Enterprise Testing using Arquillian

Rafael M. Pestano - Procergs

Agenda

Agenda

1. Testing is too hard
2. Testing is no fun
3. Testing is sloooow
4. There’s no time to test
5. Testing **sucks!**
6. Testing **sucks!**
7. Testing **sucks!**
8. Testing **sucks!**

testing...

Testing is too hard

Testing is too hard

Testing is no fun

Testing is too hard

Testing is no fun

Testing is sloooow

Testing is too hard

Testing is no fun

Testing is sloooow

There’s no time to test

Testing is too hard

Testing is no fun

Testing is sloooow

There’s no time to test

Testing **sucks!**

Think again!

You can’t fix what you can’t run

You can’t fix what you can’t debug

You can’t fix what you can’t test

You can’t develop if you can’t test

testing is
development

> The purpose of automated testing is to enable change.
>
> Verifying correctness is just a nice side-effect.
>
> — Jeremy Norris

Not testing is painful & time consuming

Testing eliminates this pain and sacrifice

Testing **rulez!**

TESTING LENS RECAP

DEVELOP BY TESTING

DON’T WASTE TIME BEING STUCK

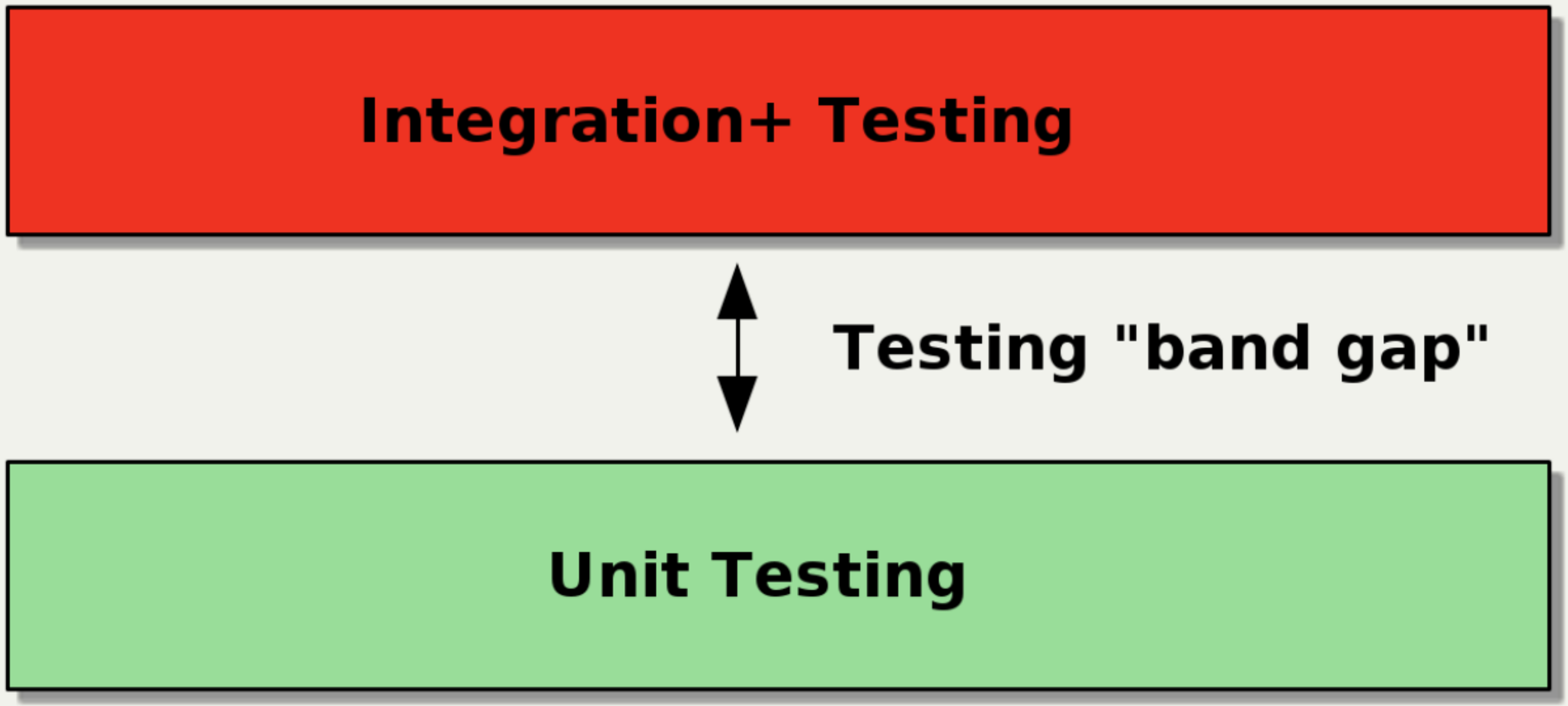
our philosophy

Tests should be portable across compatible environments

Tests should be executable from the IDE and the build tool

The test platform should reuse existing frameworks and tools

3


Integration+ Testing
Testing "band gap"
Unit Testing

Bring your tests to the runtime...

