O estado atual do Kotlin no backend

Víctor Orozco - Nabenik

30 de Novembro de 2021

@tuxtor

ACADEMIK

¿Java?

- Linguagem (Java 16)
- OpenJDK (Java Virtual Machine)
- Bibliotecas/API (Java Classpath)

As três juntas são a plataforma Java

ACADEMIK

Java no Android

- Linguagem (Java 7 + Lambdas + DateTime)
- ART/Dalvik
- Bibliotecas/API (Java+Google Classpath)

ACADEMIK

Kotlin

- Linguagem (Kotlin)
- OpenJDK (Java Virtual Machine)
- Bibliotecas/API (Java Classpath)
- kotlin-stdlib

ACADEMIK

Modelos de desenvolvimento

Desenvolvimento backend Java no 2021 - Concorrência

Thread based -e.g. EJB, CDI, JAX-RS-

```
@Inject
SayService say;

@GET
@Produces(MediaType.TEXT_PLAIN)
public String hello() {
    return say.hello();
}
```

JakartaEE/MicroProfile

Desenvolvimento backend Java no 2021 - Concorrência

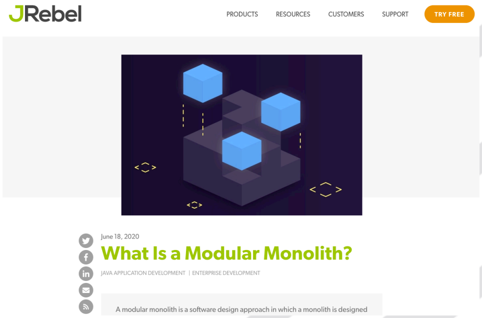
Reactive based -e.g. Async JAX-RS, Spring WebFlux, Actors, Verticles-

```java
public class Server extends AbstractVerticle {
  public void start() {
    vertx.createHttpServer().requestHandler(req -> {
      req.response()
        .putHeader("content-type", "text/plain")
        .end("Hello from Vert.x!");
    }).listen(8080);
  }
}
```

Vert.x

Desenvolvimento backend Java no 2021 - Arquitetura

- Monólito
- Microserviço
- Micromonolito



June 18, 2020

What Is a Modular Monolith?

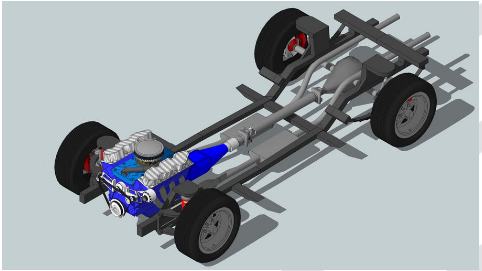
JAVA APPLICATION DEVELOPMENT | ENTERPRISE DEVELOPMENT

A modular monolith is a software design approach in which a monolith is designed

Desenvolvimento backend Java no 2021 - Runtime

"Simples/DIY"

- Servlet based -e.g. Tomcat, Jetty-
- Custom I/O -e.g. Undertow, Netty, Vert.x-


(CC BY-NC-SA3.0 GT)

ACADEMIK

Desenvolvimento backend Java no 2021 - Runtime/Framework

"Micro"

- Custom micro -e.g. Jooby, Spark, Javalin, Helidon SE-
- MicroProfile based -e.g. Quarkus, Helidon-
- Micronaut
- Spring Boot

ACADEMIK

Desenvolvimento backend Java no 2021

"Complexo"

- Java EE/Jakarta EE -e.g. JBoss, WebSphere Liberty-
- Spring
- Akka

ACADEMIK

Arquiteto Java no ano 2021

Arquiteto Java

O cara que tem que decidir entre pegar um framework pronto *complexo* ou pegar um runtime *ligero* e criar a estrutura toda -i.e Bibliotecas, estilo arquitetural, SCM (Maven)-.

O cara que tem que decidir entre chatear os dev Java tradicionais no modelo de desenvolvimento Async ou chatear os desenvolvedores Kotlin por não aproveitar os Coroutines que nem no Android.

O cara que tem que avaliar dar o pulo para o Kotlin conservando os stacks tradicionais.

