SAND2018-4932C

Developing Cloud-Native Services with the Spring Framework

PRESENTED BY

Todd Fielder

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Agenda

- Overview of Spring Boot and Spring Cloud
- Discussion of Distributed System Patterns and the components involved
- Demo of a Spring Cloud Microservices project
- Q & A

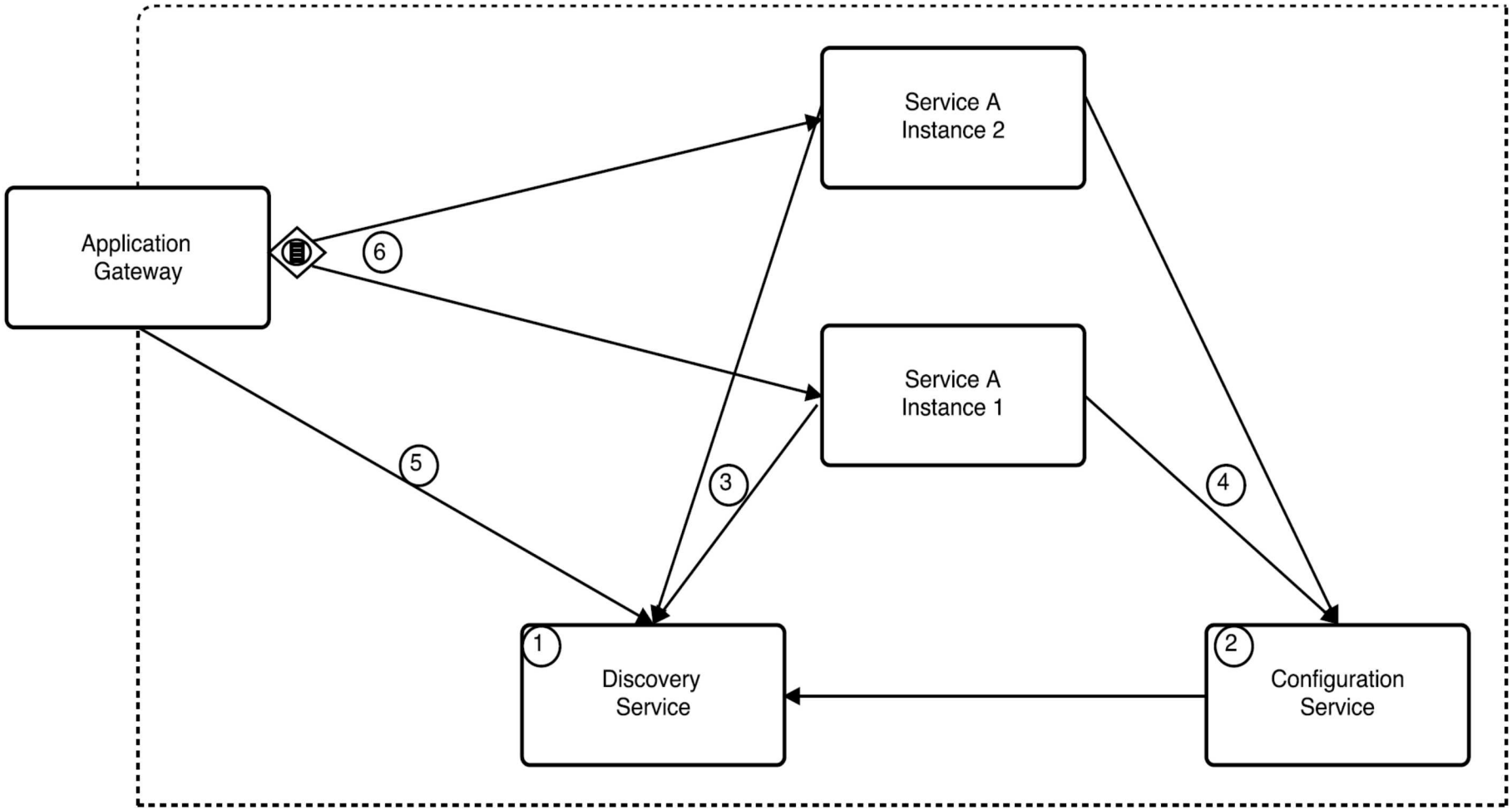
Spring Boot

- Allows developers to create "Runnable" jars to eliminate the need for an AppServer
  - Runs wherever JAVA is installed and removes the need for Middleware support or AppServer knowledge
  - No more web.xml
- Greatly Simplifies Spring Development by using Convention-over-Configuration semantics
  - Introspects pom.xml file to determine beans most likely to be needed and "Auto-Configures" those beans
  - No more application-context.xml
- Sets the "Spring Version" so that all spring Jars are at compatible and tested versions
  - Updating the Spring Boot Version updates ALL Spring Jars to associated version
  - Reduces likelihood of having multiple versions of same jar through transient dependencies

