
Making your own
Testcontainers module
for fun and profit!

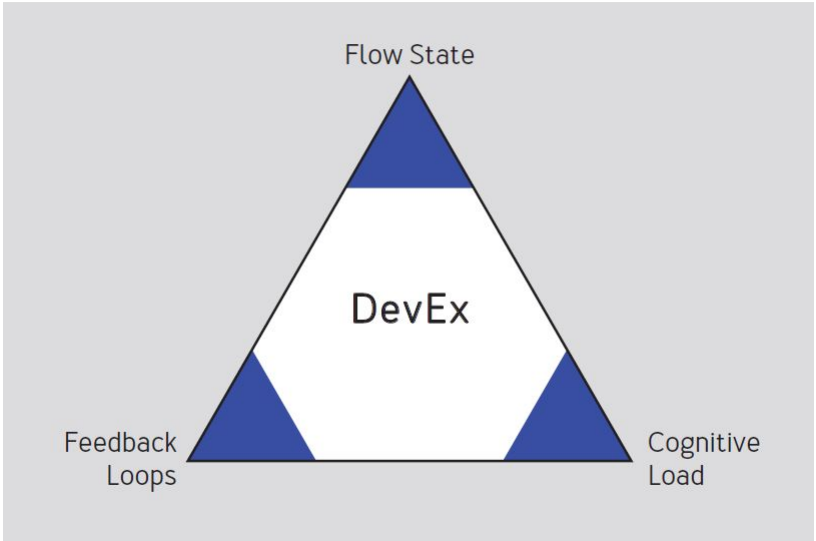
What is productivity?

The developer-centric approach to measuring and improving productivity.

Abi Noda, DX
Margaret-Anne Storey, University of Victoria
Nicole Forsgren, Microsoft Research
Michaela Greiler, DX

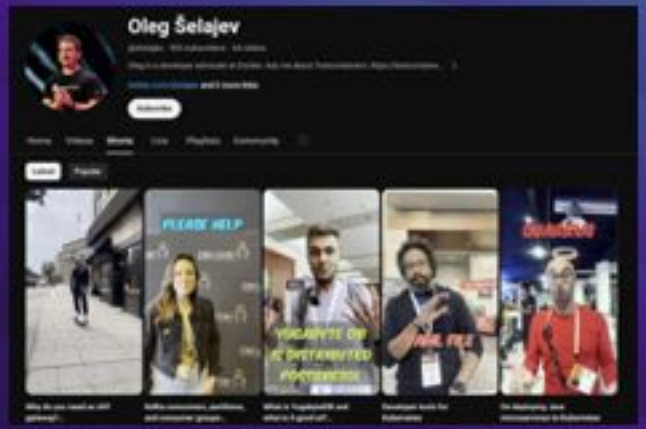
`https://queue.acm.org/detail.cfm?id=3595878`

