

Apache OpenWhisk Project Wiki

blocked URL

OpenWhisk is an Apache project which provides a complete Serverless, or Function-as-a-Service, platform along with an ecosystem of curated functional packages, client SDKs and integration tooling.

Welcome!

This Wiki is intended for our community to share their thoughts about and collaborate around any aspect of the project; whether it's about the platform architecture, an existing or planned feature or about its use cases and integrations, this is the place to do it.

Quick Links

- **Website:** <http://openwhisk.apache.org/> - The project website with latest developer, integrator or end-user information, including "getting started".
- **GitHub:** <https://github.com/apache?q=openwhisk> - The project code hosted on GitHub which includes extensive markdown documentation.
 - Project Status (build): [GitHub Repository Status](#)
- **Slack:** openwhisk-team.slack.com
- **PMC SVN:** <https://svn.apache.org/repos/private/pmc/openwhisk>

Background

Serverless computing is the evolutionary next stage in Cloud computing carrying further the abstraction offered to software developers using Container-based operating system virtualization. The Serverless paradigm enables programmers to just "write" functional code and not worry about having to configure any aspect of a server needed for execution. Such Serverless functions are single purpose and stateless that respond to event-driven data sources and can be scaled on-demand.

The Apache OpenWhisk project offers a truly open, highly scalable, performant distributed Serverless platform leveraging other open technologies along with a robust programming model, catalog of service and event provider integrations and developer tooling. Specifically, every architectural component service of the OpenWhisk platform (e.g., Controller, Invokers, Messaging, Router, Catalog, API Gateway, etc.) all is designed to be run and scaled as a Docker container. In addition, OpenWhisk uniquely leverages aspects of Docker engine to manage, load balance and scale supported OpenWhisk runtime environments including [NodeJS](#), [Go](#), [Java](#), [Scala](#), [PHP](#), [Python](#), [Ruby](#), and [Swift](#) as well as recent additions for [Ballerina](#), [.NET](#) and [Rust](#) that run Serverless functional code within Invoker compute instances, using Docker containers.

OpenWhisk's containerized design tenants not only allows it to be hosted in various IaaS, PaaS Clouds platforms that support Docker containers, but also achieves the high expectation of the Serverless computing experience by masking all aspects of traditional resource specification and configuration from the end user simplifying and accelerating Cloud application development. In order to enable HTTP requests as a source of events, and thus the creation of Serverless microservices that expose REST APIs, OpenWhisk includes an API Gateway that performs tasks like security, request routing, throttling, and logging.

How to contribute?

The Apache OpenWhisk project is actively seeking new contributors to work on all aspects of the project. We need and appreciate all contributions, including documentation help, source code development and feedback.

Code

Some basic steps for creating issues (i.e., for bugs or features):

- When you have found an issue or you want to propose a new feature, post a message to OpenWhisk's "dev" email list to discuss it <or> ask about what you are seeing in our Slack channel.
- Search existing GitHub issues (for the respective project repository) to see whether someone had already encountered the same issue. Contribute any additional info. you have on the problem or possible solution.
- If this issue is never encountered before, create a new GitHub issue.
- Include or develop a test case to demonstrate the problem and attach it to issue.
- If you have also developed a fix for the raised issue, submit a Pull Request (PR) in referencing the issue you created.

Documentation

Each sub-project of OpenWhisk has its own GitHub repository where each should have a **README** (.md, markdown) file that explains the subproject and provides information on how to install, configure and run that repository's code.

The OpenWhisk website (<https://openwhisk.apache.org/>) is generated automatically from the OpenWhisk project repository at: <https://github.com/apache/openwhisk-website>

How do I become a contributor or a committer?

Everyone can contribute to OpenWhisk project as a contributor; contributors who build up a history of successful contributions over time are invited to become committers. The difference is that committers have direct write (merge) access to the project code repositories, while contributors have read access and submit GitHub Pull Requests (or PRs) of their contributions for committers to merge following peer review. See <http://www.apache.org/dev/contributors.html> for a good overview of working as a contributor.

All Contributors need to sign an Apache Contributor License Agreement (CLA)

The OpenWhisk project asks that contributors to fill out and send into Apache, an [Individual Contributors License Agreement \(ICLA\)](#) (and perhaps also a [Corporate Contributors License Agreement \(CCLA\)](#)). It is our policy for Committers to not merge any contributions of significance (i.e., other than minor documentation, typos, or small single-line bug fixes) without verifying the submitter has a CLA on file with the Apache Software Foundation (ASF).

All Contributors should

- Subscribe to the "dev" mailing list (or its digest) and introduce themselves!
 - dev-subscribe@openwhisk.apache.org
- Sign up for a Confluence [Wiki](#) account to collaborate with fellow project contributors.
 - Ask one of the project Confluence admins via Slack | email lists (dev or private) to add to the OpenWhisk project space
 - Please provide your chosen Confluence account username
- Join our developer Slack channel and say hello...
 - <http://openwhisk.apache.org/slack>
- Assure that their First, Last names and email address are filled out in their GitHub account profile
- Enable 2-Factor Authentication within their GitHub user account

Becoming a Committer

If you're interested in committing to the project, you need to establish a history, over time, of successful contributions to the project. After such recognition, you can then be invited to become a committer by the existing Podling Project Management Committee (PMC) membership.

- Once you have been elected a committer, you will go through a process to obtain an apache committer account: See [New Committers Guide](#), [Contributors](#), and [Committers](#) for more details on responsibilities.
- After you've received an email from root@apache.org with your committer account information, you will be added to the project's appropriate access lists (i.e., LDAP) and provided with further instructions.
- After becoming a Committer, you will need to link your GitHub account to your Apache account by following instructions here: [Accessing Apache GitHub as a Committer](#)
 - Primarily this requires the use of GitBox to link your newly created Apache account to your GitHub account: <https://gitbox.apache.org/setup/>
- Committers may want to read this FAQ: <http://apache.org/dev/committers.htm>

Becoming a Podling 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. If you are invited to become a PMC member and accept the invitation:

- You will be added to the PMC membership roles which are listed here: <https://whimsy.apache.org/roster/committee/openwhisk>
- PLEASE subscribe to our private mailing list "private-subscribe@openwhisk.apache.org" with a non-blank subject and body (e.g., simply put "subscribe" in each).
- Learn about the PMC role (see <http://www.apache.org/dev/pmc.html>) and participate in the private email discussions to fulfill your role.