浅谈 FaaS 与 DevOps

Press Space for next page →

张海立

KubeSphere Ambassador, CNCF OpenFunction TOC Member

驭势科技 UISEE© 云平台研发总监，开源爱好者

云原生专注领域: Kubernetes, DevOps, FaaS, Observability, Service Mesh

zhanghaili

webup

zhanghaili0610

haili.zhang@uisee.com

ServiceUP · 语雀

如何把一个 GitHub 仓库私有化到本地 GitLab？

从一个假想的 DevOps 环境迁移的场景说起

这有何难！

- GitLab 本身就支持从 GitHub 导入并创建项目

那么，这么多 GITHUB APP 和 BOT 怎么迁移？

- 有的可以根据 Issue 中的文字自动打标签
- 有的可以根据已合并的 PR 信息自动生成 Release Note
- 有的可以通过评论修改 `README.md` 文件来添加协作者信息
- 有的可以定时检查各个 Issue 或 PR 是否长期无人处理并打上过期标签

这也没什么难的，DIY 一套呗？！

- 你可能需要创建一个或多个 微服务
- 你可能需要熟悉 GitLab Webhook 和 REST API 来打通服务的 输入和输出
- 你可能还需要一个定时任务或者异步框架来处理 异步 的任务流
- 你可能还需要关心一下这个/些服务的 运行开销，不处理任务时候能不常驻吗
- 甚至，如果维护这个/些服务的小伙伴离开团队了，另一个你会使用相同 开发语言 来持续迭代吗

FaaS, Function as a ~~Service~~ Silver bullet?

让我们从功能角度来看 FaaS 可以如何赋能 DevOps

Serverless vs. FaaS

人们常把 Serverless 和 FaaS 混为一谈

"Serverless computing = FaaS + BaaS." [1]

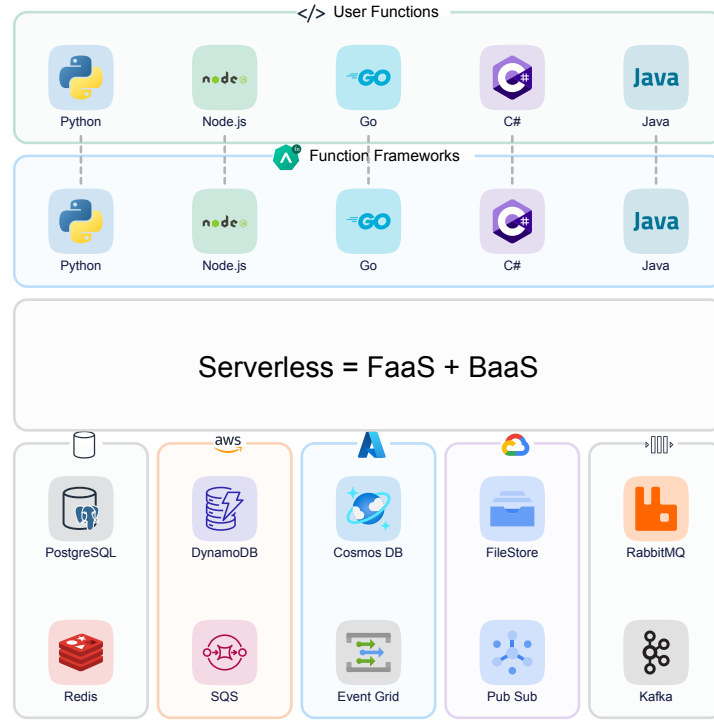
- 云函数是函数计算的主体
- 各类后端服务是函数实现业务的重要依托

🔒 云商通过提供“托管的”（Hosted）函数计算框架（FaaS）和各类中间件服务（BaaS），把开发者锁定在托管的云平台上。

🤔 这里有个潜在的 本地化及迁移 的问题，谁来填补各云商平台 BaaS 服务的接口差异?

