

# OpenWhisk Tooling

David Liu

[David.liu@cn.ibm.com](mailto:David.liu@cn.ibm.com)



# Self Introduction

- David Liu
  - IBM Open Source Developer
  - 10+ years experience in software industry
  - OpenStack contributor
  - Apache OpenWhisk committer



# Agenda

- OpenWhisk on Bluemix
- OpenWhisk Cli
- OpenWhisk Catalog
- Wskdeploy
- OpenWhisk Debugger
- Other Useful Demo&Samples&Resources
- Q&A



# OpenWhisk on Bluemix

- 1. Start
- 2. Manage
- **3. Develop**
- **4. Monitor**
- **5. API**



# Overall console-Openwhisk on IBM Bluemix

The screenshot shows a Chrome browser window with three tabs: "Creating and invoking OpenWhisk", "仪表板 - IBM Bluemix", and "OpenWhisk". The main content area displays the "IBM Bluemix OpenWhisk" interface. The left sidebar has a dropdown menu set to "入门" (Getting Started) with options: 管理 (Management), 开发 (Development), 监视 (Monitoring), and API. The main content area features a large title "IBM OpenWhisk 入门" with a subtext explaining it's an open-source serverless platform. It includes two buttons: "在浏览器中开发" (Develop in Browser) and "下载 OpenWhisk CLI" (Download OpenWhisk CLI). A graphic of a satellite dish connected to a gear with stars is shown. At the bottom, a banner says "Save costs, scale and integrate."

# OpenWhisk on Bluemix – Development

The screenshot shows the IBM Bluemix OpenWhisk developer interface in a Chrome browser. The URL is [https://console.ng.bluemix.net/openwhisk/editor?env\\_id=ibm:yp:us-south](https://console.ng.bluemix.net/openwhisk/editor?env_id=ibm:yp:us-south). The top navigation bar includes '文档', '试用还剩 408 天', '昭刘's Account', and '美国南部 : david.liu@cn.ibm.com : diu'. The main menu on the left has tabs for '入门', '管理', '开发' (which is selected), '监视', and 'API'. The central workspace displays the 'new\_helloworld' operation. It shows three actions: 'pad\_new', 'process-change', and 'My First Sequence 1'. Below the sequence is a list of rules: 'On alarms alarm --using trigger...', 'kafka-inbound-rule', and 'log-change-rule'. The code editor shows the following JavaScript code:

```
1 /**
2  * main() 将在您运行此操作时被调用。
3  *
4  * @param OpenWhisk 操作接受单个参数,
5  * 必须是 JSON 对象。
6  *
7  * 在此案例中, params 变量将类似于:
8  * { "message": "xxxx" }
9  *
10 * @return 必须是 JSON 对象。
11 * 它将是此操作的输出。
12 *
13 */
14 */
15 function main(params) {
16     return { message: 'Hello World' };
17 }
```

At the bottom right of the code editor, it says '此代码已上线!' and '查看 REST 端点'. Below the code editor are buttons for '后续步骤:', '链接到序列', and '自动执行此操作'.

# OpenWhisk on Bluemix – Monitor

The screenshot shows the IBM Bluemix OpenWhisk Monitor interface. The left sidebar has a '监视' (Monitoring) tab selected. The main area displays:

- 活动摘要 (Activity Summary):** Shows operations by name, count, and average time.
  - helloPython: 4 activations, 331ms on average
  - kafka-inbound-rule
  - kafka-sequence: 3 activations, 758ms on average
  - kafka-trigger
  - main: 3 activations, 20 activations, 187ms
  - messageHubFeed: 517ms on average
- 活动日志 (Activity Log):** Lists recent log entries for wgetPython operations.
  - wgetPython 0909c0d6f0624c86abe17023... 2017-5-25 11:24:20  
797ms  
{ "url": "https://download.ted.c..."}
  - wgetPython e81e639c758d47d08f0a82ef... 2017-5-25 11:23:03  
1.5s  
{ "url": "https://download.ted...."}
  - wgetPython 517c5a18000f1a001240a317 2017-5-24 17:05:19
- 活动时间线 (Activity Timeline):** A bar chart showing activation counts over time.
  - 16:27:44: 13 activations
  - 21:24:16: 23 activations
  - 11:24:20: 1 activation
- 过滤选项 (Filter Options):** Includes time range (Recent 50), limit (All operations), and a checkbox for excluding triggers.



# OpenWhisk on Bluemix – Monitor(log sample)



A screenshot of a Chrome browser window. The title bar shows "Creating and invoking OpenWhisk" and "OpenWhisk - 仪表板". The address bar shows "https://console.ng.bluemix.net". The main content area displays a JSON log entry:

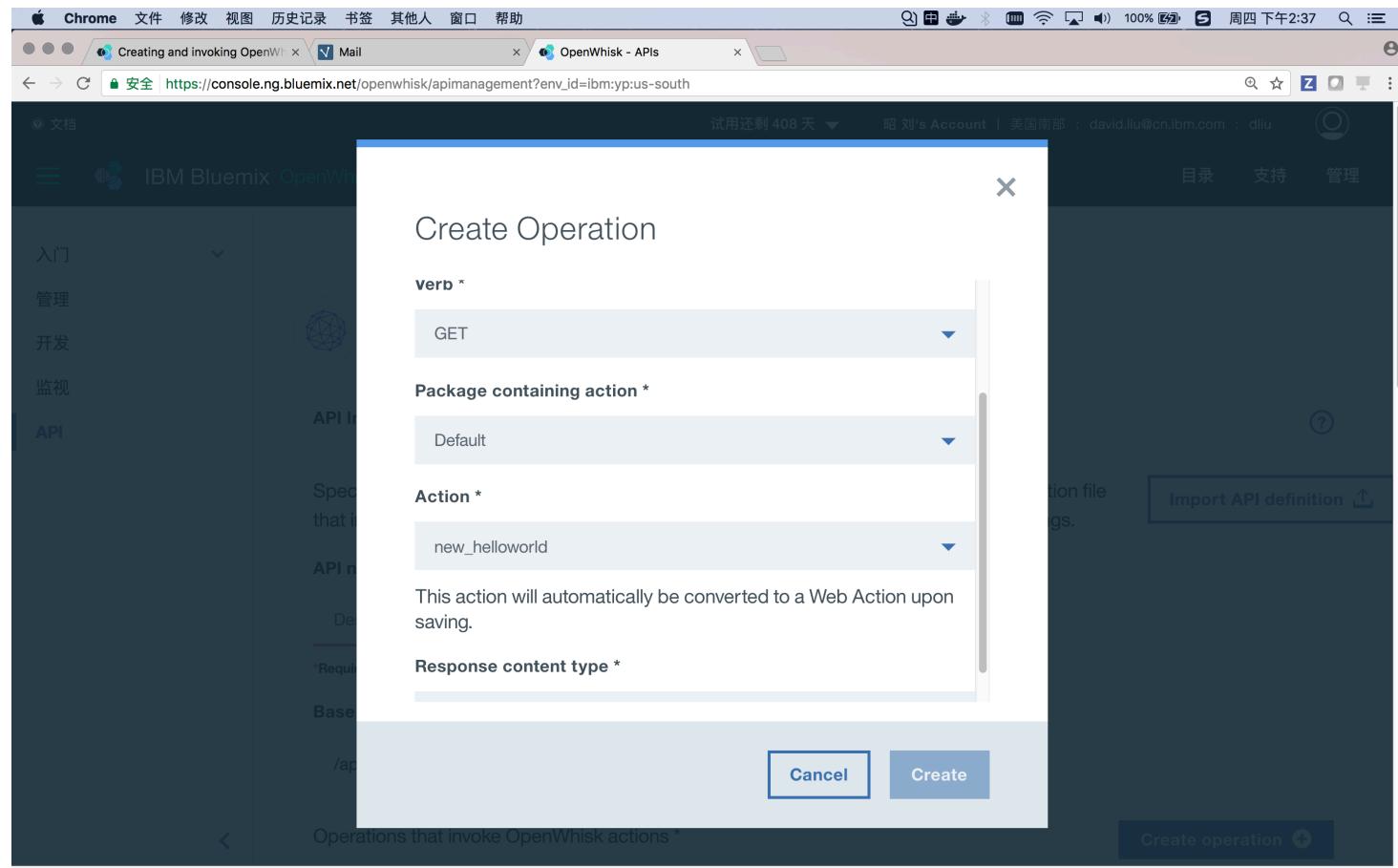
```
{"duration":115,"name":"main","subject":"david.liu@cn.ibm.com","activationId":"1955f1e87c2048c5b7e93554d7c26ca9","publish":false,"annotations":[{"key":"limits","value":{"timeout":60000,"memory":256,"logs":10}},{"key":"path","value":"david.liu@cn.ibm.com_dliu/main"}],"version":"0.0.1","response":{"result":{"extract":"China, officially the People's Republic of China (PRC), is a unitary sovereign state in East Asia and the world's most populous country, with a population of over 1."}},"success":true,"status":"success"},"end":1494057799027,"logs":[],"start":1494057798912,"namespace":"david.liu@cn.ibm.com_dliu"}
```



# OpenWhisk on Bluemix – API Creation

The screenshot shows a Chrome browser window on a Mac OS X desktop. The address bar displays the URL [https://console.ng.bluemix.net/openwhisk/apimanager?env\\_id=ibm:yp:us-south](https://console.ng.bluemix.net/openwhisk/apimanager?env_id=ibm:yp:us-south). The page title is "Creating and invoking OpenWhisk APIs". The main content area is titled "API Info" and contains fields for "API name" (with placeholder "Descriptive name for API") and "Base path for API" (with placeholder "/api"). A blue button labeled "Import API definition" is visible. Below these fields is a section for "Operations that invoke OpenWhisk actions" with a "Create operation" button. On the left sidebar, the "API" tab is selected, along with other tabs like "入门" (Getting Started), "管理" (Management), and "开发" (Development). The top navigation bar includes links for "目录" (Catalog), "支持" (Support), and "管理" (Management). The status bar at the bottom right shows the slide number "9".

