

# Cool Solution: Quick and Easy Load Testing for Novell Access Manager (NAM) 3.x

Written by *Corbin Links*, Identity Access Management (IAM) Analyst

[The Washington Post Company](#)

[linksc@washpost.com](mailto:linksc@washpost.com)

## Article Summary

Challenge:

How to quickly, reliably and easily conduct system load testing for Novell Access Manager 3.x (Hereafter referred to as “NAM”)

Solution:

Configure and run NAM tests at LoadStorm.com

## Audience

- Technical Project / Program Managers
- Quality Assurance (QA) Team Members)
- Application Systems Engineers
- Technical Marketing Personnel

## Assumptions

This article assumes you have a basic familiarity with:

- Novell Access Manager Architecture
- Basic Single Sign On (SSO) Concepts
- Basic concepts of load testing and capacity planning

- Basic functionality of web browsers and HTTP status / exit codes

## Introduction to NAM /SSO Load Testing

*Disclaimer: The author is in no way affiliated with LoadStorm.com, its partners or subsidiaries. Information provided in this article is intended solely for informational and convenience purposes. Information and statements provided in this Cool Solution are those of the author and not [The Washington Post Company](#).*

Arguably, one of the most crucial tasks of implementing any form of identity access management (IAM) framework or SSO system is the ability to verify that the system can withstand large amounts of concurrent load. Authentication (authN), Authorization (authZ) and Auditing operations though lightweight and inexpensive from a transaction standpoint, can negatively impact system performance when occurring in large quantities over short periods of time. How much or how little load your NAM system can or should handle will be defined by the requirements of your application and any corresponding Service Level Agreements (SLA) you may have in place.

Unfortunately, load testing is one of the crucial elements of any Authentication Infrastructure that can be often overlooked or short changed during pre and post implementation stages. Here are just a few of the reasons:

- Comprehensive and relevant load testing can be extremely expensive, costing anywhere from thousands of dollars, well into the tens of thousands.
- SSO deployments are extremely time, resource and cost intensive. It is not uncommon to be running behind project schedule and start dropping task items considered to be “lower priority.” These often mis-categorized items include documentation, load testing, and training – just to name three.
- Load testing tools can be very difficult to implement and set up and reuse. Slight changes in the testing platform or the particular version of NAM you are running may require brand new test cases for each scenario.
- Lack of meaningful internal benchmarks or external usage data which can guide system relevant testing requirements. Generally this scenario exists for one or more of the following reasons:
  - NAM/SSO is new to the organization. No hard SLA requirements yet exist.
  - There is little or no application analytics data to determine usage patterns, peak and valley usage, raw transaction numbers, errors, daily and per-transaction bandwidth statics and many others.
  - Future growth patterns are unknown or inadequately planned

- What does your current SLA state? (If applicable.) Does your SLA state explicit uptime requirements? Specific availability requirements for certain hours of the day?

## Determining Your Load Testing Requirements

Application load testing requirements vary tremendously between application designs, infrastructure environments, business requirements and usage requirements. The following list provides a high-level general set of testing guidelines. This is the list we have used for our NAM deployment and similar deployments in the past. Though far from comprehensive for all environments, the goal is to provide general guidelines to help you get started.

- Who is your user population? (These are the users that will be interacting with the application that NAM is protecting, not NAM-specific users)
  - Internal users only?
  - External users?
  - Mix of internal and external users
- What is the application designed to do? Is it highly transactional, RESTful, general use, highly specialized? (Limited to only a few users or a specific department for example.)
- Do you have web analytics currently installed on your present Access Management system? If not, how about the underlying client application that is protected by the Access Management System?
  - If no analytics exist, do you have reasonable use projections based on server logs or similar data?
  - Raw daily user counts, averaged over a period of 30 days
  - Time-of-day hit counts averaged over a minimum of two weeks
  - Geographic load factors
    - Example: 75% of your user population is located in the States, while the other 25% is distributed over smaller locations around the world (or vice versa)
- What is your expected annual growth percentage? Will the tests you build and run today still be valid three months from now? Six months? One year?

## Components of NAM Load Testing

At its most basic level, your NAM load testing will stress the following components:

- The *Linux Access Gateway* (if applicable)
- The *Identity Server*
- The application server SSO agent (if applicable) and/or if not using a *Linux Access Gateway*
- The application or web server which hosts your application
- All hardware and network points in between the end application page or form, and the end user browser

For the remainder of this document, we will assume the following basic transaction is common to all our NAM-protected testing scenarios:

1. Open the target application URL: <http://www.myapplication.com>
2. NAM Linux Access Gateway (LAG) or a proxy agent installed on the application's web or application server will intercept the request
3. The user's browser will redirect to a NAM-based login page generated from the Novell Identity Server
4. The user enters credentials into the page
5. The user submits the page back to the Identity Server
6. User credentials are checked in accordance with the resource's access policy. Generally, this action submits the user credentials to a backend user store (LDAP-based) to verify username and password, or other types of credentials. For our load test, we are assuming basic username/password authentication single-factor authentication
7. Identity Server validates or refuses the session
8. The resulting page is sent back to the user's browser
  - a. "Success" will open the application's home page
  - b. "Fail" will open a redirected error URL containing additional information

