

# Introduction to Serverless Computing

with Apache OpenWhisk, *an open source, distributed Serverless platform*

Upkar Lidder  
Developer Advocate, IBM

**IBM Developer**

> [ulidder@us.ibm.com](mailto:ulidder@us.ibm.com)  
> [@lidderupk](https://twitter.com/lidderupk)  
> [upkar.dev](https://upkar.dev)



**There's no serverless  
it's just someone else's container**

# What is Serverless

Serverless computing refers to the concept of building and running applications that **do not require server management.**

It describes a finer-grained deployment model where applications, **bundled as one or more functions,** are uploaded to a platform and then **executed, scaled, and billed in response to the exact demand** needed at the moment.

It refers to the idea that consumers of serverless computing **no longer need to spend time and resources on server provisioning, maintenance, updates, scaling, and capacity planning.** Instead, all of these tasks and capabilities are handled by a serverless platform and are completely **abstracted away from the developers.**

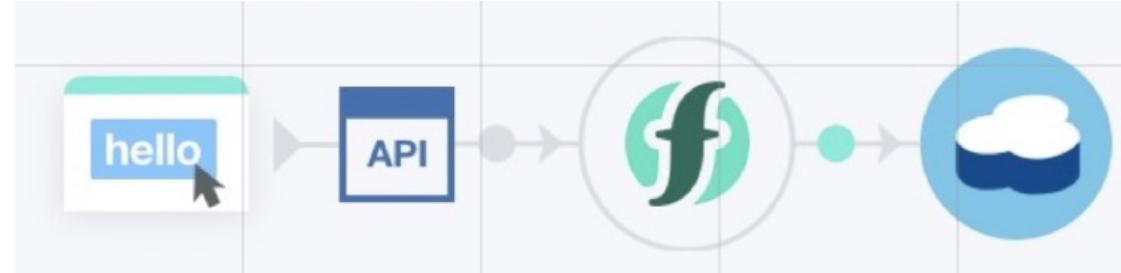
**Cloud Native Computing Foundation**

<https://github.com/cncf/wg-serverless/tree/master/whitepapers/serverless-overview>

# Use Cases

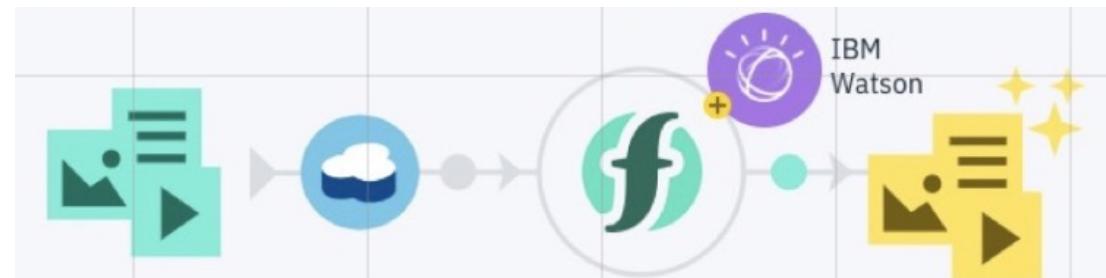
- **Secure and scalable APIs**

- API Gateway ➔ Function
- HTTP level scaling



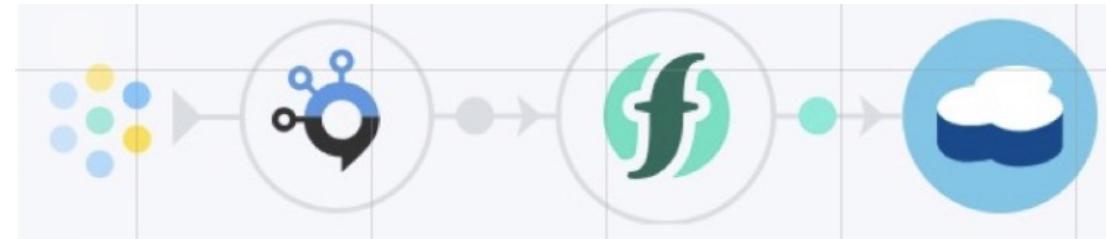
- **Backend services / micro services**

- function/service level scaling
- polyglot functions



- **Event Driven Programming**

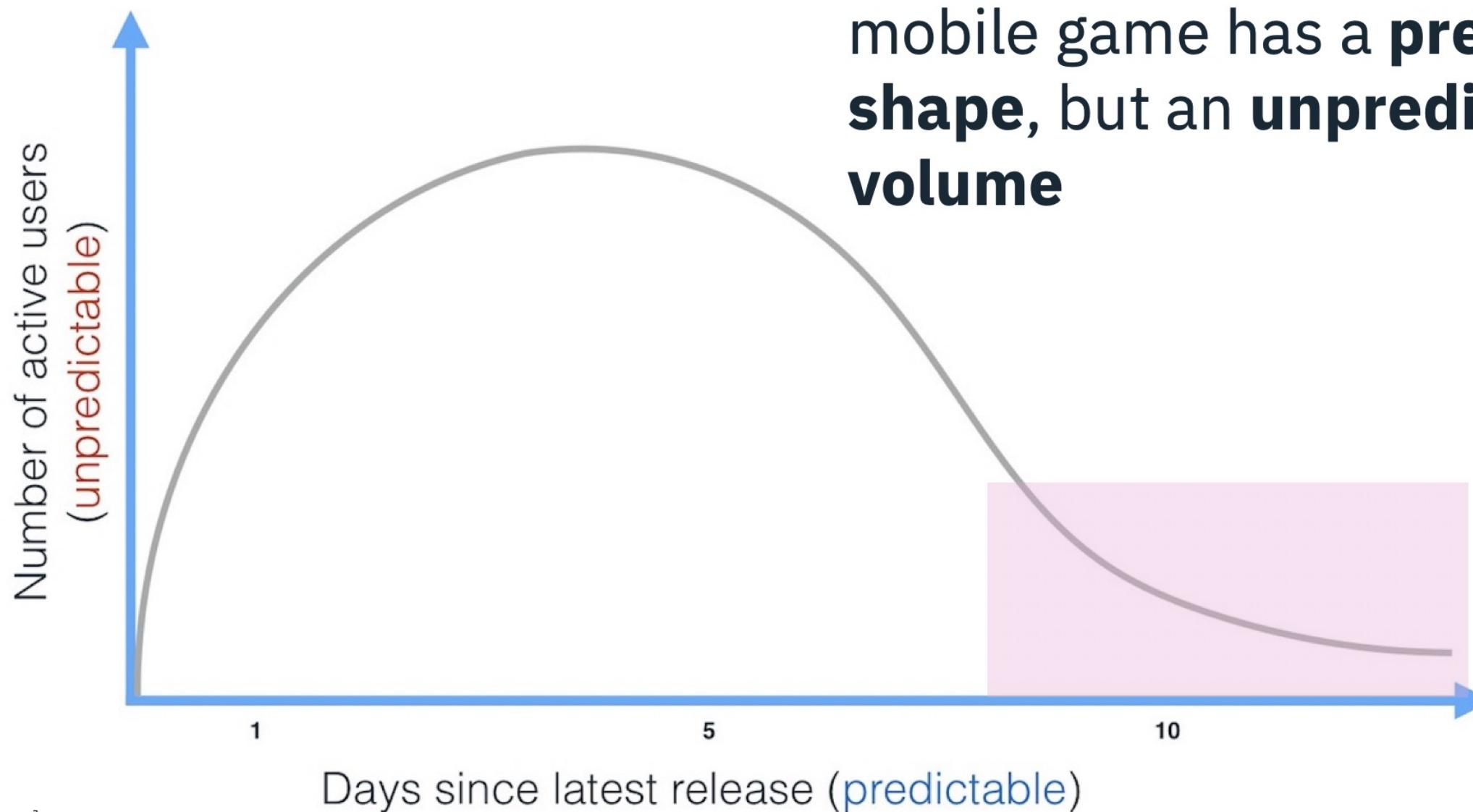
- Changes in database
- Social events



# Leans well to ...

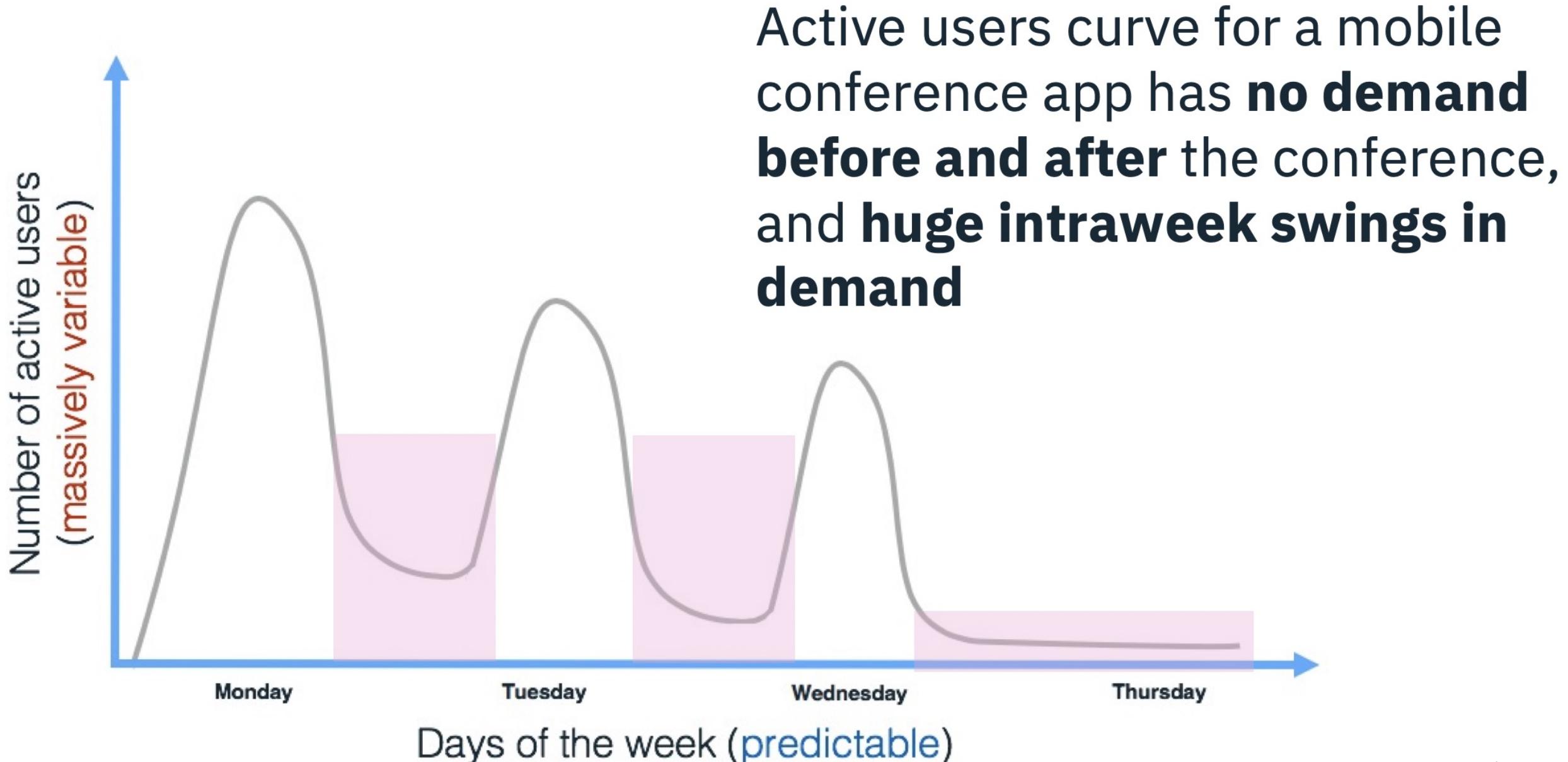
- Small, focused, asynchronous, concurrent, easy to parallelize into independent units of work
- Infrequent or has sporadic demand, with large, unpredictable variance in scaling requirements
- Stateless, ephemeral, without a major need for instantaneous cold start time
- Highly dynamic in terms of changing business requirements that drive a need for accelerated developer velocity