---

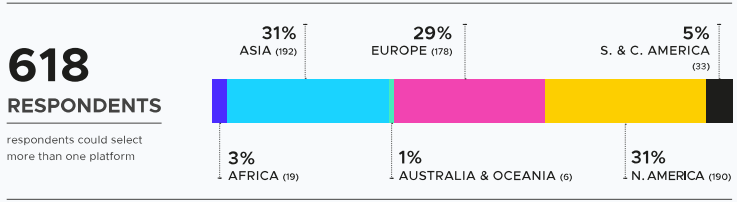
1. Cloud Programming Simplified: A Berkeley View on Serverless Computing. Feb 10, 2019.



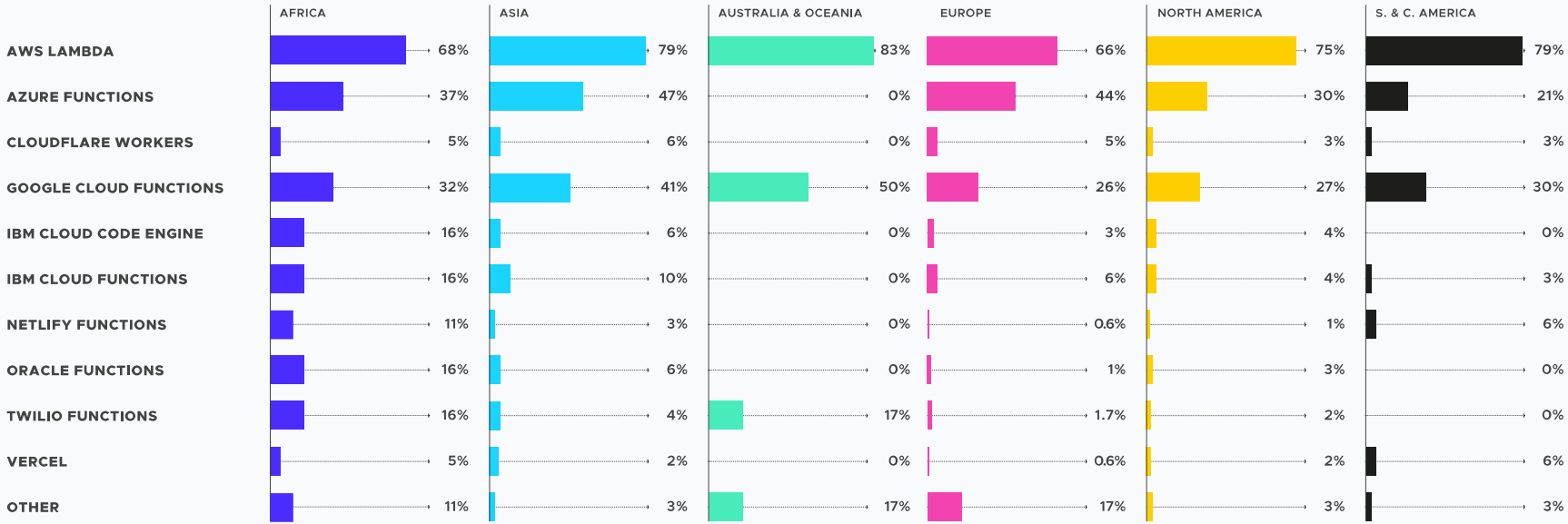
Serverless 平台的战争?

CNCF Annual Survey 2021

"使用 Serverless 技术的受访者，75% 选择了托管平台"



IF YOUR ORGANIZATION IS USING SERVERLESS VIA A 'HOSTED PLATFORM', WHICH HOSTED SERVERLESS PLATFORMS DO YOU USE?



Open? Function!

