Spring Cloud Netflix: Circuit Breaker Metric Aggregation with Turbine

Spring Cloud Netflix: Circuit Breaker Metric Aggregation with Turbine

- Turbine
- Hystrix & Turbine AMQP
- Spring Cloud Services

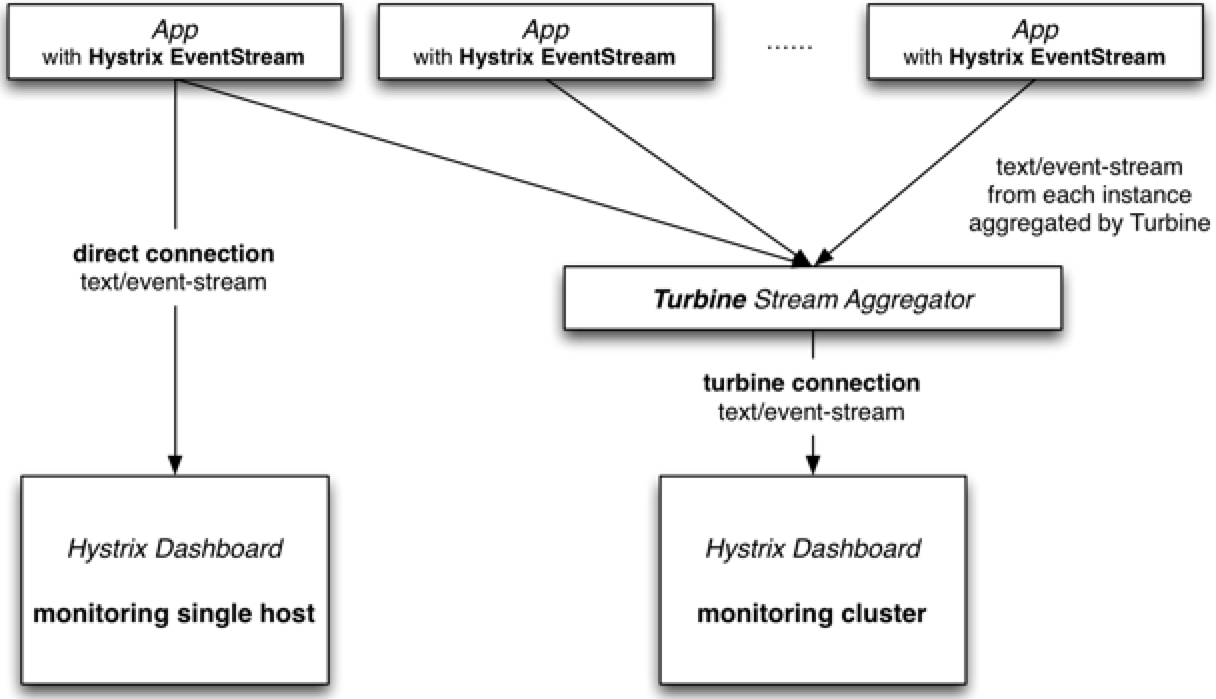
Limitations with `/hystrix.stream`

Looking at an individual instance's Hystrix data is not very useful in terms of the overall health of the system.

Turbine

- Aggregates individual **`/hystrix.stream`** endpoints into a combined **`/turbine.stream`** endpoint
- Individual instances are located via Eureka

Turbine Data Flow



Source: http://techblog.netflix.com/2012/12/hystrix-dashboard-and-turbine.html

Include Dependency

pom.xml

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-turbine</artifactId>
</dependency>
```

@EnableTurbine

```
@SpringBootApplication
@EnableTurbine
public class TurbineApplication {


    public static void main(String[] args) {
        SpringApplication.run(TurbineApplication.class, args);
    }

