

Getting Started with BlazeMeter

Lab Guide

88BLZ20010

Table of Contents

Lab 1 - Getting Started with BlazeMeter.....	1
Lab 2 - Create a simple URL/API Test & Create a Report.....	5
Lab 3 - Create a Multi Test in BlazeMeter.....	12
Lab 4 - Create, Update, and Run JMeter Tests via. APIs.....	35
Lab 5 - Record and Run a JMeter Test using Chrome Browser Plugin.....	51
Lab 6 - BlazeMeter Jenkins Integration Setup.....	59

Lab 1 – Getting Started with BlazeMeter

Goals Create projects in BlazeMeter.

Scenario Using the BlazeMeter application, you must create multiple projects and see how to switch between different projects.

You will accomplish this by completing the following tasks:

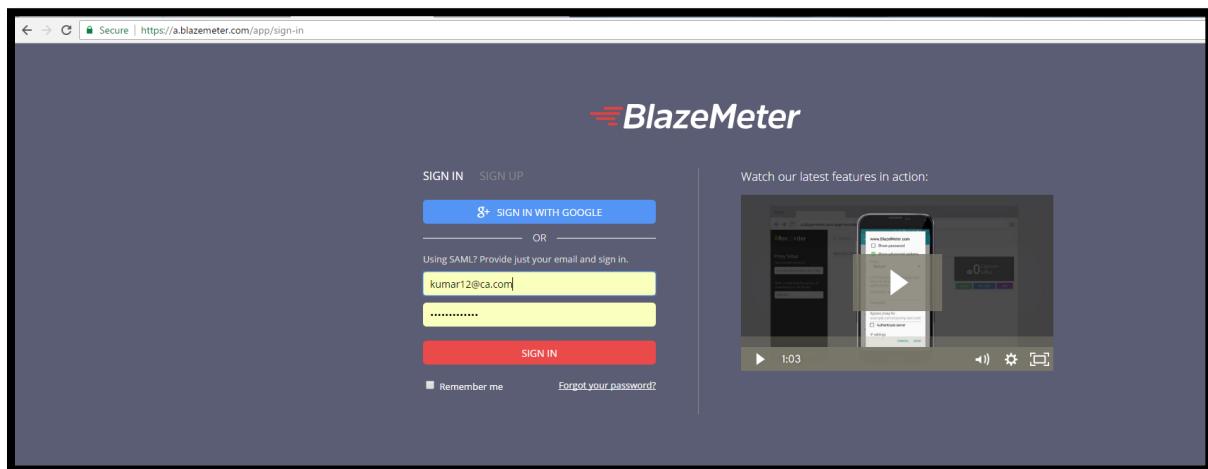
- Log into BlazeMeter
- Create two new projects
- Switch between projects

Time 10 minutes

Instructions:

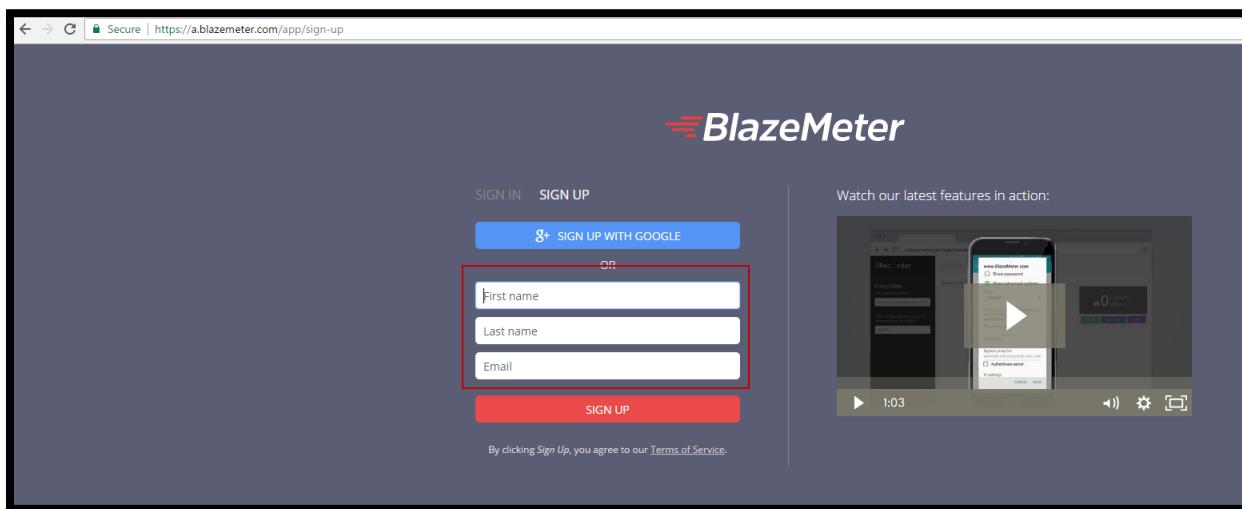
Accessing the BlazeMeter Application

1. Enter <https://a.blazemeter.com/app/sign-in> in your browser.
2. At the login prompt, enter the following details (If you are an existing user):



Field	Value
User Name	Enter your registered user name
Password	Enter a valid password

3. If you are accessing the BlazeMeter for first time, click **SIGN UP** and enter valid information to register with BlazeMeter.

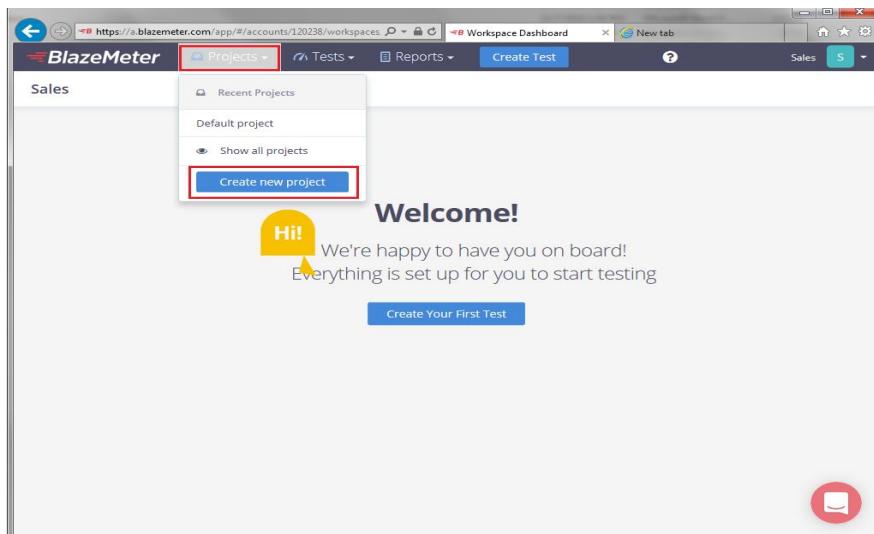


Field	Value
First Name	Enter first name
Last Name	Enter last name
Email	Enter a valid Email address

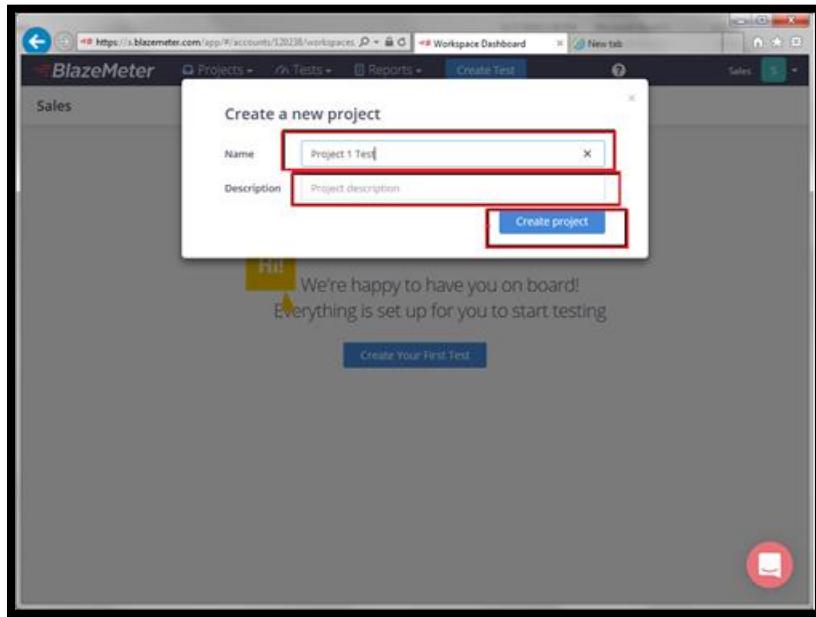
4. On clicking **SIGN IN**, the BlazeMeter application opens.

Creating New Projects

5. From the BlazeMeter UI, click **Projects > Create new project**.



6. In the Create a new project dialog box, enter **Name** and **Description**. To create a new project, click **Create Project**.

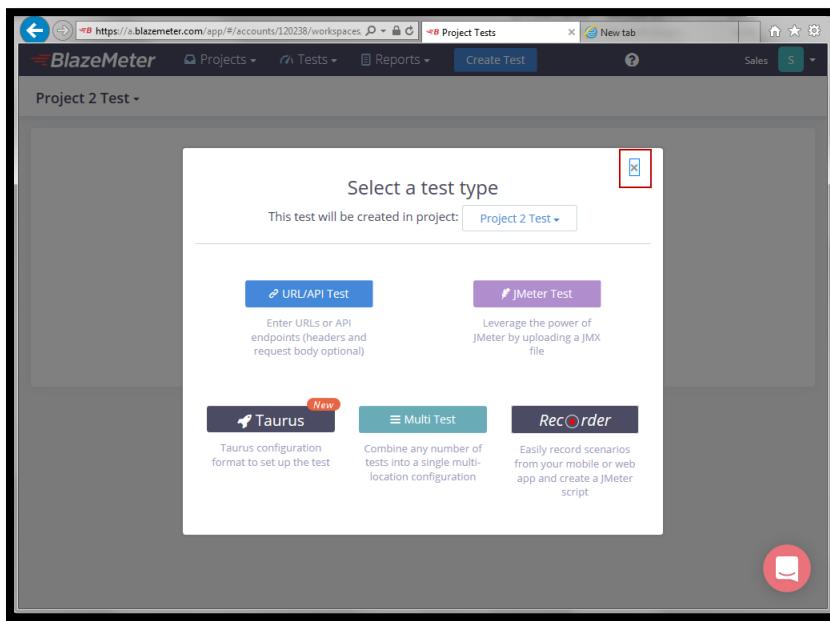


Field	Value
Name	Project 1 Test
Description	Enter description relevant to the project (It is optional)

7. To create another project, click **Projects > Create new project**.
8. In the Create a new project dialog box, enter the Name and Description. Now, click **Create Project**.

Field	Value
Name	Project 2 Test
Description	Enter description relevant to the project (It is optional)

9. From the BlazeMeter UI, click **Create New Test** to view the different tests you can create in BlazeMeter. To close the window, click 'x' button.



10. To switch from one project to another project, click the **Projects** tab in BlazeMeter UI.
11. From the drop-down menu, you can select a project that you created recently and switch between existing projects.
12. To exit the BlazeMeter UI, click **Log Out**.

Lab 2 - Create a simple URL/API Test & Create a Report

Goals

This document provides brief instructions for attendees to create a simple URL/API test from the BlazeMeter UI and create a report.

Scenario

You will accomplish this by completing the following tasks:

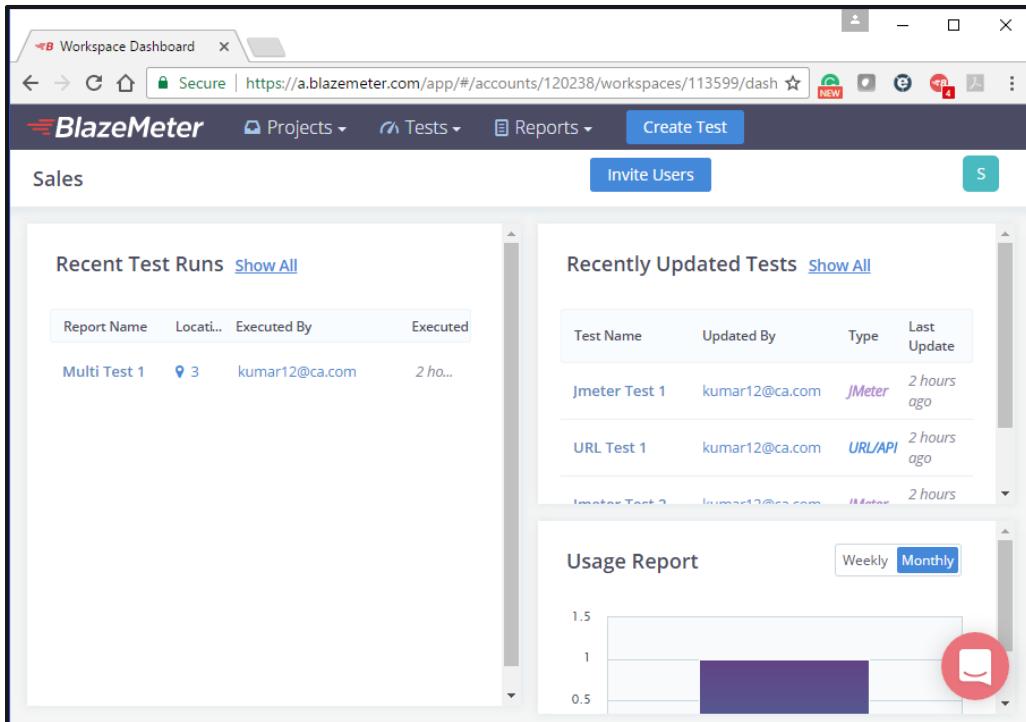
- Create a simple URL/API Test
- Create an Executive Summary Report