# Example - Game App

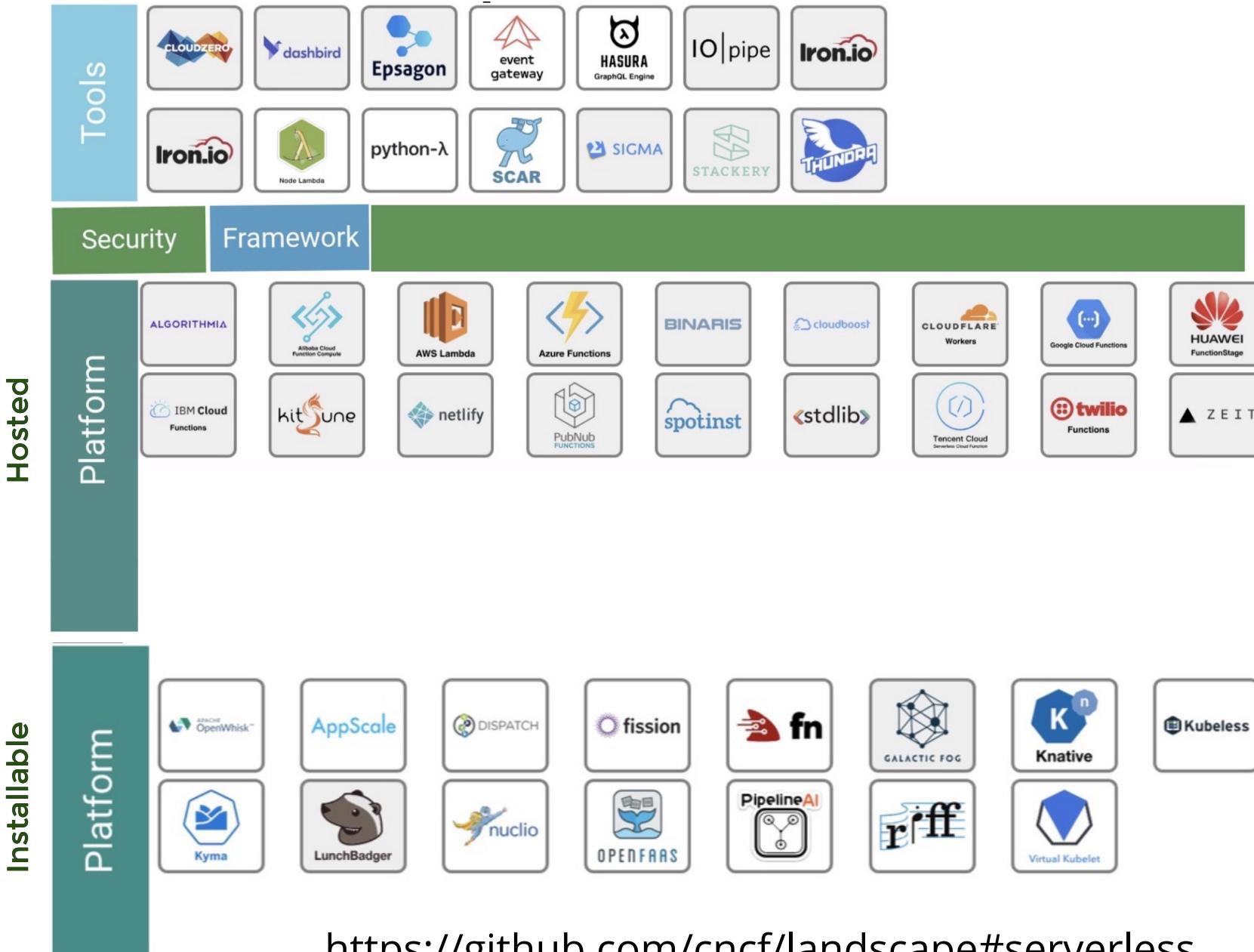


Active users curve for a new mobile game has a **predictable shape**, but an **unpredictable volume**

# Example - Conference App



# Serverless Landscape



<https://github.com/cncf/landscape#serverless>

# Serverless Landscape

## Amazon Lambda

- Node.js, Python, Java, C# and Go



## IBM Cloud Functions

- Node.js 8, Node.js 6, Python 3.6.4, Python 3.6.1, PHP 7.1, PHP 7.2, and Swift 4, Swift 4.1, Ruby 2.5, Docker
- Recent change – 2GB max memory and 15 minutes run limit.

## Microsoft Azure

- C#, F#, Node.js (in GA), Java, Python, PHP, TypeScript, Bash, PowerShell (experimental mode)

## Google Cloud Function

- Node.js, Python

**This space is constantly changing !**

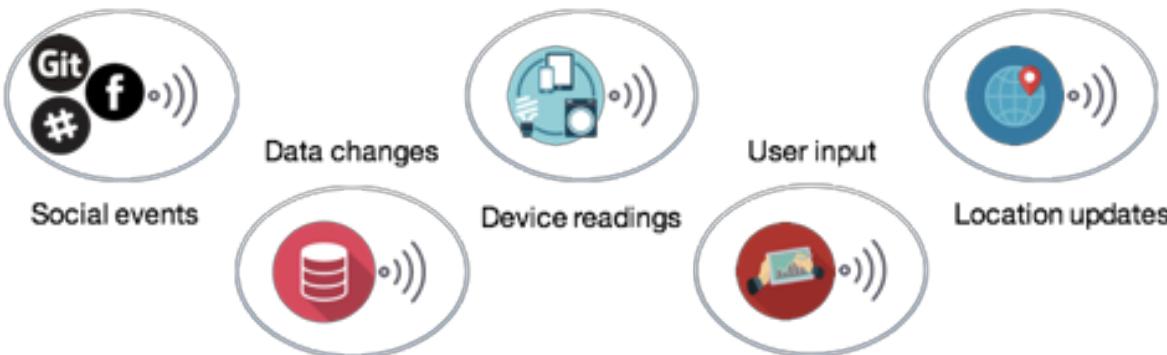
# Apache OpenWhisk

Apache OpenWhisk is an open source, distributed Serverless platform that supports a programming model in which developers write functional logic (called **Actions**), in any supported programming language, that can be dynamically scheduled and run (with **Rules**) in response to associated events (via **Triggers**) from external sources ( Feeds) or from HTTP requests.

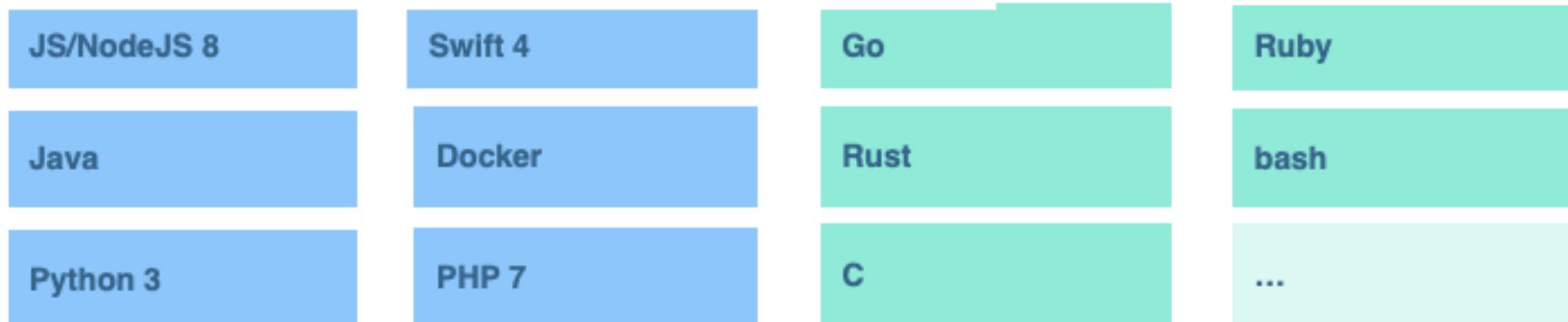


# Apache OpenWhisk

**T** A class of events that can occur

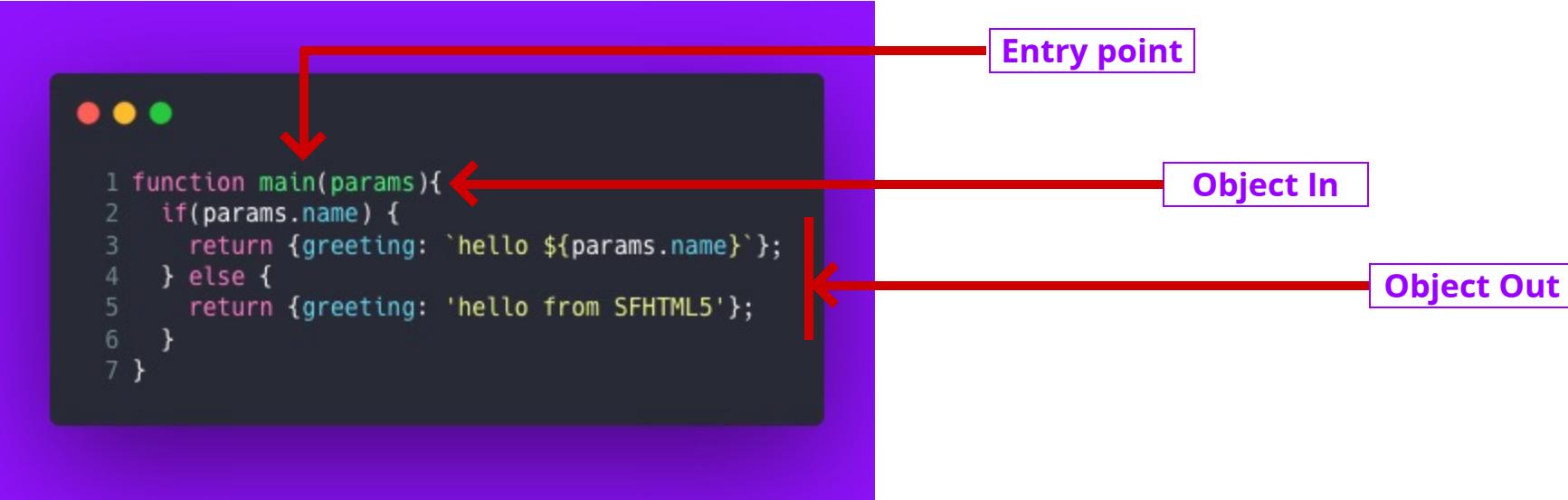


**A** Can be written in a variety of languages, such as JavaScript, Python, Java, PHP, and Swift



Packages

# Apache OpenWhisk (js)



```
→ javascript-hello wsk action create hellojs index.js --kind nodejs:10
ok: created action hellojs
```

```
→ javascript-hello wsk action list
actions
```

/upkar.ibm.watson.6@gmail.com_dev/hellojs	private nodejs:10
/upkar.ibm.watson.6@gmail.com_dev/HiveModerator	private python:3.7
/upkar.ibm.watson.6@gmail.com_dev/hellohive	private nodejs:6
/upkar.ibm.watson.6@gmail.com_dev/ibmWatsonModerator	private python:3.7
/upkar.ibm.watson.6@gmail.com_dev/upkar-action	private nodejs:6
/upkar.ibm.watson.6@gmail.com_dev/moderator-action	private python:3.7
/upkar.ibm.watson.6@gmail.com_dev/womenintech	private nodejs:6
/upkar.ibm.watson.6@gmail.com_dev/WatsonModerator	private python:3.7

# Apache OpenWhisk (js)

```
● ● ●

function main(params) {
    return new Promise((resolve, reject) => {
        // longRunningOperation()

        if (params.name === 'seattlejs') {
            console.log('status: success');
            resolve({status: 'success'});
        } else {
            console.log('status: failure');
            reject({status: 'failure'})
        }
    })
}
```

# WSK CLI

```
→ javascript-hello wsk
```



Usage:  
wsk [command]

#### Available Commands:

action	work with actions
activation	work with activations
package	work with packages
rule	work with rules
trigger	work with triggers
sdk	work with the sdk
property	work with whisk properties
namespace	work with namespaces
list	list entities in the current namespace
api	work with APIs
bluemix	bluemix integration

#### Flags:

--apihost HOST	whisk API HOST
--apiversion VERSION	whisk API VERSION
-u, --auth KEY	authorization KEY
--cert string	client cert
-d, --debug	debug level output
-h, --help	help for wsk
-i, --insecure	bypass certificate check
--key string	client key
-v, --verbose	verbose output

```
→ javascript-hello wsk action
```

work with actions

Usage:  
wsk action [command]

