



# Serverless pod strzechą

Wprowadzenie do Fn Project

19/02/2019

# Serverless

The background of the slide is a solid blue color. On the right side, there are two large, overlapping circles of a slightly darker shade of blue. The circles are positioned such that they appear to be floating or layered over the main background.

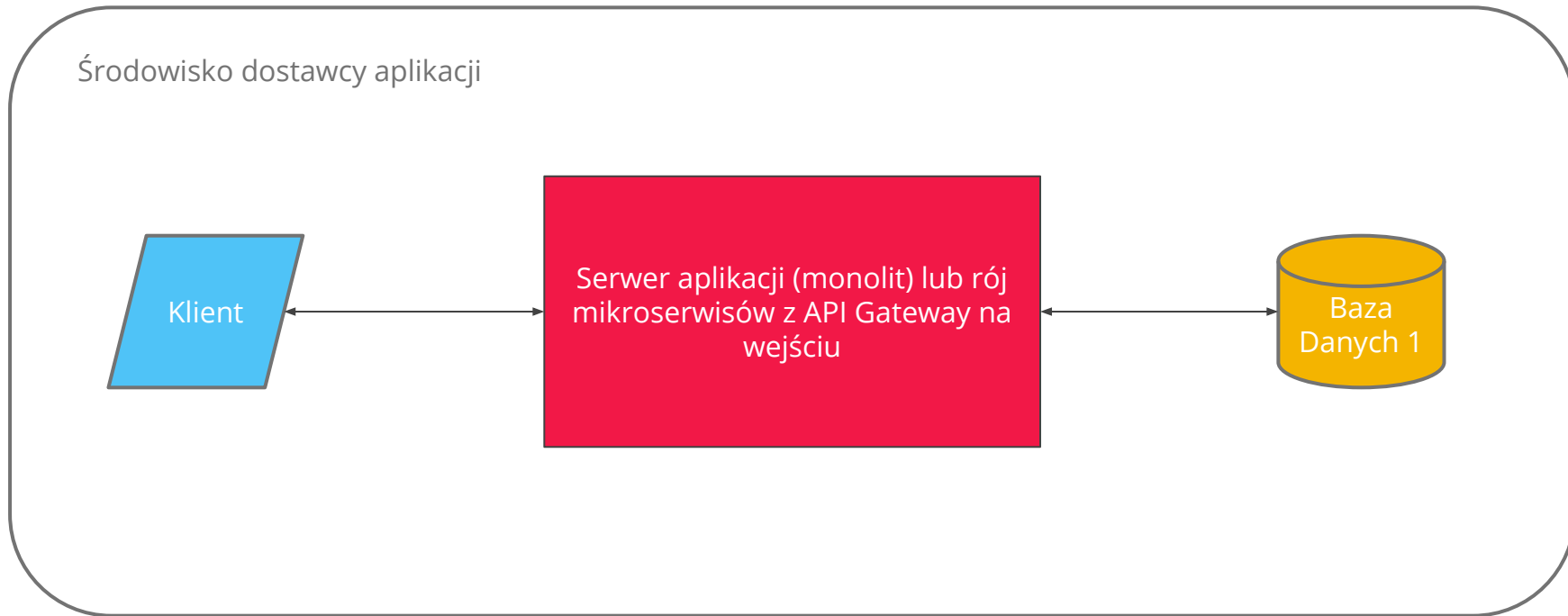
# Serverless

## Czym jest architektura Serverless?

- Buzzword, użyty po raz pierwszy w 2012, na topie w 2016
- Zakłada wyeliminowanie software'u i hardware'u
- Ogólnie, bardzo szerokie pojęcie
- Można podzielić na BaaS i FaaS

