CYREX

Load Testing Report

Prepared for: **Roby Weir, COO, Jigstack**
Prepared by: **Tim De Wachter, CTO**, *Cyrex Ltd*

09/08/2021

Table of Contents

Executive summary

Cyrex was contracted by JigStack to conduct a load test to determine the performance of the Gallery web platform on a large scale. Specifically, this test validates whether the application is ready to handle a certain volume of users. Cyrex developed load testing scripts that simulate real user behaviour within different parts of the platform:

- Login with users
- User management
    - List users
    - Show single users
- Artist management
    - Create artists
    - List artists
    - Show single artists
- NFT management
    - Create NFTs
    - Update NFT status
    - List NFTs
    - Show single NFT

Afterwards, Cyrex conducted several iterations of load testing on JigStack's infrastructure. Lastly a live demo was performed in order to demonstrate the results.

What follows is a detailed explanation of all the different tests, discovered problems and potential remediations. Most of the logs are not visible to Cyrex, thus making it hard to find the root cause of a certain bottleneck when scaling the volume of users. Nevertheless, we are eager to perform another iteration of testing in case a certain issue cannot be identified.

We are confident that the load test and this report helps the customer to raise the platform's availability and scalability to a higher level.

Conclusion

During the various iterations in the load test, different scaling issues were identified on the AWS configuration, but the vast majority of them have been resolved. There is one remaining issue, namely a 502 error that is encountered on a very limited amount of requests.

In this way it is safe to say that the platform is able to handle a load of 1000 active, concurrent users with a very small, neglectable portion of errors and thus ready for launch.

We want to thank JigStack for putting trust in our knowhow and expertise concerning load testing.

Load Test

This section lists all the load tests that were conducted by Cyrex on JigStack's infrastructure. Any issues that arose during the load test are described and whenever applicable, charts and statistics are provided.

Whenever something in the report is unclear, Cyrex is eager to help and supply additional details. Feel free to contact us at any time.

