zenvia

THE DEVELOPER'S CONFERENCE

"Micronaut"

Trilha – Arquitetura Java

Marcelo Adamatti

https://adamatti.github.io

zenvia

MICRONAUT

https://adamatti.github.io

spring

https://adamatti.github.io

Microservices

https://adamatti.github.io

2003

https://adamatti.github.io

2005/2006

https://adamatti.github.io

zenvia

https://adamatti.github.io

Why Micronaut?

- Designed for microservices and serveless
- Fast Startup time
- Low memory
- “Small jar”
- Zero Dependencies
- 12 factors
- Ahead of time (AOT) compilation (e.g. CDI)

Jar Sizes

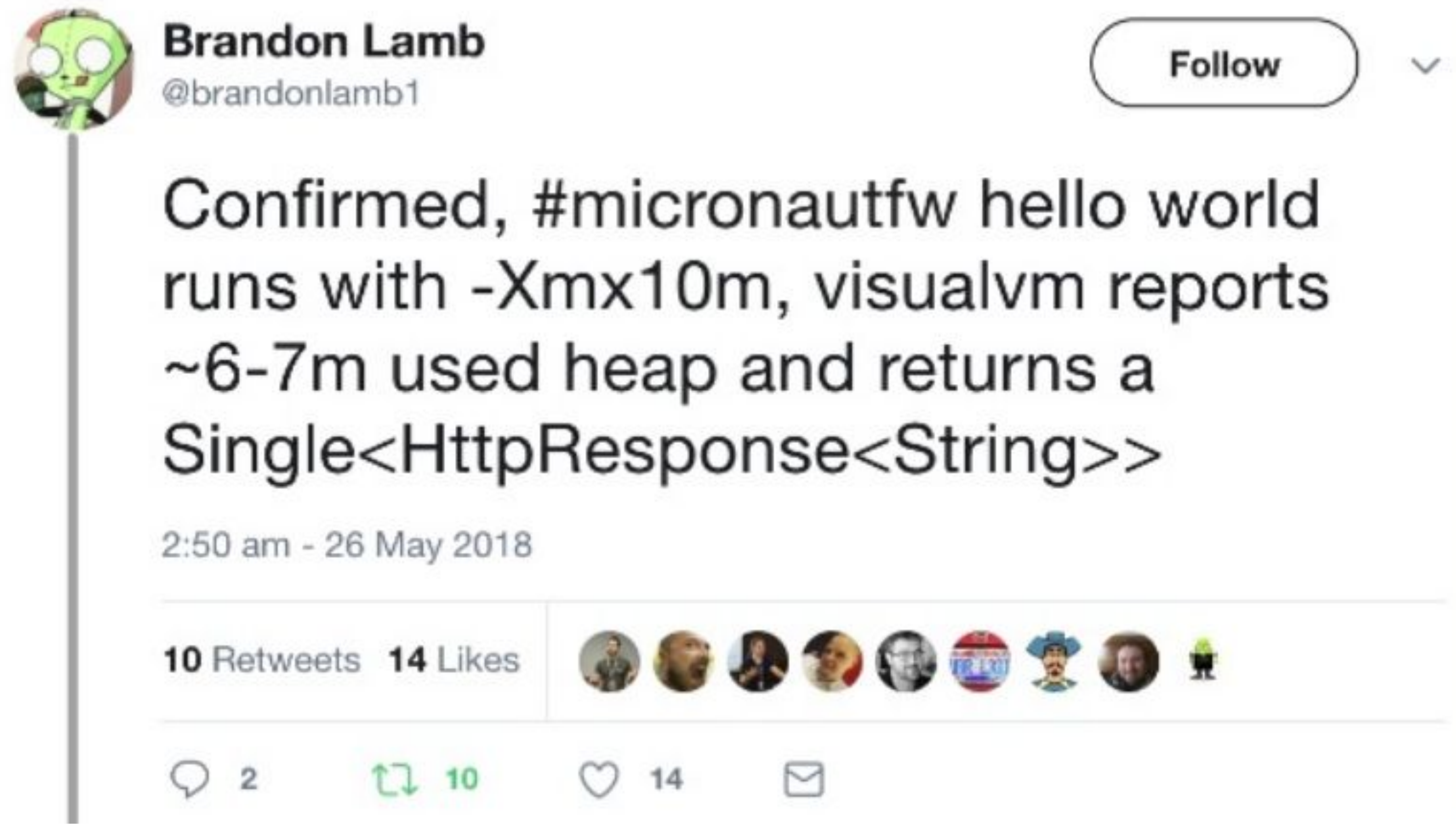
- Java: 8 mb
- Groovy: 12 mb
- Spring + Groovy: 36 mb
- Grails: 27 mb

