

# Blitz\_io

## Plugin Information

View Blitz.io [on the plugin site](#) for more information.

## Blitz.io Plugin

This plugin allows you to run web load tests from the cloud using [Blitz.io](#).

## Installation

1. go to `/pluginManager/advanced`, hit 'check now' button to get all the latest updates from update center.
2. go to `/pluginManager/available`, find the 'Blitz.io Plugin' under 'External Site/Tool Integrations' section, then install it.

<input checked="" type="checkbox"/>	<b>Blitz.io Plugin</b> This plugin allows you to run web load tests from the cloud using <a href="http://blitz.io">Blitz.io</a> .	1.03
-------------------------------------	--	------

3. after installation is done, you should be able to see 'Blitz.io Plugin' in `/pluginManager/installed` (you may need to restart Jenkins if necessary)

<input checked="" type="checkbox"/>	<a href="#">Blitz.io Plugin</a> This plugin allows you to run web load tests from the cloud using <a href="#">Blitz.io</a> .	<a href="#">1.03</a>		
-------------------------------------	---	----------------------	--	--

## Enter Blitz.io Account Info

1. go to `/configure`, find the Blitz.io section, then enter your blitz.io account user ID and API key. If you don't have a blitz.io account yet, you can register one [here](#), it's free!

## Run Blitz.io Tests After Build

1. go to your job's configure page, such as `/job/blitz.io%20plugin/configure`, in the 'Post-build Actions', check the 'Blitz.io' box.
2. if you want to run a [sprint](#) test, check the 'Sprint' box, then enter the command and the response time threshold.
3. if you want to run a [rush](#) test, check the 'Rush' box, then enter the command and the error rate threshold.

Post-build Actions	
<input type="checkbox"/>	Aggregate downstream test results <span>?</span>
<input type="checkbox"/>	Archive the artifacts <span>?</span>
<input checked="" type="checkbox"/>	Blitz.io
Sprint	<input checked="" type="checkbox"/> <span>Run a sprint test</span>
Command	<input type="text" value="http://jeff-blitz.herokuapp.com/"/> <span>?</span>
Response Time (ms)	<input type="text" value="100"/> <span>?</span>
Rush	<input checked="" type="checkbox"/> <span>Run a rush test</span>
Command	<input type="text" value="--pattern 1-250:20 -r california http://jeff-blitz.herokuapp.com/"/> <span>?</span>
Error Rate (%)	<input type="text" value="1"/> <span>?</span>

4. hit the 'Save' button

Note: you may only blitz urls that are reachable from the public web, so please don't enter "localhost", "intranet links" etc in the command line.

## Interpret Blitz.io Test Results

1. after your job's build is done, click into that build, you will see a 'Blitz.io' link on the left hand side.

Jenkins » [blitz.io plugin](#) » #1

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Blitz.io](#)

## Build #1 (Jun 5, 2012 3:49:51 PM)

 No changes.

 Started by user [anonymous](#)

2. click the 'Blitz.io' section, you will see the Sprint or/and Rush Results

## Blitz Test Report

### Sprint Test

Command	Region	Connection Time (ms)	Response Time (ms)	Threshold (ms)	Status
<a href="http://jeff-blitz.herokuapp.com/">http://jeff-blitz.herokuapp.com/</a>	virginia	41	49	100	Passed

#### GET / HTTP/1.1

**Duration:** 49 ms  
**Connect:** 41 ms  
**Response:** HTTP/1.1 200 OK

**Request headers**

- **Host:** jeff-blitz.herokuapp.com
- **X-User-IP:** 74.85.18.250
- **X-User-ID:** 5e9437ea786609055ff7df89b350f92b
- **User-Agent:** blitz.io; 5e9437ea786609055ff7df89b350f92b@74.85.18.250

**Response headers**

- **Content-Type:** text/html
- **Content-Length:** 5780
- **Connection:** keep-alive

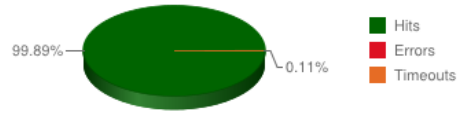
## Rush Test

Command	Region	Error Rate (%)	Threshold (%)	Status
--pattern 1-250:20 -r california http://jeff-blitz.herokuapp.com/	california	0.11%	1%	Passed

### Summary

This **rush** generated **1,822** successful hits in **20.00 seconds** and we transferred **10.81 MB** of data in and out of your app. The average hit rate of **83/second** translates to about **7,172,524** hits/day.