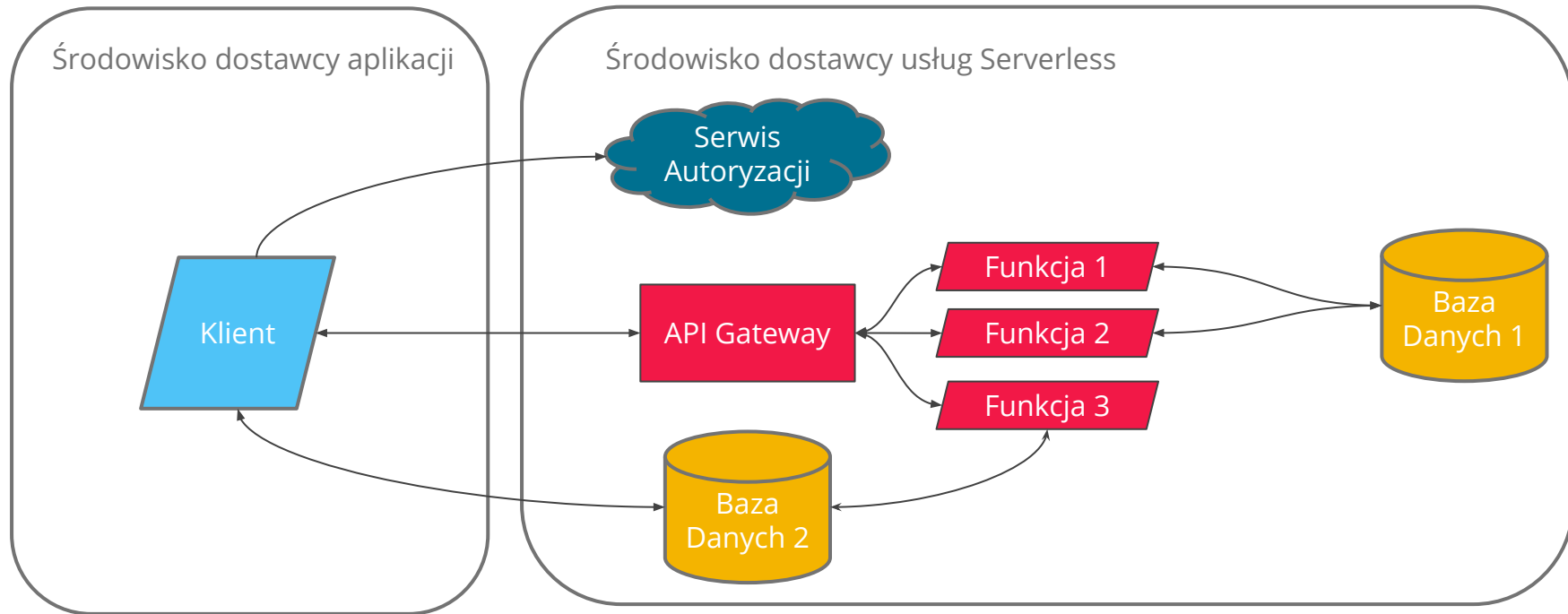
# Serverless

## Tradycyjna architektura aplikacji webowej



# Serverless

## Architektura aplikacji Serverless



# Serverless

## Najpopularniejsze rozwiązania Serverless

AWS Lambda



Microsoft Azure



Google Cloud Functions



**Google Cloud Functions**



Serverless

Alternatywy?

Tak! Jedną z nich może być np.

**Fn Project**

Fn Project



## Fn Project

“ *The Fn project is an open-source container-native serverless platform that you can run anywhere -- **any cloud or on-premise**. It's easy to use, supports every programming language, and is extensible and performant.*

— <https://fnproject.io>

Platforma Serverless wymagająca serwera?!



# Fn Project

## Czym charakteryzuje się Fn Project

- Platforma Serverless (FaaS) do instalacji na własnym serwerze lub w chmurze
- Napisana w Go
- Open source (licencja Apache 2.0)
- Posiada FDK (Functions Developer Kit) dla kilku języków: Go, Java, Node.js, Python, Ruby
- W pełni oparta na kontenerach (Docker)
- Posiada CLI i Web UI\* do zarządzania i monitorowania

# Fn Project

## Wymagania

- Docker 17.10.0-ce lub nowszy, zainstalowany i odpalony
- Konto na Docker Hub lub jakiś inny Docker Registry
- Trzeba być zalogowanym do Docker Hub (`docker login`)
- FN Project CLI\*

\*Nie jest wymagane, ale bardzo pomocne w pracy z Fn Project. W moich przykładach również korzystam z CLI. Dlatego je tutaj wymieniałem.

