Discovery Service

Eureka Discovery has been decommissioned.

- Summary
- Technical Design
  - Technical Architecture Diagram
  - Overview
  - Eureka Client Registration
    - pom.xml
    - bootstrap.properties
    - bootstrap-${profile}.properties
    - bootstrap-local.properties
    - application.properties
    - Application.java
    - Clients deployed in a Weblogic Environment
  - Testing
    - Runtime Configuration
    - bootstrap-local.properties
  - Security
  - References

Summary

Key	Value
Name	OAL-Discovery-Server
OAL Owner	srinivas.nagesh@oracle.com or raju.v.krishnamoorthy@oracle.com
Git Repository	DOCKER-OalcnDiscoveryServer.git
Deployment Context	/oalapp/services/oalcn-discovery-server
Does it expose REST API ?	Actuator: https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html Eureka: https://github.com/Netflix/eureka/wiki/Eureka-REST-operations The Eureka endpoints in the link contain a version in the URI. OAL Discovery Service doesn't have the version in the URI.
Security	None
AppCode	DISC
Interaction with external Systems	None
Root Java Package	oal.oracle.apps.ic.discovery

Technical Design

Technical Architecture Diagram

? Unknown Attachment

Overview

OAL-Discovery-Service is based on Open Source Netflix Eureka. Discovery and Registration are key in Elastic Cloud based Architectures where Services are distributed and come and go. A service registry is useful because it enables client-side load-balancing and decouples service providers from consumers without the need for DNS. Discovery serves as single source of directory for all Services to register themselves and also discover other Services. This is considered to be the static point in an otherwise infinitely* scalable cloud architecture.

Eureka Client Registration

Application Clients can register with Discovery with very minimal configuration changes.

pom.xml

The latest version of supported Eureka Client derived via Spring Cloud is documented below. Please ensure that your Client has the below dependency versions.

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.1.2.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  <java.version>1.8</java.version>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
  <spring-cloud.version>Greenwich.RELEASE</spring-cloud.version>
</properties>
```

By default Eureka Client uses Jersey for HTTP Communication. Due to Jersey v1 incompatibility with Weblogic, we have decided to exclude the Jersey dependency. This allows Spring Cloud to auto-configure RestTemplate for HTTP communication.

```
<!-- Service Discovery -->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
  <exclusions>
    <exclusion>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-client</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-core</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jersey.contribs</groupId>
      <artifactId>jersey-apache-client4</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

bootstrap.properties

```
# Discovery Configuration
eureka.client.service-url.defaultZone=http://${eureka.client.service-url.host1}${eureka.client.service-context}
/eureka,http://${eureka.client.service-url.host2}${eureka.client.service-context}/eureka
eureka.client.service-context=/oalapp/services/oalcn-discovery-server
```

bootstrap-${profile}.properties

```
# Discovery Configuration
eureka.client.service-url.host1=
eureka.client.service-url.host2=
```

bootstrap-local.properties

```
# Disable Discovery for local profile
eureka.client.enabled=false
```

application.properties

> ⚠ The instance and the registration properties noted below should be externalized to the Config Server. Do not bundle these properties with the Client Application.

```
# Discovery Configuration
# Instance Configuration
eureka.instance.hostname=${HOST_NAME}
eureka.instance.non-secure-port=${server.port}
eureka.instance.instance-id=${eureka.instance.hostname}:${spring.application.name}:${server.
port}:${SPRING_PROFILES_ACTIVE}
# Client Configuration
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.healthcheck.enabled=true
```

Application.java

```
@EnableDiscoveryClient
public class Application {
...
}
```

Clients deployed in a Weblogic Environment

Add the below entry to weblogic.xml (<prefer-application-packages>) section. This will tell Weblogic to ignore any default built in Jersey libraries. Since the application Jersey dependencies are excluded, Jersey will not be available in the environment.

```
<package-name>com.sun.jersey</package-name>
```

Testing

To test Application Client with Eureka with a Discovery Server deployed in PAAS:

Runtime Configuration

Set the following VM Options when starting up Eureka Client on local

```
-Dhttp.proxyHost=www-proxy.us.oracle.com -Dhttp.proxyPort=80
```

bootstrap-local.properties

Connect to a Remote Discovery Service by specifying the host and port

```
# Discovery Configuration
eureka.client.service-url.host1=
eureka.client.service-url.host2=
eureka.client.enabled=true
```

Validate that the Application Client is registered with Eureka by visiting the Discovery Dashboard.

Security

OAL Discovery Dashboard and API are open to all consumers in OAL PAAS Environments.

References

Environment Map

https://spring.io/guides/gs/service-registration-and-discovery/

https://cloud.spring.io/spring-cloud-netflix/multi/multi__service_discovery_eureka_clients.html

https://cloud.spring.io/spring-cloud-netflix/multi/multi_spring-cloud-eureka-server.html

https://github.com/Netflix/eureka/wiki/Eureka-at-a-glance

https://blog.asarkar.org/technical/netflix-eureka/#

https://medium.com/@fahimfarookme/the-mystery-of-eureka-health-monitoring-5305e3beb6e9

https://medium.com/@fahimfarookme/the-mystery-of-eureka-self-preservation-c7aa0ed1b799