# OpenWhisk on Bluemix – API Creation



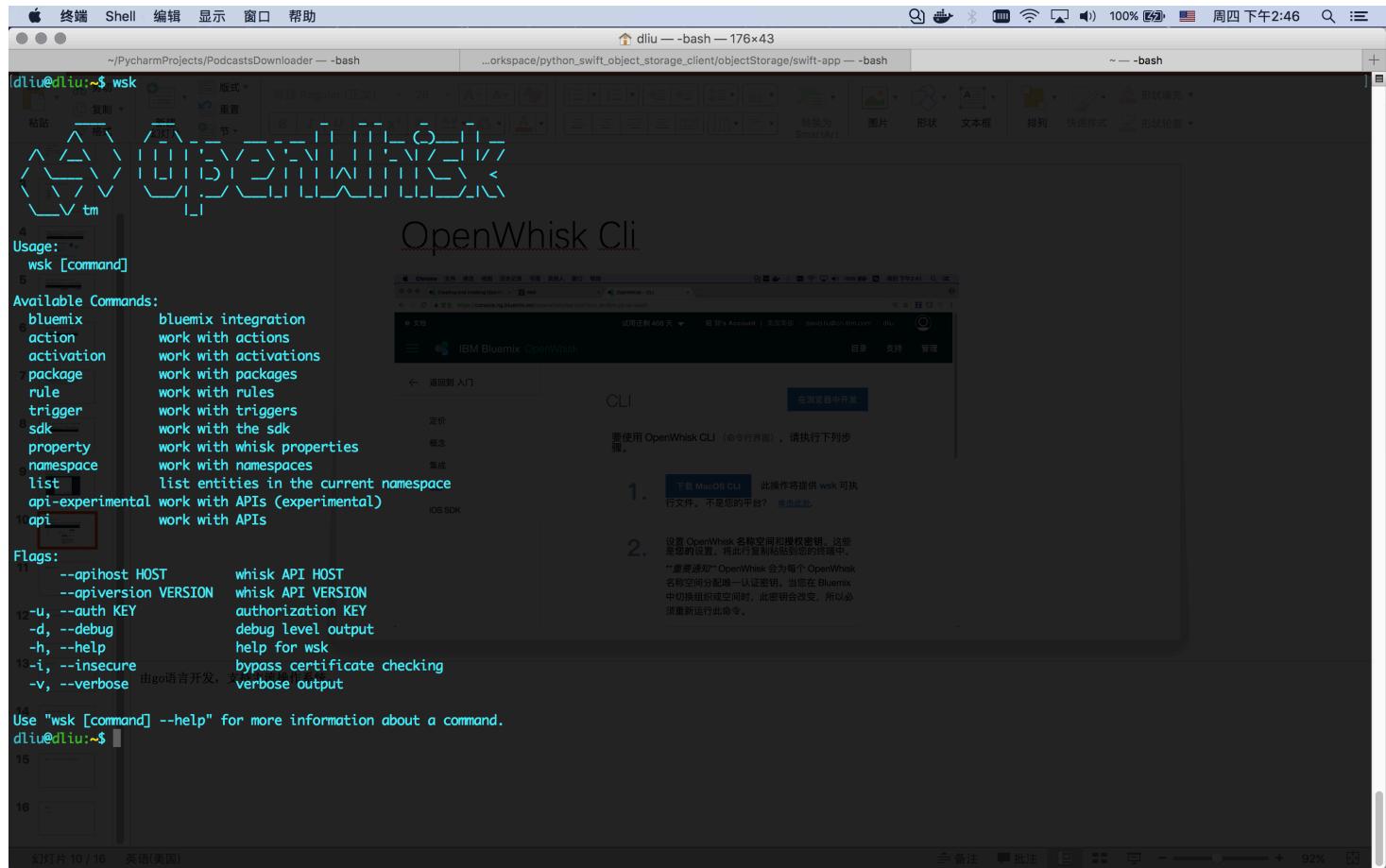
# OpenWhisk Cli

The screenshot shows a Chrome browser window displaying the IBM Bluemix OpenWhisk CLI documentation. The URL in the address bar is [https://console.ng.bluemix.net/openwhisk/learn/cli?env\\_id=ibm:yp:us-south](https://console.ng.bluemix.net/openwhisk/learn/cli?env_id=ibm:yp:us-south). The page title is "Creating and invoking OpenWhisk" and the tab title is "OpenWhisk - CLI". The main content area is titled "CLI" and contains instructions for using the OpenWhisk CLI. It includes two numbered steps: 1. Download MacOS CLI (with a note about other platforms) and 2. Set OpenWhisk namespace and auth key. A sidebar on the left lists "定价", "概念", "集成", "CLI" (which is selected), and "iOS SDK". A "在浏览器中开发" button is located in the top right corner of the main content area.

要使用 OpenWhisk CLI（命令行界面），请执行下列步骤。

1. [下载 MacOS CLI](#) 此操作将提供 `wsk` 可执行文件。不是您的平台？[单击此处](#).
2. 设置 OpenWhisk 名称空间和授权密钥。这些是您的设置。将此行复制粘贴到您的终端中。  
\*\*重要通知\*\* OpenWhisk 会为每个 OpenWhisk 名称空间分配唯一认证密钥。当您在 Bluemix 中切换组织或空间时，此密钥会改变，所以必须重新运行此命令。

# OpenWhisk Cli



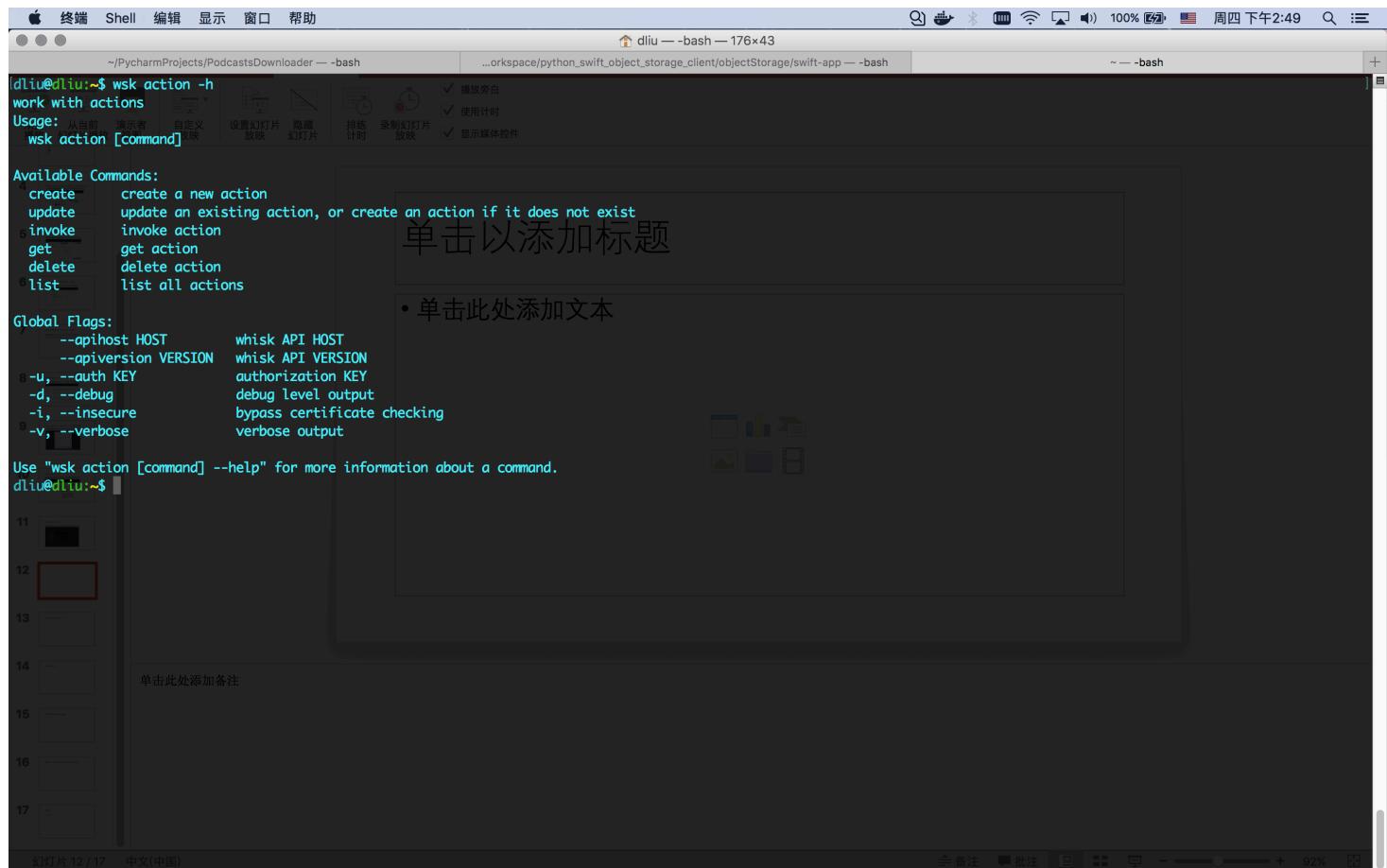
```
~/PycharmProjects/PodcastsDownloader -- bash ...orkspace/python_swift_object_storage_client/objectStorage/swift-app -- bash ~ -- bash
dliu@dliu:~$ wsk
Usage: wsk [command]
Available Commands:
  bluemix      bluemix integration
  action       work with actions
  activation   work with activations
  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-experimental work with APIs (experimental)
  api          work with APIs

Flags:
  --apihost HOST      whisk API HOST
  --apiversion VERSION  whisk API VERSION
  -u, --auth KEY      authorization KEY
  -d, --debug        debug level output
  -h, --help         help for wsk
  -i, --insecure     bypass certificate checking
  -v, --verbose      verbose output

Use "wsk [command] --help" for more information about a command.
dliu@dliu:~$
```



