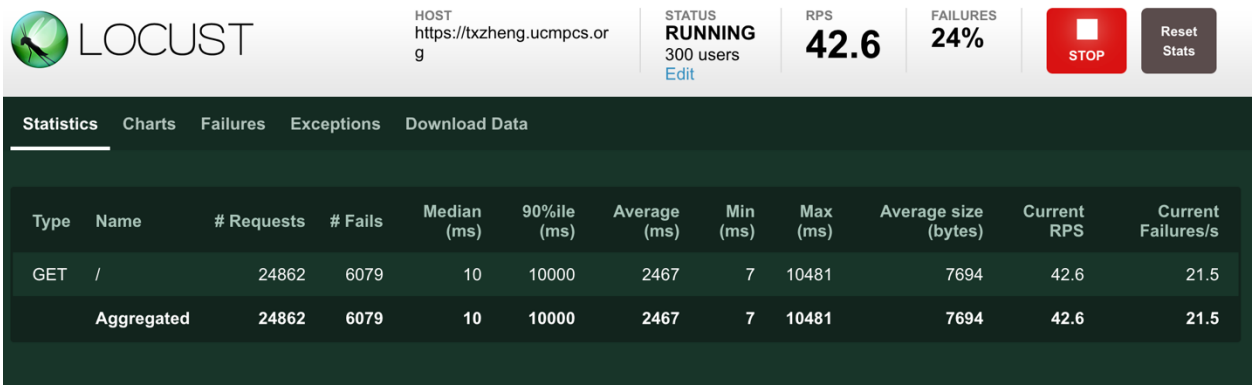
13) I started the load test for the web server with 100 users and 5 users/second, there was no failed requests and the scale-out alarm was triggered so the number of instances was scaled to 3. Then I changed to 100 users and 5 users/second. There appeared some failures and the failure rate was around 20% ~ 30%, as shown in figure1. The scale-out alarm is still shown in the in-alarm state and number of instances grew slowly as the alarm was triggered and it hasn't reached the maximum number of instances yet. Since I set the "And then wait" parameter to 300 seconds, the number of instances grew slowly at around 5 minutes per new instance, as shown in figure 2.
After it reached the maximum number of instances 10, I stopped the locust request. I expected the scale-in alarm would be triggered soon, but the two web server alarms were both showing “insufficient data” state for several minutes, after which I realized I still needed to make requests. So I set the locust load test with 1 user and 1 user/second and the scale-in alarm was triggered. The number of instances was decreasing at a fast rate as I set the "And then wait" parameter to 0, it only took the time to terminate an instance. When the number of instances was reduced to 5, it stopped decreasing as the target response time was >= 0.01 and scale-in alarm was in “OK” state. After around half an hour as the target response time was below 0.01 and the scale-in alarm was triggered and the number of instances decreased to 2 in the end.



Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	24862	6079	10	10000	2467	7	10481	7694	42.6	21.5
	Aggregated	**24862**	**6079**	**10**	**10000**	**2467**	**7**	**10481**	**7694**	**42.6**	**21.5**

