



PREPÁRATE
PARA SER EL
MEJOR



+ **ENTREMIENTO
EXPERIENCIA**



BIENVENIDOS.



Microservicios con .NET Core: Arquitectura, Contenerización y Orquestación

Sesión 06

Ing. Erick Arostegui Cunza
Instructor

earostegui@galaxy.edu.pe



AGENDA

ARQ. DE MICROSERVICIOS (ASPECTOS TRASNVER. Y OBSERVABILIDAD)

- ▶ Asegurando los microservicios (OpenID, OAuth y JWT).
- ▶ Registro y Discovery de microservicios.
- ▶ Monitoreo y estado de salud de los microservicios (HealthChecks).
- ▶ Centralización de log's.



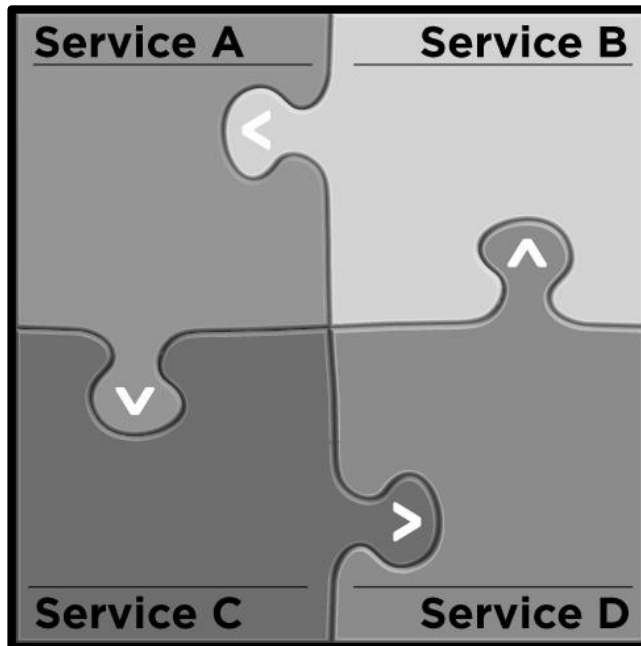
Encontrar servicios utilizando el Services Discovery



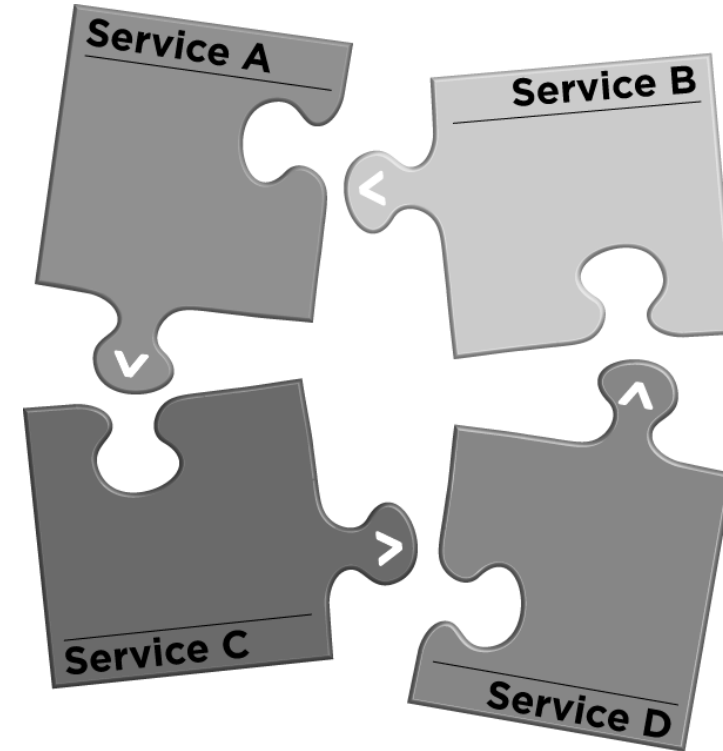
¿Qué es Service Discovery y por qué lo necesitamos?