}
```

Turbine Configuration

bootstrap.yml:

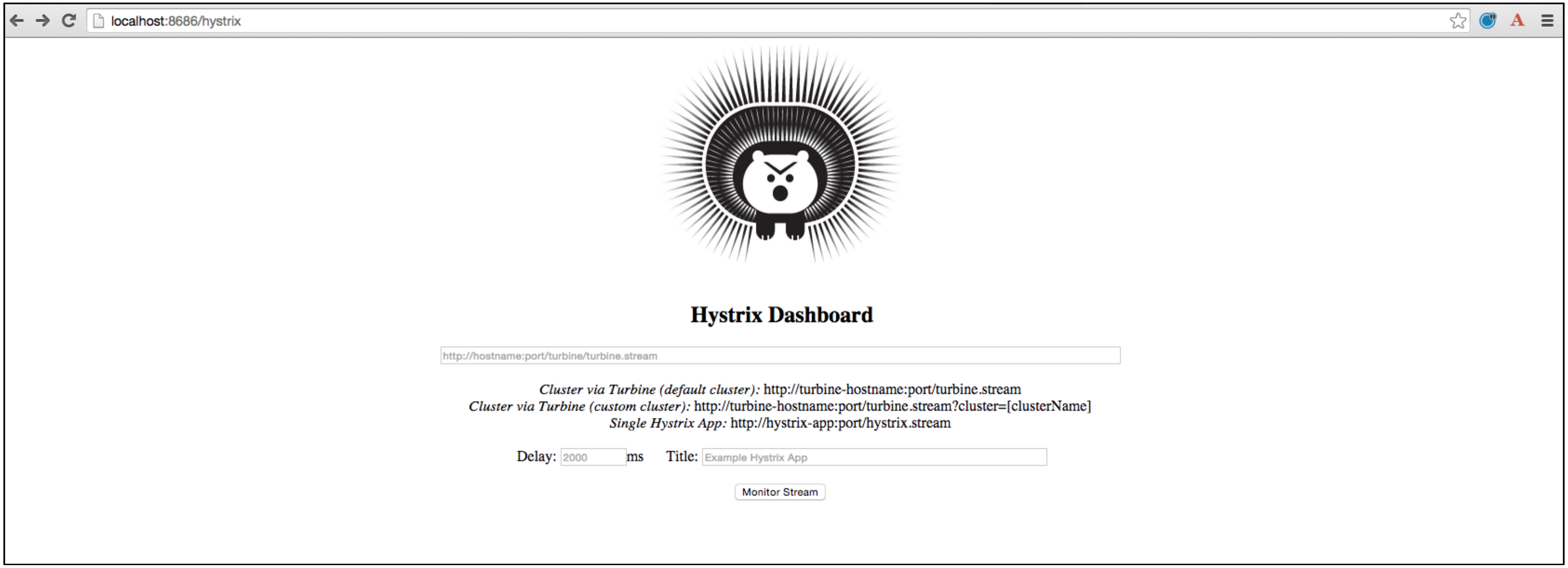
```
turbine:
  aggregator:
    clusterConfig: GREETING-HYSTRIX
  appConfig: greeting-hystrix
```

turbine.appConfig is a list of Eureka serviceIds that Turbine will use to lookup instances.

turbine.aggregator.clusterConfig is the Turbine cluster these services belong too.

Hystrix Dashboard With Turbine

Enter the the **`/turbine.stream`** endpoint.



Hystrix Dashboard

http://hostname:port/turbine/turbine.stream

Cluster via Turbine (default cluster): http://turbine-hostname:port/turbine.stream
Cluster via Turbine (custom cluster): http://turbine-hostname:port/turbine.stream?cluster=[clusterName]
Single Hystrix App: http://hystrix-app:port/hystrix.stream

Delay: 2000 ms Title: Example Hystrix App

Spring Cloud Netflix: Circuit Breaker Metric Aggregation with Turbine

- Turbine
- Hystrix & Turbine AMQP
- Spring Cloud Services

Hystrix & Turbine in Pivotal Cloud Foundry

- In Pivotal Cloud Foundry, the classic model of pulling metrics from event streams doesn’t when applications register with Eureka using their **`route`**.

- When applications register with their **`route`** every instance has the same url, making it impossible to distinguish between app instances.

- Therefore, metrics are published through a AMQP message broker on Pivotal Cloud Foundry. We'll use RabbitMQ.

Hystrix Applications

To publish metrics over AMQP, Hystrix based apps simply include the following dependency in the **`pom.xml`**

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-netflix-hystrix-amqp</artifactId>
</dependency>
```

Nothing else is needed when using a local RabbitMQ broker or running in Pivotal Cloud Foundry.

Turbine - Include Dependency

pom.xml

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-turbine-amqp</artifactId>
</dependency>
```

@EnableTurbineAmqp

For Turbine to consume metrics over AMQP, annotate your application with **`@EnableTurbineAmqp`**