...so you rule your code, not the bugs!

arquillian - the enterprise test platform

arquillian.org

Designed for humans

IDE friendly ☺

Can be fully automated

Continuous integration, FTW! ∞

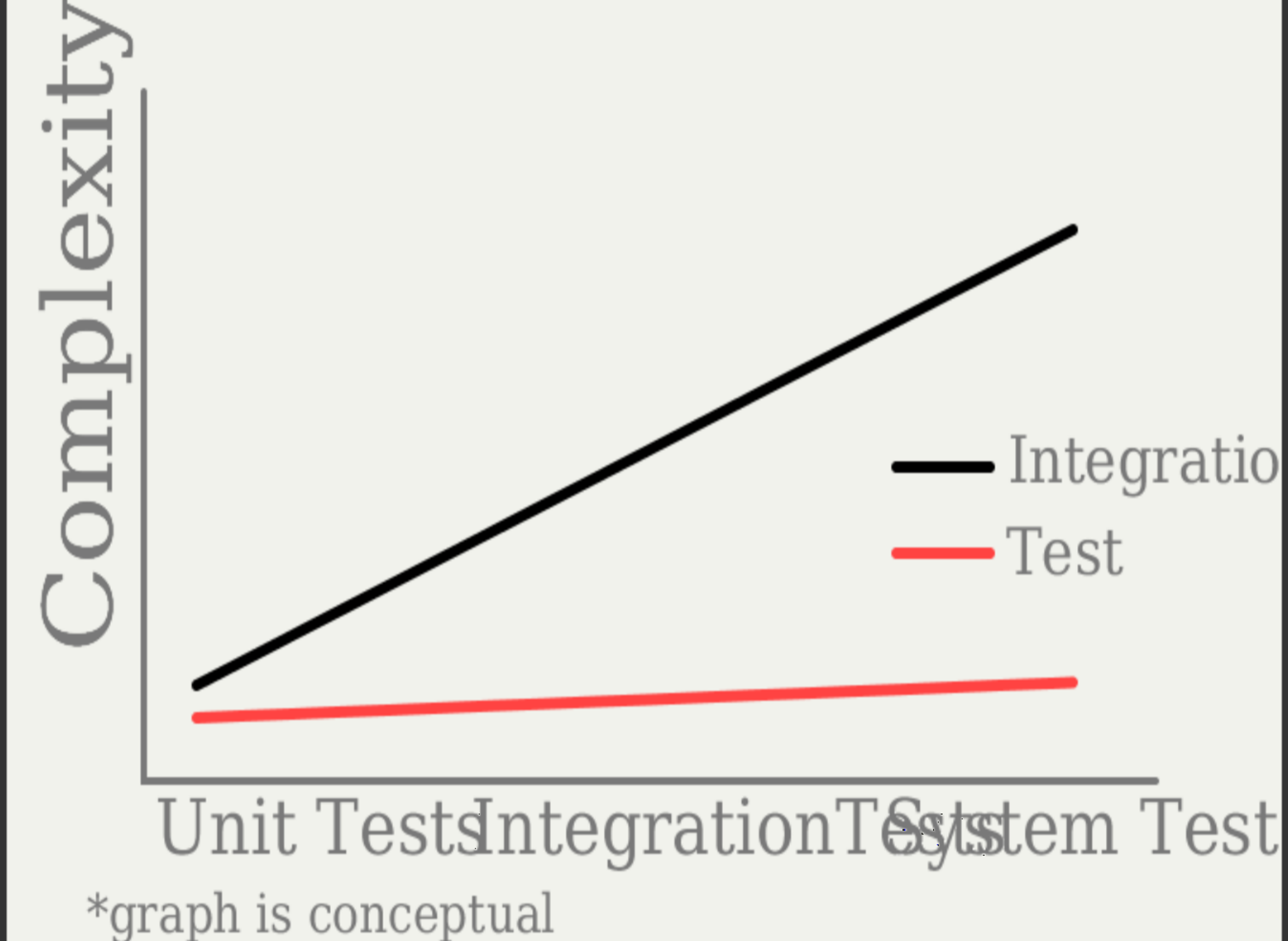
Provides a component model for testing

Dependency injection, yeah! ↠

Isolates the code under test

Tame that classpath ⋔

test continuum



*graph is conceptual

TEST RUNNER

TEST RUNNER
CONTAINER ADAPTER

TEST RUNNER
CONTAINER ADAPTER
+ TEST ENRICHER

