

API Testing Made Easy

Barkatul Mujauddin
Keploy



About me



- Developer Relation Intern, Kephloy
- Mentor, GSoC'23
- Founder, HackForCode
- Ambassador, Kubesimplify

Barkatul Mujauddin

(@barkatul_twt)



Problem

Strict Timelines



Limited Testing



Introduced Regressions



To reduce bugs...

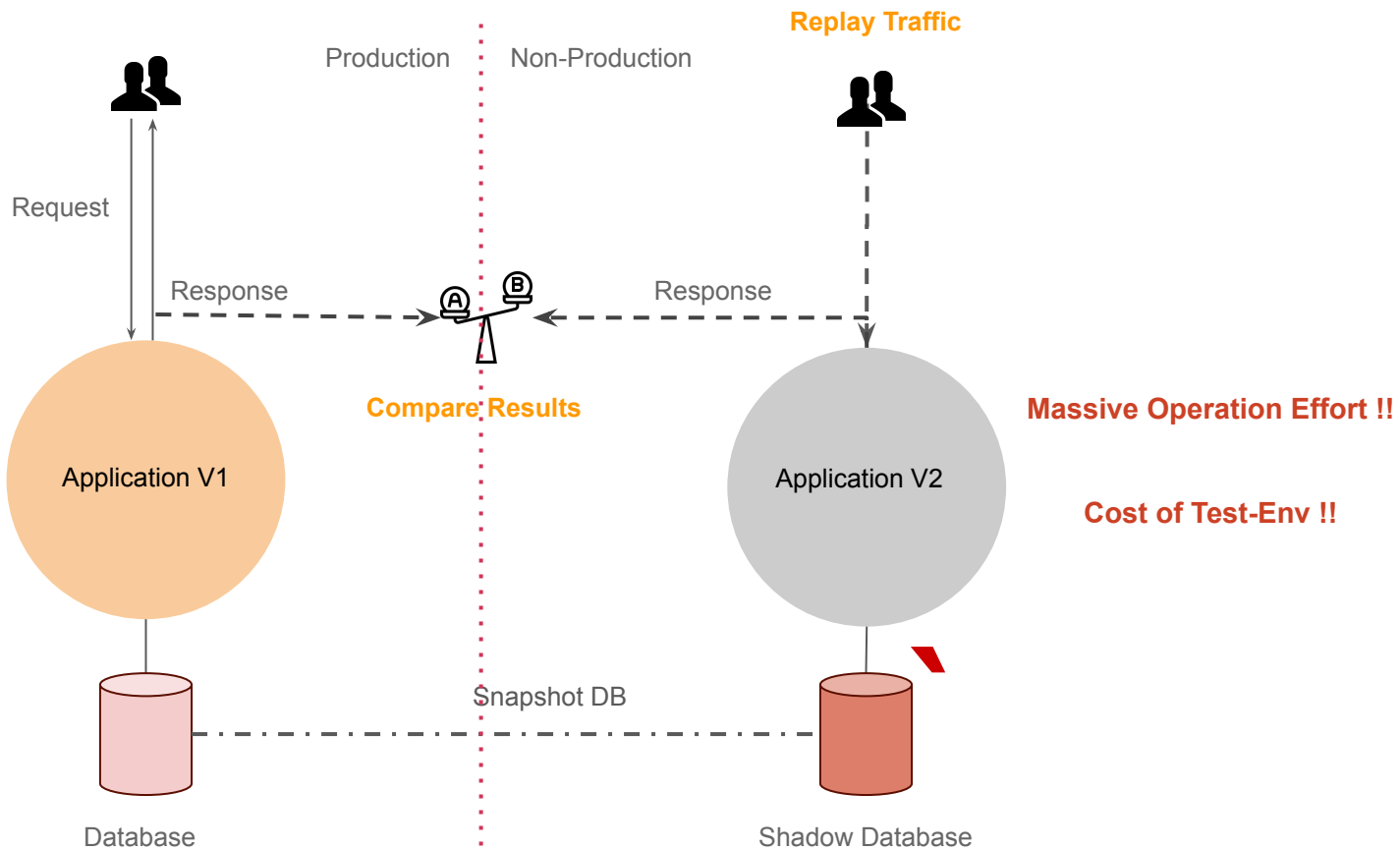
- Need real-world functional tests
- Easy to create/update tests
- Automatic Infrastructure orchestration



Let's look at...