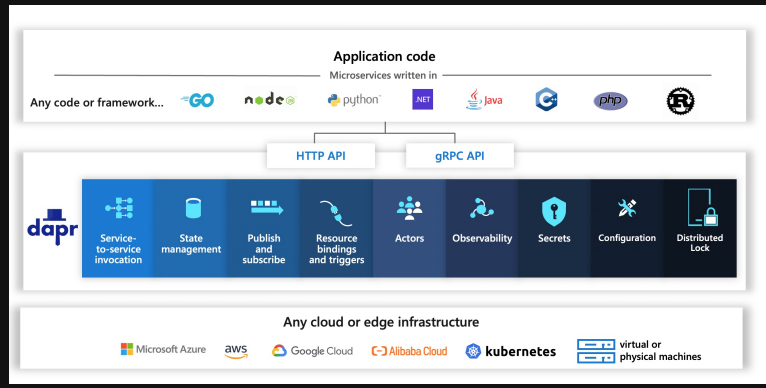
一个 CNCF Sandbox 函数计算平台项目

OpenFunction 的云原生特性：

- “云无关”：与云商的 BaaS 松耦合
  - 通过 Dapr，简化了对于云商 BaaS 的集成
- 同时支持同步和异步函数
  - 同步函数 Kubernetes Gateway API 支持的高级函数入口和流量管理
  - 异步函数可以直接从事件源消耗事件
- 支持直接从函数源码生成符合 OCI 镜像
  - 基于 Cloud Native Buildpacks 生态实现
- 在 0 和 N 之间灵活的自动缩放
  - 充分利用 KEDA、Knative

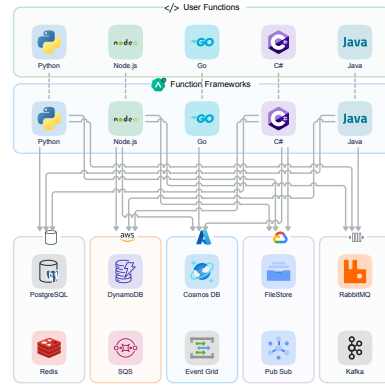
函数？应用！运行时 🎩

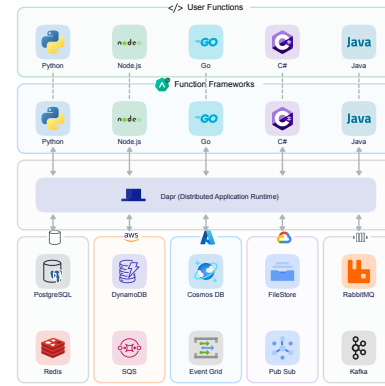
借助 Dapr 来集成各云商平台的 BaaS 服务



Dapr 是一个可移植的、事件驱动的运行时，它使任何开发人员能够轻松构建出弹性的、无状态和有状态的应用程序，并可运行在云平台或边缘计算中，它同时也支持多种编程语言和开发框架。

Dapr 是和平台无关的，这意味着您可以在 Kubernetes 集群、本地或者其它集成 Dapr 的托管环境中运行应用程序。 这使得您能够在云平台和边缘计算中运行微服务应用。





主人，未安装Flash插件，暂时无法观看直播，您可以...

下载Flash插件

事件驱动的伸缩器

Serverless 需要能伸缩到零

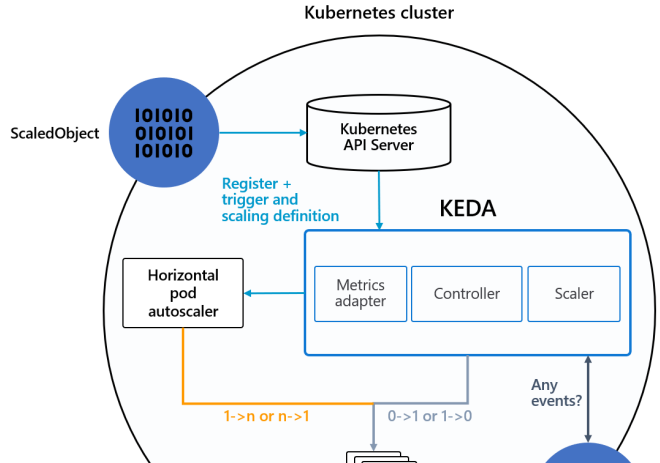
KEDA 是一个基于 Kubernetes 的事件驱动自动缩放器。使用 KEDA，您可以根据需要处理的事件数量来驱动 Kubernetes 中任何容器的扩展。

