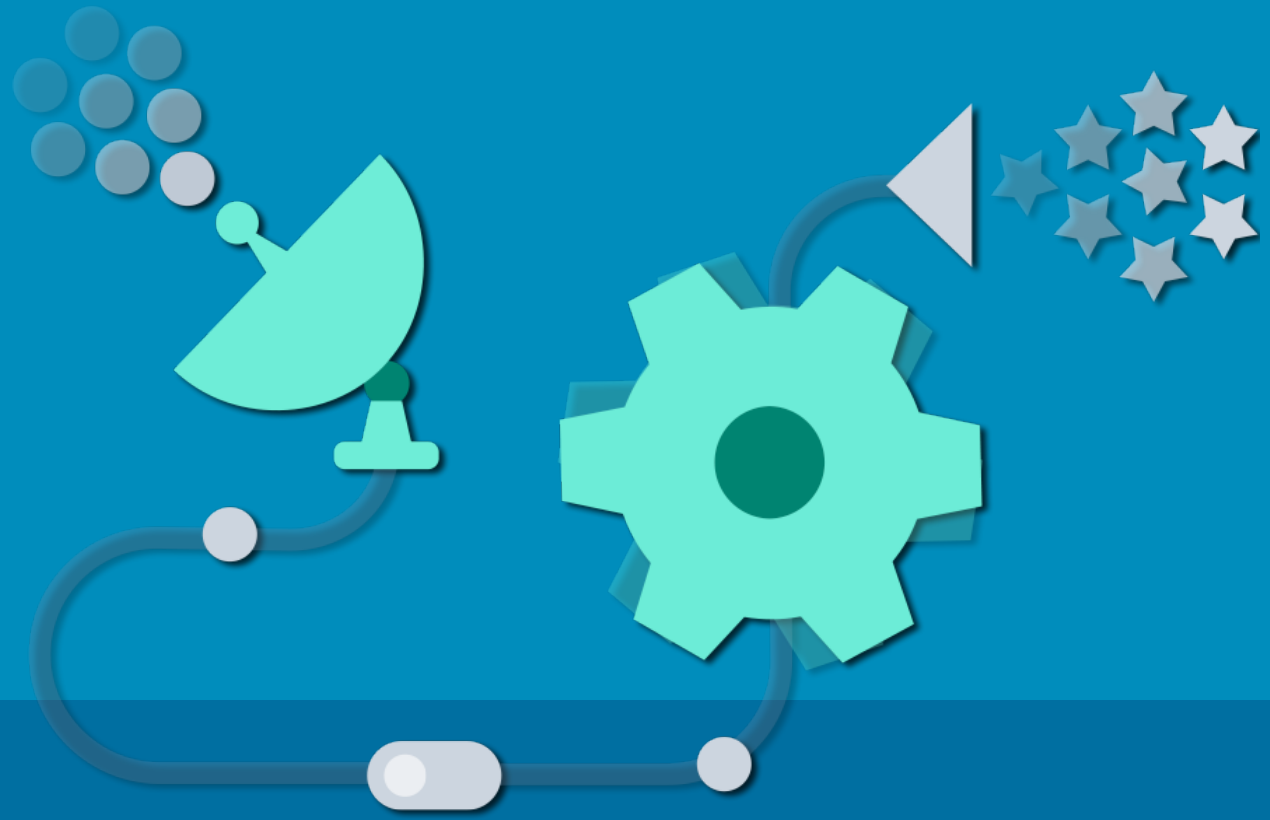
Serverless Computing

and the Apache OpenWhisk Project



http://openwhisk.org/

#openwhisk

https://openwhisk-team.slack.com/

Matt Rutkowski, *STSM, IBM Cloud Open Technologies*

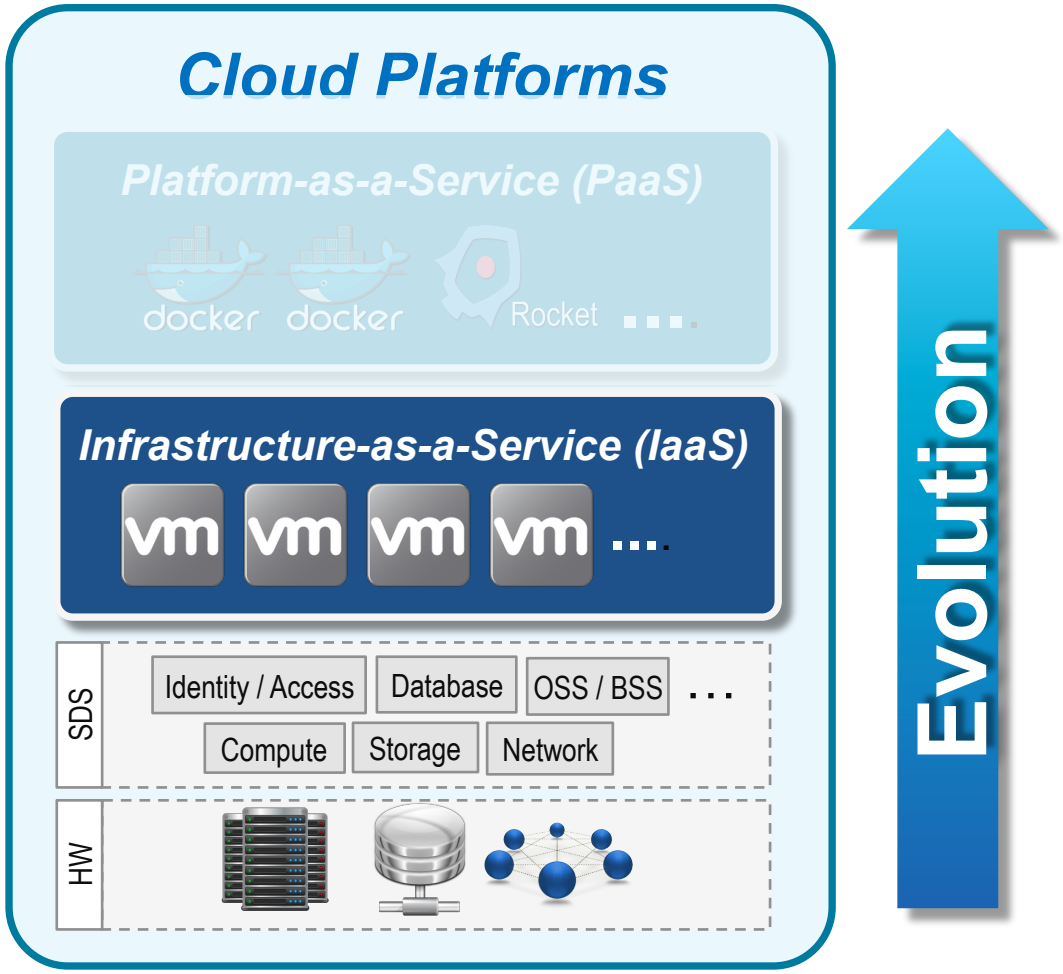
Session Topics

- What is Serverless computing?
- What is Apache OpenWhisk?
- OpenWhisk's Event-Driven Programming Model
- Apache OpenWhisk Project Overview and Architecture
- Apache OpenWhisk Community

What is Serverless Computing?

