Getting Beyond FaaS
(The PLONK Stack for Kubernetes Developers)

Alex Ellis, Founder @ OpenFaaS Ltd, CNCF Ambassador
FaaS and Furious by Forrest Brazeal

CONTAINERS

SERVERLESS
Agenda

- Not so FaaS?
- Cloud Native Stack - PLONK
- PLONK Stack Projects
- Case studies
- Over to you

@openfaas / @alexellisuk
Monoliths (the good ole days)

3-tier monolith
Cloud Native / Serverless Architecture

1. Authentication Service
2. Product Database
3. Client (browser)
4. Search Function
5. Purchase Database
Chief Kubernetes Officer (CKO)

I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.

YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE. JUST PUT IT IN CONTAINERS.

YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS. KUBERNETES.
Kubernetes for devs

• Declarative API
• Highly extensible
• Built to scale
• Rich ecosystem

For business

• Open governance
• Portable
• Used in production
• 100+ ISVs
What is PLONK?

Introducing the PLONK Stack for Cloud Native Developers

You've heard of LAMP, JAM, and MEAN, but what is the PLONK stack? And why should you be considering it for your Cloud Native Applications?
What is the PLONK Stack?

- Prometheus
- Linux/Linkerd*
- OpenFaaS
- NATS*
- Kubernetes

Cloud Native Landscape
How to build a Serverless Single Page App

Alex outlines an architecture for building a Single Page App (SPA) with Serverless Functions using Vue.js for the front-end, Postgres for storage, Go for the backend and OpenFaaS with Kubernetes for a resilient scalable compute platform.
# LEADERBOARD

<table>
<thead>
<tr>
<th>Login</th>
<th>↓ issue Comments</th>
<th>issues opened</th>
</tr>
</thead>
<tbody>
<tr>
<td>alexellis</td>
<td>1932</td>
<td>120</td>
</tr>
<tr>
<td>martindekov</td>
<td>200</td>
<td>11</td>
</tr>
<tr>
<td>LucasRoesler</td>
<td>195</td>
<td>34</td>
</tr>
<tr>
<td>derek[bot]</td>
<td>170</td>
<td>0</td>
</tr>
<tr>
<td>matipan</td>
<td>111</td>
<td>7</td>
</tr>
<tr>
<td>burton</td>
<td>98</td>
<td>3</td>
</tr>
<tr>
<td>rgee0</td>
<td>89</td>
<td>6</td>
</tr>
</tbody>
</table>
PLONK applied (Single Page App)
P - Prometheus

- Time series database
- Observe system
- Ask questions

```plaintext
rate(
    http_requests_total[1m]
    {status="200"}
)
```
P - Prometheus

rate(gateway_function_invocation_total[1m])

Graph

Execute

- Insert metric at cursor
Prometheus - Prometheus

Alerts

Show annotations

/etc/prometheus/alert.rules.yml > openfaas

APIHighInvocationRate (1 active)

alert: APIHighInvocationRate
expr: sum
  by(function_name) {rate(gateway_function_invocation_total{code="200"}[10s])}
> 5
for: 5s
labels:
  service: gateway
  severity: major
annotations:
  description: High invocation total on "{{labels.function_name}}"
  summary: High invocation total on "{{labels.function_name}}"

<table>
<thead>
<tr>
<th>Labels</th>
<th>State</th>
<th>Active Since</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertname=&quot;APIHighInvocationRate&quot; function_name=&quot;nodeinfo&quot; service=&quot;gateway&quot; severity=&quot;major&quot;</td>
<td>FIRING</td>
<td>2019-09-12 10:38:38.324404523 +0000 UTC</td>
<td>17.4</td>
</tr>
</tbody>
</table>

*Optum end-user*
P - Prometheus in OpenFaaS
P - Get Prometheus

• Add a /metrics endpoint
• HTTP / custom metrics
• Trigger alerts

Powers the auto-scaling in OpenFaaS*

CNCF: graduated status
L - Linkerd - service mesh*

- Traffic shifting
- Mutual TLS
- Insights
0 - OpenFaaS - Secure, Simple, Serverless

- 2016 genesis
- Values
- Developers-first
- Operator-friendly
- Community-centric
- Hosted by OpenFaaS Ltd

19.6k stars
250 contributors
2k Slack
<table>
<thead>
<tr>
<th>Workload</th>
<th>Node.js Function</th>
<th>Node.js Function</th>
<th>Node.js Microservice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packaging</td>
<td>Zip File</td>
<td>Zip File</td>
<td>?</td>
</tr>
<tr>
<td>Platform</td>
<td>AWS Lambda</td>
<td>Google Cloud Functions</td>
<td>?</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>AWS</td>
<td>GCP</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>
0 - OpenFaaS (Serverless 2.0)

**Any code**
Build both microservices & functions in any language. Legacy code and binaries.

**Anywhere**
Avoid lock-in through the use of Docker. Run on any public or private cloud.

**Any scale**
Auto-scale for demand or to zero when idle.
0 - Serverless 2.0 runtime contract
- Docker / OCI image
- HTTP/8080
- Stateless
A Node.js microservice for Kubernetes