#### Available Commands:

create	create a new action
update	update an existing action, or create an action if it does not exist
invoke	invoke action
get	get action
delete	delete action
list	list all actions in a namespace or actions contained in a package

#### Flags:

-h, --help	help for action
------------	-----------------

#### Global Flags:

--apihost HOST	whisk API HOST
--apiversion VERSION	whisk API VERSION
-u, --auth KEY	authorization KEY
--cert string	client cert
-d, --debug	debug level output
-i, --insecure	bypass certificate checking
--key string	client key
-v, --verbose	verbose output

# WSK SYSTEM PACKAGES

```
➔ ibmdeveloperst wsk package list /whisk.system  
packages
```

/whisk.system/slack	shared
/whisk.system/websocket	shared
/whisk.system/weather	shared
/whisk.system/samples	shared
/whisk.system/utils	shared
/whisk.system/cloudant	shared
/whisk.system/alarms	shared
/whisk.system/messaging	shared
/whisk.system/pushnotifications	shared
/whisk.system/watson-textToSpeech	shared
/whisk.system/github	shared
/whisk.system/combinators	shared
/whisk.system/watson-speechToText	shared
/whisk.system/watson-translator	shared

```
➔ ibmdeveloperst wsk package get /whisk.system/weather --summary
```

```
package /whisk.system/weather: Services from the Weather Company Data for IBM Bluemix
```

```
(parameters: *bluemixServiceName, password, username)
```

```
action /whisk.system/weather/forecast: IBM Weather Insights 10-day forecast
```

```
(parameters: host, language, latitude, longitude, password, timePeriod, units, username)
```

# WSK SYSTEM PACKAGES

```
package /whisk.system/cloudant: Cloudant database service
  (parameters: *apihost, *bluemixServiceName, dbname, host, iamApiKey, iamUrl, overwrite, password,
  username)

  action /whisk.system/cloudant/delete-attachment: Delete document attachment from database
    (parameters: attachmentname, dbname, docid, docrev, params)
  action /whisk.system/cloudant/update-attachment: Update document attachment in database
    (parameters: attachment, attachmentname, contenttype, dbname, docid, docrev, params)
  action /whisk.system/cloudant/read-attachment: Read document attachment from database
    (parameters: attachmentname, dbname, docid, params)
  action /whisk.system/cloudant/create-attachment: Create document attachment in database
    (parameters: attachment, attachmentname, contenttype, dbname, docid, docrev, params)
  action /whisk.system/cloudant/read-changes-feed: Read Cloudant database changes feed (non-continuous)
    (parameters: dbname, params)
  action /whisk.system/cloudant/delete-query-index: Delete index from design document
    (parameters: dbname, docid, indexname, params)
  action /whisk.system/cloudant/delete-view: Delete view from design document
    (parameters: dbname, docid, params, viewname)
  action /whisk.system/cloudant/manage-bulk-documents: Create, Update, and Delete documents in bulk
    (parameters: dbname, docs, params)
  action /whisk.system/cloudant/exec-query-view: Call view in design document from database
    (parameters: dbname, docid, params, viewname)
  action /whisk.system/cloudant/exec-query-search: Execute query against Cloudant search
    (parameters: dbname, docid, indexname, search)
  action /whisk.system/cloudant/exec-query-find: Execute query against Cloudant Query index
    (parameters: dbname, query)
...
feed   /whisk.system/cloudant/changes: Database change feed
  (parameters: dbname, filter, iamApiKey, iamUrl, query_params)
```



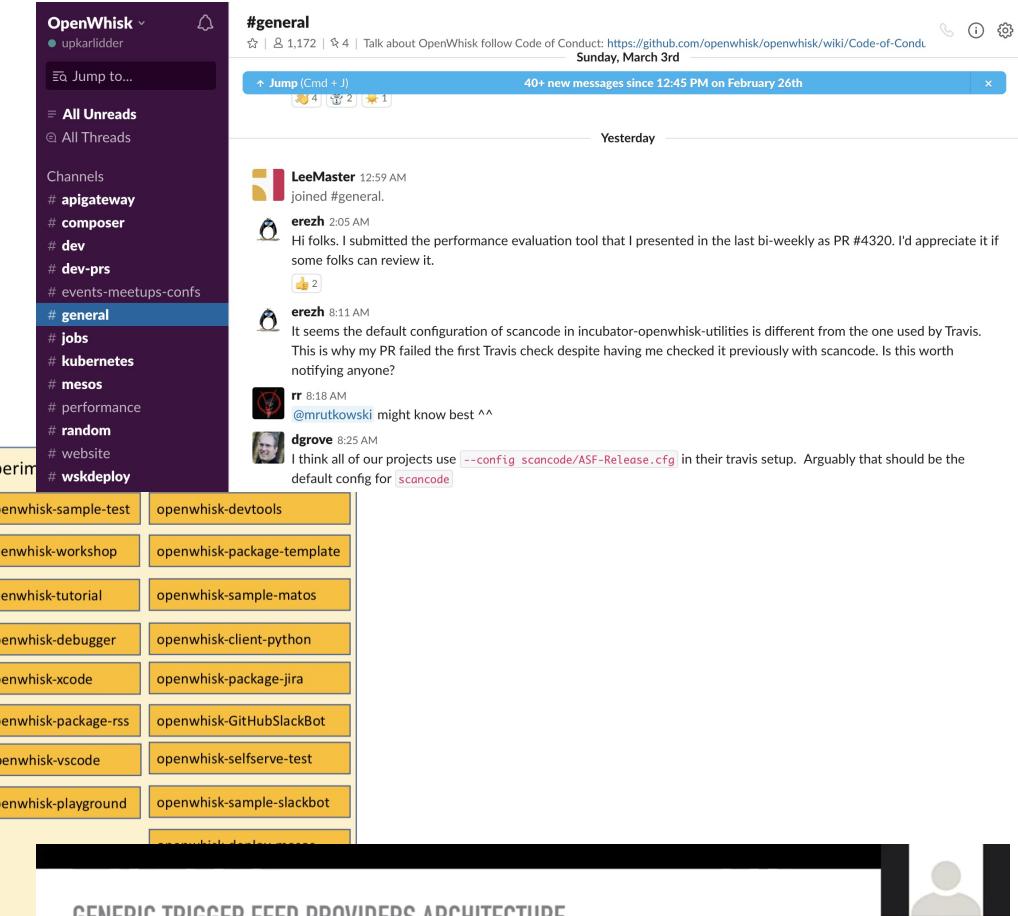
<https://github.com/lidderupk/serverless-java-mini-workshop>

# So much more ...

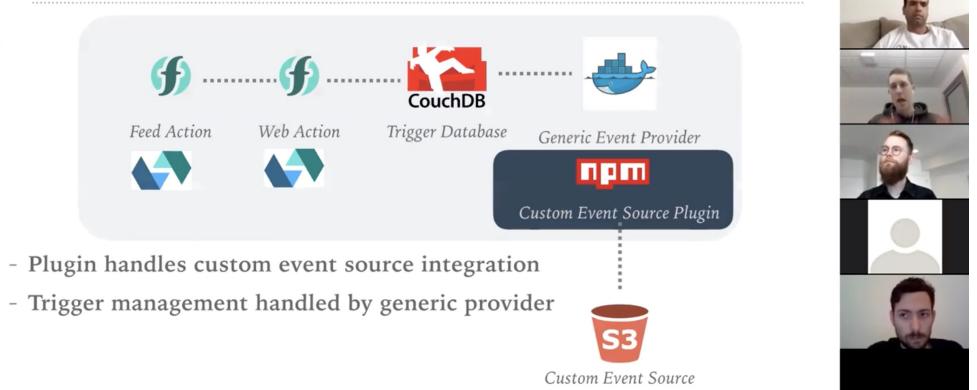
- More Compositions - Use higher-level programming constructs to declaratively chain together multiple actions.
  - while, repeat, let, retry, try, if, etc
- HTTP Actions (web == true)
- Packages community (couchdb, kafka, etc)
- API Gateway (rate limiting, authentication/authorization)
- Install external packages, debugging, etc

# Community and Help

- Slack
- <http://openwhisk-team.slack.com/>
- Github and OpenWhisk docs
- Youtube/Medium/Twitter
- Bi-weekly interexchange meetings



GENERIC TRIGGER FEED PROVIDERS ARCHITECTURE



# Some things to 🤔

- Functions are stateless. Need some sort of persistence between runs.
- Are you able to test and develop locally ? Does provider have CLI ?
- Can you easily version your functions ? Source control ?
- Can you easily monitor your functions ?
- Security and API gateway
- Avoid long-running loops / mini-monoliths ?
- Latency (cold, warm and hot loads)
- How do you track dependencies ?

# Thank you

## Let's chat !

Upkar Lidder, IBM

@lidderupk

<https://github.com/lidderupk/>

ulidder@us.ibm.com

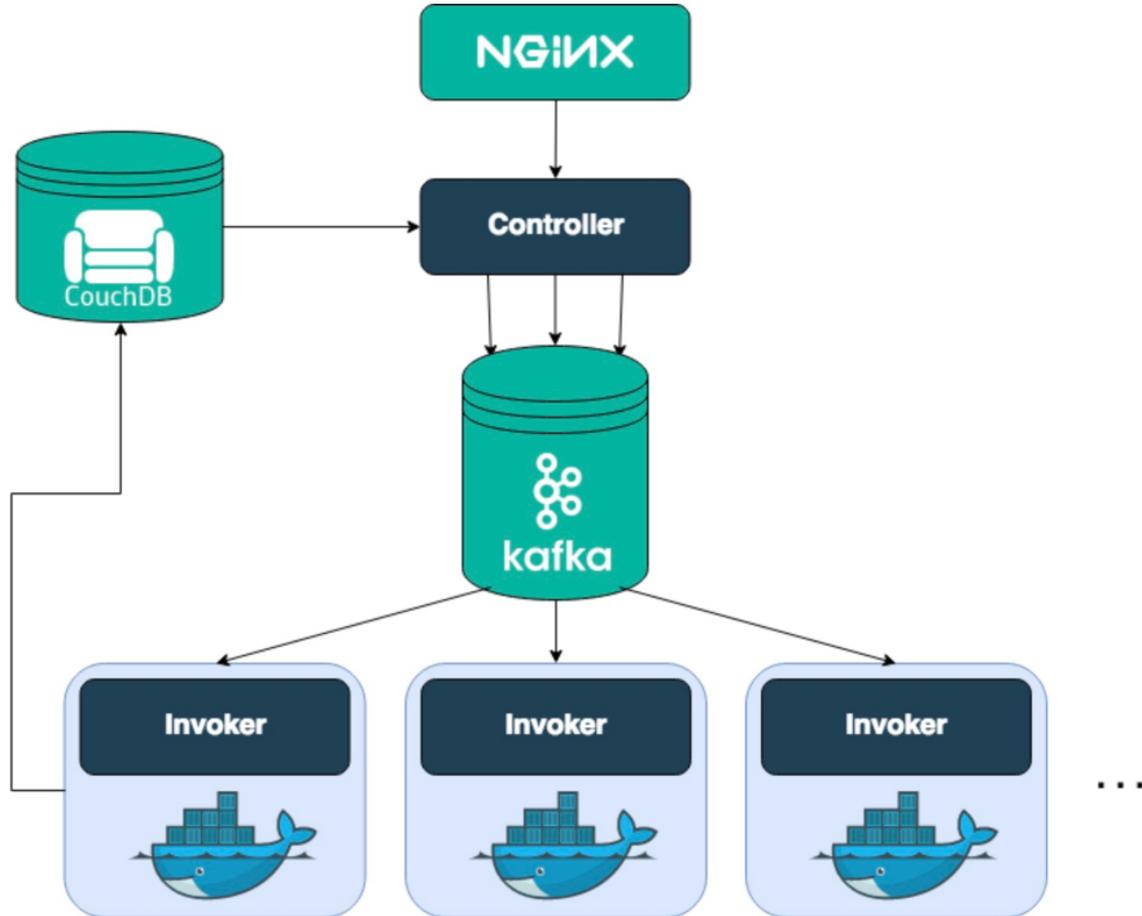
#BehindtheCode



It's about time  
developers got  
some attention.