“No Servers, Just Code”

The Evolution of Cloud Platforms



Evolution

Cloud Datacenters

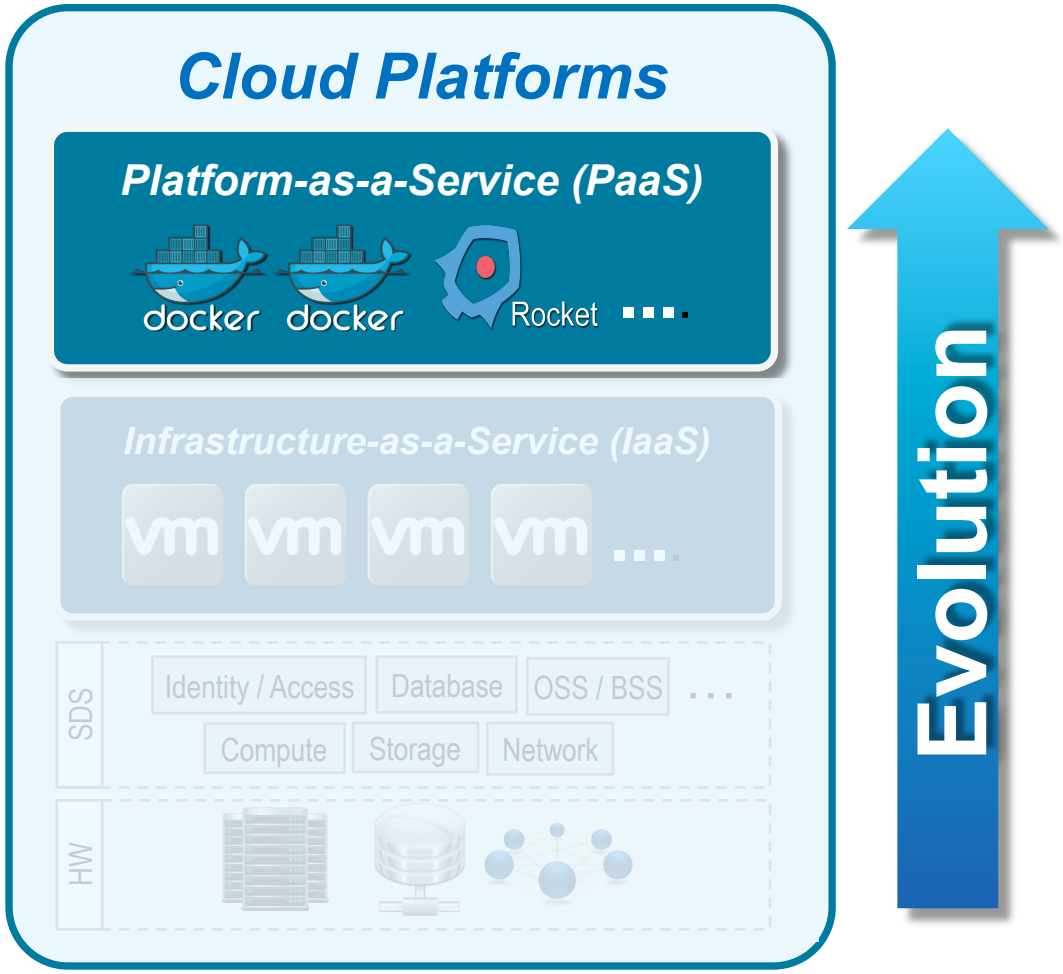
- **Software-Defined-Services (SDS)** *abstraction of*
  - ***Hardware***
    - *Compute, Network and Storage Hardware*
  - *"**Fabric" Services***
    - *Security, Data, Operational, Business, etc.*

IaaS = Virtual Machines (VMs)

- **Workloads: Heterogeneous applications**
  - *Tend to be tightly-coupled, Monolithic*
- **Configure** / **Manage** "Stacks" of applications
  - *Web/App Server, DB, logging, …*
- **Complex** Network & Storage Configurations
  - *IPs, Ports, Routes, SANs, Volumes, …*
- **Plan** / Predict for Scaling Needs
  - *CPUs, Memory, Load Balancing, …*

Lots of Configuration & Pre-planning for "worst case" Scaling & Failover

The Evolution of Cloud Platforms



PaaS = Containers

- **Workloads**: **Homogenous applications**
- Utilize abstract Configuration, Scheduling and ***Management*** frameworks
  - ***Kubernetes***, ***Apache Mesos***, …
- **Setup** Monitoring tools for Container Apps
  - ***Prometheus***, ***Logstash***, …
- **Configure** "Platform" services interactions:
  - ***Storage***, ***Access Control***, ***Key Mgmt.***, …

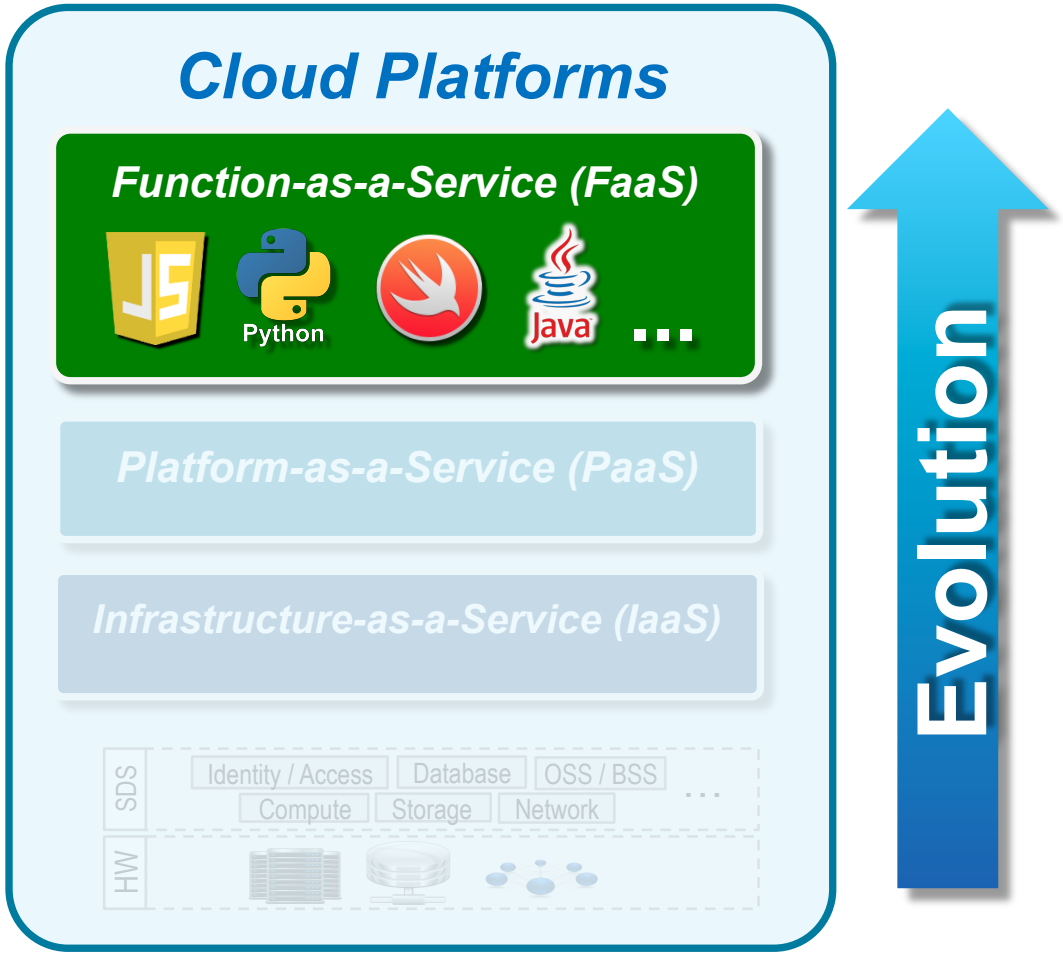
MicroServices *(often run in Containers)*

- Split large codebases into **small, standalone services**
- Single "**light weight**", RESTful endpoints or APIs
  - *"Protocol aware" (HTTP, HTTPS)*
- Built in groups or "suites" around "**resources**"
- Run inside MicroService "**Frameworks**" to handle
  - **Configuration**, logging, metrics, service registration & discovery, protocols, etc.
  - *e.g., Flask (Python), Spring (Java), Gizmo (Go)*

Still Lots of Configuration to work with Frameworks and Resources

The Evolution of Cloud Platforms

Serverless Computing is the next step for Cloud platforms



Evolution

Serverless = Functions

- **Workloads**: Simple, **single-tasked Functions**
- No "Back-end" Servers Configuration
  - ***Automatic scaling,*** *based upon load*
  - ***Driven by events,*** *and their data*
- Majority of Functions & Orchestration are "Front-end"
  - around workflows and tasks around the applications data … where
  - ***Developers Focus on ONLY writing Application and Business logic!***

There are still Servers!