Here are the steps from browser/UA (User Agent) standpoint:

1. Open a browser
2. Open <http://www.myapplication.com>

3. Enter username and password
4. Click “login” or “submit”
5. Receive application homepage in the browser
6. Enter logout url: <http://www.myapplication.com/logout> (or simulate whatever “logout” link is appropriate on the site)

## Our Testing Requirements

Before we set up the testing solution, here were the basic requirements that we wanted to test against:

1. Application: Internal Intranet Portal
2. NAM Configuration: Linux Access Gateway (LAG) based
3. Requirement: 500 concurrent users under load
4. Load Peak: no peak or special time assumed. This particular test system had a base load requirement of 500 concurrent users at full peak loading. (Using a “linear” versus a “stepped” test.)

## Our NAM Infrastructure Components

- NAM LAG
- NAM Identity Server
- Windows 2003 Active Directory Authenticator
- PeopleSoft HR Portal

## Load Testing Solution: LoadStorm.com

### Introduction

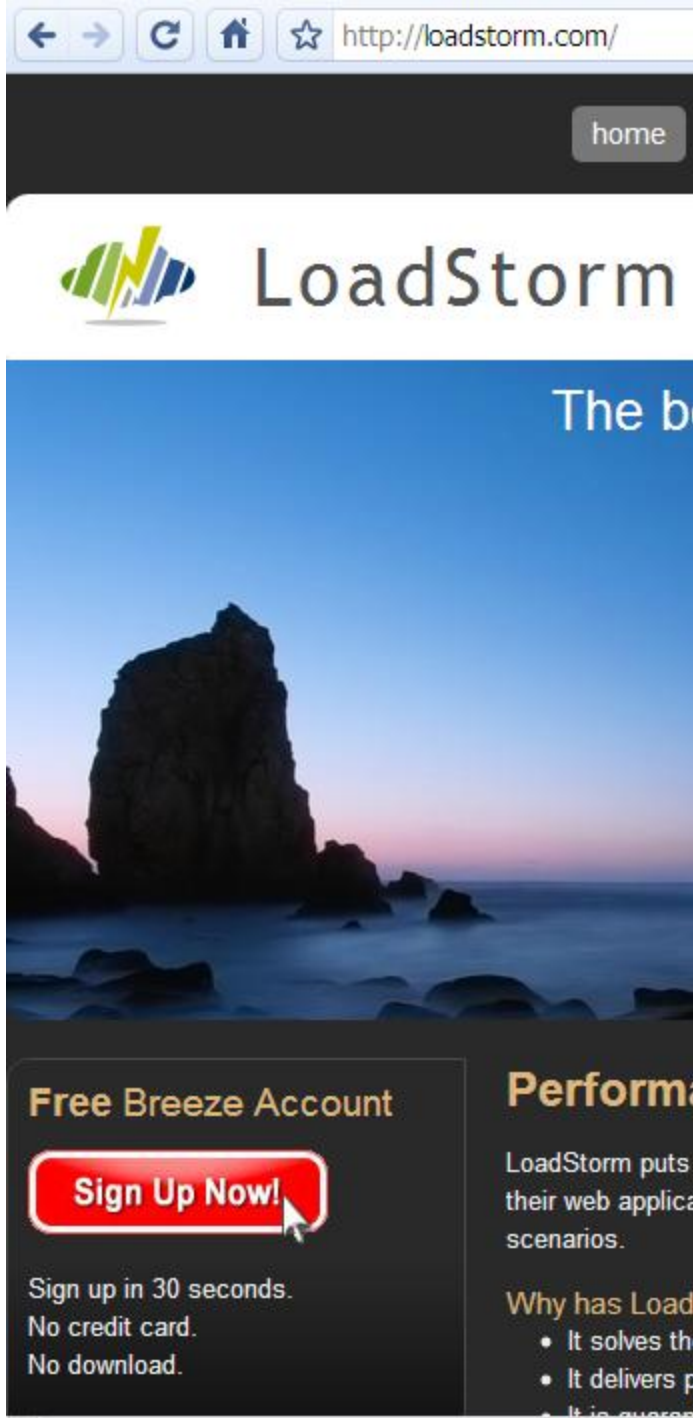
LoadStorm.com is a highly scalable cloud-based load testing service. For additional information about the LoadStorm service of company information, please visit their website at <http://www.loadstorm.com>.

## LoadStorm.com Testing Prerequisites

- A LoadStorm.com test account, which can be obtained for free (as of this writing) over at <http://www.loadstorm.com>
- A list of user ID's and passwords which are allowed to authenticate to NAM
  - NOTE: If you are testing 500 users, you will need 500 distinct and active accounts in your authentication repository
  - In most cases, all users will have the same password (set en masse)
- A test plan which can be completely run from a user browser
  - I recommend having all steps, url's and screenshots in a single document, which can be referenced when building the test case within LoadStorm
- At least one test scenario (the "Scenario") contains "Steps"
- One or more test "Steps" (each "Scenario" requires a minimum of one "Step")

## Creating a LoadStorm.com Account

1. Visit <http://www.loadstorm.com>
2. Click the big red "Sign Up Now!" button to create a "Free Breeze Account" (This free account lets you run any test as often as you want, using 25 or fewer users. 25 free users are included in the basic account.)