Spring Cloud

- Spring Boot sub-project used to create Micro-Services
  - Removes much of the boilerplate code required in developing the common patterns of distributed systems
- Most of the patterns discussed here are Spring Integrations with Netflix-OSS components
- Provides a number of support projects and libraries that are easily setup and deployed
  - Service Discovery (Eureka)
  - Config Server (Config)
  - Fault Tolerance (Hystrix)
  - Application Gateway (Zuul)
  - Load Balancing (Ribbon)
  - Distributed Tracing (Zipkin)

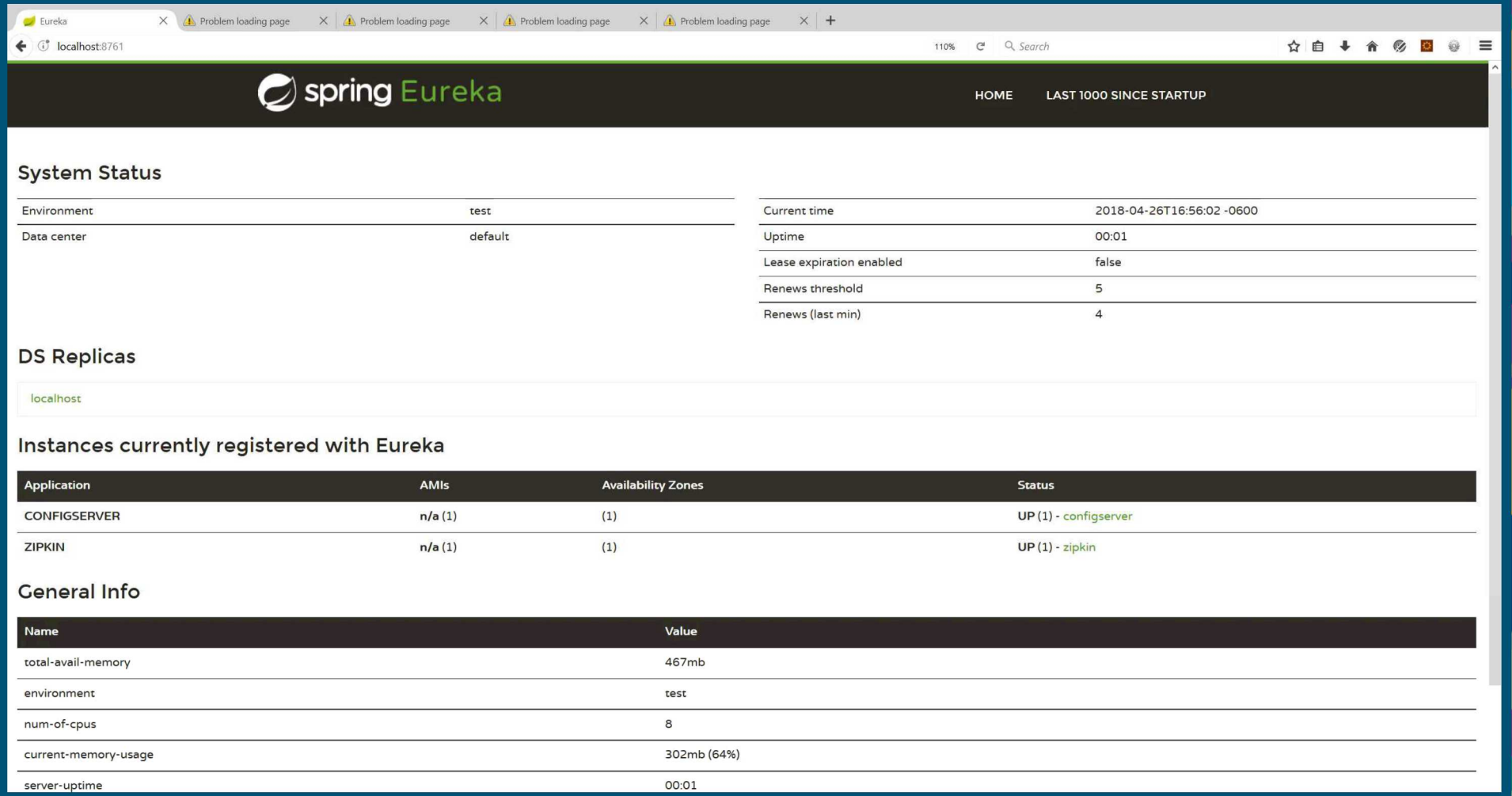
