



How to test cloud dependencies with Localstack

# About me

---

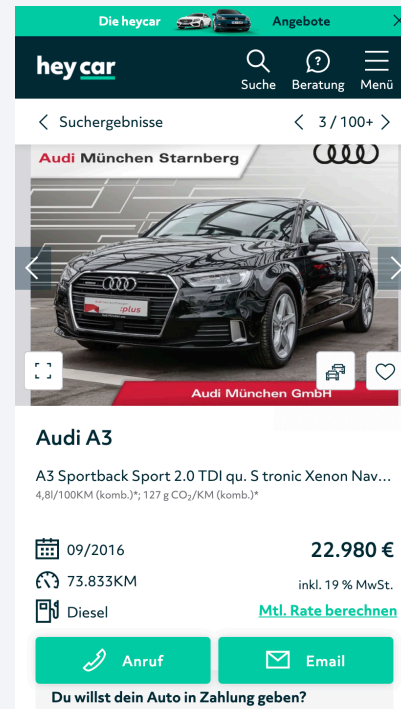
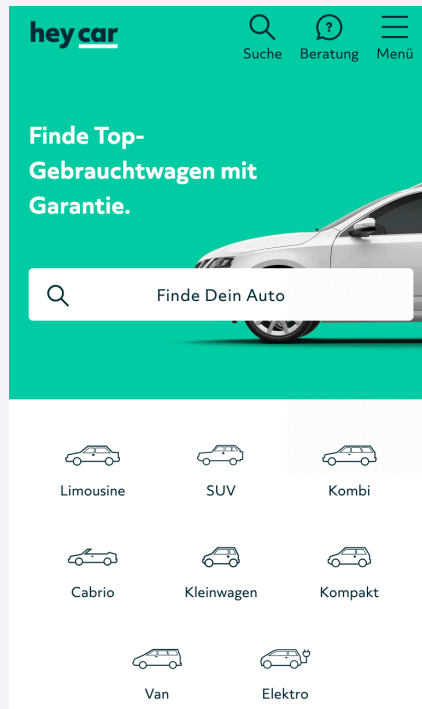
- <http://github.com/eduardompinto>
- <http://twitter.com/eduardompinto>
- Telegram: @eduardompinto
- Kotlin slack: @Eduardo Pinto
- Brazilian
- kotlin, go, python
- First talk in a foreign land/language



# About hey car

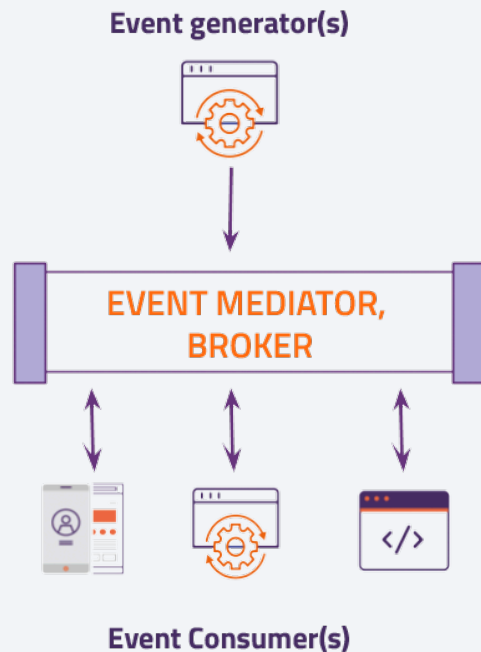
## Our engineering context

- <http://hey.car>
- <http://github.com/hey-car>
- +40 microservices
- +50 people in engineering
- +15 nationalities
- Java + Spring -> Kotlin + Spring
- Event-driven architecture
- AWS



# Event-driven

- We didn't miss a change
- Changes are streamed
  - Consumed by how wants to know about that
- Kinesis
- On the right corner is a generic image 😊



Source: <https://sensedia.com/wp-content/uploads/2019/07/Event-Driven-Components.png>

# Localstack

"LocalStack - A fully functional local AWS cloud stack"

- <https://github.com/localstack/localstack>
- SQS, Dynamodb, Kinesis, SQS ...
- Easy to setup
- Compatible with multiple languages

★ Star

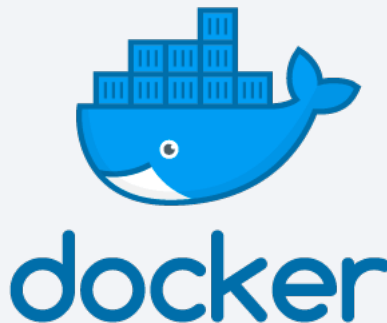
22.5k

# Setup

How to run it?

- Needs docker and docker-compose

```
version: '3.2'
services:
  localstack:
    image: localstack/localstack:latest
    ports:
      - '4560-4576:4560-4576'
      - '8082:8080'
    environment:
      - SERVICES=kinesis,sqs,dynamodb
    volumes:
      - './.localstack:/tmp/localstack/data'
      - '/var/run/docker.sock:/var/run/docker.sock'
```



# Creating resources

---

- Same syntax of using aws-cli on your console
- <https://aws.amazon.com/cli/>

```
aws-cli:  
  image: mesosphere/aws-cli:1.14.5  
  volumes:  
    - './scripts:/project/scripts'  
  environment:  
    - AWS_ACCESS_KEY_ID=foo  
    - AWS_SECRET_ACCESS_KEY=bar  
    - AWS_DEFAULT_REGION=eu-central-1  
  entrypoint: '/project/scripts/setup.sh'
```



# Creating resources

How to run it?

- Content of the setup script

```
echo "Waiting for localstack to be ready"  
# There's better ways to do this :)  
sleep 20  
aws sqs create-queue --endpoint-url=http://localstack:4576 --queue-name test-queue  
aws kinesis create-stream --endpoint-url=http://localstack:4568 --shard-count 1 --stream-name test-stream
```



# Kotlin

After all this is a kotlin meetup

---

- Localstack provides better integration tests
- [Demo](#)

```
✓ Test SQS API
▼ ✓ Given a valid queue
  ✓ when purge it should receive a 2XX response
  ✓ when delete a not existing message it should throw an AmazonSQSException
  ✓ when sends a message it should receive a messageId and an md5 of the message body
  ✓ when request 5 messages should bring 5 messages
  ✓ when delete a valid message it should receive a 2XX response
▼ ✓ Given a not existing queue
  ✓ when tries to purge it should throw a QueueDoesNotExistException
  ✓ when tries to send a message it should throw a QueueDoesNotExistException
  ✓ when tries to get messages it should throw a QueueDoesNotExistException
  ✓ when tries to delete a message it should throw a QueueDoesNotExistException
```

**hey car**

**We are hiring!**