

# MICRONAUT

Aleck BILOUNGA



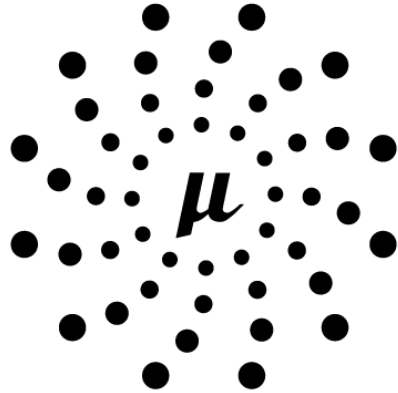


# Plan

- I. PRÉSENTATION
- II. FONCTIONNEMENT
- III. FONCTIONNALITÉS
- IV. CAS D'UTILISATIONS
- V. AVANTAGES ET INCONVÉNIENTS



# Présentation



M I C R O N A U T



OBJECT  
COMPUTING



# FONCTIONNEMENT

## Injection de dépendance et inversion de contrôle (IoC) :

- Repose sur un ensemble de transformations *AST* (pour Groovy) et de processeurs d'annotation (pour Java) qui génèrent des classes qui implémentent l'interface *BeanDefinition*.
- Permet de configurer votre application pour qu'elle fonctionne dans un environnement à faible mémoire.
- Le temps de démarrage de votre application et la consommation de mémoire ne sont pas liés à la taille de votre base de code,



# FONCTIONNALITÉS

- Injection de dépendance et inversion de contrôle(IOC):
- Aspect Oriented Programming(AOP):
- Sensible Defaults and Auto-Configuration: Fast Access Configuration
- Micronaut propose un modèle asynchrone et non bloquant et le rend donc indiqué pour développer des applications réactives. Pour cela la couche réseau est basée sur le serveur Netty qui apporte la gestion de l'évent loop .
- Configuration d'accès des données avec les bases de données mongoDB, Neo4j, Postgres, Cassandra, Redis...

Résilience et tolérance aux fautes

- Mécaniques intégrées de retry et circuit breaker

# FONCTIONNALITÉS

## Cloud

- Configuration distribuée (HashiCorp Consul, AWS Parameter Store etc)
- Service Discovery (Consul, Eureka, Kubernetes)
- Load Balancing côté client (Netflix Ribbon)
- Distributed Tracing
- fonctions Serverless