Record-Replay



So far...

Pros

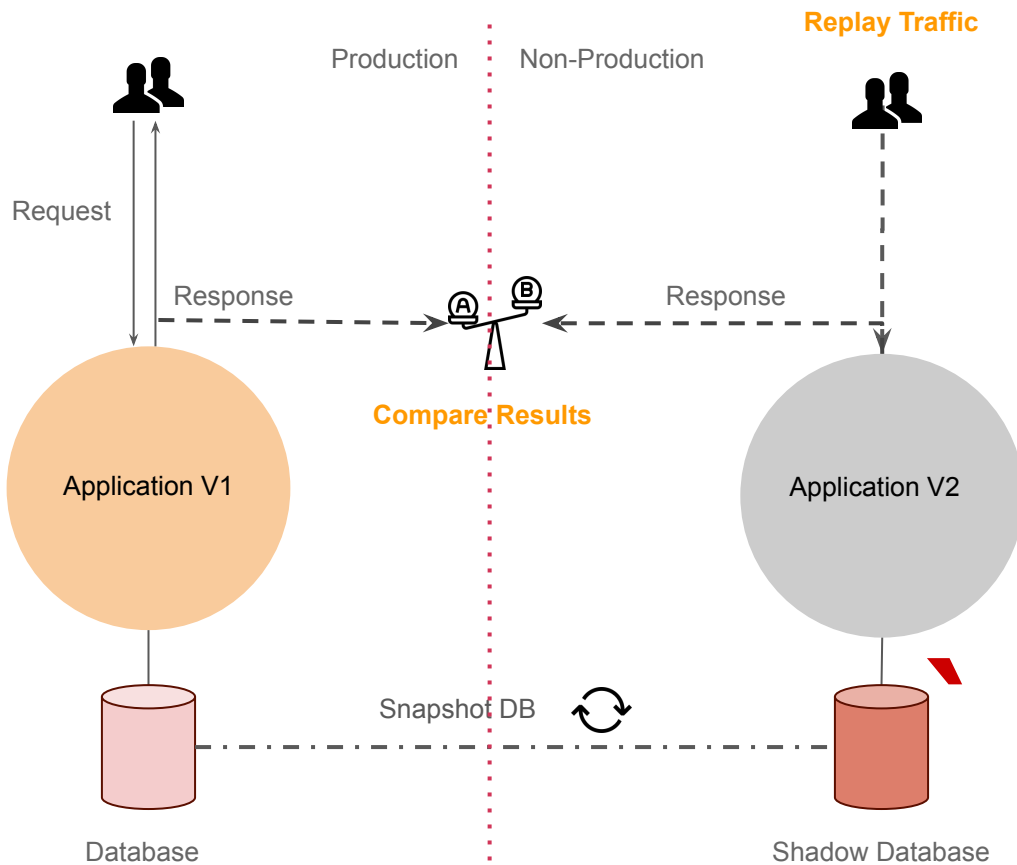
- Low code approach
- Easy to achieve high coverage
- Unlocked unexpected user flows