- KEDA 是一个单一用途的轻量级组件，可以添加到任何 Kubernetes 集群中。
- KEDA 与 HPA（Horizontal Pod Autoscaler）等标准 Kubernetes 组件一起工作，可以在不覆盖或复制的情况下扩展功能。
- 使用 KEDA，您可以明确映射您想要使用事件驱动比例的应用程序，这使得 KEDA 成为与任意数量的任何其他 Kubernetes 应用程序或框架一起运行的灵活且安全的选项。

KEDA

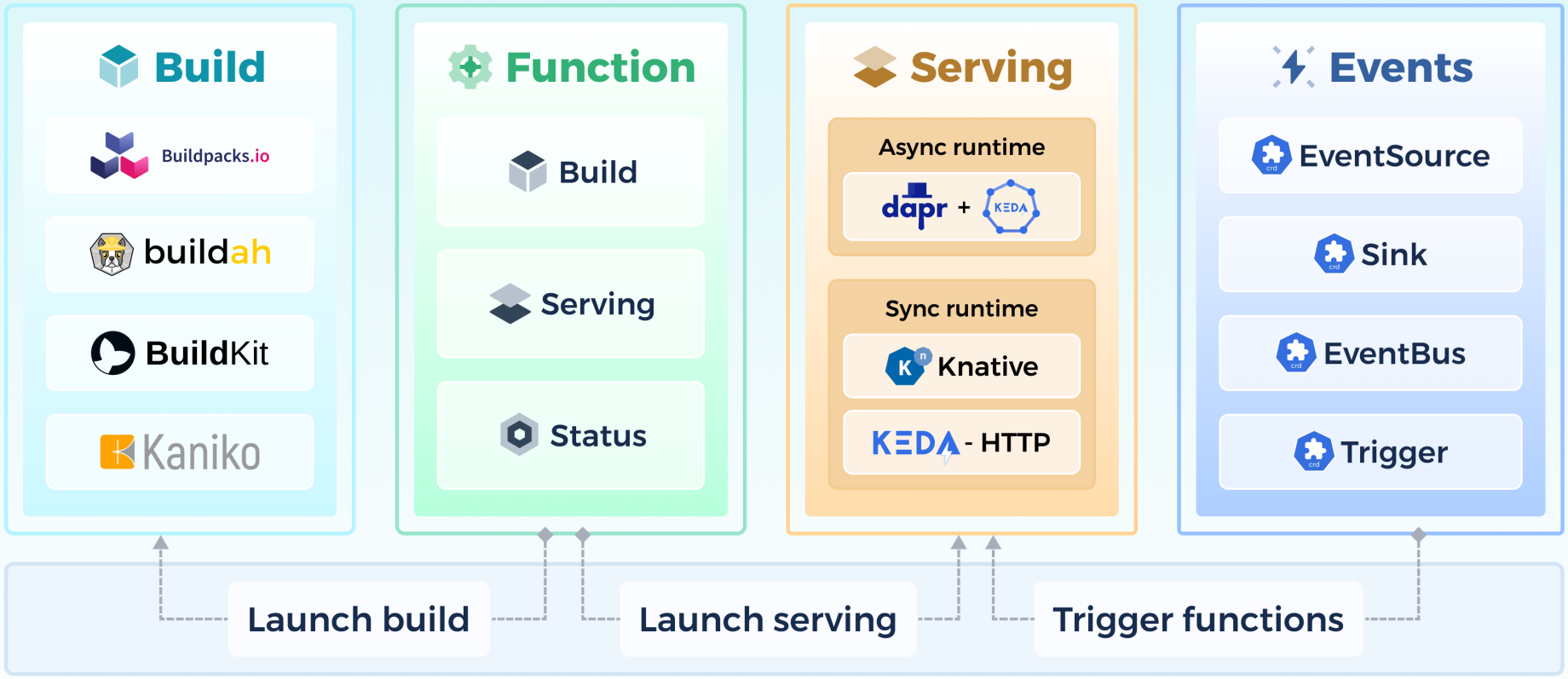
Architecture

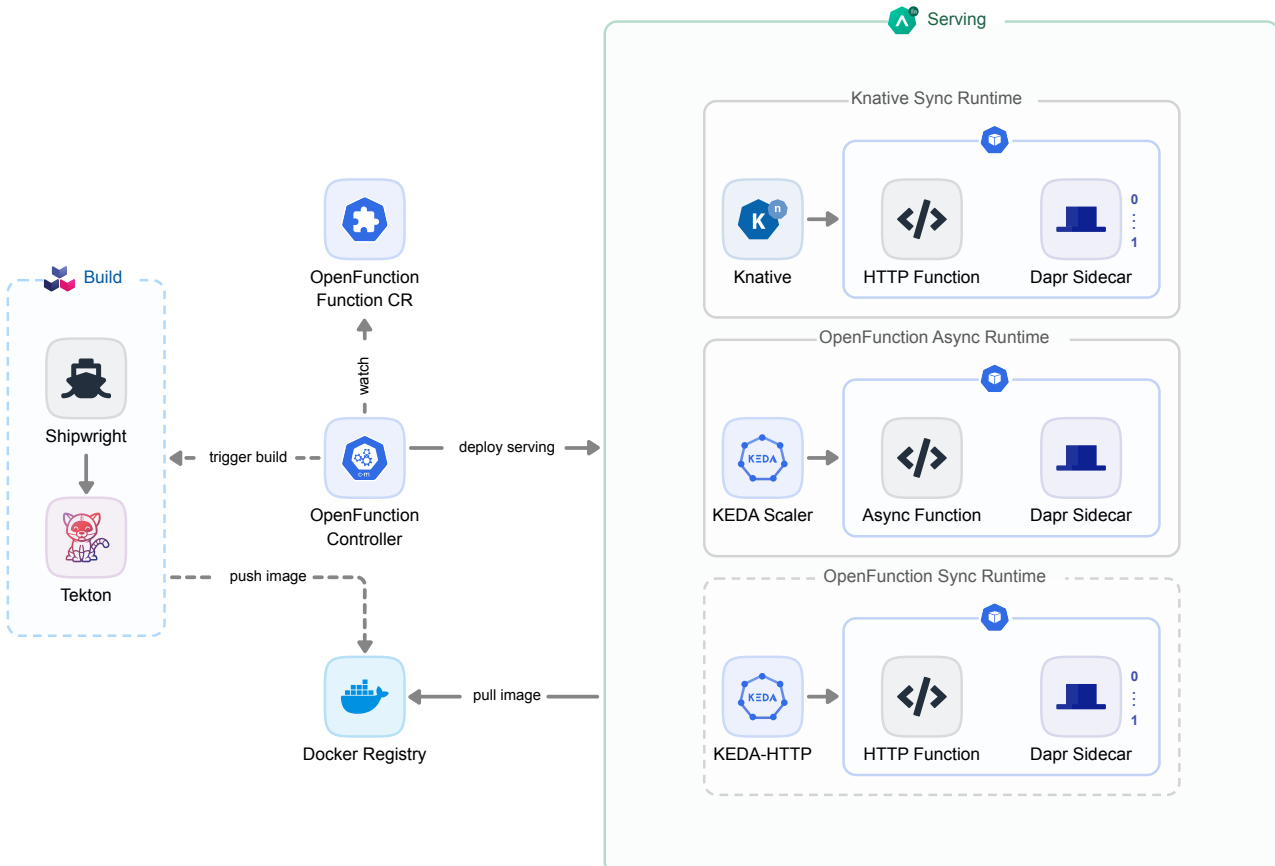
The diagram below shows how KEDA works in conjunction with the Kubernetes Horizontal Pod Autoscaler, external event sources, and Kubernetes' etcd data store:



Cloud-Native FaaS Platform

OPEN FUNCTION




Serving
Knative Sync Runtime
Knative
HTTP Function
Dapr Sidecar
0
1
OpenFunction Async Runtime
KEDA Scaler
Async Function
Dapr Sidecar
OpenFunction Sync Runtime
KEDA-HTTP
HTTP Function
Dapr Sidecar
0
1
OpenFunction
Function CR
watch
Build
Shipwright
Tekton
trigger build
OpenFunction
Controller
deploy serving
push image
pull image
Docker Registry