Distributed System Patterns



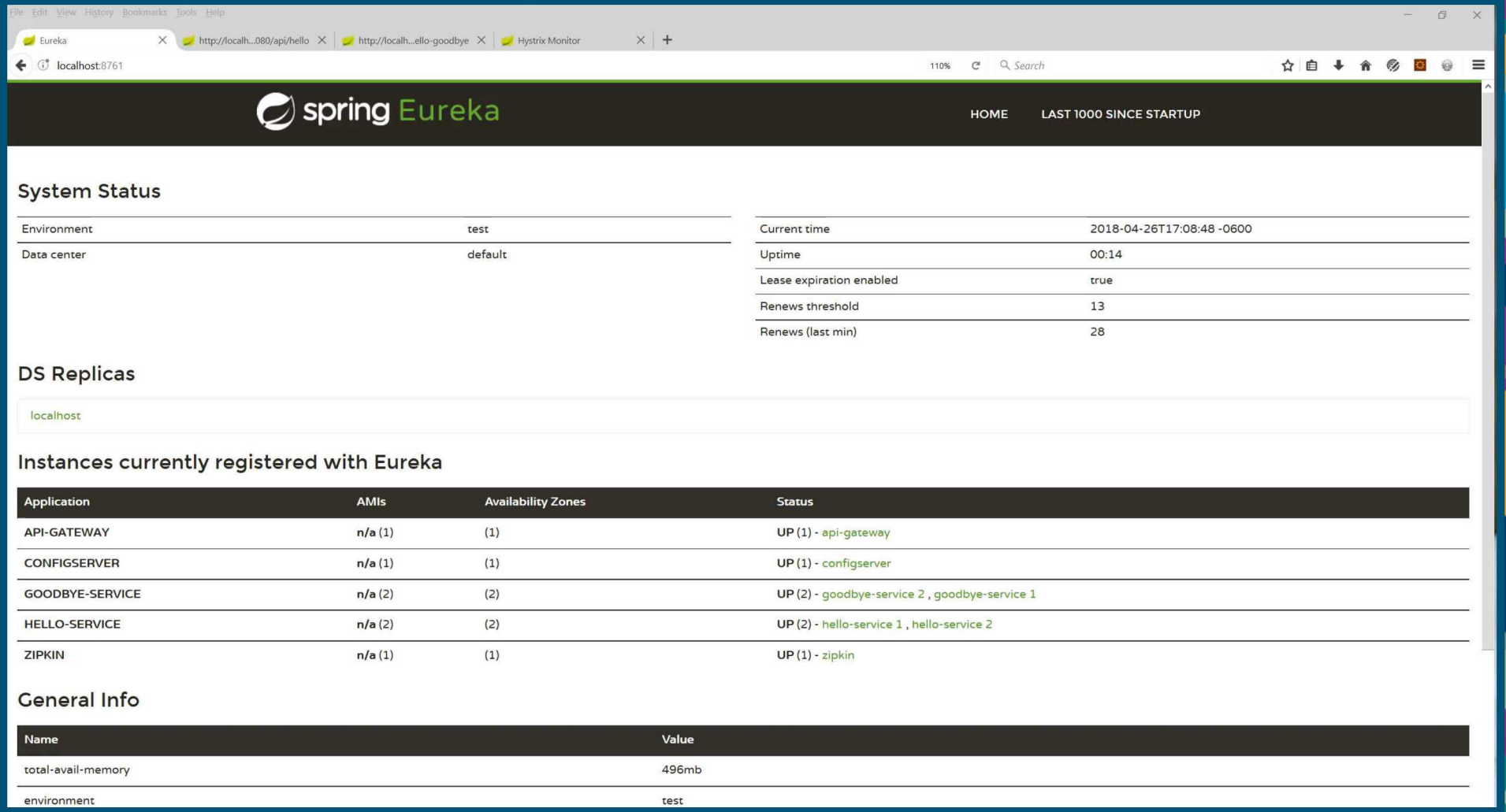
Cloud Setup


File Edit Navigate Search Project Run Window Help
Quick Access
Package Explorer
AdminConsole
ApiGateway [boot] [devtools] [spring-cloud master
CommonLib [boot] [devtools]
ConfigServer [boot] [devtools] [spring-cloud maste
DiscoveryServer [boot] [devtools] [spring-cloud ma
GoodbyeService [boot] [devtools] [spring-cloud ma
HelloService [boot] [devtools] [spring-cloud master
Spring-Cloud-Config [spring-cloud-config master]
ZipkinServer [boot] [devtools] [spring-cloud master