Arquiteto Java no ano 2021

Arquiteto Java

O cara que vai ser odiado por não dar o pulo para JavaScript só pela vontade de re-escreber a base de código que vem fazendo succeso há 15 anos

ACADEMIK

Kotlin no backend

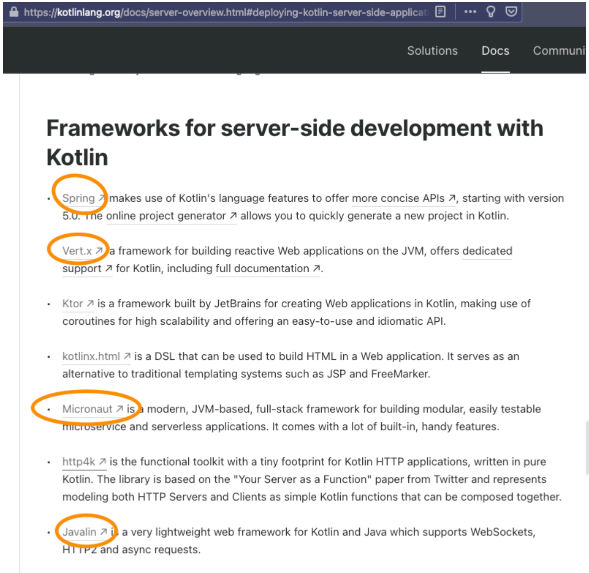
Kotlin no backend - Fatos

Fato #1

Todo framework Java pode virar framework Kotlin. Mas nem todo framework Java e testado no Kotlin.

ACADEMIK

Kotlin no backend - Fatos



https://kotlinlang.org/docs/server-overview.html#deploying-kotlin-server-side-applicat

Frameworks for server-side development with Kotlin

- Spring ↗ makes use of Kotlin's language features to offer more concise APIs ↗, starting with version 5.0. The online project generator ↗ allows you to quickly generate a new project in Kotlin.
- Vert.x ↗ a framework for building reactive Web applications on the JVM, offers dedicated support ↗ for Kotlin, including full documentation ↗.
- Ktor ↗ is a framework built by JetBrains for creating Web applications in Kotlin, making use of coroutines for high scalability and offering an easy-to-use and idiomatic API.
- kotlinx.html ↗ is a DSL that can be used to build HTML in a Web application. It serves as an alternative to traditional templating systems such as JSP and FreeMarker.
- Micronaut ↗ is a modern, JVM-based, full-stack framework for building modular, easily testable microservice and serverless applications. It comes with a lot of built-in, handy features.
- http4k ↗ is the functional toolkit with a tiny footprint for Kotlin HTTP applications, written in pure Kotlin. The library is based on the "Your Server as a Function" paper from Twitter and represents modeling both HTTP Servers and Clients as simple Kotlin functions that can be composed together.
- Javalin ↗ is a very lightweight web framework for Kotlin and Java which supports WebSockets, HTTP2 and async requests.

ACADEMIK

Kotlin no backend - Fatos

Frameworks com suporte oficial

-

Kotlin no backend - Fatos

Fato #2

Geralmente o problema são os annotation processors -e.g. Lombok-, muitos são feitos para o Java

Kotlin no backend - Fatos

Fato #3

O ecosistema Kotlin esta criando Frameworks Kotlin-first/Kotlin-exclusive

Kotlin no backend - Fatos

Reactive based - DIY

```
fun HelloWorld(): HttpHandler {
    return routes("/" bind GET to { Response(OK).body("hello world!") })
}
```

Http4k

Kotlin no backend - Fatos

Reactive based - Micro

```
fun main(args: Array<String>) {
    embeddedServer(Netty, 8080) {
        routing {
            get("/") {
                call.respondText("Hello, world!", ContentType.Text.Html)
            }
        }
    }.start(wait = true)
}
```

Ktor

Java - Dying since 1995

- Spring Boot, Micronaut, MicroProfile, GraalVM . . .
- Raw performance (Beam, Spark, Hadoop)
- Tooling - IDE, Maven, Drivers RDBMS
- JVM - (Twitter, Alibaba, Spotify, etc.)
- OpenJDK

Kotlin

Vantagens

- Código conciso enquanto você conheça as convenções
- Java inter-op
- Backend para desenvolvedores Android
- Uma nova forma de virar *fullStack*

Desvantagens

- IntelliJ IDEA Ultimate
- Precisa tempo para aprender e virar produtivo
- Tempo de compilação
- Thread-managed vs Co-routines
- Amber, Loom, Valhalla, Panama (Java 18?)