Cambios en la forma en que desarrollamos software



Desde una aplicación
monolítica



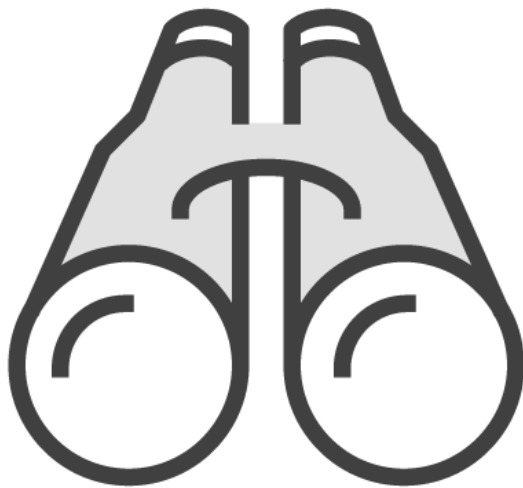
Para servicios desplegables
individualmente



El problema: ¿cómo un servicio ubica a otro?



Servicio de
aplicación A



Localización?



Servicio de
aplicación B



El enfoque simple: a través de la configuración





Multiple Instancias



Servicio de aplicación A

HTTP
REST →

Instancia 1



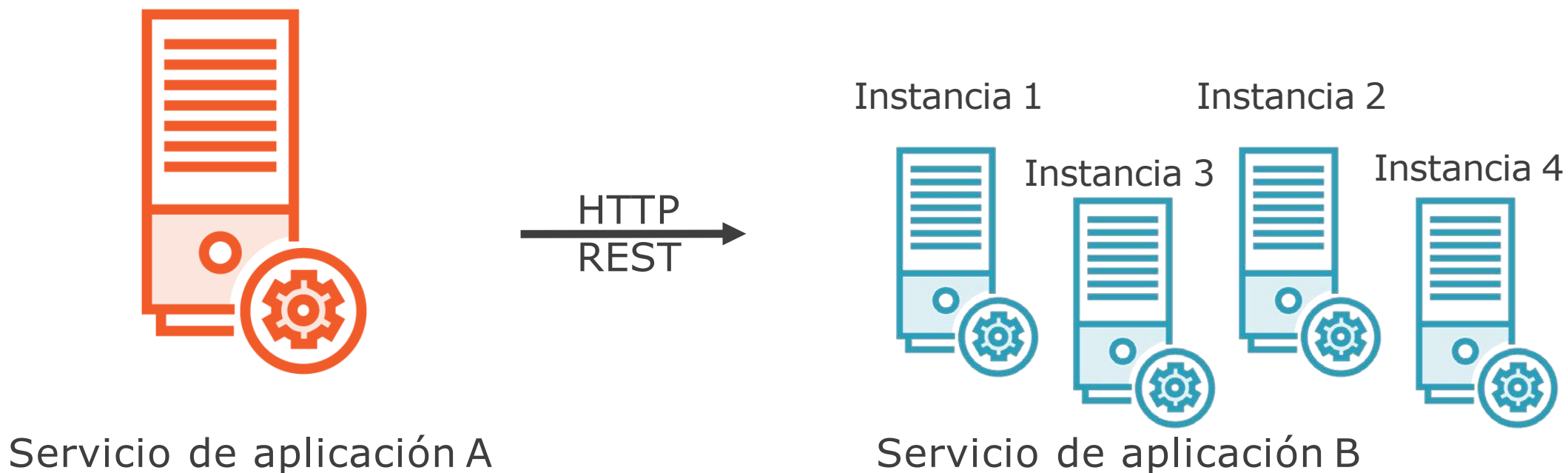
Servicio de aplicación B

Instancia 2





Las instancias van y vienen en respuesta a la demanda





Las instancias fallan



Servicio de aplicación A

HTTP
REST →

Instancia 1



Instancia 2



Servicio de aplicación B



El enfoque simple: a través de la configuración

¡El enfoque simple es
demasiado
estático (congelado
en el tiempo) para la
nube!





El enfoque simple: a través de la configuración



El descubrimiento de servicios proporciona

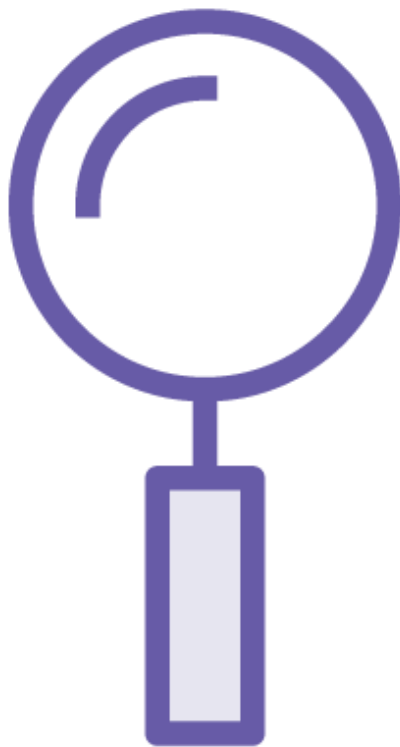
- Una forma para que un servicio se registre
- Una forma para que un servicio se desregistre
- Una manera para que un cliente encuentre otros servicios
- Una forma de verificar el estado de un servicio y eliminar instancias caídas



