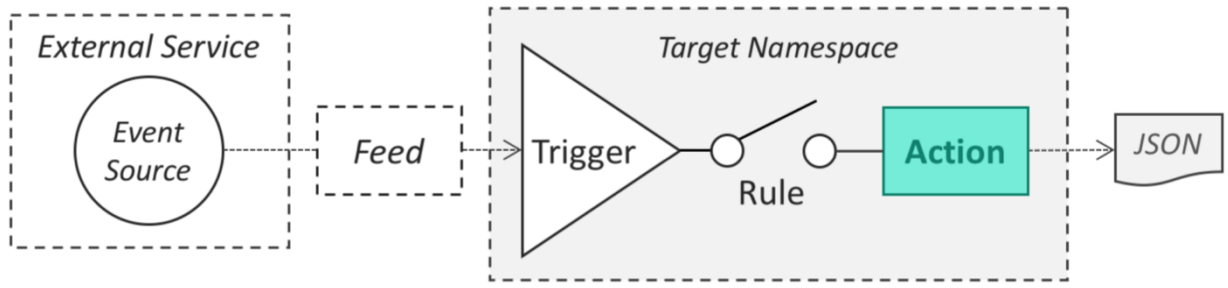
Community Analysis Report
High-performance Scientific Computing
Ali Tariq (ali.tariq@colorado.edu)

Domain background:

Serverless is a cloud computing execution model where the cloud provider dynamically manages the allocation and provisioning of servers. A serverless application runs in stateless compute containers that are event-triggered, ephemeral (may last for one invocation), and fully managed by the cloud provider. Service pricing is based on the number of executions(or time taken for execution) rather than pre-purchased compute capacity. The Serverless Framework was launched back in 2015, under a different name "Jaws" and ever since, the framework has experienced tremendous growth with over fourteen thousand stars on Github.

Project Introduction:

Apache OpenWhisk is an open-source project that spans two major areas; distributed systems and serverless. Apache OpenWhisk is driven by IBM and Adobe, is a robust Functions-as-a-Service (FaaS) platform that can be deployed in the cloud or within the data center. OpenWhisk follows event driven programming model, events drive the Serverless execution of functional code called Actions. Events can come from any Event Source or Feed service including such as Datastores, Message Queues, Mobile and Web Applications, Sensors, Chatbots, etc. The diagram below shows the control workflow logic inside OpenWhisk.



Currently supported runtimes in Apache OpenWhisk include .Net, Go, Java, Javascript, Php, Python, Ruby, Swift. While Ballerina & Rust runtimes are in the process of development.

For more information: https://openwhisk.apache.org/documentation.html

Apache OpenWhisk is a big project that targets users from multiple environments and domains, hence provides multiple deployment options for its users. Major deployments include:

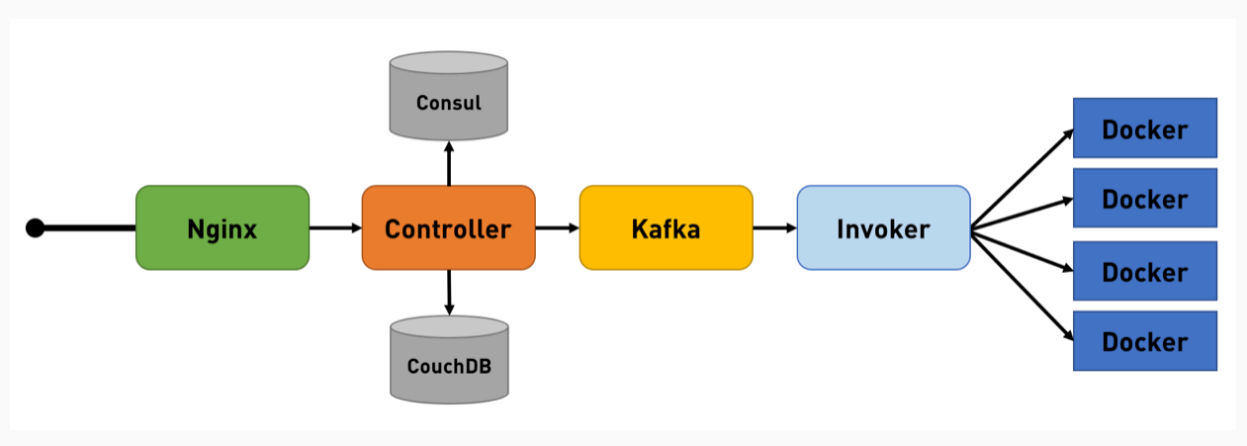
- Docker-compose
- Kubernetes-deployment

- Vagrant-deployment
- Mac/Ubuntu servers

Project Architecture:

Apache OpenWhisk is designed to act as an asynchronous and loosely coupled execution environment that can run functions against external triggers. Developers write standalone functions that are uploaded as Actions, which is completely autonomous and independent of the event sources. The project itself is built on top of other open-source projects, Nginx to Kafka to Docker, multiple technologies are powering this serverless platform.

The following diagram depicts the high-level architecture of OpenWhisk.



Project Community:

Project details:

Following on from the important version 1.0 release in 2016, the team behind the framework was able to raise VC funding to ensure the project continues to grow. The OpenWhisk project itself was initially born out of IBM, which donated its beta-level code into the Apache Incubator project in late 2016. IBM was using that codebase to support functions running on its IBM Cloud. IBM and Adobe are main driving entities, encouraging more and more community involvement in this project. Though many open-source serverless platforms have mushroomed in the recent past, OpenWhisk stands out for its robust architecture and design. Apache Openwhisk is a huge project comprised of multiple repositories. In 2019, Apache OpenWhisk has officially graduated to become an Apache Software Foundation (ASF) Top-Level-Project.

(for more information: https://github.com/apache?q=openwhisk).

For the purpose of this course, I chose the core repository providing the serverless functionality. This repository has only seen an incubating release from 2018, and first stable release has been planned for this fall. The repository has 2.6k commits (getting 5-7 commits every week now!) so far, with a total of 224 contributors.

Community members:

Developers: Developers are users that develop functions/actions, creating packages, feeds, and feed provider services using project APIs, clients and tooling.

Operators: Operators are users who deploy, configure, and host the Apache OpenWhisk Function-as-a-Service platform for development, testing and/or production.

Contributors: Developers designing and coding backend architectural components and tooling that comprise the OpenWhisk platform and its project ecosystem.

Governance:

Contributor: Everyone can contribute to OpenWhisk project as a contributor. Contributors have read access and submit GitHub Pull Requests (or PRs) of their contributions for committers to merge following peer review.

Committer: Contributors who build up a history of successful contributions over time are invited to become committers by the existing Podling Project Management Committee (PMC) membership. Committers have direct write (merge) access to the project code repositories

Project Management Committee (PMC) Member: The existing Apache OpenWhisk PMC members recognize individuals whose project contributions, across all community mediums, and outreach elevate them to a level in which their membership to the PMC is requested.

Getting Started:

To become part of the community:

1. Contributor needs to sign an Apache Contributor License Agreement (CLA)
2. Subscribe to the "dev" mailing list and introduce himself
3. Sign up for a Confluence Wiki account to collaborate with fellow project contributors.
4. Join our developer Slack channel for increased communication
5. Enable 2-Factor Authentication within their GitHub user account