Time

20 minutes

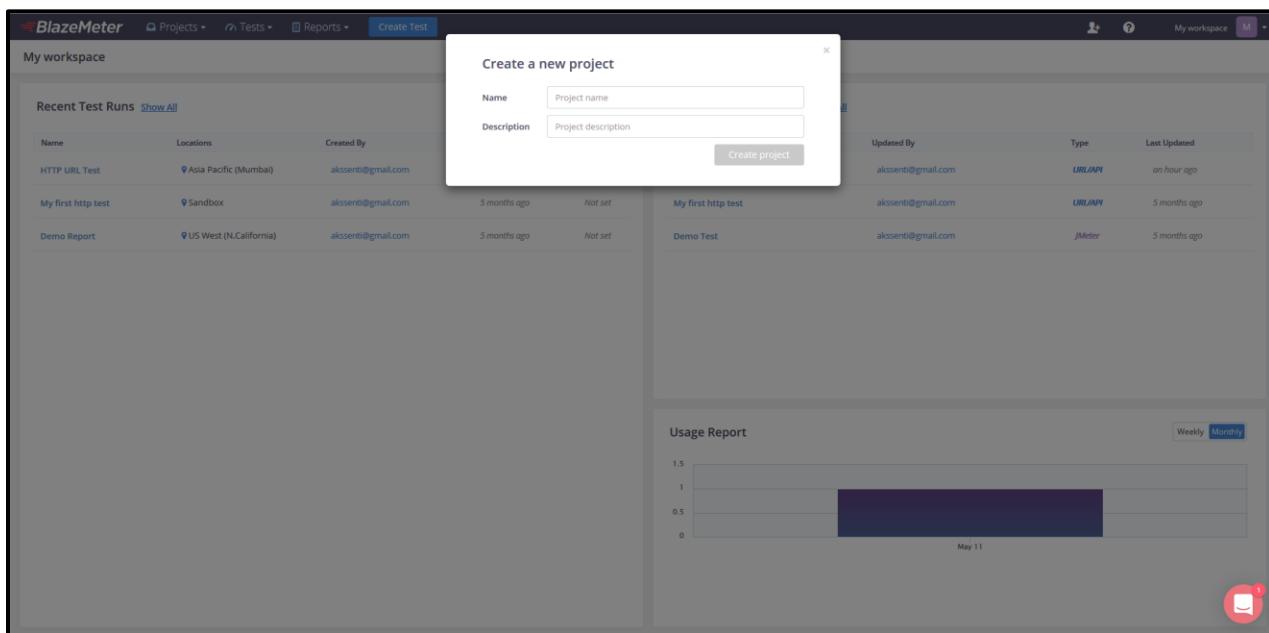
Instructions:**Run a simple URL/API Test and Create a Report**

1. Open an instance of Google Chrome browser and go to <http://www.blazemeter.com>. If you have already created a BlazeMeter account, <http://www.blazemeter.com> URL and your BlazeMeter account email and password to log into BlazeMeter.



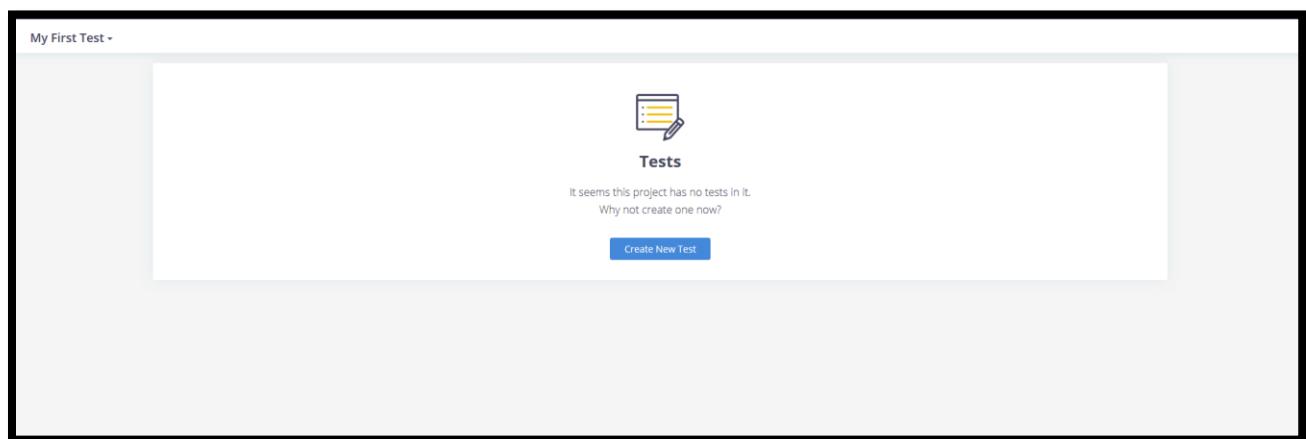
The screenshot shows the BlazeMeter workspace dashboard. At the top, there's a header with the BlazeMeter logo, navigation links for 'Projects', 'Tests', 'Reports', and 'Create Test', and a 'Create Test' button. Below the header, the workspace name 'Sales' is displayed, along with a 'Invite Users' button and a search icon. The main area is divided into two sections: 'Recent Test Runs' and 'Recently Updated Tests'. The 'Recent Test Runs' section shows a table with one entry: 'Multi Test 1' located at '3' locations, executed by 'kumar12@ca.com' 2 hours ago. The 'Recently Updated Tests' section shows a table with three entries: 'Jmeter Test 1' updated by 'kumar12@ca.com' 2 hours ago, 'URL Test 1' updated by 'kumar12@ca.com' 2 hours ago, and 'Inventory Test 2' updated by 'InventoryTest2@ca.com' 2 hours ago. At the bottom of the dashboard, there's a 'Usage Report' section with a chart showing usage over time, with a red circle highlighting a specific data point.

2. Create a new project called “My First Test”.



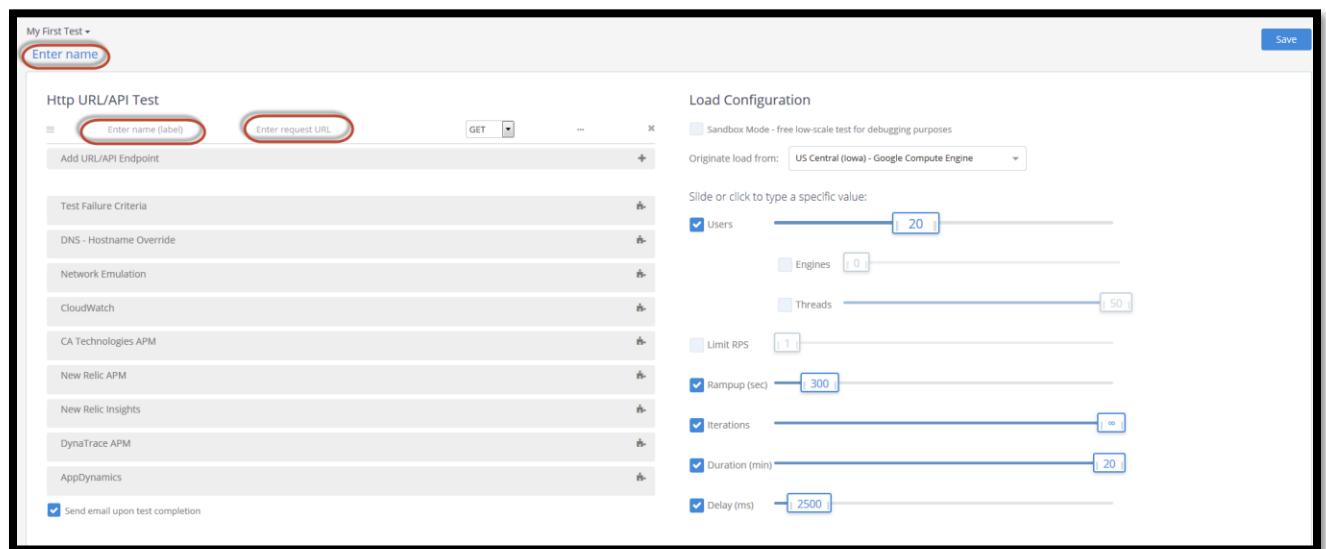
The screenshot shows the BlazeMeter dashboard with a modal window titled "Create a new project". The modal has two input fields: "Project name" and "Project description", both currently empty. Below the fields is a "Create project" button. In the background, there is a table of recent test runs and a "Usage Report" chart. A red notification badge with the number "1" is visible in the bottom right corner of the dashboard area.

3. Click on **Create New Test** button, to create a new test case.



The screenshot shows the "My First Test" project page. The main content area is titled "Tests" and features a message: "It seems this project has no tests in it. Why not create one now?". Below this message is a blue "Create New Test" button. The rest of the page is mostly blank white space.

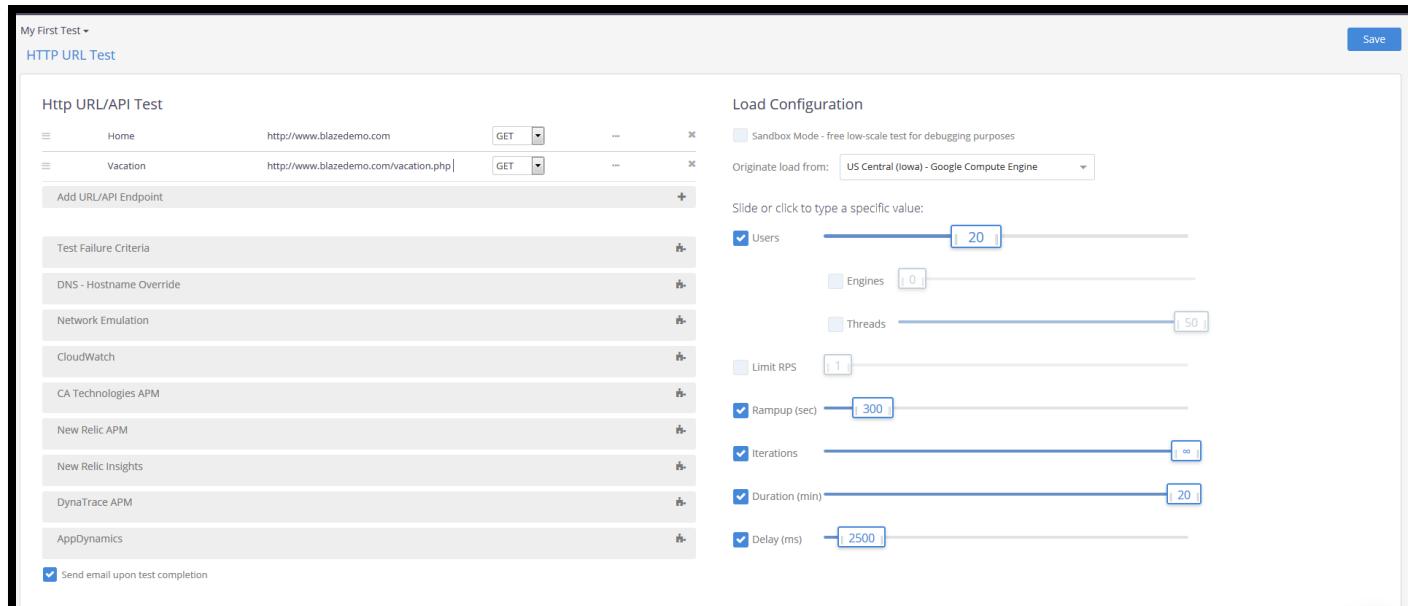
4. Click on **URL/API Test** button, to create a URL/API Test case.



The screenshot shows the 'Http URL/API Test' configuration screen. The 'Enter name' field is highlighted with a red circle. Other fields like 'Enter name (label)', 'Enter request URL', and 'Load Configuration' settings are also visible.

Field	Value
Enter name (Test Case Name)	HTTP URL Test
Enter name (label):	Home
Enter request URL	http://www.blazedemo.com

- a. Click the Add URL/API Endpoint + on right to create another URL/API Endpoint, and enter the data
- Enter name (label): **Vacation**
 - Enter request URL: <http://www.blazedemo.com/vacation.php>



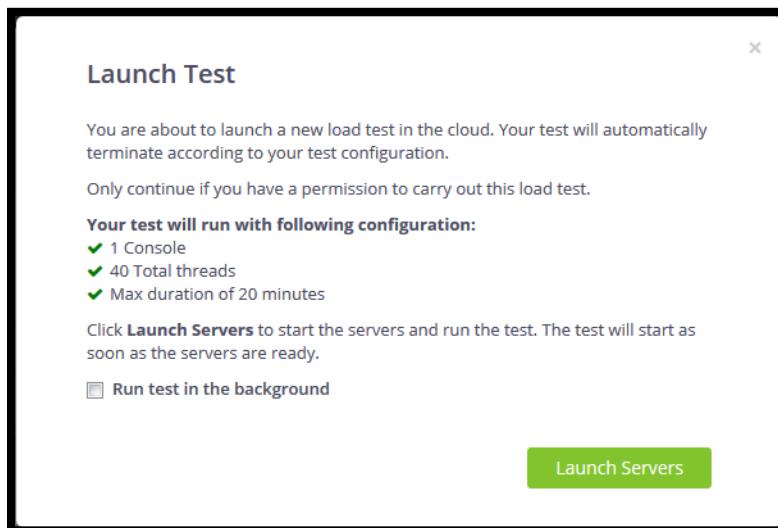
The screenshot shows the 'Http URL/API Test' configuration screen with two endpoints listed: 'Home' and 'Vacation'. The 'Vacation' endpoint has its 'Enter name (label)' set to 'Vacation' and its 'Enter request URL' set to 'http://www.blazedemo.com/vacation.php'. The 'Load Configuration' settings remain the same as in the previous screenshot.

5. Re-configure the Test Case Load Configuration as shown below:
- Select Asia Pacific (Mumbai) – **Amazon Web Services** from the Originate load from drop-down list.
 - Select the Threads check box and enter **40** for 40 Threads or Virtual Users per Engine.
 - Save the Test case using the **Save** button on top right corner.
 - You will now see a green box with an arrow next to your Test Case name.

