

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2018-4932C

Developing Cloud-Native Services with the Spring Framework



PRESENTED BY

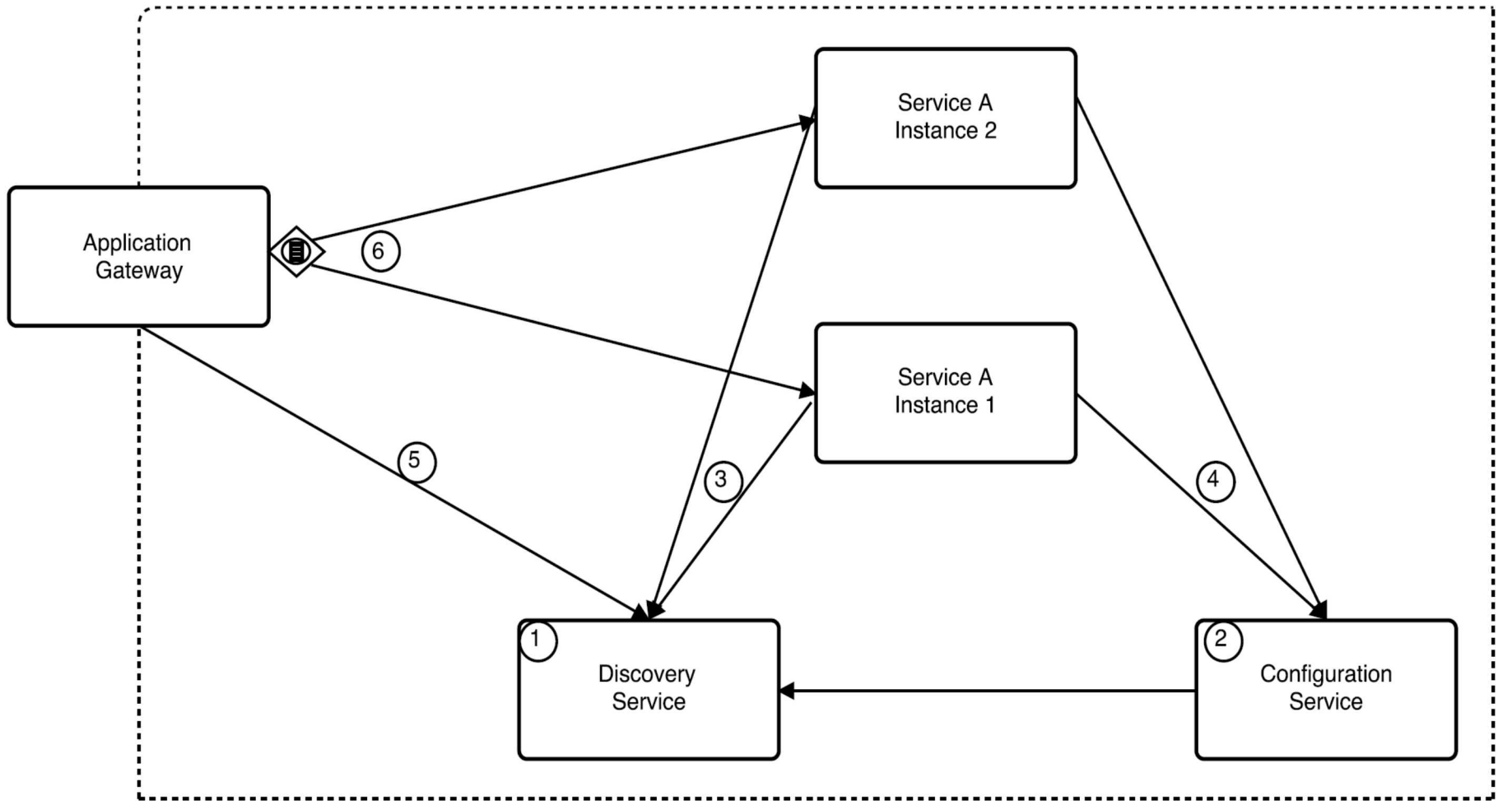
Todd Fielder

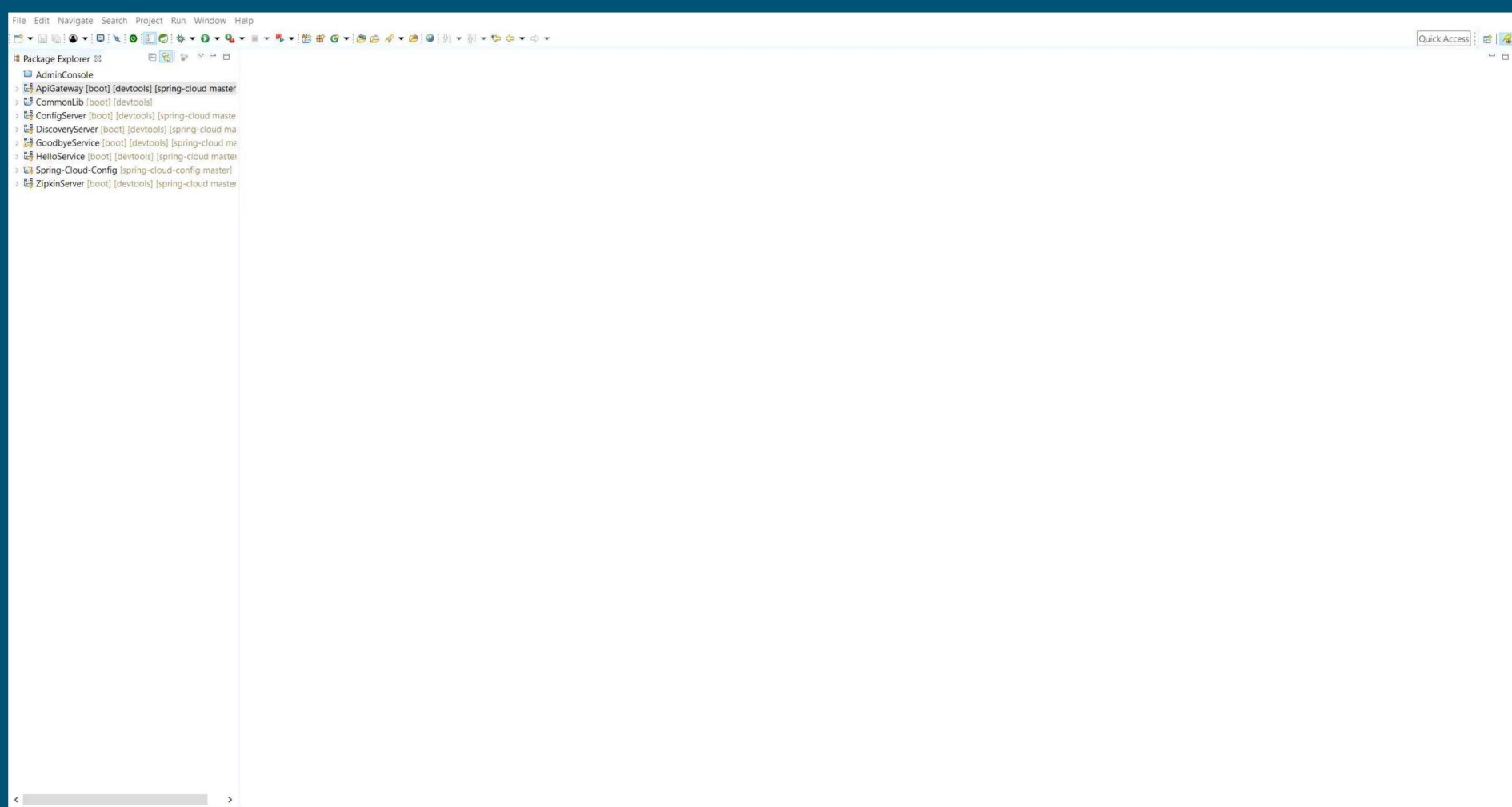
Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

- 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

- 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 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)





7 Load Balancing with Ribbon



Eureka browser interface showing system status, DS Replicas, and registered instances.

System Status

Environment	test	Current time	2018-04-26T16:56:02 -0600
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	5
		Renews (last min)	4

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
CONFIGSERVER	n/a (1)	(1)	UP (1) - configserver
ZIPKIN	n/a (1)	(1)	UP (1) - zipkin

General Info

Name	Value
total-avail-memory	467mb
environment	test
num-of-cpus	8
current-memory-usage	302mb (64%)
server-uptime	00:01

8 Fault Tolerance with Hystrix



The screenshot shows the Spring Eureka web interface. At the top, there is a navigation bar with the Spring Eureka logo and links for HOME and LAST 1000 SINCE STARTUP. Below the navigation bar, the page is divided into several sections:

- System Status:** A table showing environment (test), data center (default), current time (2018-04-26T17:08:48 -0600), uptime (00:14), lease expiration enabled (true), renews threshold (13), and renews (last min) (28).
- DS Replicas:** A search box containing the text 'localhost'.