3. Complete the basic sign up form:



# LoadStorm.com

Load testing in the cloud

Email

 \*

Password

 \*

Confirm password

 \*

First name

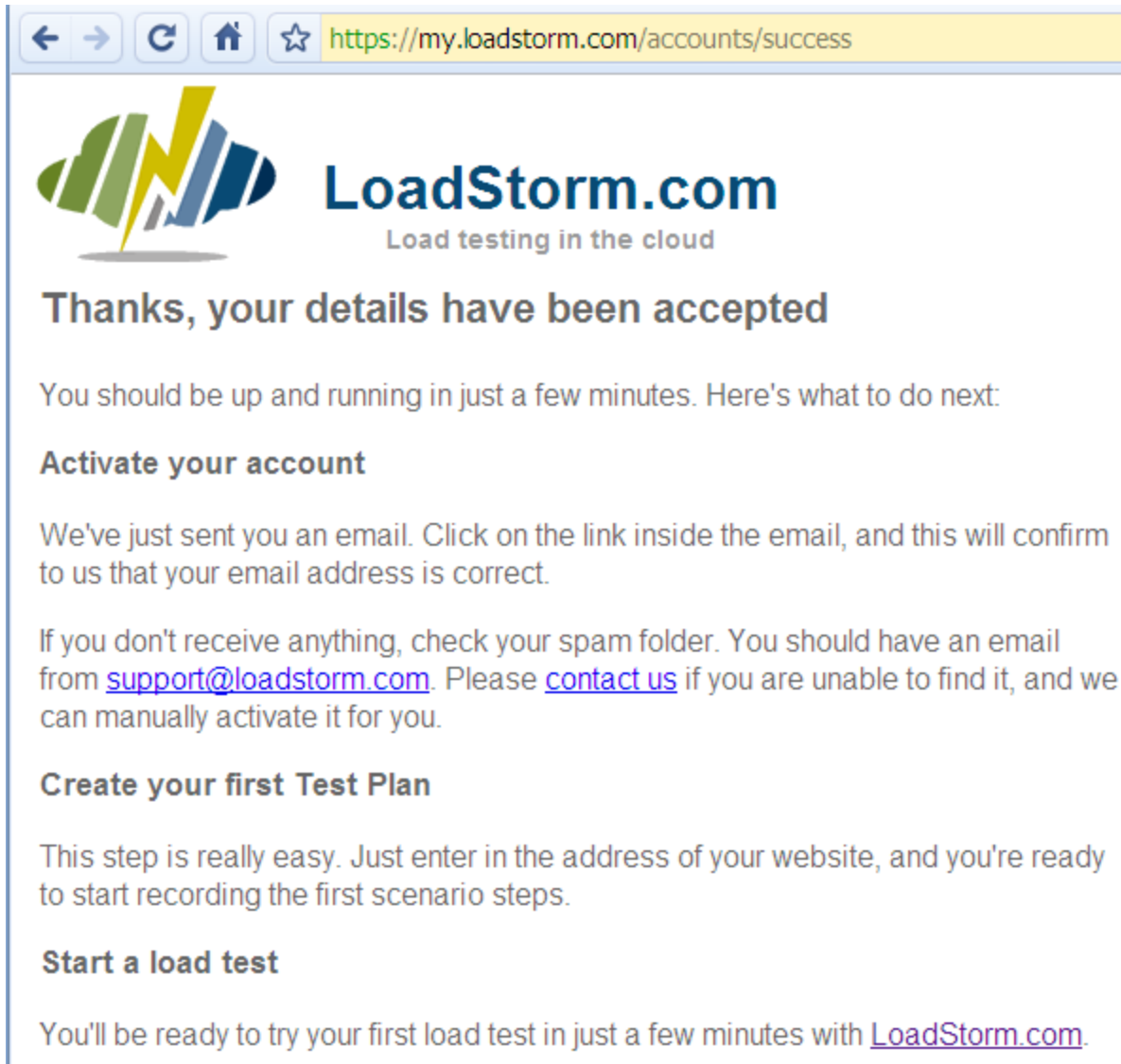
Last name

Phone number


Time zone

 \*

4. Receive the confirmation page after clicking the "Sign up" button:



← → ↻ ⬆ ☆ <https://my.loadstorm.com/accounts/success>



**LoadStorm.com**  
Load testing in the cloud

## Thanks, your details have been accepted

You should be up and running in just a few minutes. Here's what to do next:

### Activate your account

We've just sent you an email. Click on the link inside the email, and this will confirm to us that your email address is correct.

If you don't receive anything, check your spam folder. You should have an email from [support@loadstorm.com](mailto:support@loadstorm.com). Please [contact us](#) if you are unable to find it, and we can manually activate it for you.