# Fn Project

## Tworzenie i uruchamianie funkcji

- Uruchomienie kontenera fn: `fn start`
- Utworzenie pustej funkcji: `fn init --runtime java myfun`
  - Z triggerem HTTP: `--trigger http`
- Deploy funkcji na kontenerze fn: `fn deploy --app myapp`
  - Lokalnie, z pominięciem Docker Huba: `--local`
  - Bez podbijania wersji: `--no-bump`
- Wywołanie funkcji: `fn invoke myapp myfun`
  - Jeśli funkcja ma trigger http: `curl http://localhost:8080/t/myapp/myfun`
- Przebudowanie funkcji: `fn build`

# Fn Project

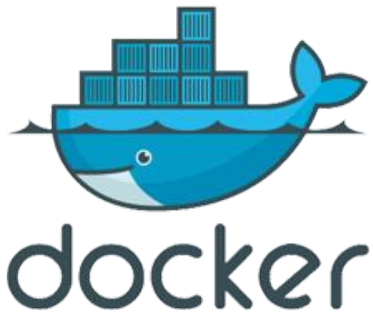
```
edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$ fn invoke wyapp simple-function
Hello, world!edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$
edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$ echo -n "Janusz" | fn invoke wyapp simple-function
Hello, Janusz!edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$
edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$ curl localhost:8080/r/wyapp/simple-function-trigger
Hello, world!
edmins-MacBook-Pro-VTAAW:simple-function szymon.winiarz$ curl -i localhost:8080/r/wyapp/simple-function-trigger
```

[github.com/SzymonWiniarz/fnproject-introduction/tree/master/simple-function](https://github.com/SzymonWiniarz/fnproject-introduction/tree/master/simple-function)

# Fn Project

## Docker HUB

Fn Project jest domyślnie zintegrowany z Docker Hubem, więc funkcje które napiszemy można uploadować do tego serwisu. Dzieje się to również automatycznie za każdym razem, kiedy instalujemy nową wersję naszej funkcji na serwerze Fn (`fn deploy`).



# Fn Project

```
---> 9e7729a917d7
Step 7/11 : RUN ["echo", "package"]
---> Using cache
---> 374dc7848ba7
Step 8/11 : FROM fnproject/fn-java-fdk:1.0.0
---> 87ebb59f2118
Step 9/11 : WORKDIR /function
---> Using cache
---> fdc1157c79c4
Step 10/11 : COPY --from=build-stage /function/target/*.jar /function/app/
---> Using cache
---> 511735d9efc7
Step 11/11 : CMD ["com.example.fn.HelloFunction::handleRequest"]
---> Using cache
---> d49777af7251
Successfully built d49777af7251
Successfully tagged srwinars/simple-function:0.0.3

Parts: [srwinars/simple-function:0.0.3]
Pushing srwinars/simple-function:0.0.3 to docker registry...The push refers to repository [docker.io/srwinars/simple-f
unction]
4a591888d81a: Pushed
8512778944d7: Pushing [=====] 20.15MB
8d3188ba7843: Pushed
977c8dc28b5e: Pushed
eed297584e43: Pushing [=====] 49.27MB/156.0MB
eed41482a727: Mounted from fnproject/fn-java-fdk-build
cd98f895a9ac: Mounted from fnproject/fn-java-fdk-build
388282b717b7: Pushing [=====] 1.43MB
bc81ab4ead84: Waiting
```



# Fn Project

## Grupowanie funkcji w aplikacji

- Tworzenie aplikacji: plik `app.yaml` z zawartością: `name: myapp`
- Deployowanie całej aplikacji: `fn deploy --all`
- Listowanie dostępnych aplikacji: `fn list apps`
- Listowanie funkcji w obrębie aplikacji: `fn list functions myapp`
- Listowanie triggerów w obrębie aplikacji: `fn list triggers myapp`

