
)
```

```go
package main_test
import (
    ...
    "github.com/testcontainers/testcontainers-go/modules/localstack"
)
func TestLocalstack(t *testing.T) {
    ctx := context.Background()
    container, err := localstack.RunContainer(ctx,
        testcontainers.WithImage("localstack:2.2.0"),
    )
    if err != nil {
        t.Fatal(err)
    }
}
```

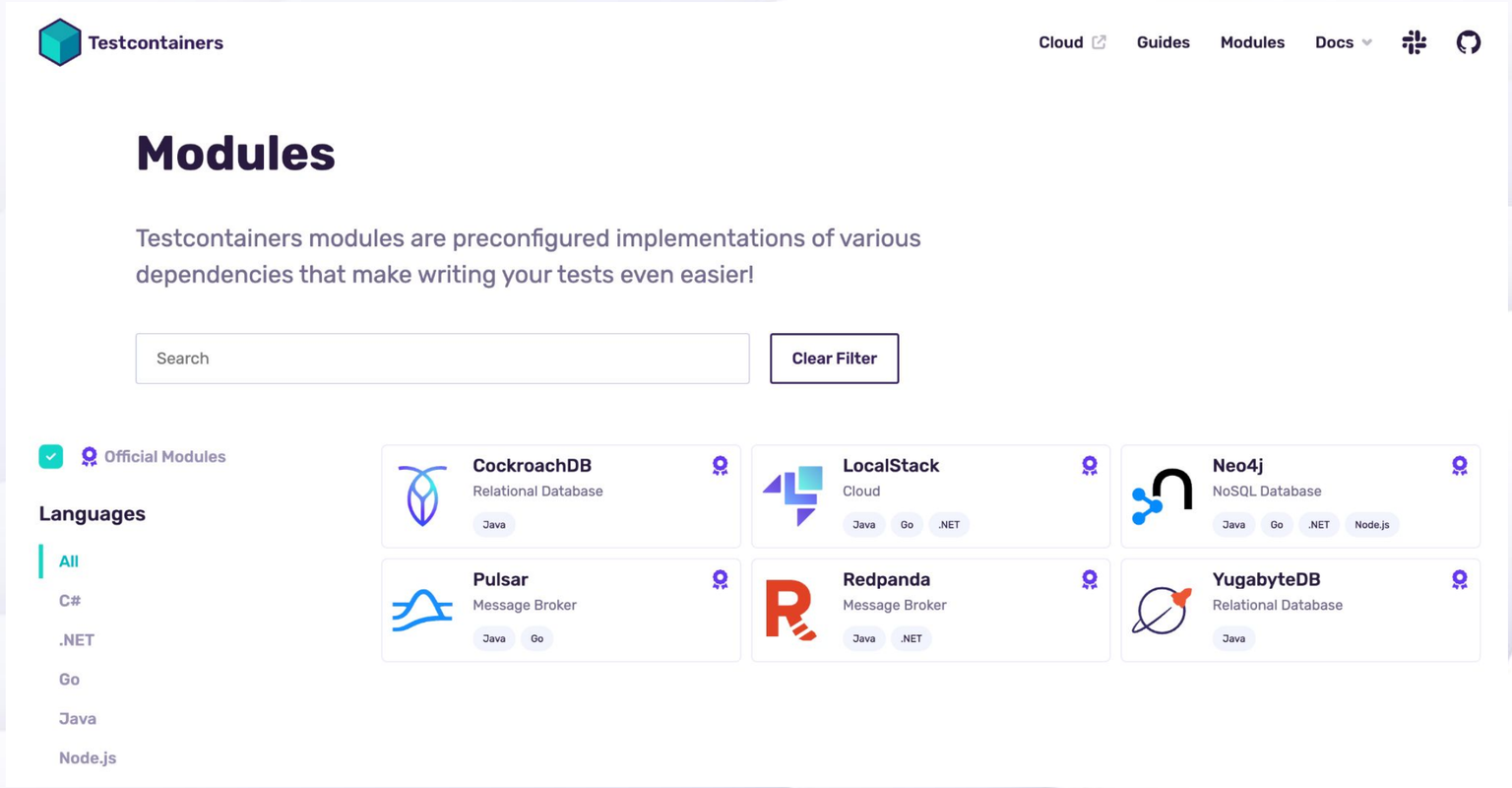
https://golang.testcontainers.org/modules/localstack/

Using Testcontainers for Go

Official Modules

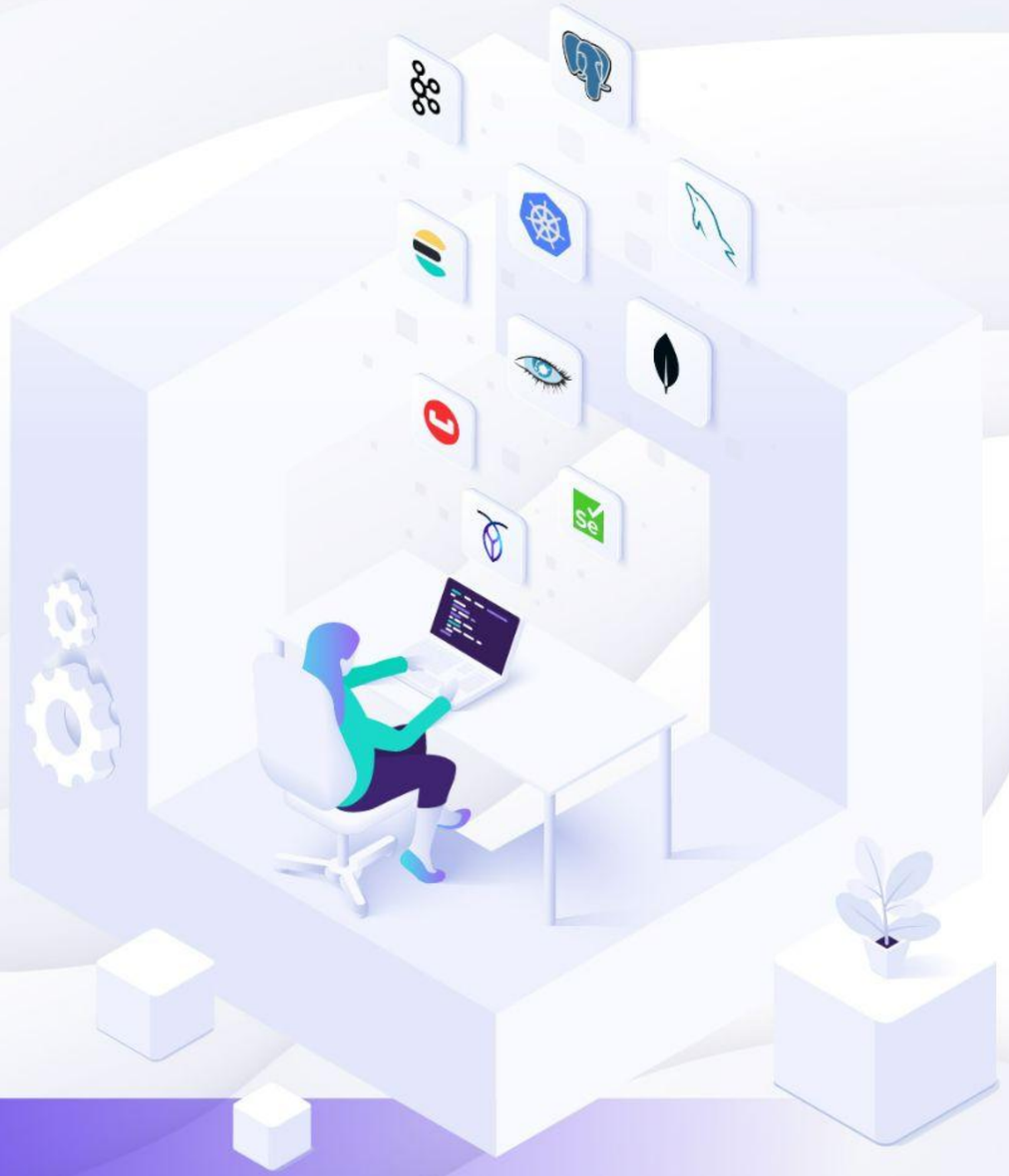
Major vendors backing the development of the modules for all the languages!

A super valuable tool for CI.



