ONICLOUD

FROM ZERO TO SERVERLESS IN 60 SECONDS

Efen Gonzalez

AGENDA

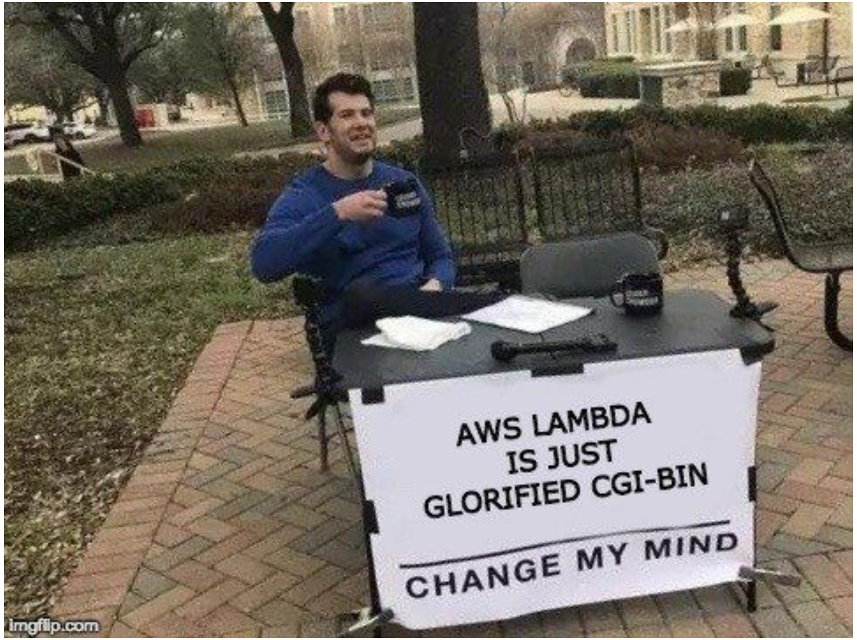
- Serverless Overview.
- What is FaaS.
- FaaS Landscape.
- FaaS Components.
- OpenFaaS Introduction.
- Demo - Installing the Open FaaS Platform.
- OpenFaaS Architecture.
- Demo 2 - Create First Functions
- Demo 2a - Development flow.
- Q & A