TEST RUNNER
CONTAINER ADAPTER
+ TEST ENRICHER

TEST RUNNER
CONTAINER ADAPTER
+ TEST ENRICHER

IN-CONTAINER TESTING

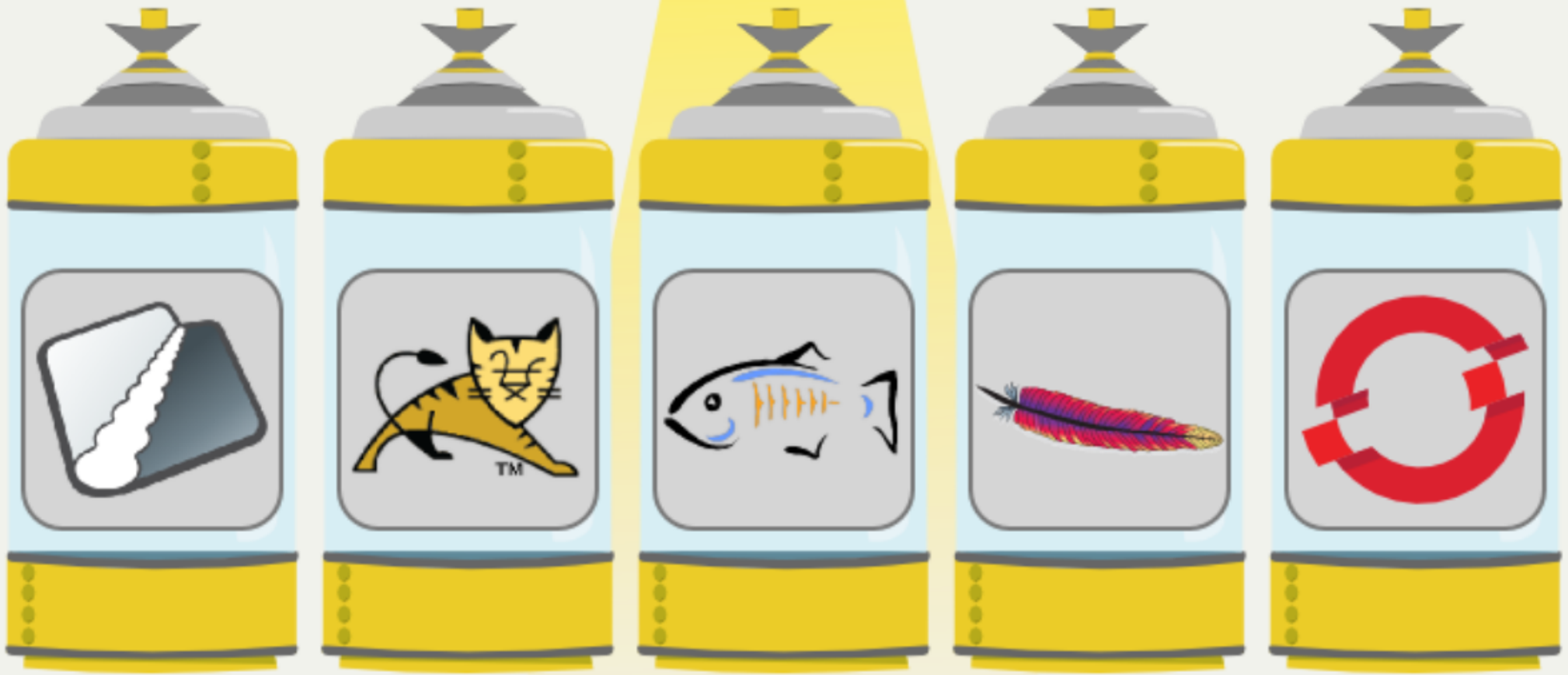
ARQUILLIAN "HELLO, WORLD!" TEST

```
@RunWith(Arquillian.class)
public class GreeterTest {

    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class, "test.jar")
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Inject
    Greeter greeter;

    @Test
    public void should_create_greeting() {
        Assert.assertEquals(
```