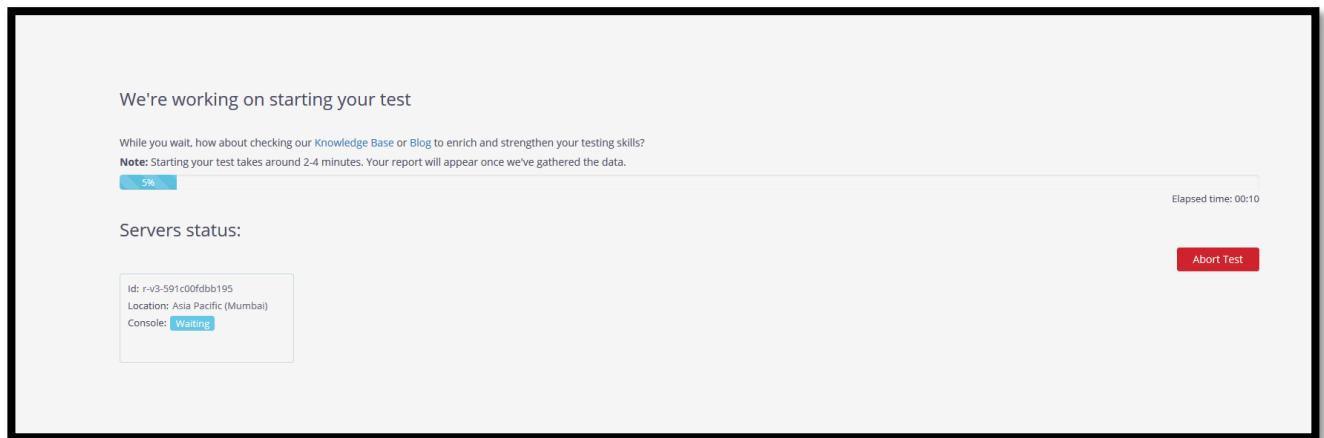
The screenshot shows the 'Http URL/API Test' configuration page. On the left, there's a tree view with nodes like 'Home', 'Vacation', and 'Enter name (label)'. On the right, the 'Load Configuration' panel is displayed. It includes fields for 'Sandbox Mode', 'Originate load from' (set to 'Asia Pacific (Mumbai) - Amazon Web Services'), 'Users' (set to 40), 'Engines' (set to 0), and 'Threads' (set to 40). Other settings include 'Rampup (sec)', 'Iterations', 'Duration (min)', and 'Delay (ms)'. A checkbox for 'Send email upon test completion' is also present at the bottom.

6. Click on the **Green Arrow** button on top left to start the URL/API test. The following dialog box will be displayed. Click on **Launch Servers** button to launch the servers that will generate the load to execute the URL/API performance test.

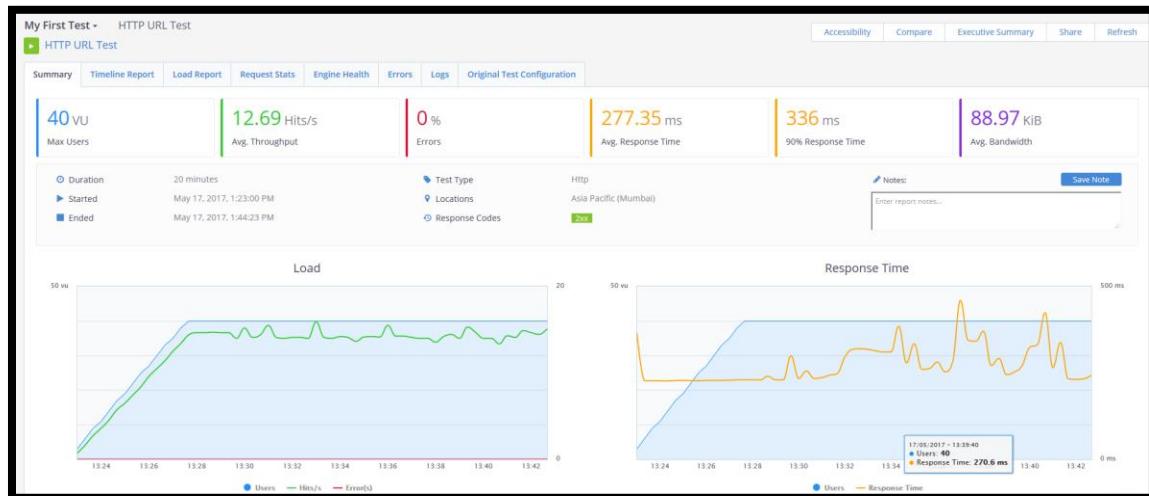
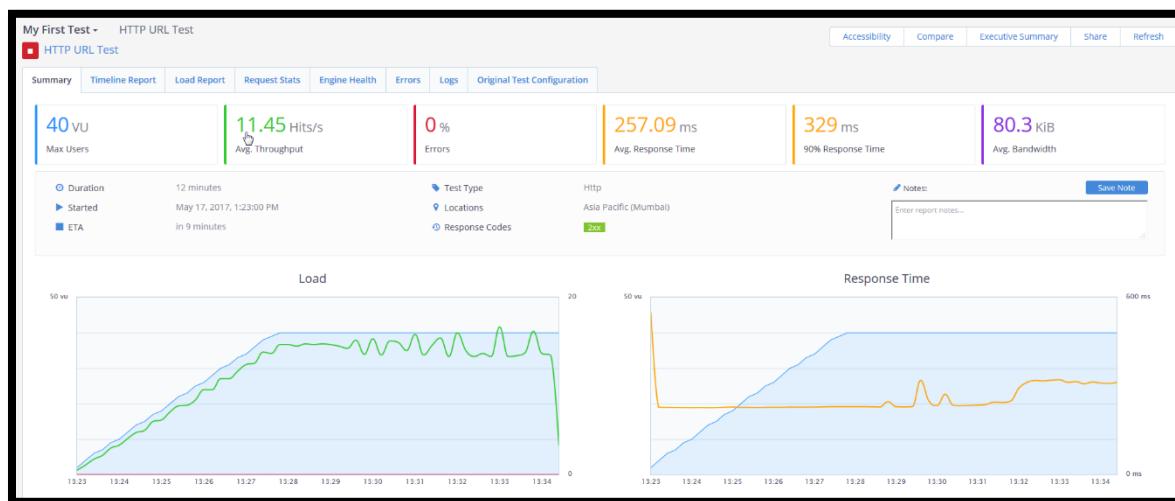
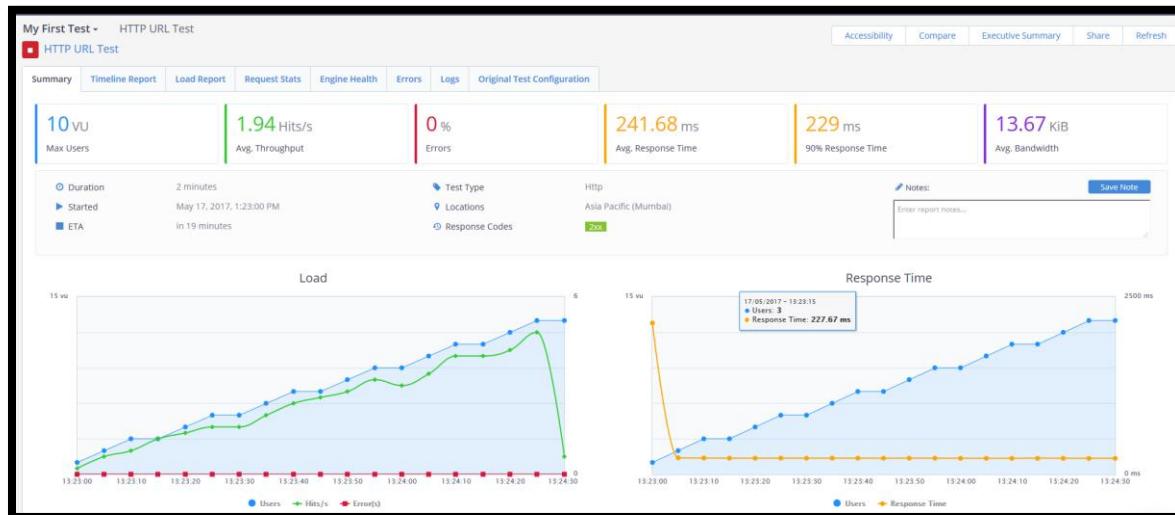
This screenshot shows the 'Test History' tab selected in the navigation bar. It displays a list of tests: 'My First Test', 'HTTP URL Test' (highlighted with a red box), 'Test Configuration', 'Test History', and 'Test Configuration'. Below this, the 'Http URL/API Test' configuration is shown, which is identical to the one in the previous screenshot. The 'Load Configuration' section is visible with its various parameters. A large green arrow icon is located in the bottom right corner of the main configuration area.



7. Observe that Engines and Console are starting.



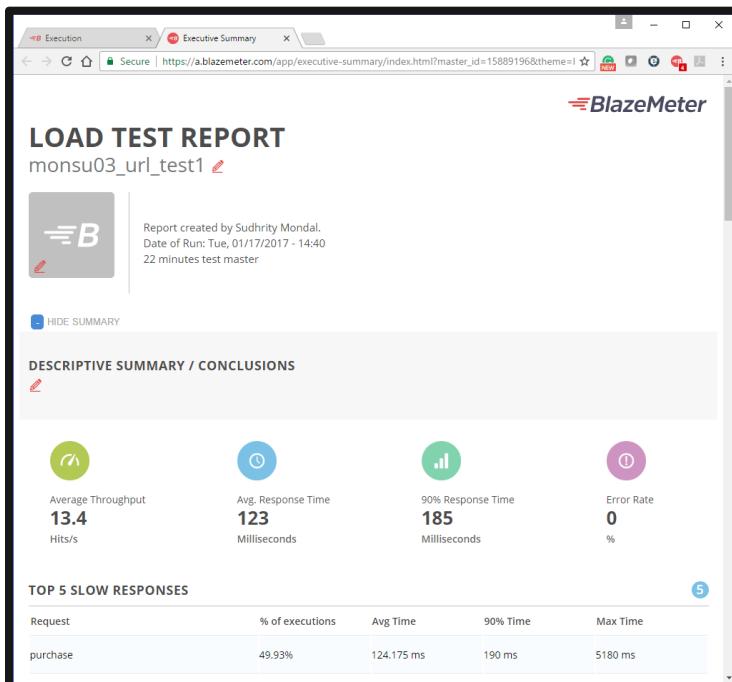
8. Once, the Engines and Consoles have started successfully and are connected, BlazeMeter takes you automatically to the Summary Reports screen as shown above. This report updates in real-time until the test completes in 20 mins. You can see the automatic ramp-up of # of Virtual Users for 300s. After that there is a flat load before the load becomes 0 at the end of the test.



9. While waiting for the test to complete, you can click on different reporting tabs to review other available reports. Description of these reports are available at: <https://guide.blazemeter.com/hc/en-us/articles/206733919-Timeline-Report-Timeline-Report>



10. After the test is complete, click on **Executive Summary** button to view the Executive Summary report. A new browser tab opens with the Executive Summary report.



11. You can click on the pencil icon next to the Test Case name to update the Test name to reflect your enterprise/customer's name. You can also upload a logo of your customer to personalize this report.
12. Right click anywhere on the page and select **Print** to save the report as a pdf document (*supported by Chrome Browser*).

Lab 3 – Create a Multi Test in BlazeMeter

Goals Create a Multi Test in BlazeMeter

Scenario The Multi Test is used for distributed load testing. Using BlazeMeter application, you will create a multi test that contains JMeter tests and a API/URL test.

The Multi Test synchronizes and aggregates the results from several test sessions into a single aggregated report. The Multi test configuration includes individual test configurations orchestrated to run simultaneously. Each test is still individual and can be run independently, regardless of the Multi Test.

You will accomplish this by completing the following tasks:

- Log into BlazeMeter
- Create JMeter tests
- Create API/URL test
- Create a Multi Test
- Run a Multi Test

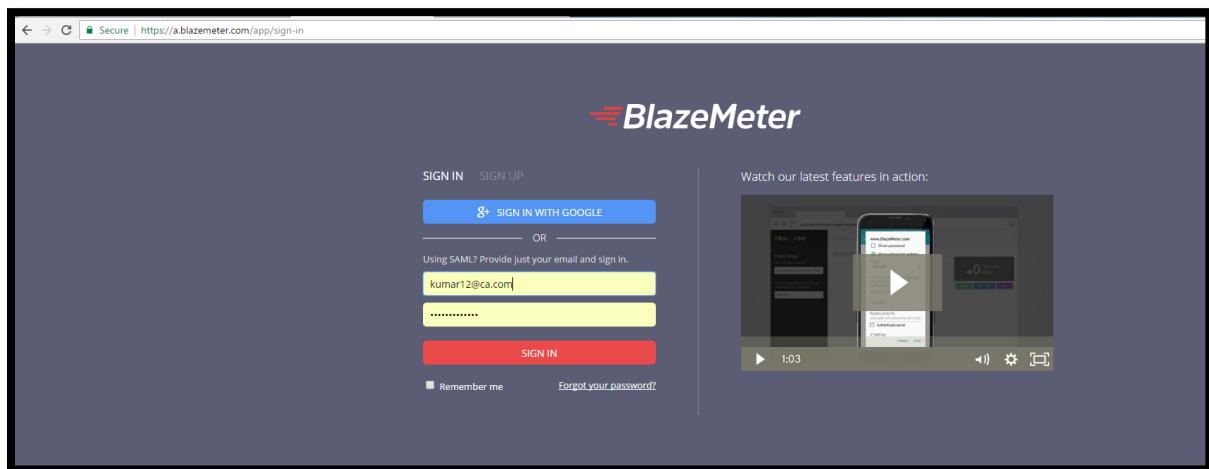
Time 20 minutes

Instructions:

Note: You cannot perform this lab if you have a free-tier account on BlazeMeter. Please watch the associated demonstration to see how this activity can be performed on BlazeMeter

Accessing the BlazeMeter Application

1. Enter <https://a.blazemeter.com/app/sign-in> in your browser.
2. At the login prompt, enter the following details (If you are an existing user):