FIGURE 1: **THREE CORE DIMENSIONS OF DEVELOPER EXPERIENCE**



@shelajev - Oleg Šelajev

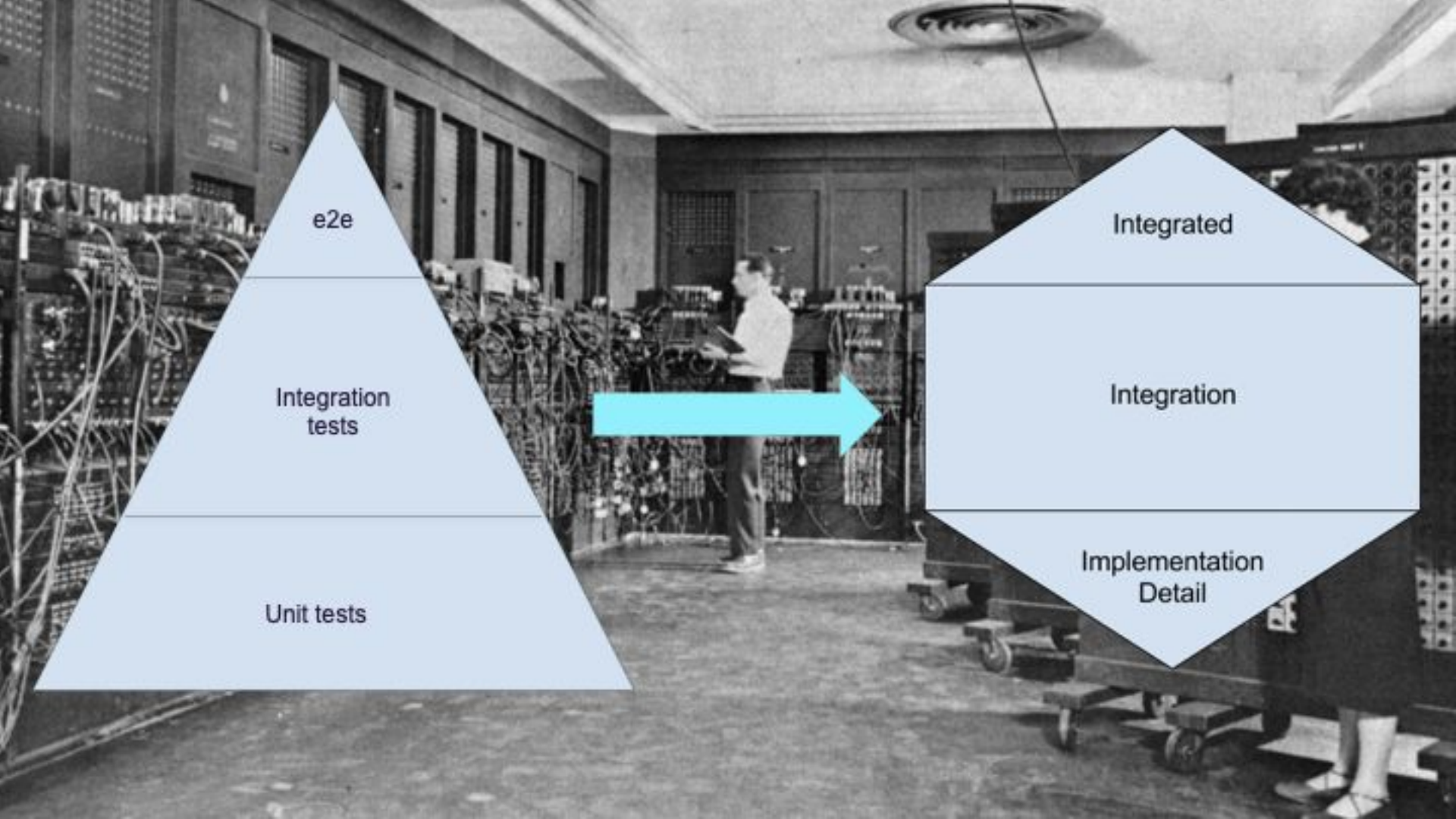
AI, Developer productivity, and Testcontainers
@ Docker



youtube.com/@shelajev


TESTS


Dev
CI
Prod


e2e
Integration tests
Unit tests
Integrated
Integration
Implementation Detail

Testcontainers

Test dependencies as code

Testcontainers is an open source library
for providing ephemeral, lightweight instances
of test dependencies.

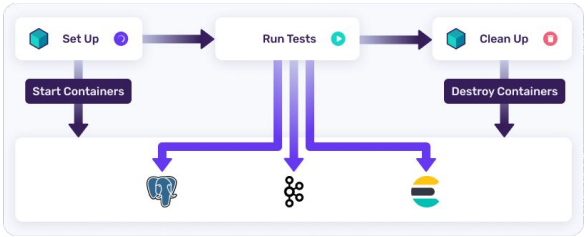
```
var redis = new GenericContainer("redis:6-alpine").withExposedPorts(6379)
```

Testcontainers: Test dependencies with real services wrapped in Docker containers

```
var granite = new GenericContainer("redhat/granite-7b-lab-gguf").withCommand("--serve")
```

