Pivotal

Cloud Foundry Meetup

June 2016

Caleb Washburn

About me….

- Advisory Solutions Architect @ Pivotal
- 18+ years in IT, mostly working within an enterprise
- Working with Spring based applications for 10+ year. Yes…when XML was the cool new thing
- Disclaimer: Work at Pivotal but my opinions are my own

Pivotal

What we plan to go over…

- Building a simple micro-service and deploying to Cloud Foundry
- Enhance that microservice to add integration with Spring Cloud Config
- Enhance that to register with Spring Service Registry
- Build a 2nd microservice that consumes the first service using Spring Service Registry
- Enhance the 2nd service to add Circuit Breaker capability

Your first microservice

Microservice getting started

1) Get started by going to start.spring.io and download a starter
2) Build it (we will use maven - mvn install)
3) Add a simple controller
4) Basic application.properties setup
5) Add a cf manifest
6) Deploy it to cloud foundry via cf push

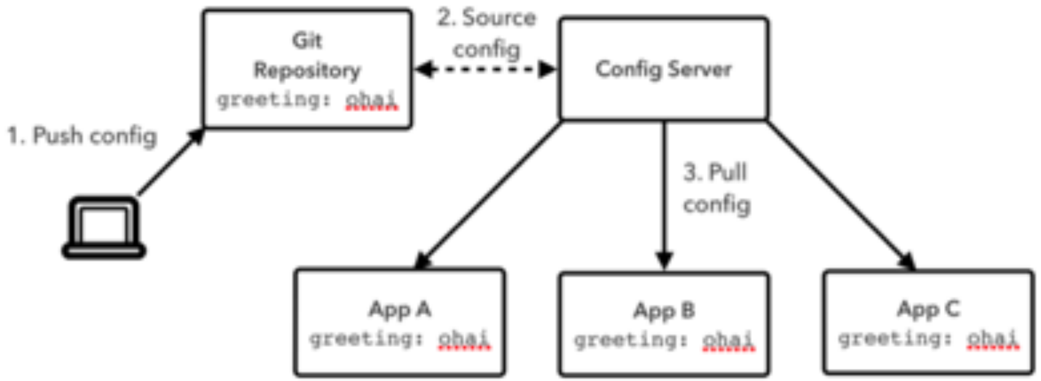
Microservice getting started

DEMO

Spring Cloud Config

Spring Cloud Config Server

Leverages SVN or GIT for application configuration. Properties that you would normally put in properties files but now can externalize with a common approach.



Spring Cloud Config Server

- Enabled through creating a service instance

cf create-service -c '{ "git": { "uri": “<Your GIT REPO URI>", "label": "master" } }' p-config-server standard config-server

- And binding to your application and restage

cf bind-service my-app config-server
cf restage

- Remember to annotate any Spring beans that you want to dynamically refresh when configuration changes with @RefreshScope so a HTTP Post to /refresh will force a reload

