Load Testing and Scaling the Applications

Now that we have our apps built and deployed to our staging environment, it's time to add some load to them. For the purpose of our Lab this will allow us to see two things:

1. observe resource consumption metrics (CPU, memory, networking traffic, etc) for each app container.
2. how the auto-scaling feature works when using Openshift Serverless deployment mode.

Introducing Hyperfoil Load Driver

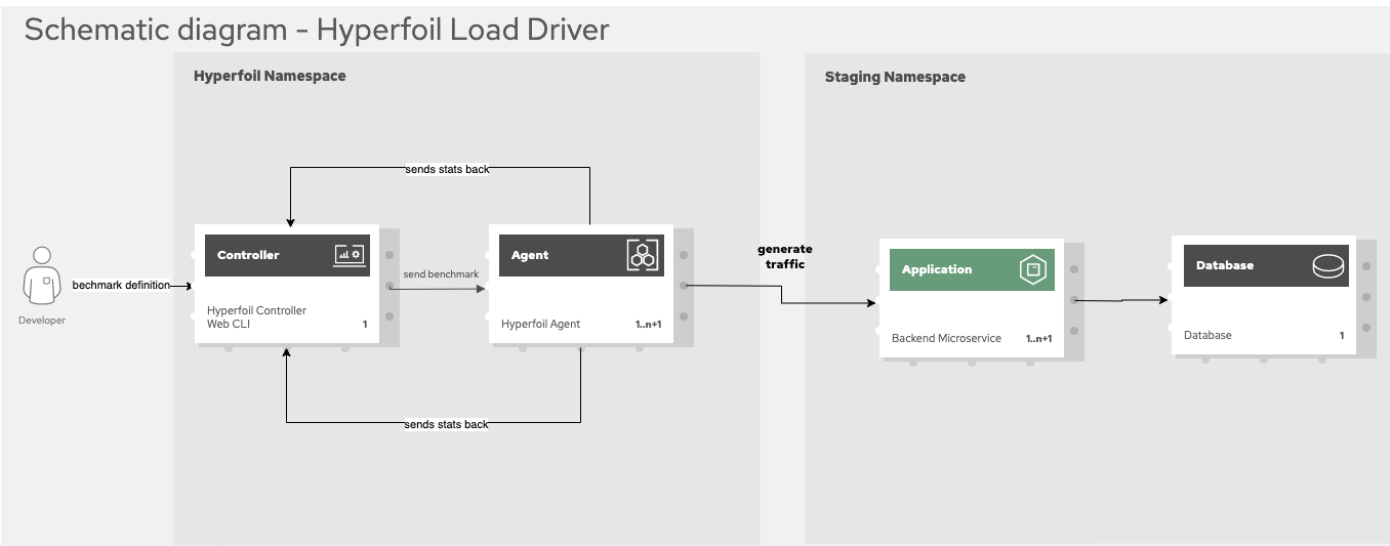
Nowadays, there are many opensource tools that can be used to generate application load towards your applications. Many of these tools were designed before Containers and Kubernetes making their deployment, configuration and scalability a challenge considering the distributed nature of Kubernetes.

For this Lab we will be using a tool called Hyperfoil.io (https://hyperfoil.io). Hyperfoil is described as a **Microservice-oriented distributed benchmark framework**. Because Hyperfoil is kubernetes-native, it's perfect for our use-case as it is easy to deploy and run on any Kubernetes environment.

> Hyperfoil is a powerful and flexible (kubernetes-native) **Load Driver** tool that can be applied to very robust use-cases. It was initially developed by the Red Hat Engineering team to test our own Software but is now available as an opensource project (https://github.com/Hyperfoil/Hyperfoil) for anyone to use and contribute.
>
> To see more details around the motivation and purpose of this tool, please see its Documentation here (https://hyperfoil.io/docs)

Hyperfoil Architecture

As mentioned early in its nature Hyperfoil is a distributed tool with leader-follower architecture (https://martinfowler.com/articles/patterns-of-distributed-systems/leader-follower.html). The **Controller** has the leader role; this is a Vert.x-based (https://vertx.io/) server with REST API. When a **benchmark** is started the Controller deploys **Agents** (according to the benchmark definition), pushes the benchmark definition to these agents and orchestrates benchmark * *. Agents execute the benchmark, periodically sending statistics to the Controller. This way the Controller can combine and evaluate statistics from all agents on the fly. When the benchmark is completed all Agents terminate.



In order to run a load testing using Hyperfoil we need to create a Benchmark (https://hyperfoil.io/userguide/benchmark.html). In Hyperfoil terms a Benchmark is a declarative YAML file where you define your load test Scenario. Lets take a look at one snippet of the benchmark definition (https://raw.githubusercontent.com/redhat-na-ssa/workshop_performance-monitoring-apps-template/main/scripts/hyperfoil/summit-load-apps.hf.yaml) we'll be using thought this section of our Lab.

YAML

```
# This is the name of the benchmark. It's recommended to keep this in sync with
# name of this file, adding extension `.hf.yaml`.
name: summit-lab-load-apps
# We must define at least one HTTP target, in this case it becomes a default
# for all HTTP requests.
http:
- host: http://quarkus-app:8080

# Distribute the connections among two agents
agents:
```