Discovering Services con Spring Cloud



El enfoque simple: a través de la configuración



Discover services con:

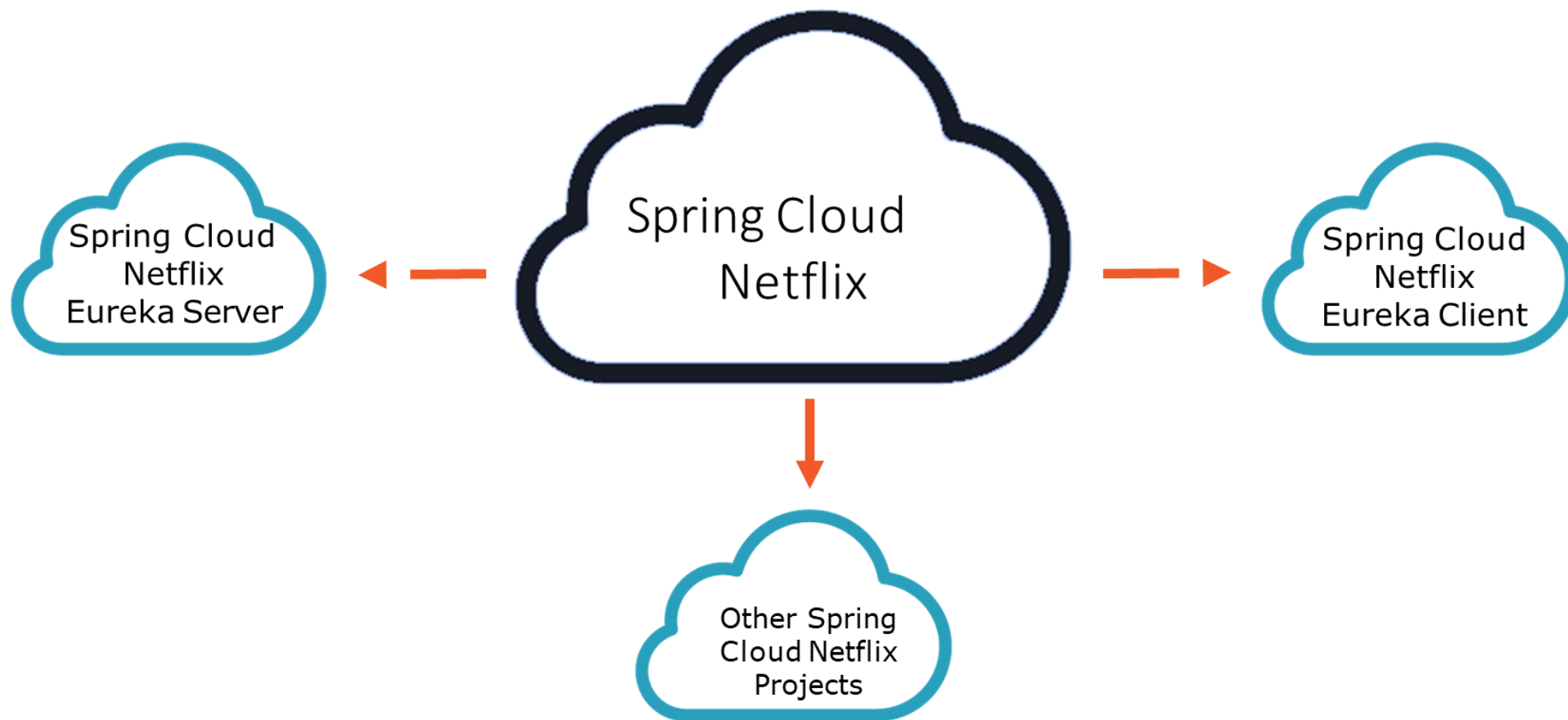
- Spring Cloud Consul
- Spring Cloud Zookeeper
- Spring Cloud Netflix



Netflix OSS + Spring + Spring Boot
=
Spring Cloud Netflix



Registro y Discovery de microservicios.