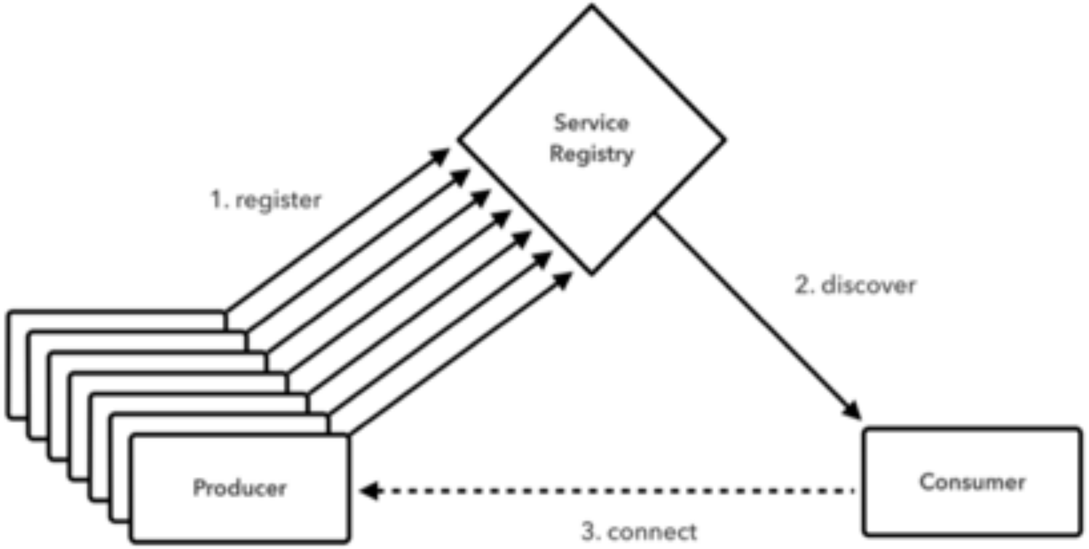
Spring Cloud Config Server

DEMO

Spring Service Registry

Spring Service Registry

Allows for dynamic registry and discovery of services within a Cloud Foundry space. This enables microservices deployed into the same space to discover each other. There are 2 approaches to discovery either direct or router. Router mode registers the Cloud Foundry Go Router url with the service registry where direct registers the container IP/Port and bypasses the network routing layer. Based on Eureka Netflix's Service discovery server/client



Spring Service Registry

- Enabled through the @EnableDiscoveryClient on your spring boot application
- For client side load-balancing via Ribbon in your microservice client. Annotate your injected RestTemplate with @LoadBalanced

```
@Autowired
@LoadBalanced
 RestTemplate restTemplate;
```

- Create a service

```
cf create-service p-service-registry standard service-registry
```

- Bind it to your application

```
cf bind-service my-app service-registry
cf restage
```

- Dashboard is created to show state of service

cf service service-registry to get dashboard url for this service instance

Spring Service Registry Service

DEMO

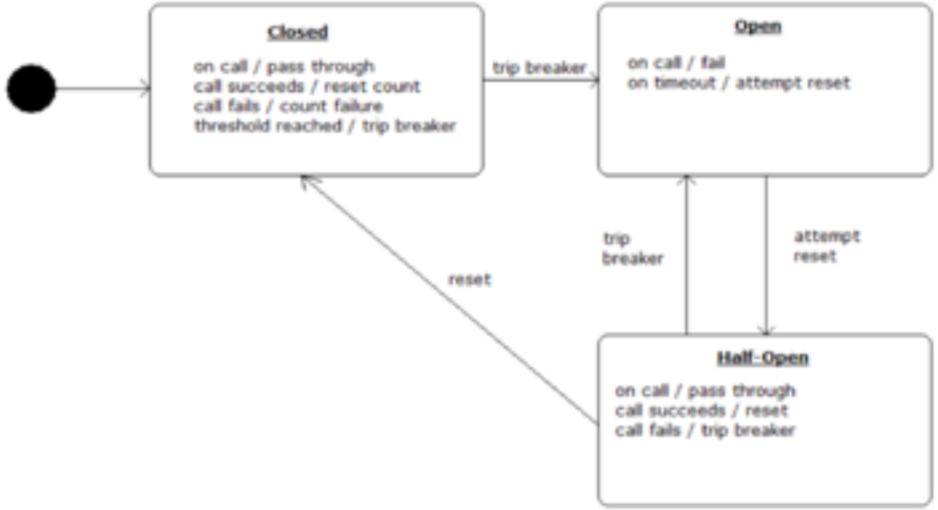
Spring Service Registry Client

DEMO

Spring Circuit Breaker Dashboard

Spring Circuit Breaker Dashboard

Allows for a microservice client to configure a fallback if the downstream microservice is unavailable. This will kick in automatically and opens the circuit after a threshold of a failing calls has been exceeded. After the service is recovered it will automatically be added back into the call chain and the circuit will be closed.   This is based on Hystrix Netflix latency and fault-tolerance library.



Pivotal

Spring Circuit Breaker Dashboard

- Enabled with @EnableCircuitBreaker on your spring boot application
- Enabled through usage of @HystrixCommand annotation on method that invokes the service

- Create a service

cf create-service p-circuit-breaker-dashboard standard circuit-breaker

- Bind it to your application

cf bind-service my-app circuit-breaker
cf restage

- Dashboard is created to show state of service

cf service circuit-breaker to get dashboard url for this service instance

Spring Circuit Breaker Dashboard

DEMO

Pivotal

Transforming How The World Builds Software

Thank You

Questions?

Pivotal

References

Spring Cloud for Cloud Foundry Documentation - http://docs.pivotal.io/spring-cloud-services/index.html

Source Code for Demo - https://github.com/calebwashburn/spring-cloud-demo