- ***But they are a 'No-Op' for you!***
  - *Provider's DevOps teams configure, manage and assure Functions scale and run efficiently.*

No Configuration *of Servers, Only Pay for Compute time functions actually use*

Characteristics of Serverless

APACHE OpenWhisk™

Workloads: Data

Functions are:

- **"Data Driven"**
  - *Event **Input***
  - *Results **Output** data*
  - *often JSON*
- **All about "Front-end" resources**
  - *What's a Server? What's a Network?*
- **Protocol (route) agnostic**
  - HTTP? What is that?
- **Event-Driven**
  - *Run only when associated Event "Triggers" them*

Serverless Apps try to:

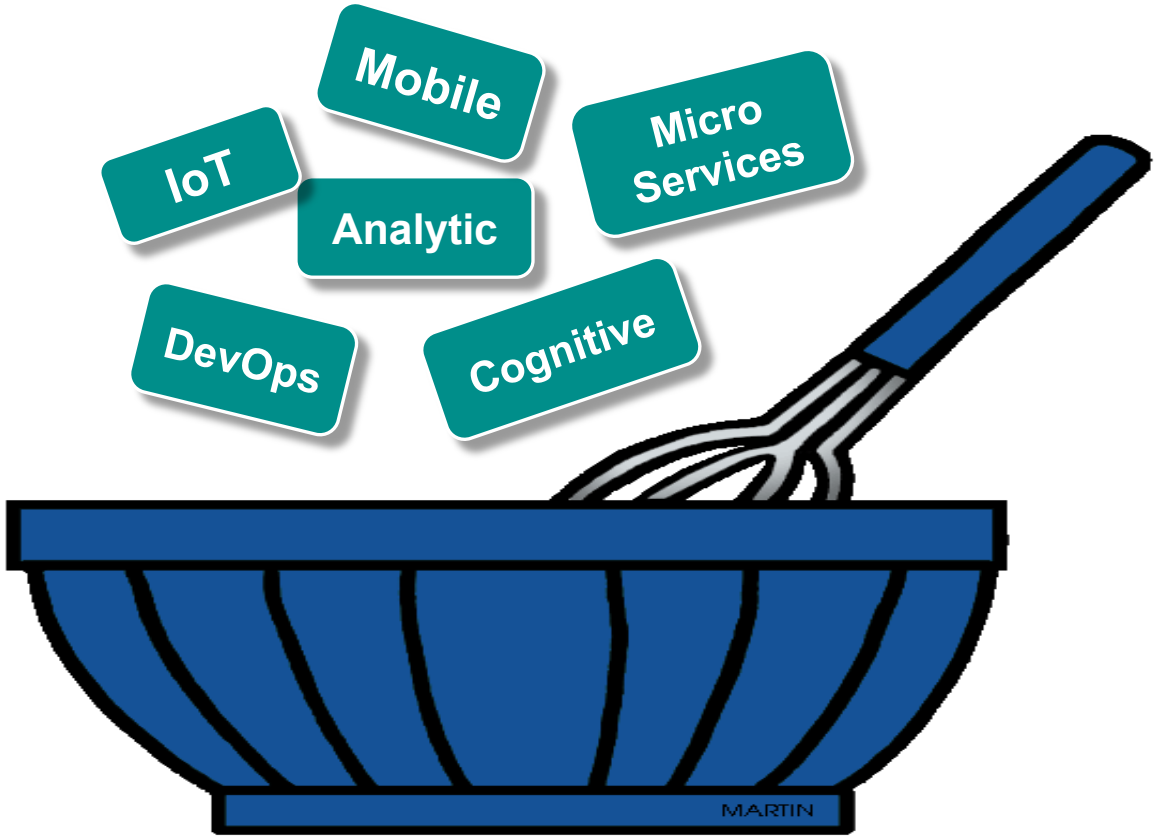
- **Maximize "stateless"**
  - *or "**pure**" functions*
  - *Process specific data & **return results quickly***
  - ***Do not block** / poll / wait for a a resource's state change (results)*
  - ***Optimistic** that Input data is correct, valid.*
- **Minimize "stateful" interactions**
  - *or "**impure**" functions*
  - *"Push" state (results) to "stateful" services, for example*
    - ***Data Stores / Message Queues***

What is Apache OpenWhisk?

whisk (v) : to move nimbly and quickly.

Apache OpenWhisk is an open source Serverless platform

Designed to connect a fast-moving, event-driven world



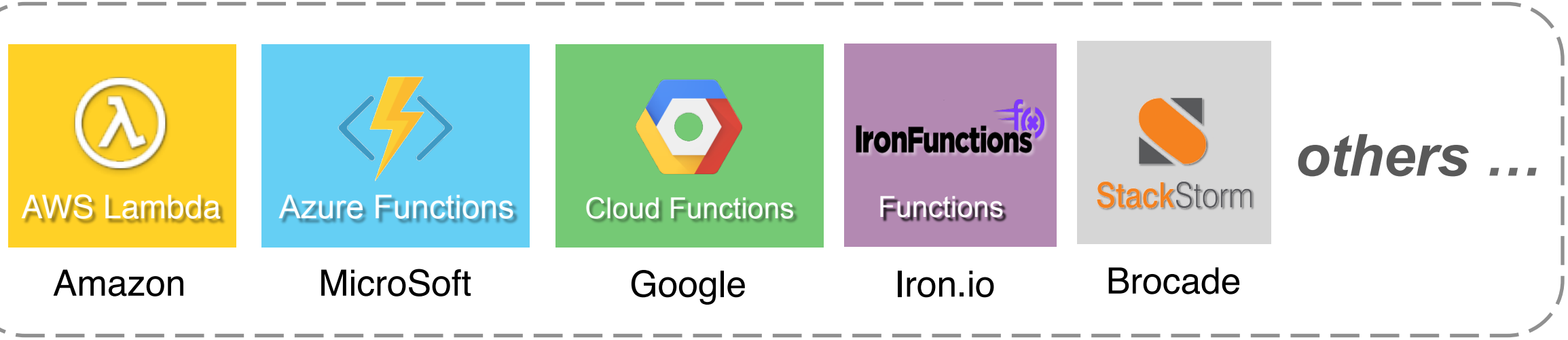
OpenWhisk provides a robust, scalable Serverless platform

*for **rapidly integrating** services and data from any domain using events that drive simple integration code.*

The name "Whisk" is also used for a kitchen utensil…

Developers, the "Chefs", can use OpenWhisk to quickly compose compelling applications, by "mixing" together interesting functions using event data

Many Cloud providers offer some form of Serverless framework



Apache OpenWhisk *offers:*

- **Apache Software Foundation** (ASF)
  - *True, community-driven open source (Apache 2 License)*
- **Proven on IBM's Bluemix Cloud Platform**
  - *Exact, same code in open source*
- **Highly Performant** *with strong, automated integration testing*
  - *< 10 millisecond schedule/load into runtime, results caching*