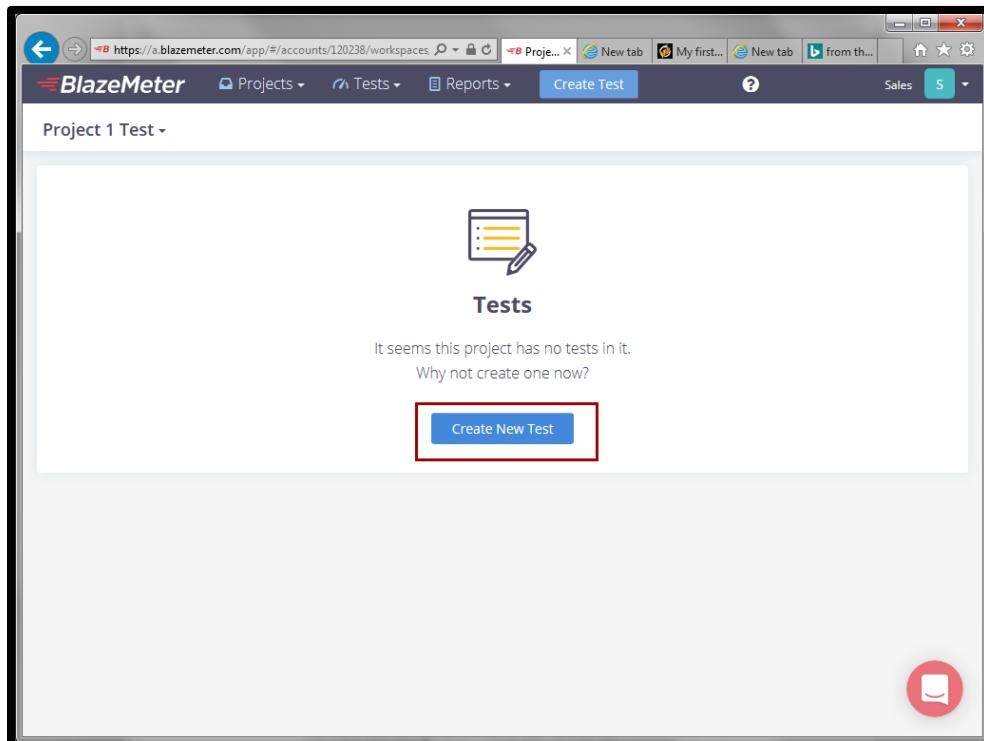


Field	Value
User Name	Enter your registered user name
Password	Enter a valid password

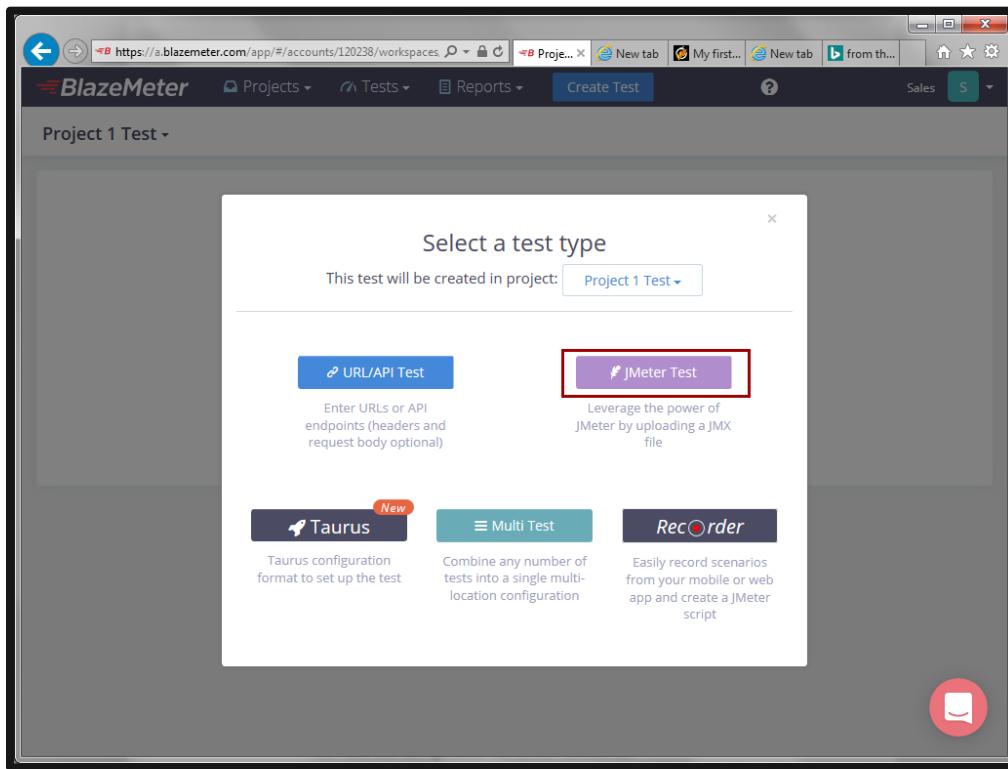
On clicking **SIGN IN**, the BlazeMeter application opens.

Creating a JMeter Test

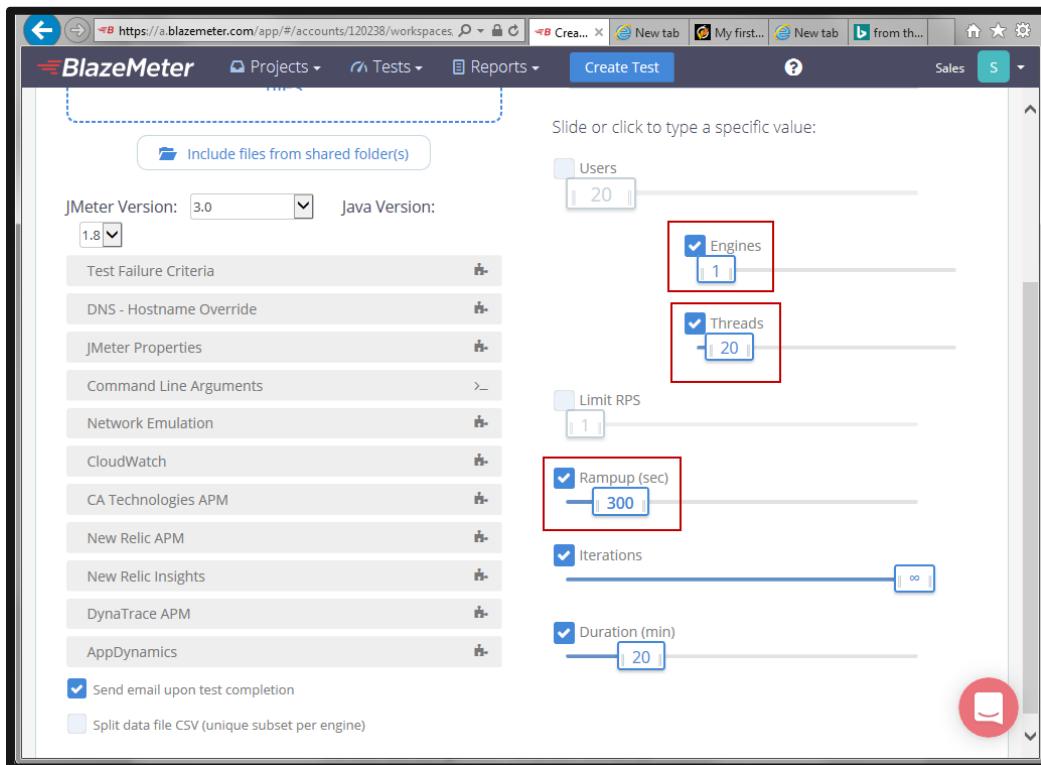
3. From the BlazeMeter UI, click **Create New Test**.



4. In the Select a test type dialog box, click **JMeter Test** to create a JMeter test



5. The JMeter Test UI is displayed. Select **Engines**, **Threads**, and **Rampup (Sec)**. Enter appropriate values in respective text boxes.

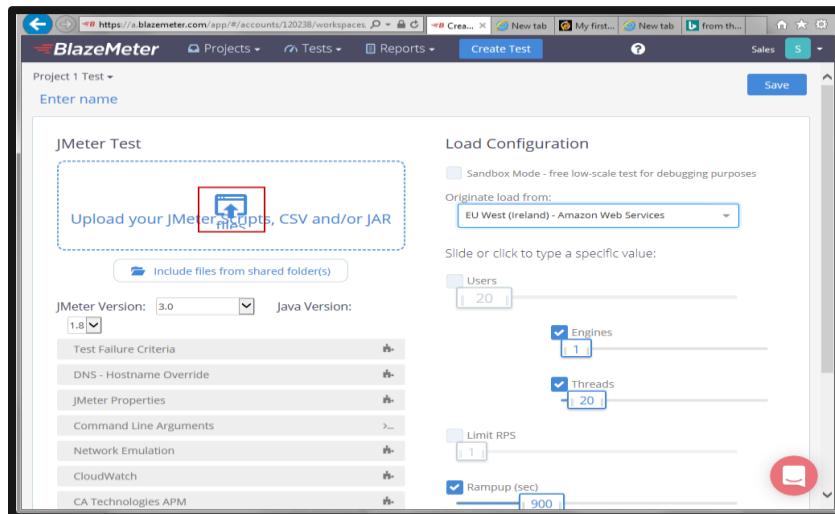


Field	Value
Engines	Manually enter the number of engines that you want to run. Enter '1'.
Threads	Manually enter the number of threads that must run per engine. Enter '20'
Rampup	Select how fast you want the test to ramp up. Enter '300'.

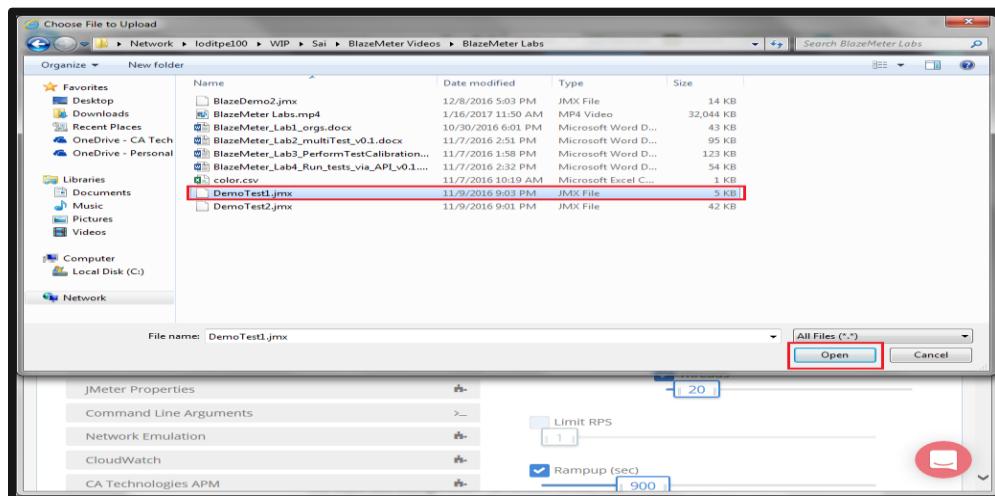
6. Under the Load Configuration section, you have the option to select the Location that you want your test to be run in. From the drop-down, select **EU West (Ireland)**.

The screenshot shows the BlazeMeter web interface for creating a JMeter test. On the left, there's a form for 'JMeter Test' with fields like 'Upload your JMeter scripts, CSV and/or JAR' (with a file upload icon), 'JMeter Version' (set to 3.0), 'Java Version' (set to 1.8), and various configuration sections like 'Test Failure Criteria', 'DNS - Hostname Override', 'JMeter Properties', etc. On the right, the 'Load Configuration' panel is open, showing a dropdown menu titled 'Originate load from:' with a list of locations. The 'EU West (Ireland)' option is highlighted with a red box. Other options in the list include 'Sandbox Mode - free low-scale test for debugging purposes', 'Sandbox (shared)', 'Amazon Web Services', 'US East (Virginia)', 'US East (Ohio)', 'US West (N.California)', 'US West (Oregon)', 'Canada (Central)', 'EU West (London)', 'EU Central (Frankfurt)', 'Japan (Tokyo)', and 'Asia Pacific (Mumbai)'. A red circle with a white exclamation mark is visible in the bottom right corner of the configuration panel.

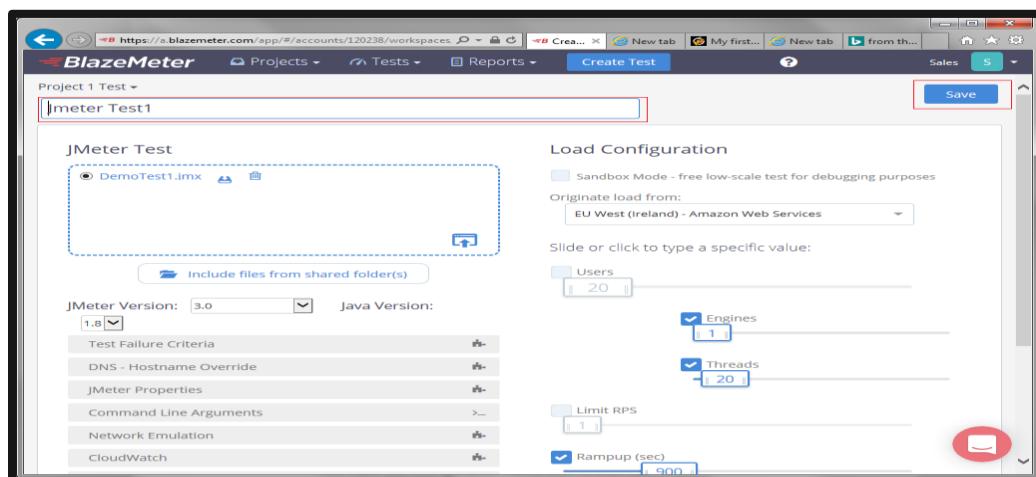
7. In the JMeter Test UI, you will find a central location where you can upload the JMX script saved on your local system. Click the **Upload** icon.



8. From the **Choose File to Upload** dialog box, select the **DemoTest1.jmx** file and click **Open** to add the JMeter script.

