Containerising Functions using Docker and OpenShift

Josef Spillner <josef.spillner@zhaw.ch>
Service Prototyping Lab (blog.zhaw.ch/icclab)

Apr 20, 2017 | Microservices Zürich Meetup @ RHCH
The FaaS Space

- AWS Lambda
- OpenWhisk
- Webtask.io
- Hook.io
- PyWren
- Zappa
- Serverless Framework
- Step Functions
- Docker-LimbCI
- Effe
- Fission
- Picasso
- Kubeless
- Funktion
- Lever OS
- X-Ray
The FaaS Space

Function-as-a-Service offerings in greater detail...

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Languages</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Lambda</td>
<td>Node.js, Java, Python / C#</td>
<td>Service</td>
</tr>
<tr>
<td>Google Cloud Functions</td>
<td>Node.js</td>
<td>Service</td>
</tr>
<tr>
<td>IBM/Apache OpenWhisk</td>
<td>Node.js, Swift, Docker* / Python</td>
<td>OSS + Service</td>
</tr>
<tr>
<td>Azure Functions</td>
<td>Node.js, C# / F#, Python, PHP, ...</td>
<td>Service</td>
</tr>
<tr>
<td>Webtask.io</td>
<td>Node.js</td>
<td>OSS + Service</td>
</tr>
<tr>
<td>Hook.io</td>
<td>Node.js, ECMAScript, CoffeeScript</td>
<td>OSS + Service</td>
</tr>
<tr>
<td>Effe</td>
<td>Go</td>
<td>OSS</td>
</tr>
<tr>
<td>OpenLambda</td>
<td>Python</td>
<td>Academic + OSS</td>
</tr>
<tr>
<td>LambCI Docker-Lambda</td>
<td>Node.js</td>
<td>OSS (re-engineered)</td>
</tr>
<tr>
<td>Lever OS</td>
<td>Node.js, Go</td>
<td>OSS</td>
</tr>
<tr>
<td>Fission</td>
<td>Node.js, Python</td>
<td>OSS</td>
</tr>
<tr>
<td>Funktion</td>
<td>Node.js</td>
<td>OSS</td>
</tr>
<tr>
<td>Kubeless</td>
<td>Python</td>
<td>OSS</td>
</tr>
</tbody>
</table>

Trend: Sooner or later → gaps will be filled
FaaS Research Questions

We spend your tax money on figuring out:

• Characteristics (technical, economical)

• Engineering applications (debugging, profiling, autotuning)

• Migrating legacy code bases (static vs. dynamic code analysis)

• Hybrid technology applications, compositions

• Tools to make all that happen
Some Fun at First

```
import boto3
import time

def lambda_handler(event, context):
    time.sleep(context.get_remaining_time_in_millis() / 1000.0 - 1)
    c = boto3.client("lambda")
    c.invoke(FunctionName="worm", Payload="{}")
```

""Lambbackup"

```
$ ./lambbackup list
1 files found with records.
refdata/Apache-2.0

$ ./lambbackup backup .gitignore
Prepared 1 chunks. Backup ...

Updating file list.
Backed up .gitignore.

$ ./lambbackup list
2 files found with records.
refdata/Apache-2.0
.gitignore
```

""Lambda Worms"

```
$ ./lambickup list
1 files found with records.
refdata/Apache-2.0

$ ./lambickup bickup .gitignore
Prepared 1 chunks. Bickup ...

Updating file list.
Bicked up .gitignore.

$ ./lambbackup list
2 files found with records.
refdata/Apache-2.0
.gitignore
```

""Lambbackup"

```
$ ./lambickup list
1 files found with records.
refdata/Apache-2.0

$ ./lambbackup backup .gitignore
Prepared 1 chunks. Backup ...

Updating file list.
Backed up .gitignore.

$ ./lambbackup list
2 files found with records.
refdata/Apache-2.0
.gitignore
```

""Lambda Worms"
Snafu: Swiss Army Knife of Serverless Computing
FaaS in PaaS: Architectural Choice

Choices:

- unprivileged (single-tenant Docker instance)
  - any user hosts functions
- semi-privileged (Open Service Broker API)
  - provider hosts functions (could be external)
- privileged (deeply integrated as Kubernetes/OpenShift resources)
  - provider hosts functions
Demo Scenario

Credits

<table>
<thead>
<tr>
<th>Promotion ID</th>
<th>Expires</th>
<th>Promotion value</th>
<th>Amount remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Trial</td>
<td>11 Mar 2015</td>
<td>$300.00</td>
<td>Expired</td>
</tr>
<tr>
<td>Google Cloud Platform Credit</td>
<td>10 Jan 2016</td>
<td>$5,000.00</td>
<td>Expired</td>
</tr>
<tr>
<td>Sales Credit</td>
<td>12 Apr 2017</td>
<td>$10,000.00</td>
<td>Expired</td>
</tr>
</tbody>
</table>

APPUIO Public Cloud

OpenShift Enterprise

Kubernetes

- Snafu (FaaS Container)
- Snafu Persistent Volume

IBM Bluemix
OpenWhisk
Demo Time - Make it or break it!

[thetalentcode.com]
One More Thing...

sources

AWS

IBM

snafu-import

Snafu

Funktion

targets

Google
OK, Demo Again...
Service Prototyping Lab

Research approach
• ambitious long-term research initiatives
  • cloud-native applications, service tooling, cloud accounting & billing
  • transfer of results into Swiss companies

Publishing approach
• preprint-first - rapid dissemination
• open source, Labsite
• blog posts, events
Future Cloud Applications

Thu, April 27, 18:30 @ Technikum Winti - All technical details about Snafu & Lambada
Further Reading and FaaS Fun

Lama, Lambbackup:
• https://arxiv.org/abs/1701.05945
Podilizer:
• https://arxiv.org/abs/1702.05510
Snafu:
• https://arxiv.org/abs/1703.07562

On arXiv Analytics:

On GitHub:

[github.com/serviceprototypinglab]

https://www.meetup.com/Future-Cloud-Applications/