WHAT IS SERVERLESS

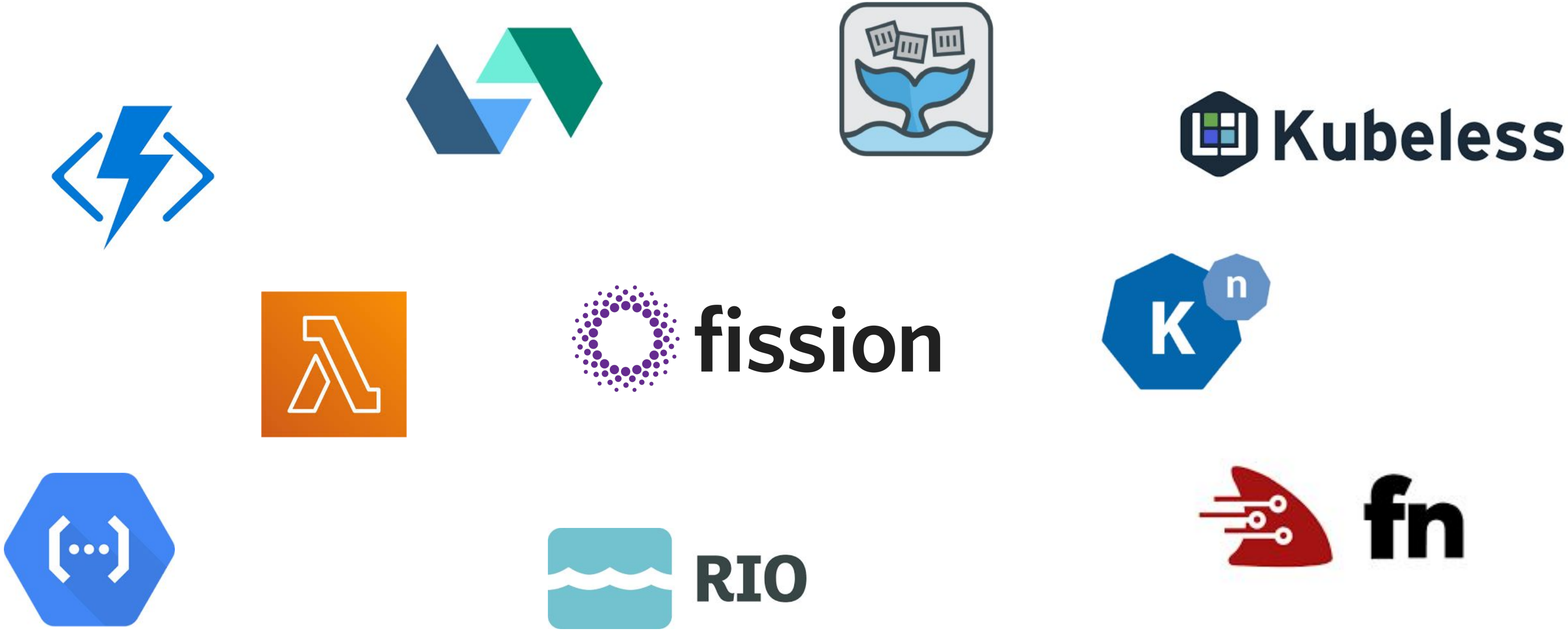
- No servers to manage or provision.
- Consumption priced, not capacity.
- Scales with usage.
- Availability and fault tolerance from the get go.

SERVERLESS FUNCTIONS

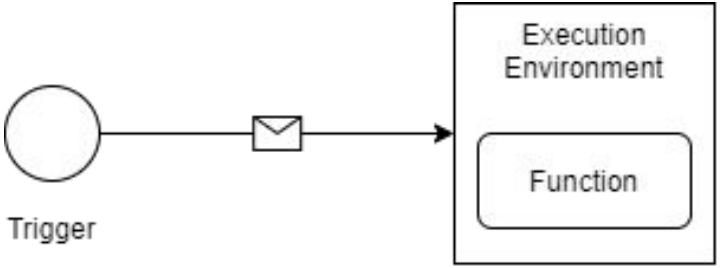
- Single purpose.
- Are hosted on managed infrastructure.
- modular chunks of functionality into the cloud that are executed independently.