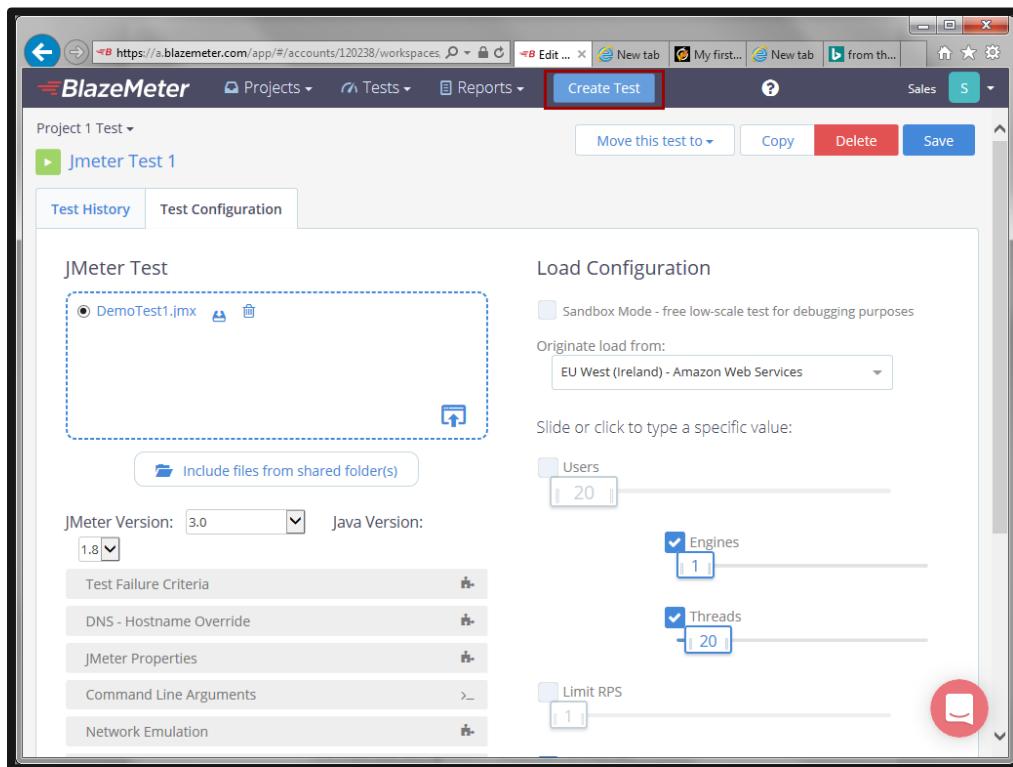


9. Enter **Jmeter Test 1** in the name text box. Click **Save** to ensure that test configuration details are saved and the test is available for execution

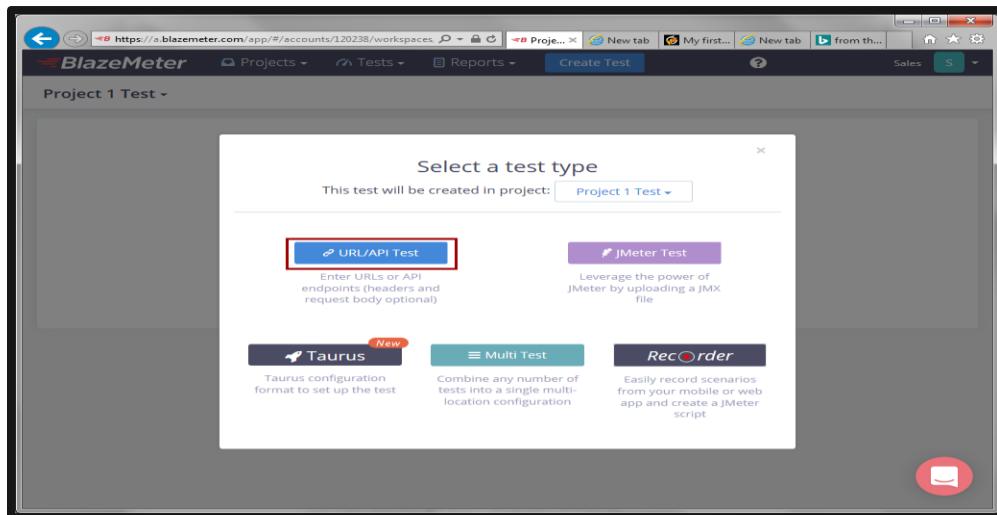


Creating a API/URL Test

- To create a URL/API test, click Create Test from the JMeter Test UI



- In the Select a test type dialog box, click URL/API Test.



- In HTTP URL/API Test UI, enter the Label name and Request URL. Select Engines, Threads and enter values in respective text boxes.

Http URL/API Test

Label 1: blazemeter.com

Method: GET

Load Configuration

- Sandbox Mode - free low-scale test for debugging purposes
- Originate load from: US Central (Iowa) - Google Compute Engine
- Slide or click to type a specific value:
- Users: 20
- Engines: 1
- Threads: 5
- Limit RPS: 1
- Rampup (sec): 300

Field	Value
Label name	Enter Label 1. Label name is entered to filter the report.
Request URL	Enter the complete URL that you want to test. Enter 'blazemeter.com'
Engines	Manually enter the number of engines that you want to run. Enter '1'.
Threads	Manually enter the number of threads that must run per engine. Enter "5"

13. In HTTP URL/API Test UI, Select **Rampup (Sec)**, **Duration** and enter values in respective text boxes.

Http URL/API Test

Label 1: blazemeter.com

Method: GET

Load Configuration

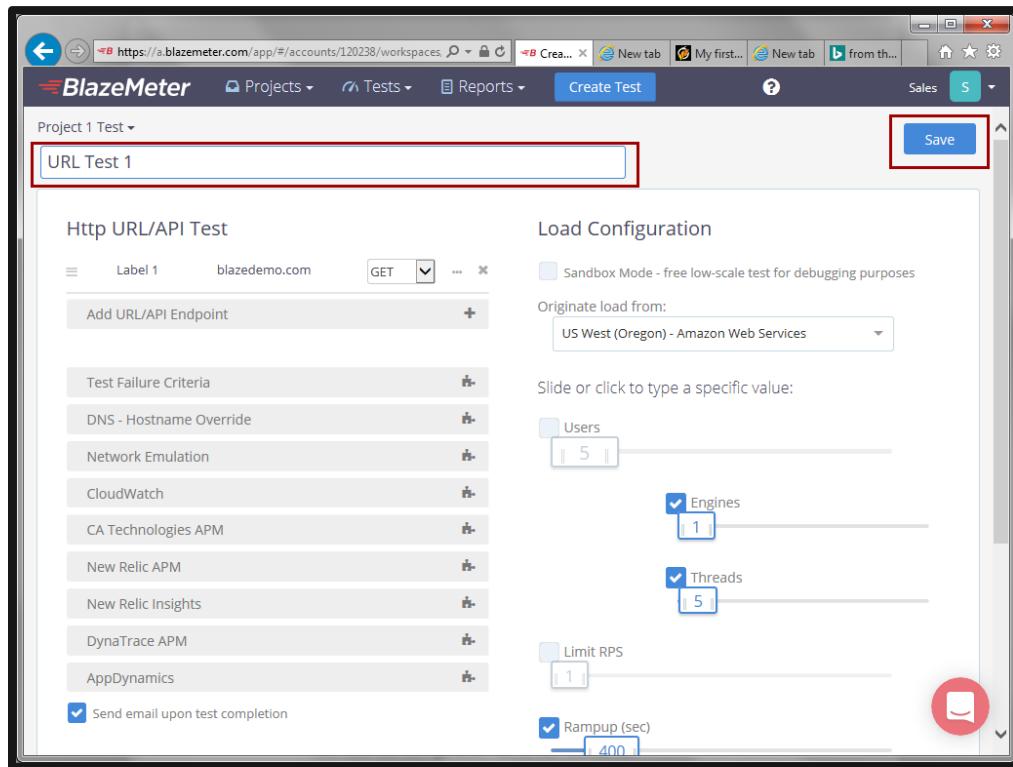
- Users: 5
- Engines: 1
- Threads: 5
- Limit RPS: 1
- Rampup (sec): 400
- Iterations: 100
- Duration (min): 20
- Delay (ms): 2500

Field	Value
Rampup (Sec)	Select how fast you want the test to ramp up. Enter '400'.
Duration	Set the full-time duration for which you want the test to run. Enter '10'.

14. Under the '**Load Configuration**' section, you have the option to select the Location that you want your test to be run in. From the drop-down list, select **US West (Oregon)**.

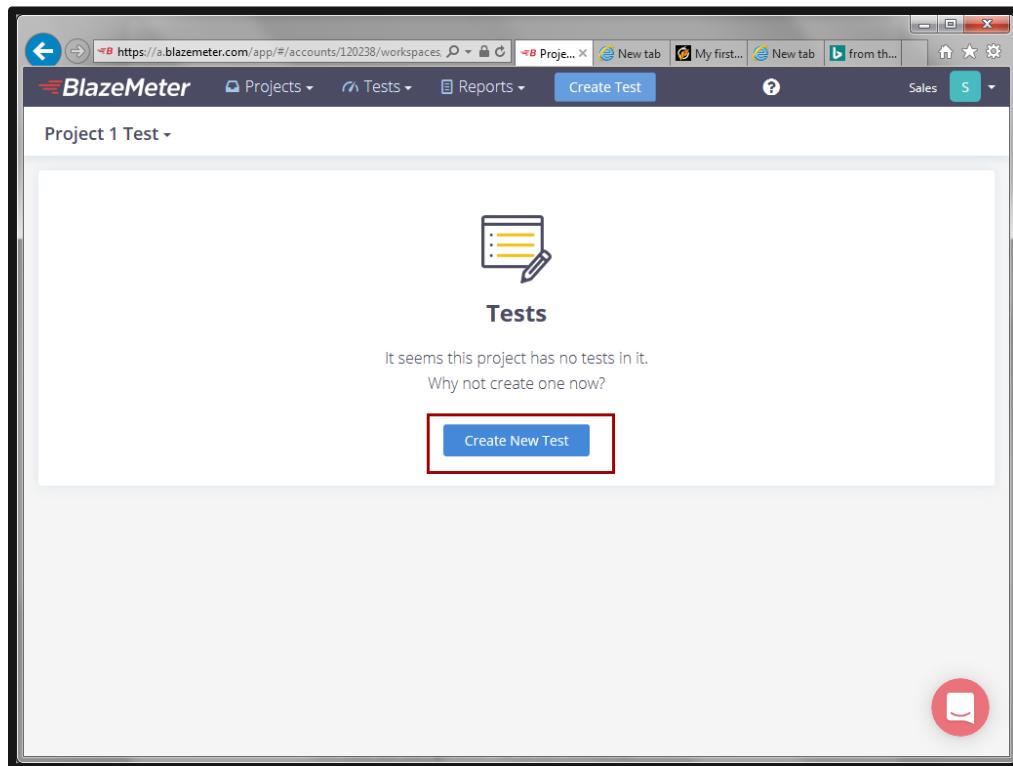
The screenshot shows the BlazeMeter web interface for creating a test. On the left, there's a sidebar with 'Project 1 Test' and a 'Create Test' button. The main area is titled 'Http URL/API Test' and contains fields for 'Label 1' (set to 'blazemeter.com'), 'Method' (set to 'GET'), and an 'Add URL/API Endpoint' button. To the right, under 'Load Configuration', there's a dropdown menu titled 'Originate load from:' with a list of locations. The 'US West (Oregon)' option is highlighted with a red box. Other options include 'Sandbox Mode - free low-scale test for debugging purposes', 'Sandbox (shared)', 'Amazon Web Services', 'US East (Virginia)', 'US East (Ohio)', 'US West (N.California)', 'Canada (Central)', 'EU West (Ireland)', 'EU West (London)', 'EU Central (Frankfurt)', 'Japan (Tokyo)', and 'Asia Pacific (Mumbai)'. A 'Save' button is visible at the top right of the configuration panel.

15. Enter **URL Test 1** in the name text box. Click **Save** to ensure that test configuration details are saved and the test is available for execution

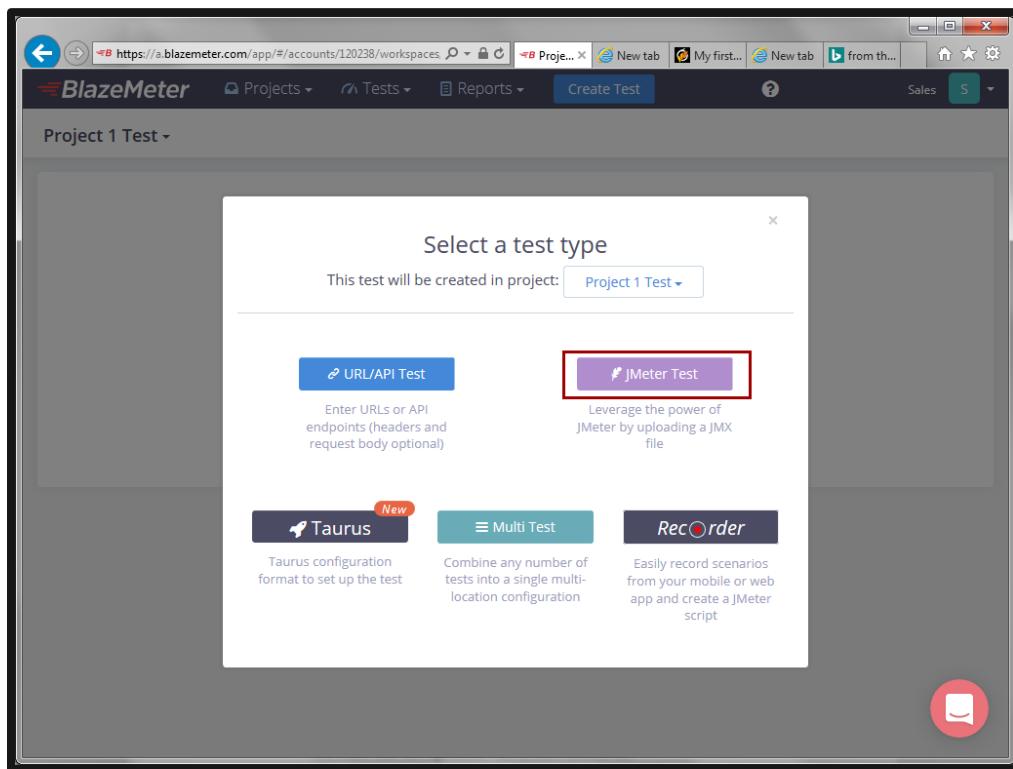


Create another JMeter Test

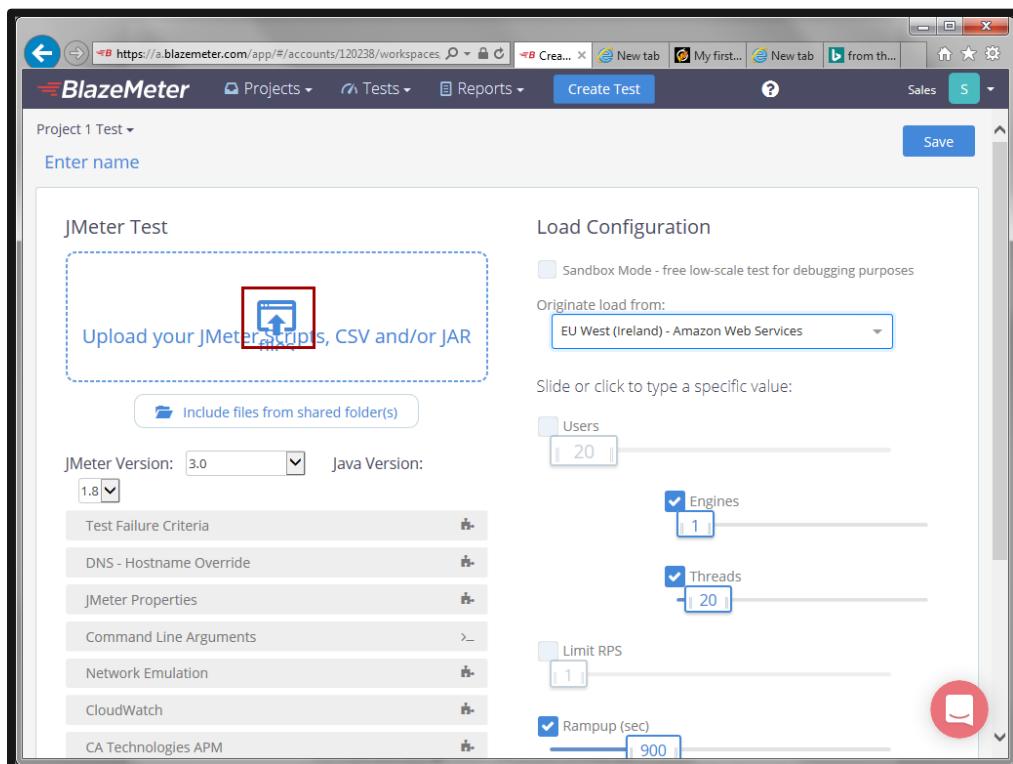
16. To create another JMeter test, click **Create new test** from the BlazeMeter UI.