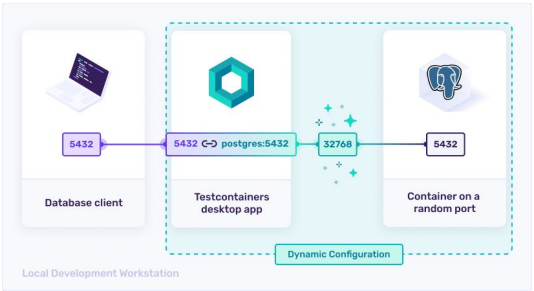
Testcontainers Library

Fast, reasistic, cost-effective dependencies



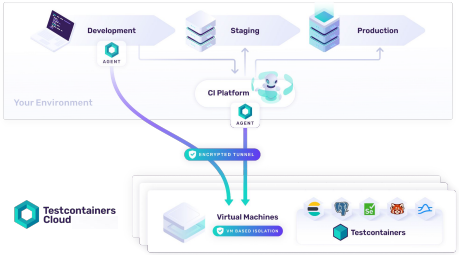
Testcontainers Desktop

Better local development and debugging experience



Testcontainers Cloud

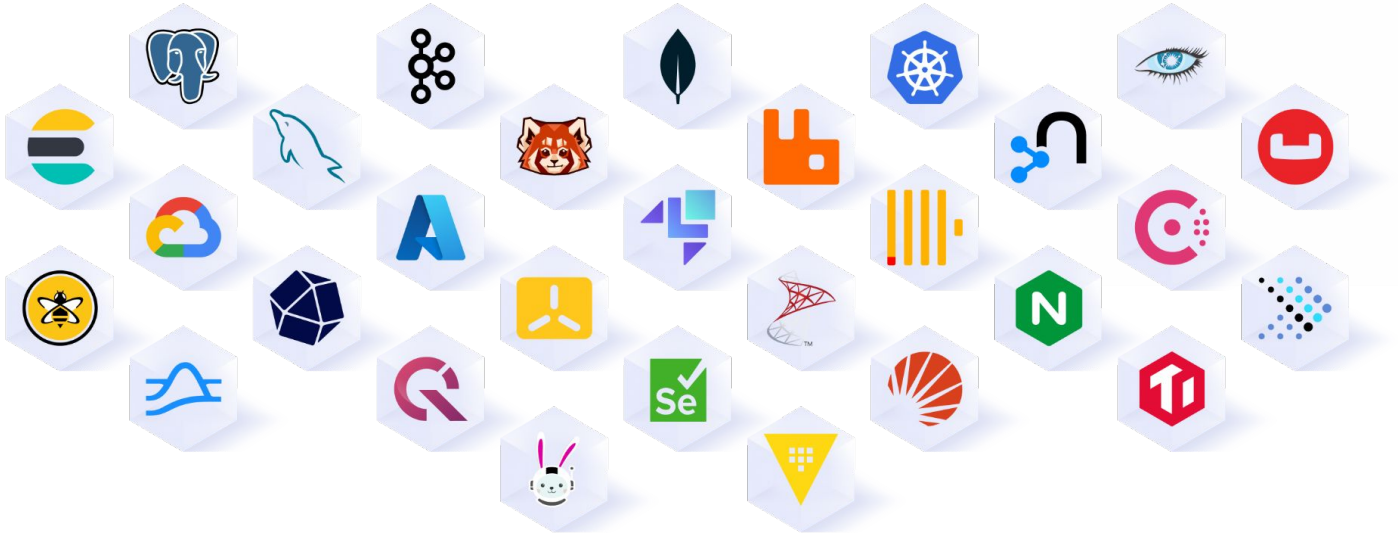
Operationalize and roll out Testcontainers across the teams