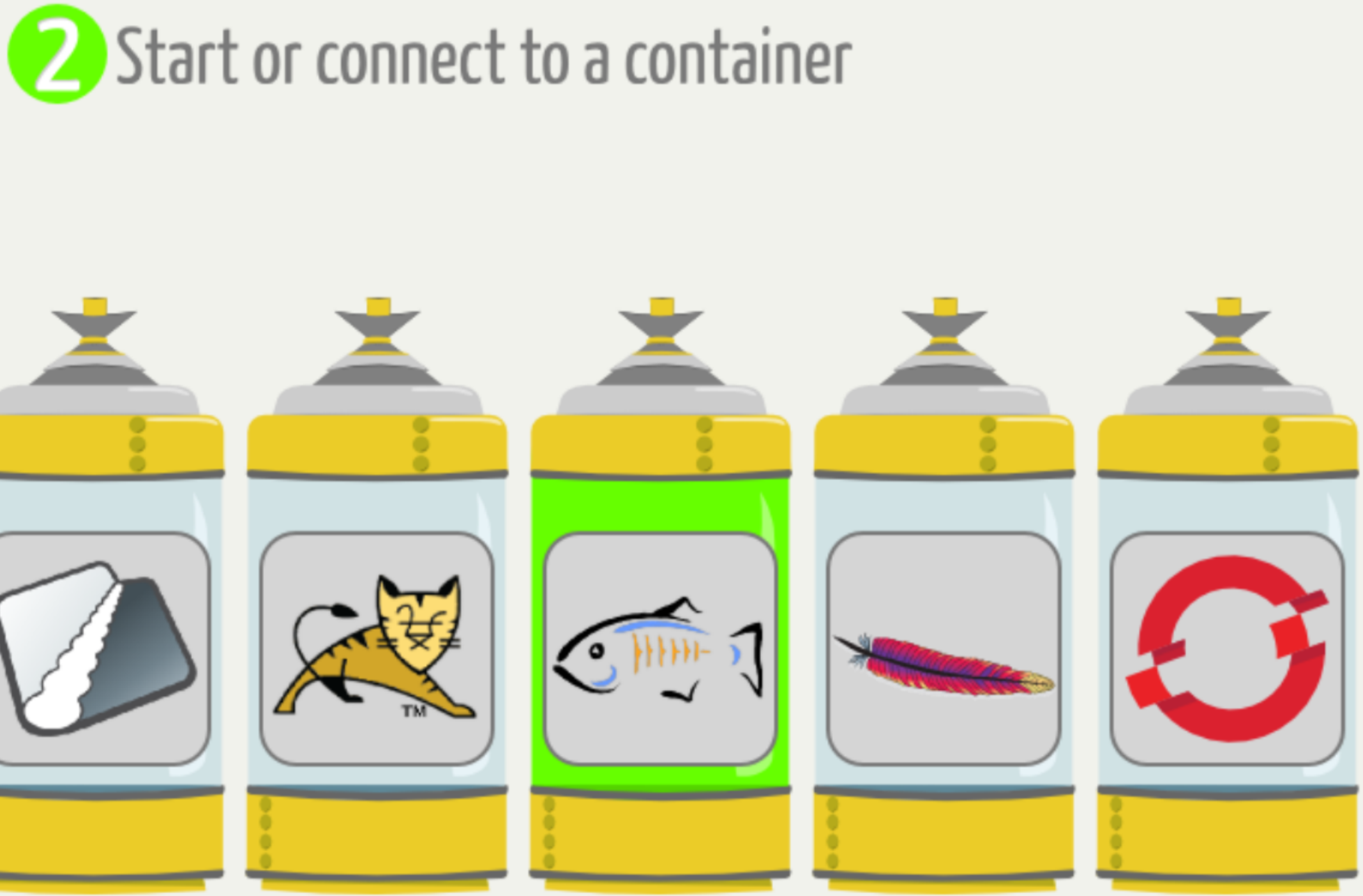
Run As... > JUnit Test

Select container

1 Select a container

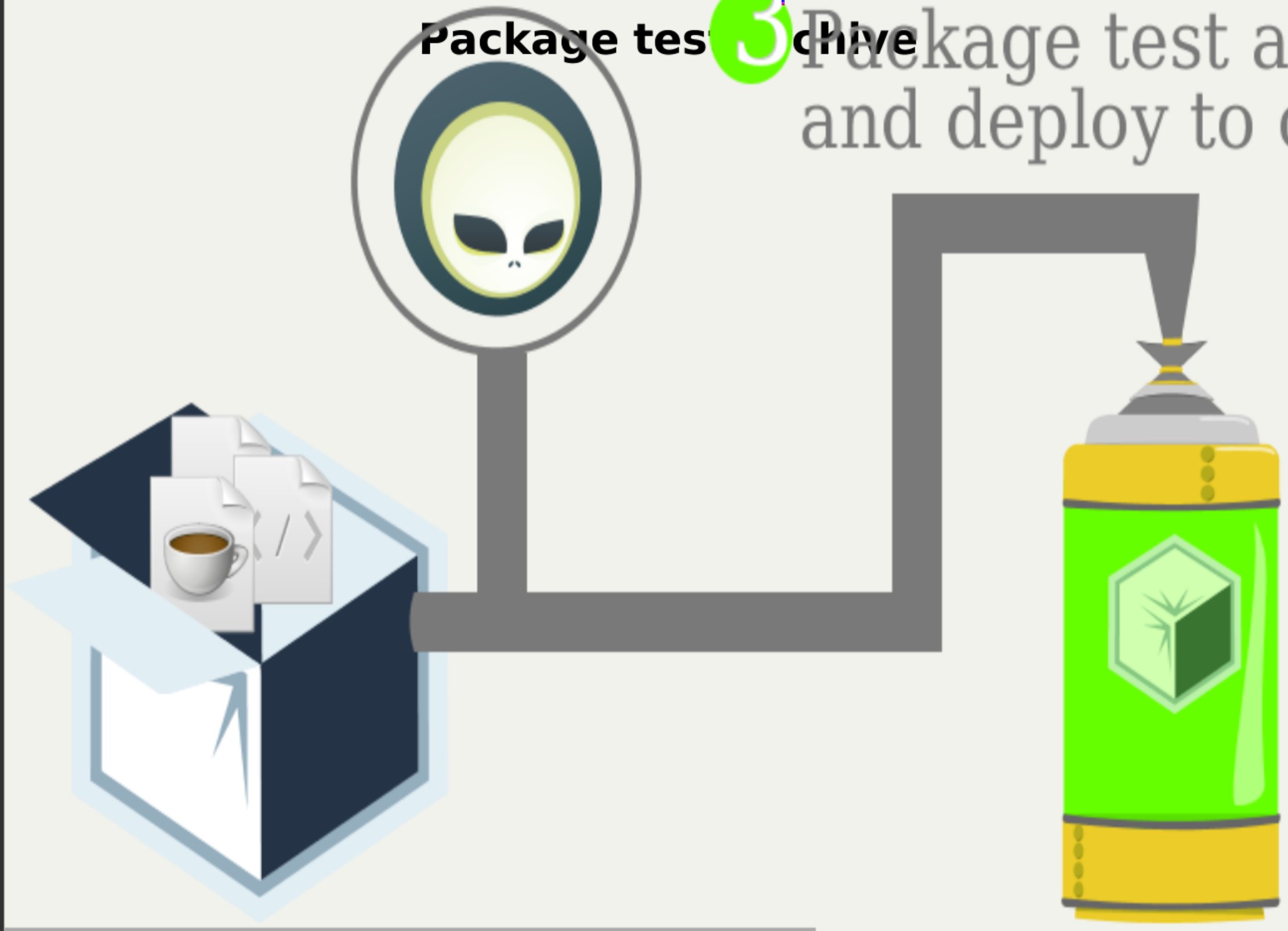