IBM  
Technology  
Platform

# Solid Foundations



## Entering the system: nginx

“an HTTP and reverse proxy server”.

## Entering the system: Controller

serves as the interface for everything a user can do, including CRUD requests for your entities in OpenWhisk and invocation of actions

## Authentication and Authorization: CouchDB

check that the user exists in OpenWhisk’s database and that it has the privilege to invoke the action myAction

## Getting the action: CouchDB... again

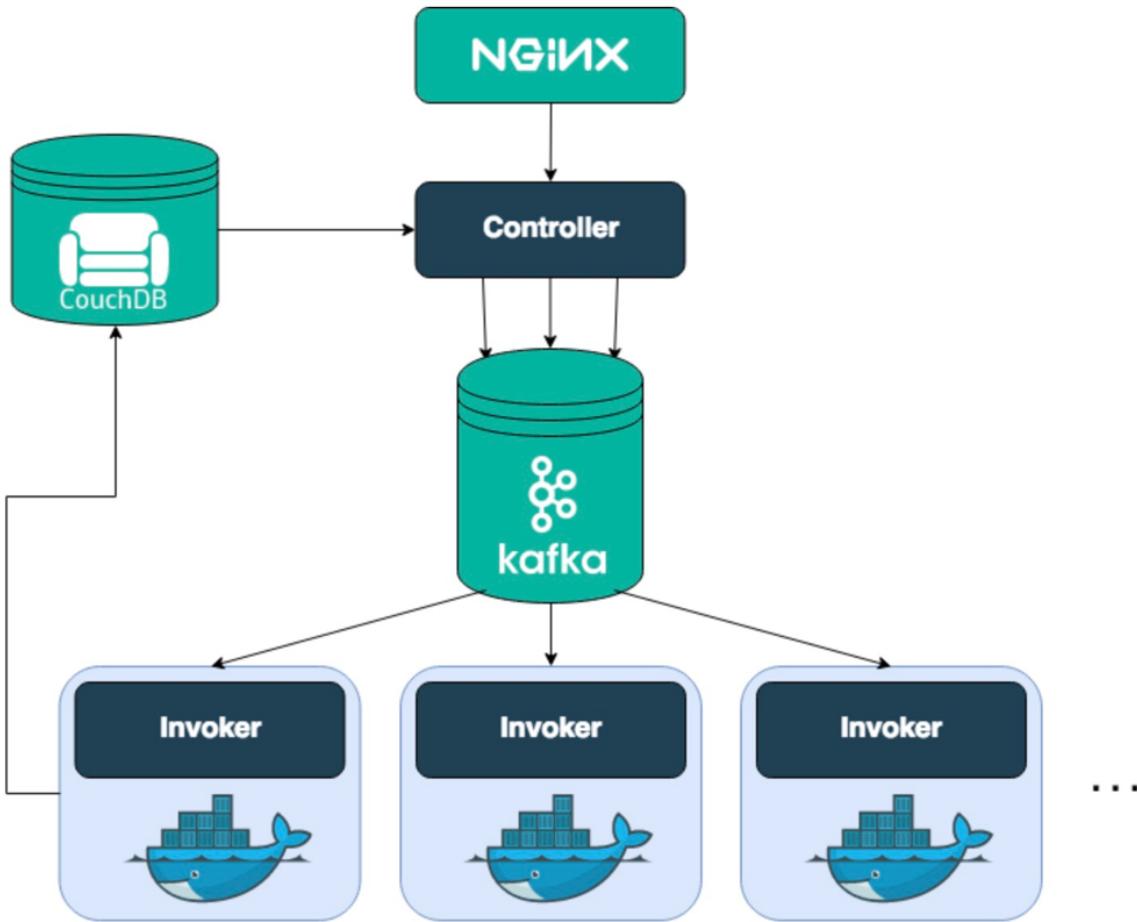
Load the action from the **whisks** database in CouchDB. The record of the action contains mainly the code to execute (shown above) and default parameters that you want to pass to your action, merged with the parameters you included in the actual invoke request. It also contains the resource restrictions imposed on it in execution, such as the memory it is allowed to consume.

## Who’s there to invoke the action: Load Balancer

has a global view of the executors available in the system by checking their health status continuously. Those executors are called **Invokers**. The Load Balancer, knowing which Invokers are available, chooses one of them to invoke the action requested.

Leverages existing opensource solutions

# Solid Foundations



## Please form a line: Kafka

"a high-throughput, distributed, publish-subscribe messaging system". Controller and Invoker solely communicate through messages buffered and persisted by Kafka.

## Actually invoking the code already: Invoker

The Invoker's duty is to invoke an action. To execute actions in an isolated and safe way it uses **Docker**.

**Example:** the Invoker will start a Node.js container, inject the code from *myAction*, run it with no parameters, extract the result, save the logs and destroy the Node.js container again.

## Storing the results: CouchDB again

As the result is obtained by the Invoker, it is stored into the **activations** database as an activation. The record contains both the returned result and the logs written. It also contains the start and end time of the invocation of the action.

The system itself mainly consists of only two custom components, the **Controller** and the **Invoker**.

**Everything else is already there, developed by so many people out there in the open-source community.**

Leverages existing opensource solutions

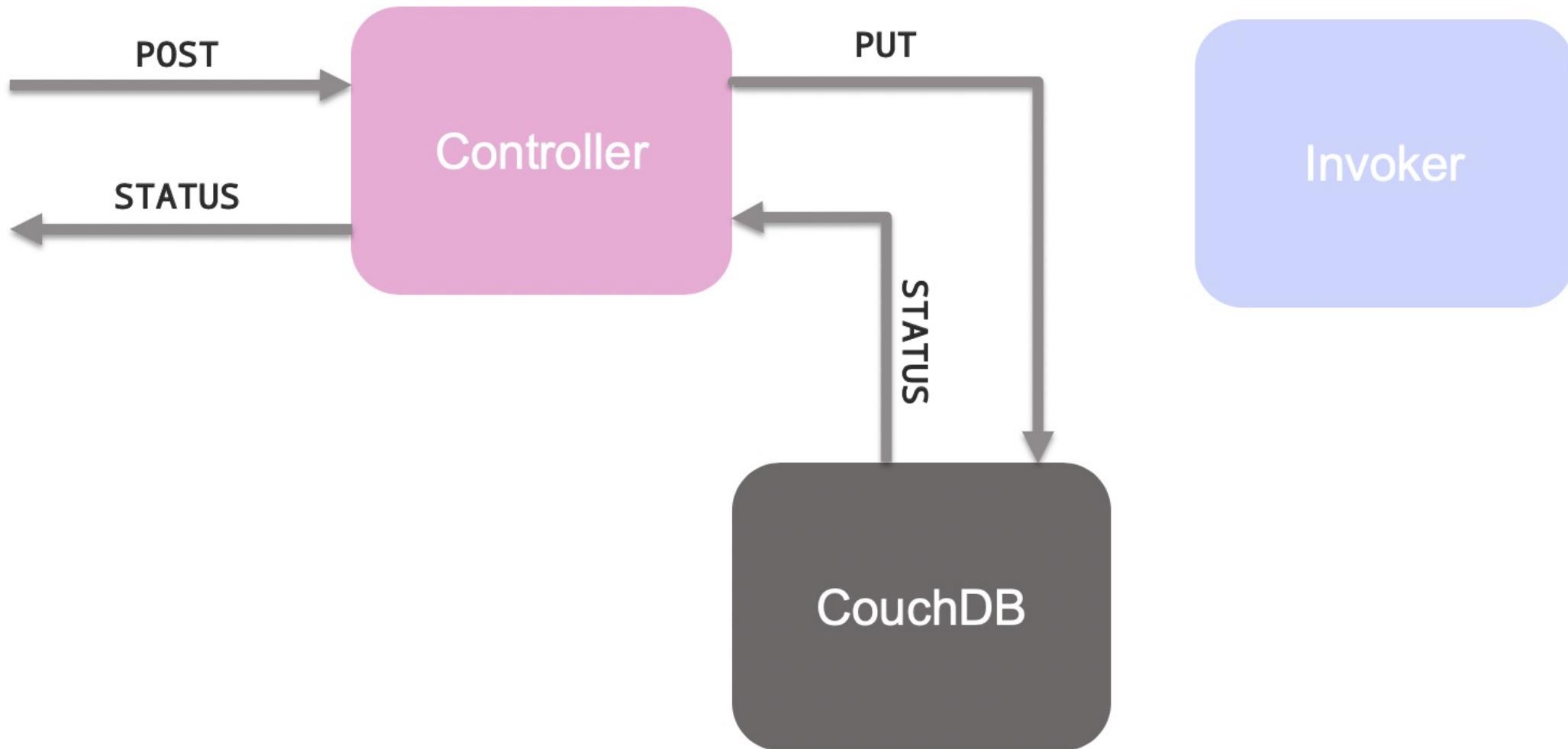
# Built on Open Source

```
$ docker ps --format "{{.ID}}: {{.Names}} {{.Image}}"
17c5d31e2c20: wsk0_60_prewarm_nodejs6
0eace484289c: wsk0_59_prewarm_nodejs6
1be725d8767c: openwhisk_apigateway_1
641cbabeb790: openwhisk_kafka-topics-ui_1
f52c25dbadd9: openwhisk_controller_1
8f0c6aa14ccc: openwhisk_invoker_1
d5274194f842: openwhisk_kafka-rest_1
40a1585f64bb: openwhisk_kafka_1
b0b0f75c6fdb: openwhisk_db_1
a7449c2edc4d: openwhisk_zookeeper_1
178abe09b793: openwhisk_redis_1
```

The Docker command `docker ps` lists the running containers. The output shows 10 containers with their names and images. The names are color-coded: wsk0\_60\_prewarm\_nodejs6 (yellow), wsk0\_59\_prewarm\_nodejs6 (yellow), openwhisk\_apigateway\_1 (green), openwhisk\_kafka-topics-ui\_1 (red), openwhisk\_controller\_1 (blue), openwhisk\_invoker\_1 (purple), openwhisk\_kafka-rest\_1 (pink), openwhisk\_kafka\_1 (magenta), openwhisk\_db\_1 (light blue), openwhisk\_zookeeper\_1 (orange), and openwhisk\_redis\_1 (yellow). The images are: openwhisk/nodejs6action:latest, openwhisk/nodejs6action:latest, adobeapiplatform/apigateway:1.1.0, landoop/kafka-topics-ui:0.9.3, openwhisk/controller, openwhisk/invoker, confluentinc/cp-kafka-rest:3.3.1, wurstmeister/kafka:0.11.0.1, couchdb:1.6, zookeeper:3.4, and redis:2.8.

