OPENSHIFT FUNCTIONS AS A SERVICE & RED HAT SERVERLESS

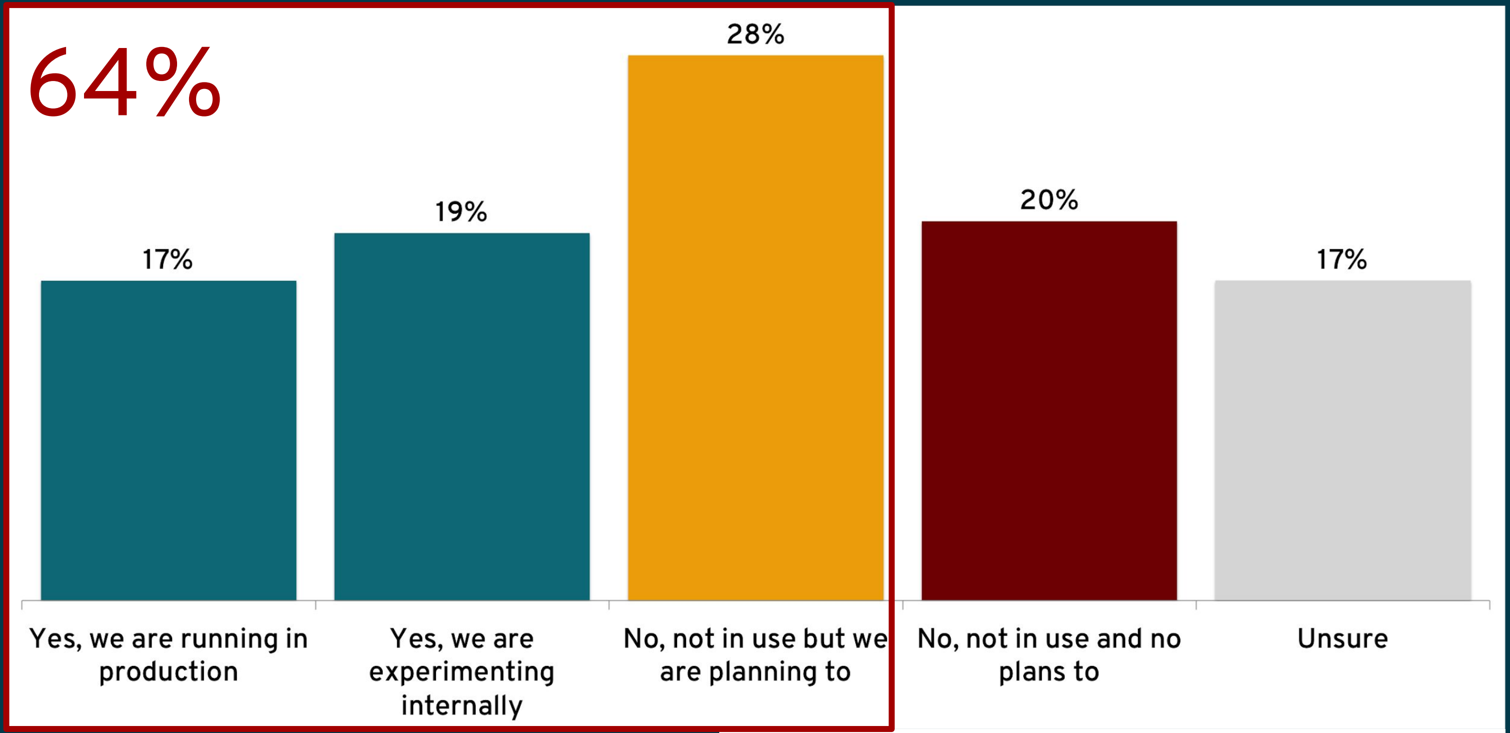
$ oc whoami
Natale Vinto
EMEA OpenShift Specialist Solution Architect
nvinto@redhat.com

T: @natalevinto #RedHat #openshift #serverless

"Serverless data center"

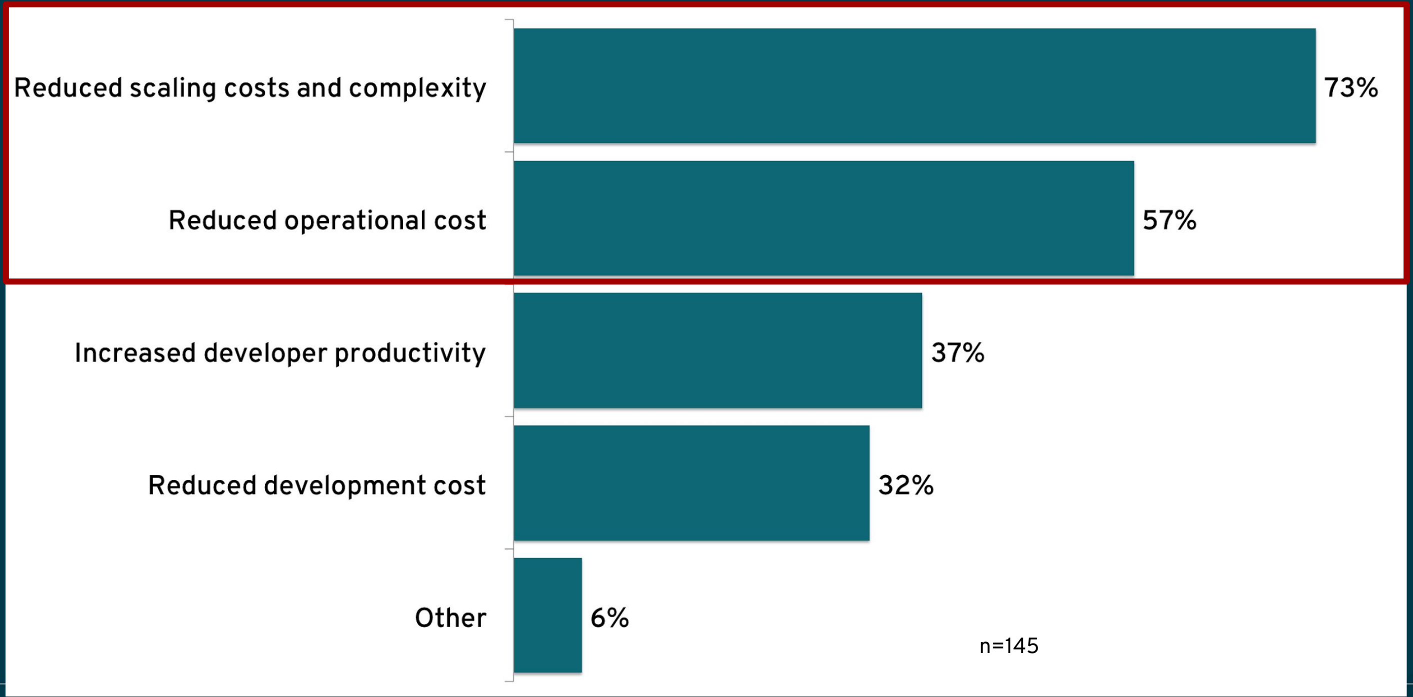
#nocode

Is your organization currently using serverless technologies?