Projeto tradicional com Kotlin

Projeto Java so que não

1. Maven
2. Dependencias (MicroProfile, Jakarta EE, Arquillian, JUnit, . . .)
3. Maven plugin (maven-compiler-plugin)
4. Kotlin plugin (kotlin-maven-plugin)

Eclipse MicroProfile com Payara 5

```
<dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <type>pom</type>
        <version>3.2</version>
        <scope>provided</scope>
</dependency>
```

Kotlin with Maven - Dependency

```xml
<dependency>
        <groupId>org.jetbrains.kotlin</groupId>
        <artifactId>kotlin-stdlib-jdk8</artifactId>
        <version>${kotlin.version}</version>
</dependency>
```

Kotlin with Maven - maven-compiler-plugin

```
<execution>
        <id>default-compile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>default-testCompile</id>
        <phase>none</phase>
</execution>
<execution>
        <id>java-compile</id>
        <phase>compile</phase>
        <goals> <goal>compile</goal> </goals>
</execution>
<execution>
        <id>java-test-compile</id>
        <phase>test-compile</phase>
        <goals> <goal>testCompile</goal> </goals>
</execution>
```

Kotlin with Maven - kotlin-maven-plugin

```
<compilerPlugins>
<plugin>all-open</plugin>
</compilerPlugins>
...
<option>all-open:annotation=javax.ws.rs.Path</option>
<option>all-open:annotation=javax.enterprise.context.RequestScoped</option>
<option>all-open:annotation=javax.enterprise.context.SessionScoped</option>
<option>all-open:annotation=javax.enterprise.context.ApplicationScoped</option>
<option>all-open:annotation=javax.enterprise.context.Dependent</option>
<option>all-open:annotation=javax.ejb.Singleton</option>
<option>all-open:annotation=javax.ejb.Stateful</option>
<option>all-open:annotation=javax.ejb.Stateless</option>
```

Ideia geral: As anotações Java viram open classes por causa do proxy-classes metapadrão

ACADEMIK

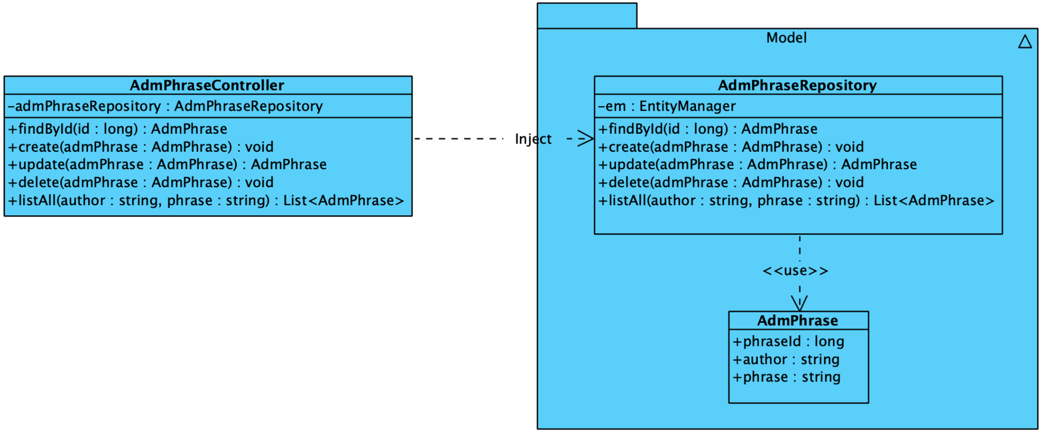
Kotlin + Jakarta EE + MicroProfile - Demo

- Kotlin 1.3
- Libraries - SLF4J, Flyway, PostgreSQL
- Jakarta EE 8 - EJB, JPA
- MicroProfile - CDI, JAX-RS, MicroProfile Config
- Testing - Arquillian, JUnit, Payara Embedded

`https://dzone.com/articles/`
`the-state-of-kotlin-for-jakarta-eemicroprofile-tra`
`https://github.com/tuxtor/integrum-ee`

ACADEMIK

Kotlin + Jakarta EE + MicroProfile - Demo


AdmPhraseController
-admPhraseRepository : AdmPhraseRepository
+findById(id : long) : AdmPhrase
+create(admPhrase : AdmPhrase) : void
+update(admPhrase : AdmPhrase) : AdmPhrase
+delete(admPhrase : AdmPhrase) : void
+listAll(author : string, phrase : string) : List<AdmPhrase>
Inject
Model
AdmPhraseRepository
-em : EntityManager
+findById(id : long) : AdmPhrase
+create(admPhrase : AdmPhrase) : void
+update(admPhrase : AdmPhrase) : AdmPhrase
+delete(admPhrase : AdmPhrase) : void
+listAll(author : string, phrase : string) : List<AdmPhrase>
<<use>>
AdmPhrase
+phraseId : long
+author : string
+phrase : string