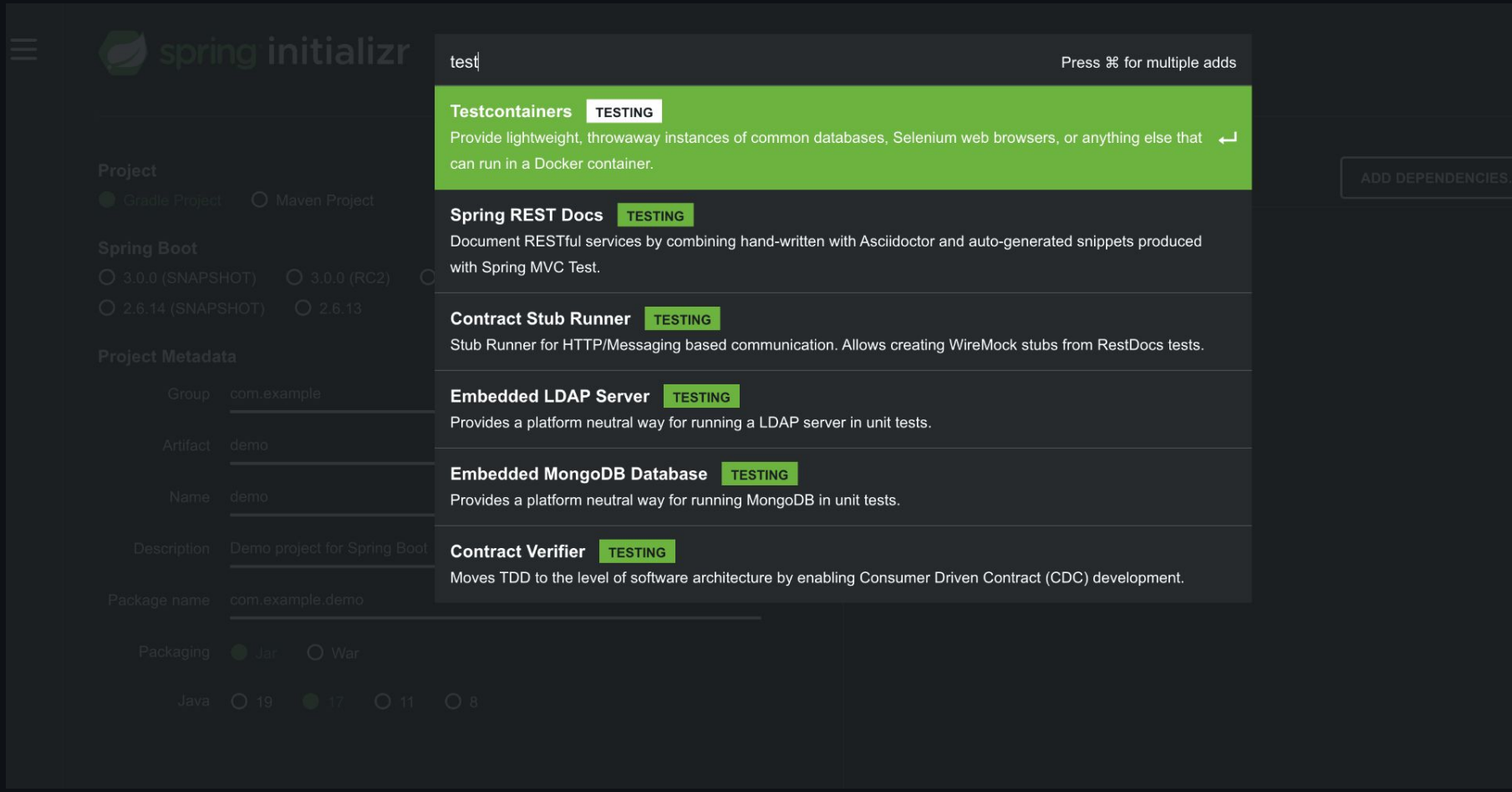


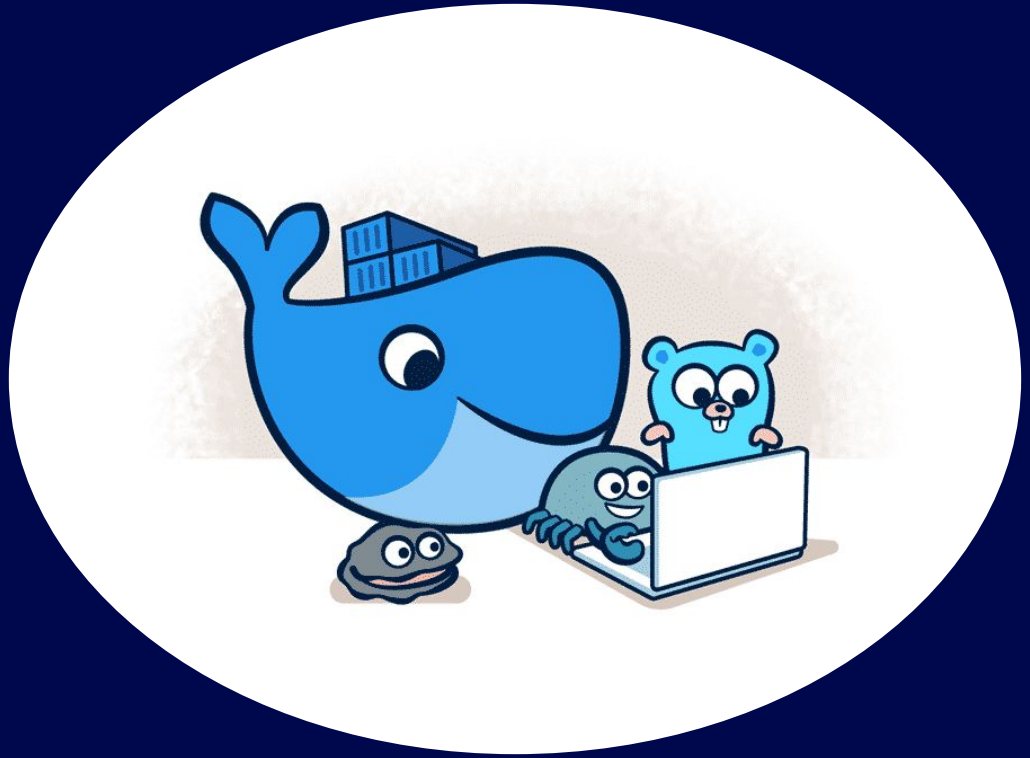
Testcontainers modules

Test against any database, message broker, browser...

or just about anything that runs in a Docker container!




spring initializr
test
Press ⌘ for multiple adds
Testcontainers
TESTING
Provide lightweight, throwaway instances of common databases, Selenium web browsers, or anything else that can run in a Docker container.
Spring REST Docs
TESTING
Document RESTful services by combining hand-written with Asciidoctor and auto-generated snippets produced with Spring MVC Test.
Contract Stub Runner
TESTING
Stub Runner for HTTP/Messaging based communication. Allows creating WireMock stubs from RestDocs tests.
Embedded LDAP Server
TESTING
Provides a platform neutral way for running a LDAP server in unit tests.
Embedded MongoDB Database
TESTING
Provides a platform neutral way for running MongoDB in unit tests.
Contract Verifier
TESTING
Moves TDD to the level of software architecture by enabling Consumer Driven Contract (CDC) development.
ADD DEPENDENCIES... ⌘ +
Project
Gradle Project
Maven Project
Spring Boot
3.0.0 (SNAPSHOT)
3.0.0 (RC2)
2.6.14 (SNAPSHOT)
2.6.13
Project Metadata
Group
com.example
Artifact
demo
Name
demo
Description
Demo project for Spring Boot
Package name
com.example.demo
Packaging
Jar
War
Java
19
17
11
8



Testcontainers module functionality

Container & Service Configuration
Lifecycle API
Convenience methods

```java
public class SupabaseContainer extends GenericContainer<SupabaseContainer> {
    public SupabaseContainer(DockerImageName dockerImageName) {

    }
```

```java
public class SupabaseContainer extends GenericContainer<SupabaseContainer> {
    public SupabaseContainer(DockerImageName dockerImageName) {



        withEnv("POSTGRES_PASSWORD", "password");
        withExposedPorts(5432);



    }
```

```java