History of OpenWhisk

Early 2015 — *IBM Research begins coding*

Feb 2016 — *IBM debuts and open sources on GitHub (openwhisk.org) & announces*

Aug 2016 — *Adobe partners with IBM on API Gateway for OpenWhisk*

Nov 10, 2016 — *IBM Bluemix beta offering*

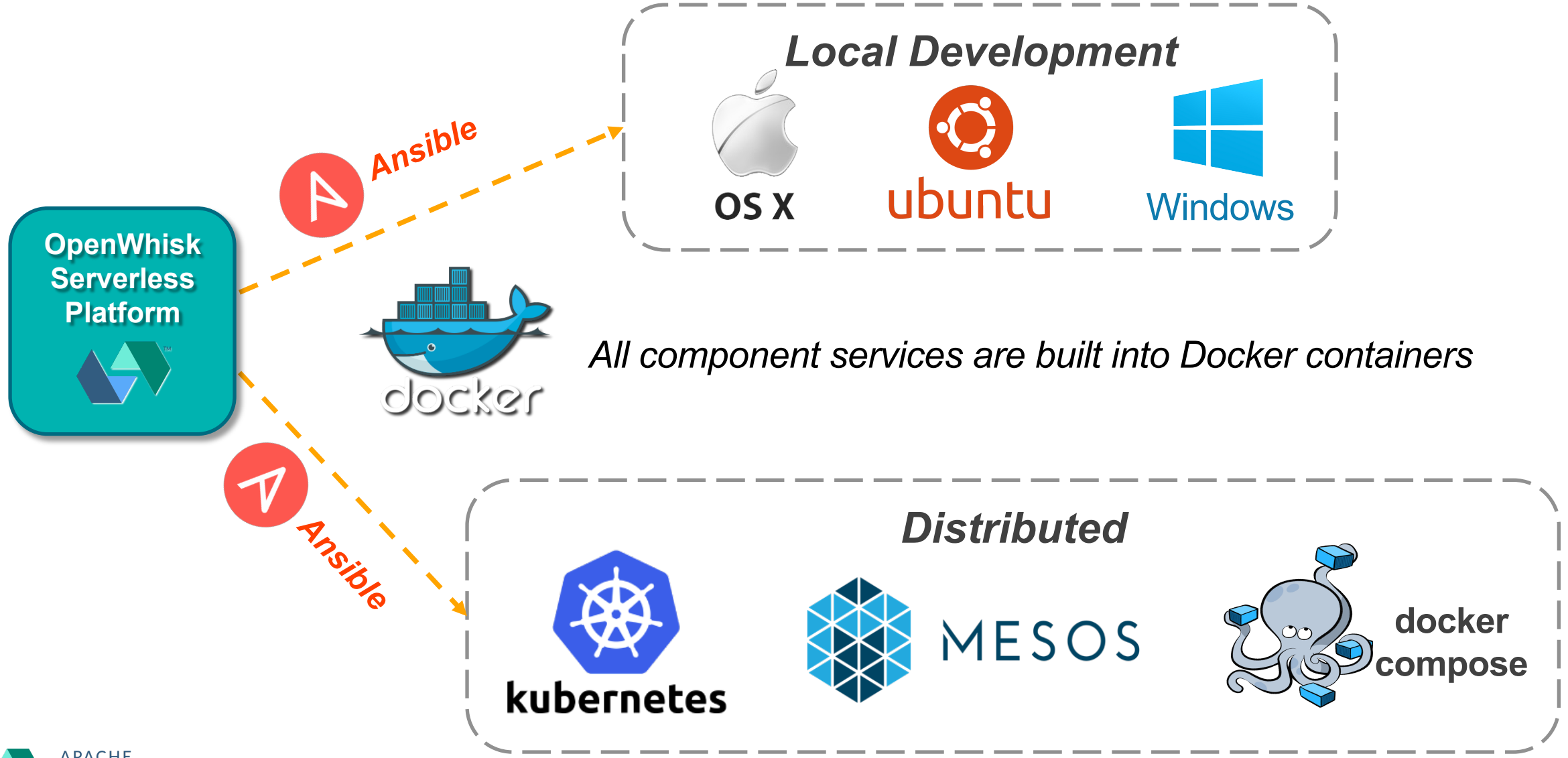
Nov 23, 2016 — *OpenWhisk Incubated at Apache Software Foundation*

Dec 14, 2016 — *General Availability (GA) on Bluemix*

Feb/Mar 2017 — *MessageHub (Kafka), Web Actions, VS Code*

May 2017 — *OpenWhisk's API Gateway integration GA*

Apache OpenWhisk has many deployment options



Event-Driven Programming Model

"performing the correct amount of work at the time needed"

Why an Event-Driven Programming Model?

Everything produces Events…

in the Natural World

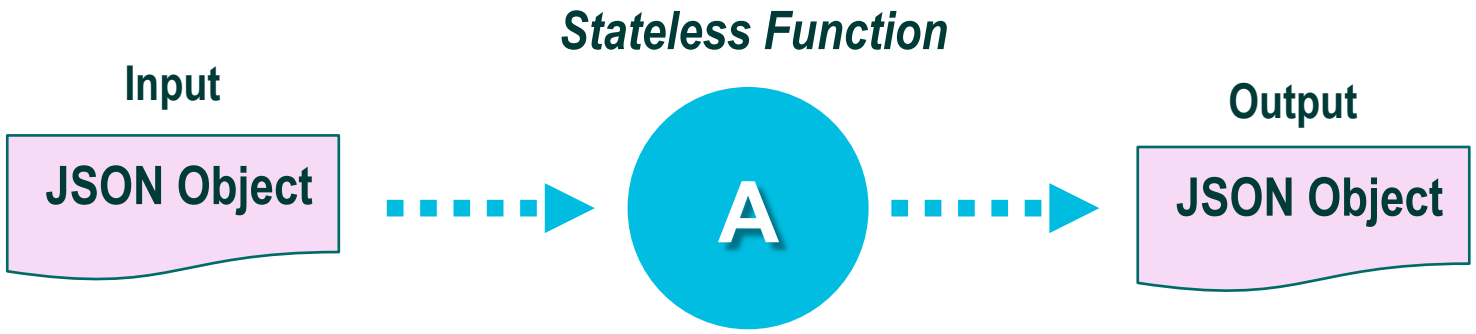
and in IT

OpenWhisk's Programming Model

APACHE OpenWhisk™

A Action :

A stateless, relatively short-running function invoked as an event handler.



Goal: *An Action's run time is measured in milliseconds,*

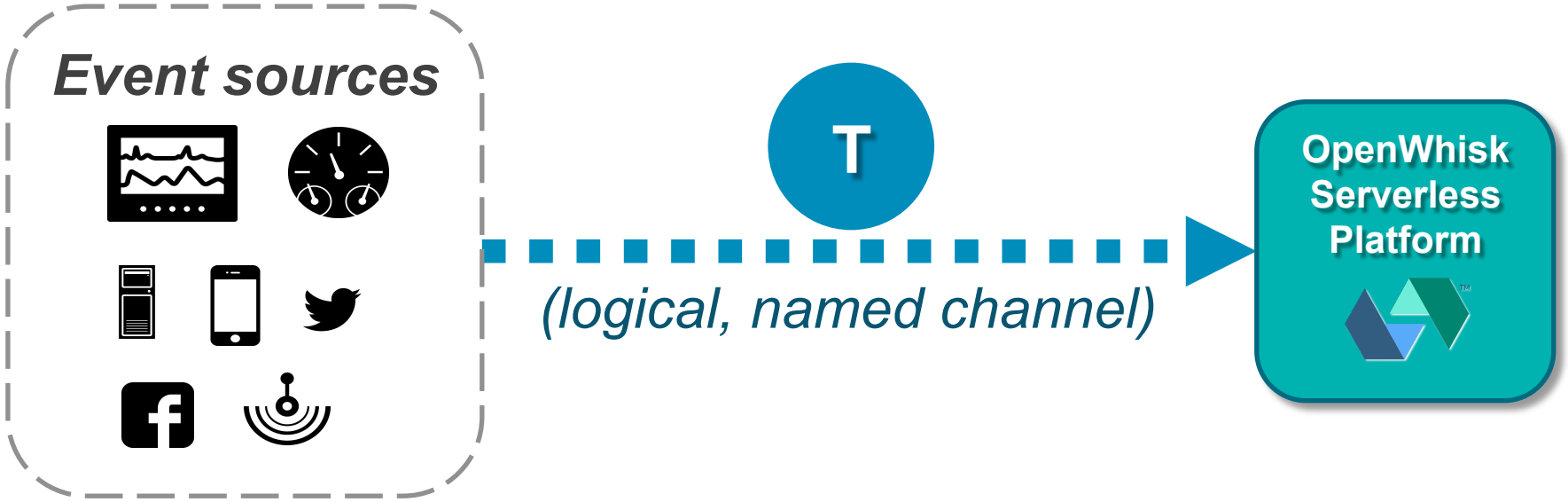
… well under the defaulted maximum of 5 minutes

OpenWhisk's Programming Model

T Trigger :

*The named channel for a **class of events**.*

Triggers represent the events (and their data) themselves without any concept of how they were generated.