# Fn Project

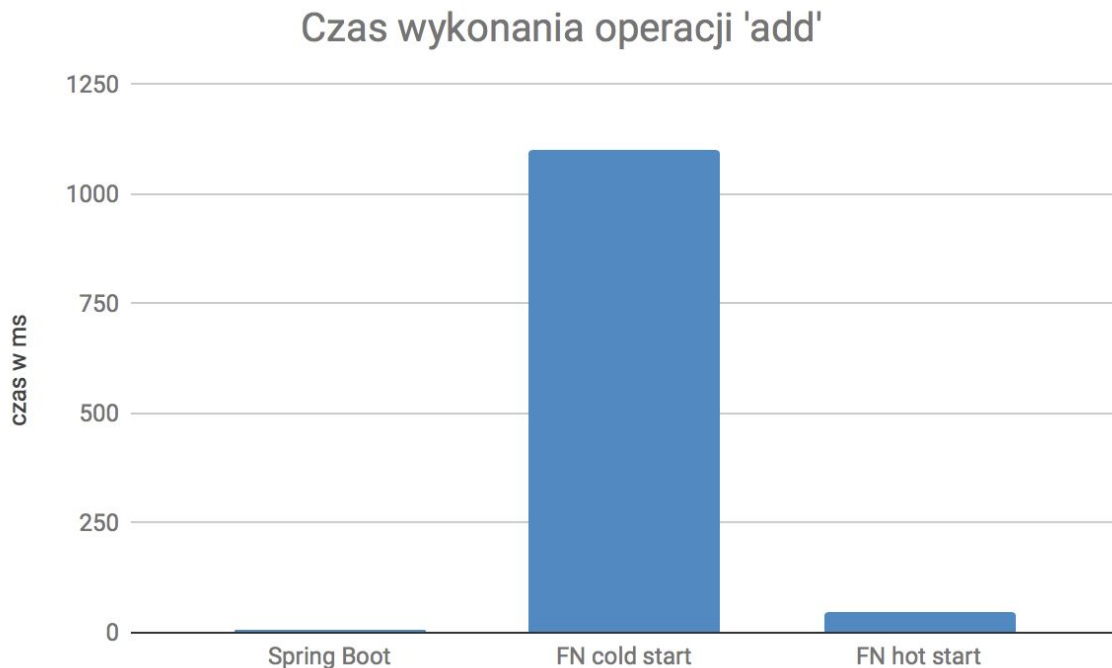
```
admins-MacBook-Pro-V29AMH:fn-calculator-app szymon.winiarz$ fn list apps
NAME      ID
calculator-app  #104110670M8C0C0P0C110000007
myapp      #10317R568M8C0C0P0C110000001
admins-MacBook-Pro-V29AMH:fn-calculator-app szymon.winiarz$ fn list functions myapp
NAME      ID
simple-function s1winia7/simple-function:0.0.0 #10317R568M8C0C0P0C110000002
admins-MacBook-Pro-V29AMH:fn-calculator-app szymon.winiarz$ fn list functions calculator-app

fn: app calculator not found

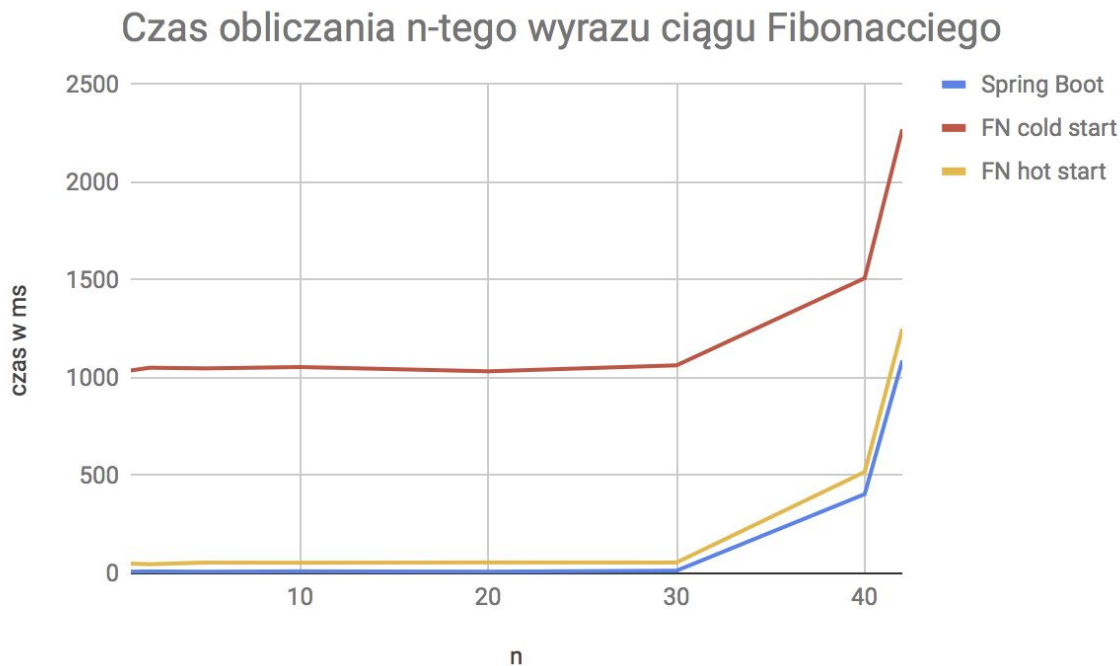
See 'fn <command> --help' for more information. Client version: 0.8.40
admins-MacBook-Pro-V29AMH:fn-calculator-app szymon.winiarz$ fn list functions calculator-app
NAME      IMAGE      ID
add-function      fndemouser/add-function:0.0.2  #104110717M8C0C0P0C110000008
divide-function   fndemouser/divide-function:0.0.2 #104110740M8C0C0P0C11000000A
fibonacci-function fndemouser/fibonacci-function:0.0.2 #104110817M8C0C0P0C11000000C
multiply-function fndemouser/multiply-function:0.0.2 #104110840M8C0C0P0C11000000E
subtract-function fndemouser/abstract-function:0.0.2 #104110917M8C0C0P0C11000000G
admins-MacBook-Pro-V29AMH:fn-calculator-app szymon.winiarz$
```