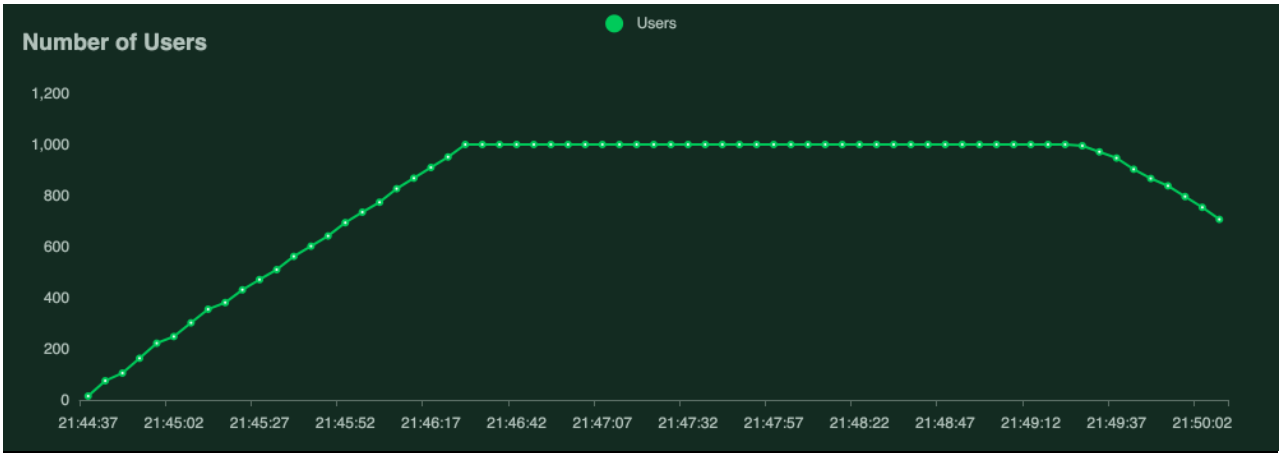
User management

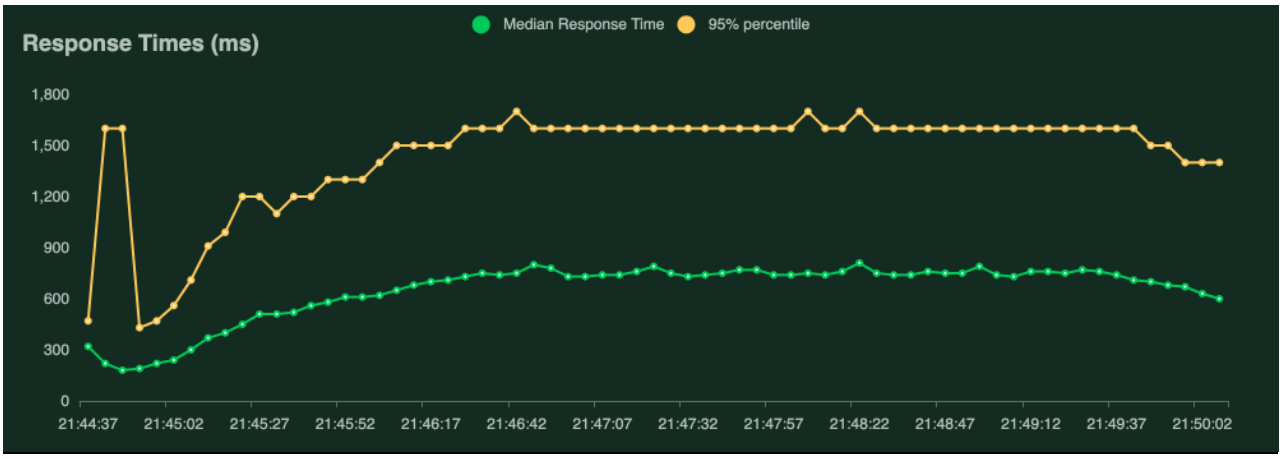
During this test, the bot farm acts as regular users would act, performing various actions like listing users, as well as requesting details of individual users.

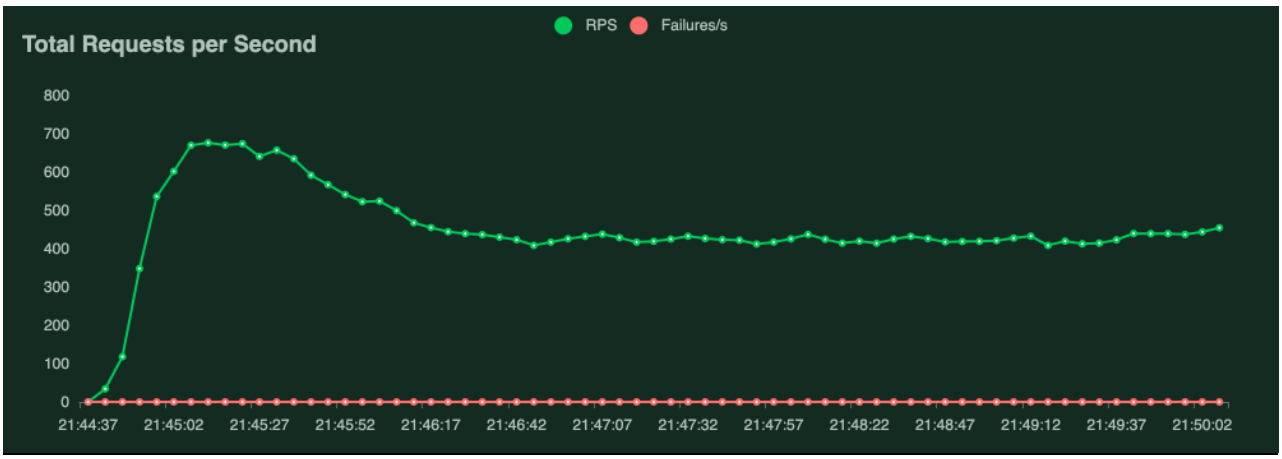
Interpreted results

No failures were encountered during this test. Response times remain between half a second up till 2 seconds. Which is very reasonable taking into account the load of 1000 concurrent users. The login request and paginated user list are definitely the most intensive ones.