Start or connect to container

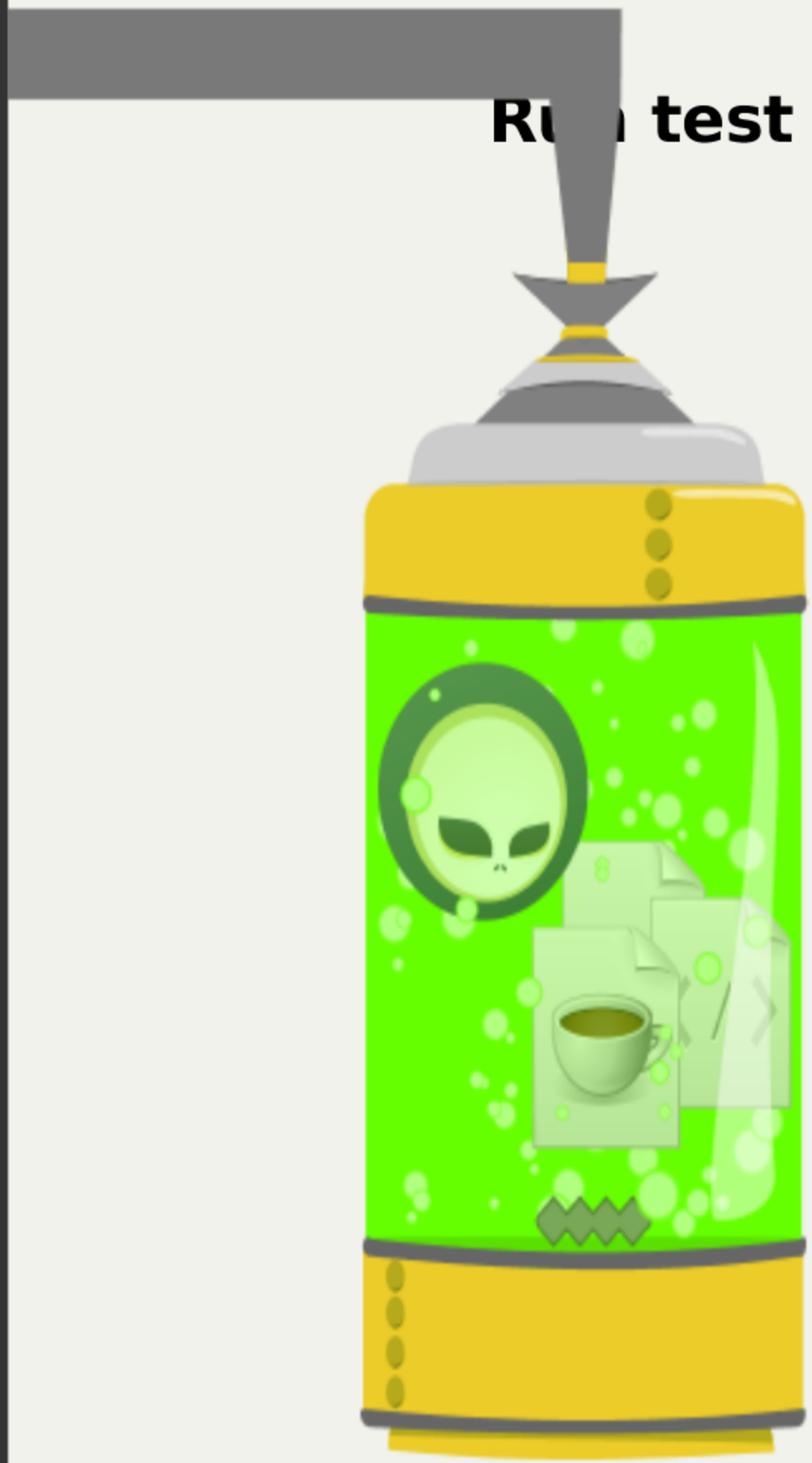


Package tes[3]chive

Package test ar
and deploy to c

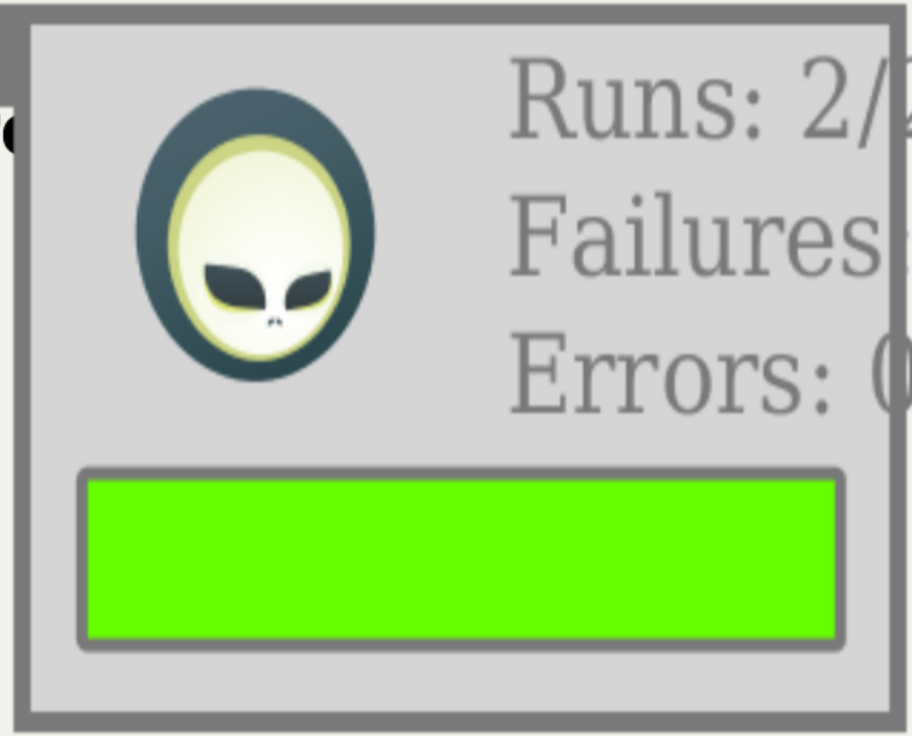


