

2.8 - Results Analysis

Results Analysis — Appvance IQ documentation.

- [Results Analysis](#)
- [Determine the Right Test Duration](#)
- [Find and Resolve Test Failures](#)
- [Evaluate Linear Scalability](#)
- [About Reports](#)
- [Review Live Results While a Test Runs](#)
- [Determine if the Test System is a Performance Bottleneck](#)
- [View Results from a Previous Test Run](#)
- [What Can Go Wrong During Results Analysis](#)

Results Analysis

Contents

- [Determining the Length of Tests](#)
- [Finding and Solving the Failures](#)
- [Evaluate Applications Ability to Achieve Linear Scalability](#)
- [About Reports](#)
- [Live Results While a Test Runs](#)
- [Determine if the Test System is a Performance Bottleneck](#)
- [Viewing Results from a Previous Test Run](#)
- [What Can Go Wrong](#)

Results Analysis and Reporting

Unified Test Platform Results Analysis Engine presents test results in several reports. This contains a set of charts showing you just where the app failed, the number of transactions, CSV Reports. This chapter shows how to determine the root cause of functional issues, performance problems, and app service delivery problems. Unified Tess Platform accomplishes this by storing the steps, times, pass / fail results, the reason for failures from every step of every test. Appvance IQ Monitor stores the CPU, memory, network, and app-specific observations from the servers running the web server, application server, database server, and load balancer.

Below are the topics that are covered:

- Determining How Long Do Test Cases Take
- Run time features
- Finding and solving the failures
- Evaluate App's Ability to Achieve Linear Scalability
- About different Reports
- Live Results

- Determine If Test system is a Performance Bottleneck
 - Viewing results from previous test runs
 - What can go wrong
-

Last updated: March 23, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Determine the Right Test Duration

Determine the Right Test Duration

Choose test duration based on the performance question being answered. A short smoke run can validate a script, while a longer run may be needed to observe steady-state behavior, resource growth, or reliability under sustained load.

Common duration patterns

- Smoke validation: short run to confirm the scenario and data work.
- Baseline run: enough time to reach steady state and capture representative metrics.
- Endurance run: longer execution used to observe stability and resource trends.

Before interpreting results

1. Exclude ramp-up and ramp-down periods when analyzing steady-state performance.
2. Confirm the run had enough valid samples for the metric being reviewed.
3. Document environment, load model, data, and duration with the result.

Last updated: September 18, 2026

Find and Resolve Test Failures

Find and Resolve Test Failures

Start failure analysis with the first meaningful failure in the execution. Later errors often come from the same root cause and can distract from the actual problem.

Failure triage

- Scenario or locator issue: the application changed or the step targets the wrong element.
- Test data issue: required data is missing, reused, expired, or invalid.
- Environment issue: the application, network, browser, or Test Node is unstable.
- Application defect: the system under test behaves incorrectly under the tested condition.

Recommended workflow

1. Open the failed execution and identify the first failed step or request.
2. Review output, screenshots, logs, and consumed test data.
3. Compare with a known good execution when available.
4. Fix the scenario only after confirming the issue is test-maintenance related.
5. Re-run a focused validation before marking the issue resolved.

Last updated: September 18, 2026

Evaluate Linear Scalability

Evaluate Linear Scalability

Linear scalability compares how the application responds as load increases. A scalable system should increase throughput predictably while keeping response times and errors within acceptable limits.

What to compare

- Virtual users or request volume at each load level.
- Throughput and transaction completion rate.
- Response time trends, including percentile behavior.
- Error rate and application or infrastructure saturation signals.

Recommended workflow

1. Run controlled tests at increasing load levels.
2. Keep test data, environment, and scenario mix consistent between runs.
3. Compare steady-state periods instead of ramp-up or unstable intervals.
4. Identify the load level where throughput stops increasing or errors rise.

Last updated: September 18, 2026

About Reports

List of Available Reports

Below you will find the list of all the Unified Test Platform Reports. The initial report screen shows eleven reports as shown below.

The screenshot displays the 'appvance IQ' interface. On the left is a sidebar with navigation options: Dashboard, Record, Build, AI Driven, Play, Analyze (highlighted), Global Options, and Tutorials / References. The main area shows a report for 'New-scenario1.scenario' dated 'Fri Jun 01 12:00:14 IST 2018'. It includes a 'Functional Report' dropdown and a 'Download JUnit XML' link. Below these are several filters: 'Iteration1', 'Testcase1', '1: 0s Success', and 'Step'. The report content is a table with the following data:

Offset	26
Testcase	Testcase1
Duration	30351ms
Output	Processing page:http://demosite.appvance.com/t/brands/rails?utf8=%E2%9C%93&per_page=
Error	
Location	127.0.0.1
Step 1 - 30351 ms	1) Untitled1
Step 2 - 4 ms	1.1) http://demosite.appvance.com/t/brands/rails?utf8=%E2%9C%93&per_page=
Step 3 - 754 ms	1.2) http://demosite.appvance.com/t/brands/rails
Step 4 - 361 ms	1.3) http://demosite.appvance.com/t/brands/spree
Step 5 - 352 ms	1.4) http://demosite.appvance.com/t/brands/apache
Step 6 - 337 ms	1.5) http://demosite.appvance.com/t/brands/ruby
Step 7 - 332 ms	1.6) http://demosite.appvance.com/t/brands/clothing
Step 8 - 346 ms	1.7) http://demosite.appvance.com/t/brands/mugs
Step 9 - 354 ms	1.8) http://demosite.appvance.com/t/brands/bags
Step 10 - 27511 ms	1.9) http://demosite.appvance.com/

In order to understand better and use these reports it is important to understand some basic concepts:

- Execution: This is the complete execution of a scenario, it is possible that it can contain several iterations.
- Iterations: A iteration indicates the number of users who run during a certain amount of time, the below screen-shot shows an execution with 2 iterations.

The screenshot shows the 'appvance IQ' 'Test Management' interface. The left sidebar is the same as the previous screenshot. The main area has tabs for 'Info', 'Test Management' (active), 'Test Cases', 'Locations', 'Monitors', 'Resources', 'Options', 'Assert', 'Store', 'Notifications', and 'Integrations'. The 'Test Management' tab contains two iteration configuration sections. The first section is for 'Iteration1' with 'Total Duration' set to 2 minutes. The second section is for 'Iteration2' with 'Total Execution' set to 1 minute. There is an 'Add Iteration' link at the bottom right of the configuration area.

Test Case: Iteration can have one or more Test Cases. Test Case can be a unit test, a selenium file, or any other supported test type by Unified Test Platform. Test Cases are executed in parallel during an iteration.