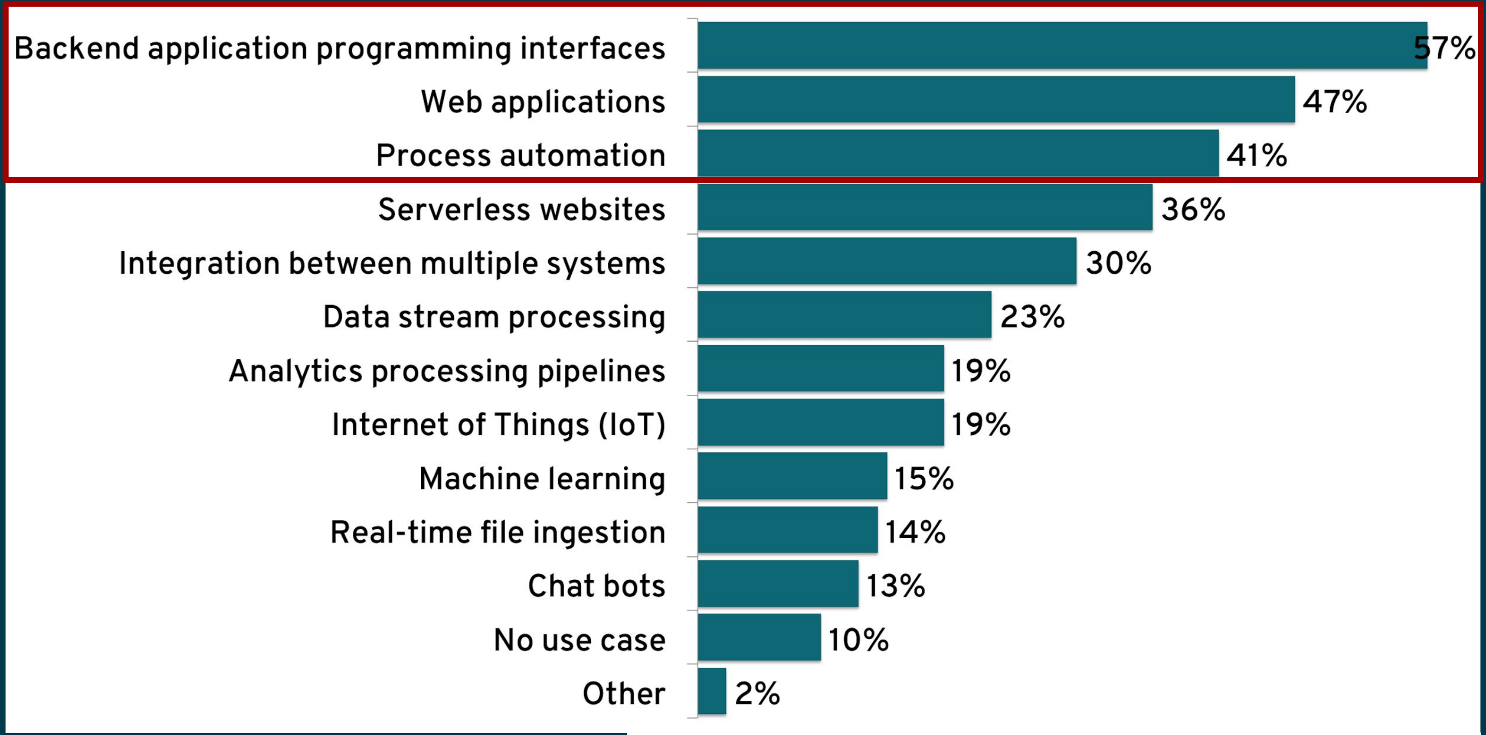
Source: Serverless Technology Research, April 2018, Red Hat

What benefits do you expect or are you already experiencing from using serverless technologies?



Source: Serverless Technology Research, April 2018, Red Hat

What are (or would be) your main use cases for using serverless technologies?



Source: Serverless Technology Research, April 2018, Red Hat

"Serverless data center"

Yes, there are servers.

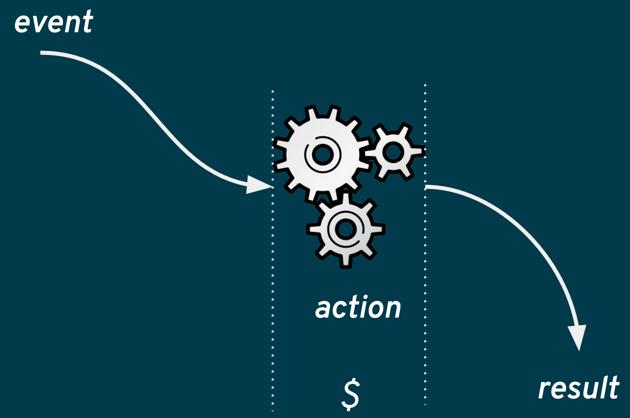
- Physical boxes running operating systems, VMs and containers.

Yes, there are servers.

- Processes listening on a TCP socket waiting for requests.

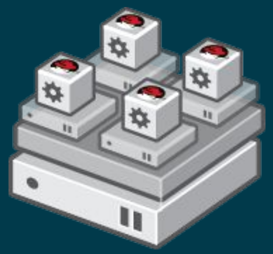
...but **the platform** takes care of provisioning, scaling, dispatching, monitoring all of those.

Serverless defined



From Wikipedia*:

“computing execution model that depends on services to manage server-side logic and state where server-side logic run in stateless, event-triggered compute containers"

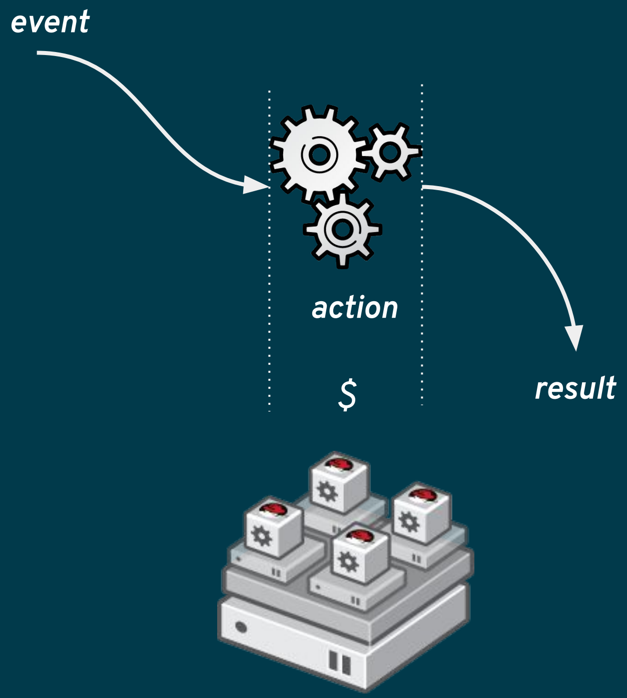


* https://en.wikipedia.org/wiki/Serverless_computing

Serverless defined

From MartinFowler.com*:

> *"…**applications where** some amount of **server-side logic** is still written by the application developer but unlike traditional architectures is **run in stateless compute containers** that are **event-triggered**, ephemeral (may only last for one invocation), and fully managed by a 3rd party"*
>
> *(Function as a Service or **FaaS**)*



* *https://martinfowler.com/articles/serverless.html*

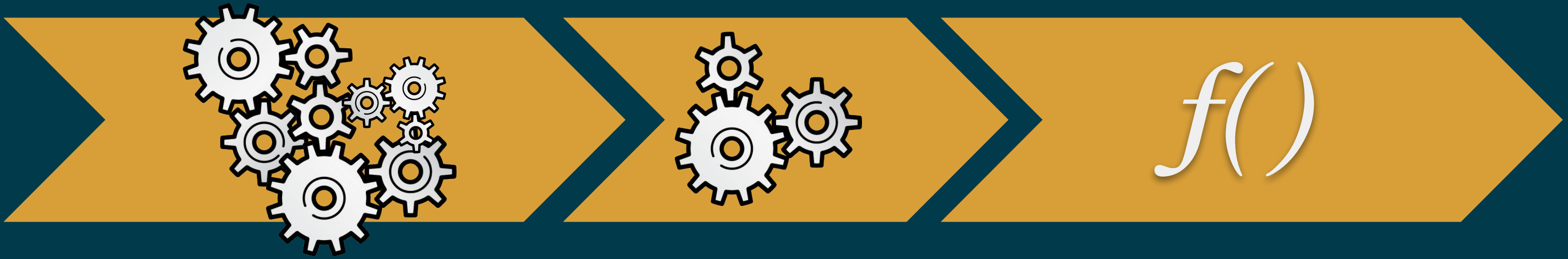
Serverless or Functions?