public class SupabaseContainer extends GenericContainer<SupabaseContainer> {
    public SupabaseContainer(DockerImageName dockerImageName) {
        super(dockerImageName);
        dockerImageName.assertCompatibleWith(DockerImageName.parse( fullImageName: "supabase/postgres"));

        withEnv("POSTGRES_PASSWORD", "password");
        withExposedPorts(5432);



    }
```

```java
public class SupabaseContainer extends GenericContainer<SupabaseContainer> {
    public SupabaseContainer(DockerImageName dockerImageName) {
        super(dockerImageName);
        dockerImageName.assertCompatibleWith(DockerImageName.parse( fullImageName: "supabase/postgres"));

        withEnv("POSTGRES_PASSWORD", "password");
        withExposedPorts(5432);

        waitingFor(
                Wait.forLogMessage( regex: ".*database system is ready to accept connections.*\\s",
                        times: 2));
    }
```

```
protected void containerIsStarted(InspectContainerResponse containerInfo) {...}   Gener
protected void containerIsStarted(InspectContainerResponse contain…   GenericContainer
protected void containerIsCreated(String containerId) {...}           GenericContainer
protected void containerIsStarting(InspectContainerResponse contai…   GenericContainer
protected void containerIsStarting(InspectContainerResponse contai…   GenericContainer
protected void containerIsStopped(InspectContainerResponse contain…   GenericContainer
protected void containerIsStopping(InspectContainerResponse contai…   GenericContainer
```

Press ^Space to see non-imported classes Next Tip

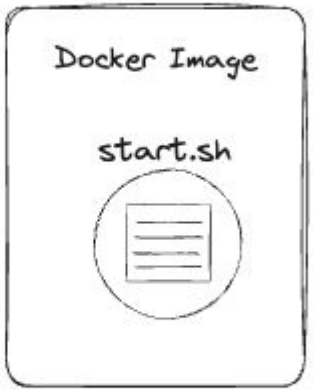
protected void configure()

```
private String password;

public SupabaseContainer withPassword(String testpassword) {
    this.password = testpassword;
    return this;
}

@Override
protected void configure() {
    super.configure();
    withEnv("POSTGRES_PASSWORD", password);
}
```