Figure 1

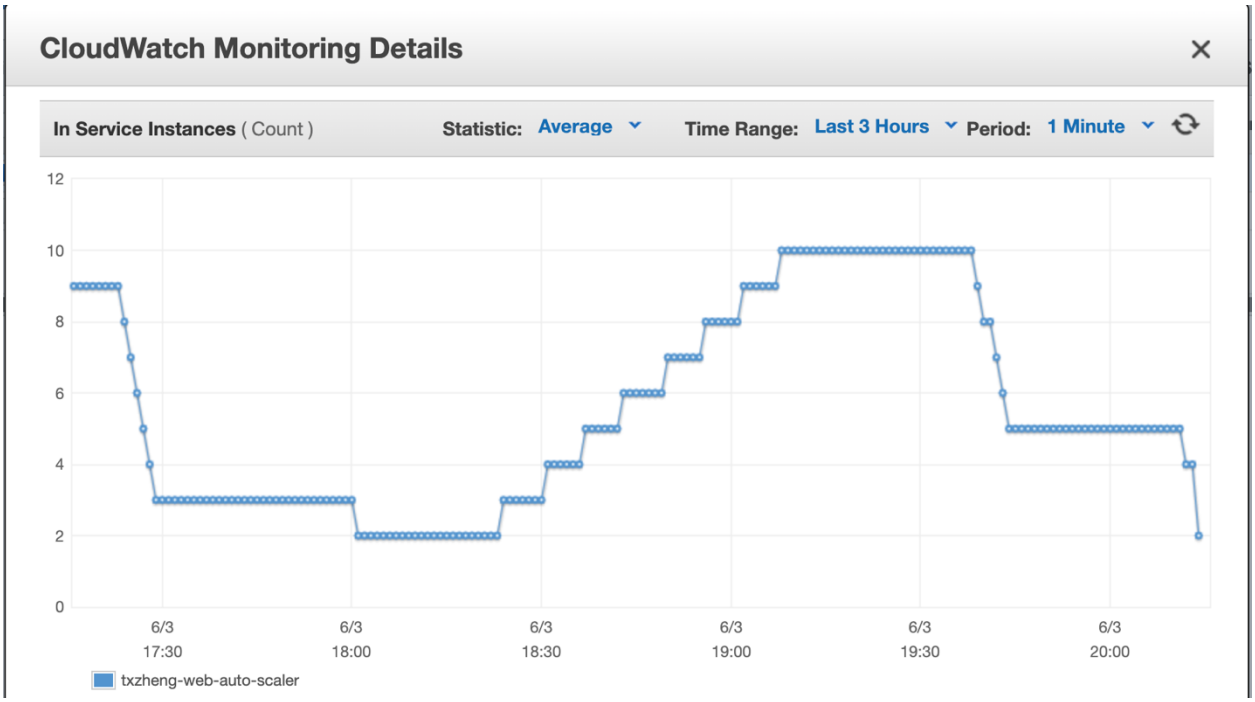


Figure 2

14) I sent messages to the job_request queue at 1/second rate. The scale-out alarm was triggered after 10 minutes and number of instances started to grow slowly. Since I set the "And then wait" parameter to 300 seconds, the number of instances grew slowly at 5 minutes per new instance, as shown in figure 3. After it reached 10 instances, I stopped sending messages to the queue and purged the queue. After around 10 minutes, the scale-in alarm was triggered, and the number of instances was decreasing quickly. As I set the "And then wait" parameter to 0, the number of instances was decreasing at 1 per 1 ~ 3 minutes, which was the time it took to terminate an instance. It took 11 minutes in total for the auto scaling group to return to 2 instances.

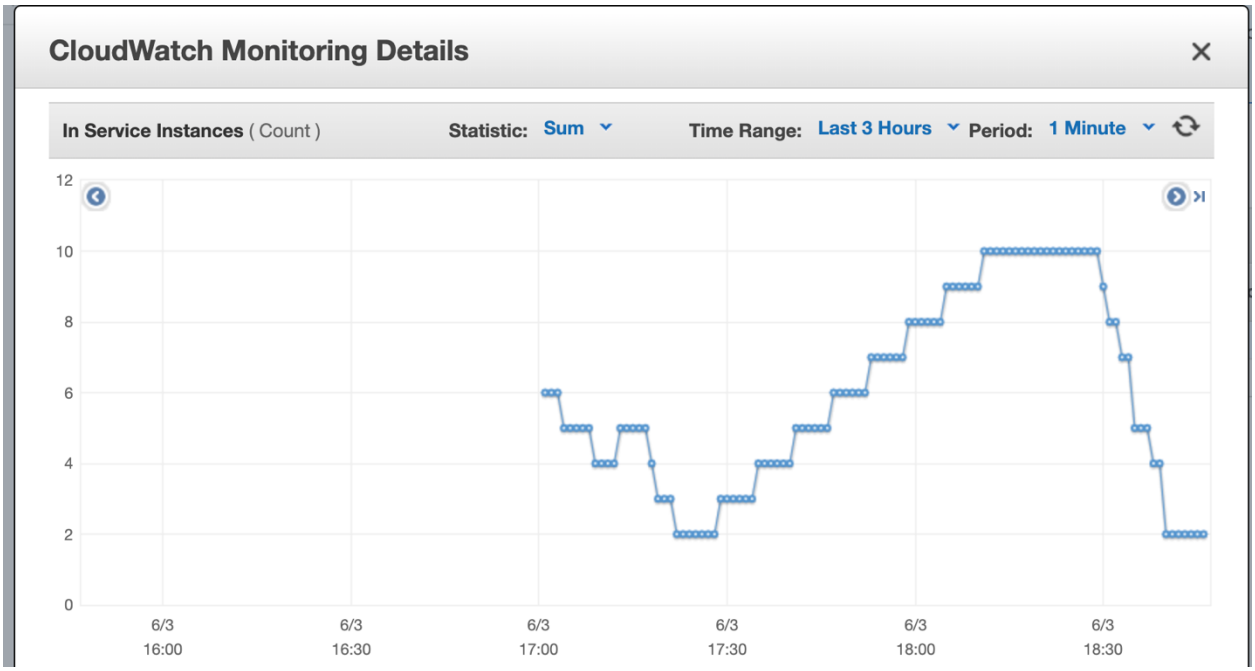

CloudWatch Monitoring Details
In Service Instances (Count)
Statistic: Sum
Time Range: Last 3 Hours
Period: 1 Minute
12
10
8
6
4
2
0
6/3
16:00
6/3
16:30
6/3
17:00
6/3
17:30
6/3
18:00
6/3
18:30

Figure 3