Both!

Functions is a **programming model**

Serverless is a **billing model** *@bibryam*

Architectural evolution



Service

> Autonomous
> Loosely-coupled

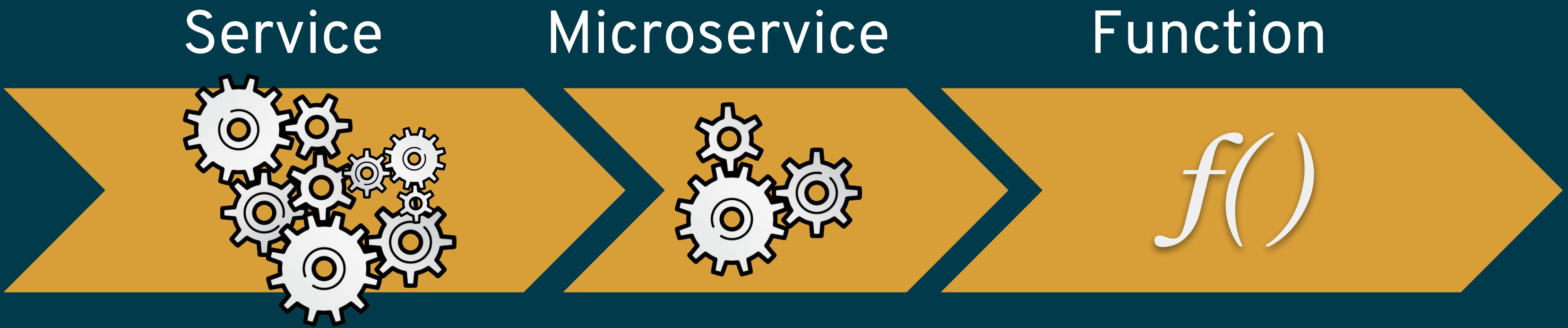
Microservice

> Single Purpose
> Stateless
> Independently Scalable
> Automated

Function

> Single Action
> Ephemeral

Architectural evolution



- > Autonomous
- > Loosely-coupled

- > Single Purpose
- > Stateless
- > Independently Scalable
- > Automated

- > Single Action
- > Ephemeral

Control & High complexity ⟷ Productivity & Low Control

PORTABILITY

Why do we need serverless ?

***Agility** of the cloud on any environment*

- ➔ *On-premise*
- ➔ *Multi-cloud*
- ➔ *Hybrid*

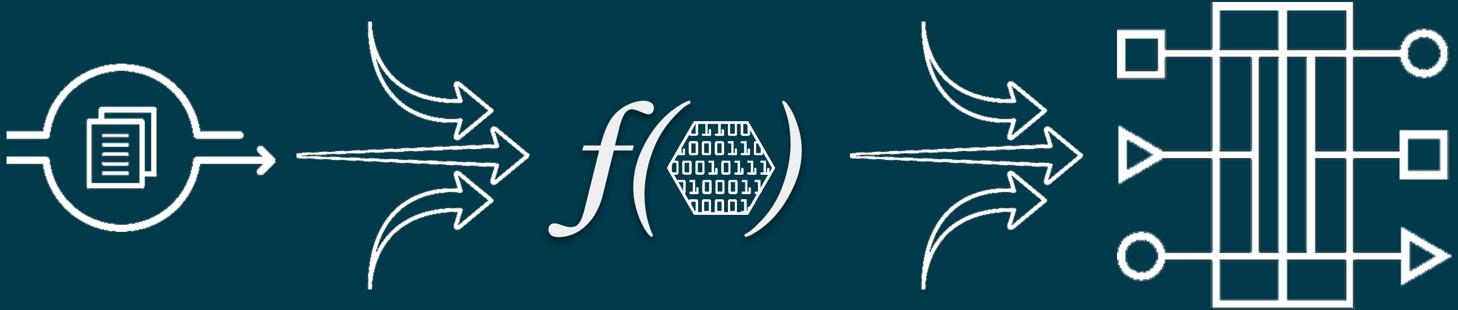
*Enable **event driven** cloud-native applications but also **integrate** with classic applications*

*Focus on business differentiation, abstract & delegate infrastructure to **platform** & **services***

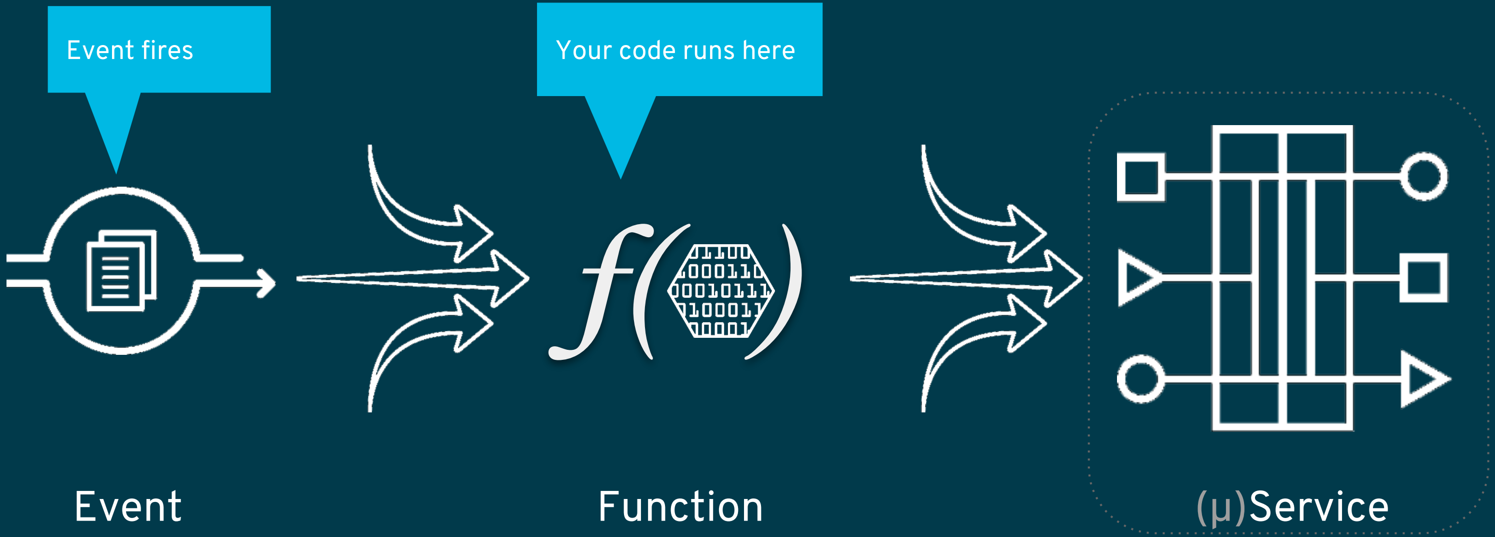
*Consistent and scalable **operations** across multiple applications*

Resource optimization & cost savings

How does it work ?

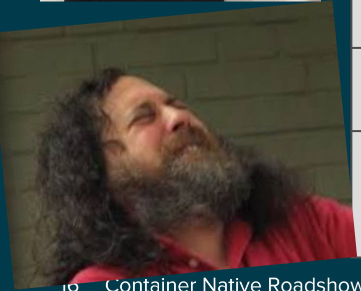


How does it work ?

