Distributing workloads in modern infrastructure - a dive into functions as a service

Jon Ander Novella
NeIC Tryggve system developer
National Bioinformatics Infrastructure Sweden

Issues with traditional infrastructure

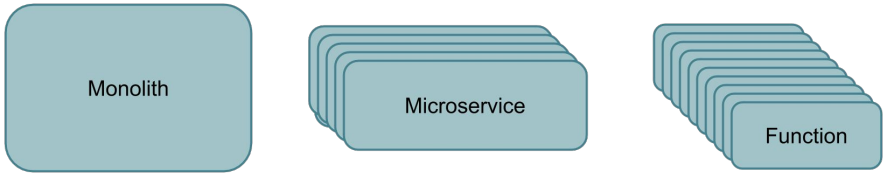
- Why should we be charged for keeping the server up even when we are ***not serving out any requests***?
- We are also in charge of the uptime and maintenance of the server and its resources
- The need for updating our services and stack stops us from achieving goals on time
- Resource usage fluctuates over time. ***Upscaling and downscaling*** are paramount

Typical challenges in research communities

- Diverse range of skills within platforms
- Lack of continuous delivery knowledge
- Need a model for ***timely web-scale delivery and reuse*** of support services: algorithms, tools for sensitive data discovery and archiving…
- Most services currently run on-premises and/or on public cloud services. It is difficult to do audit trailing

Serverless and functions as a service

- ***Serverless*** means you “can forget” about your machines
- ***FaaS*** is concept describing that stateless functions well be run “on-demand”, spinning and destroying resources as needed
- ***Stateless*** implies that your function is invoked in a new container every single time
- ***Functions triggered by events*** including http req, file uploads…