Componentes clave en Service Discovery

Discovery Server



Service



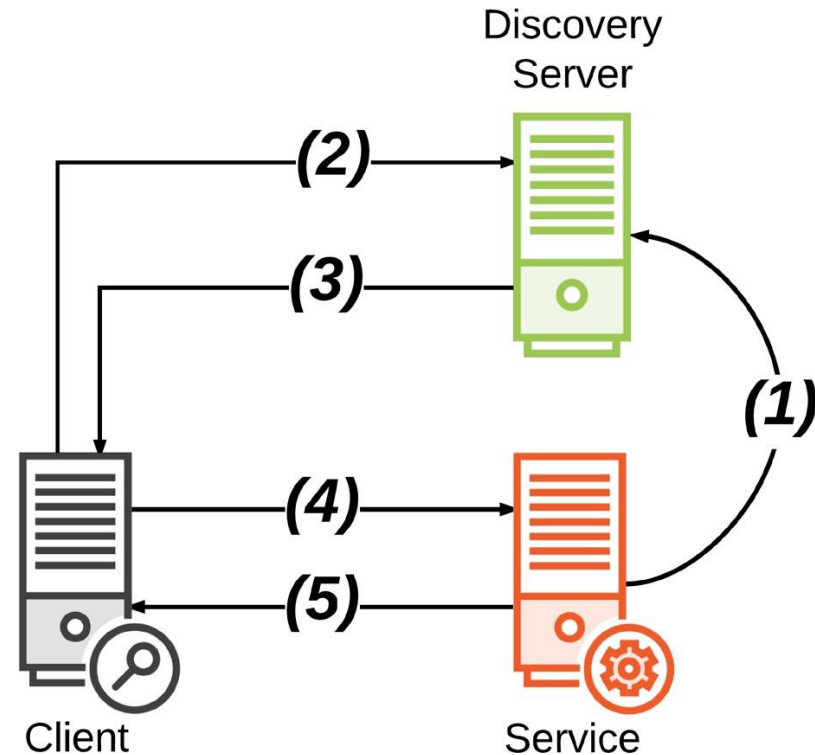
Client





El enfoque simple: a través de la configuración

1. El servicio registra la ubicación
2. El cliente busca la ubicación del servicio
3. El servidor de descubrimiento devuelve la ubicación
4. El cliente solicita servicio en la ubicación
5. El servicio envía respuesta





El enfoque simple: a través de la configuración

Discovery



Server

Un registro de ubicaciones de servicio
gestionado activamente

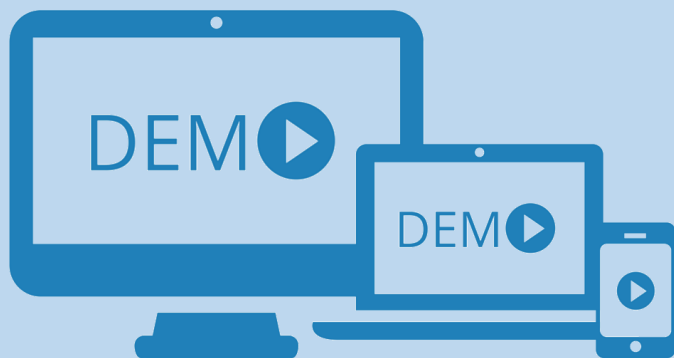
Un solo origen para una o más
instancias

Proyecto Spring Cloud:

- Servidor Spring Cloud Eureka



Registro y Discovery de microservicios.



Creando e iniciando un discovery server



Servicio de que se conecta a un Discovery Server

Application

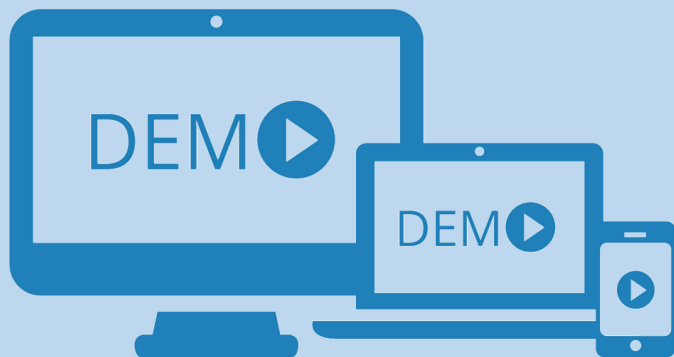


Service

- Proporcionar algunas funciones de la aplicación.
- El receptor de request
- Una dependencia de otros servicios Una o más instancias
- Uso del discovery client
 - Registrarse
 - Darse de baja



Registro y Discovery de microservicios.



