Serverless Computing with Apache OpenWhisk

Lorna Mitchell, IBM

What is Serverless?

- Code snippet
- Deployed to cloud
- Executed in response to an event
- Scaled on demand
- Costs nothing when idle

Where are the Servers?

The servers are alive and well.

(your function gets containerised, OpenWhisk grabs and runs it on demand in response to the registered triggers)

Does Serverless Solve Real Problems?

Does Serverless Solve Real Problems?

Yes! *

Does Serverless Solve Real Problems?

Yes! *

* maybe not all of them

APIs and Microservices

Serverless is a great fit for Microservices!

Each endpoint (URL and verb combination) is a serverless action

An API Gateway maps routes to actions

Each endpoint is independent

One-Off Endpoints

No need to spin up a whole server just for your:

- Alexa skill
- mailing list/slack team signup
- incoming webhook from GitHub/Twilio/Nexmo/Zapier

Just Enough Backend Code

Mobile and frontend experts often need some server-side code for APIs or secure Auth

No sysadmin skills required

Serverless and Data

When data is atomic, there's so much we can do with our easy-entry, highly scalable serverless platforms.

- Transform data
- React to database changes or incoming data
- Work with small volume, trickling data
- Work with high volume, streaming data
- Handle one-off data transformation or import

Meet Apache OpenWhisk

Meet Apache OpenWhisk

- http://openwhisk.incubator.apache.org/
- Currently an incubator project, working towards graduation
- Contributors from a variety of backgrounds

Supported Technologies

OpenWhisk is Open Source and extensible. It supports:

- Java
- NodeJS
- PHP
- Python
- Swift

Docker containers can also be deployed and run.

Getting Started With Apache OpenWhisk

@lornajane

OpenWhisk Vocabulary

- **trigger** an event, such as an incoming HTTP request
- **rule** map a trigger to an action
- **action** a function, optionally with parameters
- **package** collect actions and parameters together
- **sequence** more than one action in a row
- **cold start** time to run a fresh action

Hello World in JS

A `main` function, with a single parameter:

```
exports.main = function(args) {
    return({"body": "Hello, World!"});
};
```

Function must return an object or a Promise

Parameters

Parameters can be set:

- at deploy time
- at run time (including by the events that trigger them)

CNCF project to standardise event params:
https://github.com/cloudevents/spec

Deploying to OpenWhisk

Deploy code:

```
wsk package update demo
zip hello.zip index.js
wsk action update --kind nodejs:6 demo/hello1 hello.zip
```

Then run it:

```
wsk action invoke --result demo/hello1
```

Web-Enabled Actions

Deploy code:

```
wsk package update demo
zip hello.zip index.js
wsk action update --kind nodejs:6 --web true demo/hello1 hello.zip
```

Then get the URL and curl it:

```
wsk action get --url demo/hello1
curl https://172.17.0.1/api/v1/web/guest/demo/hello1
```

More About Packages

Packages allow us to:

- group actions together
- set parameters on packages, used by all actions

Sequences can include actions from other packages

Built In Packages

There are some packages and actions available by default:

- `utils`
- `cloudant`
- `github`
- `slack`
- `websocket`
- `samples`

Using Built-In Actions

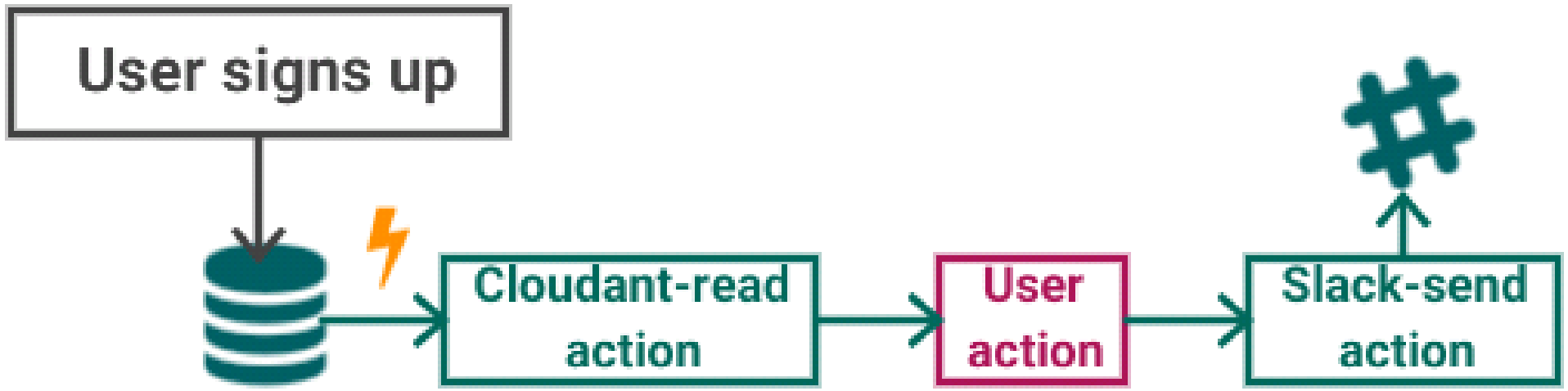
As an example: there's a `wordCount` action in the `samples` package:

```
wsk -i action invoke --result /whisk.system/samples/wordCount \
    -p payload "A good way to ruin a fine walk"
```

Returns: 8

Using Sequences

For example: notify Slack when a user registers



Serverless and Apache OpenWhisk

Resources

- http://openwhisk.incubator.apache.org/
- https://www.ibm.com/cloud/functions
- https://lornajane.net

For more: "Serverless Microservices are the New Black" tomorrow afternoon