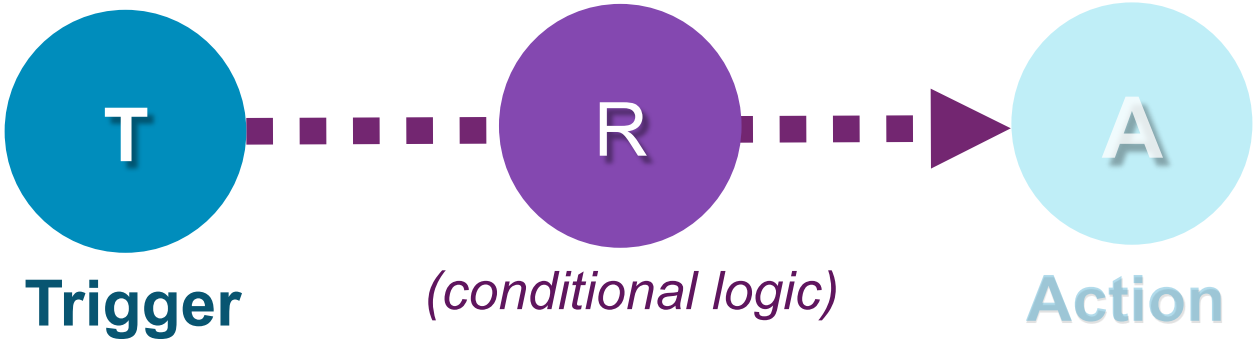
***Note**: In a pub-sub system, a trigger could be viewed as a message topic.*

OpenWhisk's Programming Model

APACHE OpenWhisk™

R Rule :

*A mapping from a **Trigger** to an **Action** which may contain simple conditional logic.*



***Note**: OpenWhisk evaluates incoming events (that belong to a **Trigger**) and invokes the assigned **Action** (event handler) associated by the **Rule**.*

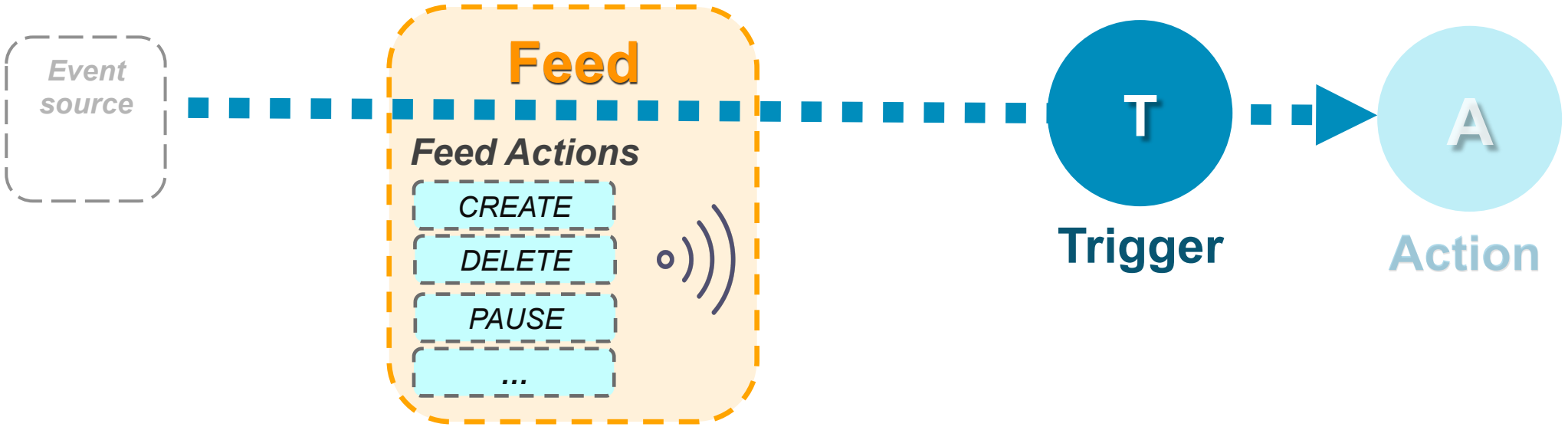
Programming Model

F Feed :

*A **Feed** manages the stream of Events from an external Event Source with some optional control operations called Feed Actions :*