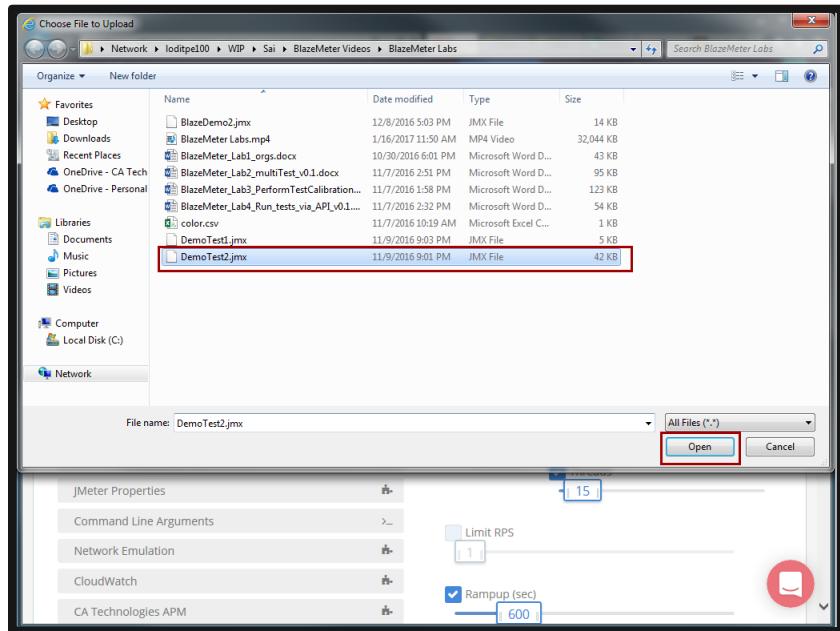
17. In the **Select a test type** dialog box, click **JMeter Test** to create a JMeter test.



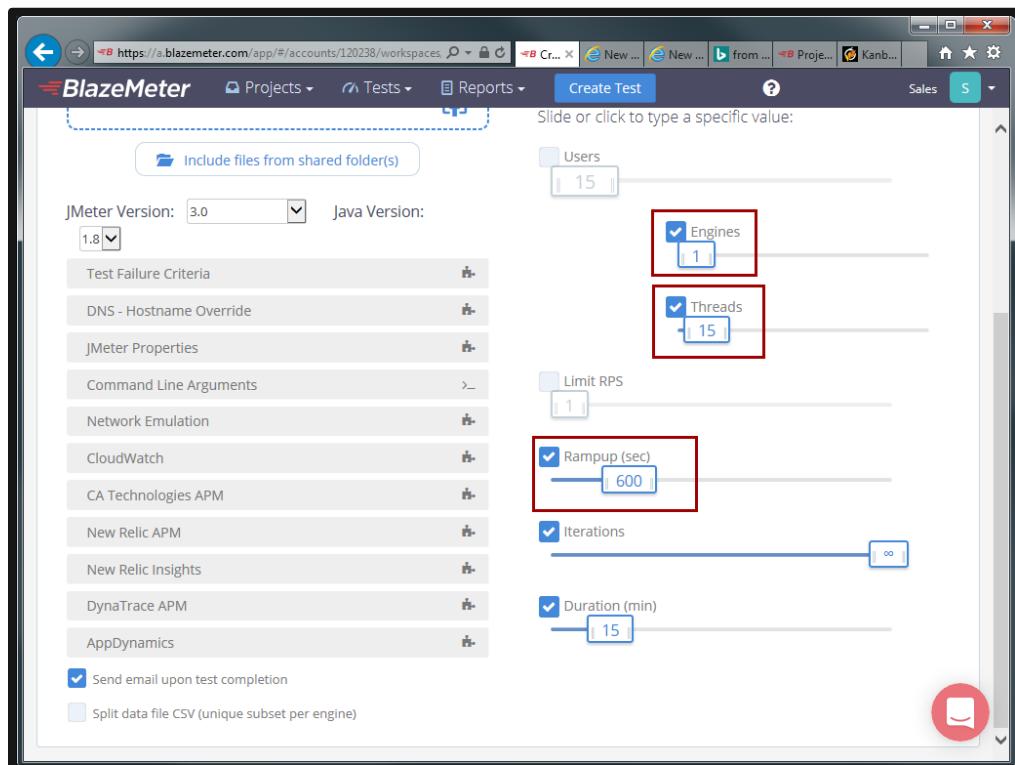
18. Click the **Upload** icon to upload the JMX file.



19. Upload the **DemoTest2.jmx file.**

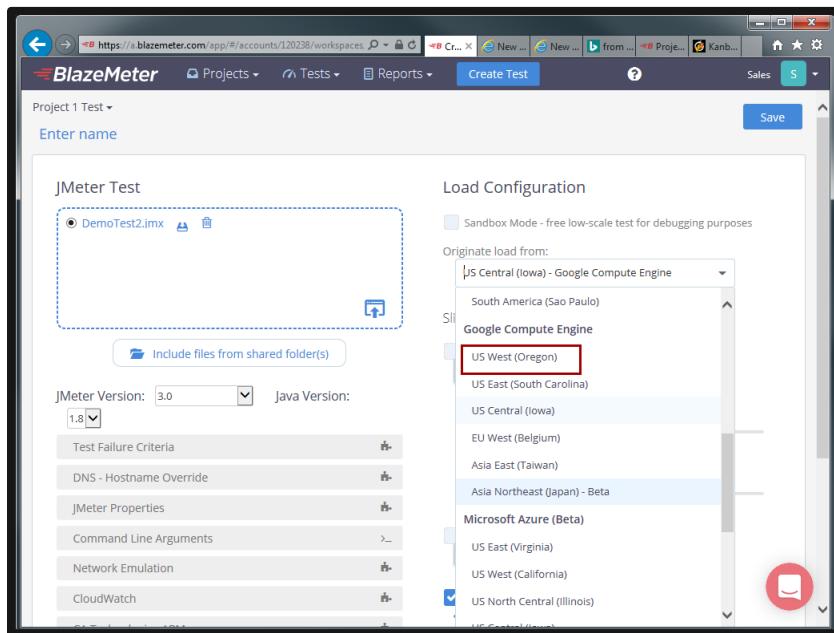


- 20. In the JMeter Test UI window, specify **Engines**, **Threads** and **Rampup (Sec)**. Enter appropriate values in respective text boxes.**

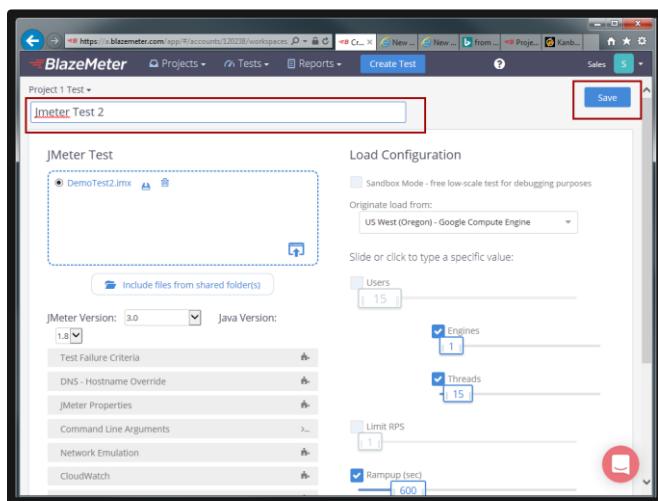


Field	Value
Engines	1
Threads	15
Rampup	600

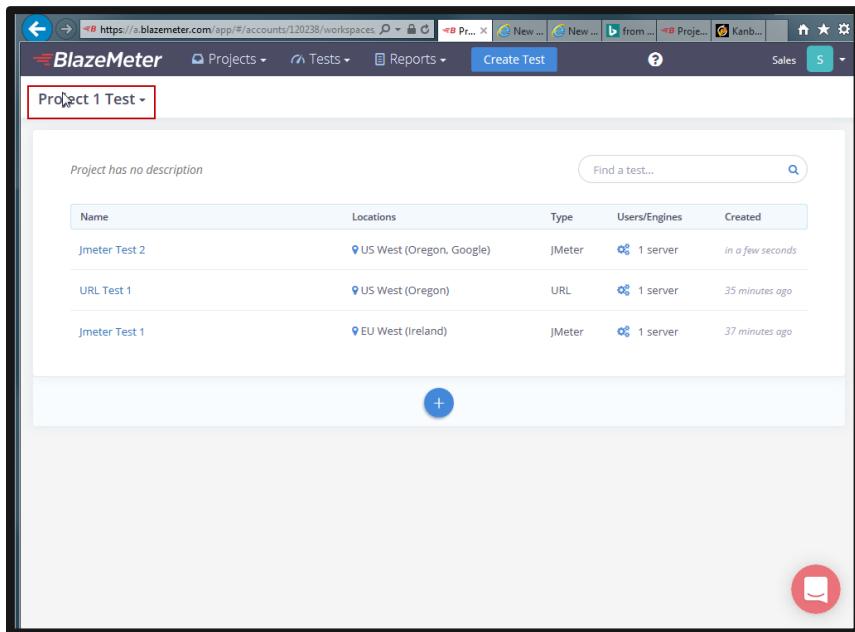
21. Ensure that the test runs in the **US West (Oregon)** location.



22. Enter **Jmeter Test 2** in the name text box and save the test.

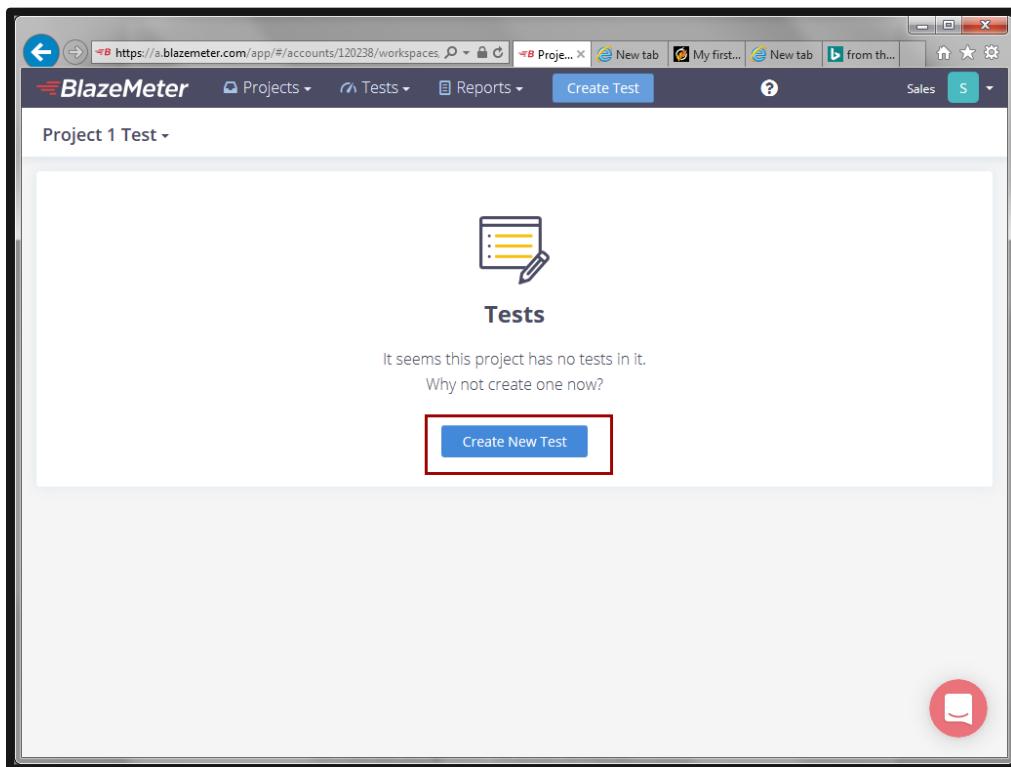


23. In the BlazeMeter UI, click **Project 1 Test** to view the different tests that you have added recently for the project.