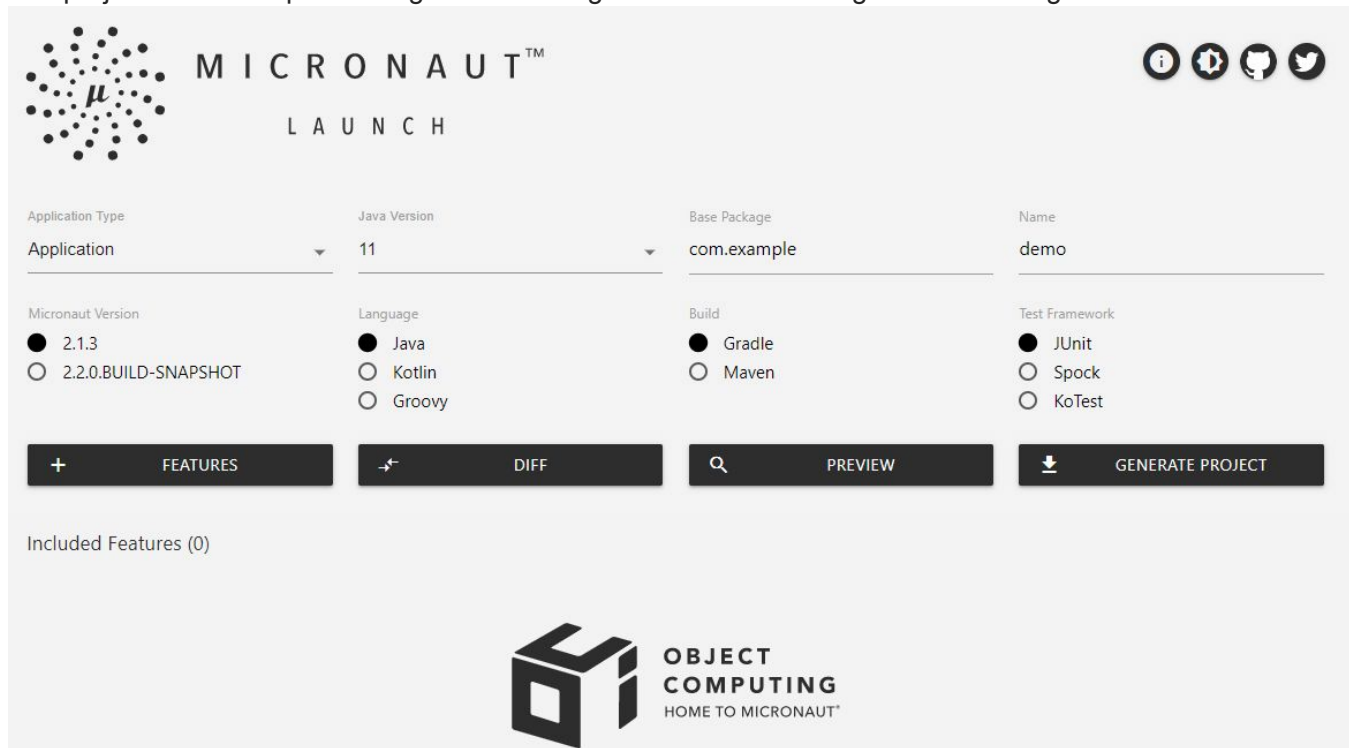
NATIVELY CLOUD NATIVE

Micronaut's cloud support is built right in, including support for common discovery services, distributed tracing tools, and cloud runtimes.



# FONCTIONNALITÉS

Les projets Micronaut peuvent également être générés à l'aide d'un générateur en ligne



The screenshot displays the Micronaut Launch web interface. At the top left is the Micronaut logo, a stylized starburst with a Greek letter mu (μ) in the center. To its right is the text "MICRONAUT™" and "LAUNCH" below it. In the top right corner, there are four social media icons: information, GitHub, and two others. Below the header, there are four configuration fields: "Application Type" (set to "Application"), "Java Version" (set to "11"), "Base Package" (set to "com.example"), and "Name" (set to "demo"). Below these are four more configuration fields: "Micronaut Version" (radio buttons for "2.1.3" and "2.2.0.BUILD-SNAPSHOT"), "Language" (radio buttons for "Java", "Kotlin", and "Groovy"), "Build" (radio buttons for "Gradle" and "Maven"), and "Test Framework" (radio buttons for "JUnit", "Spock", and "KoTest"). At the bottom of the configuration section, there are four buttons: "+ FEATURES", "↔ DIFF", "🔍 PREVIEW", and "📄 GENERATE PROJECT". Below the buttons, it says "Included Features (0)". At the bottom center, there is the Object Computing logo, a stylized 3D cube, and the text "OBJECT COMPUTING HOME TO MICRONAUT™".



# CAS D'UTILISATIONS

Avec Micronaut, on peut construire des applications de type :

- Microservices
- Serverless
- Message-Driven : axés sur les messages avec Kafka/Rabbit
- CLI : interface en ligne de commandes
- Android





# AVANTAGES

- Temps de démarrage rapide
- Réduction de l'empreinte mémoire
- Utilisation minimale de la réflexion
- Utilisation minimale des proxys
- Pas de génération d'environnement d'exécution du bytecode
- Robuste, Scalable:
- Test unitaire facile



GraalVM™



# AVANTAGES

	Memory - MB	Start-up time - ( ms )	Execution time -( ms )
Micronaut app with Native-Image	61	18	3
Micronaut app on GraalVM JIT	278	1,100	10
SpringBoot + HotSpotVM JIT	338	1,400	15



# INCONVÉNIENTS

- Certaines dépendances ne supportent pas (encore) GraalVM :
- Micronaut n'offre pas autant de fonctionnalités ou d'intégrations comme Spring Boot
- il n'existe actuellement aucune prise en charge des vues côté serveur ou des fonctionnalités typiques d'un framework MVC côté serveur traditionnel.



# DÉMONSTRATION



**MERCI POUR VOTRE ATTENTION**



# CRÉDITS

- <https://dzone.com/articles/micronaut-build-for-next-wave-of-microservices-amp>
- <https://docs.micronaut.io/1.0.0/guide/index.html#serverlessFunctions>
- <https://desosa.nl/projects/micronaut/2020/03/17/from-vision-to-architecture.html#sec-runtime-view>
- <https://micronaut.io/launch/>