Creando e iniciando un discovery server



Cliente que consume a un Discovery Server

Application

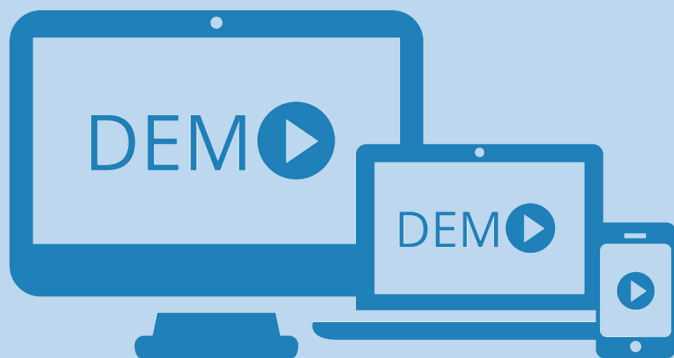


Client

- Llama a otro servicio de aplicación para implementar su funcionalidad
- El emisor de las solicitudes depende de otros servicios
- Usuario del cliente de descubrimiento
- Encuentra ubicaciones de servicio



Registro y Discovery de microservicios.



Creating a client that can discover services

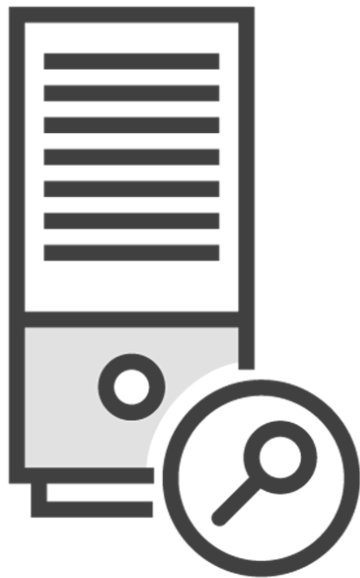


Spring Cloud Eureka Dashboard



Cliente que consume a un Discovery Server

Application



Client

Habilitado por defecto

- `eureka.dashboard.enabled=true`

Muestra metadatos útiles y estado del servicio



Servidor Eureka: ¿Estan saludables mis servicios?



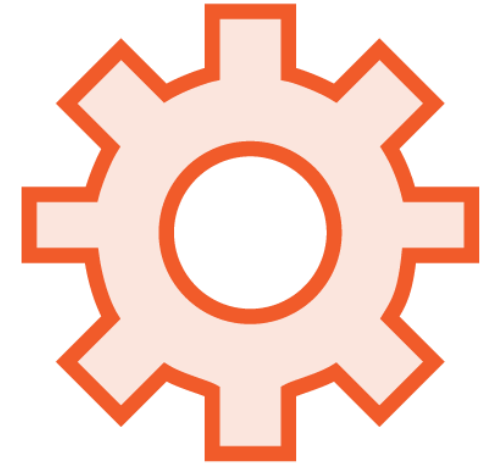
Comprueba regularmente el estado de los servicios.



Los clientes envían latidos cada 30 segundos (predeterminado)



Servicios eliminados después de 90 segundos sin latidos (predeterminado)