# OpenWhisk Cli



```
~/PycharmProjects/PodcastsDownloader -- bash ...orkspace/python_swift_object_storage_client/objectStorage/swift-app -- bash ~ -- bash
dliu@dliu:~$ wsk action -h
work with actions
Usage: 从当前 演示者 自定义 设置幻灯片 放映 幻灯片 拍摄 计时 录制幻灯片 放映 显示媒体控件
      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

Global Flags:
  --apihost HOST      whisk API HOST
  --apiversion VERSION whisk API VERSION
  -u, --auth KEY      authorization KEY
  -d, --debug          debug level output
  -i, --insecure       bypass certificate checking
  -v, --verbose        verbose output

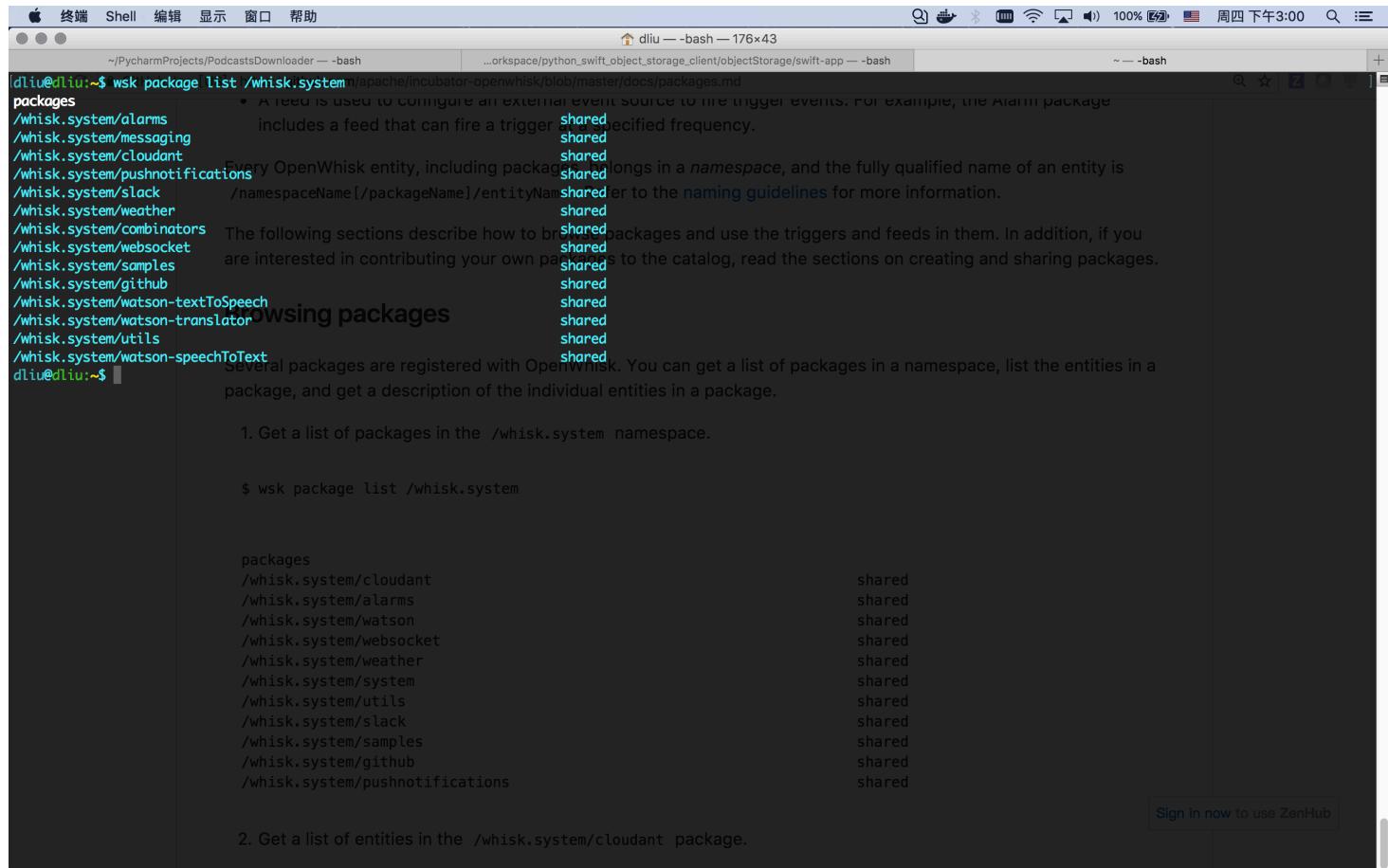
Use "wsk action [command] --help" for more information about a command.
dliu@dliu:~$
```

# OpenWhisk Cli

- <https://github.com/apache/incubator-openwhisk-client-go>
- <https://github.com/apache/incubator-openwhisk-cli>



# OpenWhisk Catalog



The screenshot shows a Mac OS X terminal window with two tabs open. The left tab shows the command `wsk package list /whisk.system` being run, and the right tab shows the output of this command. The output lists various packages under the `/whisk.system` namespace, each with a status of "shared". The packages listed are: alarms, messaging, cloudant, pushnotifications, slack, weather, combinator, websocket, samples, github, watson-textToSpeech, watson-translator, utils, and watson-speechToText.