Cons

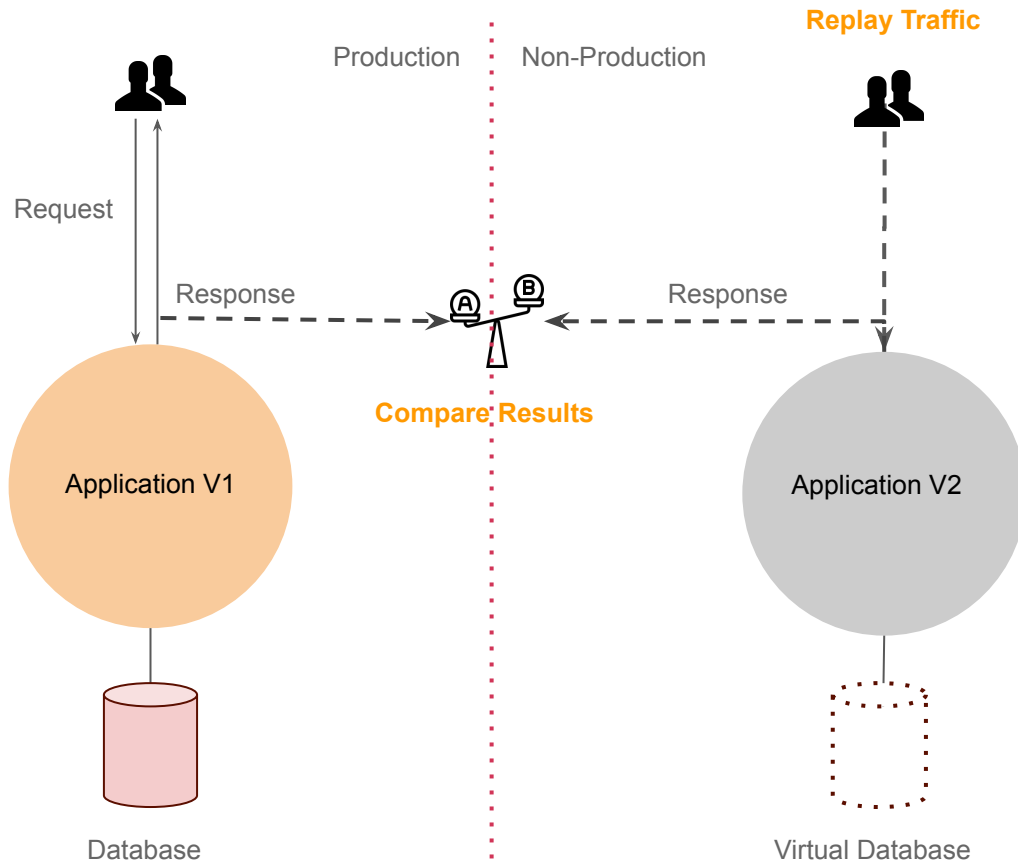
- Dependency state (DBs, internal or external APIs) causes flakiness.
- Suitable for load/stress testing but not functional tests
- Handling writes can be tricky



Record-Replay



Record-Replay



**Add Support for every
Dependency !!**

Brittle Dependencies !!

Example

Prod Env



Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }

Person (Table A)

Harry
Luna Bells
Thomson
Prince
Andrew
Archer
Kagiso

Games (Table B)

Cricket
Football
Tennis
Volleyball
Carrom
Bowling
Boxing
Basketball



mongoDB

Test Env



Request

getGames (user = Thomson)

Response

{ user Thomson not found }

Person (Table A)

Andrew
Archer
Kagiso
Quit-In

Games (Table B)

Football
Tennis



mongoDB

Example

Prod Env



Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }

Person (Table A)

Harry
Luna Bells
Thomson
Prince
Andrew
Archer
Kagiso

Games (Table B)

Cricket
Football
Tennis
Volleyball
Carrom
Bowling
Boxing
Basketball



mongoDB

Test Env



Request

getGames (user = Thomson)

Response

{ user Thomson not found }

Person (Table A)

Andrew
Archer
Kagiso
Quit-In

Games (Table B)