Results of test







Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/login	4951	0	2021	1566	2444	363	14.6	0.0
GET	/requestNonce/:address	4971	0	576	78	1812	76	14.7	0.0
GET	/users/profile/:address	47438	0	594	70	2357	108	140.0	0.0
GET	/users/show/:address	47501	0	640	88	1574	255	140.2	0.0
GET	/users?skip=N	48856	0	1127	100	2278	8602	144.2	0.0
	Aggregated	**153717**	**0**	**823**	**70**	**2444**	**2861**	**453.6**	**0.0**

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/login	2100	2100	2100	2100	2300	2300	2400	2400
GET	/requestNonce/:address	610	630	650	680	840	880	1100	1800
GET	/users/profile/:address	630	660	680	710	840	900	950	2400
GET	/users/show/:address	680	700	720	760	880	930	990	1600
GET	/users?skip=N	1300	1300	1400	1500	1600	1600	1700	2300
	Aggregated	**700**	**750**	**900**	**1300**	**1400**	**1600**	**2100**	**2400**

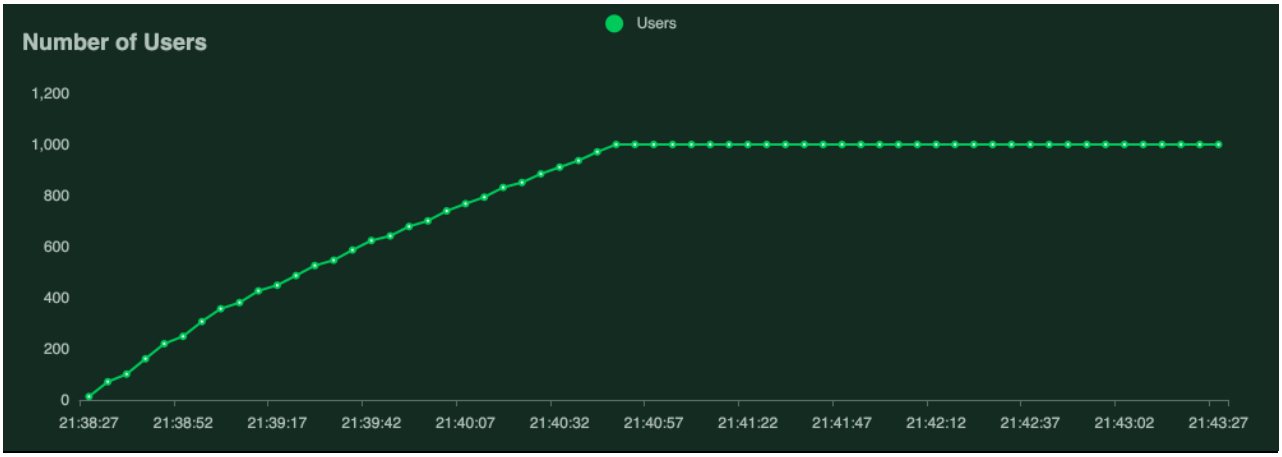
Login test

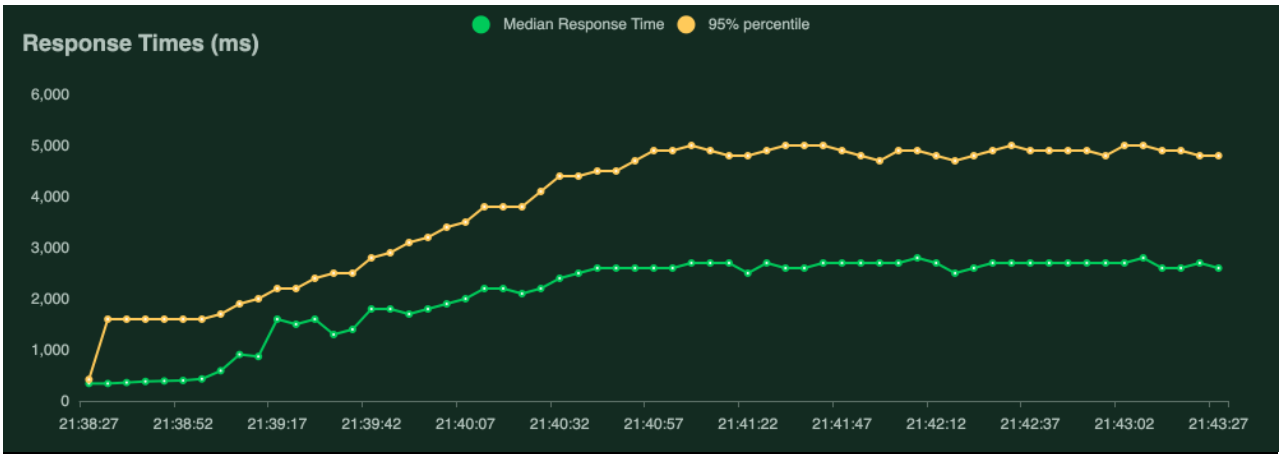
During this test, the bot farm acts similar to a lot of users logging concurrently as if several thousands of users are logging in to the application.

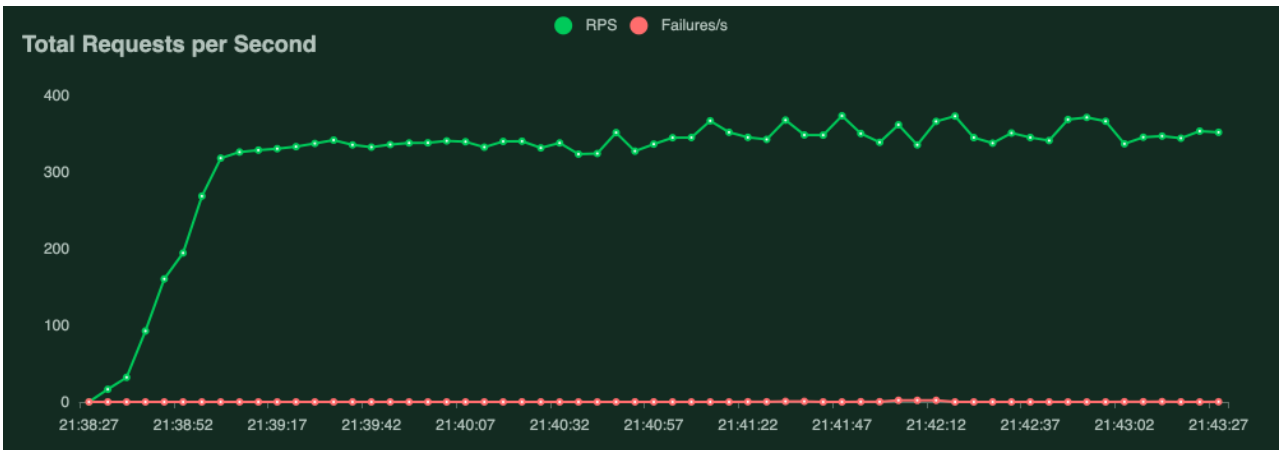
Interpreted results

During the test a very small portion of 502 errors were encountered. In total, 40 requests failed during this test. This can be neglected in comparison with the several thousand requests that have been performed but might be an area of attention whenever the application becomes very popular. Average response times for the login request are around 3 seconds, which is still very decent concerning the load.

Results of test







Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/login	49773	20	3247	443	5529	363	162.2	0.1
GET	/requestNonce/:address	50063	20	1381	74	9008	76	163.2	0.1
	Aggregated	**99836**	**40**	**2312**	**74**	**9008**	**219**	**325.4**	**0.1**

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/login	3200	3600	3900	4300	4700	4900	5200	5500
GET	/requestNonce/:address	1300	1600	1900	2200	2600	2800	3300	9000
	Aggregated	**2200**	**2600**	**3000**	**3600**	**4300**	**4700**	**5100**	**9000**

Failure statistics

Method	Name	Status code	Error	Occurrences
GET	/requestNonce/:address	502	<title>502 Bad Gateway</title> 502 Bad Gateway	20
POST	/login	502	<title>502 Bad Gateway</title> 502 Bad Gateway	20

