Fun With Serverless JS

Lorna Mitchell, IBM

The Serverless Revolution

The big secret is: there ARE servers!

What is Serverless?

Backend functions, deployed to the cloud, scaling on demand. Pay as you go.

The Serverless Revolution

FaaS: Functions as a Service

Developer focus:

- the outputs
- the inputs
- the logic in between

Charges are usually per GBsec

When To Go Serverless

- For occasional server needs (contact form on static site)
- For very variable traffic levels (millions of browsers requesting update)
- To provide extra compute resource without extending existing platform (classic example: PDF generation)

Serverless Platforms

- Amazon Lambda
- IBM Cloud Functions (aka OpenWhisk)
- Twilio Functions
- Azure Functions
- Google Cloud Functions
- Iron Functions
- .... and more. Every day there are more.

Getting Started

Amazon Lambda

- Install `awscli` command line tool (there is also a web interface)
- Set up permissions via IAM and then use `aws configure` to get that set up

- Write some code, then zip it (e.g. `index.js` -> `hello.zip`)

Amazon Lambda

Create your lambda function by supplying the zip file and some options:

```
aws lambda create-function \
        --function-name hello1 \
        --runtime nodejs6.10 \
        --role "arn:aws:iam::283476131276:role/service-role/Alexa"
        --description "A demo first Lambda function" \
        --handler index.handler \
        --zip-file fileb://hello.zip
```

Amazon Lambda

(if you want to edit your code and redeploy it)

```
aws lambda update-function-code \
        --function-name hello1 \
        --zip-file fileb://hello.zip
```

Run your lambda function:

```
aws lambda invoke --function-name hello1 output.txt
```

Hello World Lambda

```
exports.handler = function(event, context) {
    context.succeed("Hello, World!");
};
```

Hello World OpenWhisk

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

IBM Cloud Functions

Get the `wsk` CLI tool or the `bx` tool with `wsk` plugin, then log in

Zip and deploy/update your code like this:

```
zip hello.zip index.js

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

`demo` is the package name

Bluemix OpenWhisk

Run your action from the CLI:

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

Enable web access and web request your action:

```
wsk action update demo/hello1 --web true

curl https://openwhisk.ng.bluemix.net/api/v1/web/ \
        Lorna.Mitchell_Working/demo/hello1.json
```

Serverless + HTTP

FaaS + HTTP = Microservices!

The Fun Part

Alexa: Amazon Echo

You speak, the device sends the sound to the cloud and speaks back the response

Invoking Alexa

Alexa, ask **[blah]** to/about **[blah]** with/on **[blah]**

Invoking Alexa

Invoking Alexa

Invoking Alexa

Example: Project Codename

Existing npm library:
https://www.npmjs.com/package/project-name-generator

Skill code on GitHub:
https://github.com/lornajane/alexa-project-codename

"Alexa, ask Project Codename for a new project name"

Project Codename: Code

```
function main(args) {
    var generate = require('project-name-generator');
    var random = generate().spaced;
    var response = {
        "version": "1.0",
        "response" :{ "shouldEndSession": true,
            "outputSpeech": {
                "type": "PlainText",
                "text": "project codename. " + random } }
    }
    return(response);
}
exports.main = main;
```

Project Codename: Run

```
wsk action invoke --blocking alexa/project-codename
```

... random 2-word responses appear.

Redis Integration

As an example, storing the new project codename in Redis:

```
if(args.redisURL) {
  bluebird.promisifyAll(redis.RedisClient.prototype);
  var client = redis.createClient(args.redisURL);
  return client.setAsync(["codenames", random])
          .then(function (result) {
    return response;
  });
} else {
  return response;
}
```

Serverless In The Real World

Beyond the trivial example, many things are possible:

- connect to a datastore
- make an API call
- trigger other actions
- ... your imagination is the limit

The Serverless Revolution

Resources

- Project codename skill: https://github.com/lornajane/alexa-project-codename
- Serverless framework: https://github.com/serverless/serverless
- IBM Cloud Functions: https://www.ibm.com/cloud-computing/bluemix/openwhisk
- My blog: https://lornajane.net