### Create your first Test Plan

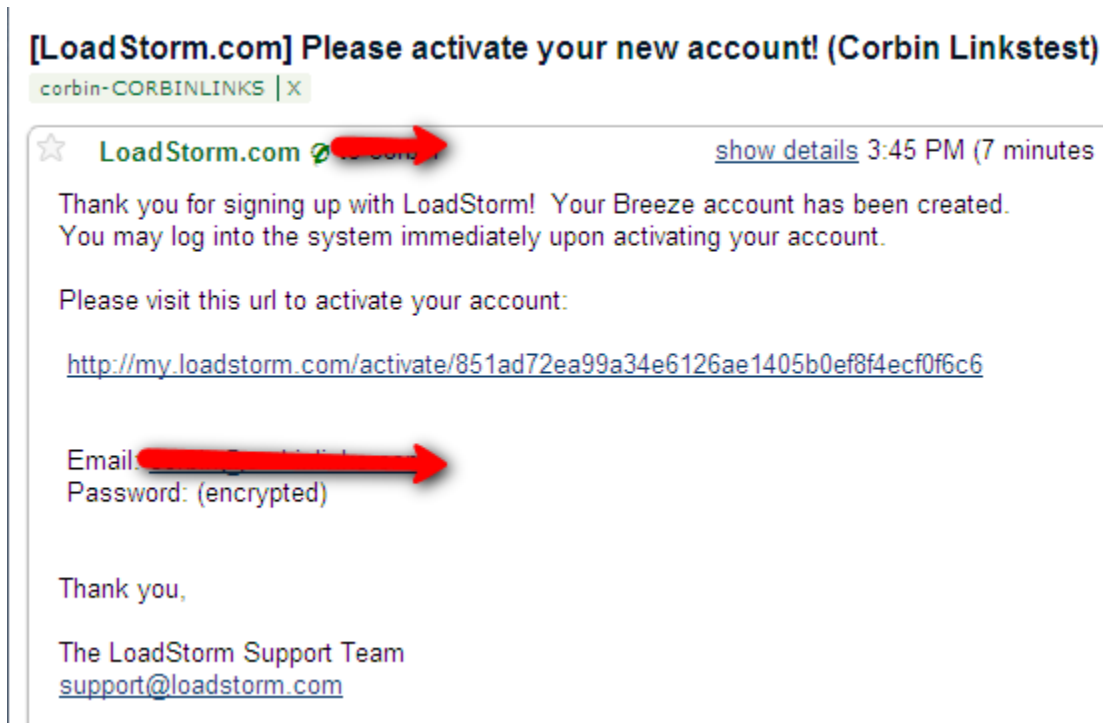
This step is really easy. Just enter in the address of your website, and you're ready to start recording the first scenario steps.

### Start a load test

You'll be ready to try your first load test in just a few minutes with [LoadStorm.com](#).

5. Look for the confirmation email in your inbox

6. Confirm the confirmation email



7. Log in to your account

## Configuring LoadStorm.com

Once logged into LoadStorm.com, you can start configuring and running test cases. Since this *Cool Solution* is not intended as a detailed tutorial on all the options and possibilities within LoadStorm.com, we will instead focus on the core elements of creating and running a NAM-centric test case.

If you have not previously used LoadStorm, it is a good idea to experiment and familiarize yourself with one of the existing demo cases, such as the "CIA Factbook" case. Running a demo case will give you a pretty good feel for how the tool operates and how it outputs results. Additionally, the demo cases are designed to be copied to new cases and customized to fit other use environments.

Before building and configuring your first test case in LoadStorm, here are a few points to keep in mind:

- The case testing model is : *Plan* → *Scenario* → *Steps*
- As of this writing, LoadStorm's current version requires a full successful run of each step in a test before a subsequent step can be run. If a particular step does not complete, it follows that attempts to add a dependent step will fail.

- Each step within the load scenario will complete and deliver results to the active LoadStorm scenario.
- Every time a new URL from a new domain is added, LoadStorm considers it a new server. To protect your environment from unscrupulous people using LoadStorm as DDoS method against your network, LoadStorm requires all servers to be verified. This can be done in one of two ways:
  1. Place a special file in the root of your application/web server and Novel Identity Server
  2. Add a string of code to the root page on your site
    - For NAM deployments, it's easiest to use the "touch" command in \*NIX systems, or the "copy con" command in Windows-based systems to create a 0-byte file with the correct name. Then click the "Verify" button to verify server ownership and away you go!

Servers

Name	Address	Status
Server friendly name	Server url	Verified

Verification Status

- Results returned on completion of each step in the scenario will determine what actions are available next. For example, if your test is a simple scenario like ours (hit a website, enter credentials on the NAM-redirected form, authenticate, redirect to an application target page, logout,) then the NAM login form would not be available to populate until after the first "visit the site" step.
  - NOTE: As of this writing, LoadStorm supports three possible test actions:
    - *Open a page*
    - *Click a link*
    - *Process a form*
- NAM's use of iFrames can cause some additional configuration challenges, because LoadStorm *must* have the ability to parse the HTML of any page to load the form and parse the elements within. This is further complicated by the extensive differences in login page composition between NAM 3.0 and 3.1. In our environment, we were in the process of upgrading from 3.0 to 3.1 and had to configure separate test runs for both environments. I recommend opening your full NAM redirect link first in your test scenario step, before attempting to directly open your target page. For instance, instead of making the first LoadStorm step: "Open <http://www.myapplication.com>", instead "Open"






<https://namids.myapplication.com/nidp/idff/sso>, (or relevant link if you are remapping NAM-standard links.) This approach will make the form accessible to LoadStorm and LoadStorm can parse the NAM login form fields "Ecom\_User\_ID" and "Ecom\_Password" and read input from a single constant, or a CSV input file.