举几个 Node.js 函数的例子 

OpenFunction 现已支持 Go, Node.js, Python, Java, .Net 等多种语言

一个 Node.js 同步函数

INDEX.MJS

```
// Standard Express style HTTP sync function
export const tryKnative = (req, res) => {
  res.send(`Hello, ${req.query.u || 'World'}!`);
};
```

🔍 参见 Express 的 request 和 response 对象的使用

PACKAGE.JSON

```
{
  "main": "index.mjs",
  "scripts": {
    "start": "functions-framework --target=tryKnative"
  },
  "dependencies": {
    "@openfunction/functions-framework": "^0.6.0"
  }
}
```

FUNCTION CR (RAW MANIFEST)

```
apiVersion: core.openfunction.io/v1beta1
kind: Function
metadata:
  name: sample-node-knative
spec:
  version: v2.0.0
  image: '<image-repo>/<image-name>:<image-tag>'
  build:
    builder: openfunction/builder-node:v2-16.15
    env:
      FUNC_NAME: tryKnative
    srcRepo:
      url: https://github.com/OpenFunction/samples.git
      sourceSubPath: functions/async/mqtt-io-node
      revision: main
  # app port default to "8080"
```

🔍 参见 OpenFunction Function CRD 定义

一个 Node.js 异步函数

INDEX.MJS

```
// Standard Express style HTTP sync function
export const tryKnative = (req, res) => {
  res.send(`Hello, ${req.query.u || 'World'}!`);
};

// Async function
export const tryAsync = (ctx, data) => {
  // Function Framework received and parsed "data"
  console.log('Data received: %o', data);
  // Use "send" utility to broadcast data to outputs
  ctx.send(data);
};
```

💡 多个函数可以被放在同一个 JS 文件中，并在 CR 中动态指定

FUNCTION CR (RAW MANIFEST)

```
apiVersion: core.openfunction.io/v1beta1
kind: Function
metadata:
  name: sample-node-async
spec:
  version: v2.0.0
  image: '<image-repo>/<image-name>:<tag>'
  serving:
    # default to knative
    runtime: async
    annotations:
      # default to "grpc"
      dapr.io/app-protocol: http
    template:
      containers:
        - name: function
```

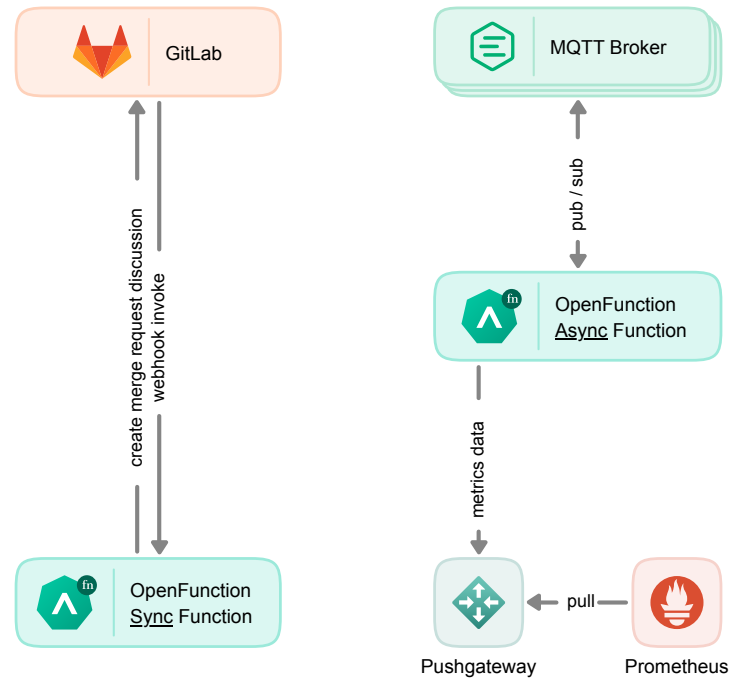
🔍 OpenFunction 函数定义中的输入输出部分：可以直接引用 Dapr Components 的元数据定义