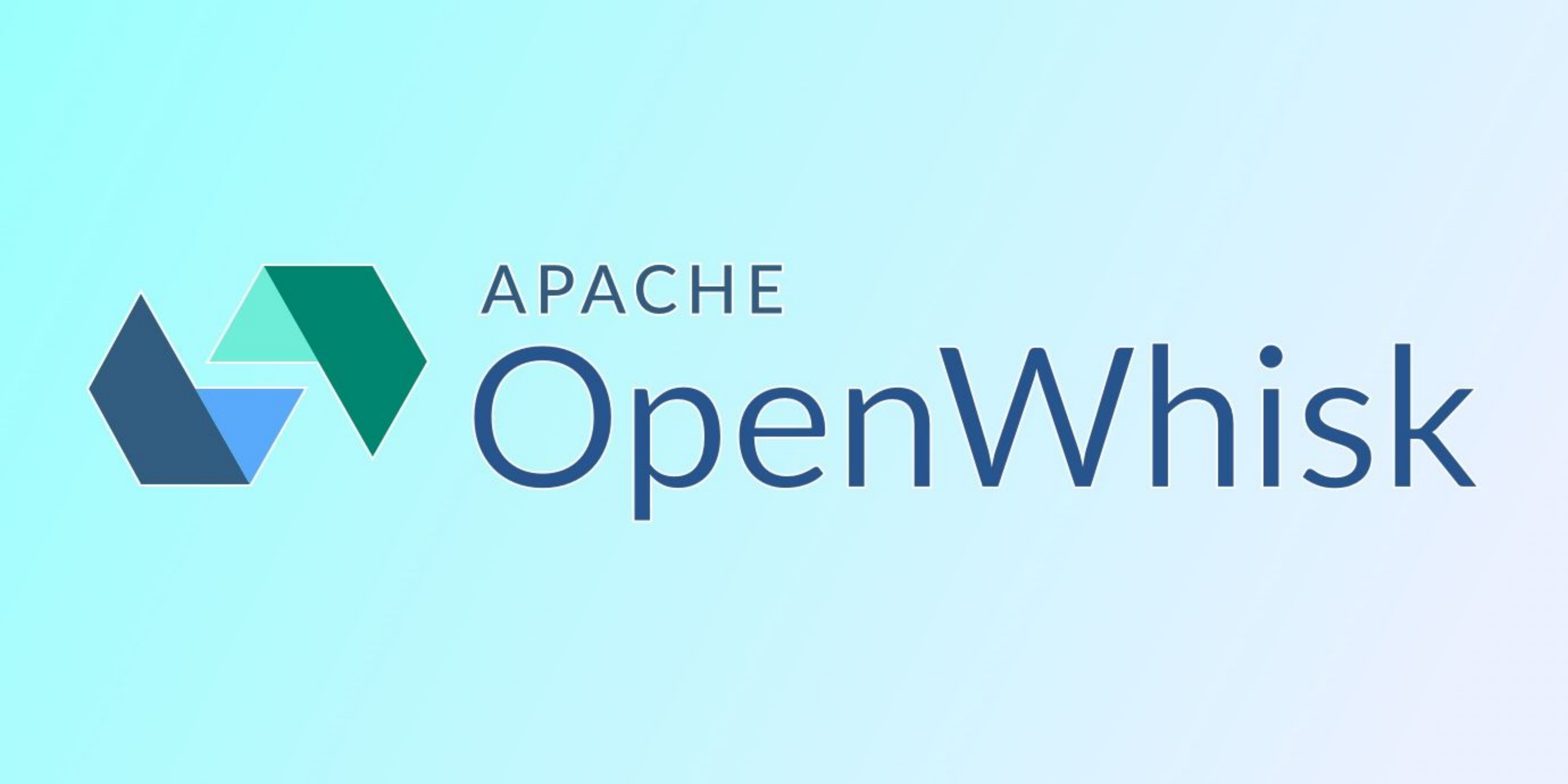


Serverless Solutions

Serverless scorecard

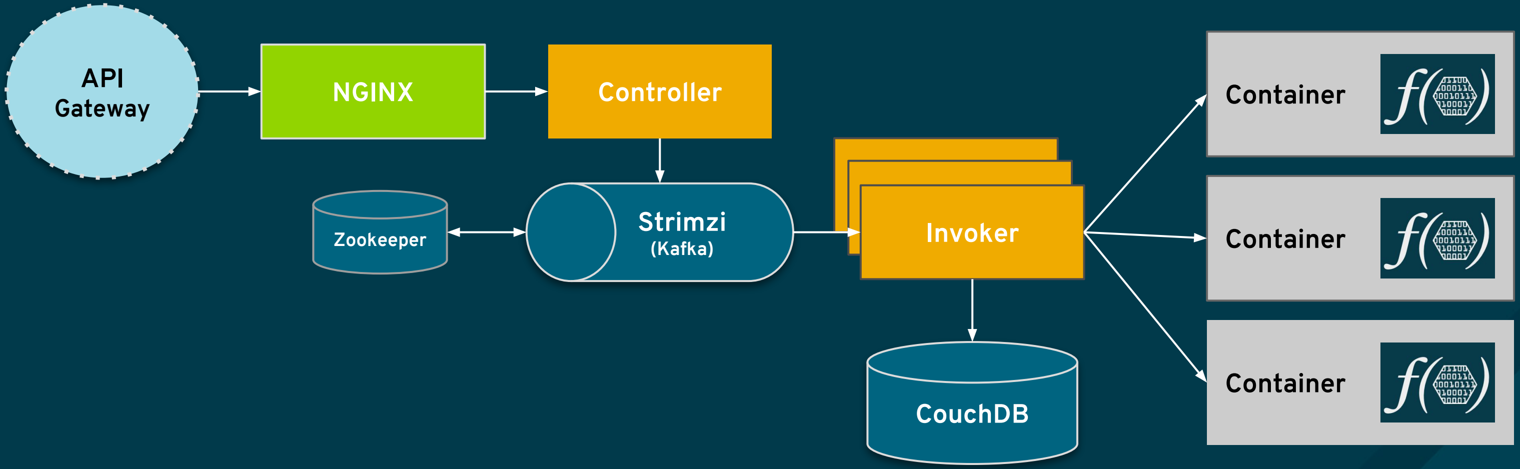


Project	Open Source	Kubernetes Support	Community Size	Feature Set	Started
Apache OpenWhisk	Yes	Yes	Large	★★★★	2015
Fission	Yes	Yes	Small	★★	2016
Funktion	Yes	Yes	Tiny	★★	2017
Project Riff	Yes	Yes	Tiny	★★	Late 2017
Amazon Lambda	No	No	Large	★★★★	2014
Azure Functions	No	No	Small	★★★	Late 2016
Google Cloud Functions *(beta)*	No	No	Small	★★★	2016