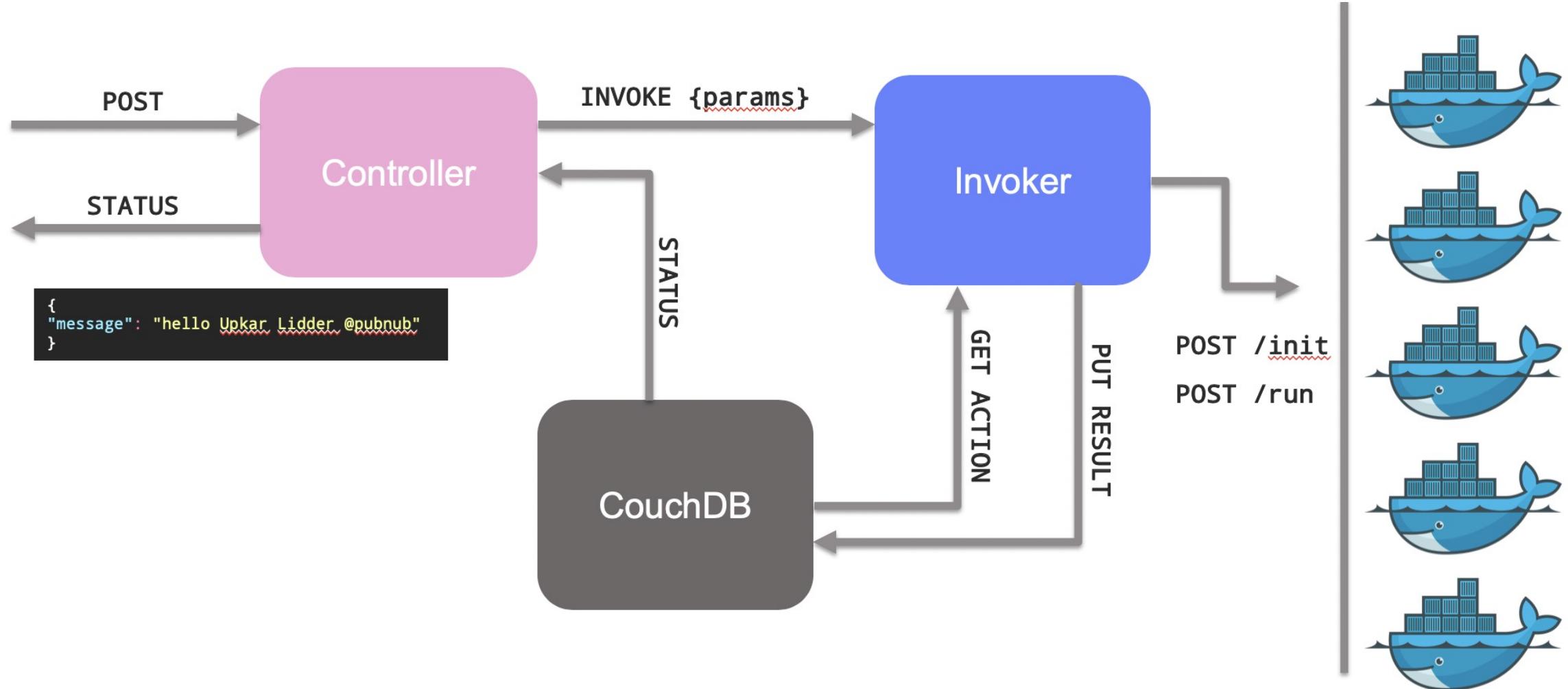
# Apache OpenWhisk (create)

wsk action create helloJava target/hello-world-java.jar --main com.example.FunctionApp

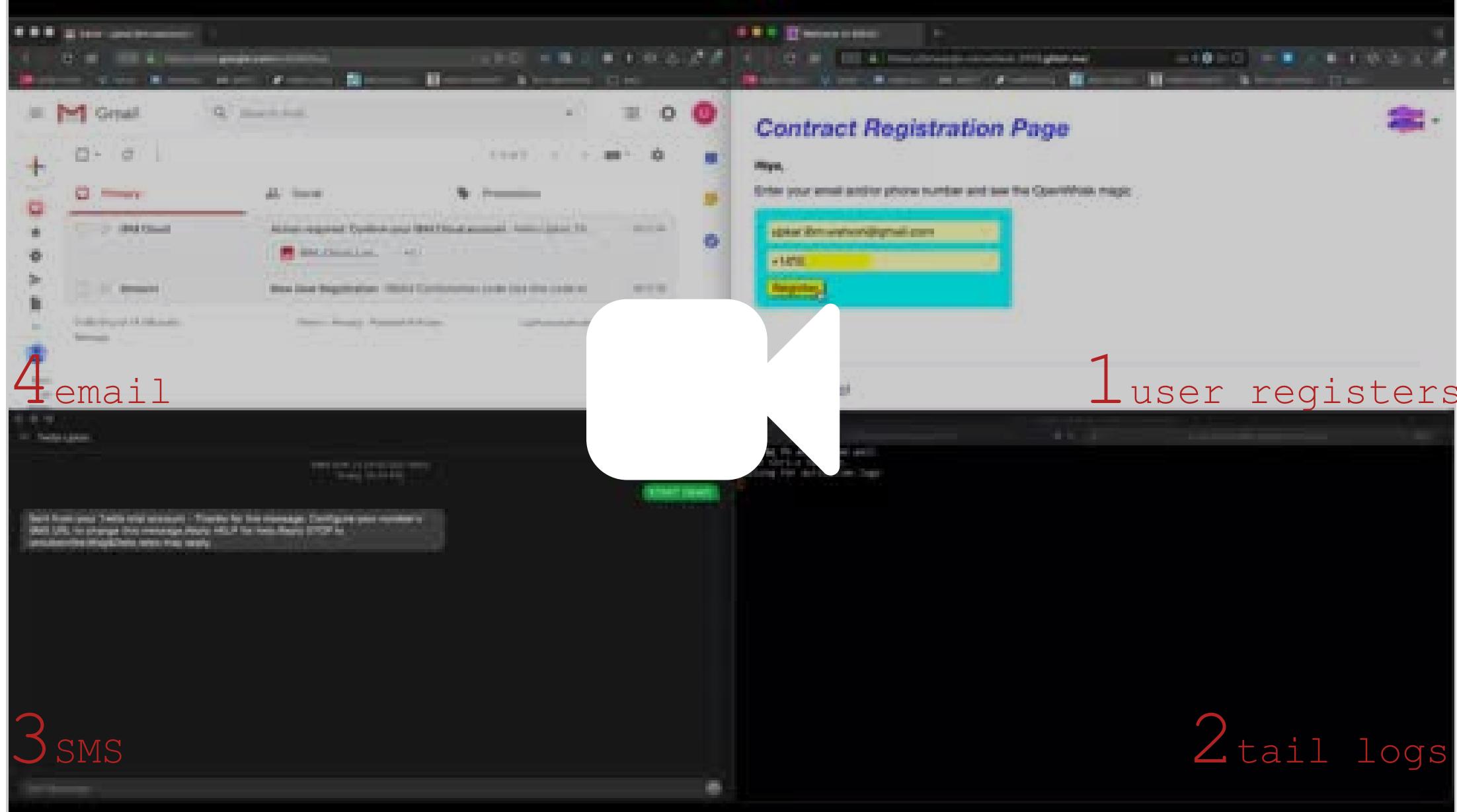


# Apache OpenWhisk (invoke)

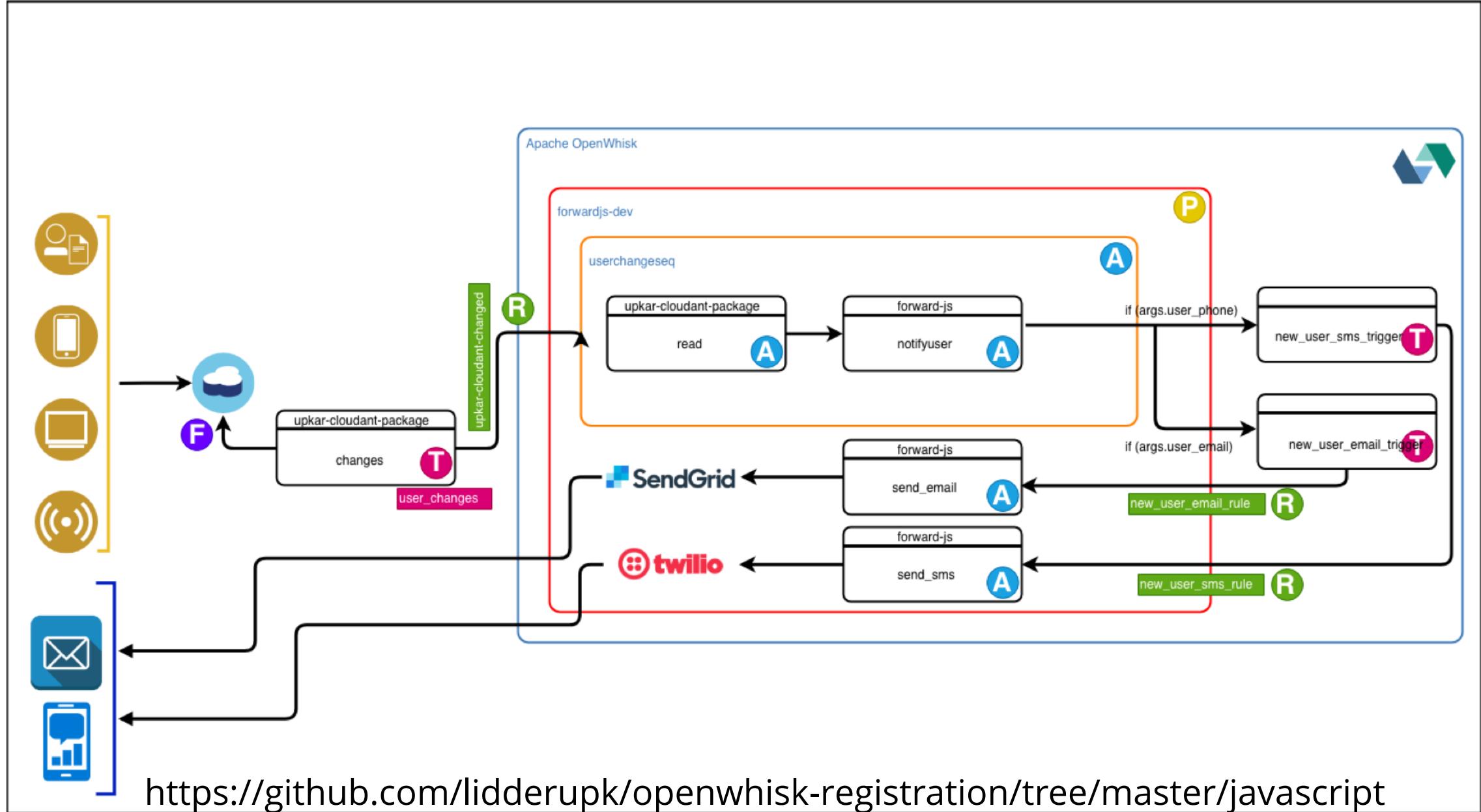
wsk action invoke --result helloJava --param name World



# Apache OpenWhisk - more real example



# Apache OpenWhisk - more real example



# Composition / Orchestration



# Composition / Orchestration

```
→ openwhisk-example-js compose demo.js > demo.json
→ openwhisk-example-js deploy demo demo.json -w
ok: created actions /_/authenticate,/_/success,/_/failure,/_/demo
```

```
→ openwhisk-example-js wsk action invoke demo -p password notpassword -r
{
  "message": "failure"
}
→ openwhisk-example-js wsk action invoke demo -p password abc123 -r
{
  "message": "success"
}
```



```
Activation: 'demo' (1fb6070e222b4edfb6070e222b3edfed)
[
  "b9edf39f2b424a28adf39f2b423a2862",
  "456eed5271e34c8eaeed5271e3ac8e9e",
  "0332bb22dab14533b2bb22dab1e5338a",
  "878b7bb4f10d48f08b7bb4f10db8f045",
  "e43e607f69cd48b6be607f69cd78b6ec"
]
```

# KUI Shell

```
> wsk action list
```

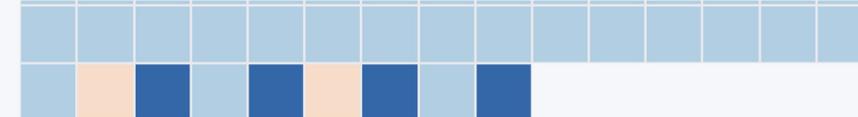
## ACTIONS

NAME	KIND
demo	python:3
failure	python:3
success	python:3
authenticate	python:3
python-annotate	python:3.7
hello-js	nodejs:6
pythonbotupkar	python:3.6
pythonbotforslack	python:3.6
my-python-bot-action	python:3.7
lajugaction	java
hackerdojojobot	python:3.7
hackerdojo-hello	nodejs:10
getBotJSON	nodejs:10
sfhtml5action	nodejs:6
HiveModerator	python:3.7
hellohive	nodejs:6
ibmWatsonModerator	python:3.7
upkar-action	nodejs:6
moderator-action	python:3.7
womenintech	nodejs:6
WatsonModerator	python:3.7