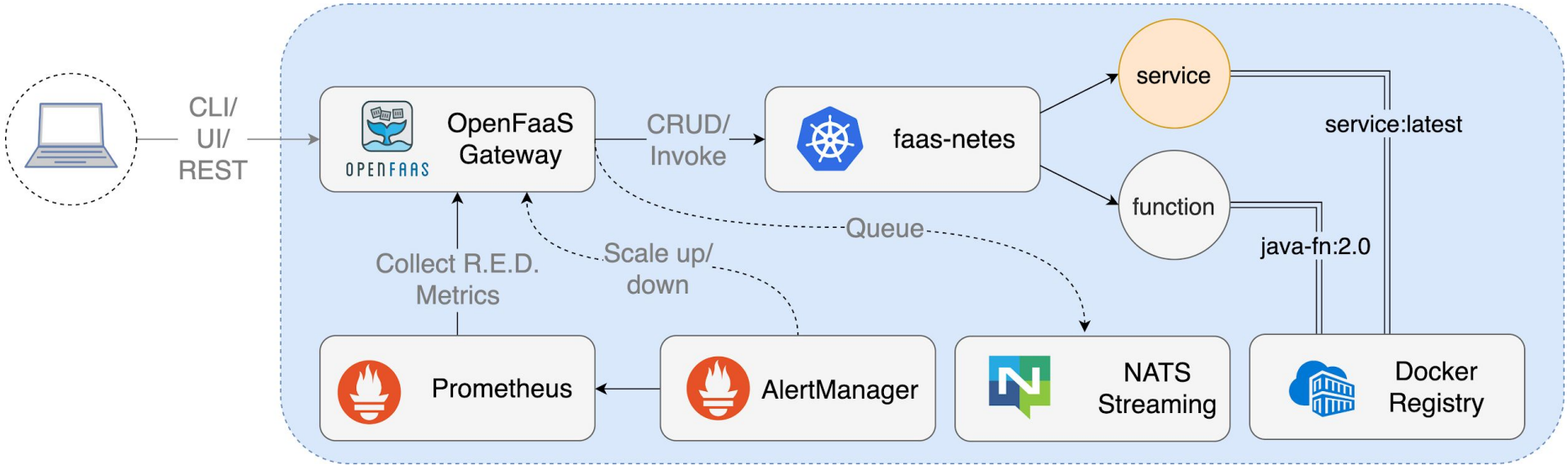


OpenFaaS: a lightweight but powerful framework

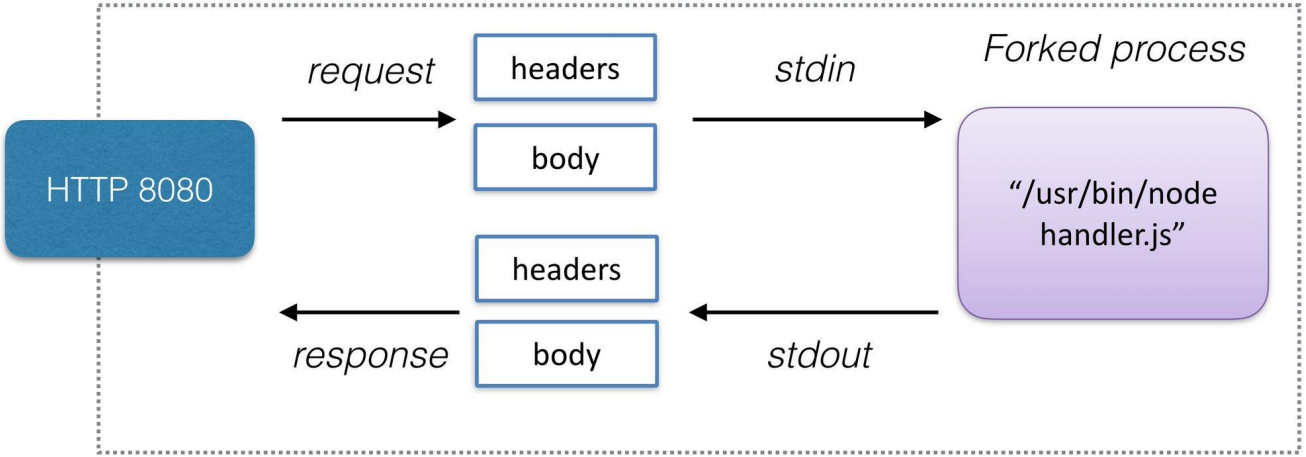
- ***Simple*** and ***lightweight***
- Built on top of ***Kubernetes*** and integrated with ***Helm***
- ***Deliver fast*** *without repetitive, boiler-plate coding*
- ***Scale*** as demand increases including to zero

OPENFAAS

Architecture and design of OpenFaaS



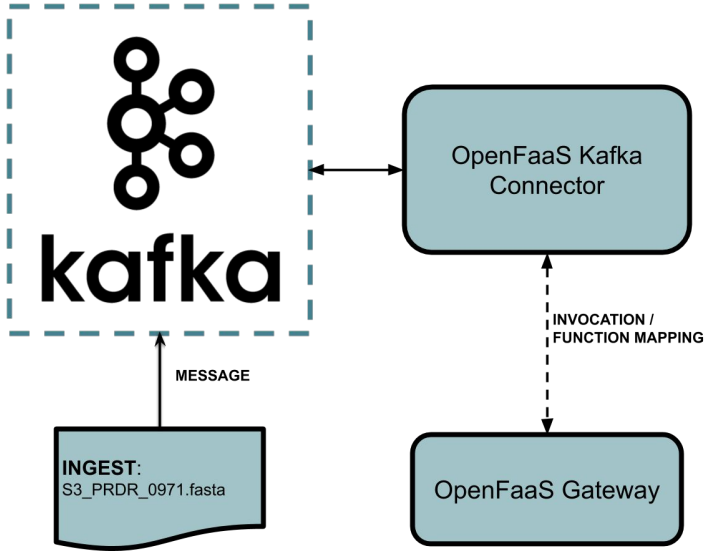
The OpenFaaS Watchdog



- The watchdog is a pass-through proxy with health-checks, metrics and compliance

Function-based (secure) workloads

- Workloads can be invoked sync or async via NATS streaming
- One of the challenges is to ***set accurately the timeouts*** of functions
- An alert manager which reads in performance metrics can scale the functions accordingly (***even to or from zero***)
- Workloads can be easily secured using a ***service mesh solution*** like Istio, and invocations can be authenticated using keyed-hash message authentication code



OpenFaaS hands-on

- The workshop playground can be found at: https://github.com/neicnordic/serverless-workshop
- The only requirements are Vagrant and a virtualization tool such as libvirt or virtualbox
- Under the exercise folder, you will find the instructions
- The OpenFaaS docs are available at: https://docs.openfaas.com/

*Special credit to **Alex Ellis** - Founder of OpenFaaS - for the diagrams used in this presentation*