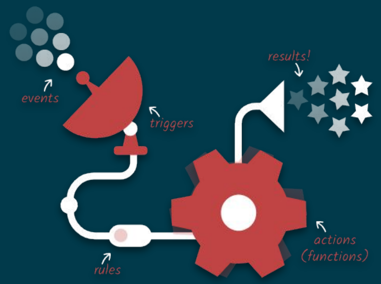

APACHE
OpenWhisk

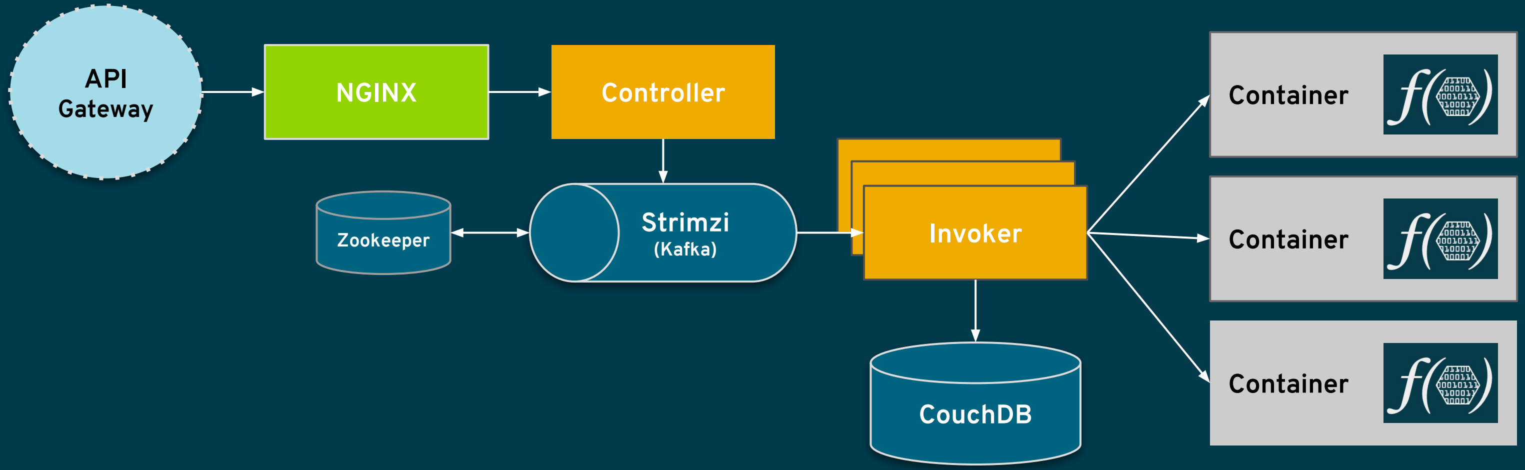
What is Apache OpenWhisk

- Complete Serverless solution
- Incubating project under Apache Software Foundation
- Started by IBM but with Adobe and Red Hat as contributors

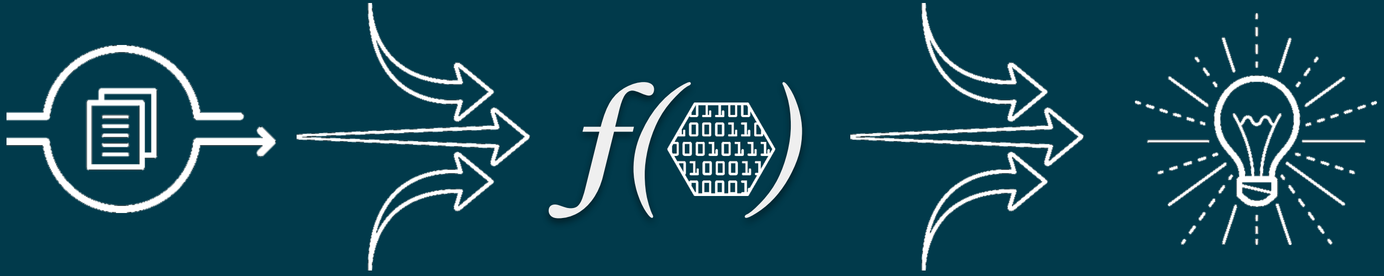


What is Apache OpenWhisk





How does it work ?

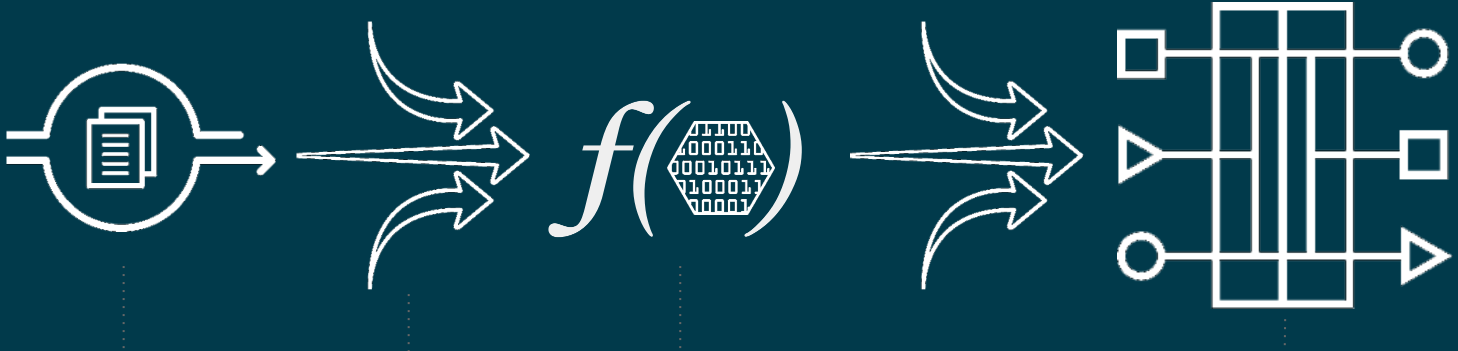


Event

Function

Results

How does it work ?



Event

Function

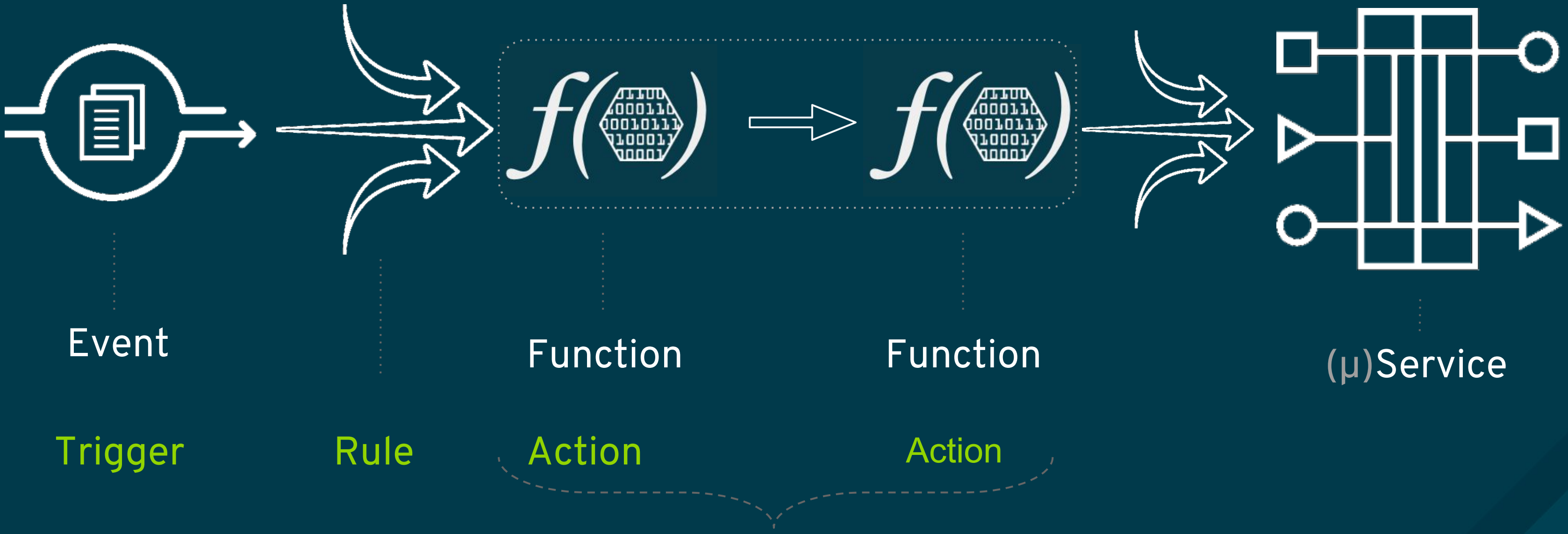
(μ)Service

Trigger

Rule

Action

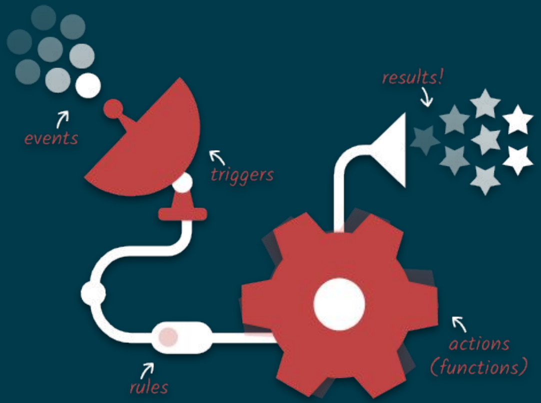
How does it work ?