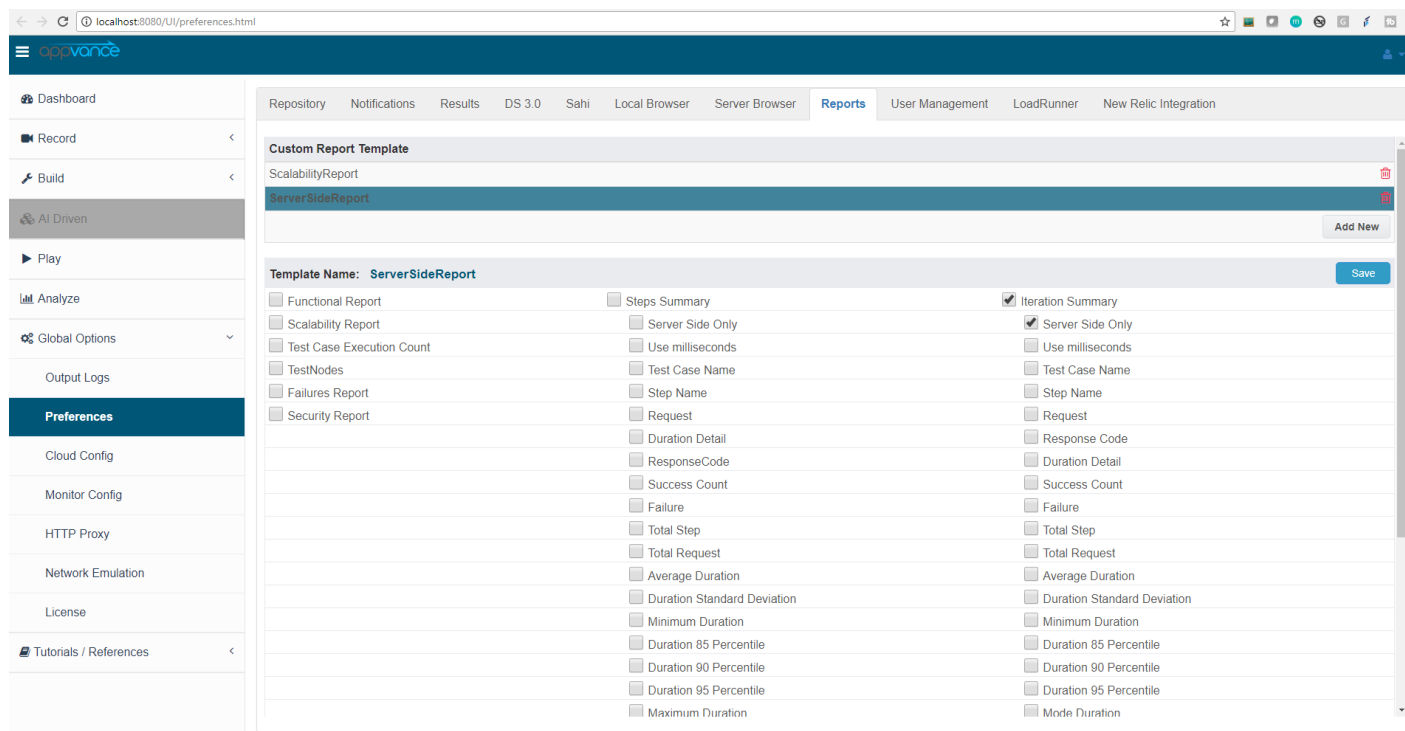
Template Reports

Users can create their own Template Reports and create Reports by using these Templates. See the section on [Report Templates in the Global Options section of the documentation for information on creating report templates.](#)

Server Side Report

“Server-Side Only” is a filter where every “row”(step registry) that has a value = 0 in the column “totalRequest” won’t be added in the report. This Report can be used for filtering the Reports.

With this not only the “set_value” steps are filtered, but every step where “totalRequest” values are zero gets filtered and becomes a Server Side Report.



Functional report

Useful for functional testing. Shows first transaction detailed information

appvance

Dashboard

Record

Build

AI Driven

Play

Analyze

Global Options

Tutorials / References

New-scenario.scenario

Functional Report

Download JUnit XML

Iteration1

Testcase1

1: 0s Success

Request

Duration

5869ms

Output

Processing page:http://demosite.appvance.com/

Error

Location

127.0.0.1

Step 1 - 5869 ms

1) Untitled1

Step 2 - 5869 ms

1.1) http://demosite.appvance.com/

http://demosite.appvance.com/

http://demosite.appvance.com/assets/spree/frontend/all-fe935964ffc1a8877767afcdc98fdeacd3e24552a5986e52ad8f3e1722df9f.js

http://demosite.appvance.com/assets/logo/spree_50-0ae687ccd486b63b01687b1df9e5394430f6fce3ab40c28250b179be4363ddc1.png

http://demosite.appvance.com/spree/products/21/small/lor_tote.jpeg?1499978138

http://demosite.appvance.com/spree/products/23/small/lor_bag.jpeg?1499978140

http://demosite.appvance.com/spree/products/24/small/lor_baseball.jpeg?1499978140

http://demosite.appvance.com/spree/products/26/small/lor_rj_spaghetti.jpeg?1499978142

http://demosite.appvance.com/spree/products/29/small/lor_ringer.jpeg?1499978144

http://demosite.appvance.com/spree/products/34/small/ruby_baseball.png?1499978147

http://demosite.appvance.com/spree/products/33/small/apache_baseball.png?1499978146

http://demosite.appvance.com/spree/products/41/small/spree_jersey.jpeg?1499978152

http://demosite.appvance.com/spree/products/40/small/spree_spaghetti.jpeg?1499978152

http://demosite.appvance.com/spree/products/38/small/spree_ringer_t.jpeg?1499978150

http://demosite.appvance.com/spree/products/36/small/spree_tote_front.jpeg?1499978149

http://demosite.appvance.com/spree/products/35/small/spree_bag.jpeg?1499978148

http://demosite.appvance.com/assets/bootstrap/glyphicons-halflings-regular-fe185d11a49676890d47bb783312a0cda5a44c4039214094e7957b4c040ef11c.woff2

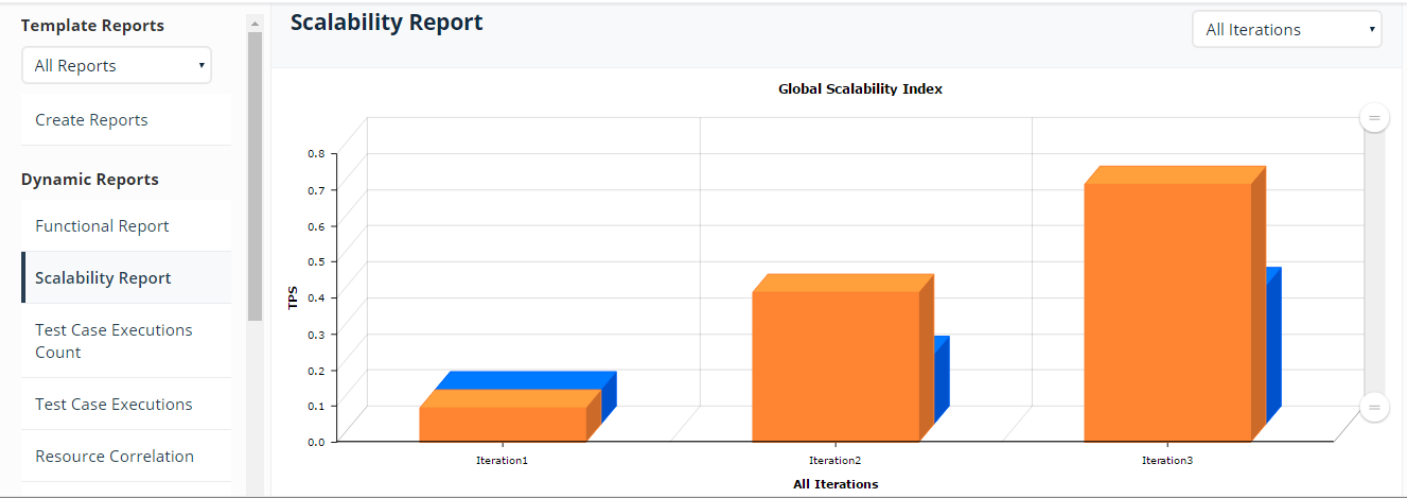
http://demosite.appvance.com/cart_link

Scalability Report

Scalability Report shows how the application scales during the test. Scaling means the more users use the service, the more transactions will be achieved.

Scalability Report is provided in Transactions per second (TPS for short). Good environments should have linear scalability, meaning if X user generates Y transactions, N*X users will generate N*Y transactions.