- *CREATE, DELETE, PAUSE, UNPAUSE*

The running Feed "fires" events to an associated Trigger

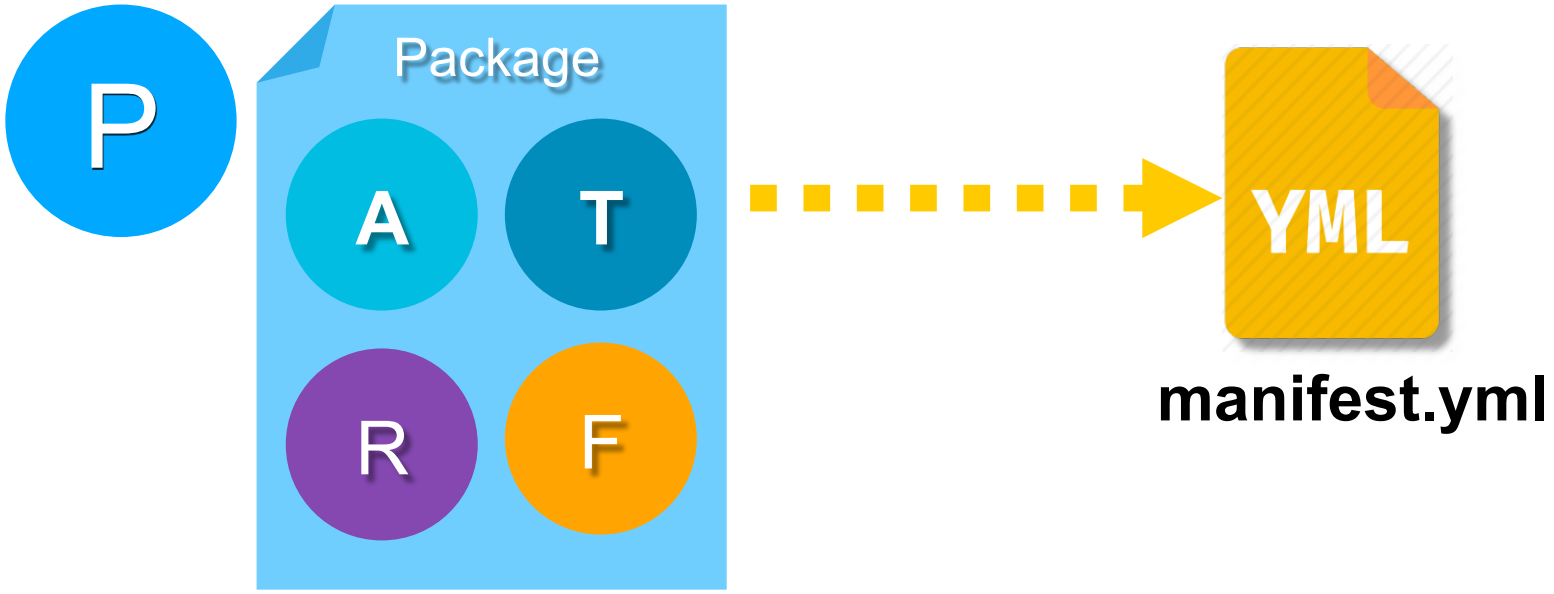


***Note**: **Feeds** can "Fire" **Triggers** without an Event Source such as the OpenWhisk **Alarm** package.*

Programming Model

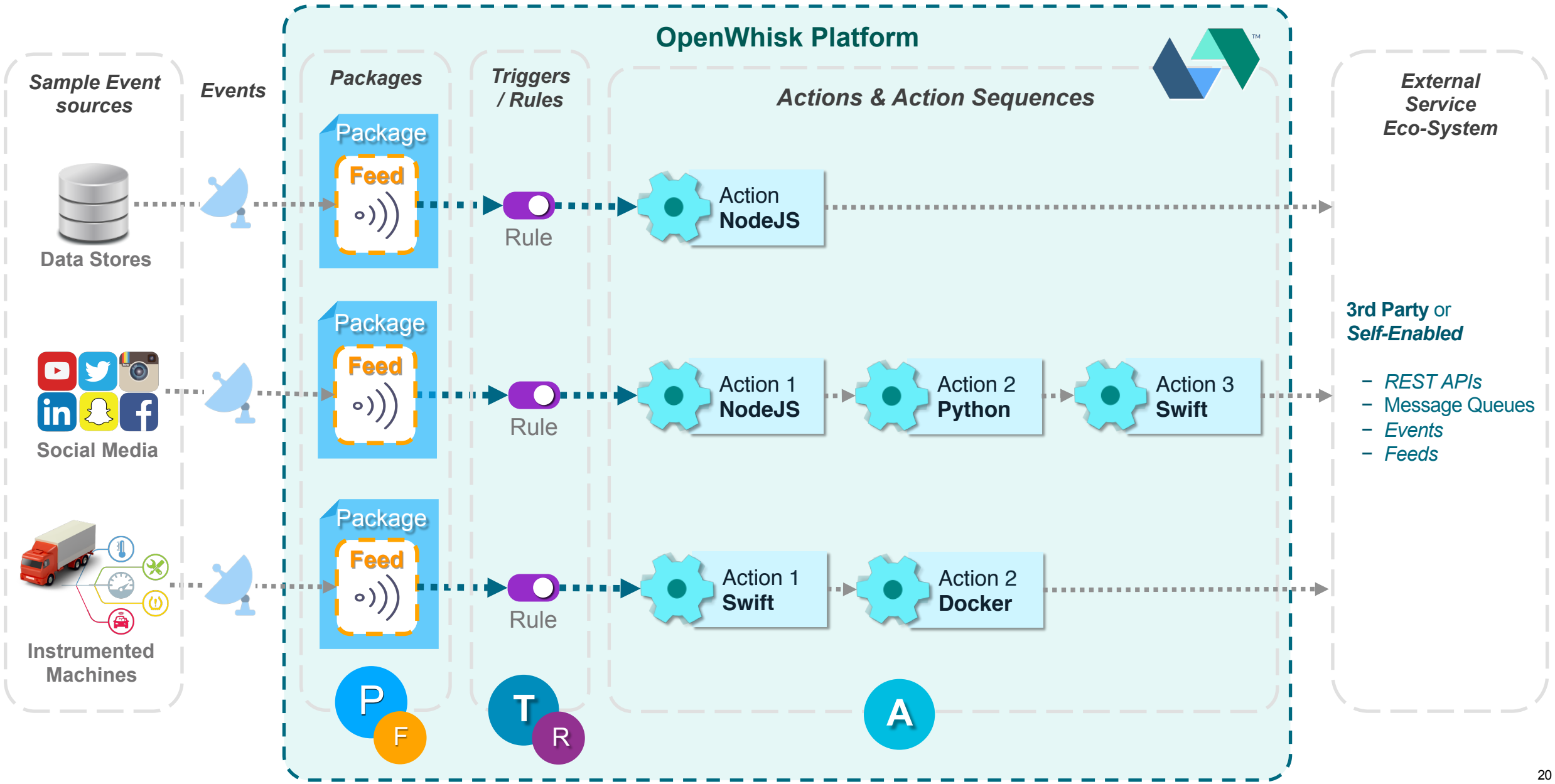
P Package:

A named, shared collection of (Namespaced) Actions, Triggers, Feeds, Rules, …

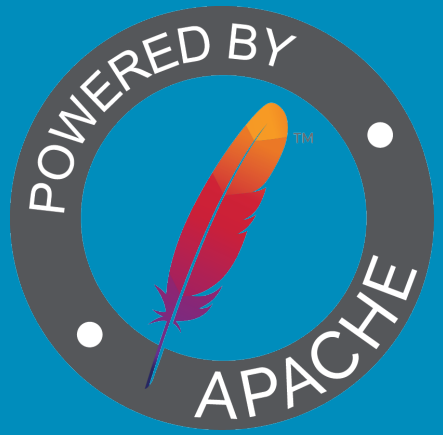


***Note**: Designed as a first-class entity in the OpenWhisk Model, being used by new Whisk Deploy tool ("**wskdeploy**") for 1-click Deployments*

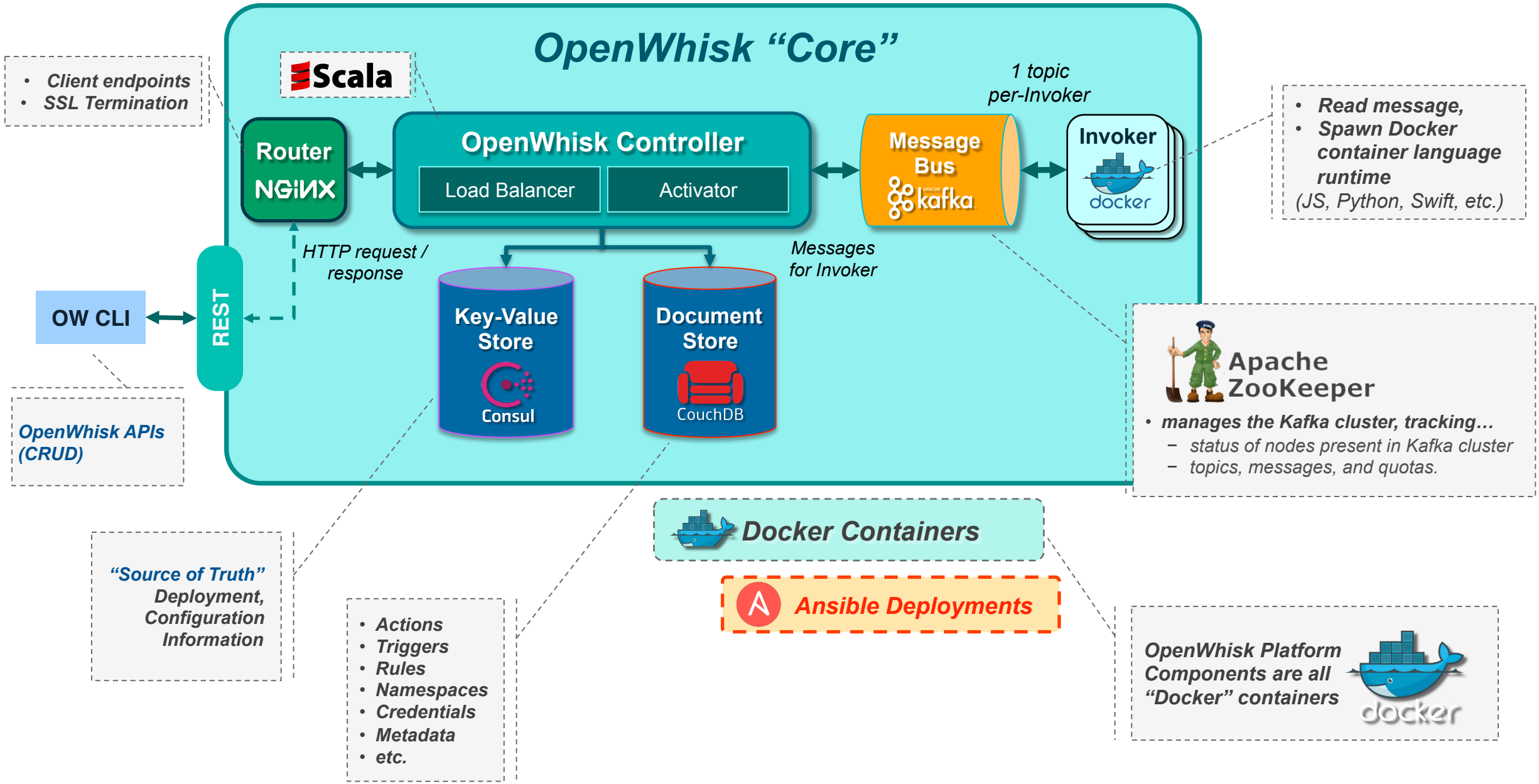
Apache OpenWhisk – Event-Trigger-Rule-Action Processing