Sequence

What is Apache OpenWhisk

Core concepts



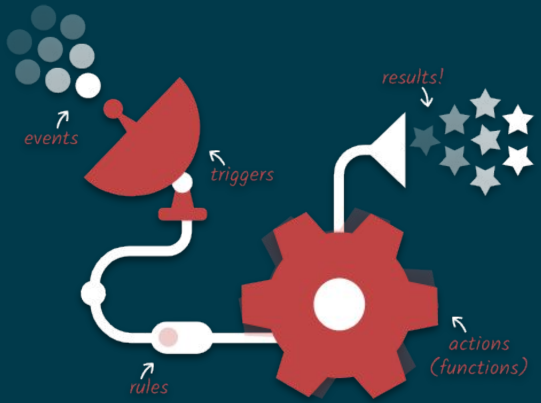
- **Triggers** - Class of events that can happen to start an action.
  - When a new person joins a chat room (`newPersonJoin`)

- **Actions** - The event handler, an ephemeral piece of code that runs in response to an event.
  - A Javascript function that prints *"hello! welcome $event"*

- **Rules** - Association between a trigger and an action.
  - Associate that when "`newPersonJoin`" is triggered call "`hello.js`".

What is Apache OpenWhisk

More concepts

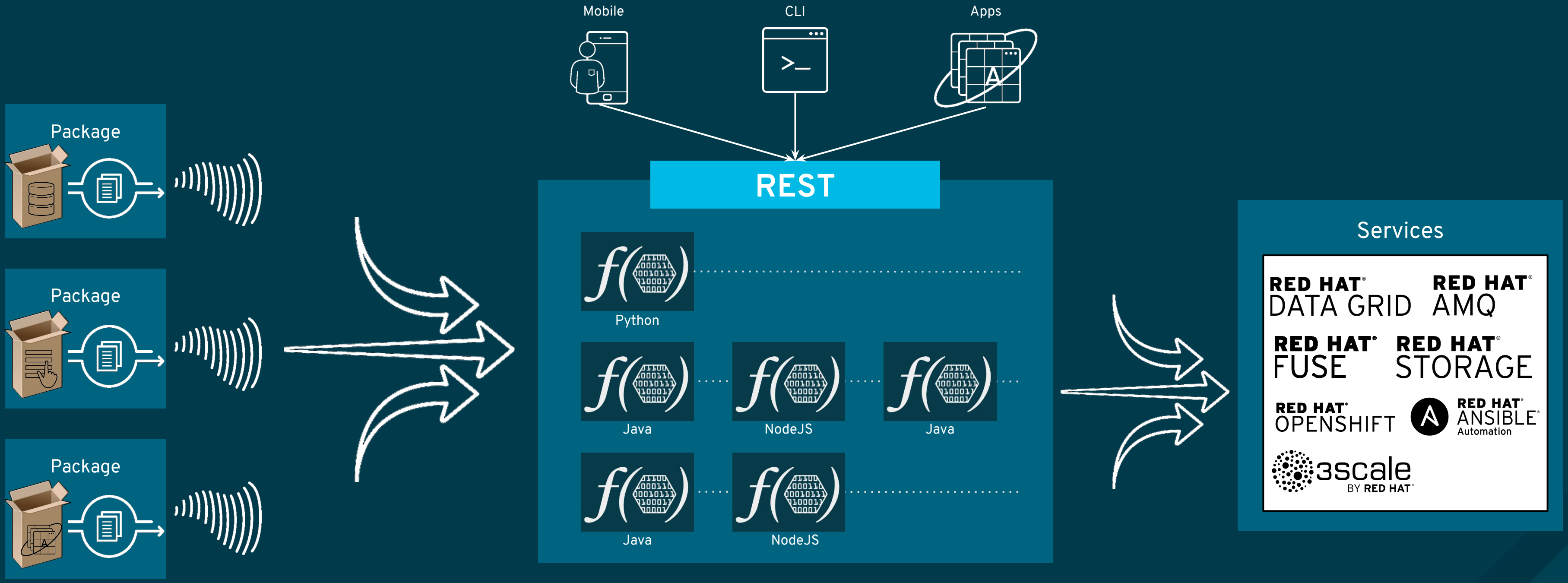


- **Sequences -** Orchestration of a group of functions
  - Function A calls Function B and sends result to Function C.

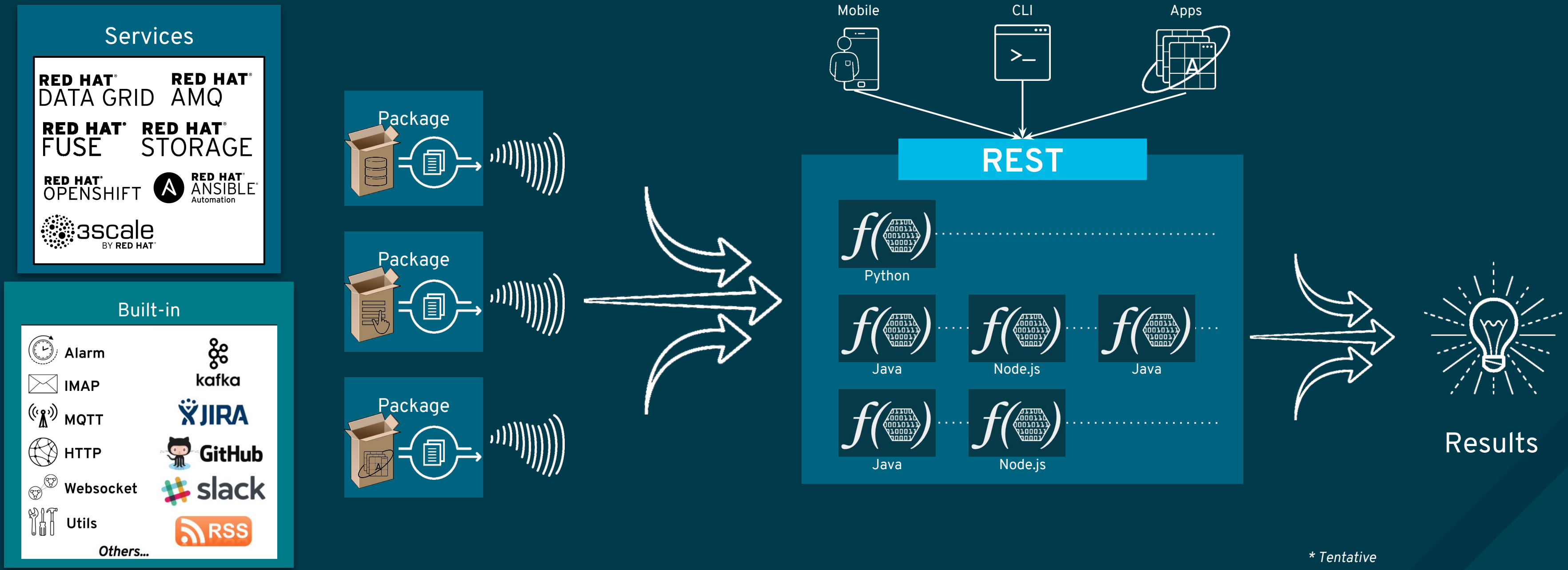
- **Feeds -** Stream of events that can start Triggers through polling, webhooks, etc.
  - *A data grid continuous queries triggers multiple functions*
  - *A clickstream from a web application*

- **Packages** - Bundle a set of actions, feeds and rules.
  - Example: Slack, GitHub, Red Hat Data Grid...