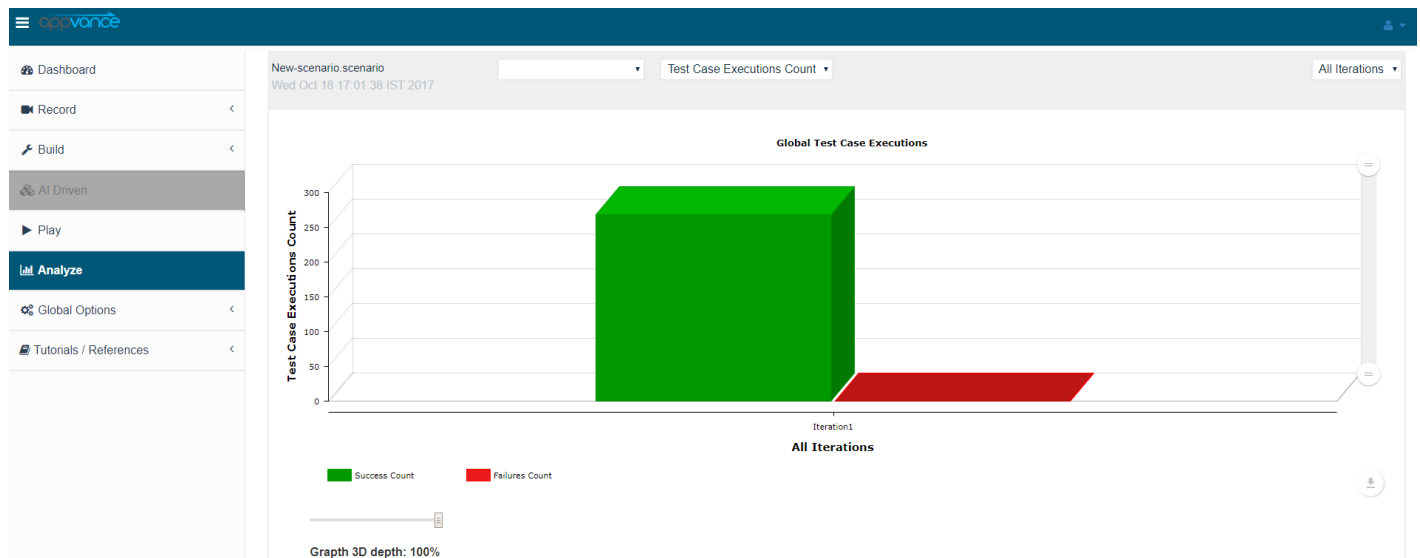
Automatically shows the expected Linear Scalability vs the actual Transactions Per Second achieved.



TestCase Execution Count

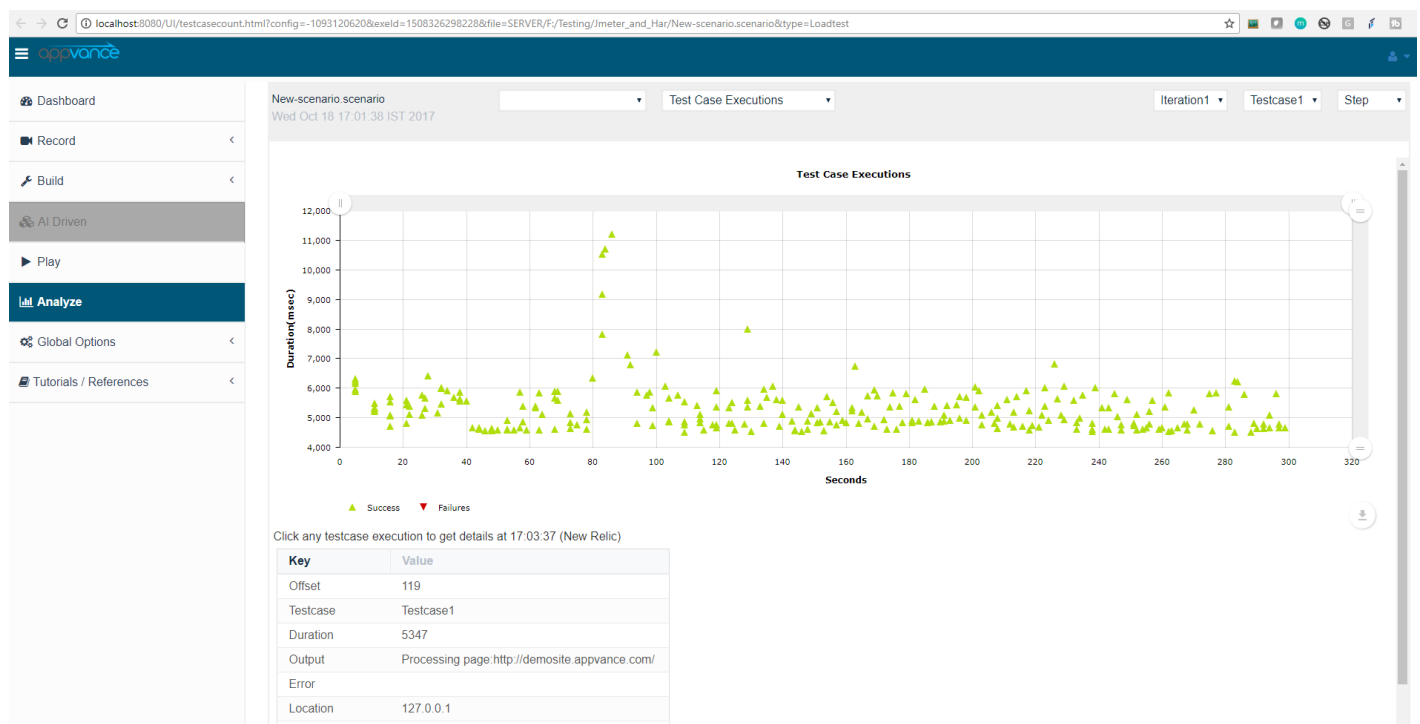
TestCase Execution Count Report shows total successful transactions versus total failures transactions. Each Iteration can be selected from the drop-down.

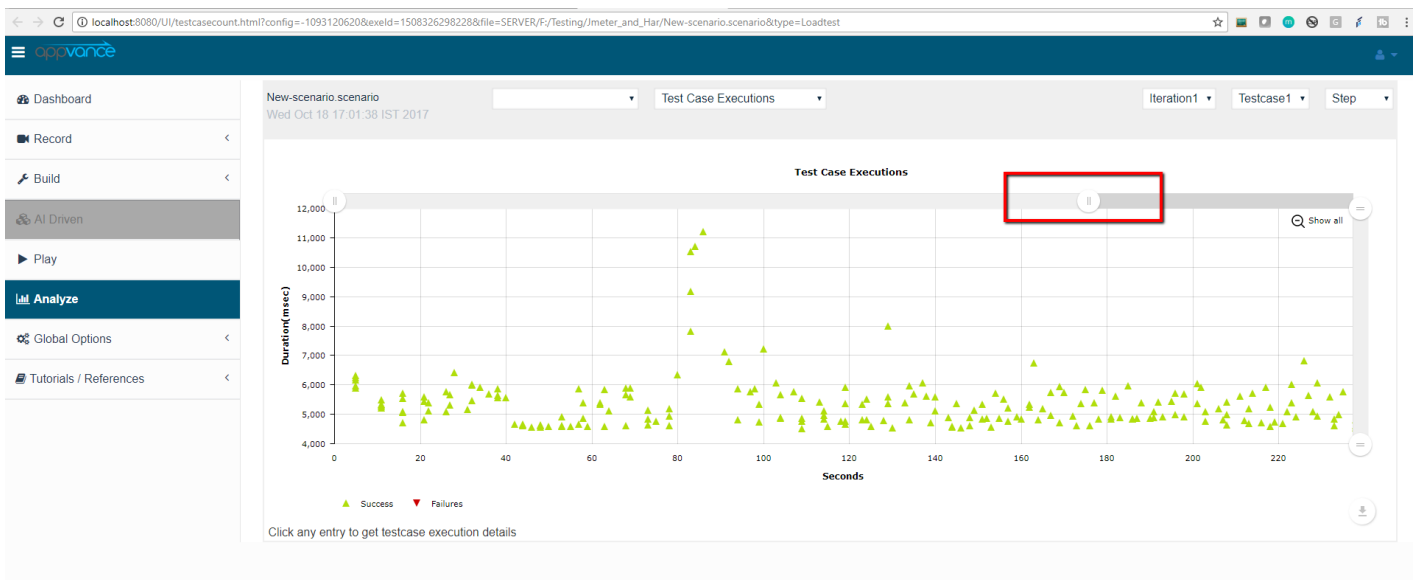
It is useful to quickly see if there are spikes in failures and the overall performance of the service.



TestCase Execution

TestCase Execution Report shows transactions duration every iteration. Iteration needs to be selected from the drop-down as well the Test Case.



