Spring Cloud

Gene Kuo

Topics

- Service discovery
- Edge server
- Centralized configuration
- Circuit breaker
- Distributed tracing
- Integration with Spring Security (OAuth 2.0)

Service discovery

- Netflix Eureka and Spring Cloud load balancer
- Registration of microservices with Netflix Eureka
- Scaling up the number of instance of a microservice
- Load-balancing traffic over available instances based on round-robin scheduling
- Eureka web UI

Edge server

- Spring Cloud Gateway and Spring Security OAuth
- Security: hiding private services and protecting public services in a microservice landscape
- URL path-based routing
- Protecting endpoints via the use of OAuth 2.0 and OpenID Connect (OIDC)
- Spring Cloud Gateway based on non-blocking APIs from Spring 5, Project Reactor and Spring Boot 2

Edge server

Centralized configuration

- Spring Cloud Config
- Storing configuration files:
  - Git repo
  - Local file system
  - HashiCorp Valut
  - JDBC Database
- Hierarchical structure configurations

Centralized configuration

- Detection of changes in the configuration and push notifications (Spring Cloud Bus) to the microservices
- Spring Cloud Bus is based on Spring Cloud Stream and supports RabbitMQ and Kafka
- Encryption of sensitive information

Centralized configuration

Circuit breaker

- `Resilience4j` is a open source-based fault tolerance library
- Circuit breaker: prevents a chain of failure reactions if a service stops to respond
- Rate limiter: limits the number of requests during a specified time period
- Bulkhead: limits the number of concurrent requests to a service

Circuit breaker

- Retries: handles random errors happened from time to time
- Timeouts: avoids waiting too long from slow or not responding services

Example

```
curl $HOST:$PORT/actuator/health -s | jq
.details.myServiceCircuitBreaker
```

Distributed tracing

- Spring Cloud Sleuth and Zipkin
- Tracks and visualizes requests and messages flow between microservices
- Spring Cloud Sleuth Marks requests and messages in a flow with correlation ID (TraceId)
- Spring Cloud Sleuth sends tracing data to Zipkin for storage and visualization
- trace tree and spans
- Synchronous over HTTP and asynchronous using RabbitMQ or Kafka

Distributed tracing