[github.com/SzymonWiniarz/fnproject-introduction/tree/master/fn-calculator-app](https://github.com/SzymonWiniarz/fnproject-introduction/tree/master/fn-calculator-app)

## Porównanie wydajności z tradycyjną aplikacją opartą o Servlety



## Porównanie wydajności z tradycyjną aplikacją opartą o Servlety

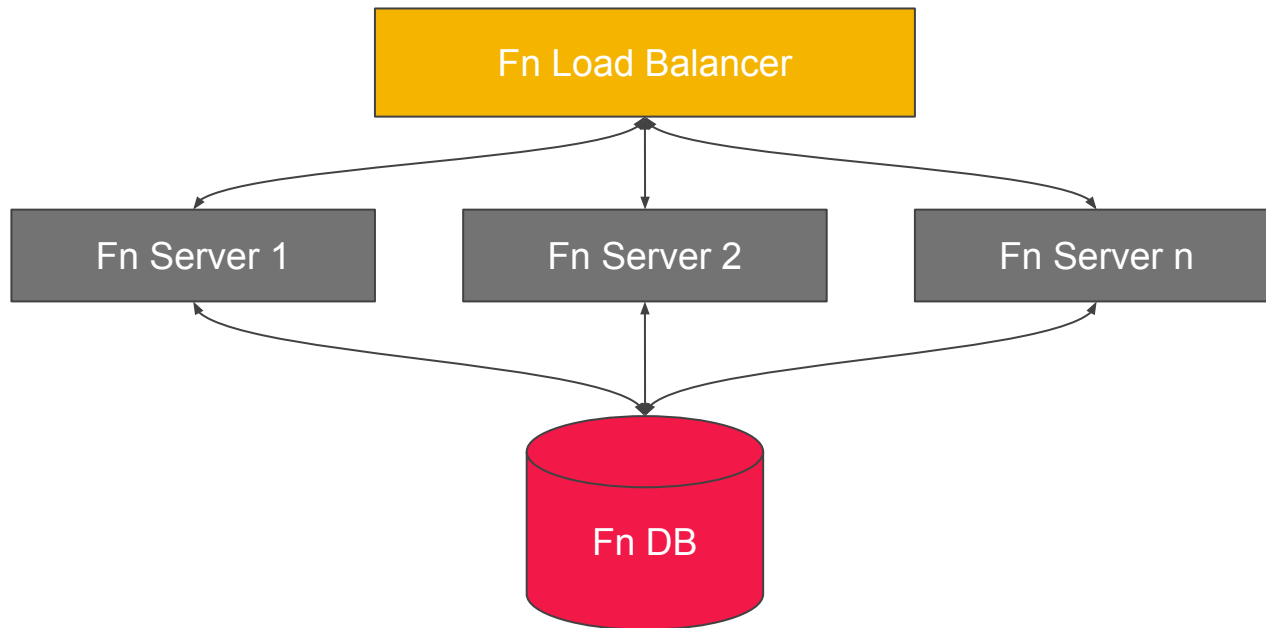


The background is a solid blue color. On the right side, there is a large, faint, light-blue circle. The entire background is covered with a pattern of thin, light-blue diagonal lines running from the top-left to the bottom-right.

Fn Project - tematy zaawansowane

# Fn Project - tematy zaawansowane

## Fn Project w klastrze



# Fn Project - tematy zaawansowane

## Fn Project w klastrze

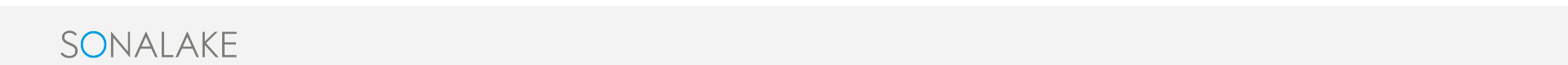
- Start wielu instancji serwera Fn:

```
docker run --rm -i --name fnserverX -v FN_ROOT_DIR/.fn/iofs:/iofs -e  
FN_IOFS_DOCKER_PATH=FN_ROOT_DIR/.fn/iofs -e FN_IOFS_PATH=/iofs -v  
FN_ROOT_DIR/.fn/data/fn.db:/app/data/fn.db -v  
/var/run/docker.sock:/var/run/docker.sock --privileged -p 808X:8080 --entrypoint  
./fnserver fnproject/fnserver:latest
```

- Start load balancera:

```
docker run --rm --name fnlb -p 8080:8081 fnproject/fnlb:latest --nodes  
HOST:8081,HOST:8082,(...)
```