Convenience methods

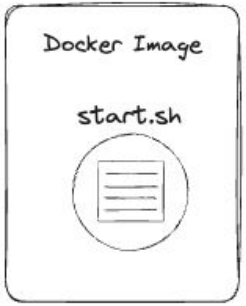
```
public String getJdbcUrl() {
    return "jdbc:postgresql://" + getHost() + ":" +
            getMappedPort( originalPort: 5432) + "/postgres";
}

public String getUsername() {
    return "postgres";
}

public String getPassword() {
    return password;
}
```

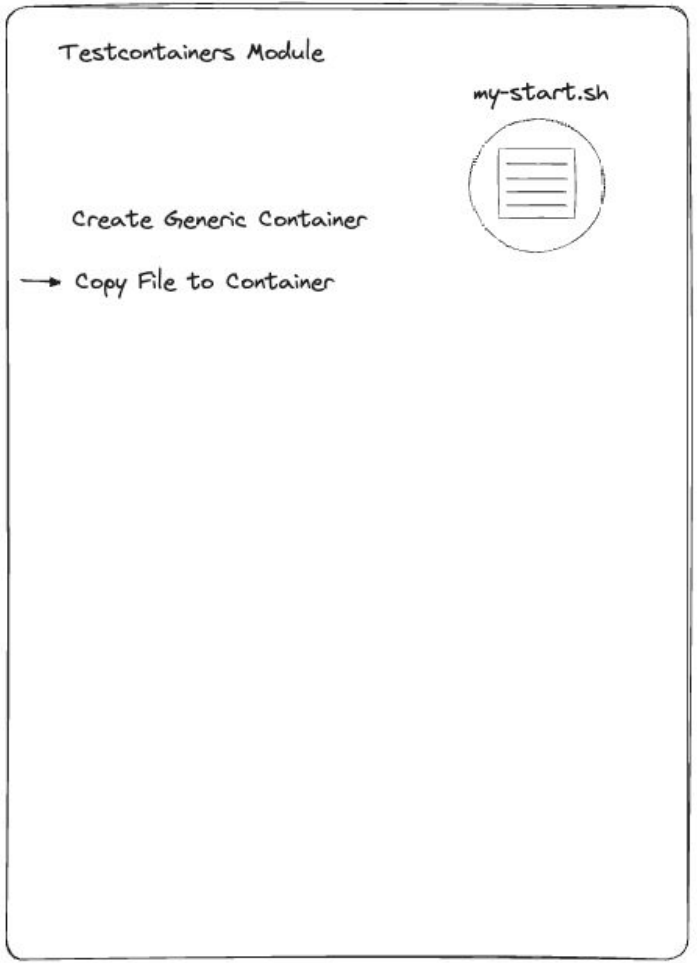

WARNING
Security
Cameras In Use
ARROW
Sanitary Service
503-257-3993
A WASTE CONNECTIONS COMPANY
CAUTION
CAUTION
GARBAGE
TEST!