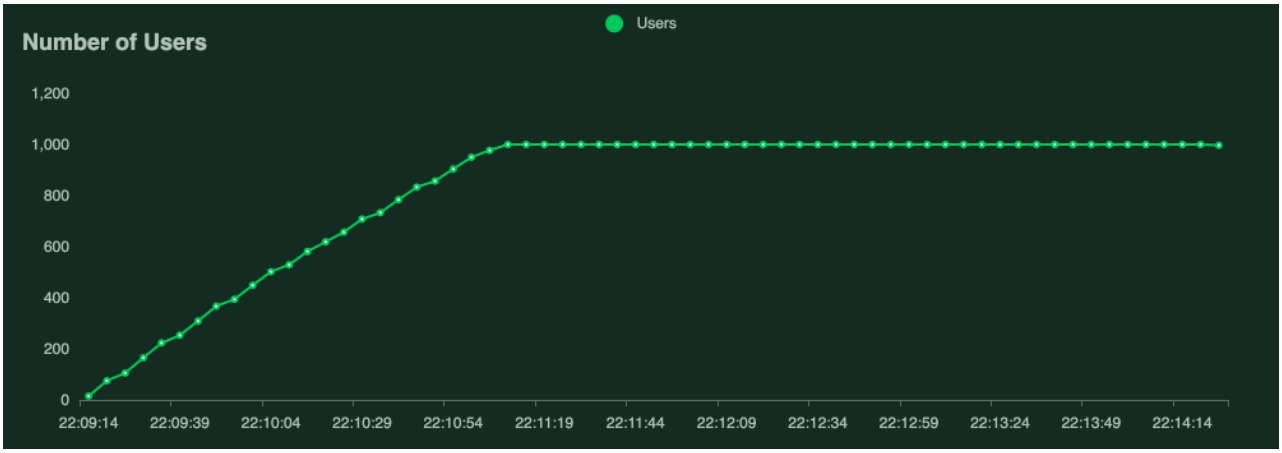
Artist management

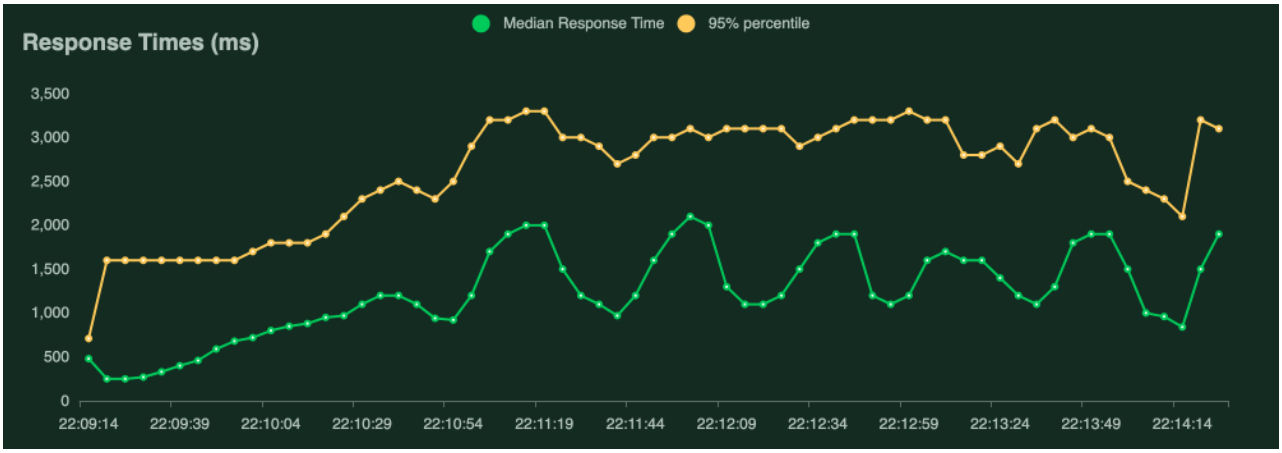
During this test, the bot farm acts as regular users would act, performing various actions like creating and listing artists, as well as requesting details of individual artists.