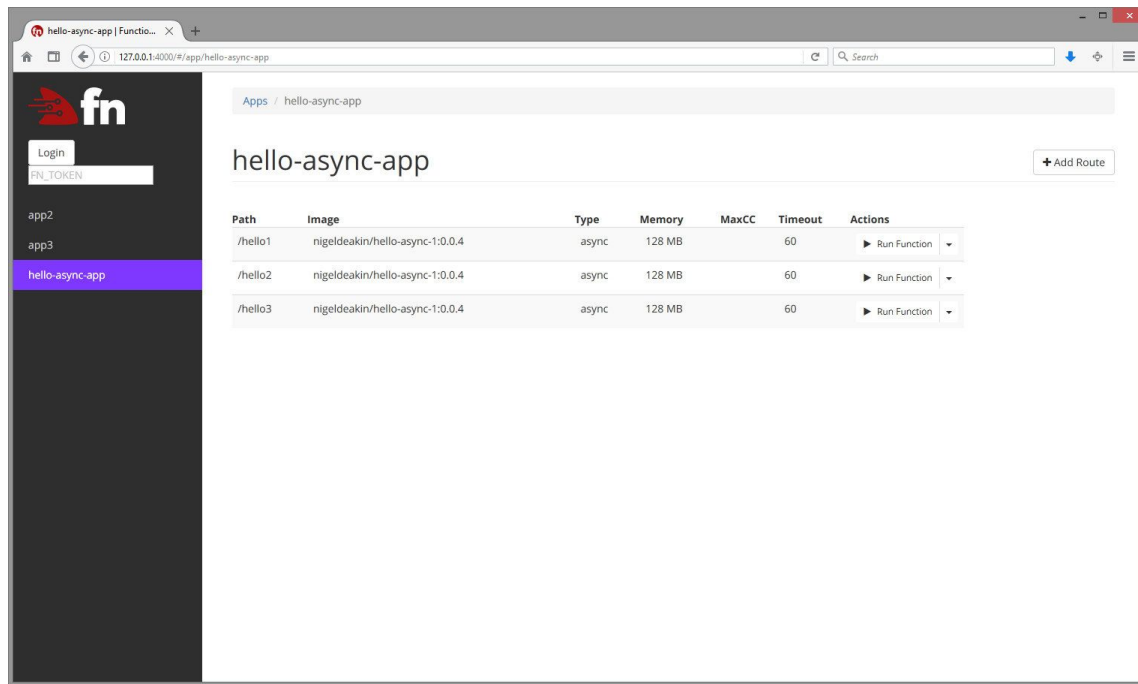
Figure 1





# Fn Project - tematy zaawansowane

## Web UI



# Fn Project - tematy zaawansowane

## Web UI

Uruchomienie:

- `fn start` - uruchamia serwer Fn który będzie monitorowany
- `docker run --rm -it --link fnserver:api -p 4000:4000 -e "FN_API_URL=http://api:8080" fnproject/ui` - uruchamia UI serwer

# Fn Project - tematy zaawansowane

## Web UI

Na chwilę obecną jest tylko jedno “ale”...

📖 README.md

⚠️ **WARNING: this repo has gone stale and the UI is currently broken. It is not intended to be left for dead forever, feel free to fix it up, this has become a lower priority item at the moment however for the fn team.**

UI for Fn  PASSED

# Fn Project - tematy zaawansowane

## Natywne funkcje w Javie - Fn Project + GraalVM

W skrócie:

- GraalVM wykorzystany do kompilacji Javy do natywnego kodu
- Natywne funkcje w Javie wypadają lepiej pod względem szybkości działania i zużycia pamięci w porównaniu z funkcjami w Go
- Rozmiar plików wykonywalnych porównywalny z funkcjami w Go dzięki zastosowaniu `scratch` jako bazowego obrazu Dockerowego

Więcej tutaj:

<https://medium.com/criciumadev/serverless-native-java-functions-using-graalvm-and-fn-project-c9b10a4a4859>

# Podsumowanie

# Podsumowanie

## Fn Project:

- Platforma Serverless (FaaS) do zainstalowania na własnym serwerze
- Umożliwia pisanie funkcji w dowolnym języku programowania
- Oparta na kontenerach (Docker)
- Posiada CLI przyspieszające i ułatwiające pracę
- Wydajność zbliżona do porównywalnej aplikacji Spring Boot
- Wielomodułowe aplikacje - brak wsparcia out of the box
- Słaby domyślny exception handling
- Skalowalna (dedykowany load balancer `fnlb`)
- ~~• Dostarcza Web UI do łatwiejszego monitorowania~~

# Źródła i inne ciekawe materiały

Artykuł o Serverless - <https://martinfowler.com/articles/serverless.html>

Artykuł z pierwszą wzmianką o Serverless z 2012 -

<https://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>

Strona Fn Project - <https://fnproject.io>

Fn Load Balancer - <https://github.com/fnproject/lb>

Web UI dla Fn Project - <https://github.com/fnproject/ui>

Private registry zamiast Docker Hub -

[https://github.com/fnproject/docs/blob/master/fn/operate/private\\_registries.md](https://github.com/fnproject/docs/blob/master/fn/operate/private_registries.md)

Artykuł o GraalVM + Fn -

<https://medium.com/criciumadev/serverless-native-java-functions-using-graalvm-and-fn-project-c9b10a4a4859>

Artykuł o Load Balancingu w Fn -

<https://hackernoon.com/part-ii-fn-load-balancer-585babd90456>

# Dziękuję!

Pytania?