Page-3593-RabbitMQ

RabbitMQ

RabbitMQ is a message broker: it accepts and forwards messages. You can think about it as a post office: when you put the mail that you want posting in a post box, you can be sure that Mr. Postman will eventually deliver the mail to your recipient. In this analogy, RabbitMQ is a post box, a post office and a postman

```
ln -svf /usr/local/opt/rabbitmq/*.plist ~/Library/LaunchAgents
```

```
launchctl load ~/Library/LaunchAgents/homebrew.mxcl.rabbitmq.plist
```

4056
Hystrix itself does not care what kind of command gets wrapped by it and it does not support the idea of retries. Example behind the idea: If your command (that wraps a REST request) is parametrised it could be that some resource endpoints should be retried while others not. It won't be nice to have either two commands that do more or less the same nor a technical parameter to activate a retry. Additionally this will add some extra complexity to the project.
To get around this problem and stick with Hystrix you might want to take a look into SpringRetry if you are working on a Spring application.
Another possible solution is resilience4j which could be seen as a combination of Hystrix and SpringRetry in this context.

SARGE: https://github.com/jhalterman/sarge

Retry
https://myadventuresincoding.wordpress.com/2014/07/30/java-creating-a-simple-retry-command-with-function-passing-in-java-8/

RabbitMQConfiguration

http://www.baeldung.com/introduction-to-hystrix

Retry
https://gist.github.com/mlui/817505

```
ConnectionFactory factory = new ConnectionFactory();
// configure various connection settings

try {
  Connection conn = factory.newConnection();
} catch (java.net.ConnectException e) {
  Thread.sleep(5000);
  // apply retry logic
}


 factory.setNetworkRecoveryInterval(10000);
factory.setAutomaticRecoveryEnabled(false);
```

https://www.rabbitmq.com/heartbeats.html#heartbeats-timeout
Heartbeat Timeout Value

Day-3
With Spring boot
https://stackoverflow.com/questions/39281842/retry-configuration-for-rabbitlistener-programatically-specify-dead-letter-queu

As per my observation in local:

I created RabbitMQ broker in one system and it is up and running.
Now I am listening to that broker from other system.
I stop the RabbitMQ broker then the application getting connection exceptions as expected.
Once the broker is restarted then application is rebinding to the broker and getting the connection as expected.

Thursday

Dealing with pff instances:

Restarting one instance follow this:
https://stackoverflow.com/questions/48635211/how-to-kill-a-specific-instance-on-pivotal-cloud-foundry

Crash Events

If an app instance crashes, CF automatically restarts it by rescheduling the instance on another container three times. After three failed restarts, CF waits thirty seconds before attempting another restart. The wait time doubles each restart until the ninth restart, and remains at that duration until the 200th restart. After the 200th restart, CF stops trying to restart the app instance.
https://docs.cloudfoundry.org/devguide/deploy-apps/app-lifecycle.html

Friday

```
cf restart-app-instance APP_NAME INDEX
$ cf restage YOUR-APP
```

The IP address can be found using the Cloud Foundry CLI command given below.

```
CF_TRACE=true cf app <APP_NAME>
```

Monday
LOGS

Cloud Foundry aggregates logs for all instances of your applications as well as for requests made to your applications through internal components of Cloud Foundry. For example, when the Cloud Foundry Router forwards a request to an application, the Router records that event in the log stream for that app. Run the following command to access the log stream for an app in the terminal:

```
cf logs APP-NAME
```

Filtering logs

```
$ cf logs APP-NAME --recent | grep -v RTR
```

We can create an alert using **new-relic** or **circuit breaker**

Worked on restarting a single instance in PCF instead of restarting the total application.
Background: SpringBoot application is deployed in PCF with 12 instances using scaling. Previously If two instances are crashed or down we are restarting the application.

https://confluence.gapinc.com/display/LPP/Monitoring+logs+and+Requested+endpoints+to+specific+PCF+instances

https://docs.cloudfoundry.org/cf-cli/cf-help.html
https://confluence.gapinc.com/display/LPP/Monitoring+logs+and+Requested+endpoints+to+specific+PCF+instances

Now rabbitmq is migrating from PCF to AZURE. In Azure creating cluster for the RabbiMQ broker. Its may need some code changes in rabbitmq configuration classes. As per our discussion with Ravi, what are all changes needed just let me know.

As per the observations RabbitMQ broker is automatically reconnecting to MicroService when the connection lost and get the connection. RabbitMQ monitoring, recovery and clustering will be implemented as part of this https://jira.gapinc.com/browse/PAG-4106 story.