

# Serverless Supercomputing

## Project Logistics:

Mentors: Dan McPherson email: dmcphers@redhat.com; email: ;

Min-max team size: 2-4

Expected project hours per week (per team member): 6-8

Will the project be open source: yes

## Preferred Past Experience:

Serverless/FaaS Valuable

Containers/Docker/Kubernetes/OpenShift Nice to have

Git Valuable

JavaScript Valuable

## Project Overview:

### *Background:*

Serverless/FaaS computing really started taking off with the launch of AWS Lambda. More recently Red Hat and other companies have made a bet on an Open Source solution that will run across all cloud and on prem environments in Apache OpenWhisk. The reason for making such a bet is that there's way too much lock-in from a proprietary solution offered by any of the cloud providers. When you pick a solution like Lambda, you can no longer easily move your application to another provider and you have no control over your cost.

Red Hat is currently making an investment to offer OpenWhisk on top of OpenShift. The benefit being that OpenShift can run anywhere and will provide a platform that OpenWhisk can utilize and integrate with. Because FaaS doesn't really work without existing services to interact with, in the MOC, OpenWhisk will also be able to take advantage of the features from OpenStack such as the Swift object store.

### *Project Specifics:*

The goal of this project is to build an on demand "supercomputer" out of OpenWhisk on OpenShift on OpenStack in the MOC. Namely, given a task that is highly parallelizable (TBD which task), rather than spin up virtual machines or containers to solve the problem, we can instead use OpenWhisk/FaaS to have an on demand supercomputer. The goal would be to give a small portion of the work to each function, and spin up 1000s of workers to accomplish the job as quickly as possible. Such a model has huge benefits from a cost perspective in that you aren't paying for any overhead for running the job except for the actual execution time (no spin up or spin down cost). The problem has been solved with

virtual machines many times but you get billed by the hour, min, or second with virtual machines and the time to start and stop the machines can make it cost prohibitive to spin up too many instances at once. Using containers is better, but using FaaS is really the ultimate solution. What's needed though is a framework and examples to show how this is possible and in an environment (OpenWhisk) which is generically applicable.

**Some Technologies you will learn/use:**

OpenWhisk (FaaS)

Containers/Docker/Kubernetes/OpenShift

Software Engineering (Agile, Scrum, Git, etc.)

OpenStack/MOC

Parallel Computing