ACADEMIK

Kotlin - JPA entity

```kotlin
@Entity
@Table(name = "adm_phrase")
@TableGenerator(...)
data class AdmPhrase(
        @Id
        @GeneratedValue(strategy = GenerationType.TABLE,
                generator = "admPhraseIdGenerator")
        @Column(name = "phrase_id")
        var phraseId:Long? = null,
        var author:String = "",
        var phrase:String = ""
)
```

Data Clases, Nullable Types

ACADEMIK

Kotlin - CDI Repository

```
@RequestScoped
class AdmPhraseRepository {

        @Inject
        private lateinit var em:EntityManager

        ...

}
```

Lateinit (nullable type)

Kotlin - CDI Repository

```
fun create(admPhrase:AdmPhrase) = em.persist(admPhrase)

fun update(admPhrase:AdmPhrase) = em.merge(admPhrase)

fun findById(phraseId: Long) =
em.find(AdmPhrase::class.java, phraseId)

fun delete(admPhrase: AdmPhrase) = em.remove(admPhrase)
. . .
```

Single expression functions (One line methods)

ACADEMIK

Kotlin - CDI Repository

```
fun listAll(author: String, phrase: String):
        List<AdmPhrase> {

        val query = """SELECT p FROM AdmPhrase p
        where p.author LIKE :author
        and p.phrase LIKE :phrase
        """

        return em.createQuery(query, AdmPhrase::class.java)
                .setParameter("author", "%$author%")
                .setParameter("phrase", "%$phrase%")
                .resultList
}
```

Multiline string

ACADEMIK

Kotlin - JAX-RS Controllers

```
@Path("/phrases")
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
class AdmPhraseController{

        @Inject
        private lateinit var admPhraseRepository: AdmPhraseRepository

        @Inject
        private lateinit var logger: Logger
        ...

}
```

Kotlin - JAX-RS Controller

```
@GET
fun findAll(
@QueryParam("author") @DefaultValue("%") author: String,
@QueryParam("phrase") @DefaultValue("%") phrase: String) =
        admPhraseRepository.listAll(author, phrase)

@GET
@Path("/{id:[0-9][0-9]*}")
fun findById(@PathParam("id") id:Long) =
        admPhraseRepository.findById(id)

@PUT
fun create(phrase: AdmPhrase): Response {
        admPhraseRepository.create(phrase)
        return Response.ok().build()
}
```

ACADEMIK

Kotlin - JAX-RS Controller

Elvis operator as expression

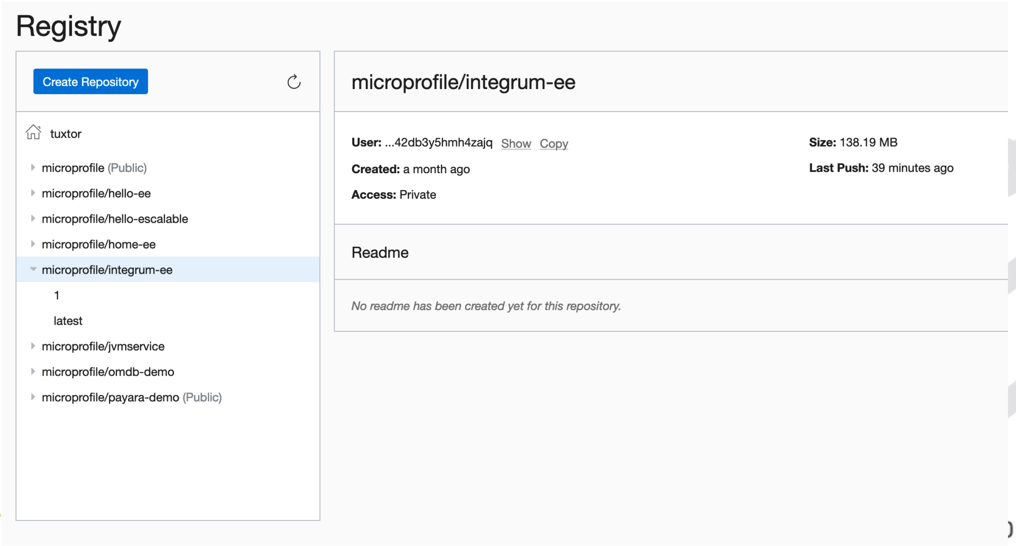
```
@POST
@Path("/{id:[0-9][0-9]*}")
fun update(@PathParam("id") id: Long?, phrase: AdmPhrase)
        :Response {
        if(id != phrase.phraseId)
                return Response.status(Response.Status.NOT_FOUND).build()

        val updatedEntity = admPhraseRepository.update(phrase)
        return Response.ok(updatedEntity).build()
}
```