Apache OpenWhisk
Project Architecture & Overview



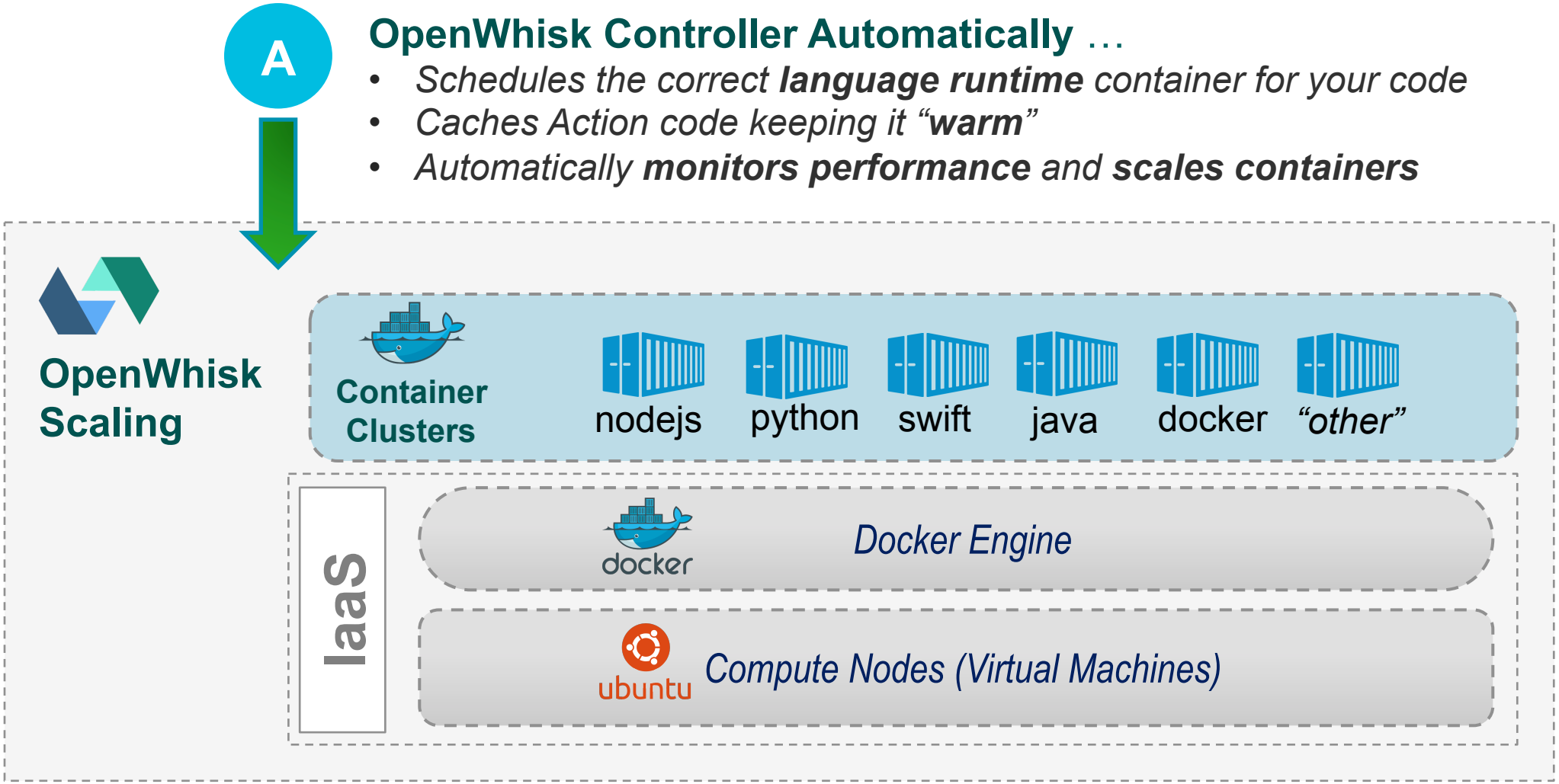
OpenWhisk Platform Architecture



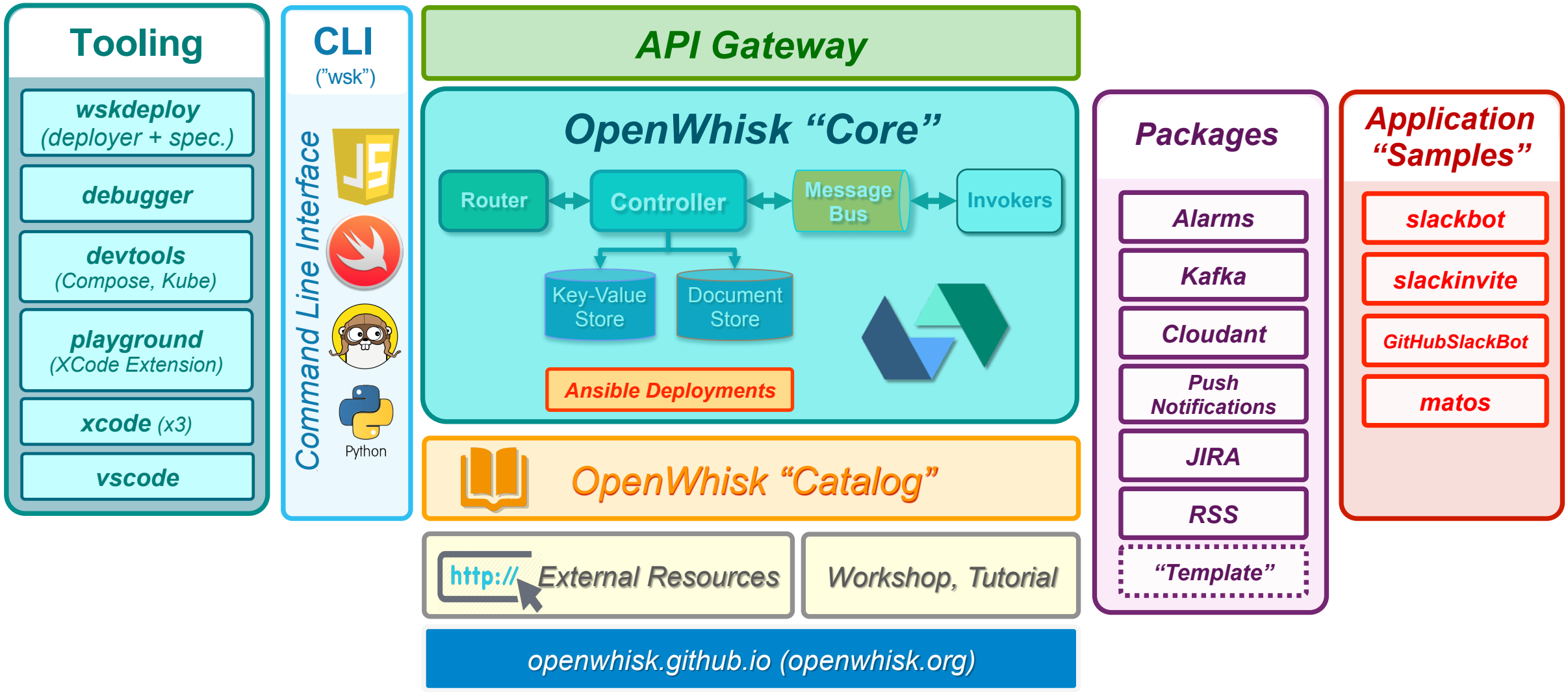
Invokers – Run Language-Specific Docker Containers for Actions

OpenWhisk supports many languages (runtimes) for Actions

- *JavaScript, Swift, Java, Python and more in the future…*



Apache OpenWhisk "Eco-System" Overview



APACHE OpenWhisk™

“Core” OpenWhisk Platform Repository

incubator-openwhisk

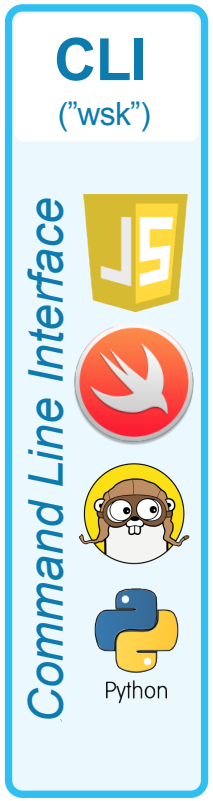
- ***Contains the OpenWhisk Serverless Platform code***
  - ***Scala*** *is the Primary Language of the “Controller”*
  - ***Manages*** *Whisk System Entities / State;* ***Load balances*** *Invoker requests; Provides* ***Logging***
- **Ansible run books for various deployment environments**
  - *i.e., ”Local” (single Vagrant VM), “Distributed”, “Mac”*