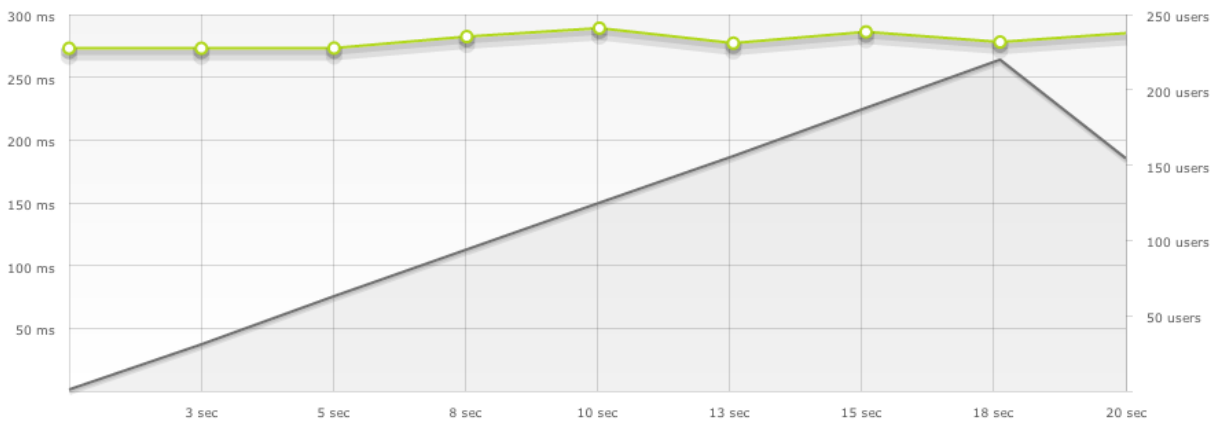
You got bigger problems though: **0.11%** of the users during this **rush** experienced timeouts or errors!



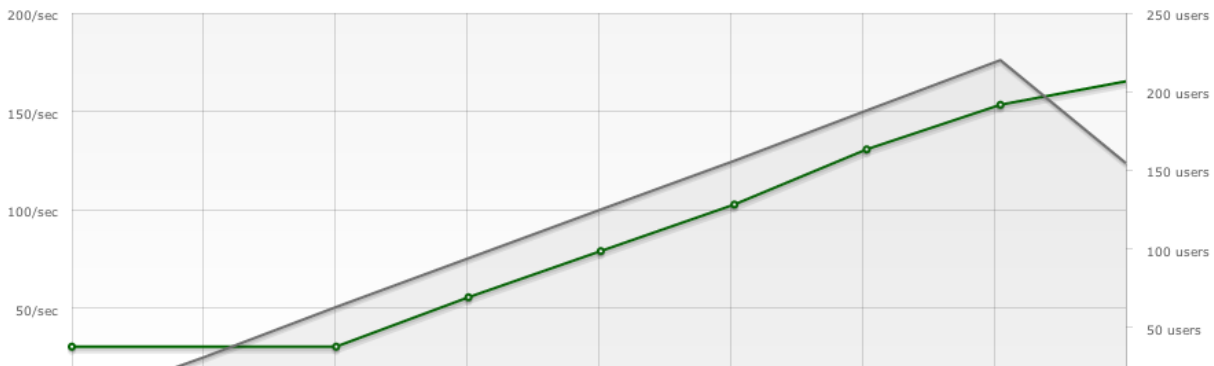
### Timeouts

The first timeout happened at **17.62 seconds** into the test when the number of concurrent users was at **220**. Looks like you've been rushing with a timeout of **1 second**. Timeouts tend to increase with concurrency if you have lock contention of sorts. You might want to think about in-memory caching using [redis](#), [memcached](#) or [varnish](#) to return stale data for a period of time and asynchronously refresh this data.

### Response Times



### Hit Rate



3. your build will be marked as 'success' if all the tests passed, otherwise it will be marked as 'failure'