Heap Size

- Java: 7 mb
- Groovy: 19 mb
- Spring + Groovy: 33 mb
- Grails: 49 mb

Startup Time

- ~1 second
- Spring / Grails: ~3-4 seconds



Brandon Lamb
@brandonlamb1

Follow

Confirmed, #micronautfw hello world runs with -Xmx10m, visualvm reports ~6-7m used heap and returns a Single<HttpResponse<String>>

2:50 am - 26 May 2018

10 Retweets 14 Likes

2 10 14

https://adamatti.github.io

GraalVM™

14 ms

https://adamatti.github.io

https://adamatti.github.io

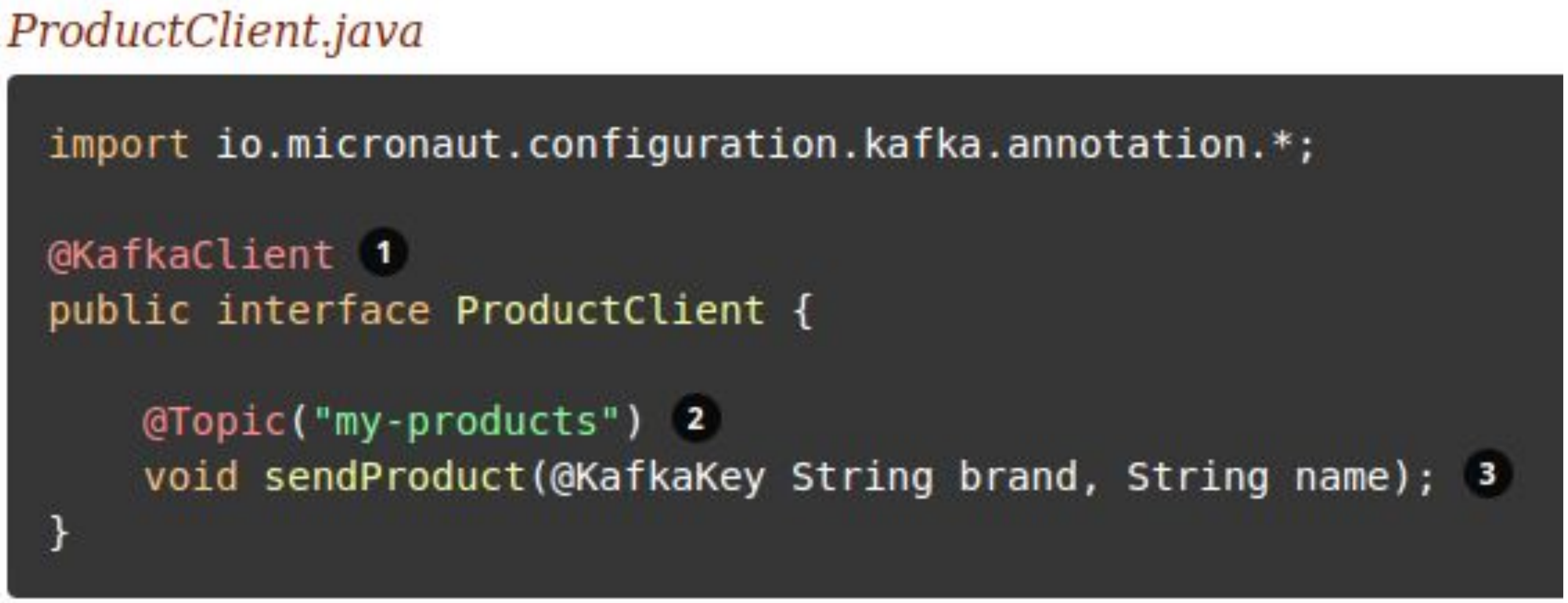
https://adamatti.github.io

References

- http://micronaut.io
- Micronaut Announcement: https://www.youtube.com/watch?v=56j_f3OCg6E
- https://www.technipelago.se/blog/show/The-road-to-Micronaut
- AWS Lambda with Micronaut and without any framework - billing in serverless architecture https://github.com/asc-lab/aws-lambda-billing
- https://www.slideshare.net/alvarosanchezmariscal/conoce-micronaut-un-framework-para-microservicios-jvm-commit-conf-2018