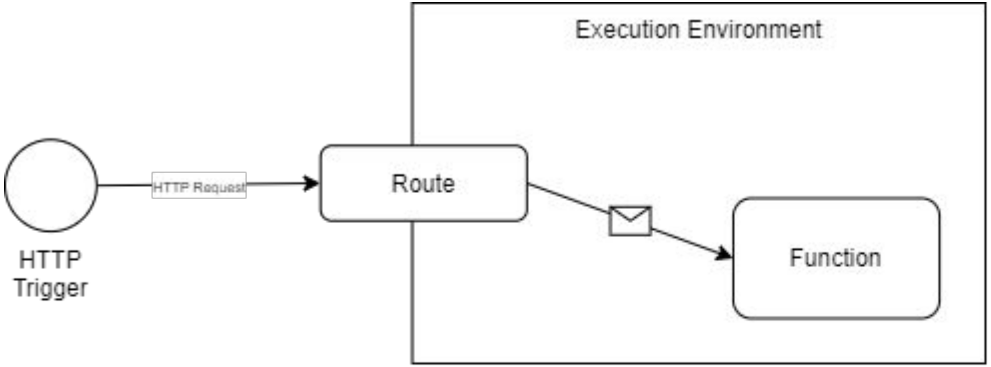
FAAS PROVIDER LANDSCAPE



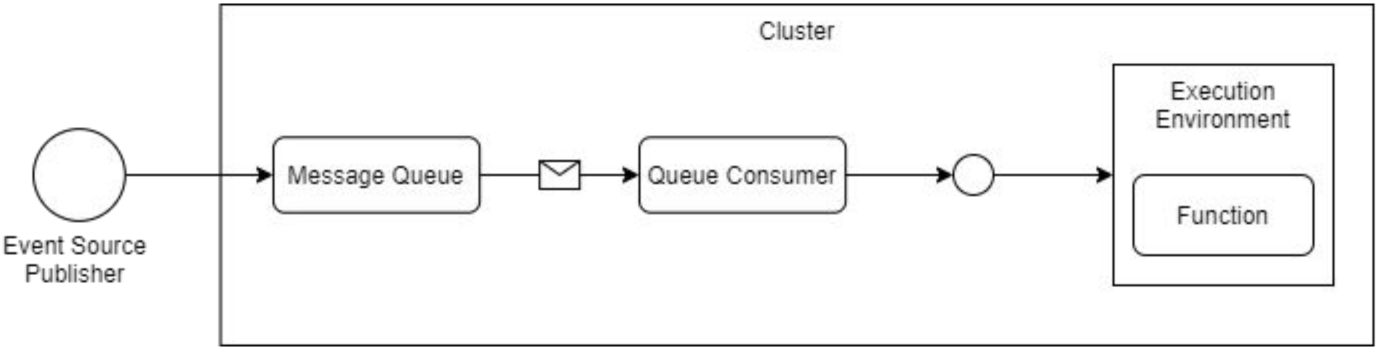
FAAS EXECUTION MODELS



FAAS EXECUTION MODELS - HTTP TRIGGER



FAAS EXECUTION MODELS - MESSAGE QUEUE TRIGGER



OPENFAAS

Serverless Functions, Made Simple.

OPENFAAS

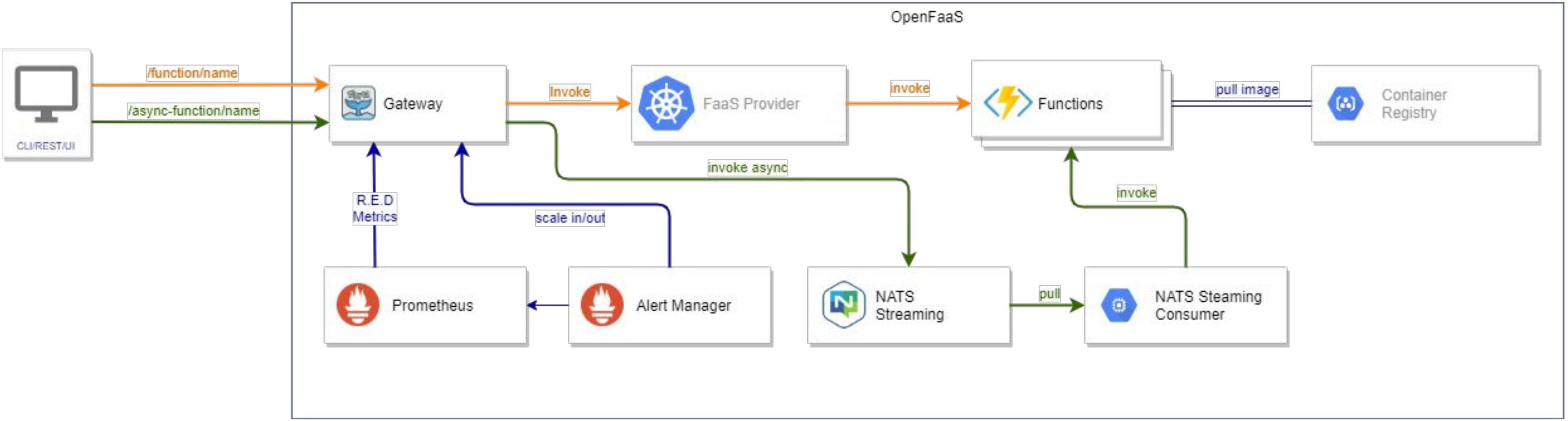
OpenFaaS® makes it easy for developers to deploy event-driven functions and microservices to Kubernetes.