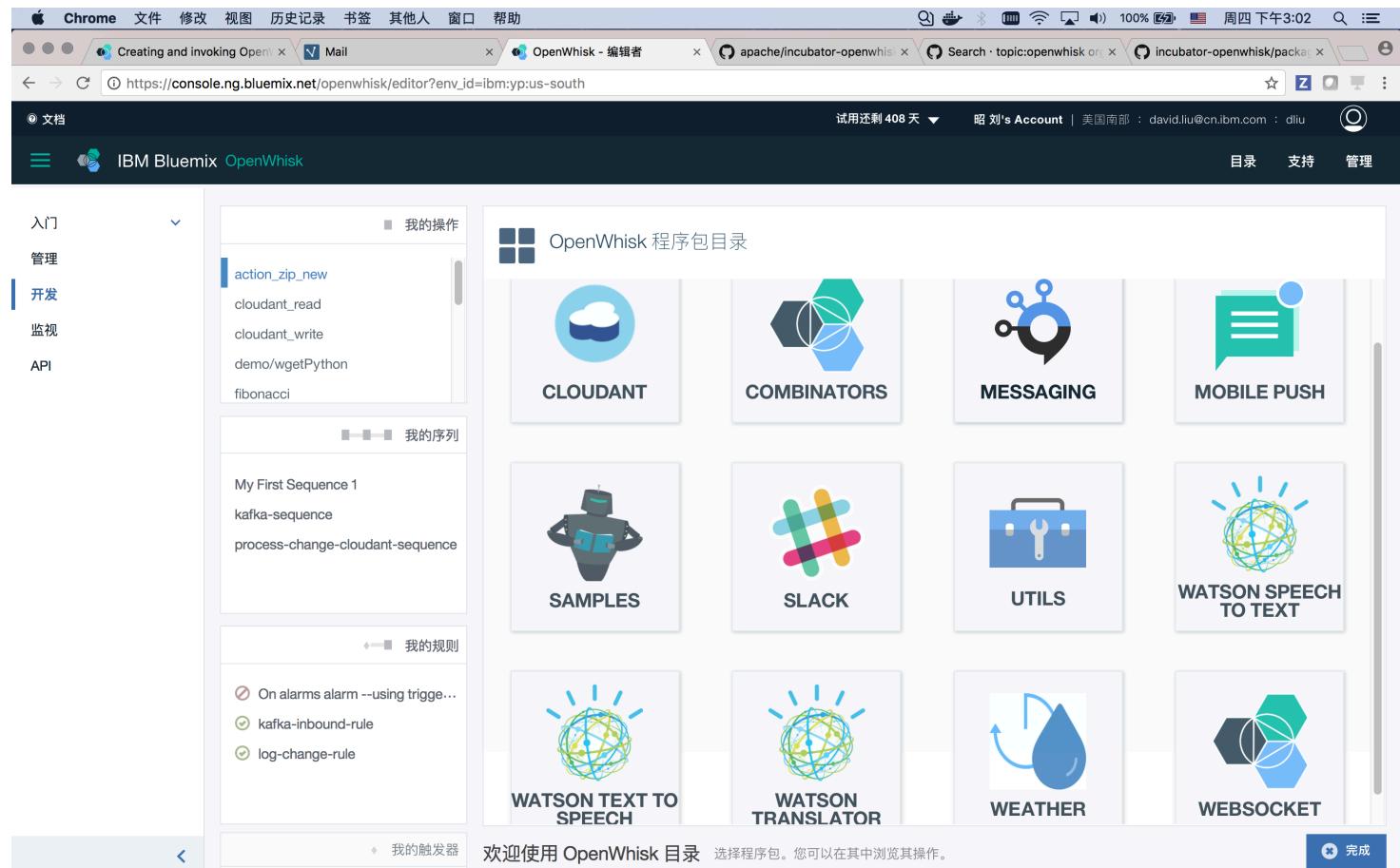
```
~/PycharmProjects/PodcastsDownloader -- bash ...orkspace/python_swift_object_storage_client/objectStorage/swift-app -- bash ~ -- bash
dliu@dliu:~$ wsk package list /whisk.system
packages
/wk.system/alarms shared
/wk.system/messaging shared
/wk.system/cloudant shared
/wk.system/pushnotifications shared
/wk.system/slack shared
/wk.system/weather shared
/wk.system/combinators shared
/wk.system/websocket shared
/wk.system/samples shared
/wk.system/github shared
/wk.system/watson-textToSpeech shared
/wk.system/watson-translator shared
/wk.system/utils shared
/wk.system/watson-speechToText shared
Several packages are registered with OpenWhisk. You can get a list of packages in a namespace, list the entities in a package, and get a description of the individual entities in a package.

1. Get a list of packages in the /whisk.system namespace.

$ wsk package list /whisk.system

packages
/wk.system/cloudant shared
/wk.system/alarms shared
/wk.system/watson shared
/wk.system/websocket shared
/wk.system/weather shared
/wk.system/system shared
/wk.system/utils shared
/wk.system/slack shared
/wk.system/samples shared
/wk.system/github shared
/wk.system/pushnotifications shared
2. Get a list of entities in the /whisk.system/cloudant package.
```

# OpenWhisk Catalog



# OpenWhisk Catalog

- <https://github.com/apache/incubator-openwhisk-catalog>



# Wskdeploy

- Target: Deploy Apache OpenWhisk Apps in just one click
- Use yaml file to define serverless app structure, entities, configs, deps etc.
- Go-lang
- Cobra
- Depends on  
<https://github.com/apache/incubator-openwhisk-client-go>



# Wskdeploy- yaml sample



The screenshot shows a Mac OS X desktop with a terminal window open. The terminal window has three tabs:

- Tab 1: incubator-openwhisk-wskdeploy — bash — 176x43
- Tab 2: ~ / go\_work / src / github.com / apache / incubator - openwhisk - wskdeploy — bash
- Tab 3: ~/workspace/openwhisk-tutorial — bash

The content of the second tab is:

```
dliu@dliu:~/go_work/src/github.com/apache/incubator-openwhisk-wskdeploy$ cat tests/usecases/helloworld/manifest.yaml
package:
  name: helloworld
  version: 1.0
  license: Apache-2.0
actions:
  hello:
    version: 2.0
    location: actions/greeting.js
    runtime: nodejs:6
    inputs:
      - name: Paul
        place: Boston
  helloworld:
    version: 1.0
    location: actions/greeting2.js
    runtime: nodejs:6
    inputs:
      - name: Bernie
        place: Vermont
triggers:
  trigger1:
    name: trigger1
  trigger2:
    name: trigger2
```



# Wskdeploy – UI

```
~/PycharmProjects/PodcastsDownloader -- bash ~/go_work/src/github.com/apache/incubator-openwhisk-wskdeploy -- bash ~/workspace/openwhisk-tutorial -- bash ... +  
dliu@dliu:~/go_work/src/github.com/apache/incubator-openwhisk-wskdeploy$ go run main.go -h  
A tool to deploy openwhisk packages with a manifest and/or deployment yaml file.  
wskdeploy without any commands or flags deploys openwhisk package in the current directory if manifest.yaml exists.  
  
Usage:  
wskdeploy [flags]  
wskdeploy [command]  
  
Available Commands:  
add      Add an action, feed, trigger or rule to the manifest  
init    Init helps you create a manifest file on OpenWhisk  
publish Publish a package to a registry  
report   Returns summary of what's been deployed on OpenWhisk in specific namespace  
undeploy Undeploy assets from OpenWhisk  
version  Print the version number of openwhisk-wskdeploy  
  
Flags:  
-a, --allow-defaults      allow defaults  
-i, --allow-interactive  allow interactive prompts (default true)  
--apihost string          whisk API HOST  
--apiversion VERSION     whisk API VERSION  
-u, --auth KEY            authorization KEY  
--config string           config file (default is $HOME/.wskdeploy.yaml)  
-d, --deployment string  path to deployment file  
-m, --manifest string    path to manifest file  
-p, --pathpath string    path to serverless project (default ".")  
-t, --toggle               Help message for toggle  
-v, --verbose              verbose output  
  
Use "wskdeploy [command] --help" for more information about a command.  
dliu@dliu:~/go_work/src/github.com/apache/incubator-openwhisk-wskdeploy$
```