- Collections of users for running different sizes load of load tests can be configured at the bottom of the "Build" tab.
- If you use the "Copy" action to copy existing test cases to new test cases, or existing scenarios to new scenarios, you will have to manually run the copied scenario again for LoadStorm to recognize all of the available actions within the test (for instance, if a form is available for parsing or not)
- *Important Note:* As of this writing, LoadStorm treats any URL delivered in over 35 seconds as a "408 timeout error" condition. When an error condition occurs, LoadStorm automatically "fails" the current virtual user and moves to the next user on the list. This is important to understand because LoadStorm makes no assumption, rightly or wrongly, as to whether or not the URL *could* have been loaded had it been given enough time. When we stress tested our NAM sandbox environment, we found that the bulk of our errors related to the server having to work too hard to manage authentications and return results.

## ADD SERVER

At the bottom of the "Build" tab, you will see the option to add servers or additional "Form Data Sets." "Form Data Sets" are CSV files containing header rows, usernames and passwords. View the "Tiny" or "Small" files included with your LoadStorm account to get the idea

## Form Data Sets

Name	Number of rows	
TRN_UAT-25_v_users	25	<a href="#">details</a> 
loadstorm_peoplesoft-test1.csv	25	<a href="#">details</a> 
test_190_users_for_LOGOUT_testing	194	<a href="#">details</a> 
400_user_slam	400	<a href="#">details</a> 
500_user_slam	500	<a href="#">details</a> 
Tiny (system)	10	<a href="#">details</a>
Small (system)	100	<a href="#">details</a>
Huge (system)	5000	<a href="#">details</a>

## UPLOAD DATA

Uploads CSV to the LoadStorm system

## Test Case List View in the LoadStorm UI

Home **Build** Run Analyze

**Test Plans**

	Plan Name	Scenarios	Load Tests
<b>DETAILS</b>	<a href="#">CIA Factbook on S3 (example)</a>	3	
<b>DETAILS</b>	<a href="#">DMOZ excerpt on EC2 (example)</a>	3	

Sample demo cases

# of scenarios per case

## Test Case Scenario and Step View in the LoadStorm UI

Home **Build** Run Analyze

[Build](#) > CIA Factbook on S3 (example)

**Plan: CIA Factbook on S3 (example)** **EDIT** **COPY PLAN**

**Description:** Amazon's Simple Storage Service is a web service that provides online storage capacity.

**Server(s):** [CIA Factbook on Amazon S3](#) (verified)

Scenario	Weighting	Steps	Average Duration	Estimated transfer
<a href="#">Read 3 pages</a>	1 (33.3%)	3		
<a href="#">Read 8 pages</a>	1 (33.3%)	8		
<a href="#">Read a bunch of pages</a>	1 (33.3%)	20		0 KB

ADD SCENARIO NEW LOAD TEST

Scenarios within a Test Plan

These are actually three separate scenarios, rather than steps as described in the annotation red box. Steps are shown once you click on a scenario.

Now, let's set up a more useful test. Here is what we did:

## Real Case: ERP Time Reporting Test with NAM 3.1 Login and Logout

uatgateway.wponet.com (verified)

Estimated transfer: 1,062 KB

Duration: 5.939 s

Highlighted words are LoadStorm.com "Keyword" actions that can be performed within the test script. Actions include "Open", "Process Form", "Click Link"

#	Step	Transfer	Duration
1	Open <a href="https://ids.mvaapplication.com/nido/idff/sso">https://ids.mvaapplication.com/nido/idff/sso</a>	34 KB	1.746 s
2	Process form This step will parse the form and enter username and password	11 KB	1.601 s
3	Open <a href="http://www.myapplication.com/target-url">http://www.myapplication.com/target-url</a>	455 KB	12.155 s
4	Open &twp_ Open any additional URL here	289 KB	10.275 s
5	Process form Process any other forms here	185 KB	2.220 s
6	Open <a href="http://www.myapplication.com/?cmd=logout">http://www.myapplication.com/?cmd=logout</a>	87 KB	3.453 s

ADD STEP REORDER STEPS

Open the URL to tear down the session so user can be re-used

### A Look inside the "Process Form" Step which reads in "Ecom\_User\_ID" (userID) and "Ecom\_Password" (password)

Home Build Run Analyze

Build > Test Name Scenario Name Step #

### Process Form

Form 1

This form submits to the URL "http://www.myapplication.com/step/login" using the POST method.