```
@SpringBootApplication
@EnableTurbineAmqp
public class TurbineApplication {

    public static void main(String[] args) {
        new SpringApplicationBuilder(TurbineApplication.class).
          web(false).run(args);
    }
```

Turbine AMQP configuration

`turbine.appConfig` and **`turbine.aggregator.clusterConfig`** no longer need to be configured when using Turbine AMQP.

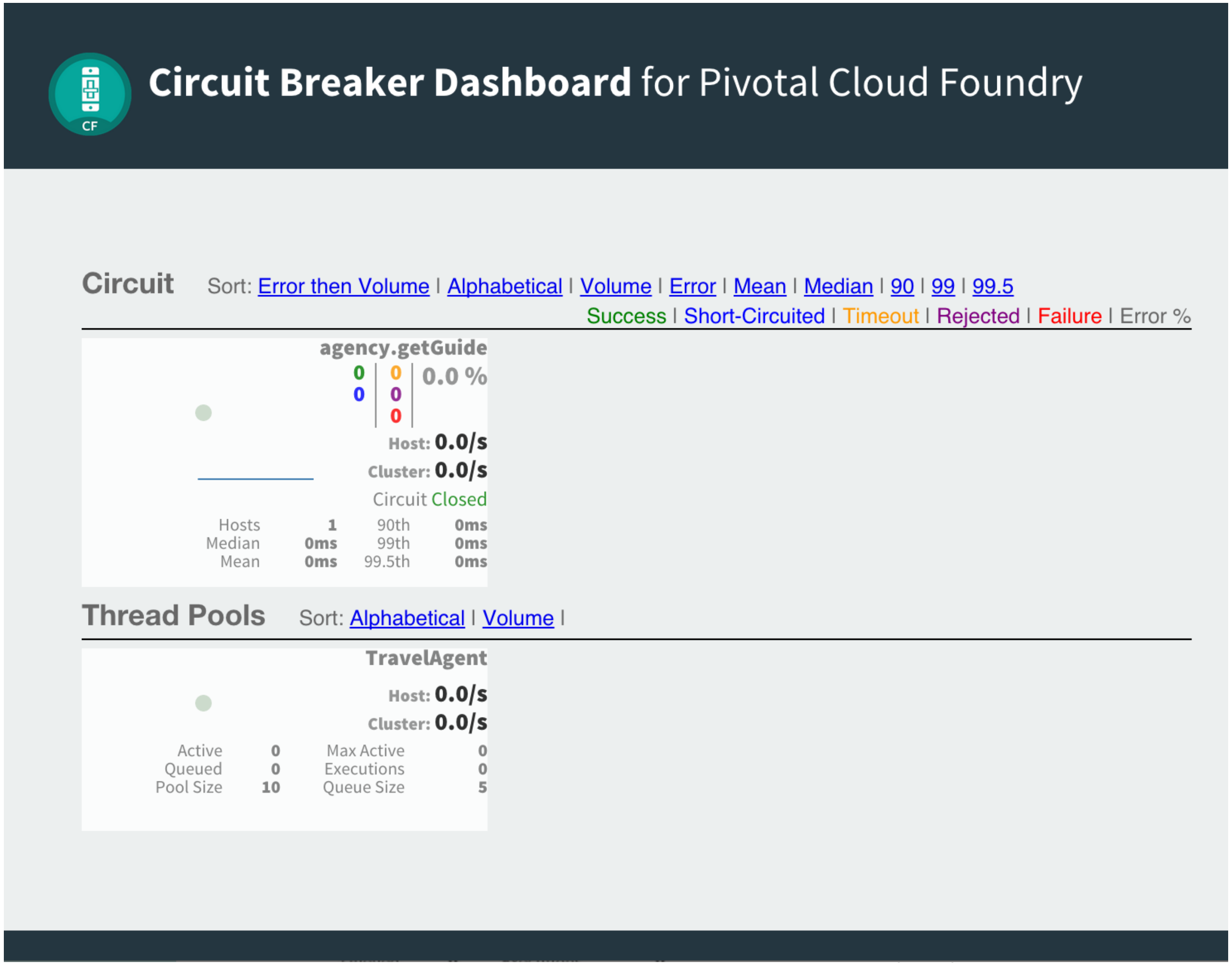
Spring Cloud Netflix: Circuit Breaker Metric Aggregation with Turbine

- Turbine
- Hystrix & Turbine AMQP
- Spring Cloud Services

Spring Cloud Services

- Brings Spring Cloud to Pivotal Cloud Foundry
- Includes: Config Server, Service Registry & Circuit Breaker Dashboard services

Spring Cloud Services: Circuit Breaker Dashboard



Circuit Breaker Dashboard for Pivotal Cloud Foundry

Circuit Sort: Error then Volume | Alphabetical | Volume | Error | Mean | Median | 90 | 99 | 99.5

Success | Short-Circuited | Timeout | Rejected | Failure | Error %

agency.getGuide

0 0 0.0 %
0 0
0

Host: 0.0/s
Cluster: 0.0/s
Circuit Closed

Hosts	1	90th	0ms
Median	0ms	99th	0ms
Mean	0ms	99.5th	0ms

Thread Pools Sort: Alphabetical | Volume |

TravelAgent

Host: 0.0/s
Cluster: 0.0/s

Active	0	Max Active	0
Queued	0	Executions	0
Pool Size	10	Queue Size	5

Spring Cloud Services: Circuit Breaker Dashboard

1) Include dependency:

```
<dependency>
  <groupId>io.pivotal.spring.cloud</groupId>
  <artifactId>spring-cloud-services-starter-circuit-breaker</artifactId>
</dependency>
```

2) Create Circuit Breaker Dashboard service instance

3) Bind the service to the app.