Feeds and Services



Triggers and Services



** Tentative*

Function Runtimes

Where functions are executed

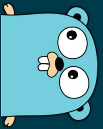
Your code running on:

- Fully tested
- Supported
- Based on:
  - CentOS (Community)
  - RHEL (Product)

python™
Java™
node
JS®

More runtimes that *we can* support:

- Go, Swift, Rust, Scala



APACHE OpenWhisk™

- Donated to Apache from IBM
  - Red Hat, IBM, Adobe…
- Currently incubating in Apache
- Runs anywhere
- No prescriptive platform
- Vibrant community

RED HAT® OPENSHIFT CLOUD FUNCTIONS

- Enterprise ready
- Optimized for **OpenShift**
- Integrations with Red Hat Portofolio
- Repackaged with **fully-supported** runtimes
- Available in OpenShift Online** and OCP*
- Dev tools with Che support

Public ***Hybrid*** ***Private***

* Dev-preview for Red Hat Summit 2018
** Tech-preview November 2018

OpenShift Cloud Functions

Versus the competition...

It's **your** FaaS

Custom memory & timeout limits

- More flexible than cloud providers.

Run **on-premise** or **any cloud**

Available on OpenShift Online.

Local development environment

- Support for Minishift

Enterprise ready

Robust security and authentication

Fully tested, patched and supported

Integrated monitoring interface*

Repackaged runtimes (CentOS/RHEL)