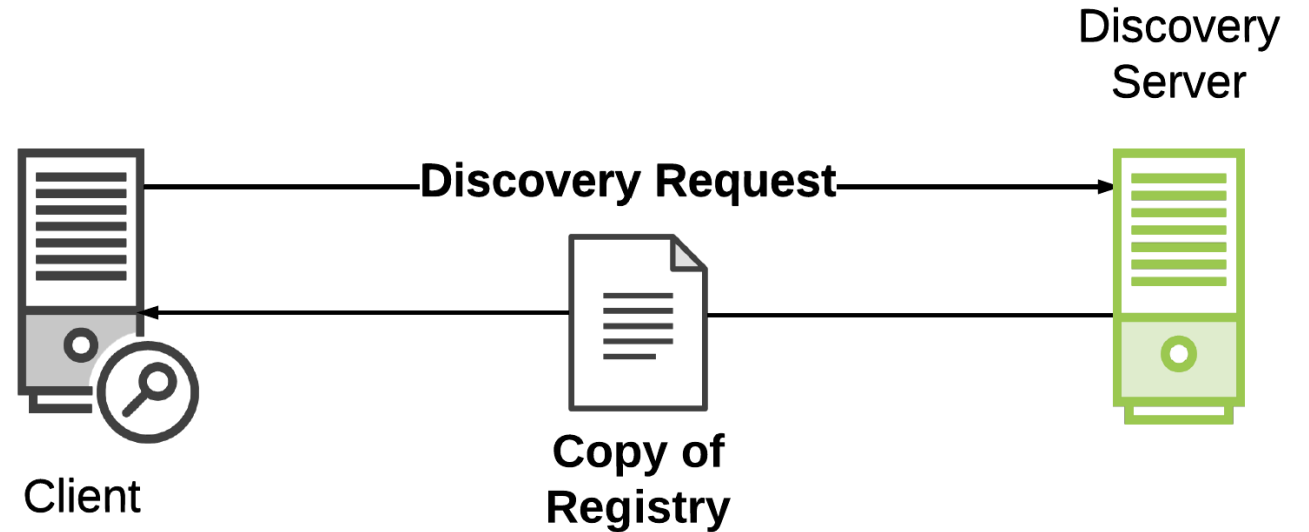


Puede personalizar la configuración para usar
/health endpoint
eureka.client.
healthcheck.enabled



Servidor Eureka: ¿Estan saludables mis servicios?

- El registro se distribuye (almacena en caché localmente en cada cliente)
- Los clientes pueden operar sin servidor de descubrimiento
- Obtiene deltas para actualizar el registro





Spring Cloud Eureka AWS Support



Spring Cloud Eureka AWS Support

AWS-specific instance data

Multi-zone aware

Multi-region aware

Elastic IP Binding



EC2 Dashboard

us-east-1

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

1 to 2 of 2

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
<input checked="" type="checkbox"/>	spring-cloud-discovery-server-1	i-782b7576	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None
<input type="checkbox"/>	spring-cloud-discovery-server-2	i-b9bcba41	t2.micro	us-east-1e	● running	✓ 2/2 checks ...	None

```
eureka.client.availability-zones.us-east-1=us-east-1b,us-east-1e
```

Availability Zones Configuration in `application.properties`

```
eureka.client.availability-zones.[region]=[az1],[az2],[az3]
```



Elastic IP Dashboard

us-east-1

Allocate New Address

Actions ▾

↺

⚙

?

Filter

VPC addresses ▾

🔍

Search Elastic IPs...

✕

⏪ < 1 to 2 of 2 Elastic IPs > ⏩

<input type="checkbox"/>	Address ▴	Allocation ID ▾	Instance ID ▾	Network Interface ID ▾	Scope ▾	Private Address ▾
<input type="checkbox"/>	34.192.167.121	eipalloc-bdf9e782	i-782b7576	eni-2d3231d2	vpc	172.31.52.243
<input type="checkbox"/>	34.193.24.166	eipalloc-59e3fd66	i-b9bcba41	eni-f1e0c418	vpc	172.31.33.117

```
eureka.client.service-url.us-east-1b=  
    http://ec2-34-192-167-121.compute-1.amazonaws.com:8761/eureka/  
eureka.client.service-url.us-east-1e=  
    http://ec2-34-193-24-166.compute-1.amazonaws.com:8761/eureka/
```

Service URL Configuration in `application.properties`

```
eureka.client.service-url.[zone]=http://[eip-dns]/eureka  
- *Use EIP DNS name. Do not use IP (as of version Eureka 1.4)
```



Eureka Dashboard: AWS Multi-zone Discovery Servers

DS Replicas

ec2-34-192-167-121.compute-1.amazonaws.com

ec2-34-193-24-166.compute-1.amazonaws.com

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DISCOVERY-SERVER-1	ami-40d28157 (1)	us-east-1b (1)	UP (1) - i-782b7576
DISCOVERY-SERVER-2	ami-40d28157 (1)	us-east-1e (1)	UP (1) - i-b9bcba41