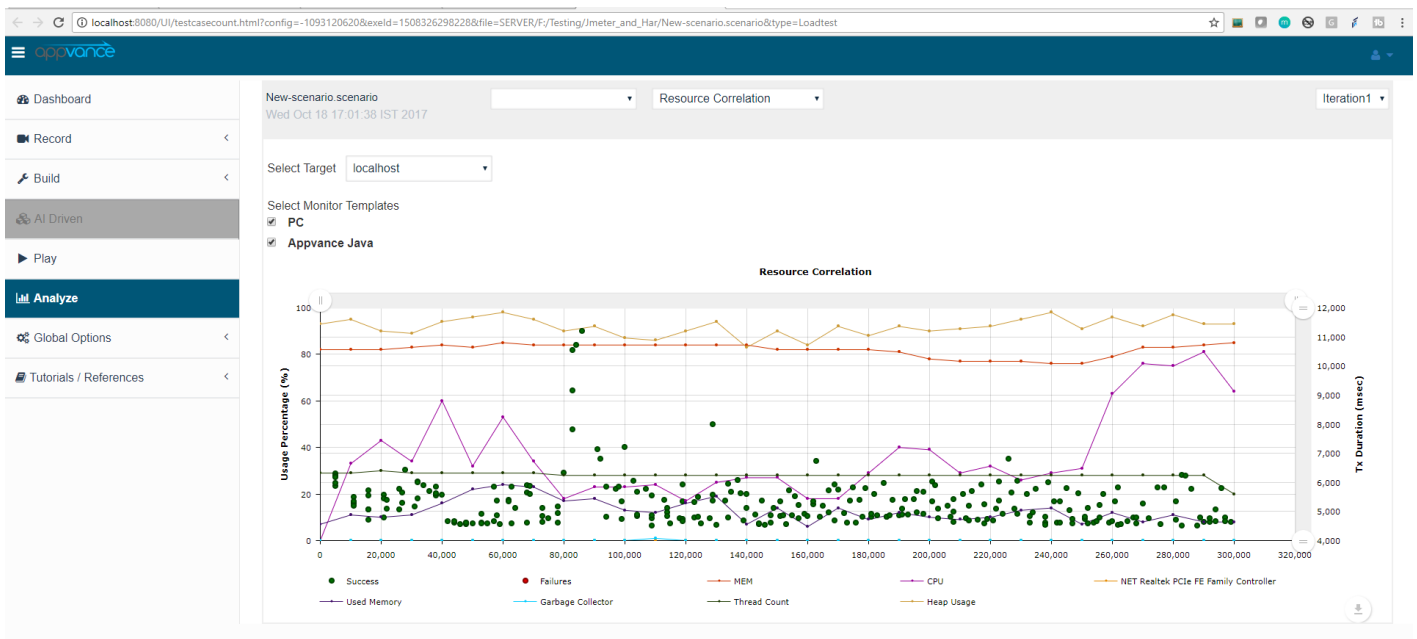


Resource correlation

Correlated successful transactions vs. resource uses. It is useful to pinpoint bottlenecks on a machine. Several machines can be added for resource monitoring. AIQ recommends adding monitors to all the machines that are used during the test. They might include databases, web servers, load balancers, etc.

In order to add machines, a UTPMonitor service should be installed on the desired machine.

Appvance IQ support APCMonitor for Windows, Linux, and Mac Operating Systems.



Resource Usage

Resource usage Report shows machine resource uses during a test. It is useful to pinpoint bottlenecks on a machine. Several machines can be added for resource monitoring. AIQ recommends adding monitors to all the machines that are used during the test. They might include databases, web servers, load balancers, etc.

In order to add machines, an APCMonitor service should be installed on the desired machine.

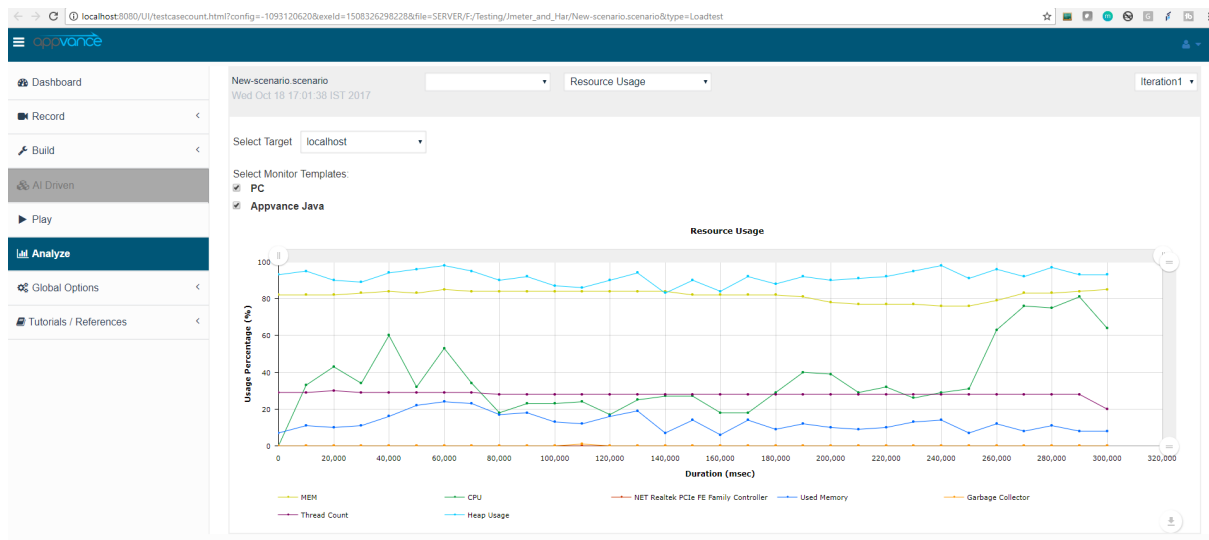
AIQ supports APCMonitor for Windows, Linux, and Mac machines.

JVM METRICS:

1. Heap Usage
 1. This is a JVM metric that shows the used Heap Memory. The heap memory is a run-time data area from which memory for all class instances and arrays are allocated. AIQ by default initializes the heap size limit as 1500 Mb.
2. Garbage Collector
 1. JVM Garbage Collector is a process that automatically handles disposing of memory is no longer used. This metric shows the CPU usage of this process.
3. Thread count
 1. Shows the current number of active threads in JVM

OS METRICS:

1. CPU Usage
 1. CPU usage as reported by the operating system.
2. MEM
 1. Percentage of memory used (against total memory) as reported by the operating system.
3. NET
 1. Net traffic as reported by the operating system.