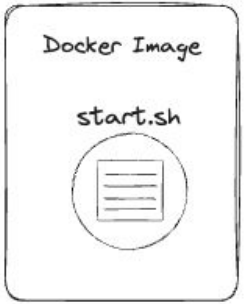


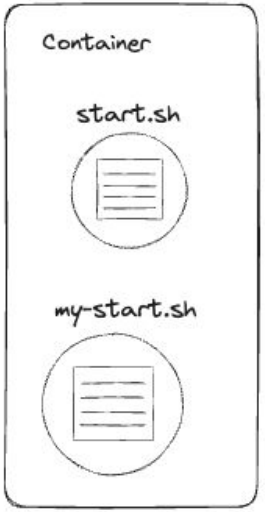
Custom Commands

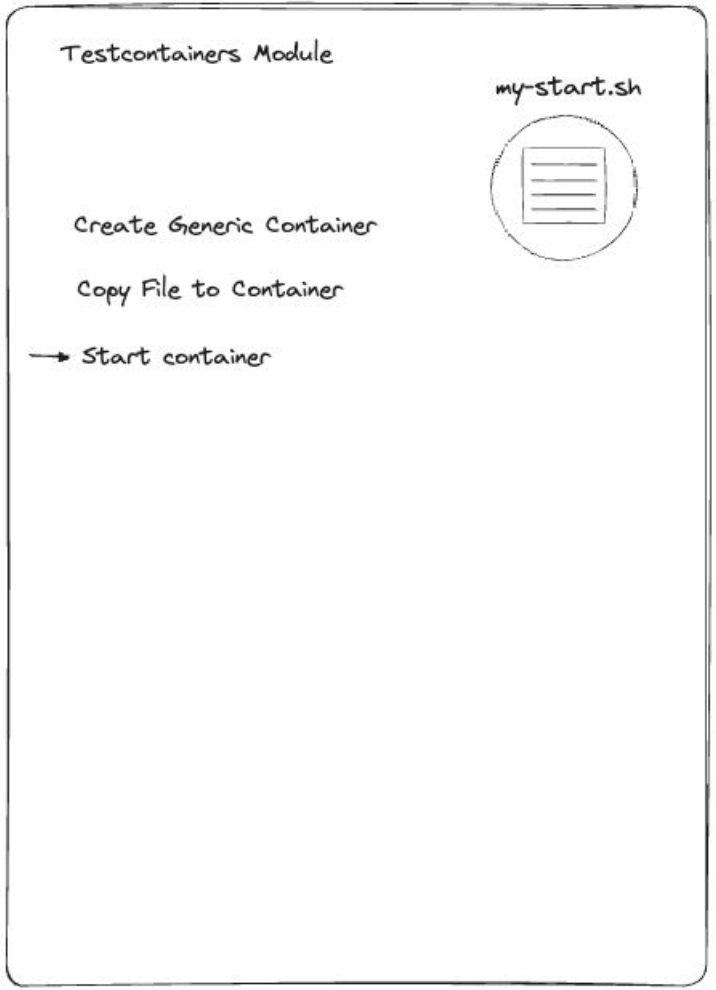

Testcontainers Module
my-start.sh
Create Generic Container

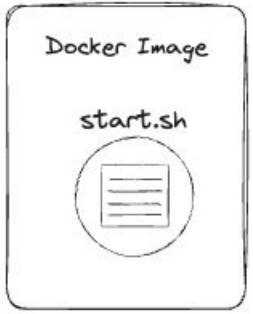

Docker Image
start.sh

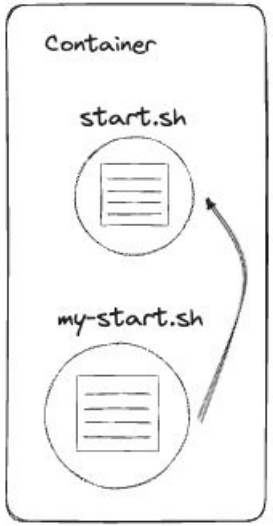

Testcontainers Module
my-start.sh
Create Generic Container
Copy File to Container


Docker Image
start.sh


Container
start.sh
my-start.sh


Testcontainers Module
my-start.sh
Create Generic Container
Copy File to Container
Start container


Docker Image
start.sh


Container
start.sh
my-start.sh

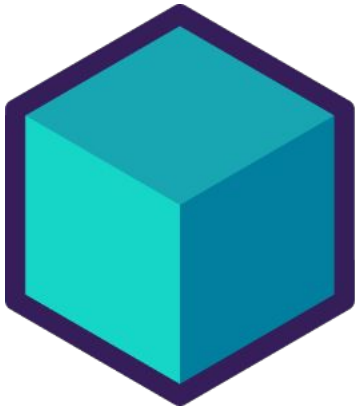
KafkaContainer

```
this.setCommand(new String[]{"-c",
"while [ ! -f /testcontainers_start.sh ];
do
  sleep 0.1;
done;
/testcontainers_start.sh"});
```

KafkaContainer

```
String kafkaAdvertisedListeners = String.join( delimiter: ",", advertisedListeners);
String command = "#!/bin/bash\n";
command = command + String.format("export KAFKA_ADVERTISED_LISTENERS=%s\n", kafkaAdvertisedListeners);
if (!this.kraftEnabled || this.isLessThanCP740()) {
    command = command + "echo '' > /etc/confluent/docker/ensure \n";
}

if (this.kraftEnabled) {
    command = command + this.commandKraft();
} else if (this.externalZookeeperConnect == null) {
    command = command + this.commandZookeeper();
}

command = command + "/etc/confluent/docker/run \n";
this.copyFileToContainer(Transferable.of(command, fileMode: 511), containerPath: "/testcontainers_start.sh");
```