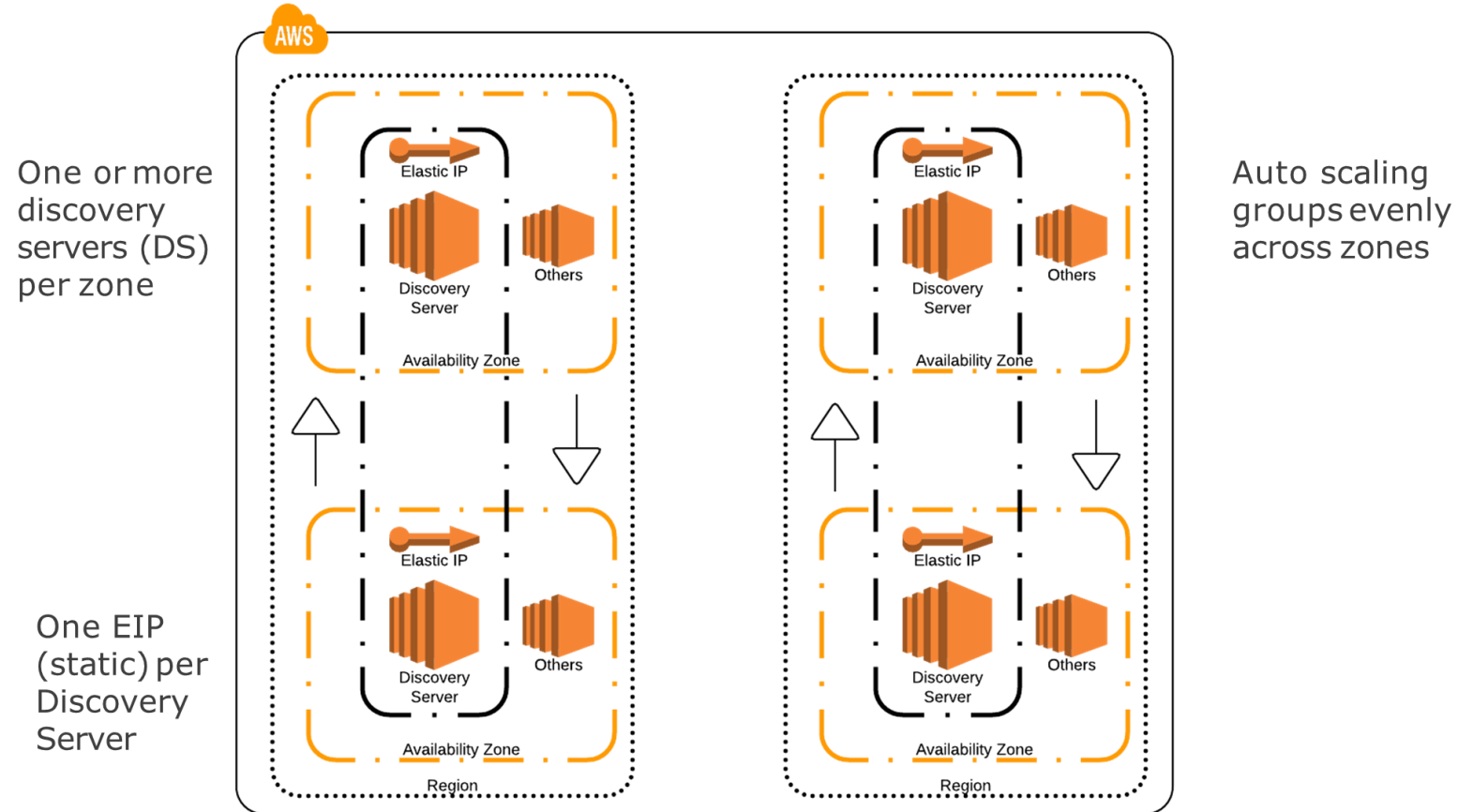


Eureka Dashboard: AWS Instance Data

Instance Info	
Name	Value
public-ipv4	34.192.167.121
public-hostname	ec2-34-192-167-121.compute-1.amazonaws.com
instance-id	i-782b7576
instance-type	t2.micro
ami-id	ami-40d28157
ipAddr	172.31.52.243
status	UP
availability-zone	us-east-1b



Eureka Dashboard: AWS Instance Data





GALAXY
TRAINING