4 Run test inside the container



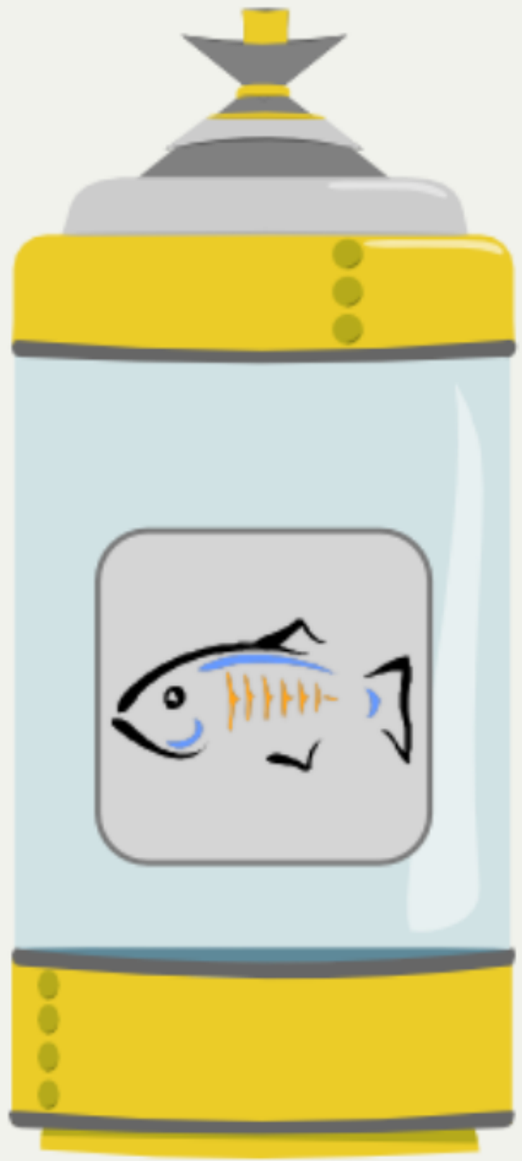
Run test in-cont

Collect test r



5 Capture and rep
test results

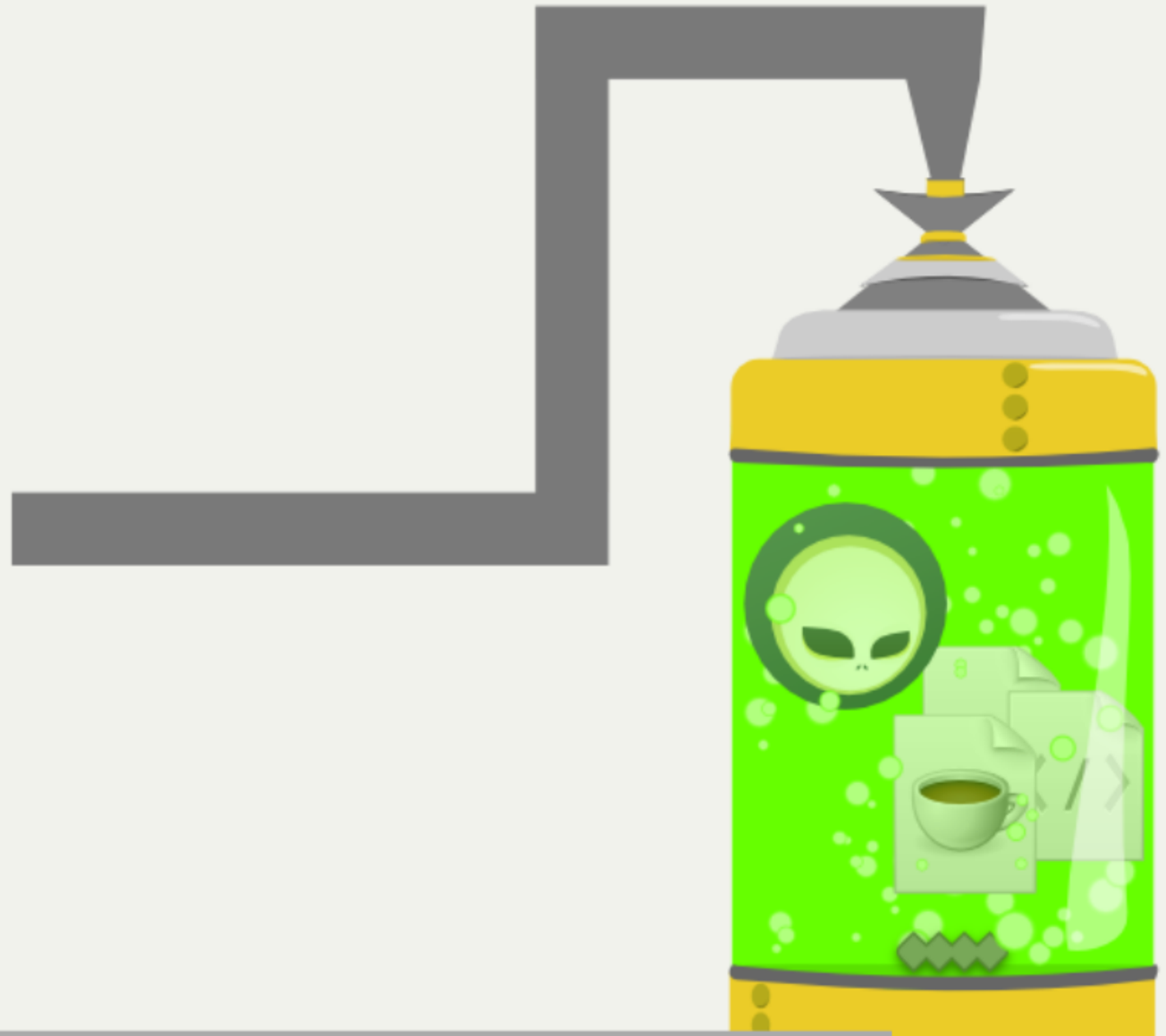