Football
Tennis



mongoDB

Dependency Mocking

Prod Env

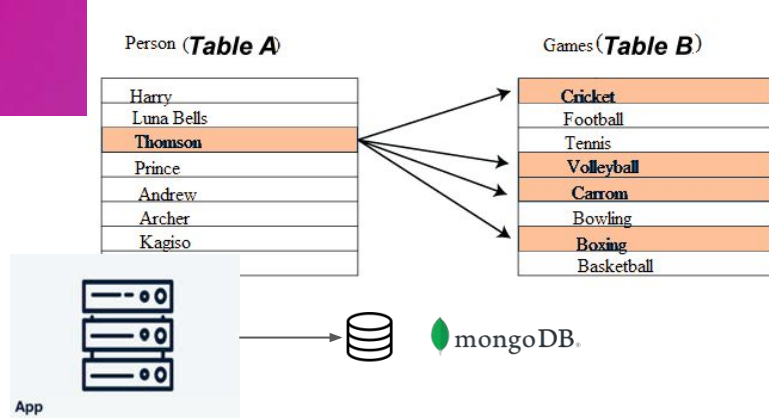


Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }



VIRTUAL Test Env

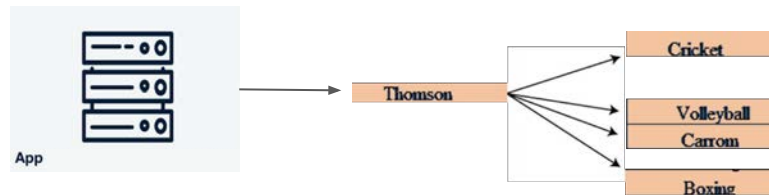


Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }



Dependency Mocking

Prod Env

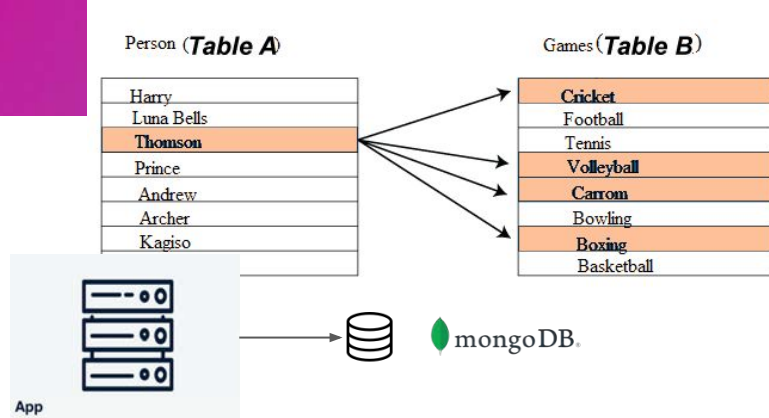


Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }



VIRTUAL Test Env

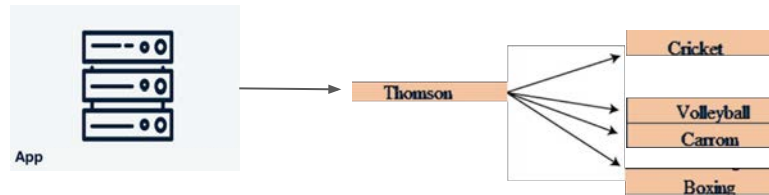


Request

getGames (user = Thomson)

Response

{ Cricket, Volleyball, Carrom, Boxing }



- Converts **API Calls** to **Test Cases**.
- Automatically **mock** external **dependencies**.
- Accurate **Noise** Detection.



Demo



Welcome to Keploy 🙌



Contributions Welcome Go passing go report slack docs Gitpod ready-to-code

Keploy

Keploy is a functional testing toolkit for developers. It generates E2E tests for APIs (KTests) along with mocks or stubs(KMocks) by recording real API calls. KTests can be imported as mocks for consumers and vice-versa.



Start keploy server (macOS)

```
curl --silent --location "https://github.com/keploy/keploy/releases/latest/download/keploy-darwin.tar.gz" && tar -xzf keploy-darwin.tar.gz && mv /tmp/keploy /usr/local/bin && keploy
```

Start keploy server (Linux)

```
curl --silent --location "https://github.com/keploy/keploy/releases/latest/download/keploy-linux.tar.gz" && tar -xzf keploy-linux.tar.gz && mv /tmp/keploy /usr/local/bin && keploy
```

Setup Employee-Manager App

```
git clone https://github.com/keploy/samples-java
```

Start PostgreSQL instance

```
docker-compose up -d
```


Maven clean install

```
mvn clean install
```

Run the application

```
mvn spring-boot:run
```




Find Keploy on Github 



Join us on Slack 



Thank you!