Fields parsed by LoadStorm's "Process Form" action

**Ecom\_User\_ID**   
 userid Source: constant | form data

**Ecom\_Password**   
 password Source: constant | form data

name=button, value=Login  
 Submit form without button click  
 Go to next step without submit

Save

From where will the user accounts and passwords be read? Use "constant" for testing scenarios with individual users, and "form data" for bulk load tests

## Running the Tests

LoadStorm runs scenario test steps dynamically as you build your test. Each preceding step must be run successfully before the next can be executed. If there are any errors such as unreachable URL's or forms that cannot be processed, LoadStorm will provide this information in the script build page. Since our internal LoadStorm NAM tests include URL's and data we don't necessarily want to share in *this Cool Solution*, let's run the "CIA" test case and see what displays and how it displays.

To Run a LoadStorm Test:

1. From inside your LoadStorm account page, click the "Build" tab
2. Underneath "Plan Name", click the "CIA Factbook on S3 (example) link
3. Click the "Run" tab
4. Click the "ADD LOAD TEST" button

5. Choose the Test Plan you want, and configure the load test:

## New Load Test

Plan:

CIA Factbook on S3 (example) ▼

Plan Name

Start:



When should the test start?  
Can schedule for any date/  
time in the future, or "Start  
ASAP"

(YYYY-MM-DD hh:mm AM/PM)

This date/time should be at least 15 minutes in the future to allow the system time to prepare resources. Multiple tests may be scheduled, but may not run concurrently and should have at least 5 minutes between them.

Start load test as soon as possible

Description:

Brief description and should be < 255 characters

Pattern:

linear ▼

How will users be fed to the test? continually in linear increments, or stepped in numbers over a set interval? For our NAM tests, we used "linear"

Expected duration:

20 (max 60 minutes)

How long will the test run? After the first loop is completed through your list of users, the test will continually loop through the list for this number of minutes

Start users:

500

Peak users:

500 (max 500 VUsers)

BUY MORE VUSERS

Step time (unused):

minutes

Start and peak users. Start users are useful when "stepping" up user numbers over time. For our NAM test, we used a full 500 users at all at once, with a 500 user peak". The last two: "Step time" and "Step increase" are not necessary unless you are using the "step up" Pattern

Step increase (unused):

[Cancel](#)

6. Click the "Save" button

7. The "Load Test Run" screen will display, showing when the test will start and what the test parameters are:

[Run](#) > CIA Factbook on S3 (example)



Load Test saved, and will start in 2 minutes

## Load Test

**EDIT**

**Plan:** [CIA Factbook on S3 \(example\)](#)

**Servers:** [CIA Factbook on Amazon S3](#)

**Scheduled Start:** 28 seconds

**Expected Duration:** 5 minutes

**Description:** Quick test

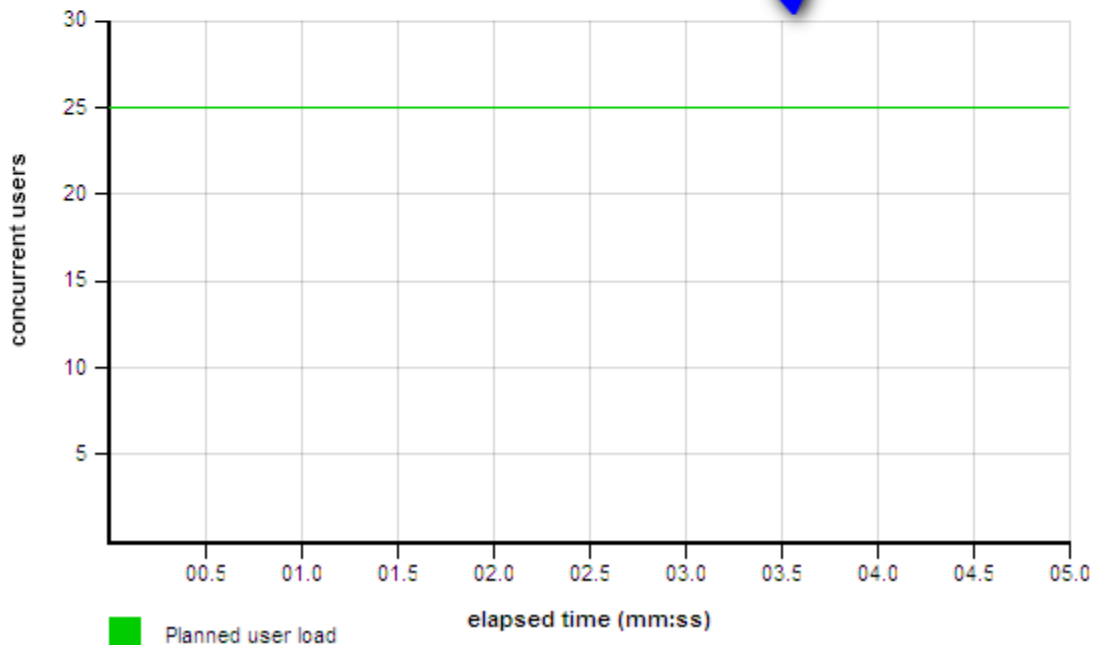
**Pattern:** linear

**Expected load:** starting at 25 users, finishing at 25 users

Nazme of plan

Which server(s) are included

Graph populates in real time when test runs



8. To view completed tests, click the "Analyze" tab

9. The “Analyze” tab stores a permanent record of all tests runs and their results. Here is a sample from our account:

Home Build Run **Analyze** Help

Analyze Load Tests

	Completed	Actual Peak Users	Peak Requests/min	Total KBs Transferred	
<b>DETAILS</b>	2010-05-26 02:22 PM	25	686	36,851	CIA Factbook on S3 (example) Quick test
<b>DETAILS</b>	2010-05-18 11:36 AM	194	3774	699,717	NAM 3.0 to 3.1 post-upgrade load test