OpenFunction 案例探索

让我们看几个现实可用的 FaaS 赋能场景

同步：GITLAB WEBHOOK ➡️ CHATOPS

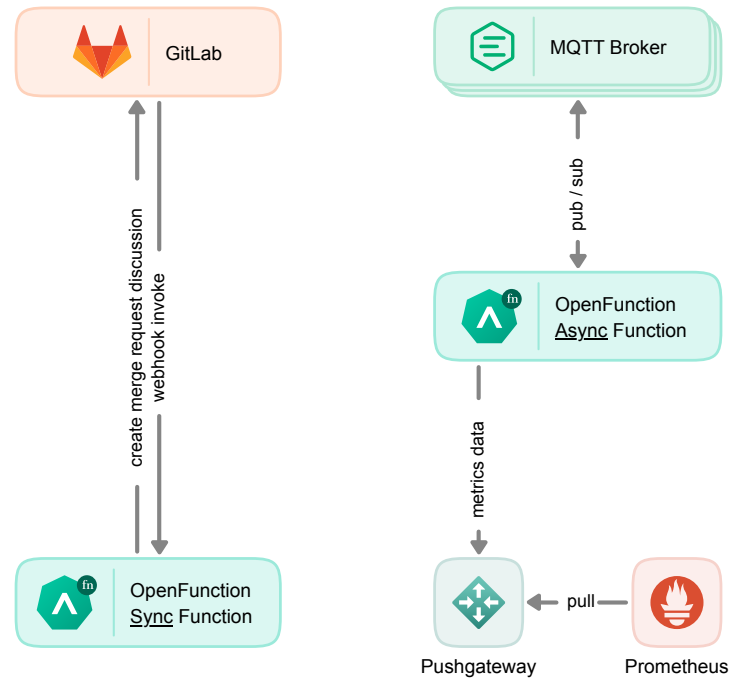
```
import (
  // ...
  "github.com/OpenFunction/functions-framework-go
  "github.com/xanzy/go-gitlab"
)

var GitlabClient *Client

func init() {
  functions.HTTP("HelloWorld", SendBeginUnittest,
  GitlabClient, _ = gitlab.NewClient("4g9gZCsex36
}

func SendBeginUnittest(w http.ResponseWriter, r *
  // ...
  projectMergeRequestList, _, err := GitlabClient
  mergeRequest := projectMergeRequestList[0]

  msg := "大佬又提交代码了!! 666666666"
  mergeRequestDiscussion := &CreateMergeRequestDi
  res, _, err := GitlabClient.Discussions.CreateM
}
```