Oracle Cloud

```
<groupId>io.fabric8</groupId>
<artifactId>docker-maven-plugin</artifactId>
<version>0.30.0</version>
...
<image>
        <name>iad.ocir.io/tuxtor/microprofile/integrum-ee</name>
        <build>
                <dockerFile>${project.basedir}/Dockerfile</dockerFile >
        </build>
</image>
```

ACADEMIK

Oracle Cloud



Registry

tuxtor

- microprofile (Public)
- microprofile/hello-ee
- microprofile/hello-escalable
- microprofile/home-ee
- microprofile/integrum-ee
  - 1
  - latest
- microprofile/jvmservice
- microprofile/omdb-demo
- microprofile/payara-demo (Public)

microprofile/integrum-ee

User: ...42db3y5hmh4zajq Show Copy

Created: a month ago

Access: Private

Size: 138.19 MB

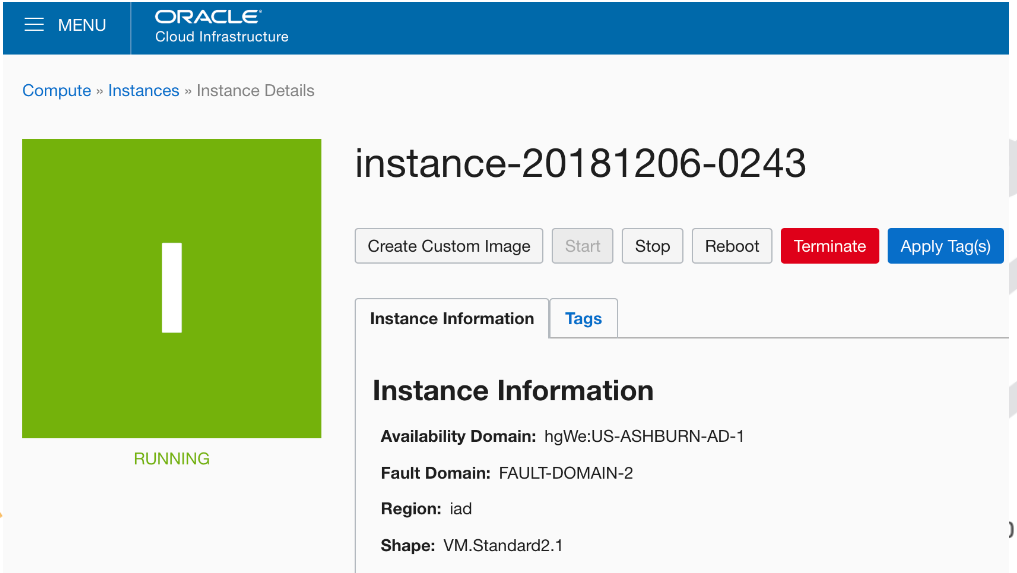
Last Push: 39 minutes ago

Readme

No readme has been created yet for this repository.

) GT)