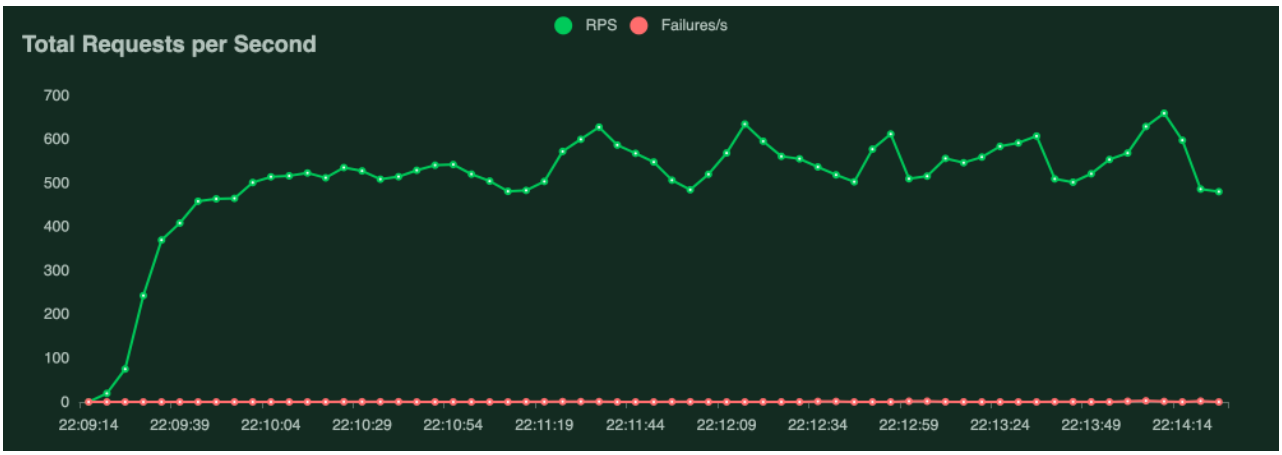
Interpreted results

During the test a very small portion of 502 errors were encountered were encountered on various endpoints. In total, 101 requests failed during the test. This can be neglected in comparison with the several thousand requests that have been performed but might be an area of attention whenever the application becomes very popular. Response times remain between half a second up till 2 seconds. Which is very reasonable taking into account the load of 1000 concurrent users. The login request and paginated artist list are definitely the most intensive ones.

Results of test







Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/artists/new	7798	1	1289	243	5632	349	24.4	0.0
GET	/artists?skip=N	72596	50	1940	106	5843	8212	227.4	0.2
POST	/login	7761	4	2294	226	4055	363	24.3	0.0
GET	/requestNonce/:address	7798	17	855	76	3086	76	24.4	0.1
GET	artists/show/:address	67669	29	797	71	2583	541	212.0	0.1
	Aggregated	**163622**	**101**	**1401**	**71**	**5843**	**3905**	**512.5**	**0.3**

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/artists/new	1200	1300	1500	1700	2100	2400	3100	5600
GET	/artists?skip=N	2000	2200	2300	2600	2900	3200	3700	5800
POST	/login	2200	2400	2500	2700	2900	3100	3500	4100
GET	/requestNonce/:address	730	900	1100	1300	1600	1900	2500	3100
GET	artists/show/:address	770	860	980	1100	1300	1400	1800	2600
	Aggregated	**1200**	**1500**	**1900**	**2200**	**2600**	**2900**	**3500**	**5800**

Failure statistics

Method	Name	Status code	Response	Occurrences
POST	/artists/new	502	<title>502 Bad Gateway</title> 502 Bad Gateway	1
GET	/artists?skip=N	502	<title>502 Bad Gateway</title> 502 Bad Gateway	50
GET	/requestNonce/:address	502	<title>502 Bad Gateway</title> 502 Bad Gateway	17
GET	artists/show/:address	502	<title>502 Bad Gateway</title> 502 Bad Gateway	29
POST	/login	502	<title>502 Bad Gateway</title> 502 Bad Gateway	4