Stop or disconnect from container



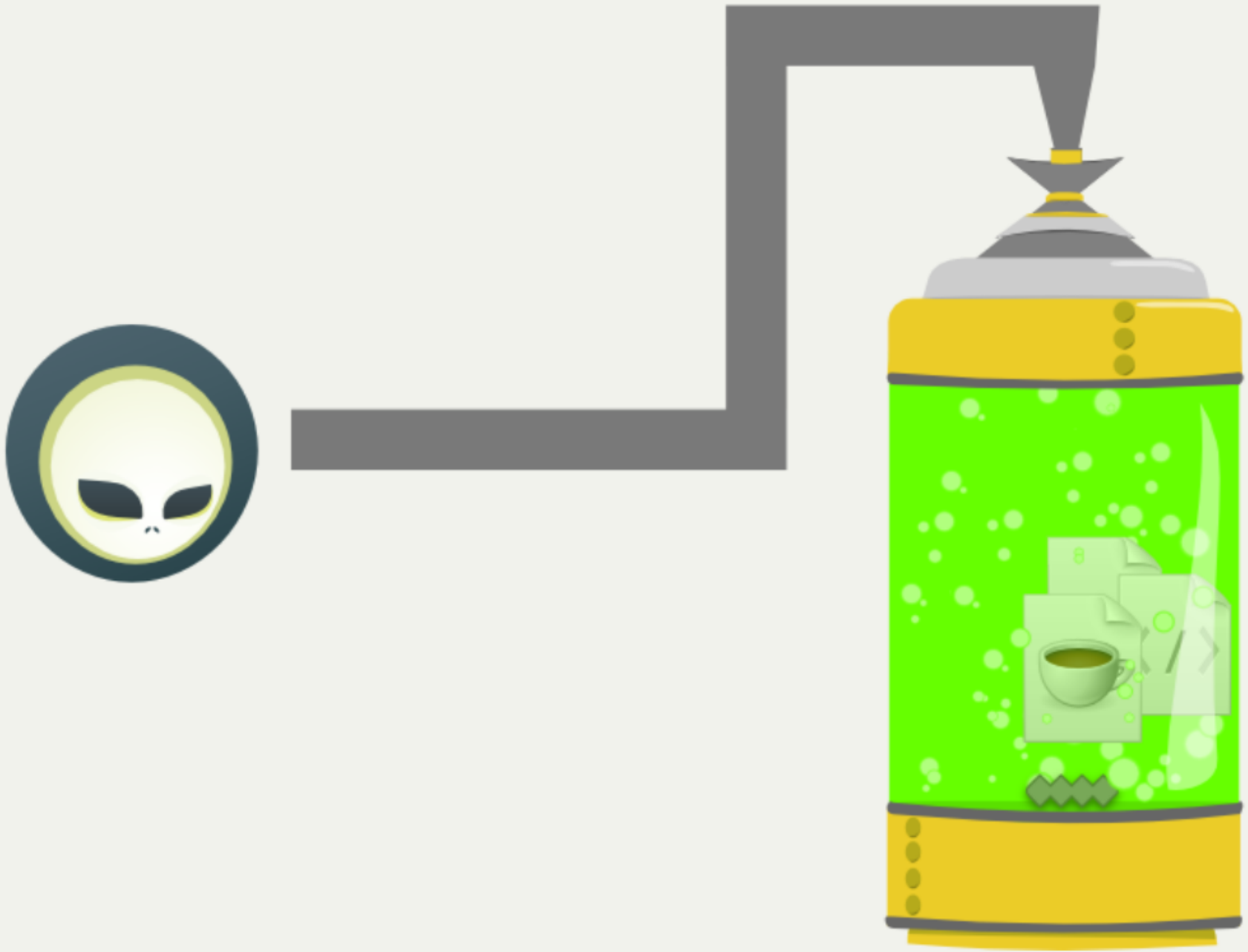
6 Undeploy test archive and stop or disconnect from container