https://adamatti.github.io

Backup
Ctrl



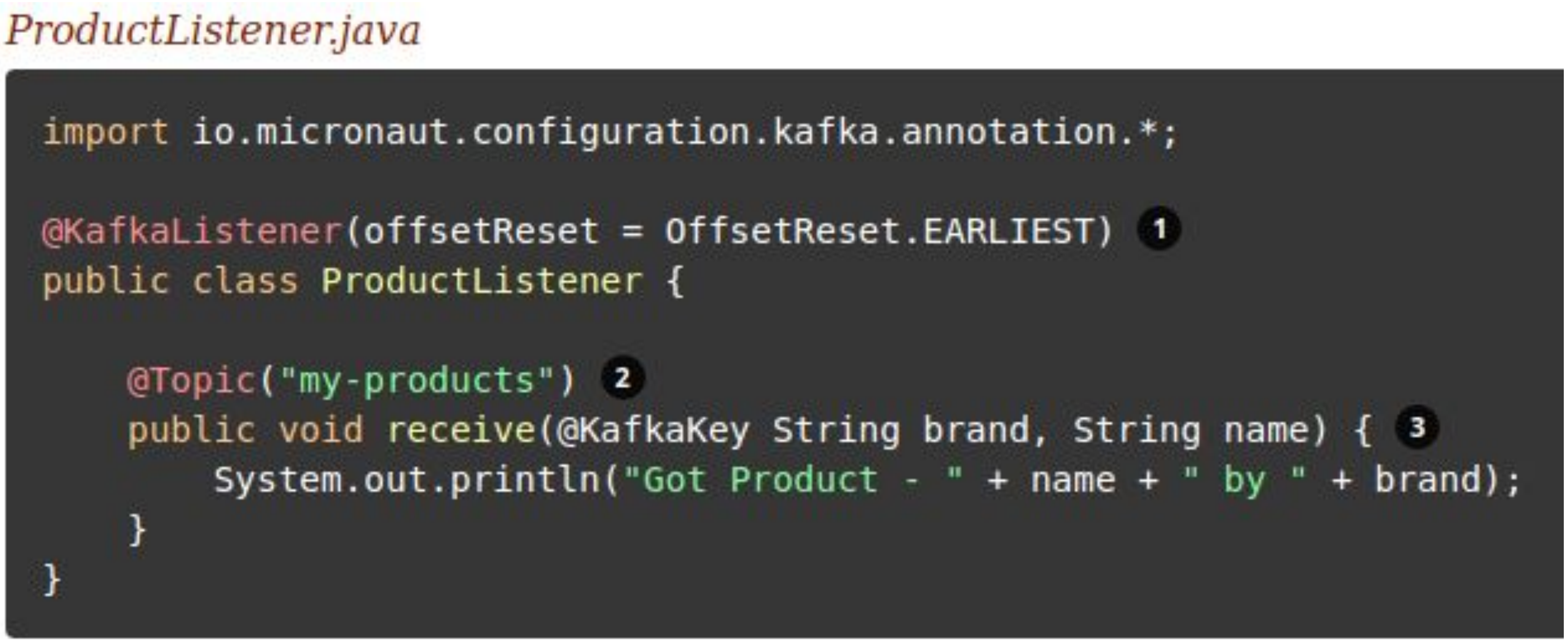
ProductClient.java

```
import io.micronaut.configuration.kafka.annotation.*;

@KafkaClient 1
public interface ProductClient {

    @Topic("my-products") 2
    void sendProduct(@KafkaKey String brand, String name); 3
}
```

https://adamatti.github.io



ProductListener.java

```java
import io.micronaut.configuration.kafka.annotation.*;

@KafkaListener(offsetReset = OffsetReset.EARLIEST) ❶
public class ProductListener {

    @Topic("my-products") ❷
    public void receive(@KafkaKey String brand, String name) { ❸
        System.out.println("Got Product - " + name + " by " + brand);
    }
}
```

https://adamatti.github.io