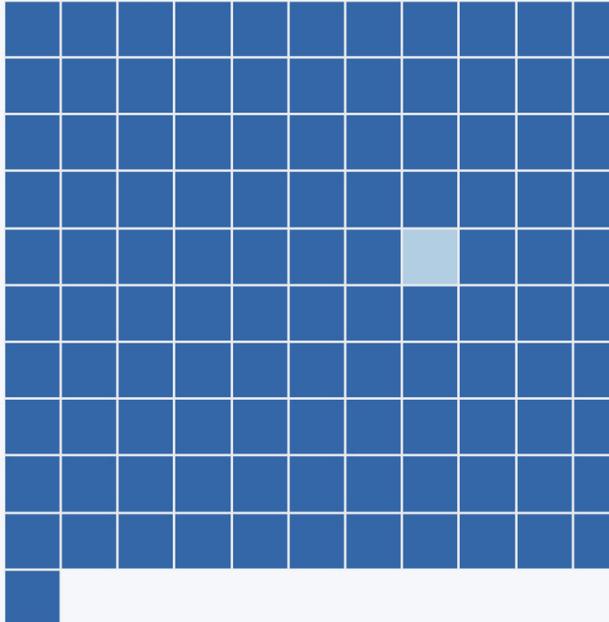
GRID

## Recent Activity

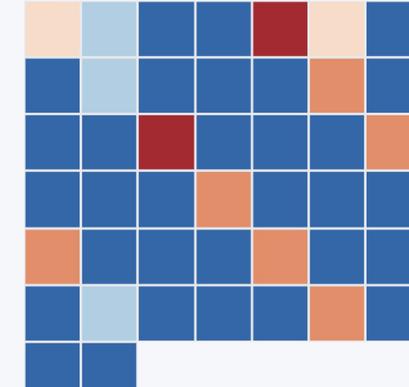
Showing 400 activations from Fri, Mar 8, 2:04:58 PM spanning ~2d



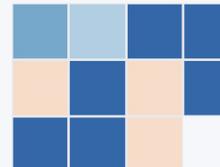
my-python-bot-action



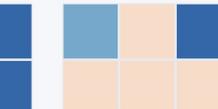
demo



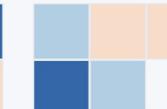
authenticate



failure



success



hello-js



python-compose



python-annotate



Summary Grid

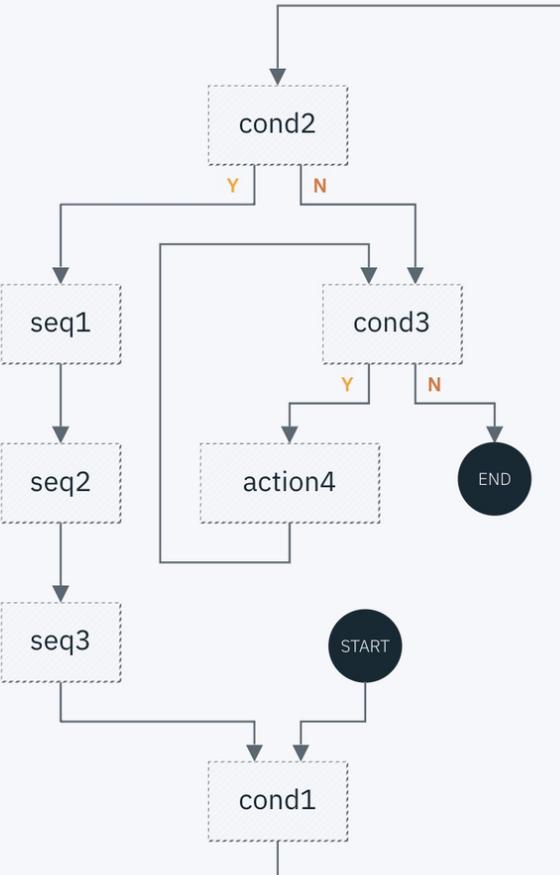


# KUI Shell

PREVIEW

## looper.js

This is a preview of your composition, it is not yet deployed



# Future of Serverless ?

The screenshot shows a GitHub repository page for 'kelseyhightower / nocode'. The repository has 2,223 issues, 360 pull requests, 0 projects, and 1 contributor. It uses the Apache-2.0 license. The README.md file contains the text: "The best way to write secure and reliable applications. Write nothing; deploy nowhere." Below the repository stats, there's a list of files: CONTRIBUTING.md, Dockerfile, LICENSE, and README.md. The README.md file is expanded, showing its content: "No Code is the best way to write secure and reliable applications. Write nothing; deploy nowhere." and "Getting Started: Start by not writing any code." A note at the bottom says: "This is just an example application, but imagine it doing anything you want. Adding new features is easy too:".

kelseyhightower / nocode

Watch 259 Unstar 27,297 Fork 2,350

Code Issues 2,223 Pull requests 360 Projects 0 Wiki Insights

The best way to write secure and reliable applications. Write nothing; deploy nowhere.

3 commits 1 branch 1 release 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find File Clone or download

**kelseyhightower add Docker support** 9 Latest commit ed6c73f on Feb 6, 2018

**CONTRIBUTING.md** add no code a year ago

**Dockerfile** add Docker support a year ago

**LICENSE** add no code a year ago

**README.md** add windows support a year ago

## No Code

No code is the best way to write secure and reliable applications. Write nothing; deploy nowhere.

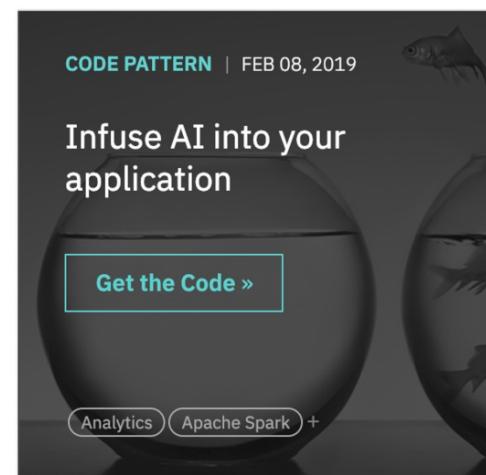
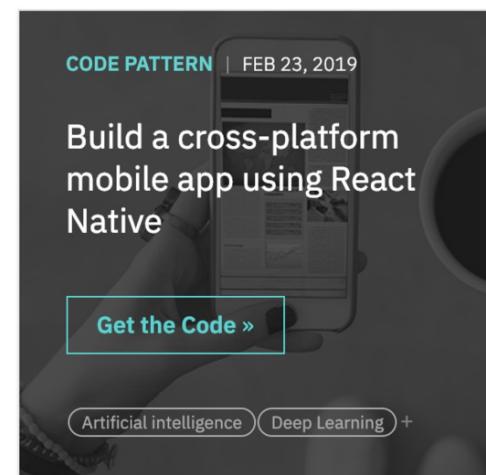
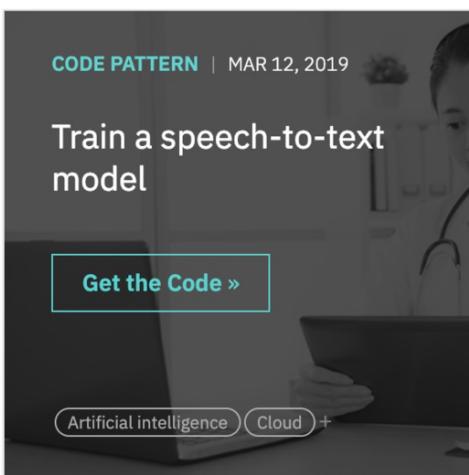
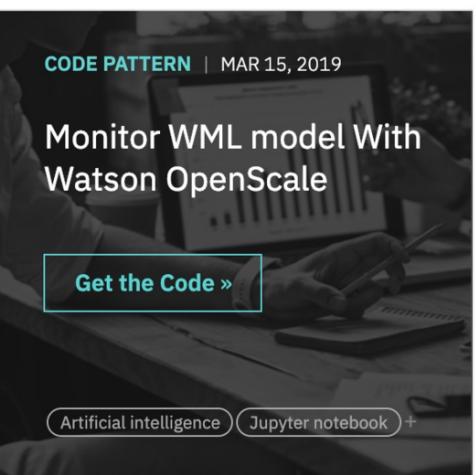
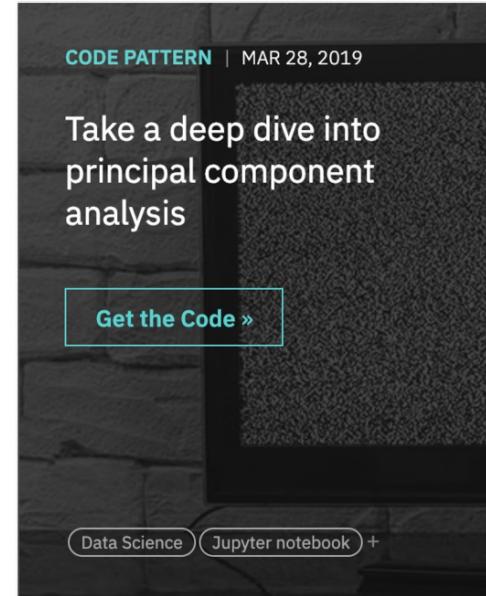
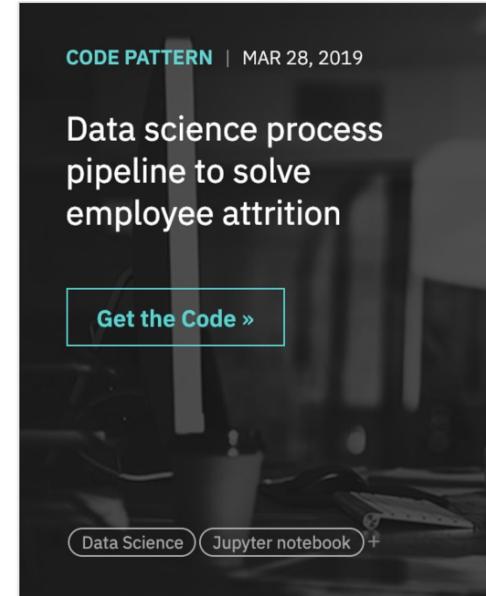
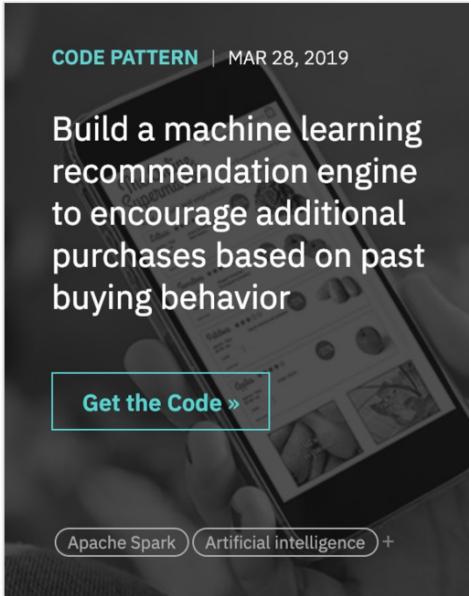
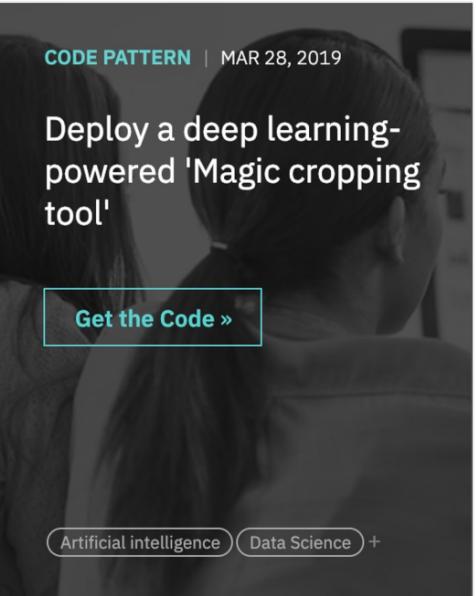
## Getting Started

Start by not writing any code.

This is just an example application, but imagine it doing anything you want. Adding new features is easy too:

# IBM Code Patterns

Machine Learning 



# IBM Partners

Enabling Independent Software Vendors (ISVs)  
and tech companies for growth

## Target audience

- ISVs and tech companies building and selling cloud solutions
- New to IBM Cloud
- Startups who aspire to build and sell their own solutions

## Offers to help you get started



### Build with up to \$12,000 of free IBM Cloud™ credits (\$1,000 per month for 12 months)

Integrate your solutions with leading-edge IBM Cloud technologies to deliver more innovation and value to your clients. Access more than **130 unparalleled services** including Watson™, Analytics and Security.



### Build with 10TB of IBM Cloud Object Storage at no charge

Build data capability into your offering. IBM Cloud Object Storage is designed for high durability, resiliency and security.



### Build with IBM Watson Assistant with a 1-year free trial

Receive access to 100K API calls per month plus 10 workspaces. Build and deploy chatbots quickly and efficiently with IBM Watson Assistant's advanced capabilities and seamless interface.