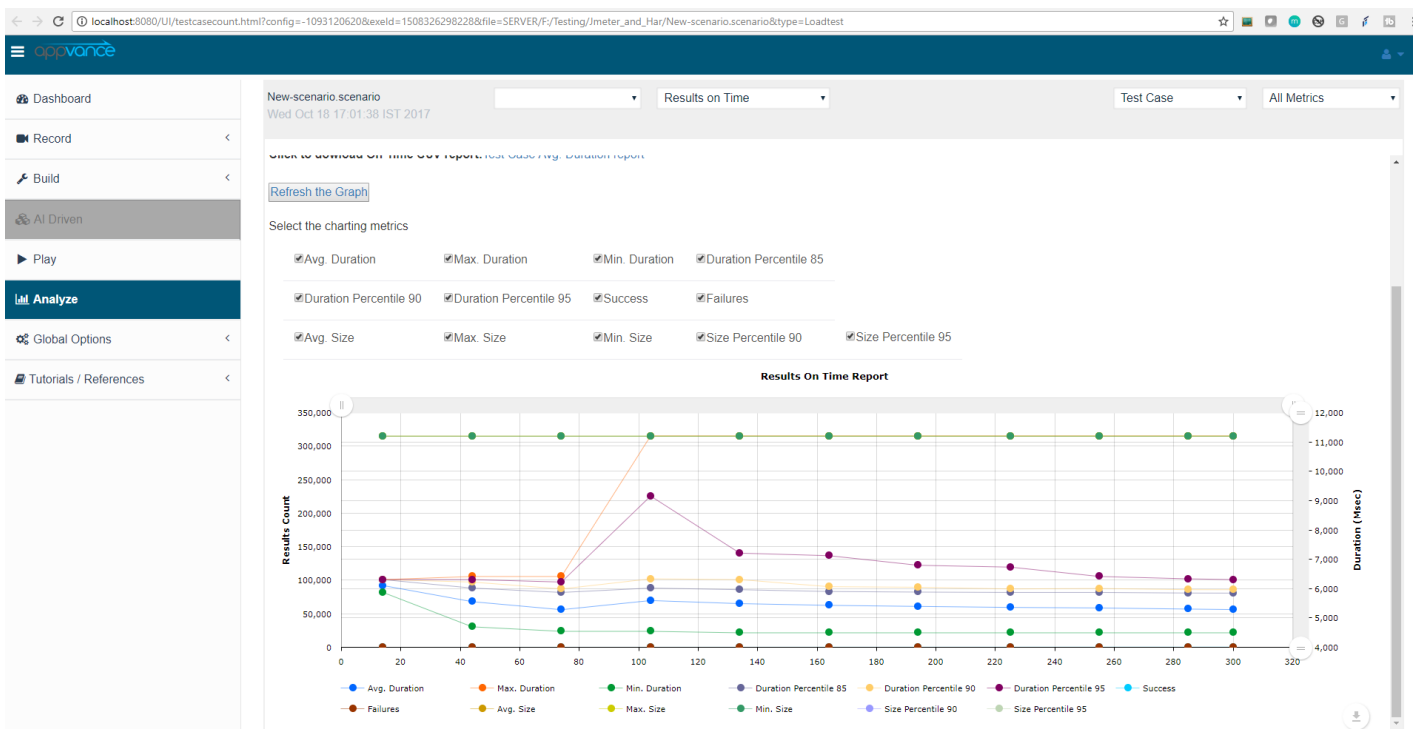
Results on Time

Results on Time graph gives you the option of choosing the "Element type" with "Metrics" that are available in the drop-down. Element type contains Test Case, Step, and Request whereas Metrics drop-down contains Size related, Count related, and duration related. Users can select the combination from either 2 and create a graph.

In addition to the graph, the user can also download CSV reports depending upon the metrics selected in the "Results on Time" Graph. If the user has selected Duration-related metrics the user can download CSV reports of Avg duration, min duration, and so on. So the CSV reports can be downloaded depending upon the metric selected.

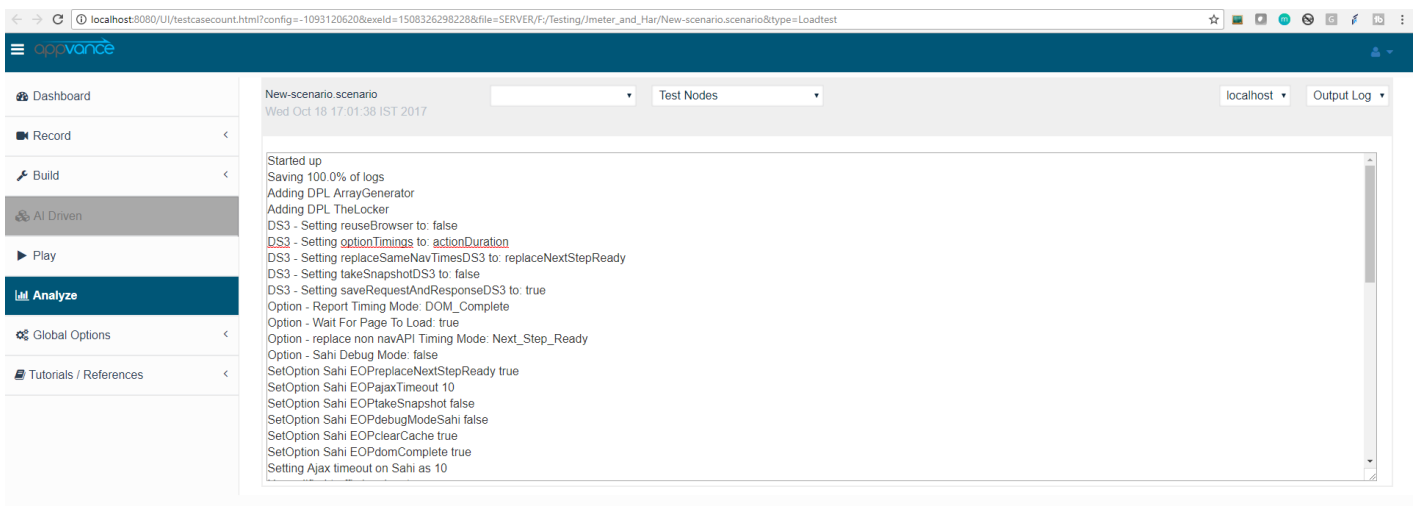
The data points for response times and hits/sec are calculated from the total number of executions for a specific time slice (granularity value). The granularity value is automatically chosen, depending on the duration of the test.

For example, if the chosen granularity value is one minute, then the system will create an average response time value, by averaging the response times of Test Case executions occurring every minute. This graph is useful to monitor the state of each Test Case, while the test is still running.



TestNodes

TestNodes report shows quickly the output and error of the execution TestNodes. It might be useful for debugging purposes.



Step Times

Step Times Report shows transaction's step information per Iteration. The step information showed is the maximum, minimum, average step time. It also includes the duration standard deviation.

Finally, it shows the 85, 90, and 95 duration percentile. This means the maximum duration of 85, 90, and 95% of the steps. It is useful to set limits and confirm them. E.g. 90% of login steps should take less than 10 seconds.



Step Times CSV Report

Unified Test Platform also provides a CSV report of the step times. It can be opened in almost any spreadsheet editor.

Note: In order to capture total throughput in the CSV Report, Error logging should be enabled at the Options panel.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Duration	3 m 6 s																			
Start Time	Sat Jul 30 07:53:52 UTC 2016																			
End Time	Sat Jul 30 07:57:13 UTC 2016																			
Total User	60																			
Total Hits	2380																			
Total Thro	200071.46 KB																			
Average H	12.8 hits/s																			
Average T	1075.65 KB/s																			
Consumer	8811.75 kbps																			
Testcases	60																			
Testcases	0																			
Testcases	0																			
Step pass	360																			
Step fail	0																			
Step error	0																			
All durations and sizes are for successes only																				
Testcase	Total User	User Perc	Active Use	Step nam	Request n	nDuration	cResponse	Success C	Failure Co	Total Step	Total Req	Average C	Standard I	Minimum	Median Di	Mode Dur	Duration	Duration	Duration	Maxim
Testcase1	60		60	1) Untitled1				60	0	360	2380	11828	290469	9089	13145	11828	13145	13499	15674	197
				1.1) navigateTo('http://www.apple.com')				60	0	60	2380	11828	290469	9089	13145	11828	13145	13499	15674	197
				myriad-set-pro_semibold.woff				60	0		60	19	0	17	20	18	20	23	27	
								200	0		0	19	0	17	20	18	20	23	27	
				image_large.svg				600	0		600	21	14961	17	34	18	34	35	35	

