

Servicio Orders - Testing de escalabilidad

Propósito:

El objetivo de este testing es probar la capacidad de escalado horizontal del servicio Orders a la hora de recibir una cantidad de peticiones, para así saber si el mismo puede manejar cargas aumentadas de información.

Contenido:

- **Objetivos de la prueba:** El propósito de la prueba de escalabilidad es validar que el servicio sea capaz de escalar horizontalmente al recibir una mayor cantidad de peticiones de lo normal.
- **Resultado Clave:** El servicio es capaz de escalar horizontalmente acorde a la demanda de peticiones por los usuarios según la configuración aplicada, sin verse afectado el servicio. Posteriormente a la caída de la demanda, el servicio des escala correctamente.

Entorno de Pruebas

Se realiza las pruebas en instancias con CPU de 256 unidades y 512 MegaBytes de memoria RAM.

Contenido:

- **Descripción del Entorno:** Los cluster donde trabajan las instancias son de tipo "FRAGATE", las configuraciones de red están completamente abiertas tanto la entrada como salida. Cuenta con un balanceador de carga.
- La configuración del auto escalado sea realiza teniendo 1 tarea como mínimo y un máximo de 3 tareas escaladas. La política de escalado es de tipo seguimiento de destino, con una métrica de peticiones con un valor de 2000 peticiones y perdidos de recuperación de escalado y desescalada horizontal de 30 segundos.
- Se utiliza la herramienta JMeter (V 5.6.3) para realizar las pruebas de carga.

Ajuste automáticamente el recuento deseado del servicio hacia arriba y hacia abajo dentro de un rango especificado en respuesta a las alarmas de CloudWatch. Puede modificar la configuración del escalado automático de servicios en cualquier momento para satisfacer las necesidades de la aplicación.

Configurar el escalado automático de servicios para ajustar el recuento deseado del servicio

El límite inferior al que el escalado automático de servicios puede ajustar el recuento deseado del servicio.

1

El límite superior al que el escalado automático de servicios puede ajustar el recuento deseado del servicio.

3

Eliminar

Seguimiento de destino

Auto-Scaling-Policy-dev-orders-service

ALBRequestCountPerTarget

2000

30

30

☐ Desactivar la acción de desescalar horizontalmente

- Escenario de prueba: 3000 usuarios simultáneos en un tiempo de Ramp up de 30 segundos durante 10 minutos.

The screenshot shows the JMeter GUI configuration for a Thread Group named "Grupo de Hilos". The "Action to be taken after a Sampler error" section has four radio button options: "Continue" (selected), "Start Next Thread Loop", "Stop Thread", "Stop Test", and "Stop Test Now". The "Thread Properties" section includes fields for "Number of Threads (users)" set to 3000, "Ramp-up period (seconds)" set to 30, and "Loop Count" set to "Infinite" (checked). There are three checked checkboxes: "Same user on each iteration", "Specify Thread lifetime", and "Delay Thread creation until needed". The "Duration (seconds)" field is set to 600, and the "Startup delay (seconds)" field is empty.

Resultado de la prueba

- Se observa en el reporte de Jmeter el tiempo transcurrido de 10 minutos y un envío total de 86 mil peticiones aproximadamente con un índice de éxito del 100%.

Summary Report

Name: Summary Report

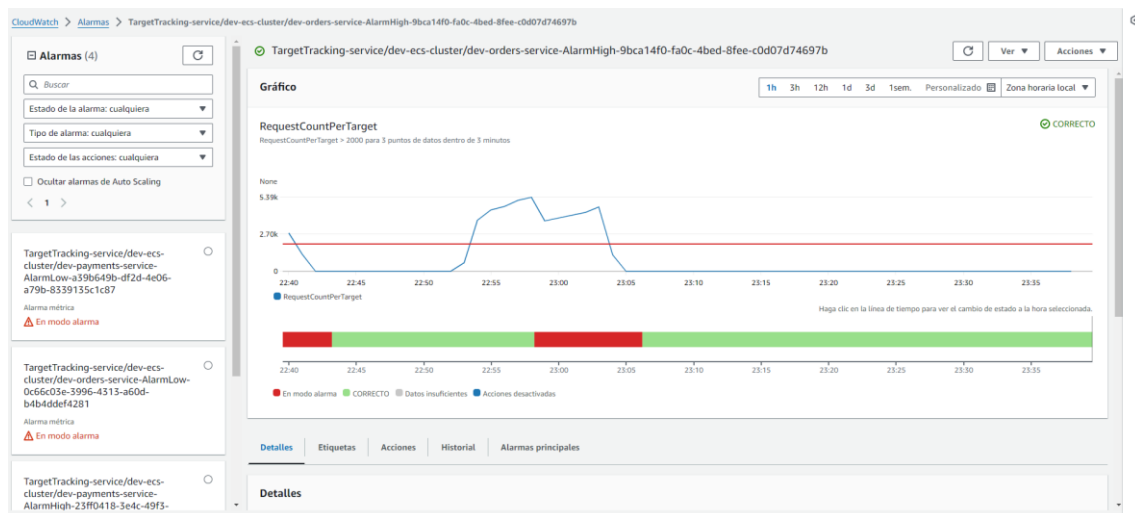
Comments:

Write results to file / Read from file

Filename: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Selected HTTP POST	86168	20805	138	46249	13577.98	0.08%	137.0/sec	32.03	34.63	239.4
TOTAL	86168	20805	138	46249	13577.98	0.08%	137.0/sec	32.03	34.63	239.4

- En cloud watch se observan picos de 5.39 mil peticiones anteriores al escalado.



- Se puede observar como de una instancia se pasa a tres instancias en ejecución paralela, indicando el correcto escalamiento según la demanda que se tenía y la configuración.