Adding a new service (the hard way)
Four pillars of Object Oriented Programming.
• Encapsulation
• Abstraction
• Inheritance
• Polymorphism

Functions Store
An ecosystem for sharing, reusing and collaborating on functions

Templating system
Reduce boilerplate code, share code in the templates store

Functions or Microservices
Deploy existing microservices using Express.js, Sinatra, ASP.NET Core or simplify with functions.
0 - OpenFaaS: template system

```
functions:
  hello-fn:
    lang: node10-express
    handler: ./hello-fn
    image: hello-fn:latest

handler.js
```

- Secrets / environment
- Memory / CPU limits
- Labels / CRD
const axios = require("axios")

module.exports = async (event, context) => {

    let res = await axios.get("https://api.ipify.org?format=json")

    return context
        .status(200)
        .headers({"Content-Type": "application/json"})
        .succeed({"result": res.data, "status":"🌮"});
}
0 - Zero to production in < 5 mins (ketchup)

k3sup app install nginx-ingress
k3sup app install cert-manager
k3sup app install openfaas
k3sup app install openfaas-ingress
   --domain plonk.dev --email webmaster@plonk.dev

faas-cli login --gateway https://plonk.dev
$ faas new --lang node12 ping

module.exports = (event, context) => {
  return {"status": "pong"}
}

handler.js

{"dependencies": {"axios": "^0.19.0"}}

package.json

$ faas up

Gateway

Docker Image

Build

Push

Deploy

Registry
O - OpenFaaS: CRD?

```yaml
faas-cli generate -f express-flag.yml

---
apiVersion: openfaas.com/v1alpha2
kind: Function
metadata:
  name: express-flag
spec:
  name: express-flag
  image: alexellis2/express-flag:latest
  environment:
    debug: "1"
  secrets:
    - api-key

faas-cli generate -f express-flag.yml
  --api=serving.knative.dev/v1
```
Case study - Ed Wilde @ Form3

Likes 😊
• Fast, online, bank transfers
• Platform engineering
• GitHub & Go

Dislikes: 😣
• Change Advisory Board (CAB)
• Jira
1. feature request

2. develop

3. approval

4. deploy
Form3 - PLONK to the rescue!

PLONK replaced Jira 😊
A new contributor Channing Conner of Netflix put together an awesome example repo with the new streaming bash template. It uses ffmpeg to generate webm previews, create contact sheets, and to apply a watermark.
#!/bin/bash

SESSION=`uuidgen`

function clean () {
    rm -rf .tmp/"${SESSION}"*.png
}

if [ ! -d "./tmp" ]; then
    mkdir .tmp
fi

trap clean SIGINT SIGTERM

# Create a screenshot every minute
ffmpeg -i -vf fps=fps=1/60 -f image2 .tmp/"${SESSION}"_%03d.png
montage .tmp/"${SESSION}"_*.png -geometry +0+0 -tile 3x -background transparent png:-
rm -f .tmp/"${SESSION}"_*.png
0 - OpenFaaS event-connectors

- Kafka
- AWS SQS*/SNS
- RabbitMQ
- CRON
- .. and more

```yaml
functions:
  email-customer:
    image: email:1.0
  annotation:
    topic: onboarded
```
Case-study: “VEBA”
Case-study: PLONK + vCenter-connector

*Nvidia end-user
NATS

Use case:
• Messaging
• Streaming
• IoT / Edge

CNCF status: Incubating
N - Invocation without NATS

Case-study: Vision Banco, Paraguay
N - Invocation with NATS

Case-study: Vision Banco, Paraguay
https://gateway/function/create-statement

https://gateway/async-function/create-statement
PLONK Stack users*

*Blinker optional
CI/CD for PLONK

- Jenkins
- GitLab
- GitHub Actions
- Travis/Drone/Circle CI
- Tekton
- Argo
- Weave Flux

OpenFaaS Cloud

https://github.com/openfaas/openfaas-cloud
“Kubernetes is not for developers”

- Highly configurable
- Declarative API
- Eventually consistent

- Sharp edges 😵
- Steep learning curve
<table>
<thead>
<tr>
<th>Plonk Layers</th>
<th>Platform Layer</th>
<th>Infrastructure Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OpenFaaS</strong></td>
<td>OpenFaaS Gateway</td>
<td>Kubernetes</td>
</tr>
<tr>
<td></td>
<td>Prometheus</td>
<td>Docker</td>
</tr>
<tr>
<td></td>
<td>NATS Streaming</td>
<td>Container Registry</td>
</tr>
</tbody>
</table>
Why PLONK is easier

- 📈 Convention over configuration
- ✔️ Sane defaults
- 😎 More maintainable
- 🧑‍💻 Great DX
- 🧑‍💼 Trusted in production
Connect & Learn 🎬 📸

• 👍 Join us: https://slack.openfaas.io/

• 🎞️ Try the free OpenFaaS workshop
  https://github.com/openfaas/workshop/

• 🌮 KubeCon sessions:
  https://www.openfaas.com/blog/kubecon-san-diego/