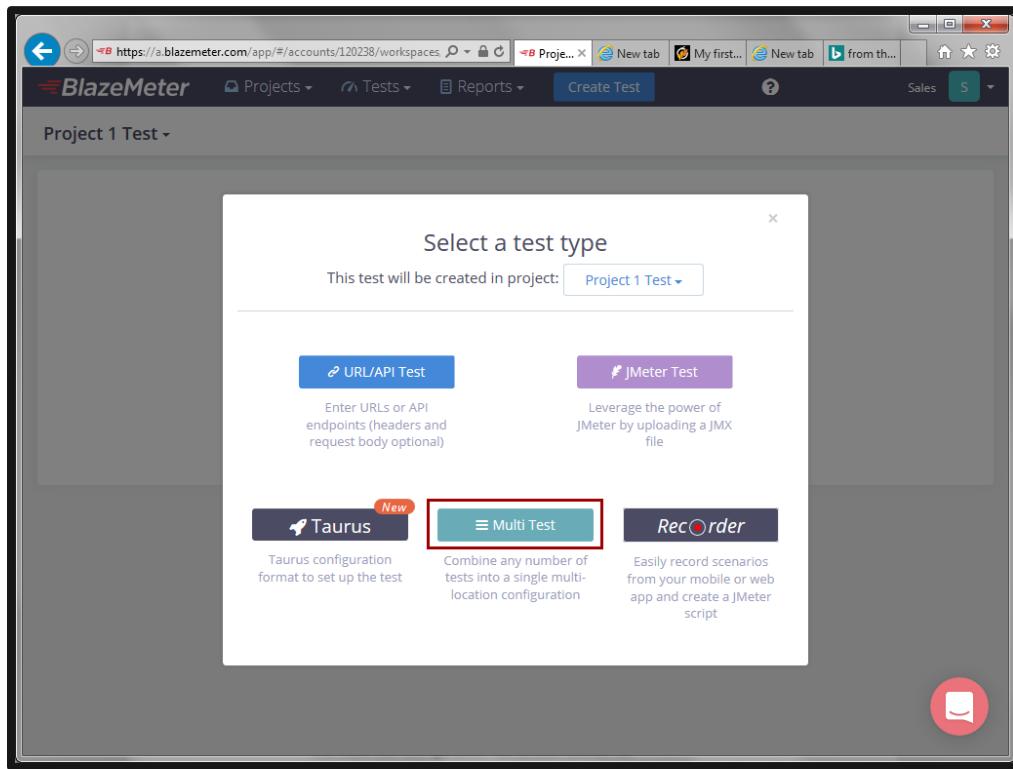
The screenshot shows the BlazeMeter web interface. At the top, there's a navigation bar with links for 'Projects', 'Tests', 'Reports', 'Create Test', and 'Sales'. Below the navigation is a search bar labeled 'Find a test...'. A table lists three existing tests: 'Jmeter Test 2' (US West, JMeter, 1 server, created 'in a few seconds'), 'URL Test 1' (US West, URL, 1 server, created '35 minutes ago'), and 'Jmeter Test 1' (EU West, JMeter, 1 server, created '37 minutes ago'). A large blue '+' button is located in the center of the main content area.

24. To create a Multi Test, click **Create New Test** from the BlazeMeter UI.

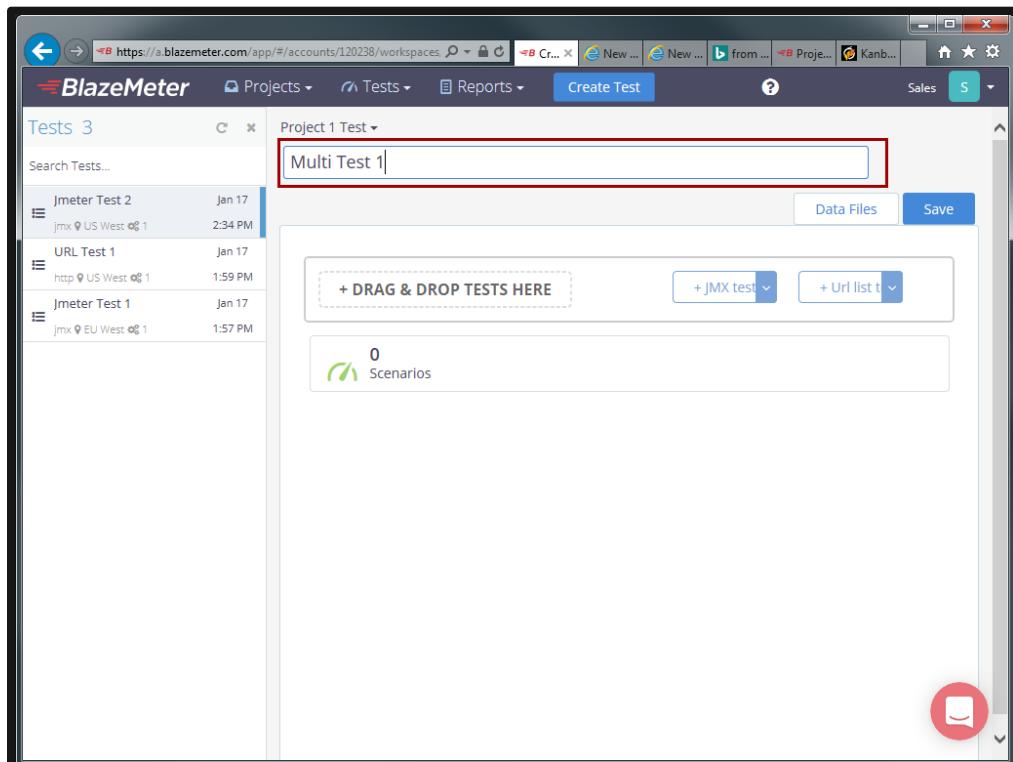


The screenshot shows the same BlazeMeter interface as the previous one, but the 'Create Test' button has been clicked, leading to a new screen. This screen features a large icon of a notepad with a pencil. Below the icon, the word 'Tests' is displayed. A message states 'It seems this project has no tests in it.' followed by 'Why not create one now?'. A prominent blue 'Create New Test' button is centered at the bottom of the screen. A red box highlights this button.

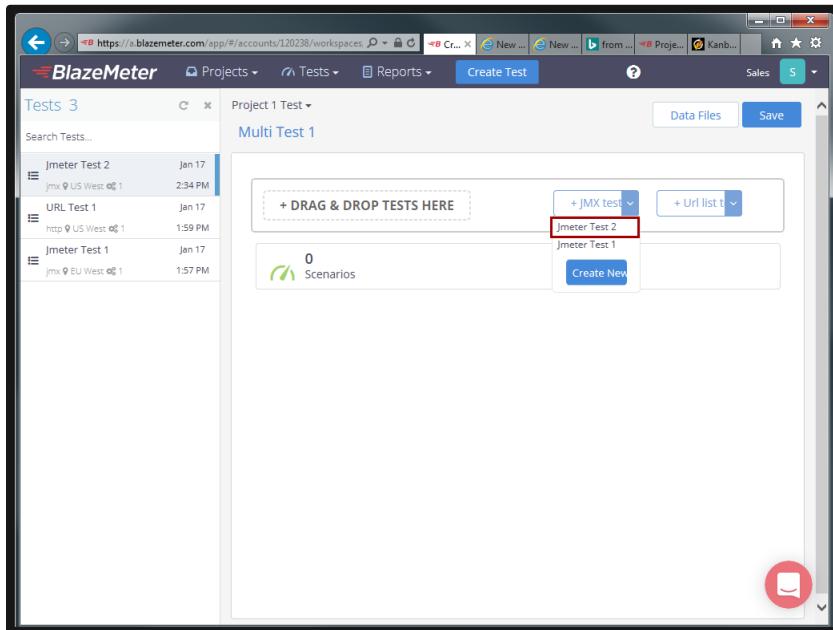
25. In the **Select a test type** dialog box, click **Multi Test**.



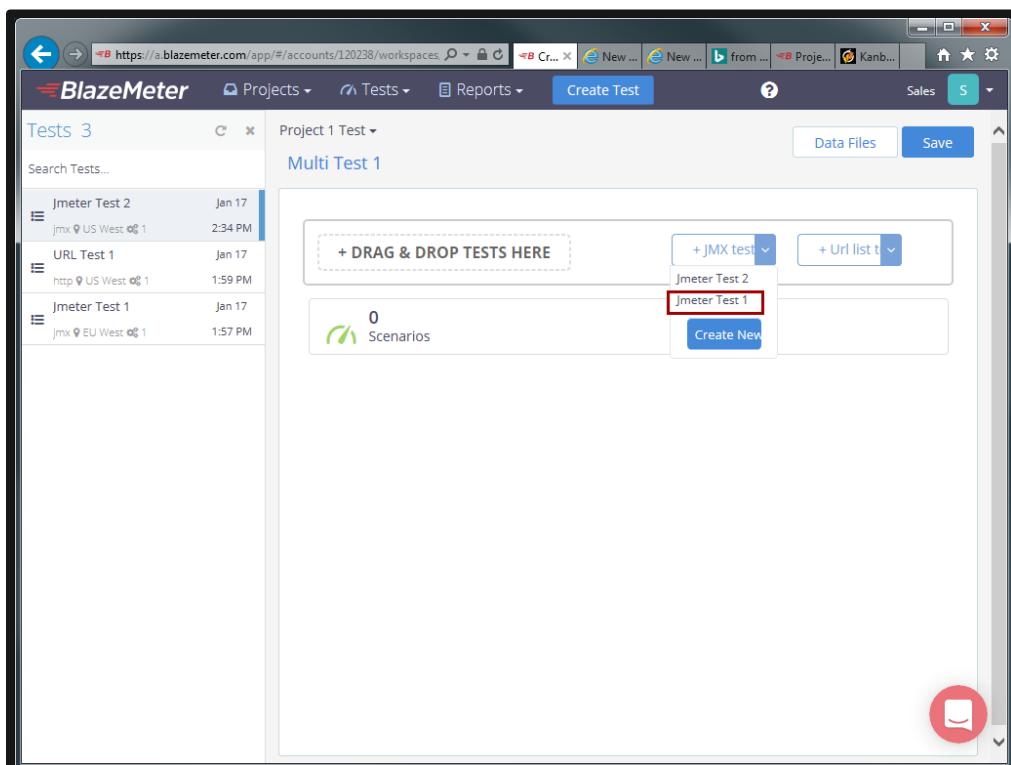
26. In the Name text box, enter **Multi Test 1**.



27. To add different tests to the Multi Test, you can do a simple drag and drop the tests displayed in the left pane. The other way is to click **+JMX test** to add JMeter scripts and **+URL List test** to add URL/API tests. Click **+ JMX test** and select **Jmeter Test 2** from the drop-down list



28. Click **+ JMX test** to add another JMeter Test. Select **Jmeter Test 1** from drop-down list



29. To add URL/API Test, click **+Url list test** and select **URL Test 1**.

The screenshot shows the BlazeMeter web interface for creating a multi-test. On the left, a sidebar lists existing tests: 'Jmeter Test 2' (Jan 17), 'URL Test 1' (Jan 17), and 'Jmeter Test 1' (Jan 17). The main area is titled 'Multi Test 1'. It contains a 'DRAG & DROP TESTS HERE' box, a 'Scenarios' section (2 Scenarios, EU West (Ireland), US West (Oregon, Google)), and two test configurations:

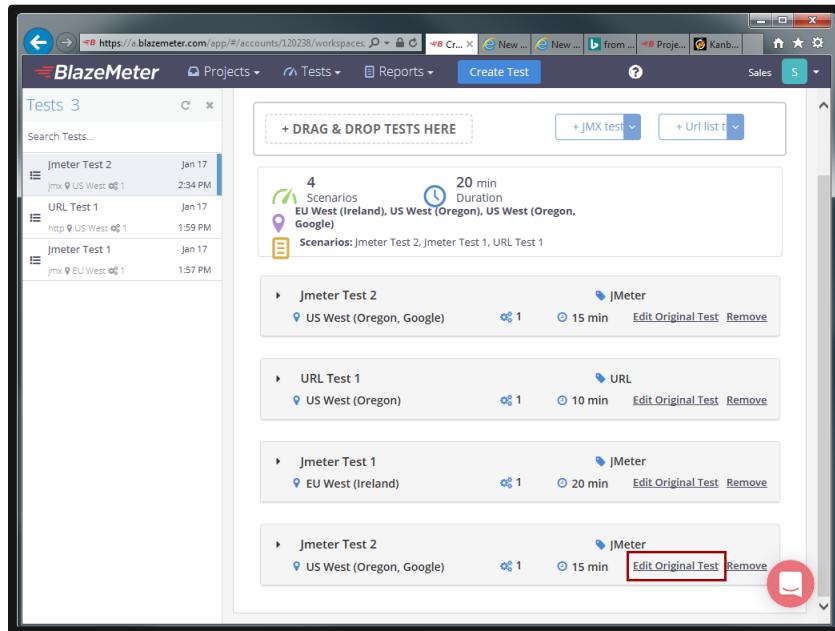
- Jmeter Test 1:** JMeter, EU West (Ireland), 1 user, 20 min duration.
- Jmeter Test 2:** JMeter, US West (Oregon, Google), 1 user, 15 min duration.

Editing JMeter Test Configuration Details.

30. Based on the requirement, you can add each test multiple times. To add another JMeter Test, click **+ JMX test** and select **Jmeter Test 2**. All the tests that have been added are displayed here. You can even remove a test and edit the test configuration details of a specific test. Click **Remove** to delete a test and click **Edit Original Test** to change the configuration details.