- Instances currently registered with Eureka:** A table listing applications and their status.
- General Info:** A table showing system configuration details.

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - api-gateway
CONFIGSERVER	n/a (1)	(1)	UP (1) - configserver
GOODBYE-SERVICE	n/a (2)	(2)	UP (2) - goodbye-service 2 , goodbye-service 1
HELLO-SERVICE	n/a (2)	(2)	UP (2) - hello-service 1 , hello-service 2
ZIPKIN	n/a (1)	(1)	UP (1) - zipkin

Name	Value
total-avail-memory	496mb
environment	test

9 Distributed Tracing with Zipkin



Eureka browser interface showing system status, DS Replicas, and registered instances.

System Status

Environment	test	Current time	2018-04-30T13:16:32 -0600
Data center	default	Uptime	03:25
		Lease expiration enabled	true
		Renews threshold	13
		Renews (last min)	28

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - api-gateway
CONFIGSERVER	n/a (1)	(1)	UP (1) - configserver
GOODBYE-SERVICE	n/a (2)	(2)	UP (2) - goodbye-service 2 , goodbye-service 1
HELLO-SERVICE	n/a (2)	(2)	UP (2) - hello-service 3 , hello-service 2
ZIPKIN	n/a (1)	(1)	UP (1) - zipkin

General Info

Name	Value
total-avail-memory	475mb
environment	test



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 master]
- DiscoveryServer [boot] [devtools] [spring-cloud master]
- GoodbyeService [boot] [devtools] [spring-cloud master]
- HelloService [boot] [devtools] [spring-cloud master]
- Spring-Cloud-Config [spring-cloud-config master]
- ZipkinServer [boot] [devtools] [spring-cloud master]