System & Utility Catalog Repository

incubator-openwhisk-catalog

- Catalog of built-in “system”, “utility” and sample Actions and Feeds
  - ***Utilities****: Combinators (forwarder, retry, etc.), Git, Slack, Weather, Web Socket, IBM Watson*
  - ***Samples****: cat, countdown, curl, greeting, hello (variants), httpGet, sort, split, trigger, wordcount, …*

Command Line Interface (CLI) Repositories



incubator-openwhisk-cli

- *Implements "wsk" CLI using the "**Cobra**" framework*
  - *Integrated Syntax Checking (POSIX flags) / Auto-completion / Validation / Help*
- *Supports "Go" API Client as a "plug-in"*
- *Supports client SDKs… (component = docker | swift 3 | iOS 9 &10 | OSX)*

incubator-openwhisk-client-go

- *Implements HTTP-REST APIs for OpenWhisk*
  - *Actions. Triggers, Rules, Activations, Packages, Namespaces,*
  - *Api Gateway, SDKs, more…*

Http API Clients for other languages (Python, Swift, JavaScript)

- ***incubator-openwhisk-client-python***
- ***incubator-openwhisk-client-swift***
- ***incubator-openwhisk-client-js***

Package Repositories

APACHE OpenWhisk™

Packages

openwhisk-package-alarms

- Alarm (Feed) that fires a (trigger) event at a specified frequency (UNIX “crontab”).

openwhisk-package-kafka

- *Package for c*ommunicating (**Producer-Consumer**) with **Kafka** *(or IBM Message Hub)* instances…

openwhisk-package-push-notifications

- **Push Service**: *Google Cloud Messaging (GCM), Apple Push Notification* (*APNs*), Firefox/Chrome/ Safari

openwhisk-package-jira

- Integration with **JIRA** events. Simple issue creation supported

openwhisk-package-rss

- Subscribe to **RSS/ATOM** feeds and receive events when a new feed item is available.

openwhisk-package-cloudant

- Enables interaction (read / write / monitor) with a **Cloudant** database

API Gateway Repository

API Gateway

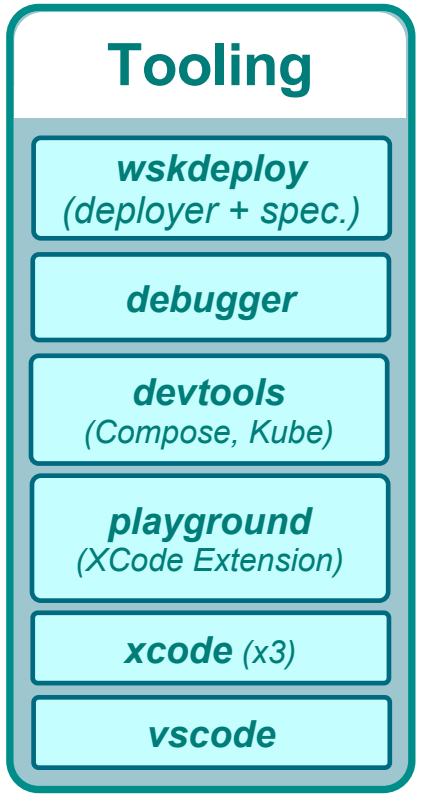
incubator-openwhisk-apigateway

- Main configuration files and a Dockerfile to build all modules into a single container.
  - leverages technologies like Nginx, OpenAPI (Swagger), LUA Modules (plug-ins)
- Combined with OpenWhisk it provides an easy way to build back-ends without the need to manage servers by mapping APIs to actions
- *Now supported via Whisk CLI using the "api" command (no longer "api-experimental")*

Recent Activities

- *Management interface to configure actions as APIs*
- *OAuth2 support*
- *CORS support*
- *Profiling tools used to measure performance*

“Tooling” Repositories



incubator-openwhisk-wskdeploy

- OpenWhisk deployment tool “**wskdeploy**”, 1-click deployments
- Deploys OpenWhiske ”Packages” (“manifest.yml” files)

incubator-openwhisk-devtools

- Experimental deployments for testing OpenWhisk functions locally.
- using **Docker Compose, NodeJS**

incubator-openwhisk-deploy-kube

- **Kube deployment** – on Ubuntu (localhost) - as of May 16th
  - *All **Docker** images now **shared** between both Compose and Kubernetes*

incubator-openwhisk-debugger

- Debug Actions locally using the ”**wskdb**” command using NodeJS (NPM package)

incubator-openwhisk -xcode / -playground

- collection of OpenWhisk tools for OS X implemented in Swift 3

incubator-openwhisk-vscode

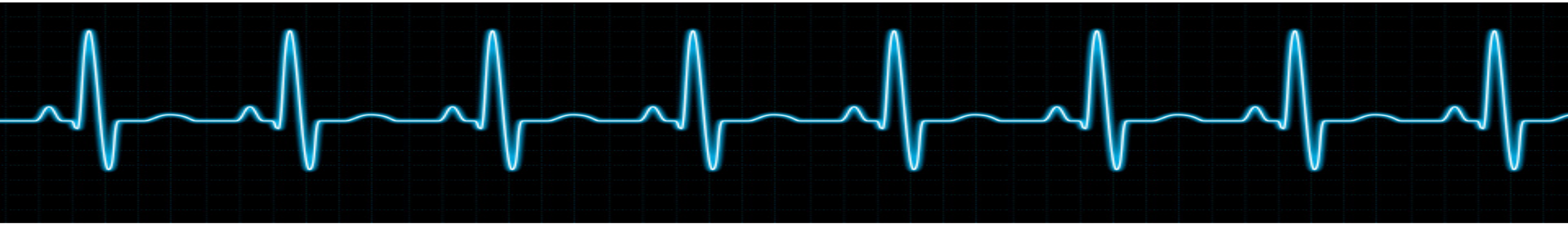
- Prototype extension for Visual Studio Code

APACHE OpenWhisk™

Apache OpenWhisk Community

OpenWhisk Project Vital Statistics

- **GitHub** *("Core" Repo.)*
  - ***Stars***
    - **1493**: *https://github.com/apache/incubator-openwhisk/stargazers*
    - *and growing ~10-20 per week on average*
  - ***Stats***
    - ***311 forks*** *https://github.com/apache/incubator-openwhisk/network*
    - ***Contribution Graphs***: *https://github.com/apache/incubator-openwhisk/graphs/contributors*
    - ***85+ Contributors*** *(all repos.)*
  - *plans to formalize releases (and binary downloads) from Apache so we can start tracking users…*



OpenWhisk – Become a Contributor

Find all OpenWhisk Repositories: https://github.com/apache?q=openwhisk

My personal

- ✓ **More Integrations !** *(Queues, Devices, Data Stores)*
- ✓ **More Language Runtimes**
- ✓ **More Packages !** (*Twilio, Facebook, Twitch, etc.)*
- ✓ **More Samples** *(Workflows of 3+ Actions)*
- ✓ *Verify / Fix / Create / Improve* ***Documentation***
- ✓ **Whisk Deploy** *(wskdeploy)* **Package Manifests**
  - *for all Curated Packages / Samples*
- ✓ **Web User Interface** *(Compose, Deploy, Debug via UI)*
- ✓ **More Web Actions** *(Mime Types)*
- ✓ **Whisk Package Catalog** *(NPM-like Distributed Registry)*

OpenWhisk.org – Connect with us!

- ***Latest Events & Meetups!***
- ***Links to …***
  - ***YouTube***: *Channel "OpenWhisk"*
  - ***SlideShare***: *https://www.slideshare.net/OpenWhisk*
  - ***Twitter***: #openwhisk
  - ***Slack***:
    - Invite yourself using a Whisk Action! http://slack.openwhisk.org/
  - ***Medium*** *(blogs)*: https://medium.com/openwhisk
  - ***StackOverflow***: tag "openwhisk"

Thank you!

Questions?