Circuit Breakers

Table of Contents

Adding circuit breakers via the Hystrix library helps you fail fast or provide a fallback if any dependent service either goes down or goes too slow.

Hystrix is a library rather than anything else, which means that it can just be easily added in **any** java program. Additionally there are frameworks that provide integration with Hystrix:

- Kubeflix
- Wildfly Swarm - Netflix *

Spring Cloud - Netflix

Standalone applications

Using Hystrix is as simple as implementing the HystrixCommand interface. Borrowed from hystrix examples here is a simple *Hello World* implementation:

```
public class CommandHelloWorld extends HystrixCommand<String> {

    private final String name;

    public CommandHelloWorld(String name) {
        super(HystrixCommandGroupKey.Factory.asKey("ExampleGroup"));
        this.name = name;
    }

    @Override
    protected String run() {
        return "Hello " + name + "!";
    }
```

The command can be now executed:

```
new CommandHelloWorld().execute();
```

This is enough to implement a circuit breaker.

Exposing Hystrix metrics

To expose metrics from the circuit breaker one needs to expose the HystrixMetricsStreamServlet, which can be found inside:

```
        <dependency>
            <groupId>com.netflix.hystrix</groupId>
            <artifactId>hystrix-metrics-event-stream</artifactId>
            <version>1.4.9</version>
        </dependency>
```

To register the servlet one needs to simply add the following inside the web.xml:

```
    <servlet>
        <display-name>metrics</display-name>
        <servlet-name>metrics</servlet-name>
        <servlet-
class>com.netflix.hystrix.contrib.metrics.eventstream.HystrixMetricsStreamServlet</ser
vlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>metrics</servlet-name>
        <url-pattern>/hystrix.stream</url-pattern>
    </servlet-mapping>
```

Using the Hystrix dashboard

An application that is exposing its hystrix metrics stream can take advantage of the visualization capabilities of the Hystrix Dashboard.

This is as simple as pointing the dashboard to the url of the hystrix metrics stream.

Using Turbine to aggregate multiple hystrix stream

To make the most out of the Hystrix dashboard you can aggregate multiple streams and have the dashboard visualize multiple circuit breakers at once.

The aggregation is performed by Turbine.

Kubeflix

Everything that has been mentioned so far is something that can easily used inside simple java application. But if those application are to be run inside Kubernetes, there will be some additional requirements:

- Circuit Breaker discovery
- Turbine Server Docker image
- Hystrix Dashboard Docker image

Circuit Breaker Discovery

In most cloud environments ip addresses are not known in advanced and kubernetes is no exception. This means that we can't have Turbine pre-configured with a fixed set of urls but instead we need a discovery mechanism. This mechanism is provided by Kubeflix and it pretty much allows to:

- Discovery all endpoints in the current that have been labeled as `hystrix.enabled`.
- Define multiple clusters that are composed by multiple endpoints accross multiple namespaces.

Turbine Server Docker image

Having a discovery implementation for turbine is not enough. We also need a turbine server app packaged as a docker container and of course the required Kubernetes configuration. Kubeflix provides both (image and configuration).

The image is a simple webapp, pre-configured with the Circuit Breaker discovery as described above. The great thing about this app is that the default configuration can be modified by:

- Environment Variables
- ConfigMap

and that makes it easier for the user to define his own clusters or tune turbine to his own needs.

Hystrix Dashboard Docker image

For the Hystrix dashboard we also need to package it as a Docker container and create the required Kubernetes configuration. Again Kubeflix provides both. On top of that web console is configured to reference Turbine Servers DNS name **out of the box**.

For more details, please visit the kubeflix project.

—

Wildfly Swarm Netflix

Taken directly from the Wildfly Swarm Website:

```
Swarm offers an innovative approach to packaging and running JavaEE applications by packaging them with just enough of the server runtime to "java -jar" your application.
```

One of the available modules is Wildfly Swarm Netflix which provides integration with the Netflix components.

A **Hello World** example of Hystrix with Wildfly Swarm.

It's important to note that this example is `Kubeflix-ready` which means that regardless of how it has been implemented it will be able to integrate with the rest fo the Kubeflix bits. This is also the the

case for the next framework in line....

—

Spring Cloud Netflix

This project provides integration between Spring Cloud and the Netflix components, including Hystrix as a **Circuit breaker** implementation. On top of that it provides integration with Ribbon and makes it easy to compose rest application that communicate with each other.

For Spring Cloud users it worths mentioning Spring Cloud Kubernetes that provides Kubernetes integration with Spring cloud and it allows you to use everything together (Spring Cloud, Kubernetes, Netflix components).