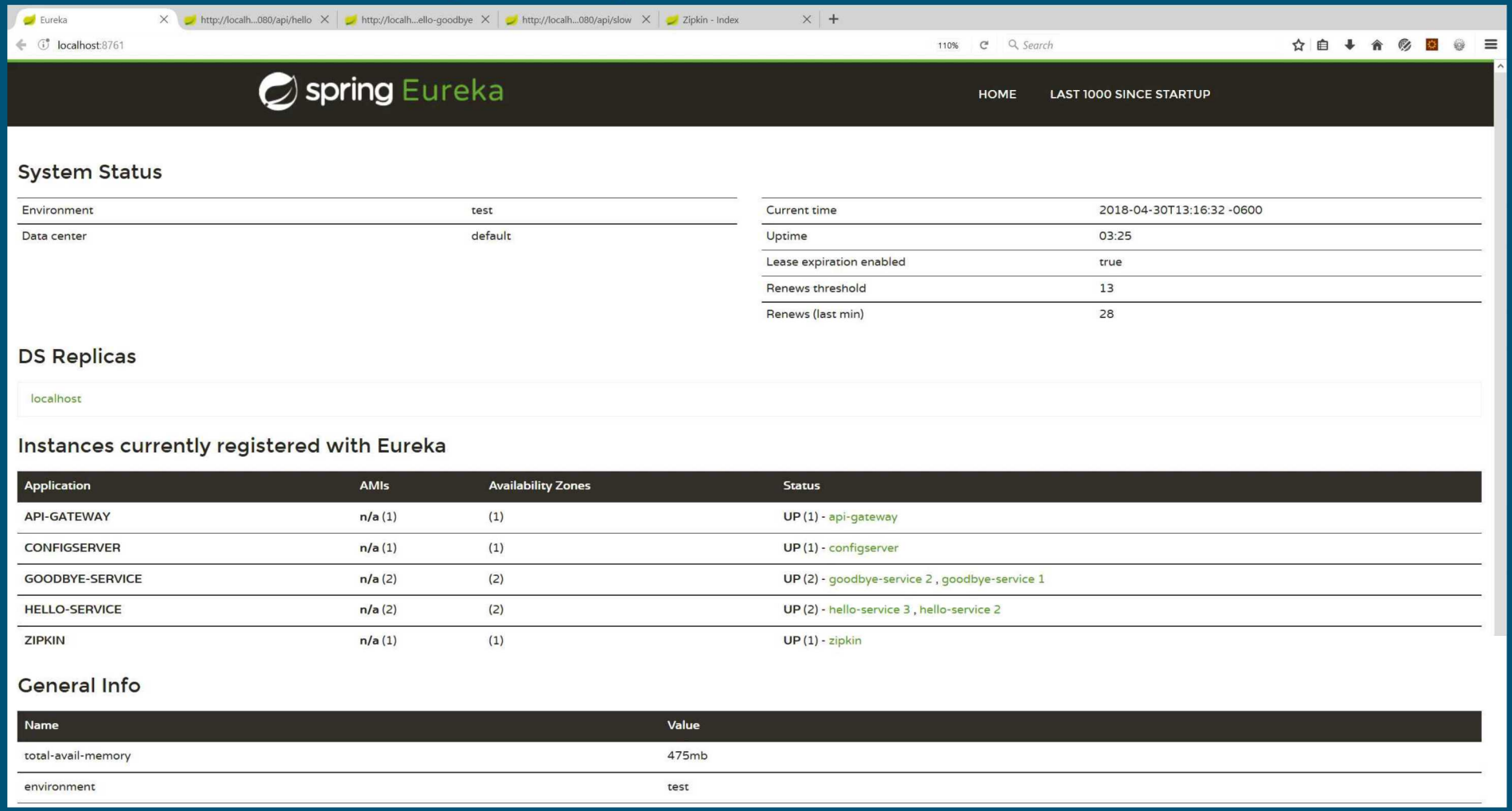
Load Balancing with Ribbon



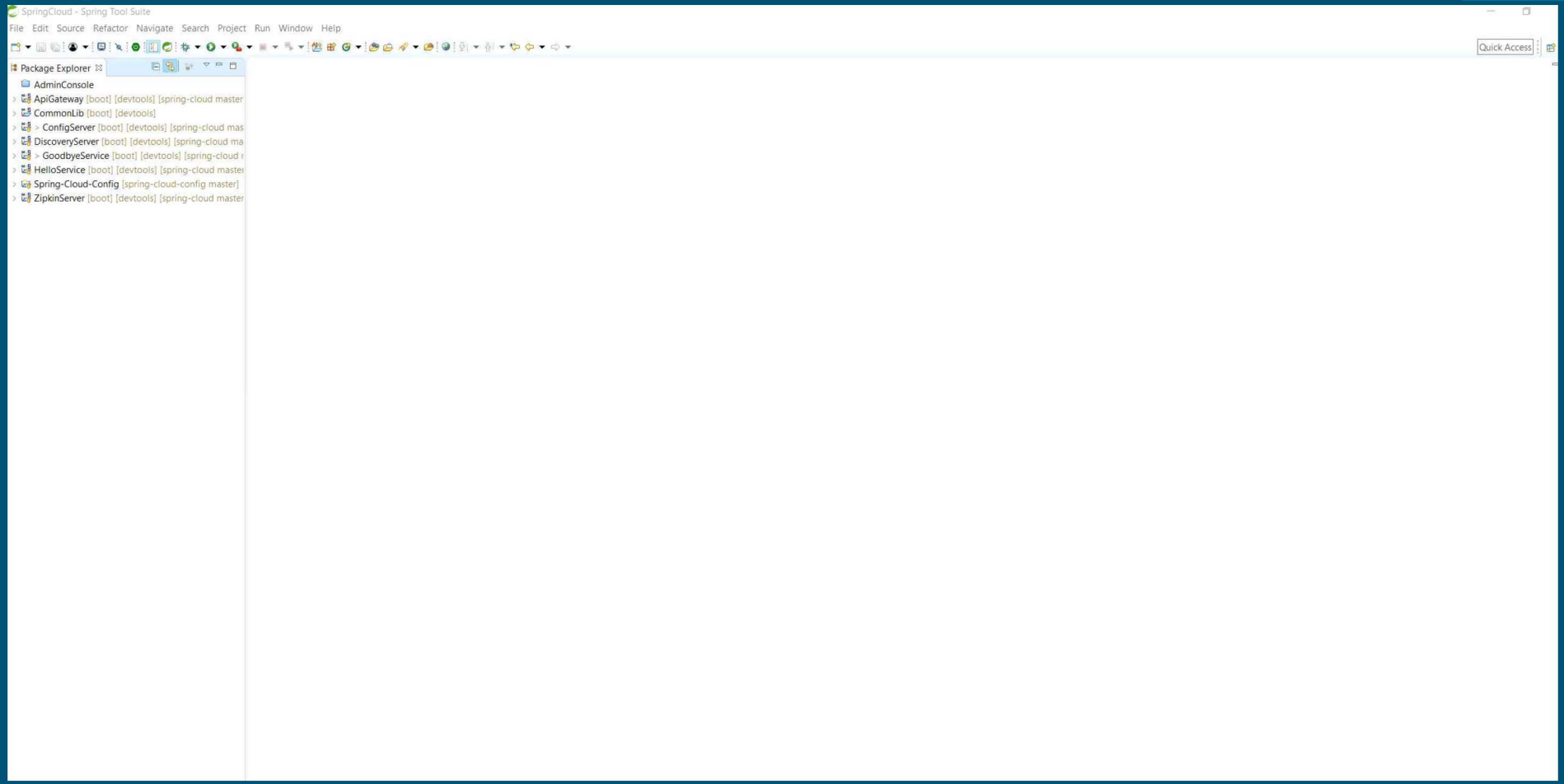
Fault Tolerance with Hystrix



Distributed Tracing with Zipkin



Code Review


SpringCloud - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
AdminConsole
ApiGateway [boot] [devtools] [spring-cloud master
CommonLib [boot] [devtools]
ConfigServer [boot] [devtools] [spring-cloud mas
DiscoveryServer [boot] [devtools] [spring-cloud ma
GoodbyeService [boot] [devtools] [spring-cloud
HelloService [boot] [devtools] [spring-cloud master
Spring-Cloud-Config [spring-cloud-config master]
ZipkinServer [boot] [devtools] [spring-cloud master

Questions?