## Get started

Experience IBM's countless partner benefits. Start building and selling with IBM today.

Learn more and access offers at  
[ibm.com/partners/start](http://ibm.com/partners/start)



### Build with IBM Cloud Kubernetes Service with a 1-year free trial

Containerize your solution with 1TB of block storage. Ship all your applications in one agile, well-defined structure with IBM Cloud Kubernetes Service.



### Build with IBM Blockchain with a 6-month free trial

Build a network with up to 3 organizations to prototype. Build a secure business transaction network for your clients using blockchain and smart contracts.



### Finished building and testing? Go-to-market with IBM

Access Provider Workbench, attend an orientation session and join the premier network of over 400 partners who are already listing their solutions on the IBM Marketplace.



### Is your business a Startup? Build with up to \$120,000 in IBM Cloud credits

If your business revenue in the last 12 months is less than \$1M and you've been in business for fewer than five years, then you may qualify for Startup with IBM.

# Apache OpenWhisk - simple action



```
1 package openwhisk.java.example;
2
3 import com.google.gson.JsonObject;
4 import com.google.gson.JsonPrimitive;
5
6 import java.util.logging.Logger;
7
8 /**
9  * HelloWorld with Apache OpenWhisk
10 */
11 public class HelloWorld {
12
13     private static final Logger logger = Logger.getLogger("ibm");
14     public static JsonObject main(JsonObject args) {
15         JsonObject response = new JsonObject();
16         JsonPrimitive nameArg = args.getAsJsonPrimitive("name");
17         String result;
18         if (nameArg == null) {
19             result = "Hello! Welcome to OpenWhisk";
20         } else {
21             result = "Hello " + nameArg.getAsString() + " welcome to Dawscon!!!";
22         }
23         response.addProperty("greetings", result);
24
25         logger.info("invoked with params:");
26         return response;
27     }
28 }
```

Entry point

JsonObject In

JsonObject Out

# Apache OpenWhisk - sample pom.xml

```
1 <dependencies>
2     <dependency>
3         <groupId>com.google.code.gson</groupId>
4         <artifactId>gson</artifactId>
5         <version>2.8.2</version>
6     </dependency>
7     <dependency>
8         <groupId>junit</groupId>
9         <artifactId>junit</artifactId>
10        <version>4.12</version>
11        <scope>test</scope>
12    </dependency>
13 </dependencies>
14
15 <build>
16     <plugins>
17         <plugin>
18             <groupId>org.apache.maven.plugins</groupId>
19             <artifactId>maven-shade-plugin</artifactId>
20             <version>3.1.0</version>
21             <executions>
22                 <execution>
23                     <phase>package</phase>
24                     <goals>
25                         <goal>shade</goal>
26                     </goals>
27                 </execution>
28             </executions>
29         </plugin>
30     </plugins>
31 </build>
```

JsonObject In

Fat JAR

# Apache OpenWhisk - make jar

```
→ serverless-java-mini-workshop git:(master) mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:hello-world-java >-----
[INFO] Building hello-world-java 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ hello-world-java ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ hello-world-java ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ hello-world-java ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ hello-world-java ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ hello-world-java ---
[INFO] Surefire report directory: /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/surefire-reports

-----
T E S T S
-----
Running com.example.FunctionAppTest
Mar 05, 2019 2:37:00 PM com.example.FunctionApp main
INFO: invoked with params:
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.076 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ hello-world-java ---
[INFO] Building jar: /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/hello-world-java.jar
[INFO]
[INFO] --- maven-shade-plugin:3.1.0:shade (default) @ hello-world-java ---
[INFO] Including com.google.code.gson:gson:jar:2.8.2 in the shaded jar.
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/hello-world-java.jar with /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/target/hello-world-java-1.0-SNAPSHOT-shaded.jar
[INFO] Dependency-reduced POM written at: /Users/ulidder/Downloads/temp/java-openwhisk/serverless-java-mini-workshop/dependency-reduced-pom.xml
[INFO]
[INFO] BUILD SUCCESS
```

# Apache OpenWhisk - Runtime Exception

The screenshot shows a development environment with an IDE and a terminal window.

**IDE Editor:** The code editor displays `FunctionApp.java` with the following content:

```
31
32     JSONObject response = new JSONObject();
33     JsonPrimitive nameArg = args.getAsJsonPrimitive("name");
34     String result;
35     if (nameArg == null) {
36         result = "Hello! Welcome to OpenWhisk";
37     } else {
38         result = "Hello " + nameArg.getAsString() + " Welcome to OpenWhisk";
39     }
40     response.addProperty("greetings", result);
41
42     double me = 1/0; // Line 42 highlighted with a red box
43 
```

A warning message is shown at the bottom of the editor:

You, 2 minutes ago • Uncommitted changes

**Terminal:** The terminal window shows the command:

```
→ serverless-java-mini-workshop git:(master) ✘ ibmcloud fn action invoke helloJava -r
```

The output shows an error activation:

```
Activation: 'helloJava' (8ea2fb3161c14493a2fb3161c1949325)
[
    "2019-07-18T19:41:10.294892Z  stderr: java.lang.Arithme
    "2019-07-18T19:41:10.295020Z  cException: divide by zero",
    "2019-07-18T19:41:10.295065Z  stderr: at com.example.FunctionApp.main(FunctionApp.java:4
    "2019-07-18T19:41:10.295092Z  2)",
    "2019-07-18T19:41:10.295133Z  stderr: at sun.reflect.NativeMethodAccessorImpl.invoke0(Na
    "2019-07-18T19:41:10.295194Z  tive Method)",
    "2019-07-18T19:41:10.295221Z  stderr: at sun.reflect.NativeMethodAccessorImpl.invoke(Na
    "2019-07-18T19:41:10.295246Z  tiveMethodAccessorImpl.java:62)",
    "2019-07-18T19:41:10.295287Z  stderr: at sun.reflect.DelegatingMethodAccessorImpl.invoke(
    RunHandler.handle(Proxy.java:157)",
    "2019-07-18T19:41:10.295324Z  DelegatingMethodAccessorImpl.java:43)",
    "2019-07-18T19:41:10.295350Z  stderr: at java.lang.reflect.Method.invoke(Method.java:498
    "2019-07-18T19:41:10.295350Z  )",
    "2019-07-18T19:41:10.295350Z  stderr: at org.apache.openwhisk.runtime.java.action.JarLoa
    "2019-07-18T19:41:10.295350Z  der.invokeMain(JarLoader.java:76)",
    "2019-07-18T19:41:10.295350Z  stderr: at org.apache.openwhisk.runtime.java.action.Proxy$1
    "2019-07-18T19:41:10.295350Z  handle(Proxy.java:157)",
    "2019-07-18T19:41:10.295350Z  stderr: at com.sun.net.httpserver.Filter$Chain.doFilter(Fi
    "2019-07-18T19:41:10.295350Z  lter.java:79)",
    "2019-07-18T19:41:10.295350Z  stderr: at sun.net.httpserver.AuthFilter.doFilter(AuthFil
    "2019-07-18T19:41:10.295350Z  ter.java:83)",
    "2019-07-18T19:41:10.295350Z  stderr: at com.sun.net.httpserver.Filter$Chain.doFilter(Fi
```

# Apache OpenWhisk (py)

```
py hello-world.py x
```

```
1 def main(args):  
2     return {'greeting': 'hello world'}
```

Entry point

Dictionary

Dictionary

```
~/L/M/c/D/c/o/openwhisk-py ➤ wsk action create hellopy hello-world.py --kind python:3  
ok: created action hellopy
```

```
~/L/M/c/D/c/o/openwhisk-py ➤ wsk action list
```

actions	
/upkar.ibm.watson@gmail.com_dev/hellopy	private python:3
/upkar.ibm.watson@gmail.com_dev/webAction	private nodejs:8
/upkar.ibm.watson@gmail.com_dev/helloUI	private nodejs:8
/upkar.ibm.watson@gmail.com_dev/custom_docker_action2	private python:3.6
/upkar.ibm.watson@gmail.com_dev/hello-world/helloworld	private python:3.7
/upkar.ibm.watson@gmail.com_dev/skeleton	private blackbox
/upkar.ibm.watson@gmail.com_dev/upkar-skeleton-python3	private python:2
/upkar.ibm.watson@gmail.com_dev/hello-upkar-js	private nodejs:6
/upkar.ibm.watson@gmail.com_dev/myAction	private sequence
/upkar.ibm.watson@gmail.com_dev/BoulderAction	private java
/upkar.ibm.watson@gmail.com_dev/hello	private nodejs:6

# OpenWhisk

limit	description	configurable	unit	default
timeout	a container is not allowed to run longer than N milliseconds	per action	milliseconds	60000
memory	a container is not allowed to allocate more than N MB of memory	per action	MB	256
logs	a container is not allowed to write more than N MB to stdout	per action	MB	10
concurrent	no more than N activations may be submitted per namespace either executing or queued for execution	per namespace	number	100
minuteRate	no more than N activations may be submitted per namespace per minute	per namespace	number	120
codeSize	the maximum size of the actioncode	configurable, limit per action	MB	48
parameters	the maximum size of the parameters that can be attached	not configurable, limit per action/package/trigger	MB	1
result	the maximum size of the action result	not configurable, limit per action	MB	1