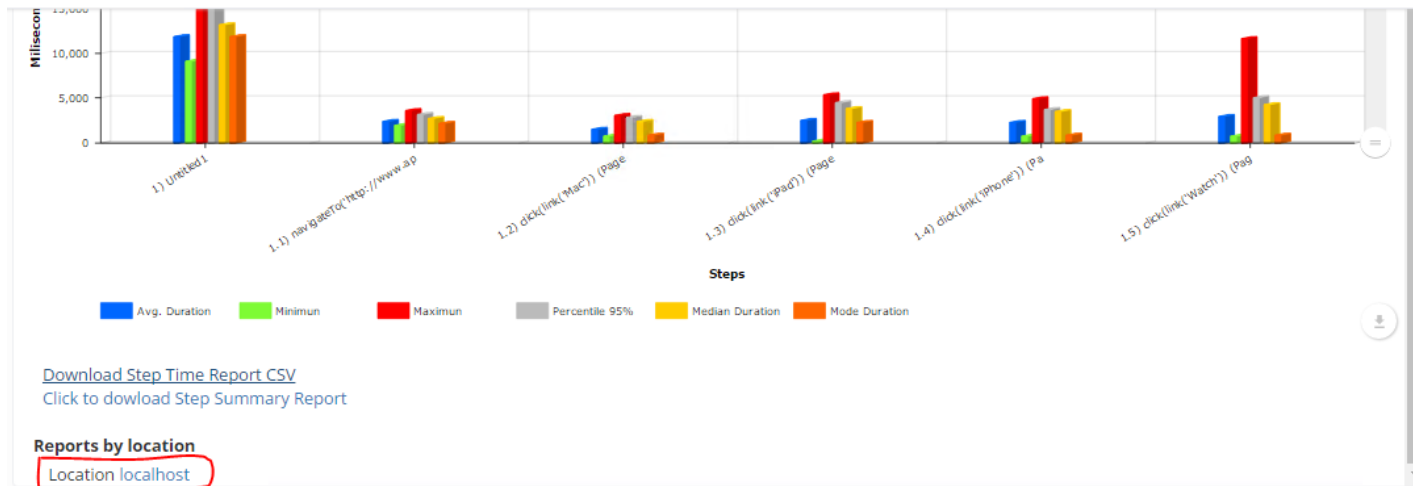
Step Summary Report

Unified Test Platform also provides a CSV report of the Step Summary. It can be opened in almost any spreadsheet editor.

	A	B	C	D	E	F	G
1	Step Name	Avg. Duration (Msec)	Minimum Duration (Msec)	Maximum Duration (Msec)	Percentile 95% (Msec)	Median Duration (Msec)	Mode Duration (Msec)
2	1) Untitled1	11828	9089	19761	15674	13145	11828
3	1.1) navigateTo('http://www.apple.com') (Page Fully Loaded)	2428	1973	3579	3156	2754	2204
4	1.2) click(link('Mac')) (Page Fully Loaded)	1549	785	3084	2829	2369	881
5	1.3) click(link('iPad')) (Page Fully Loaded)	2557	204	5386	4495	3811	2316
6	1.4) click(link('iPhone')) (Page Fully Loaded)	2350	806	4913	3779	3513	911
7	1.5) click(link('Watch')) (Page Fully Loaded)	2942	783	11645	5050	4254	907
8							
9							
10							
11							
12							
13							
14							
15							
16							

Location-Based Report

Unified Test Platform also provides a CSV report of the step times based on each location. It can be opened in almost any spreadsheet editor. There will be a separate link to download the CSV report for each location available.



Clicking on the location link will download the location-based report in CSV format as shown below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Duration	3 m 6 s																			
2	Start Time	Sat Jul 30 07:53:52 UTC 2016																			
3	End Time	Sat Jul 30 07:57:13 UTC 2016																			
4	Total User	110																			
5	Total Hits	20580																			
6	Total Thro	2244264.87 KB																			
7	Average H	110.65 hits/s																			
8	Average T	12065.94 KB/s																			
9	Consumes	98844.18 kbps																			
10	Testcases	110																			
11	Testcases	0																			
12	Testcases	0																			
13	Step pass	660																			
14	Step fail	0																			
15	Step error	0																			
16	All durations and sizes are for successes only																				
17																					
18																					
19	Testcase r	Total User	User Perc	Active Use	Step nam	Request n	Duration c	Response	Success C	Failure Co	Total Step	Total Req	Average C	Standard I	Minimum	Median D	Mode Dur	Duration E	Duration S	Duration F	Maxim
20	Testcase1	60		60					60	0	360	2380	11828	290469	9089	13145	11828	13145	13499	15674	197
21					1) Untitled1				60	0	60	2380	11828	290469	9089	13145	11828	13145	13499	15674	197
22					1.1) navigateTo('http://www.apple.com')				60	0	60	1020	2428	149615	1973	2754	2204	2754	3010	3156	35
23					myriad-set-pro_semibold.woff				60	0	60	19	0	0	17	20	18	20	23	27	
24						200	60	0	0	0	0	19	0	0	17	20	18	20	23	27	
25					image_large.svg				600	0	600	21	14961	17	34	18	34	35	35	35	

Note: In order to capture the location-based reports, the User has to enable the locations by going to the Test Cases tab and selecting "Users by Location" and assigning the users in each location.

Users by Location

Remove

Close

Save

localhost (users)

10

Node1 (users)

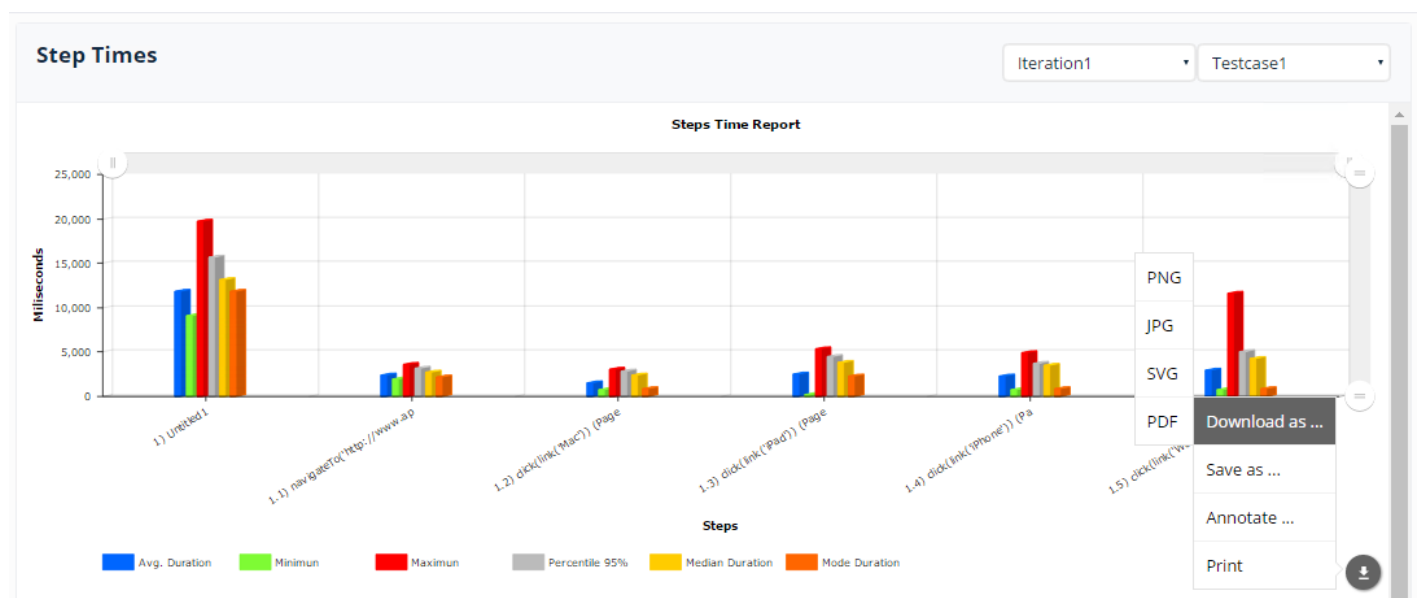
20

Node2 (users)