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

Testcontainers Cloud BY ATOMICJAR

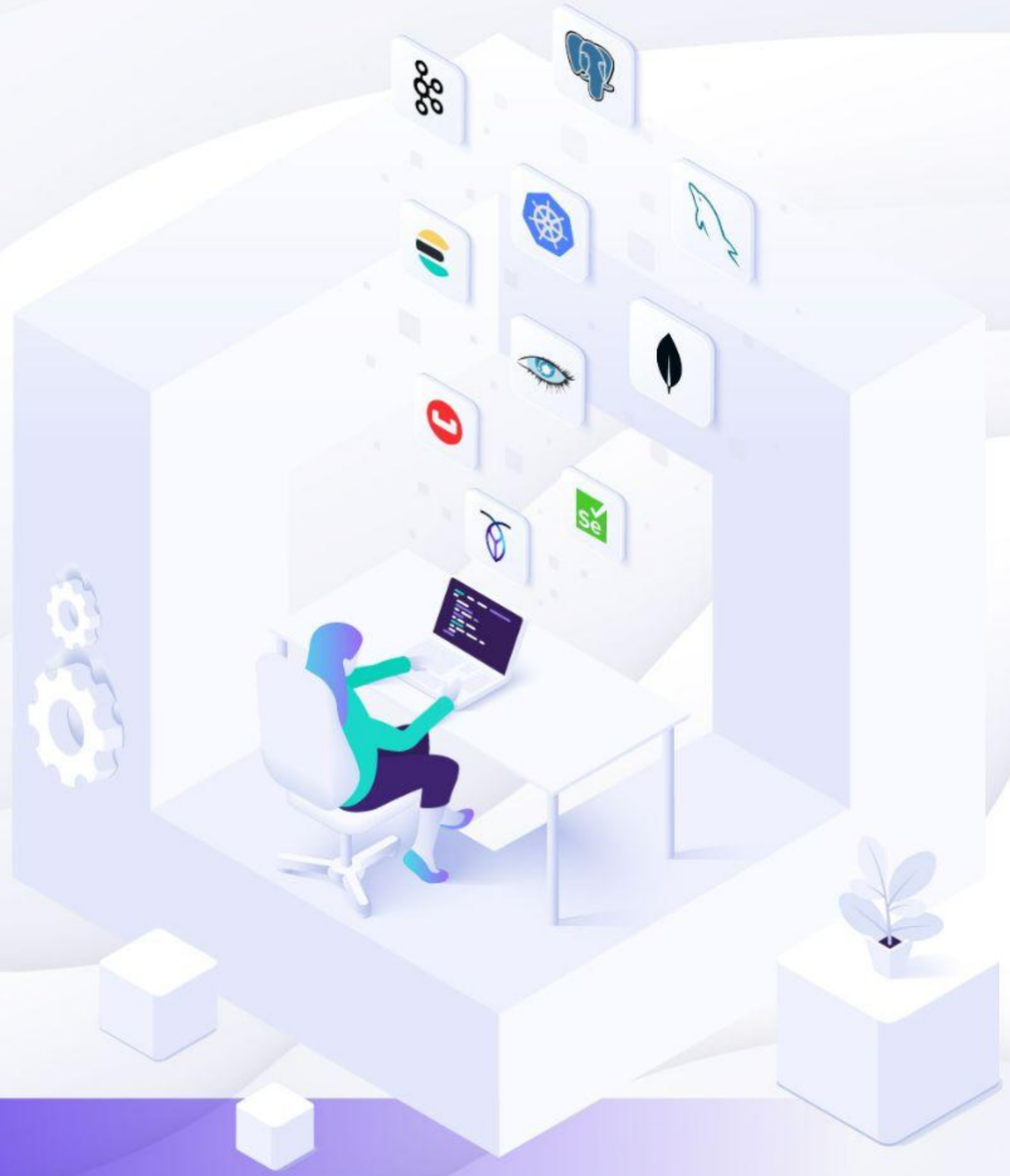


Go to testcontainers.cloud and start testing!

Testcontainers Cloud

BY ATOMICJAR

https://bit.ly/tcc-commit-conf-2023



Go to testcontainers.cloud and start testing!

Demo

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

Thanks!

https://slack.testcontainers.org

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

https://golang.testcontainers.org

https://bit.ly/tcc-commit-conf-2023

@mdelapenya everywhere

ATOMICJAR