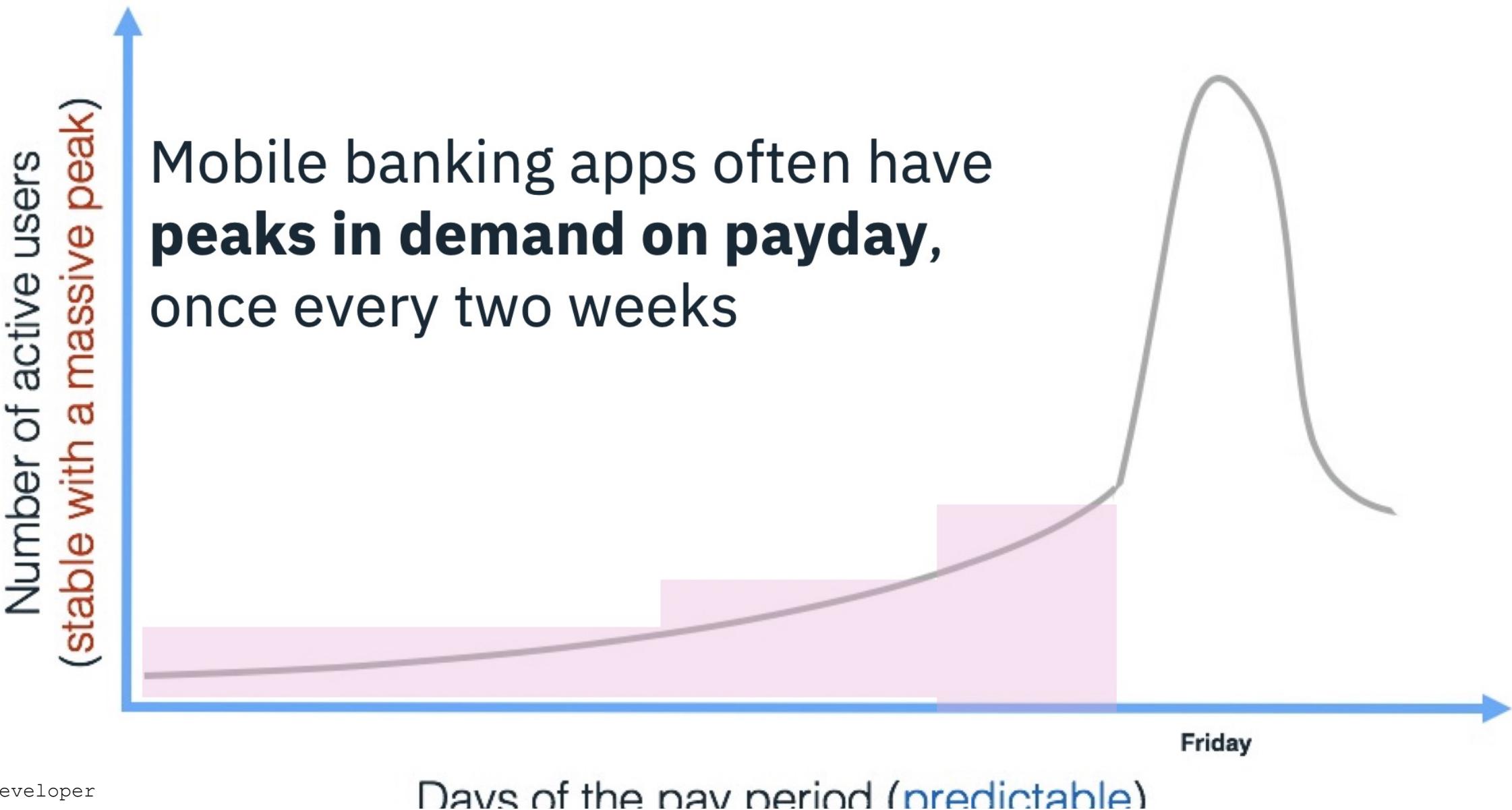
# IBM Cloud Function

Limit	Description	Default	Min	Max
<a href="#"><u>codeSize</u></a>	The maximum size of the action code in MB.	48	1	48
<a href="#"><u>concurrent</u></a>	No more than N activations can be submitted per namespace either executing or queued for execution.	1000	1	1000*
<a href="#"><u>logs</u></a>	A container is not allowed to write more than N MB to stdout.	10	0	10
<a href="#"><u>memory</u></a>	A container is not allowed to allocate more than N MB of memory.	256	128	2048
<a href="#"><u>minuteRate</u></a>	No more than N activations can be submitted per namespace per minute.	5000	1	5000*
<a href="#"><u>openulimit</u></a>	The maximum number of open files for an action.	1024	0	1024
<a href="#"><u>parameters</u></a>	The maximum size of the parameters that can be attached in MB.	1	0	1
<a href="#"><u>proculimit</u></a>	The maximum number of processes available to an action.	1024	0	1024
<a href="#"><u>result</u></a>	The maximum size of the action invocation result in MB.	1	0	1
<a href="#"><u>sequenceMaxActions</u></a>	The maximum number of actions that comprise a given sequence.	50	0	50*
<a href="#"><u>timeout</u></a>	A container is not allowed to run longer than N milliseconds.	60000	100	600000

# IBM Cloud Functions

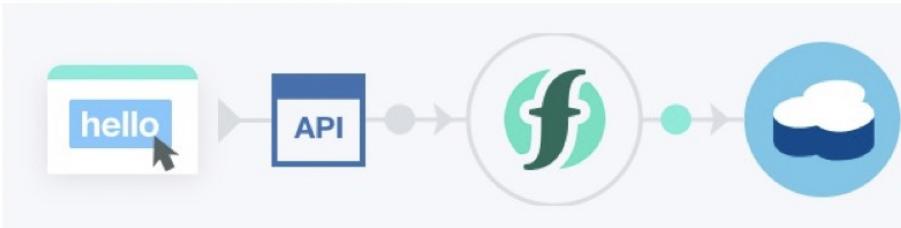
	Open source	Hosted service
Serverless engine	<a href="#">Apache OpenWhisk</a>	<a href="#">IBM Cloud Functions</a>
API Gateway	<a href="#">LoopBack</a>	<a href="#">IBM API Gateway</a>
Databases	<a href="#">Apache CouchDB</a> <a href="#">MySQL</a>	<a href="#">IBM Cloudant</a> <a href="#">IBM Compose</a>
Message streams	<a href="#">Apache Kafka</a>	<a href="#">IBM Message Hub</a>

# Example - Banking App



# More Examples

## Serverless Backends



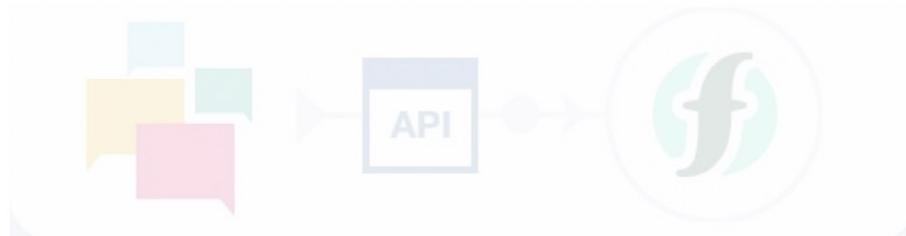
## Mobile Backend



## Data Processing



## Conversational Scenarios



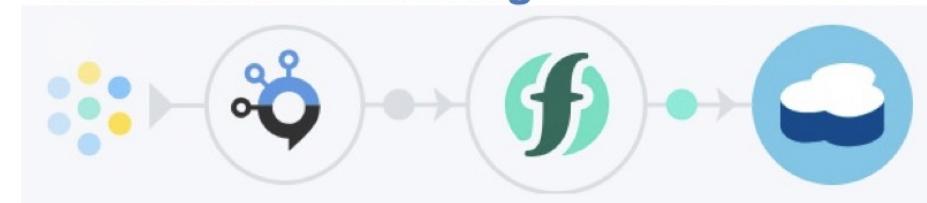
## Cognitive Data Processing



## IoT Ready



## Event Stream Processing



## Scheduled Tasks



# Apache OpenWhisk (java)

```
→ serverless-java-mini-workshop git:(master) wsk action list  
actions
```

```
/upkar.ibm.watson.6@gmail.com_dev/my-python-bot-action  
/upkar.ibm.watson.6@gmail.com_dev/hackerdojobot  
/upkar.ibm.watson.6@gmail.com_dev/hackerdojo-hello  
/upkar.ibm.watson.6@gmail.com_dev/getBotJSON  
/upkar.ibm.watson.6@gmail.com_dev/sfhtml5action  
/upkar.ibm.watson.6@gmail.com_dev/HiveModerator  
/upkar.ibm.watson.6@gmail.com_dev/hellohive  
/upkar.ibm.watson.6@gmail.com_dev/ibmWatsonModerator  
/upkar.ibm.watson.6@gmail.com_dev/upkar-action  
/upkar.ibm.watson.6@gmail.com_dev/moderator-action  
/upkar.ibm.watson.6@gmail.com_dev/womenintech  
/upkar.ibm.watson.6@gmail.com_dev/WatsonModerator  
/upkar.ibm.watson.6@gmail.com_dev/bot-action
```

```
private python:3.7  
private python:3.7  
private nodejs:10  
private nodejs:10  
private nodejs:6  
private python:3.7  
private nodejs:6  
private python:3.7  
private nodejs:6  
private python:3.7  
private nodejs:6  
private python:3.7  
private python:3.7
```

1

```
→ serverless-java-mini-workshop git:(master) wsk action create helloJava target/hello-world-java.jar --main com.example.FunctionApp  
ok: created action helloJava
```

```
→ serverless-java-mini-workshop git:(master) wsk action invoke helloJava  
ok: invoked /_helloJava with id ad0755b737c94c5b8755b737c98c5b35  
→ serverless-java-mini-workshop git:(master) wsk action invoke helloJava --param name "Upkar Lidder"  
ok: invoked /_helloJava with id 3fcfd956377d34f558d956377d37f55ef  
→ serverless-java-mini-workshop git:(master) wsk action invoke -r helloJava
```

2

```
{  
    "greetings": "Hello! Welcome to OpenWhisk"  
}  
→ serverless-java-mini-workshop git:(master) wsk action invoke --result helloJava --param name "Upkar Lidder"  
{  
    "greetings": "Hello Upkar Lidder welcome to Dawscon!!!"  
}
```

3

```
→ serverless-java-mini-workshop git:(master) wsk activation list  
activations  
c6b6e7ea179745f0b6e7ea179715f06a helloJava  
13290db4acbc4343a90db4acbc034349 helloJava  
3fcfd956377d34f558d956377d37f55ef helloJava  
ad0755b737c94c5b8755b737c98c5b35 helloJava  
7928a66c6511456aa8a66c6511156a8b helloJava  
8806ee39d9c8429786ee39d9c8e297e7 helloJava  
ef335974d0084811b35974d00898117c helloJava  
c8c76a0f35174a53876a0f35175a5358 helloJava
```

4

# Apache OpenWhisk (tail log)

```
→ expenses wsk activation poll
Enter Ctrl-c to exit.
Polling for activation logs

Activation: 'helloJava' (ad0755b737c94c5b8755b737c98c5b35)
[
  "2019-03-05T22:46:16.483411123Z stderr: Mar 05, 2019 10:46:16 PM com.example.FunctionApp main",
  "2019-03-05T22:46:16.483437074Z stderr: INFO: invoked with params:"
]

Activation: 'helloJava' (3fcd956377d34f558d956377d37f55ef)
[
  "2019-03-05T22:46:25.568213011Z stderr: Mar 05, 2019 10:46:25 PM com.example.FunctionApp main",
  "2019-03-05T22:46:25.568233909Z stderr: INFO: invoked with params:"
]

Activation: 'helloJava' (13290db4acbc4343a90db4acbc034349)
[
  "2019-03-05T22:46:30.792737853Z stderr: Mar 05, 2019 10:46:30 PM com.example.FunctionApp main",
  "2019-03-05T22:46:30.792770166Z stderr: INFO: invoked with params:"
]

Activation: 'helloJava' (c6b6e7ea179745f0b6e7ea179715f06a)
[
  "2019-03-05T22:46:38.218979004Z stderr: Mar 05, 2019 10:46:38 PM com.example.FunctionApp main",
  "2019-03-05T22:46:38.218998834Z stderr: INFO: invoked with params:"
]
```

# Apache OpenWhisk (js)

```
→ javascript-hello wsk action invoke hellojs
ok: invoked /_/hellojs with id 40be998f7fea4aaebe998f7feacaaef7
```

1

```
→ javascript-hello wsk action invoke hellojs -r
{
  "greeting": "hello from SFHTML5"
}
```

2

```
→ javascript-hello wsk action invoke hellojs -p name "Upkar Lidder"
ok: invoked /_/hellojs with id e27d4c77de414293bd4c77de41e29376
```

3

```
→ javascript-hello wsk activation list hellojs
activations
```

```
e27d4c77de414293bd4c77de41e29376 hellojs
edcddc28c59043598ddc28c590c3599d hellojs
40be998f7fea4aaebe998f7feacaaef7 hellojs
9a95abe401b94b5195abe401b9fb5132 hellojs
ae349d738a4f4d42b49d738a4fc4237 hellojs
e23976d21491453bb976d21491f53bdc hellojs
2ceabee51a5e4a5eaabee51a5e4a5e67 hellojs
f7a70f8f92174d2da70f8f92170d2dbf hellojs
e69ddd58e9b041719ddd58e9b0517183 hellojs
```

4

```
→ javascript-hello wsk activation poll
Enter Ctrl-c to exit.
Polling for activation logs
```

```
Activation: 'hellojs' (40be998f7fea4aaebe998f7feacaaef7)
[]
```

5

```
Activation: 'hellojs' (edcddc28c59043598ddc28c590c3599d)
[]
```

```
Activation: 'hellojs' (e27d4c77de414293bd4c77de41e29376)
[]
```

# Apache OpenWhisk (py)

```
~/L/M/c/D/c/o/openwhisk-py ➤ wsk action invoke hellopy -r
{
  "greeting": "hello world"
}
~/L/M/c/D/c/o/openwhisk-py ➤ wsk action invoke hellopy
ok: invoked /_hellopy with id c6d989394b9347ec9989394b93c7ec91
~/L/M/c/D/c/o/openwhisk-py ➤ wsk activation list hellopy
activations
c6d989394b9347ec9989394b93c7ec91 hellopy
6ef945a768674f39b945a768674f390a hellopy
d7f4d60804b748d0b4d60804b7d8d08a hellopy
ece76f21cc33427aa76f21cc33f27ab5 hellopy
e440649928354af180649928351af1e9 hellopy
e866a6fad87f4940a6a6fad87f29406c hellopy
83ed616f013f4899ad616f013fa899a3 hellopy
2a71ab1e6f534712b1ab1e6f53b712e2 hellopy
dc4ba6c6eb00470d8ba6c6eb00e70d2e hellopy
➤ wsk activation poll
an 8 20:07:02 2019
Enter Ctrl-c to exit.
Polling for activation logs

Activation: 'hellopy' (6ef945a768674f39b945a768674f390a)
[]
Activation: 'hellopy' (c6d989394b9347ec9989394b93c7ec91)
[]
```