# Wskdeploy

- <https://github.com/apache/incubator-openwhisk-wskdeploy>

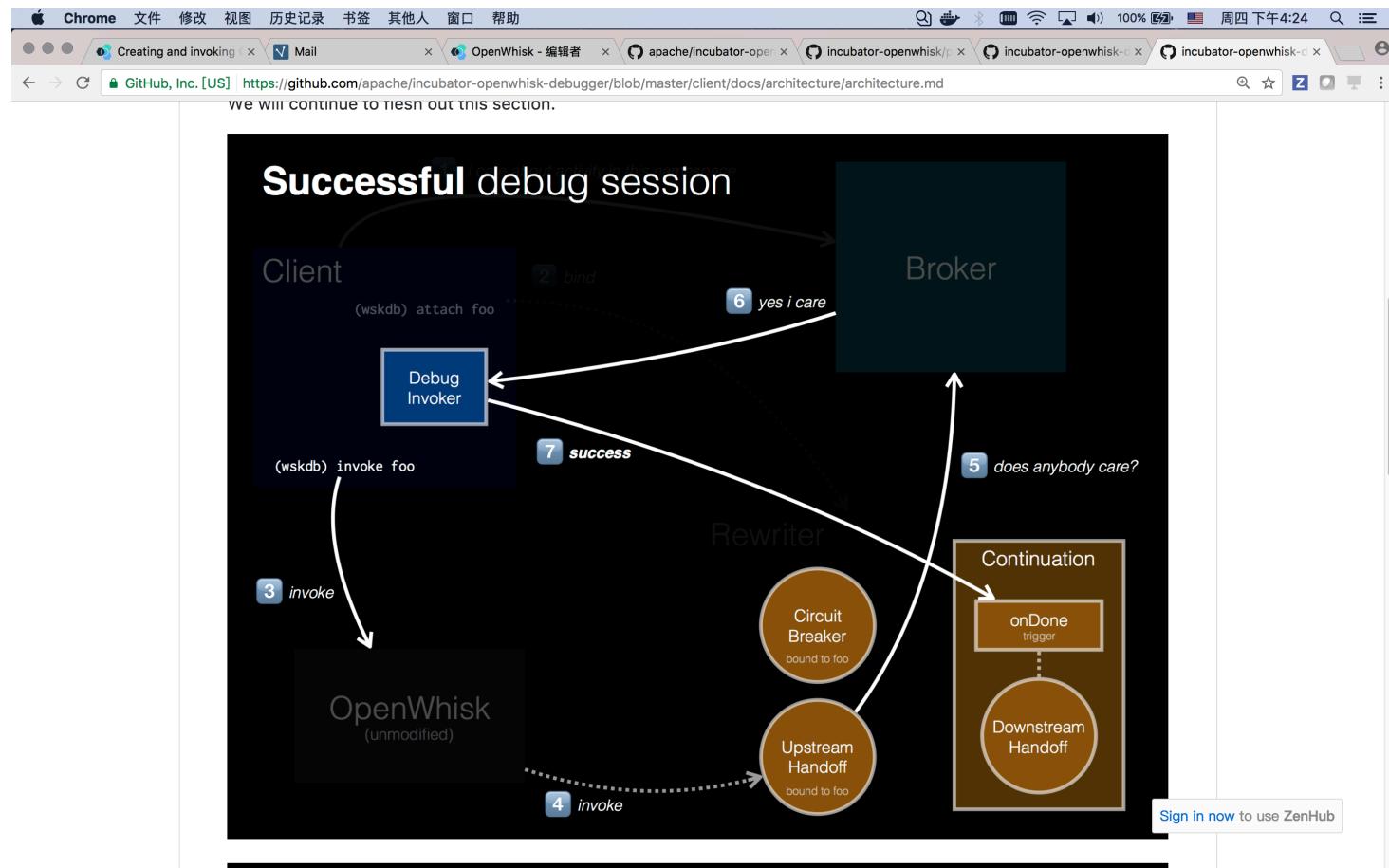


# OpenWhisk Debugger

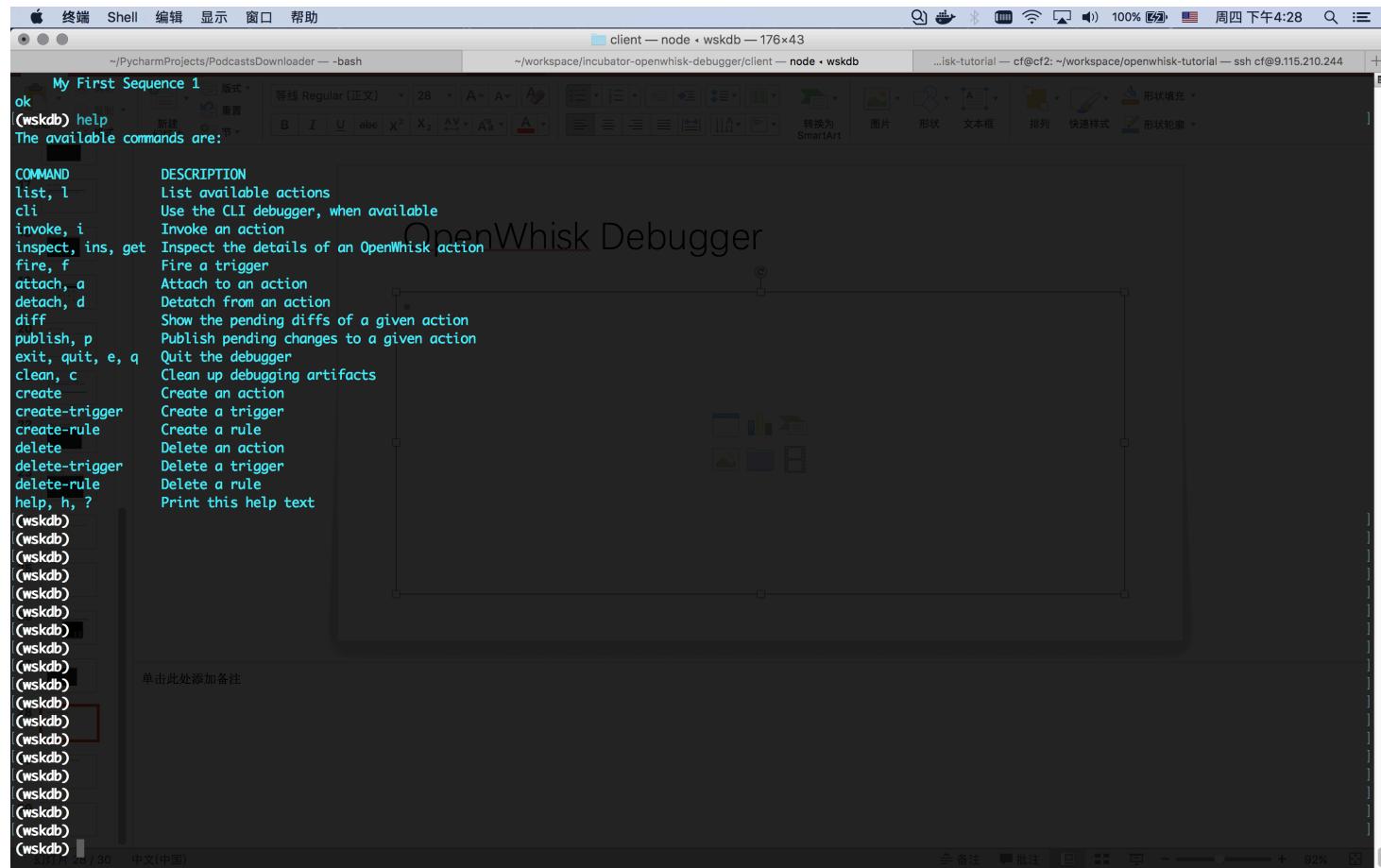
- Help debug OpenWhisk Actions
- Based on Nodejs, Debuggers are OpenWhisk Actions themselves
- Now support Nodejs, Python, Swift



# OpenWhisk Debugger



# OpenWhisk Debugger



A screenshot of a PyCharm terminal window titled "client — node • wskdb — 176x43". The terminal shows the output of the "help" command for the OpenWhisk Debugger. The help text is as follows:

```
My First Sequence 1
ok
(wskdb) help
The available commands are:
COMMAND      DESCRIPTION
list, l       List available actions
cli          Use the CLI debugger, when available
invoke, i     Invoke an action
inspect, ins, get Inspect the details of an OpenWhisk action
fire, f       Fire a trigger
attach, a     Attach to an action
detach, d     Detach from an action
diff          Show the pending diffs of a given action
publish, p    Publish pending changes to a given action
exit, quit, e, q Quit the debugger
clean, c      Clean up debugging artifacts
create        Create an action
create-trigger Create a trigger
create-rule   Create a rule
delete       Delete an action
delete-trigger Delete a trigger
delete-rule  Delete a rule
help, h, ?   Print this help text
(wskdb)
```



# OpenWhisk Debugger

- <https://github.com/apache/incubator-openwhisk-debugger>



# Other Useful Demos and Samples&Resources

- <https://github.com/apache/incubator-openwhisk-external-resources>
- <https://github.com/IBM/openwhisk-action-trigger-rule>
- <https://github.com/IBM/openwhisk-data-processing-message-hub>
- <https://github.com/lzbj/openwhisk-data-processing-message-hub-watson>
- openwhisk-package-alarms  
openwhisk-package-kafka  
openwhisk-package-push-notifications  
openwhisk-package-jira  
openwhisk-package-cloudant
- <https://github.com/apache/incubator-openwhisk-package-rss>
- <https://github.com/apache/incubator-openwhisk-tutorial>



# Other Useful Demos and Samples&Resources

the platform using NodeSchool toolchain.