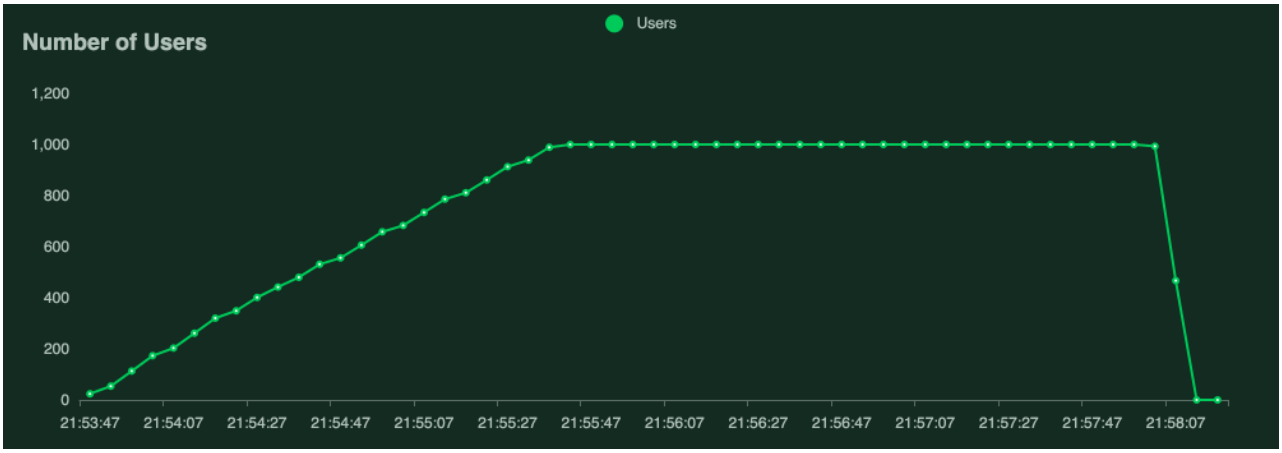
NFT management

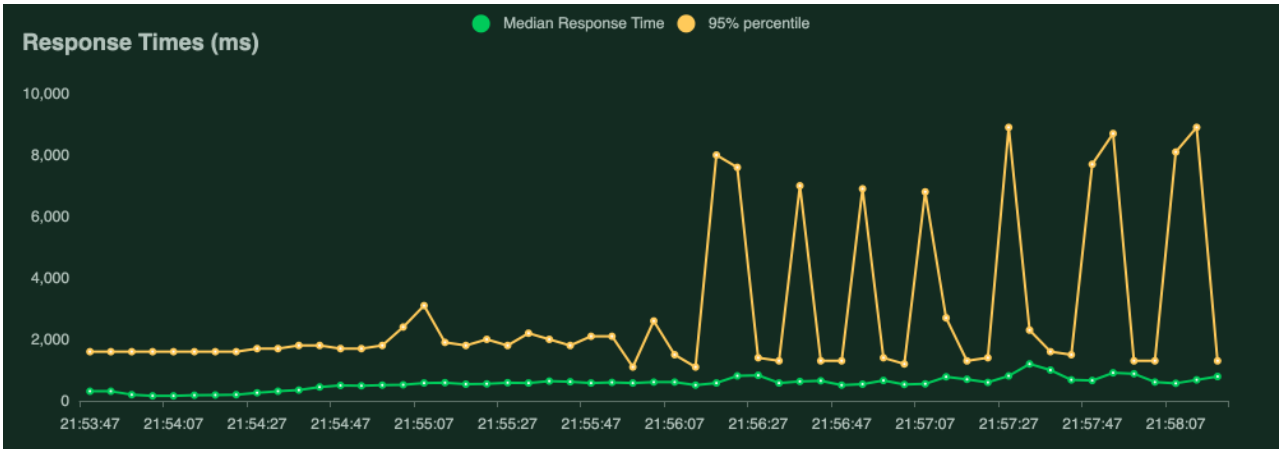
During this test, the bot farm acts as regular users would act, performing various actions like creating and listing of NFTs, as well as requesting details of individual nfts.

