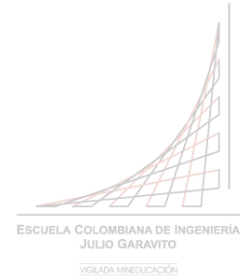


INFORME - LAB 09

ARSW



1. ESCENARIO DE CALIDAD DE LA ESCALABILIDAD

Cuando un conjunto de usuarios consulta un enésimo número (superior a 1000000) de la secuencia de Fibonacci de forma concurrente y el sistema se encuentra bajo condiciones normales de operación, todas las peticiones deben ser respondidas y el consumo de CPU del sistema no puede superar el 70%.

2. ANÁLISIS RESULTADOS CON 10 PETICIONES CONCURRENTES

Agregamos la colección de postman ajustada a la función y su environment con la url de la misma.

A. NEWMAN

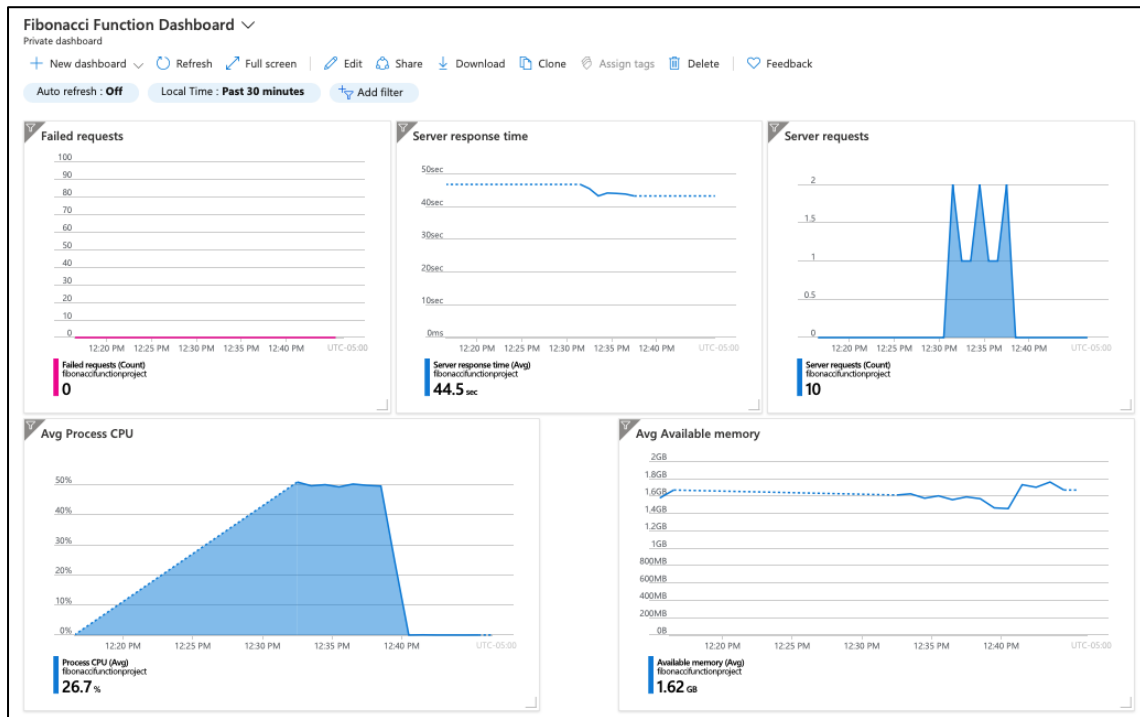
```
→ fibonacci
GET https://fibonaccifunctionproject.azurewebsites.net/api/fibonacci?nth=1000000 [200 OK, 204.32KB, 43.9s]
```

	executed	failed
iterations	10	0
requests	10	0
test-scripts	10	0
prerequest-scripts	0	0
assertions	0	0
total run duration: 7m 32.7s		
total data received: 1.99MB (approx)		
average response time: 45.1s [min: 43.6s, max: 48.6s, s.d.: 1459ms]		

En las ejecuciones paralelas realizadas se obtuvo 10/10 peticiones exitosas con un tiempo de respuesta promedio de 45s.

B. COMPORTAMIENTO DE LA FUNCIÓN EN AZURE

Dashboard



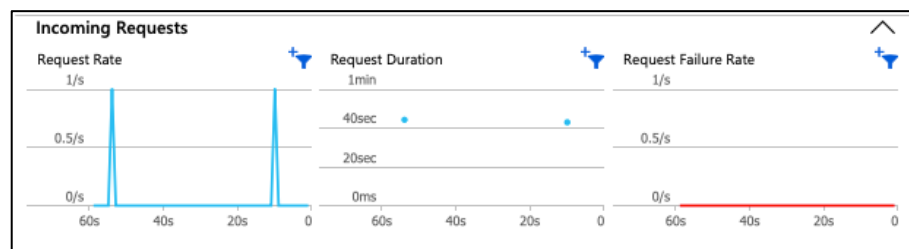
Se obtuvo en promedio un consumo de CPU del 26,7%, y en memoria de 1,62GB y todas las solicitudes exitosas.

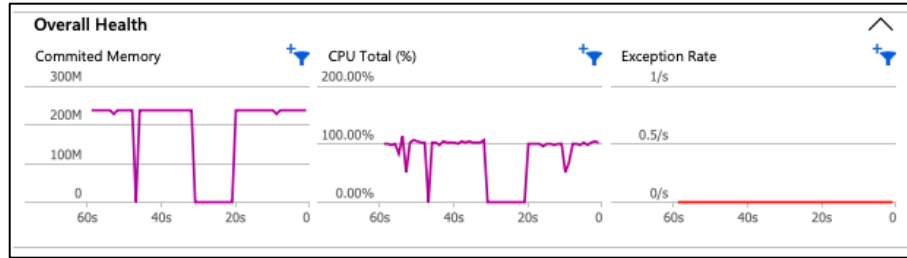
Performance

CLOUD ROLE/INSTANCE	CPU	AVAILABLE MEMORY	PROCESS IO RATE	DEPENDENCIES	AVG DEPENDENCY ...	REQUESTS	AVG REQ DURATION	ACTIONS
fibonaccifunctionproject	53.41%	1.51 GB	16.65 KB/s	0	0.0 ms	10	44 sec	
fd39869222317c16b91f1e5f	53.41%	1.51 GB	16.65 KB/s	—	—	10	44 sec	

El promedio del consumo de CPU no supero el 60%, procesando alrededor de 17 KB/s en 44 s

Live Metrics





El rendimiento de la función fue muy variable, en consumo y memoria.

Conclusión

En conclusión, se lograron 10/10 peticiones concurrentes exitosas, y como se muestra su comportamiento en live metrics con el servicio serverless, el consumo de CPU aumentaba y reducía, aún cuando superó el 70% en un tiempo aprox de 45 s.