OPENFAAS HIGHLIGHTS

- Ease of use through UI portal and one-click install
- Write functions in any language.
- Portable - runs on existing hardware or public/private cloud with Kubernetes or containerd
- CLI available with YAML format for templating and defining functions
- Auto-scales as demand increases

DEMO
INSTALLING OPENFAAS

OPENFAAS ARCHITECTURE



CREATING FUNCTIONS

CREATING FUNCTIONS

- Functions are created using templates
- Templates can be installed from the oficial openfaas repo or from a custom source
- The developer can create its own templates.
- It is a good practice to organize your functions by language and features and tag them if they are part of a service.

CREATING FUNCTIONS

```
$ faas-cli template store list
$ faas-cli template store list --url https://my.server.net/templates.json
```

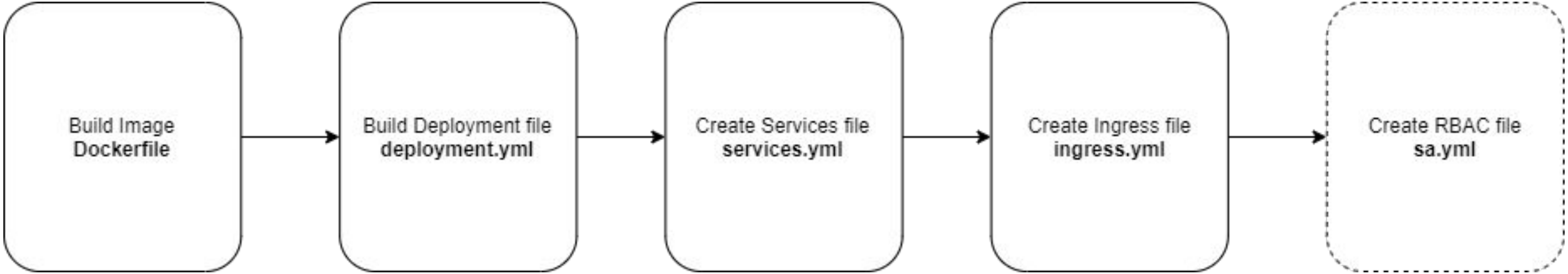
```
$ faas-cli template store pull template-name
$ faas-cli template store pull template-name --url https://my.server.net/templates.json
```

```
$ faas-cli new send-welcome-email --lang netcore50-function --prefix docker-hub-account
```

DEMO
CREATING FUNCTIONS

DEPLOYING FUNCTIONS

Developer Deploy Flow - Without OpenFaas



OPENFAAS

FUNCTIONS ARE DEPLOYED TO A GATEWAY

BUILDING AND DEPLOYMENT

```
$ faas-cli build -f my-function.yml
$ faas-cli push -f my-function.yml
$ faas-cli deploy -f my-function.yml -g https://my-function-gateway:8080
```

or

```
$ faas-cli up -f my-function.yml -g https://my-function-gateway:8080
```

DEMO
DEPLOYING FUNCTIONS

PROS, CONS AND OTHER STUFF

Benefits

- Runs in any cloud, PaaS or In Prem.
- Multiple languages supported.
- One stack for all functions.
- Leverage the portability of containers.
- Subsecond sale from zero.
- Template and functions store.
- Easy to jump in.

Drawbacks

- Scale to Zero opt-in.
- Containers always consume compute and memory.
- No hot deployment and reload of the function.
- No blue/green or canary deployments.
- Gateway functionality is bare minimal.

ONICLOUD

QUESTIONS