## Sample Applications

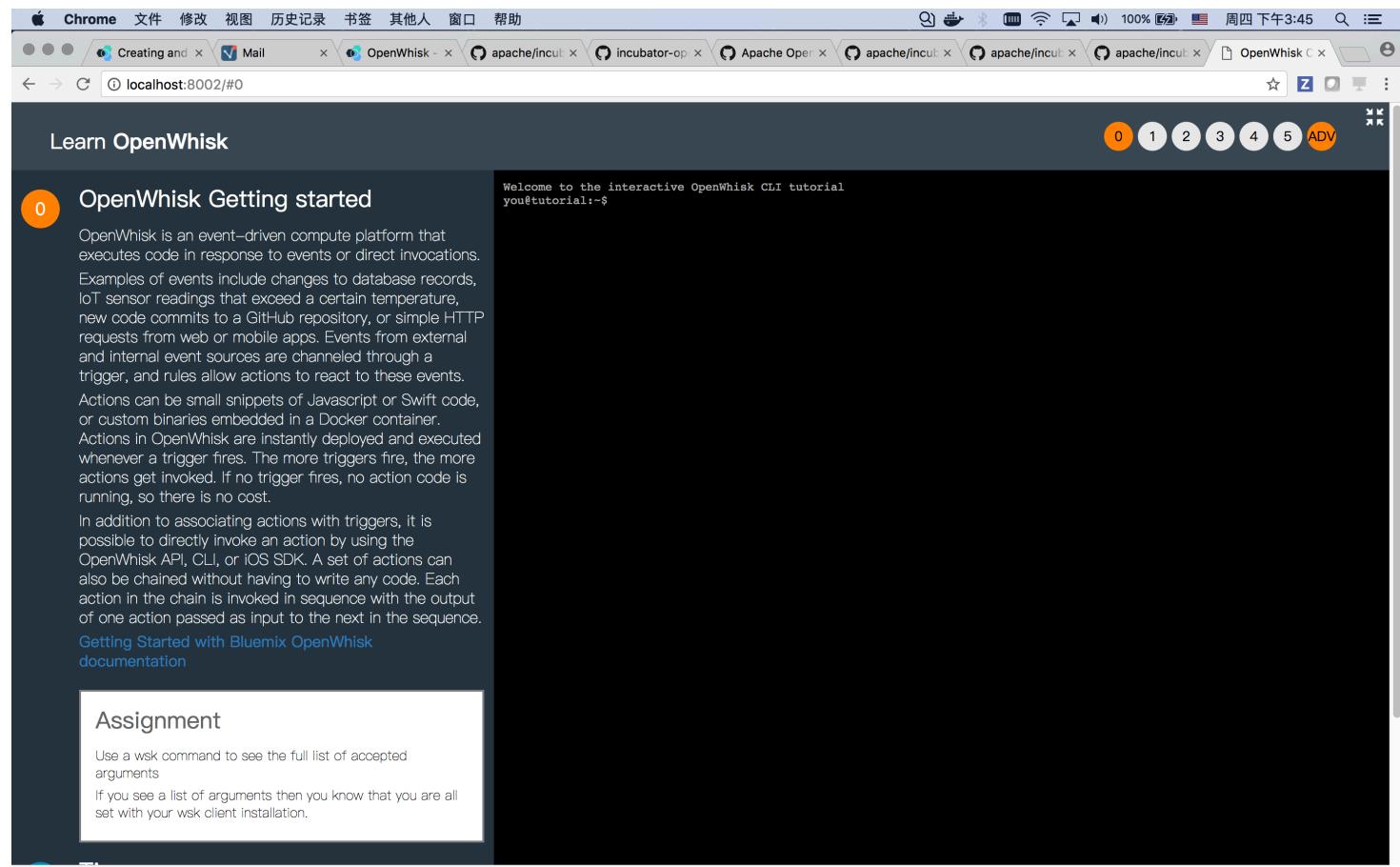
Sample open-source projects built using the OpenWhisk platform

- [Your first Action, Trigger, and Rule](#) - Simple Hello World style demo showing OpenWhisk actions, triggers, and rules.
- [Message Hub and Kafka Data Processing](#) - Create a serverless, event-driven application with OpenWhisk that executes code in response to messages or to handle streams of data records from Apache Kafka or IBM Message Hub.
- [OpenWhisk and Serverless APIs](#) - Create a serverless, event-driven application with OpenWhisk that executes code in response to HTTP REST API calls.
- [Cloudant Data Processing](#) - Create a serverless, event-driven application with OpenWhisk that executes code in response to database changes from Apache CouchDB or IBM Cloudant.
- [Transit IoT](#) - Comprehensive example of using OpenWhisk for IoT data processing that uses Docker, Node-RED, Message Hub (Kafka based), Object Storage, Spark and Bluemix Data Science Experience services for data analytics.
- [Emoting](#) - Sample "user feedback" app that uses OpenWhisk actions via REST API with pages hosted on GitHub and database in Cloudant.
- [Logistics Wizard](#) - Enterprise-grade sample application which leverages OpenWhisk and CloudFoundry to build 12-factor style applications. It is a smart supply chain management solution that aims to simulate an environment running an ERP system. Also see [related blog post](#).
- [Web Actions](#) - This sample shows how to use OpenWhisk to build a complete Web App. See [related blog post](#).
- [Dark vision](#) - Application that shows how to use OpenWhisk, Cloudant, Watson Visual Recognition, Object Storage to process video frames, tag and recognize scenes.
- [openwhisk-monitoring](#) - Example of using OpenWhisk to monitor HTTP server status.
- [openwhisk-publisher](#) - Hosting static sites with Jekyll, Object Storage and OpenWhisk.
- [skylink](#) - Connect and control a DJI drone aircraft over the Internet with OpenWhisk.

[Sign in now to use ZenHub](#)



# OpenWhisk Tutorial



# Q&A

- <http://openwhisk.org/>
- <https://github.com/apache?q=incubator-openwhisk>
- Thanks!





添加 **IBMOpenTech**  
请求入群  
与讲师同学互动



扫码填写  
本课程调查问卷

下讲预告  
6月22日晚8点  
**《用Apache OpenWhisk搭建微服务架构》**

如需体验**Apache OpenWhisk**  
请到[bluemix.net](#)注册并体验  
任何问题, 请微信咨询  
**IBMOpenTech**