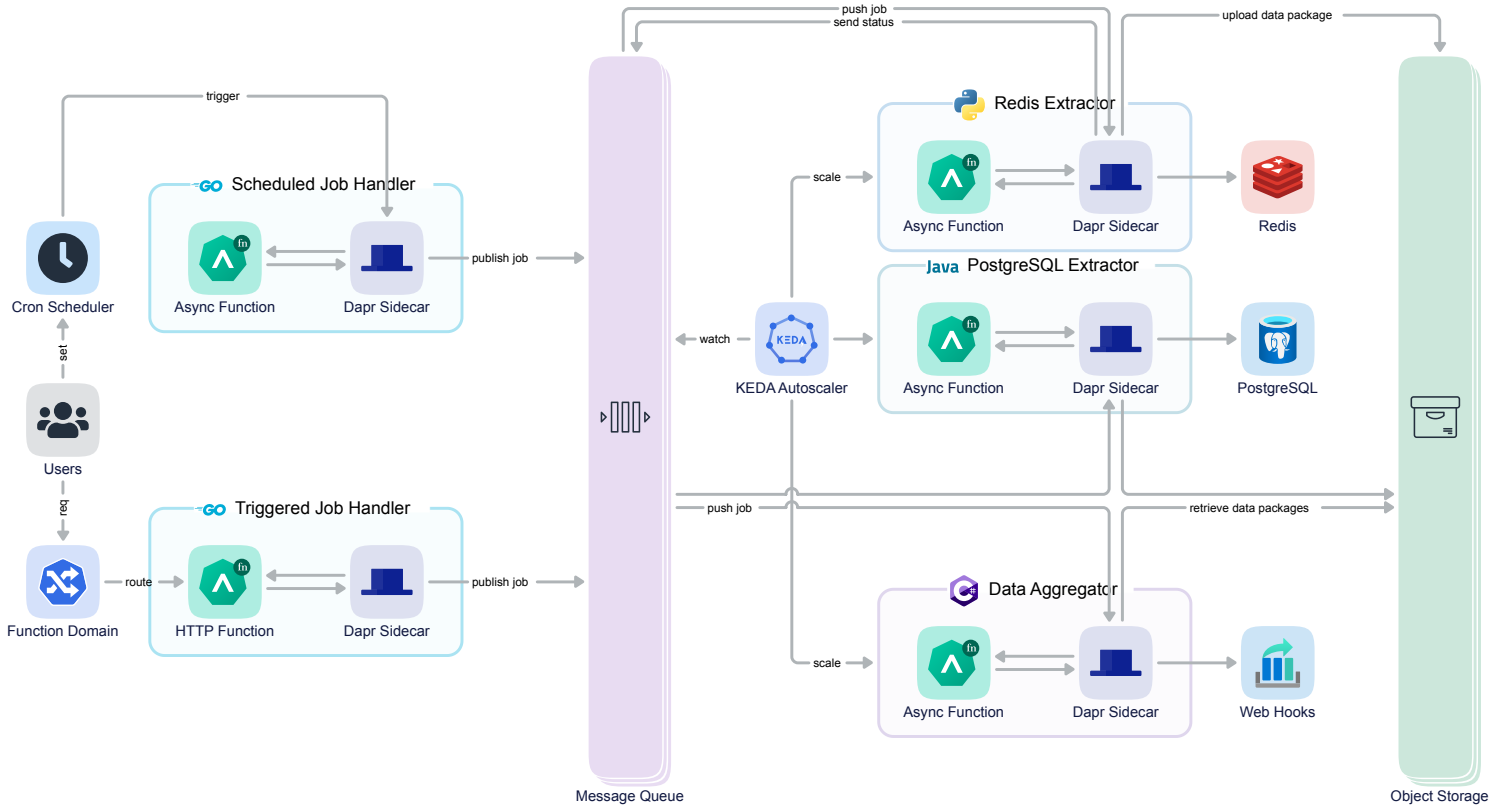
onemodel

异步：消息队列实时数据 ➡️ PROMETHEUS 指标

```
import (
  ofctx "github.com/OpenFunction/functions-framew
  "github.com/prometheus/client_golang/prometheus
)

var speedGauge prometheus.Gauge

func init() {
  speedGauge = prometheus.NewGauge(prometheus.Gau
    Namespace: namespace,
    Subsystem: subsystem,
    Name:      "speed",
    Help:      "vehicle speed",
  })
}

func trackingHandler(ctx ofctx.Context, vehicleTr
  // Helper to push data to gauge
  setAndPushGauge(speedGauge, vehicleTracking.Par
  return ctx.ReturnOnSuccess(), nil
}
```



onemodel


push job
send status
upload data package
trigger
Scheduled Job Handler
Redis Extractor
scale
Async Function
Dapr Sidecar
Redis
Cron Scheduler
publish job
Java PostgreSQL Extractor
watch
KEDA Autoscaler
PostgreSQL
set
Users
req
Triggered Job Handler
push job
retrieve data packages
Data Aggregator
route
Function Domain
HTTP Function
Web Hooks
Message Queue
Object Storage

Announcing OpenFunction 0.8.0: Speed up function launching with Dapr Proxy

Friday, October 28, 2022

One of the unique features of OpenFunction is its simple integration with various backend services (BaaS) through Dapr. Currently, OpenFunction supports Dapr pub/sub and bindings building blocks, and more will be added in the future.

In OpenFunction v0.7.0 and versions prior to v0.7.0, OpenFunction integrates with BaaS by injecting a dapr sidecar container into each function instance's pod, which leads to the following problems：

- The entire function instance's launch time is slowed down by the launching of the dapr sidecar container.
- The dapr sidecar container may consume more resources than the function container itself.

Learn More

Discord · Slack

OpenFunction · Node.js Functions Framework