Fun With Serverless Python

Lorna Mitchell, IBM
https://speakerdeck.com/lornajane

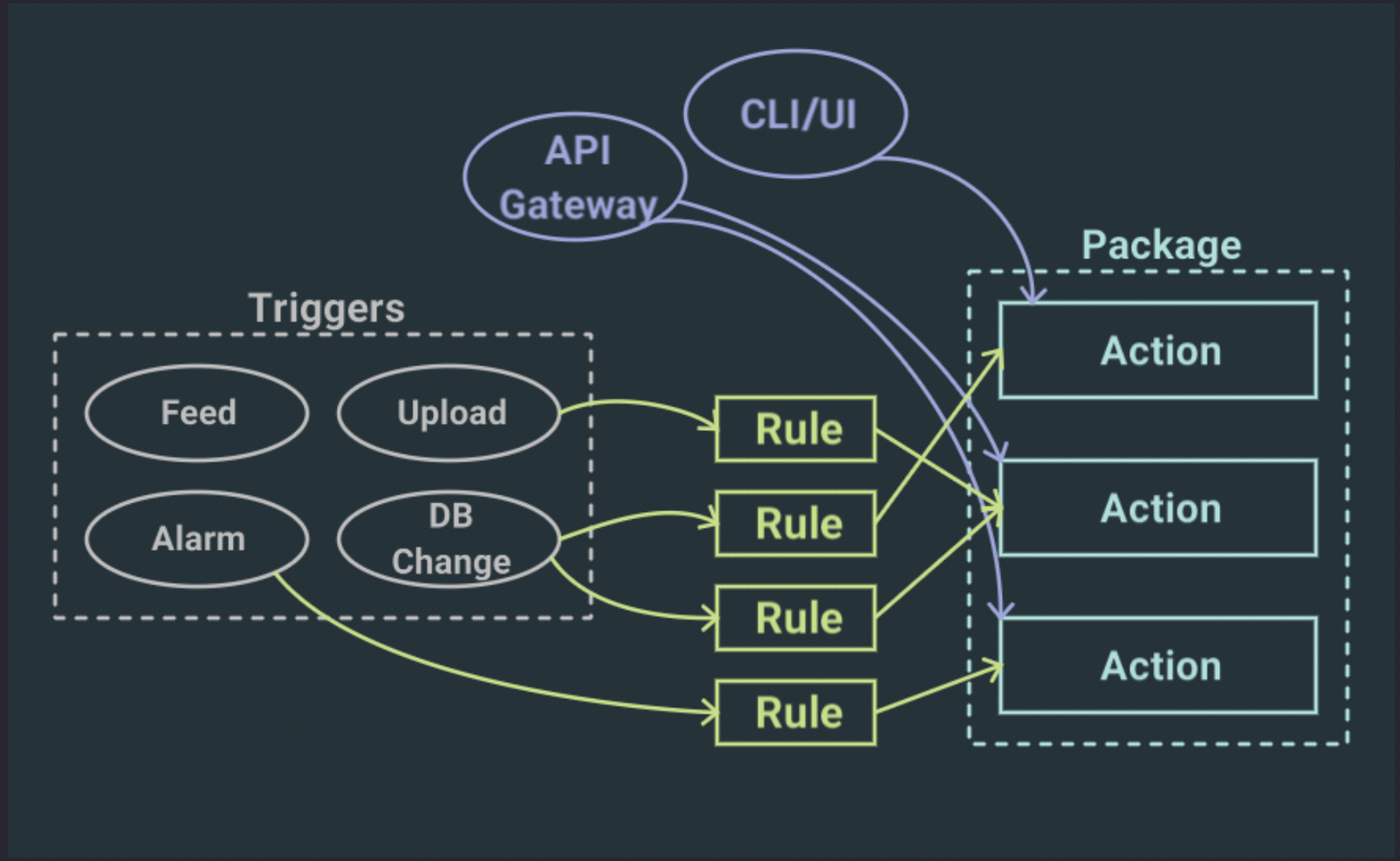
The Serverless Revolution

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



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 Apache OpenWhisk)
- Google Cloud Functions
- Azure 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. `hello.py` -> `hello.zip`)

Amazon Lambda

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

```
aws lambda create-function \
    --function-name hipy \
    --runtime python3.6 \
    --role arn:aws:iam::283476131276:role/service-role/Alexa
    --description "Hello World in Python" \
    --handler hello.main \
    --zip-file fileb://hello.zip
```

Amazon Lambda

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

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

Run your lambda function:

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

Hello World Lambda

```
def main(event, context):
    return {
        'message':'Hello World'
    }
```

Hello World OpenWhisk

```
def main(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 __main__.py

bx wsk action update --kind python:3 pydemo/hipy hello.zip
```

`pydemo` is the package name

IBM Cloud Functions

To handle dependencies, add more files to your zip

e.g to include the virtualenv:

```
zip -r hello.zip __main__.py virtualenv
```

IBM Cloud Functions

Run your action from the CLI:

```
bx wsk action invoke --blocking pydemo/hipy
```

Enable web access and web request your action:

```
bx wsk action update pydemo/hipy --web true

curl https://openwhisk.eu-gb.bluemix.net/api/v1/web/ \
    Lorna.Mitchell_dev/pydemo/hipy.json
```

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: Airport Codes

Skill code on GitHub:
https://github.com/lornajane/alexa-airport-codes

"Alexa, ask Airporter which airport has code CWL"

Airport Codes: Code

```
 1 import redis
 2
 3 def main(args):
 4     redis_client = redis.StrictRedis(...)
 5     code = args['request']['intent']['slots']['Code']['value'];
 6     a = redis_client.hgetall("code:" + code)
 7     airport_info = "Airport code " + code
 8     airport_info += " is for " + a['name']
 9     airport_info += " in " + a['country']
10
11     speech = {"type": "PlainText", "text": airport_info}
12     response = {"shouldEndSession": "true", "outputSpeech": spe
13     return {"version": "1.0", "response": 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

- IBM Cloud Functions: https://www.ibm.com/cloud-computing/bluemix/openwhisk
- GitHub repo for Airport Codes skill: https://github.com/lornajane/alexa-airport-codes
- OpenWhisk: https://openwhisk.incubator.apache.org/
- Redis: https://redis.io/

- My blog: https://lornajane.net