arquillian - the extensible test platform

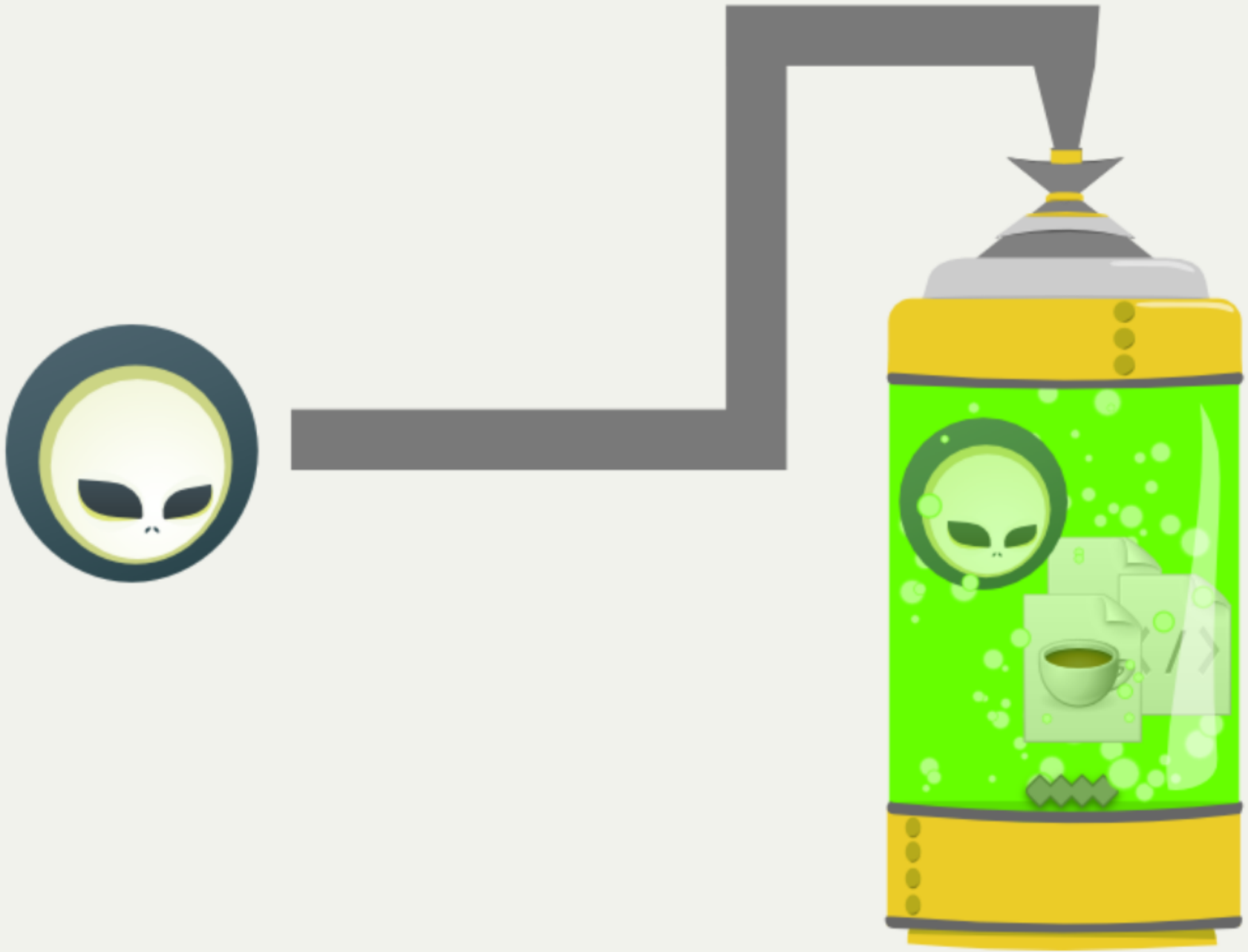
Arquillian Persistence



Arquillian Drone



Arquillian Warp



Reject complex, jacked up test suites

TEST CONTINUUM RECAP

YOU WORK IN THE IDE

CI SERVERS WORK THE BUILD

test to learn

Question everything

arquillian - a learning test platform

write real tests

Real tests increase certainty

Real tests build confidence

Real tests increase robustness

Testing with mocks only proves you know how to write mocks

LEARNING RECAP

USE TESTS TO LEARN

KEEPS KNOWLEGE IN THE SYSTEM

Java enterprise testing + arquillian

=

child’s play

thanks!

Dan Allen

Twitter or GitHub:
@mojavelinux