10. Let’s open the “CIA Factbook on S3 (example)” Test. Click the “DETAILS” link in the first row (pictured above)

[Analyze](#) > CIA Factbook on S3 (example)

**Plan:** [CIA Factbook on S3 \(example\)](#)

**Servers:** [CIA Factbook on Amazon S3](#)

**Started at:** May 26, 2010 at 02:17PM

**Expected Duration:** 5 minutes

**Description:** Quick test

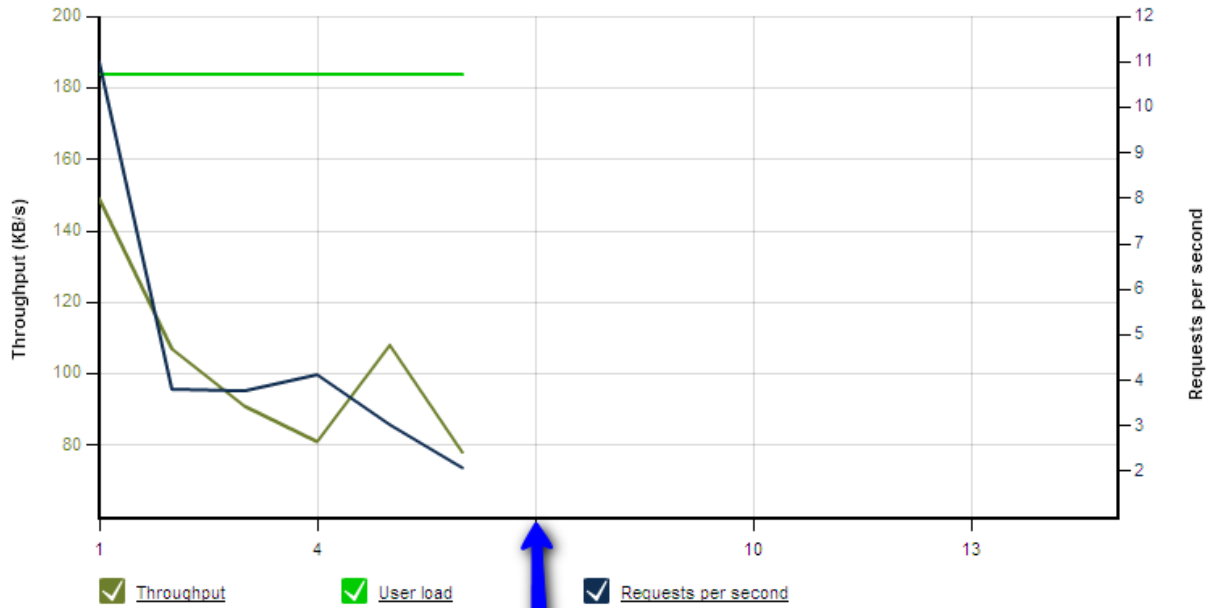
Total requests

**Pattern:** linear

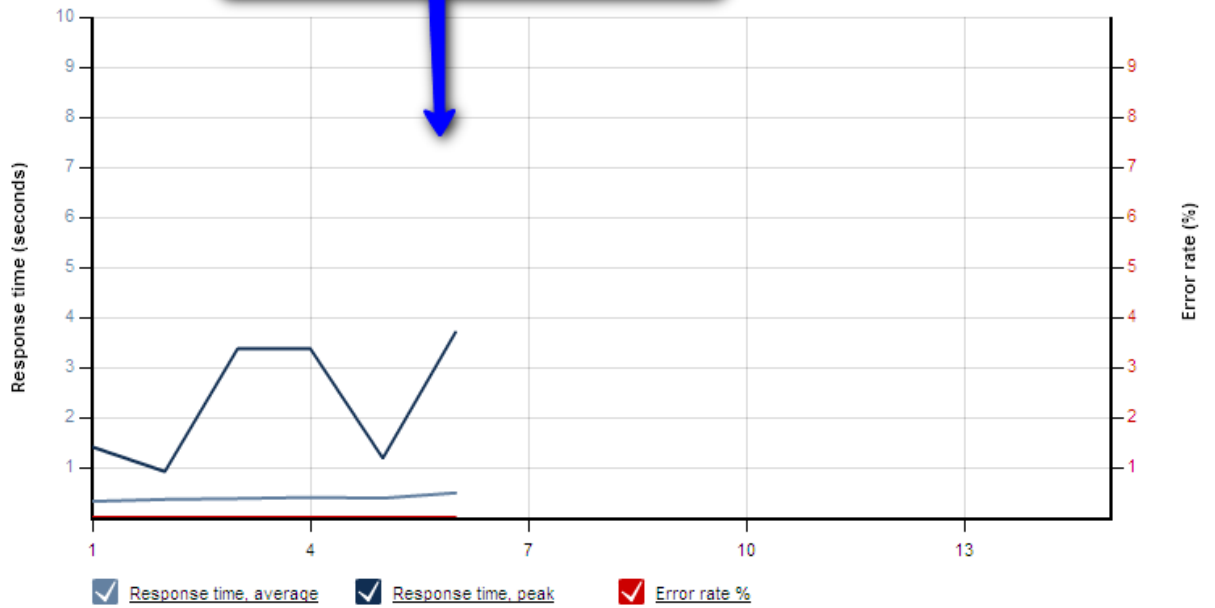
**Expected load:** starting at 25 users, finishing at 25 users