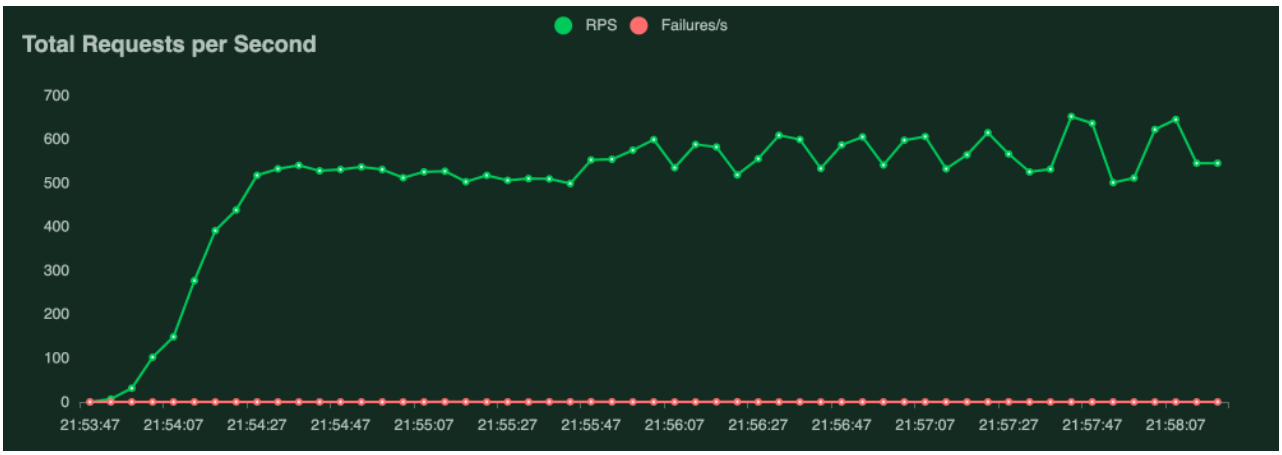
Interpreted results

During the test a very small portion of 502 errors were encountered on various endpoints. In total, 13 requests failed during the test. This can be neglected in comparison with the several thousand requests that have been performed but might be an area of attention whenever the application becomes very popular. Response times remain between half a second and 2 seconds. Which is very reasonable considering a load of 1000 concurrent users. Apart from the creation of NFTs, which can easily take up till 10 seconds. But we need to take into account that users are generally aware that whenever they upload something, the request might take a while.

Results of test







Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/login	1000	0	1721	1564	2297	363	3.7	0.0
POST	/nfts/new	6287	2	8343	250	16798	819	23.5	0.0
GET	/nfts/show/:id	58883	6	435	67	1410	930	220.0	0.0
GET	/nfts?skip=N	60897	4	799	74	3067	14749	227.5	0.0
GET	/requestNonce/:address	7758	1	478	75	2471	76	29.0	0.0
	Aggregated	**134825**	**13**	**980**	**67**	**16798**	**7113**	**503.7**	**0.0**

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/login	1700	1700	1800	1800	1900	2000	2200	2300
POST	/nfts/new	8200	9300	11000	12000	13000	14000	15000	17000
GET	/nfts/show/:id	410	450	500	580	690	780	930	1400
GET	/nfts?skip=N	780	860	950	1100	1200	1400	1600	3100
GET	/requestNonce/:address	410	460	530	620	800	1100	1800	2500
	Aggregated	**580**	**680**	**790**	**940**	**1200**	**1800**	**11000**	**17000**

Failure statistics

Method	Name	Status code	Response	Occurrences
POST	/nfts/new	502	<title>502 Bad Gateway</title> 502 Bad Gateway	2
GET	/requestNonce/:address	502	<title>502 Bad Gateway</title> 502 Bad Gateway	1
GET	/nfts/show/:id	502	<title>502 Bad Gateway</title> 502 Bad Gateway	6
GET	/nfts?skip=N	502	<title>502 Bad Gateway</title> 502 Bad Gateway	4