Networks



Testcontainers

Docker configuration

```
Info info = this.dockerClient.infoCmd().exec();
Map<String, RuntimeInfo> runtimes = info.getRuntimes();
if (runtimes != null) {
    if (runtimes.containsKey("nvidia")) {
        withCreateContainerCmdModifier(cmd -> {
            cmd
                    .getHostConfig()
                    .withDeviceRequests(
                            Collections.singletonList(
                                    new DeviceRequest()
                                            .withCapabilities(
                                                    Collections.singletonList(
                                                            Collections.singletonList("gpu")))
                                            .withCount(-1)
                            )
                    );
        });
    }
}
```

```
        .withCreateContainerCmdModifier(createContainerCmd -> {
                var hostConfig = new HostConfig();
                hostConfig.withMemory(memoryLimitInMB * 1024L * 1024L);
                hostConfig.withCpuCount(1L);
                createContainerCmd.withHostConfig(hostConfig);
```

```
@CsvSource({
        ".*SerialGC.*true.*, 1791",
        ".*G1GC.*true.*, 1792"}
)
@ParameterizedTest
void doSomethingWithCreate(String gcRegex, long memoryLimitInMB) throws IOException {
    try (var container =
                 new GenericContainer<>( dockerImageName: "eclipse-temurin:17-jdk")
                         .withCreateContainerCmdModifier(createContainerCmd -> {
                                 var hostConfig = new HostConfig();
                                 hostConfig.withMemory(memoryLimitInMB * 1024L * 1024L);
                                 hostConfig.withCpuCount(1L);
                                 createContainerCmd.withHostConfig(hostConfig);
                             }
                         )
                         .withCommand( cmd: "java -XX:+PrintFlagsFinal -version && sleep infinity")
                         .withStartupCheckStrategy(new IndefiniteWaitOneShotStartupCheckStrategy())
    ) {
        container.start();
        var logs = container.getLogs();
        Assertions.assertThat(logs).containsPattern(gcRegex);
    }
}
```

Tools in containers

```
public CloudflaredContainer(DockerImageName dockerImageName, int port) {
    super(dockerImageName);
    dockerImageName.assertCompatibleWith(
            DockerImageName.parse( fullImageName: "cloudflare/cloudflared"));
    withAccessToHost( value: true);
    Testcontainers.exposeHostPorts(port);
    withCommand( ...commandParts: "tunnel", "--url",
            String.format("http://host.testcontainers.internal:%d", port));
    waitingFor(Wait.forLogMessage( regex: ".*Registered tunnel connection.*",  times: 1));
}
```

Tools in containers

```
public CloudflaredContainer(DockerImageName dockerImageName, int port) {
    super(dockerImageName);
    dockerImageName.assertCompatibleWith(
            DockerImageName.parse( fullImageName: "cloudflare/cloudflared"));
    withAccessToHost( value: true);
    Testcontainers.exposeHostPorts(port);
    withCommand( ...commandParts: "tunnel", "--url",
            String.format("http://host.testcontainers.internal:%d", port));
    waitingFor(Wait.forLogMessage( regex: ".*Registered tunnel connection.*",  times: 1));
}
```


WARNING
Security
Cameras In Use
ARROW
Sanitary Service
503-257-3993
A WASTE CONNECTIONS COMPANY
CAUTION
CAUTION
GARBAGE
TEST!

https://github.com/testcontainers/java-module-workshop
https://github.com/testcontainers/workshop
https://testcontainers.com/desktop
https://slack.testcontainers.com

Booth 35!


Testcontainers
OPEN SOURCE
Test with real dependencies
Self-contained integration and end-to-end tests
50+ pre-configured modules for databases, message brokers, cloud emulators, and more...
Reduce context switching when testing any app—including your GenAI projects
Testcontainers is used by
Spotify
NETFLIX
Uber
docker
DEVELOP FASTER. RUN ANYWHERE.
TEST
BUILD
SHARE
RUN
SESSION
Making your own Testcontainers modul
for fun and profit!
Monday 28th Oct
11:30 am-12:15 pm
Testcontainers
Join our public slack!
Observabili
helps solve
you couldn'