30

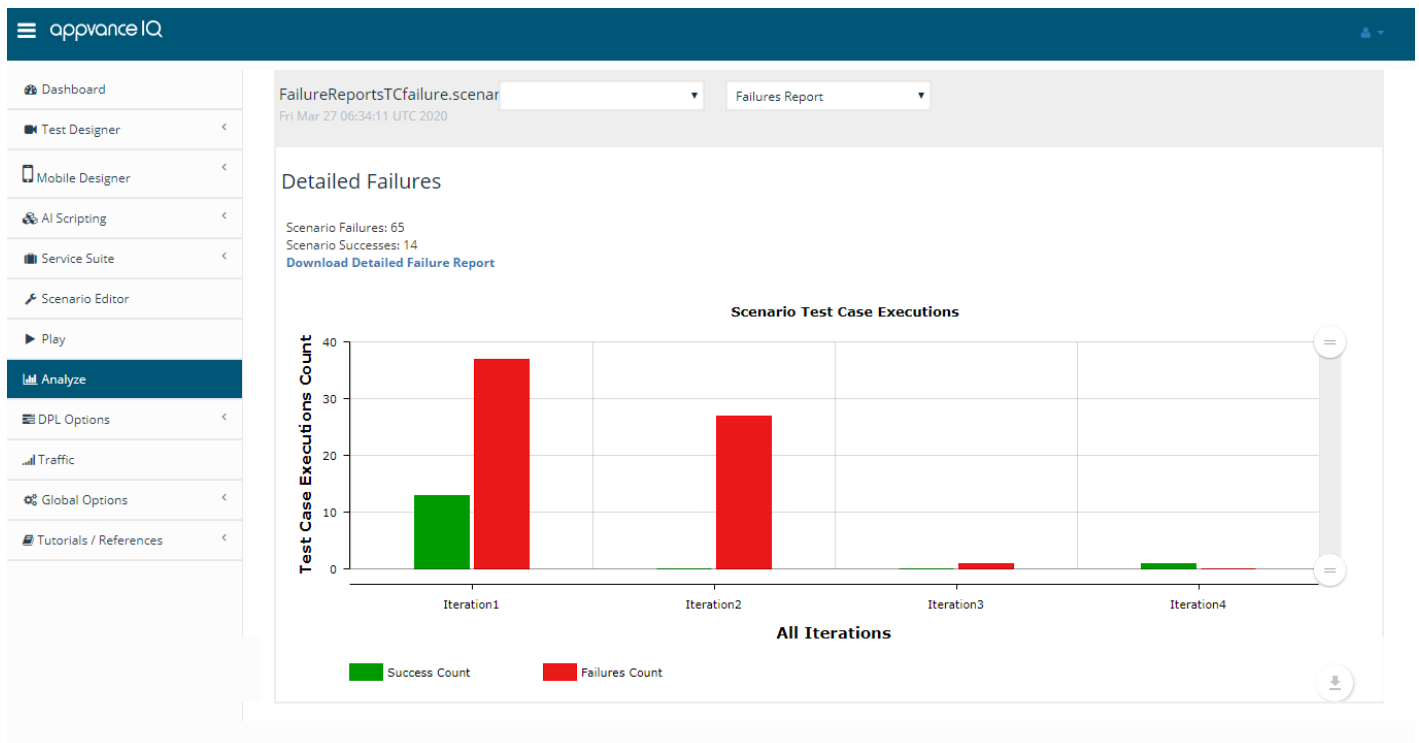
Downloading Graph

Unified Test Platform also provides a feature wherein the user can download the step times graph make changes to the graph like annotate etc.



Failures report

Failure report shows the number of successes and failures from all the iterations (Scenario wise). Just like the step times report, a bar graph is shown with the list of successes and failures.



To check the failures in detail, you can click the **Download Detailed Failures Report** link which will download an HTML report showing all the required data in detail.

AIQ Failure Report

Scenario: FailureReportsTCfailure.scenario

Date: Fri Mar 27 06:34:11 UTC 2020

Failure rate: 82.28% Successes: 14 Failures: 65

Scenario Type: Loadtest

Execution Id: 1585290851510

Iteration: Iteration1

Testcase: TC1 12 failures

Script: 1) TC1-S1 8 failures

Failed Step	Reason Detail	Reason	Qty
1.2) Random step failed 2	Error: random step failed 2	Error	5
1.3) Random step failed 3	Error: random step failed 3	Error	3

Script: 2) TC1-S2 4 failures

Failed Step	Reason Detail	Reason	Qty
<no inner step got executed>	Broken!	Script failed to start executing.	4

Iteration Summary

Iteration Summary is a CSV document that shows execution information during the iterations.

Iteration Summary

Select an Iteration

Iteration1

Select a specific Time Frame (minutes) to get the Iteration Summary report

3min : 6sec

Click to download the Iteration report in Comma Separated Value (CSV) format: [Iteration1](#)

Location [localhost](#)

Compare Summaries

Select the comparison criteria:

☒ Average Duration

☒ Success

☒ Failures

Base Line Type

Iteration Summary

Please load an iteration summary CSV file to compare against the current iteration summary

The iteration summary is the most complete report. It contains several rows showing each Test Case with the total transactions achieved, the total failures, and success. It also contains the average, maximum, minimum, and average duration of each Test Case and the Standard deviation.

It also indicates the number of users who executed each Test Case during the iteration.

Also, it shows the 85, 90, and 95 percentile of each transaction. This means the maximum duration of 85, 90, and 95% of the transactions per Test Case. Percentile is useful to set limits and confirm them. E.g. 90% of Test Case "Working Sequence" should take less than 10 seconds.

Finally, it shows all the above information for the iteration as a whole and not only its Test Cases.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Testcase r	Step nam	Request n	Success C	Failure Co	Total Step	Total Req	Average C	Standard I	Minimum	Duration	Duration	Duration	Maximum	Average S	Standard I	Minimum	Size 85	Pe Size 85	Pe Size 85	Pe Maxim
2	HAR Use Case			1636	0	6544	6544	382	238	52	602	652	823	1476	38	0	38	38	38	38	
3		1) http://localhost:8	1636	0	1636	1636	1636	98	76	14	173	193	226	654	1	0	1	1	1	1	1
4		http://loc	1636	0		1636	1636	0	0	0	0	0	0	0	1	0	1	1	1	1	1
5		1) http://localhost:8	1636	0	1636	1636	1636	67	61	10	120	141	180	796	2	0	2	2	2	2	2
6		http://loc	1636	0		1636	1636	0	0	0	0	0	0	0	2	0	2	2	2	2	2
7		2) http://localhost:8	1636	0	1636	1636	1636	55	57	10	88	104	139	565	1	0	1	1	1	1	1
8		http://loc	1636	0		1636	1636	0	0	0	0	0	0	0	1	0	1	1	1	1	1
9		2) http://localhost:8	1636	0	1636	1636	1636	89	80	13	150	171	236	816	32	0	32	32	32	32	32
10		http://loc	1636	0		1636	1636	0	0	0	0	0	0	0	32	0	32	32	32	32	32

Compare Summaries

The iteration Summary tab also allows the user to compare the reports of the same test that was run previously. This helps to compare the changes in response times.

By default, the comparison takes place between the current execution and the execution which was run previously.

The screenshot of the UI is shown below.

Click to download the Iteration report in Comma Separated Value (CSV) format: [Iteration1](#)

Location localhost

Compare Summaries

Select the comparison criteria:

- ☒ Average Duration **Base Line Type** Iteration Summary ▾
- ☒ Success
- ☒ Failures

Please load an iteration summary CSV file to compare against the current iteration summary [?](#)

Loaded File

Load Baseline

Compare Summaries

The user loads the baseline and clicks on Compare Summaries and a report in HTML format will be opened which gives the comparison between executions.

Summaries Comparison - New_apple.scenario

	Success Count	
	Baseline	Sat Jul 30 UTC 2016
Testcase1	60.0	0.0
1) Untitled1	60.0	0.0
1.1) navigateTo('http://www.apple.com') (Page Fully Loaded)	60.0	0.0
appleicons_thin.woff	60.0	0.0
image_large.svg	600.0	0.0
myriad-set-pro_extrathin.woff	60.0	0.0
myriad-set-pro_semibold.woff	60.0	0.0
myriad-set-pro_text.woff	60.0	0.0
myriad-set-pro_thin.woff	60.0	0.0
status	120.0	0.0
1.2) click(link('Mac')) (Page Fully Loaded)	60.0	0.0
ac-analytics.js	1.0	0.0

Security Report

The security report tab is only available whenever you execute an App Penetration test type, another kind of scenario will not display this tab since there will be no report available. This report is generated based on the ZAP OWASP open-source tool which is included in AIQ as one of our engines to perform web security tests. It is a formatted PDF file with the list of vulnerabilities of a website returned by OWASP.

Security Report

Scenario: AppPenOwasp.scenario

Mon Jul 18 11:15:09 CST 2016

Alert Detail

Low(Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://www.bosquenuboso.org/bos.css
Other information	This issue still applies to error type pages (401, 403, 500, etc) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scanner will not alert on client or server error responses.
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.
WASC Id	15
Medium(Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

All about requests and responses

Below are some of the details of requests and responses.

All the links below a page are requests with size in bytes - This is in the controller window after running a test.