Supported OpenShift Online & OCP

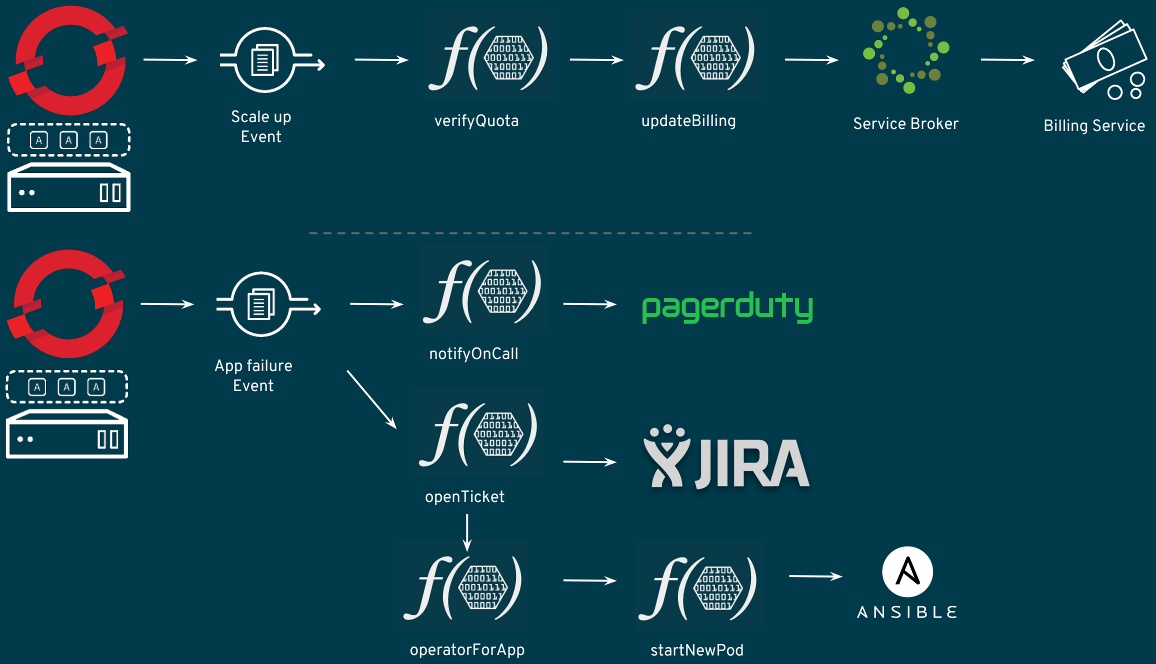
Dev tools with Che support

** Roadmap*

Serverless Use Cases

Serverless use cases

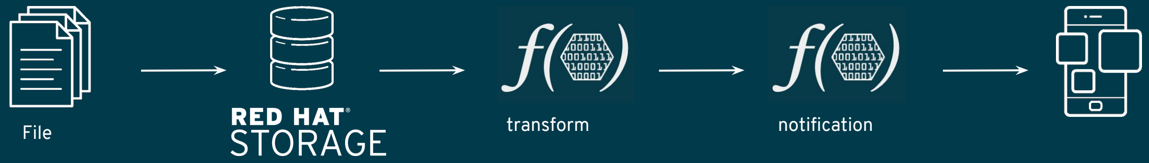
OpenShift event monitoring



Serverless use cases

Storage

File System / S3 Events



File

RED HAT® STORAGE

transform

notification

Machine Learning

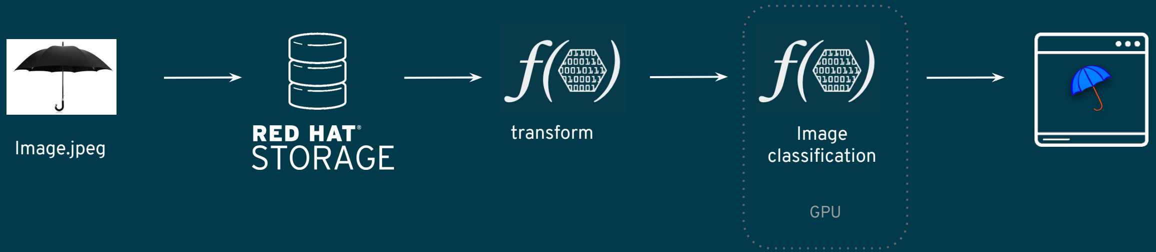


Image.jpeg

RED HAT® STORAGE

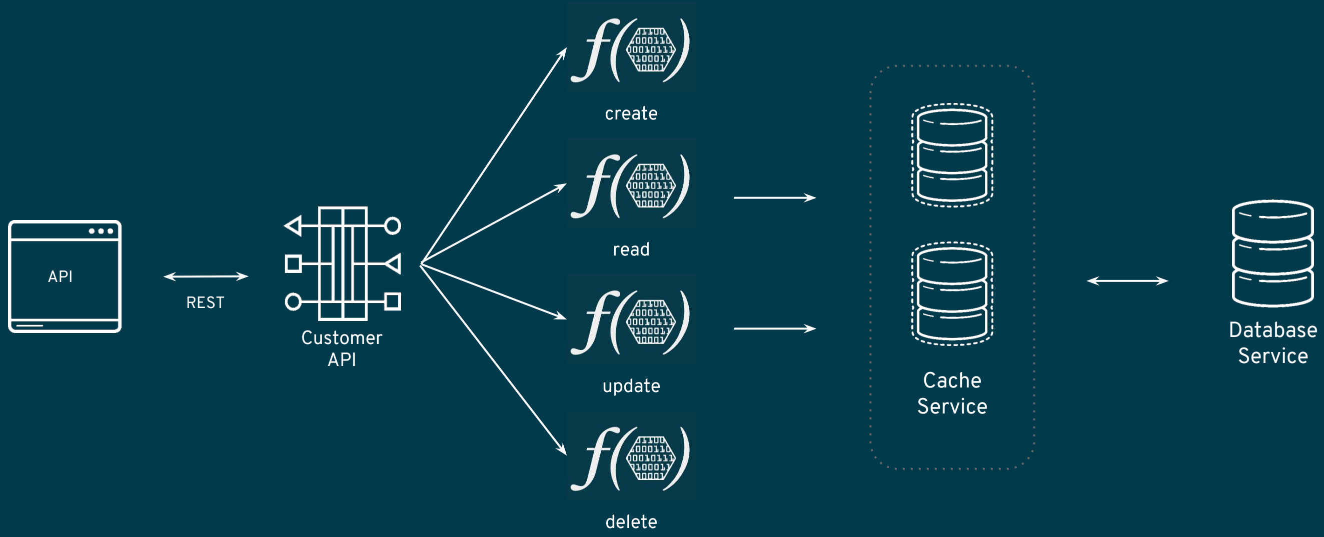
transform

Image classification

GPU

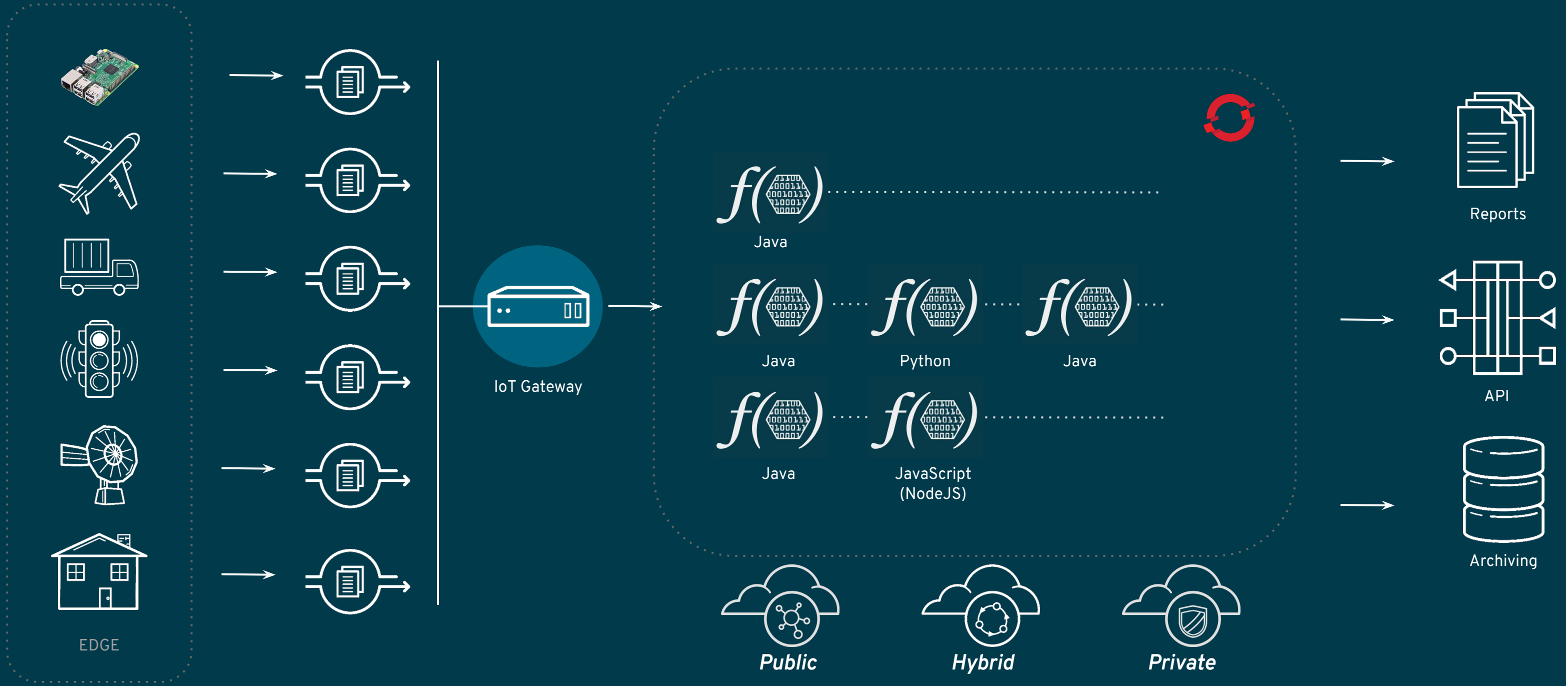
Serverless use cases

Web APIs



Serverless use cases

IoT and Sensors

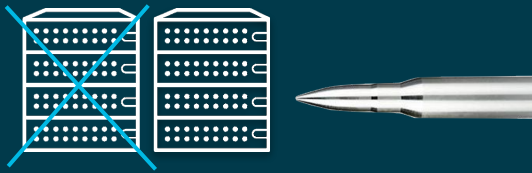


Serverless use cases

Other common use cases...

- Processing web hooks
- Scheduled tasks (a la cron)
- Data transformation
- Mobile image manipulation (compression, conversion, and so on)
- Voice packet to JSON transformation (Alexa, Cortana, and so on)
- Mobile video analysis (frame-grabbing)
- PDF generation
- Mobile/MBaaS /single-page apps
- Chat bots

When not to use serverless

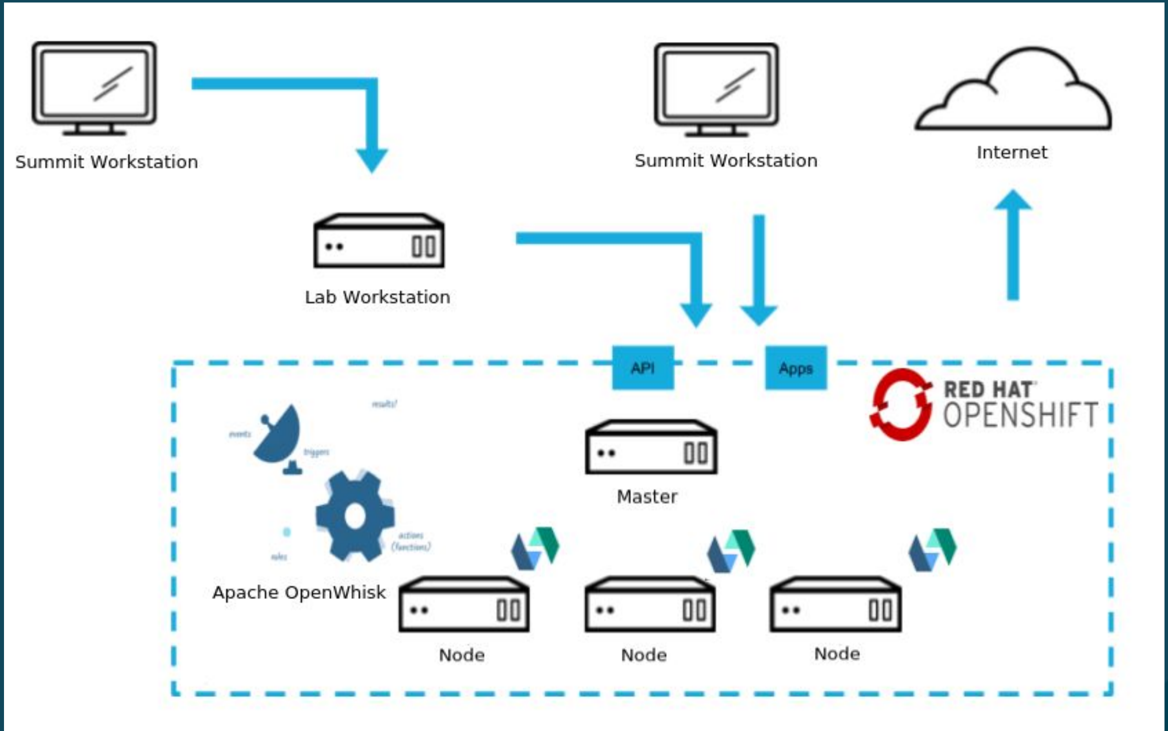


- *Real-time, ultra-low latency applications*
- *Long running tasks that can't be split into steps*
- *Advanced or complex observability and monitoring requirements*
- *Memory or CPU requirements are very demanding and specific*
- *Can't deal with cold-start...*

CNCF, CloudEvents and Serverless

Serverless Hands on

http://bit.ly/roadshow-serverless-paas-lab



RED HAT CLOUD FUNCTIONS

APACHE OPENWHISK + OPENSHIFT = RED HAT® CLOUD FUNCTIONS Technical Preview

ENTERPRISE GRADE HYBRID, MULTI-CLOUD SERVERLESS

LEARN MORE :

https://learn.openshift.com/serverless/

"Serverless data center"

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos