Performance test report - Mar 12, 2024 (#10)

Open in Postman

Postman collection: NBA GraphQL

Report exported on: Mar 12, 2024, 14:56:46 (CDT)

Test setup

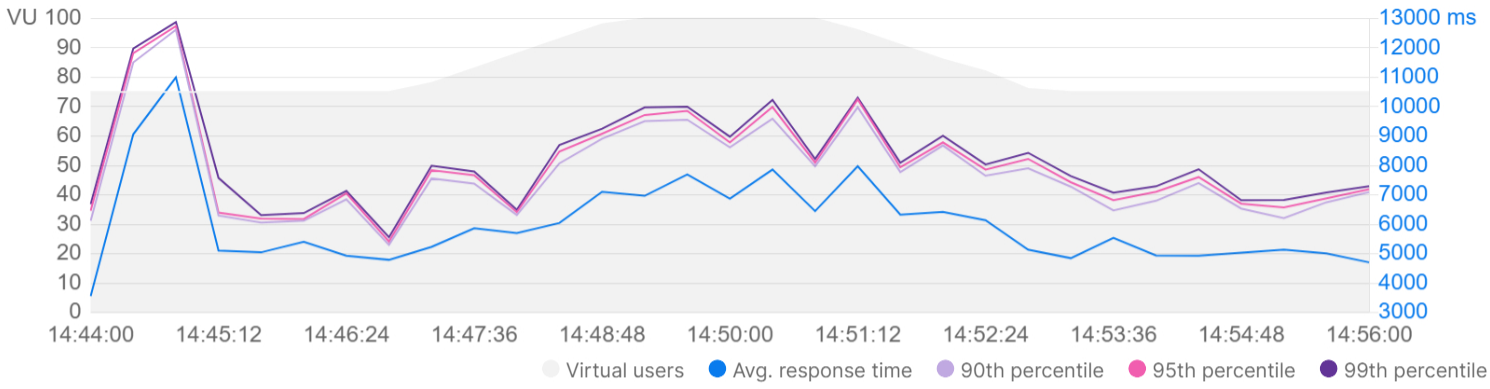
Virtual users	Start time	Load profile
100 VU	Mar 12, 14:44:08 (CDT)	Peak
Duration	**End time**	**Environment**
12 minutes	Mar 12, 14:56:15 (CDT)	GraphQL

1. Summary

Total requests sent	Throughput	Average response time	Error rate
9,504	13.07 requests/second	5,938 ms	0.00 %

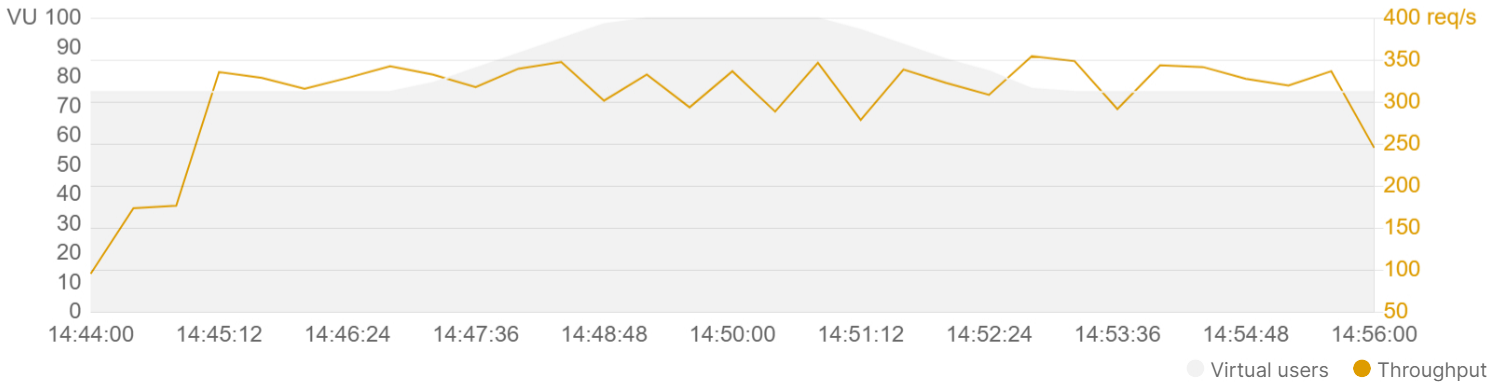
1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.



1.3 Requests with slowest response times

Top 5 slowest requests based on their average response times.

Request	Resp. time (Avg ms)	90th (ms)	95th (ms)	99th (ms)	Min (ms)	Max (ms)
GET LeBron Advanced + Total {{url}}/graphql/	6,622	9,055	10,280	11,504	2,952	12,779
GET Kobe Advanced + Total {{url}}/graphql/	6,472	8,960	9,970	12,553	3,044	12,963
GET Harden Advanced + Total Copy {{url}}/graphql/	6,366	8,484	9,314	11,595	3,100	11,807
GET Iverson Advanced + Total {{url}}/graphql/	5,707	8,068	9,097	9,915	3,202	10,361
GET Advanced LeBron {{url}}/graphql/	5,304	7,727	8,381	9,456	455	10,157

2. Metrics for each request

The requests are shown in the order they were sent by virtual users.

Request	Total requests	Requests/s	Min (ms)	Avg (ms)	90th (ms)	Max (ms)	Error %
GET Advanced LeBron {{url}}/graphql/	1,643	2.26	455	5,304	7,727	10,157	0
GET Totals Harden {{url}}/graphql/	1,600	2.20	2,215	5,205	7,388	10,245	0
GET Iverson Advanced + Total {{url}}/graphql/	1,572	2.16	3,202	5,707	8,068	10,361	0
GET Kobe Advanced + Total {{url}}/graphql/	1,566	2.15	3,044	6,472	8,960	12,963	0
GET LeBron Advanced + Total {{url}}/graphql/	1,564	2.15	2,952	6,622	9,055	12,779	0
GET Harden Advanced + Total Copy {{url}}/graphql/	1,559	2.14	3,100	6,366	8,484	11,807	0

3. Errors

This run has no errors

All requests were sent successfully and returned a 2xx response code.

Testing API performance on Postman

Postman enables you to simulate user traffic and observe how your API behaves under load. It also helps you identify any issues or bottlenecks that affect performance.

Learn more about testing API performance.