New-scenario.scenario
Wed Oct 18 17:01:38 IST 2017 [Edit Scenario](#)

Status: Execution done | Iteration: Iteration1 | Succeeded: 268 | Failed: 0 | [Play Scenario](#)

Output | Live Reports | **Test case(s) Executions** | Testnode | Active Virtual Users | Validation Suite | Autocorrelation

Test case(s) Executions | Most recent 5 test cases executions

Result	Test case	Duration	TestNode
Success	Testcase1	4641	localhost
Success	Testcase1	4788	localhost
Success	Testcase1	4658	localhost
Success	Testcase1	5799	localhost
Success	Testcase1	5068	localhost

Output | Error | **Request & Response** | Snapshots

Click to show response

1) Untitled1. 4641 msec.
1.1) http://demosite.appvance.com/. 4641 msec.

http://demosite.appvance.com/
Date: Wed, 18 Oct 2017 11:36:45 GMT Server: thin 1.6.2 codename Doc Brown X-Frame-Options: SAMEORIGIN X-Frame-Options: SAMEORIGIN X-XSS-Protection: 1; mode=block X-Content-Type-Request-Id: 72c10e3-97a9-41ed-aabc-be096b3a5568 X-Runtime: 0.042171 Set-Cookie: _htdocs_session=NViXtDhVUTBGUitKRkNMK2JHNM0xS3NveGtwVE5ISDFHWkQvNVRUVigzYjM2d1IzMxE5ZERTWTh5WkRObGRjNU1Tb2p1TmtBamNuRFE2QWpKNG9yVORCT0s1cDdLYWxl-5730a7bb7efbdc922d2bbc2fcb80b5c774947; path=/; HttpOnly Vary: Accept-Encoding Keep-Alive: timeout=5, max=49 Connection: Keep-Alive

http://demosite.appvance.com/assets/spree/frontend/all-fe935964ffc1a8877767afcdc98fdeadc3e2455f2a5986e52ad8f3e1722d9f.js
Date: Wed, 18 Oct 2017 11:36:45 GMT Server: Apache X-Frame-Options: SAMEORIGIN Vary: Cookie Accept-Encoding Last-Modified: Fri, 12 May 2017 09:34:53 GMT ETag: "2d369-54f506ae62940"

http://demosite.appvance.com/assets/logo/spree_50-0ae687ccd486b63b01687b1df9e5394430f6fce3ab40c28250b179be4363ddc1.png
Date: Wed, 18 Oct 2017 11:36:46 GMT Server: Apache X-Frame-Options: SAMEORIGIN Vary: Cookie Last-Modified: Fri, 12 May 2017 09:34:44 GMT ETag: "872f-54f506ae5cd500" Accept-Ranges: b

http://demosite.appvance.com/spree/products/21/small/ror_tote.jpeg?1499978138
Date: Wed, 18 Oct 2017 11:36:46 GMT Server: Apache X-Frame-Options: SAMEORIGIN Vary: Cookie Last-Modified: Thu, 13 Jul 2017 20:35:39 GMT ETag: "11fc-55438e02cfe53" Accept-Ranges: b

request (2) | Show all | X

Report - Iteration summary report - provides details in a CSV sheet, This shows the request name, the response codes of each request, a number of occurrences of this request under a URL, what time it took to load that request (Average duration), the minimum and maximum duration is also provided in case of the performance load test.

Filter response content:

We also have the option to filter out the response contents to be stored in the database or not with this feature.

Visit Global Options for more details.

Plot Graphs:

We can use Results on time graphs to plot graphs based on a Request from a particular step.

There are a lot more that we could offer with requests and responses, but again since we have all these data with us, presenting them to the customer as they requested does not take much time for the Appvance team.

Assertions:

Assertions can be added to Requests as well, please refer to Validation and Store Test Suite - HAR for more details.

Last updated: July 10, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Review Live Results While a Test Runs

Review Live Results While a Test Runs

Use live results to monitor a test while it is running. Live data helps detect invalid runs early, but final analysis should use completed results after the run has ended and all data has been processed.

Monitor during execution

- Active users and ramp behavior.
- Error rate and repeated failure messages.
- Response-time trend changes.
- Test Node or environment health symptoms.

When to stop a run

1. Stop if credentials, data, or environment selection is wrong.
2. Stop if the application under test becomes unstable outside the intended test scope.
3. Stop if setup errors make the result invalid.

Last updated: September 18, 2026

Determine if the Test System is a Performance Bottleneck

AIQ is a distributed test environment where tests run remotely on TestNodes. A TestNode is software running on a server that uses threads to operate multiple virtual users concurrently.

As the number of concurrent virtual users increases, the TestNode consumes additional CPU, memory, and network resources. If the TestNode reaches its resource limits, it can become a performance bottleneck and affect the results of the test.

To determine how many virtual users a TestNode can support before reaching its resource limits, AIQ recommends using **Calibration Testing**.

The calibration testing methodology consists of three steps:

1. **Identify the test use cases.**

A test use case contains the steps required to operate the application.

2. **Identify the optimal throughput of the application and test environment.**

Gradually increase the number of virtual users and monitor the application's performance and the resources consumed by the test environment.

3. **Run the complete performance test and analyze the results.**

Use the results from the calibration testing to determine whether the TestNode or the application infrastructure becomes a bottleneck during the complete performance test.

For detailed instructions, see the **Calibration Test Methodology** chapter of the AIQ User Guide.

Estimating Server Resources Required to Support a User Population

Performance testing with **AIQ Monitor** can be used to identify the server-side resources required to support a population of application users.

AIQ Monitor is an agent-based service that observes CPU, network, and memory utilization while AIQ executes a performance test. AIQ stores the monitored values in the Results Repository, normally using MySQL or Oracle.

The **Resource Utilization** report correlates the application's performance with the resources consumed by the monitored servers. This helps identify whether application infrastructure becomes a bottleneck as the number of virtual users increases.

Use the **Resource** panel in the **Scenario Editor** to identify the AIQ Monitors from which you want to collect results. For example, AIQ Monitor can be configured on:

- Web servers
- Application servers
- Database servers
- Load balancers

Viewing Resource Utilization

The **Resource Utilization** report is available from the **Analyze** section.

To view the resource utilization for a test execution:

1. Open **Analyze** from the AIQ Start Screen.
2. Select the execution you want to analyze.
3. Select **Resource Utilization** from the list on the left.

Use the report to compare application performance with CPU, memory, and network utilization. This information can help determine whether the TestNode or application infrastructure is limiting the number of concurrent virtual users that the environment can support.

Last updated: August 17, 2026

If you have any feedback on this topic, please contact us at documentation@appvance.ai.

View Results from a Previous Test Run

View Results from a Previous Test Run

Use previous test run results to compare baselines, investigate past failures, and track performance trends over time. Confirm that the compared runs used the same environment, scenario version, data assumptions, and load model before drawing conclusions.

Recommended workflow

1. Locate the run by project, suite, execution date, or execution ID.
2. Open the completed result and review summary status first.
3. Compare key metrics or failed steps against the current run.
4. Record relevant links or execution IDs in the related defect, report, or maintenance note.

Comparison notes

- Avoid comparing runs that used different environments or data sets unless that difference is the point of the analysis.
- Use steady-state periods for performance comparisons.
- Document any scenario or application changes between runs.

Last updated: September 18, 2026

What Can Go Wrong During Results Analysis

Recommended guidance

Results analysis can be misleading when test data, environment, load model, monitoring, or reporting context is incomplete. Review failures and performance results with enough evidence to separate application defects from test or environment issues.

Common causes

- Unstable test data, exhausted accounts, or missing cleanup.
- Browser, Test Node, network, database, or integration issues.
- Incorrect assertions, waits, ramp-up, pacing, or load model assumptions.
- Small sample sizes, high error rates, or incomplete monitoring.
- Comparing runs from different environments or configurations.

Review workflow

1. Start with the first meaningful failure, not the final cascading error.
2. Compare with a known-good run and check environment changes.
3. Use logs, reports, screenshots, metrics, and test data evidence together.
4. Document conclusions, open questions, and retest criteria.

Last updated: October 7, 2026