Chatrin Yoonchalard 6631304721

1. Understanding overload

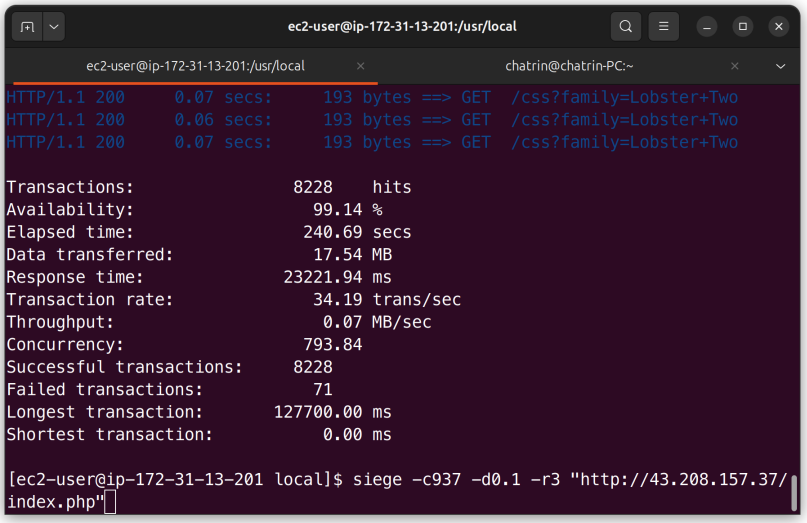
1.1. What option did you use to run siege in order to trigger overload on your web servers and get HTTP status errors or the server becomes unresponsive.

Ans: I use `siege -c937 -d0.1 -r3 "http://43.208.157.37/index.php"`

This command runs with 937 clients. This was obtained by binary search between 120 to 1024, the normal siege configuration limit client number at 255 so I edited the siege configuration file and set the limit up to 1024.

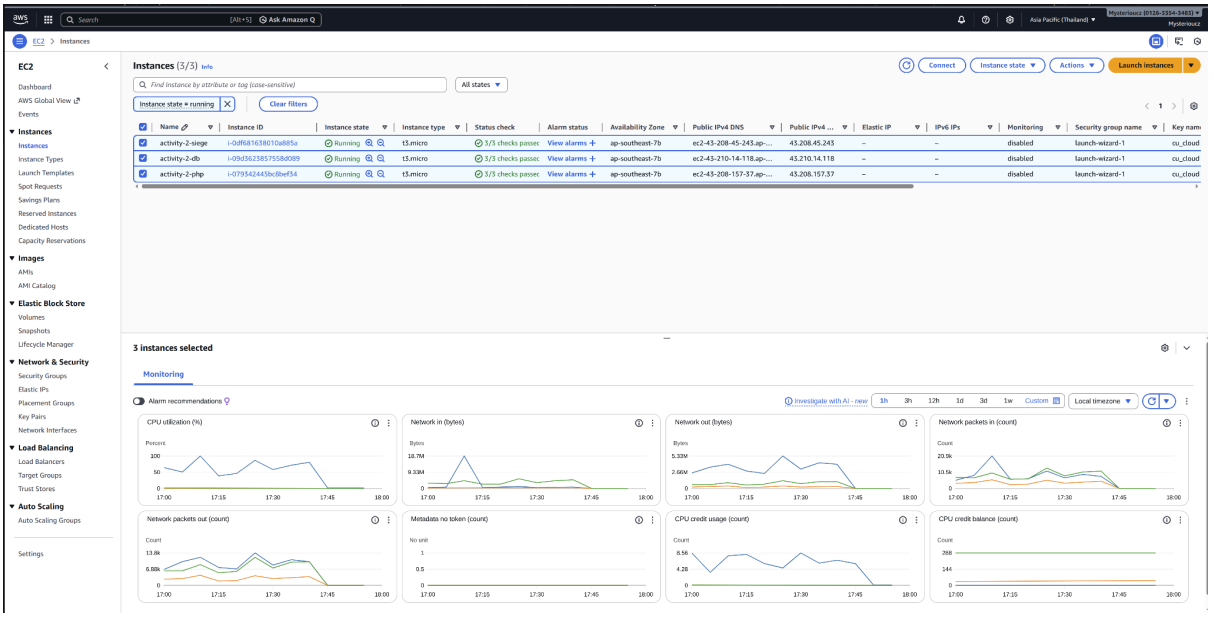
1.2. For the experiment using the siege options in 1.1, what is the average response time, transaction rate, and concurrency? How many of your transactions were successful vs. failed?

Ans: The average response time is 23.221 seconds, transaction rate is 34.19 trans/sec, and concurrency is 793.84. The successful transaction is 8228 transactions and 71 failed transactions.



2. Take screenshots from the monitoring panel for your 3 instances to show load changes and point out when you start to HTTP status errors (not HTTP 200 OK) or when the server becomes unresponsive.

Ans: The server become unresponsive at the time around 17.10



3. Do you think Amazon is providing sufficient resources for this instance size, price point, and number of clients you would generally have for a web application?

Ans: In my opinion at this price it's very sufficient. During the load test the response time from the server is increasing with the number of siege's clients (more than 3 seconds when tested with 120 clients) at this price it can handle around 120 clients with 3 seconds response time is very fair.

4. Where is the bottleneck in this system: your client, your web server, or your database server? Why do you think so? Provide evidence.

Ans: The bottleneck is at the web server we can clearly see in the CPU Utilization graph the Web server instance reaches 100%. This is because the index.php contains matrix multiplication code and with 937 clients it overwhelms the instance's CPU..