Oracle Cloud



instance-20181206-0243

Instance Information

Availability Domain: hgWe:US-ASHBURN-AD-1

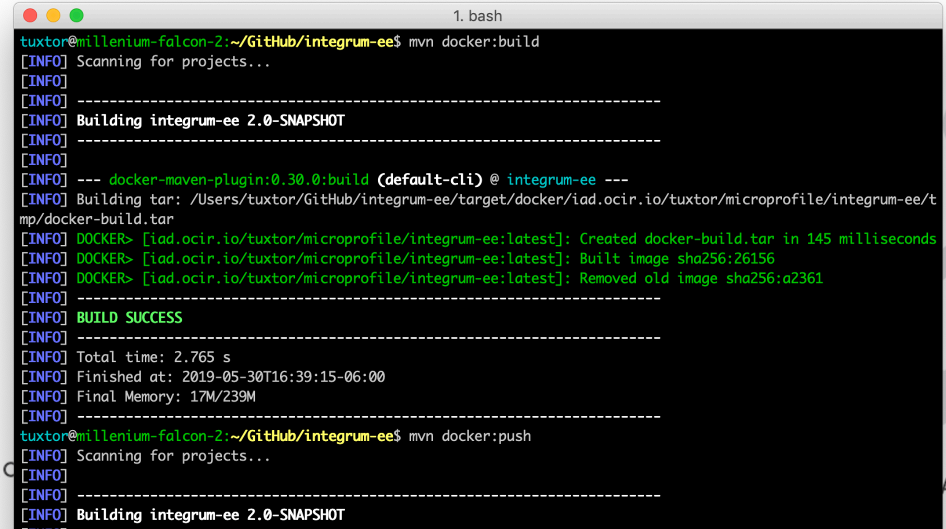
Fault Domain: FAULT-DOMAIN-2

Region: iad

Shape: VM.Standard2.1

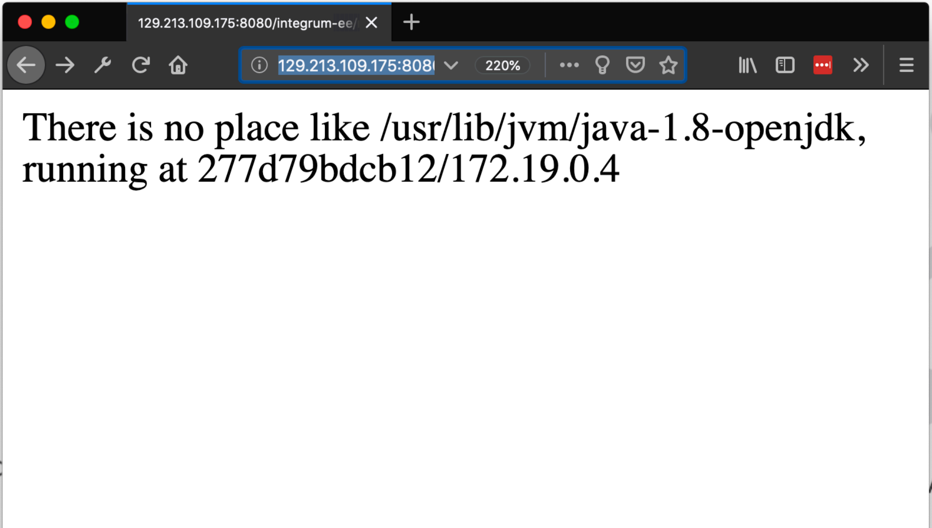
) GT)

Oracle Cloud



```
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:build
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- docker-maven-plugin:0.30.0:build (default-cli) @ integrum-ee ---
[INFO] Building tar: /Users/tuxtor/GitHub/integrum-ee/target/docker/iad.ocir.io/tuxtor/microprofile/integrum-ee/t
mp/docker-build.tar
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Created docker-build.tar in 145 milliseconds
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Built image sha256:26156
[INFO] DOCKER> [iad.ocir.io/tuxtor/microprofile/integrum-ee:latest]: Removed old image sha256:a2361
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.765 s
[INFO] Finished at: 2019-05-30T16:39:15-06:00
[INFO] Final Memory: 17M/239M
[INFO] ------------------------------------------------------------------------
tuxtor@millenium-falcon-2:~/GitHub/integrum-ee$ mvn docker:push
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building integrum-ee 2.0-SNAPSHOT
```

Oracle Cloud



129.213.109.175:8080/integrum-ee/

129.213.109.175:808

220%

```
There is no place like /usr/lib/jvm/java-1.8-openjdk,
running at 277d79bdcb12/172.19.0.4
```

Kotlin

Kotlin

- Static typing
- Java inter-op
- OO + FP
- Null safety
- Extension functions
- Operator overloading
- Data classes
- One line methods

Kotlin - Fatos interesantes

- Effective Java - Immutability, builder, singleton, override, final by default, variance by generics
- Elvis - Groovy
- Type inference - Scala
- Immutability - Scala
- Identifiers - Scala
- Null values management - Groovy
- Functions - Groovy

ACADEMIK

Víctor Orozco

Duke's Choice Award
2016 Winner
Java

Oracle Groundbreakers

Java Champions

ORACLE
Certified Professional
Java SE 8 Programmer

ORACLE
Certified Associate
Java SE 8 Programmer

ACADEMIK

- vorozco@nabenik.com
- @tuxtor
- https://vorozco.com