Average response time throughout the entire test cycle

	Response (average)	Errors	Requests	RPS (average)	RPS (peak)	Throughput (average)	Throughput (peak)	Total Transfer
HTML	0.511	0	418	1	1	55	71	19 MB
Other *	0.322	0	1,274	4	11	47	115	17 MB
<b>Total</b>	0.368	0	1,692	5	11	102	149	36 MB



Response and throughput graphs



## Requests by response time

Resource	Requests	Average size	Average response time	Max response time
/geos/in.html	1	128,379 bytes	3.703 s	3.703 s
/print/vq.html	1	42,329 bytes	3.378 s	3.378 s
/print/jn.html	1	17,345 bytes	3.346 s	3.346 s
/fields/2020.html	1	90,472 bytes	1.666 s	1.666 s
/fields/2116.html	1	370,190 bytes	1.038 s	1.038 s
/geos/ro.html	1	118,097 bytes	1.008 s	1.008 s

Requests by response time, size and resource (click the "Requests by response time" link)

## Interpreting Test Results

Let's examine a "Response time" graph taken from a live ERP Time Reporting test we ran recently. This test was run against one of the sandbox servers and used the basic NAM login / logout test discussed previously, but with a couple of additional steps.



[Show requests by elapsed time](#) [Show requests by error code](#) [Show requests by response time](#)

The dark blue line is showing the "Peak Response Time" which means that it measures the slowest responding transaction during that minute interval. This line highlights your outliers of poor

performance. The light blue line is showing the “Average Response Time” and will almost always be much lower than the peak response time because your system will respond quickly to some requests. For example, your login page may be slow due to processing bottlenecks at the database, while a small static image should be much faster since it requires no extra processing for the web server to deliver. Generally, database access is one of the common performance offenders in a load test.

The graph displays a direct correlation between the Average Response Time and the Error Rate. We can see the Peak Response Times hit the 35 second threshold at about the 40 minute mark as our system starts to take longer to respond.

By clicking on the “Show requests by error code” link above, we can look at a couple of URL’s that failed a large number of times: Our application login page and the Novel Identity Server redirect page:



408	Application Login Page - # of errors and error code	26	33.76 s	35.0 s
408	Identity Server redirect page	97	34.435 s	35.0 s

At a high level, the graphic tells us that users trying to log in again much later in a test are waiting a very long time to receive login pages (35 seconds or more.) Either the NAM session isn’t being correctly torn down, or our NAM sandbox instance is being staggered by load. We will investigate further and re-run the tests after any configuration issues are mitigated.

## LoadStorm itself is a Load Tool, Not a Diagnostic Tool – But, it can definitely help with the diagnosis or find issues early before your production users do...

When analyzing LoadStorm test results (always available on LoadStorm’s “Analysis” tab,) it may be tempting to view LoadStorm or any load testing platform as a diagnostic tool. LoadStorm does not directly fulfill this role, though when used correctly can provide extensive information to help diagnose and trouble various NAM and application bottlenecks. In our case, we made a number of policy tweaks and graphic size reductions based on errors we encountered during the load tests. Natively, LoadStorm provides key data such as URL’s, response times and HTTP Error Codes (most often related to the 400 series of codes), and the data can be downloaded from LoadStorm.com as a CSV file. However, the data collected from LoadStorm, when used in conjunction with web/application server and Novell Identity Server Logging can help pinpoint key bottlenecks in the system and identify potential throughput problems before you flip the switch to go live in production.

## Conclusion

This *Cool Solution* has just scratched the surface of what is possible with external load testing tools such as LoadStorm. When load testing a NAM environment, be sure to consider estimated and actual peak users, while simulating real-world authentication and authorization patterns. There is no need to throw 5000 concurrent users at a system just because you can, or have a handy tool supports it. Testing 5000

concurrent authentication actions for duration of 45 – 60 minutes can literally push 20,000 or more total transactions through your system within the test duration period. This level of concurrency is a very large number represents uncommon load levels for internally-facing intranet applications. However, if you *do* need this level of concurrency or higher, LoadStorm can “handle the load” nicely.

Happy testing!

For additional information, please visit:

- The Washington Post Company: <http://www.washpostco.com>
- Novell: <http://www.novell.com>
- LoadStorm: <http://www.loadstorm.com>