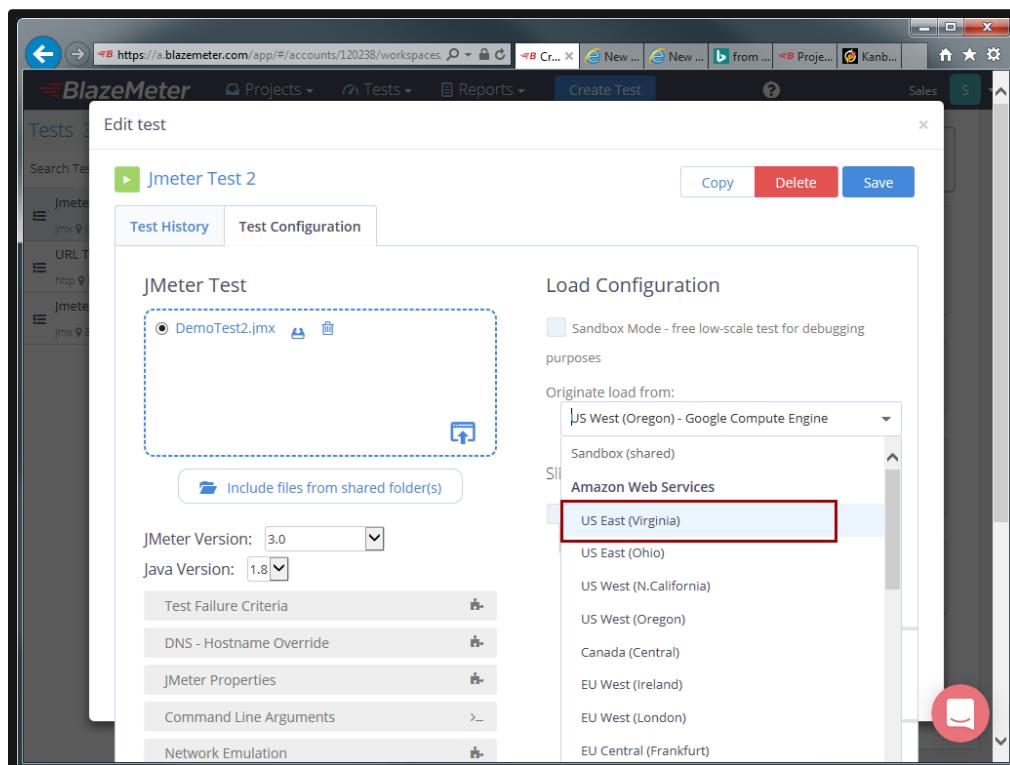
This screenshot shows the same 'Multi Test 1' configuration as the previous one, but now it includes three 'Jmeter Test 2' entries. The 'Add Test' dropdown menu at the top right has 'Jmeter Test 2' selected and highlighted with a red box. The list of tests in the main area is also highlighted with a large red box.

31. To edit the test configuration details, click **Edit Original Test**.



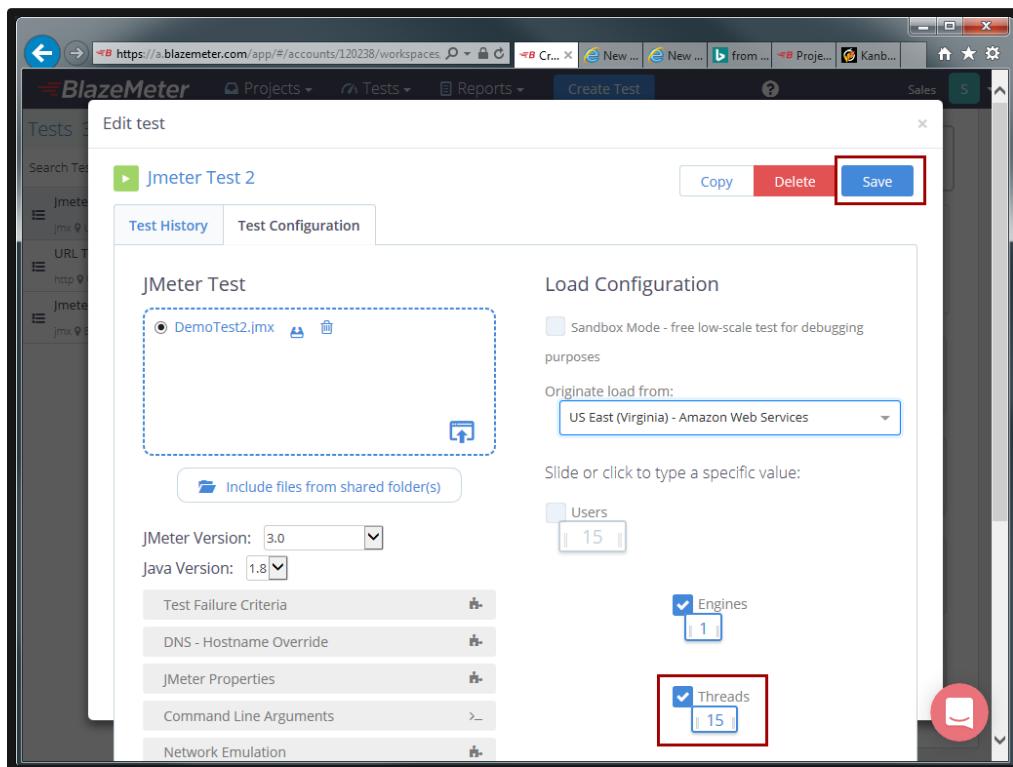
The screenshot shows the BlazeMeter interface with the 'Tests' tab selected. A list of existing tests is on the left, and the main area displays a summary of a multi-test configuration. The configuration includes four scenarios: 'Jmeter Test 2' (US West, 1 thread, 15 min duration), 'URL Test 1' (US West, 1 thread, 10 min duration), 'Jmeter Test 1' (EU West, 1 thread, 20 min duration), and 'Jmeter Test 2' (US West, 1 thread, 15 min duration). The 'Edit Original Test' link for the second 'Jmeter Test 2' entry is highlighted with a red box.

32. Under the 'Load Configuration' section, configure the test to run in the **US East (Virginia)** location.

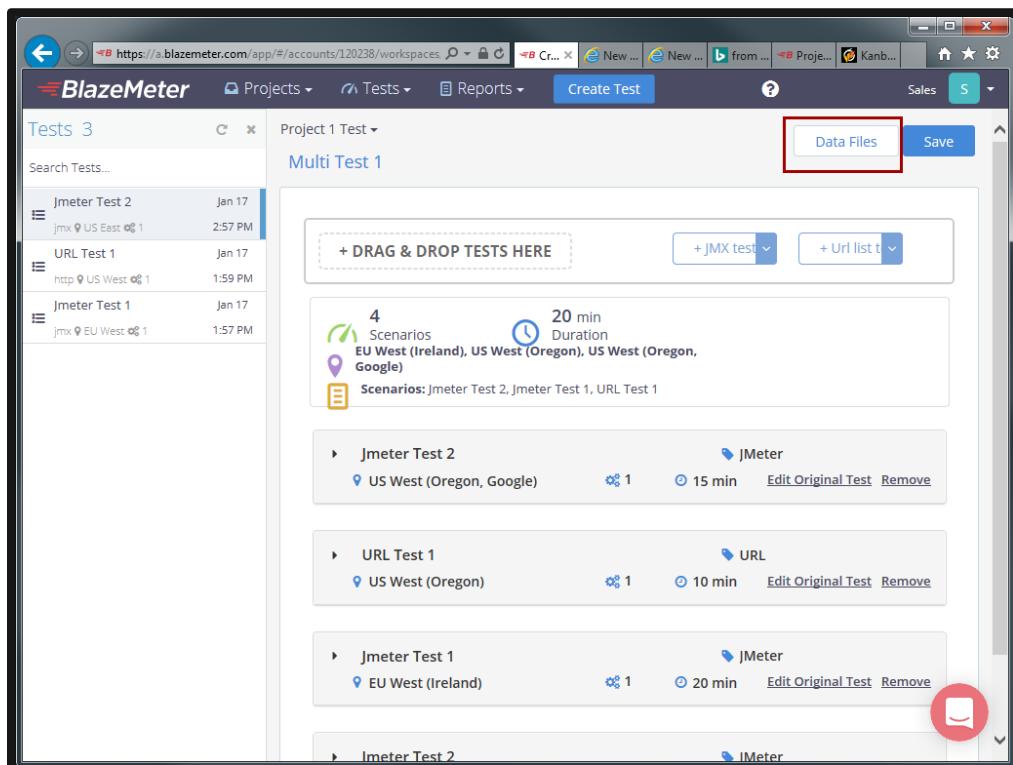


The screenshot shows the 'Edit test' dialog for 'Jmeter Test 2'. In the 'Test Configuration' tab, the 'Load Configuration' section is open. It shows a dropdown menu for 'Originate load from:' with 'US East (Virginia)' selected. This option is highlighted with a red box. Other available locations listed in the dropdown include US West (Oregon) - Google Compute Engine, US East (Ohio), US West (N. California), US West (Oregon), Canada (Central), EU West (Ireland), EU West (London), and EU Central (Frankfurt).

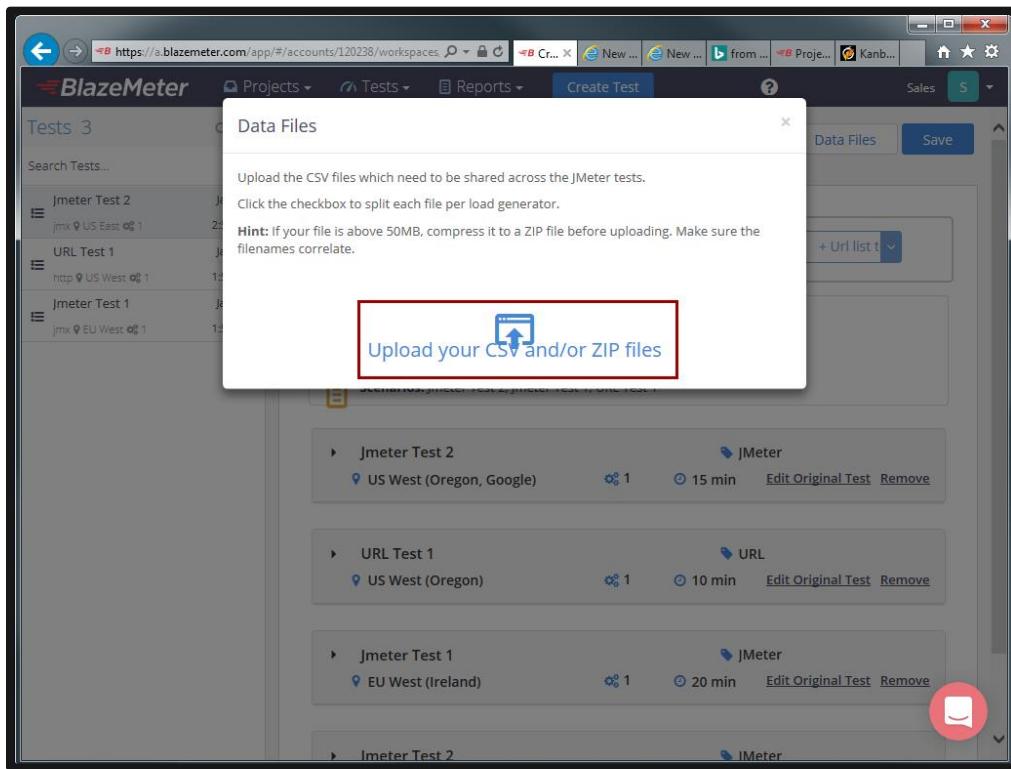
33. Select **Threads** and enter '**15**'. Click **Save** to ensure that updated test configuration details are saved.



34. In the Multi Test UI, click **Data Files**. If you run a multi engine test, and you wish to use a single CSV file for all tests without creating duplicate values, use the 'split CSV' option.

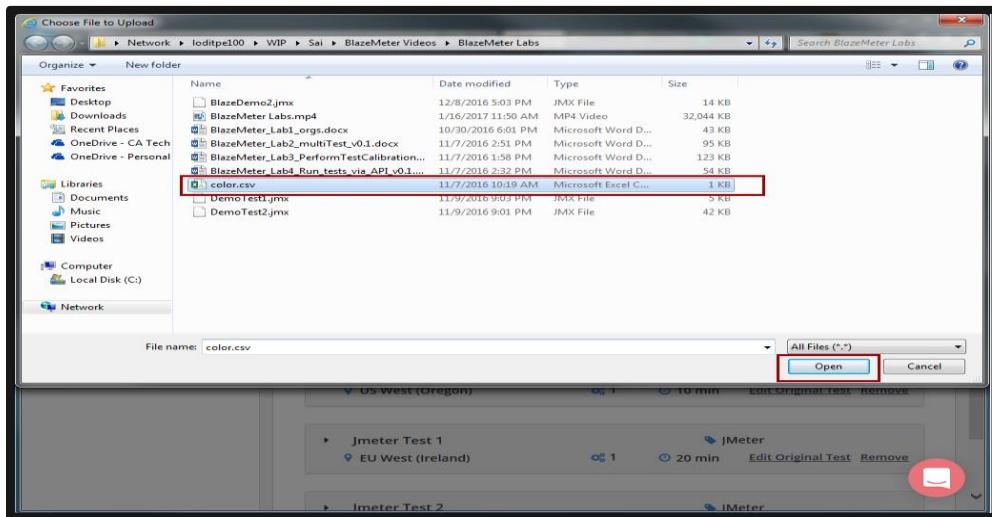


35. Click **Upload** icon to add the CSV file that need to be shared across different JMeter tests.

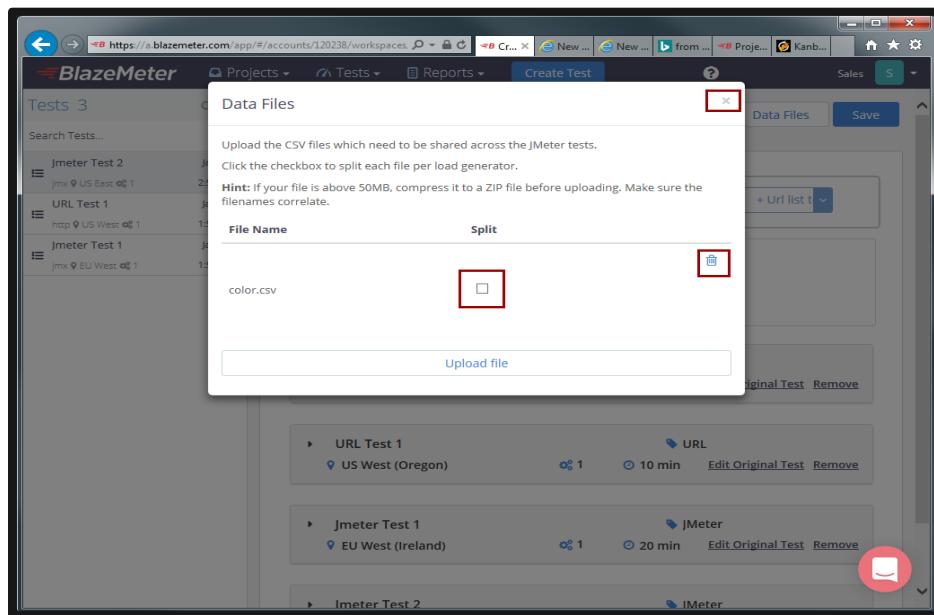


Note: If your CSV file size is above 50MB, it is suggested to compress it to a zip file before uploading it. Make sure that the filenames correlate.

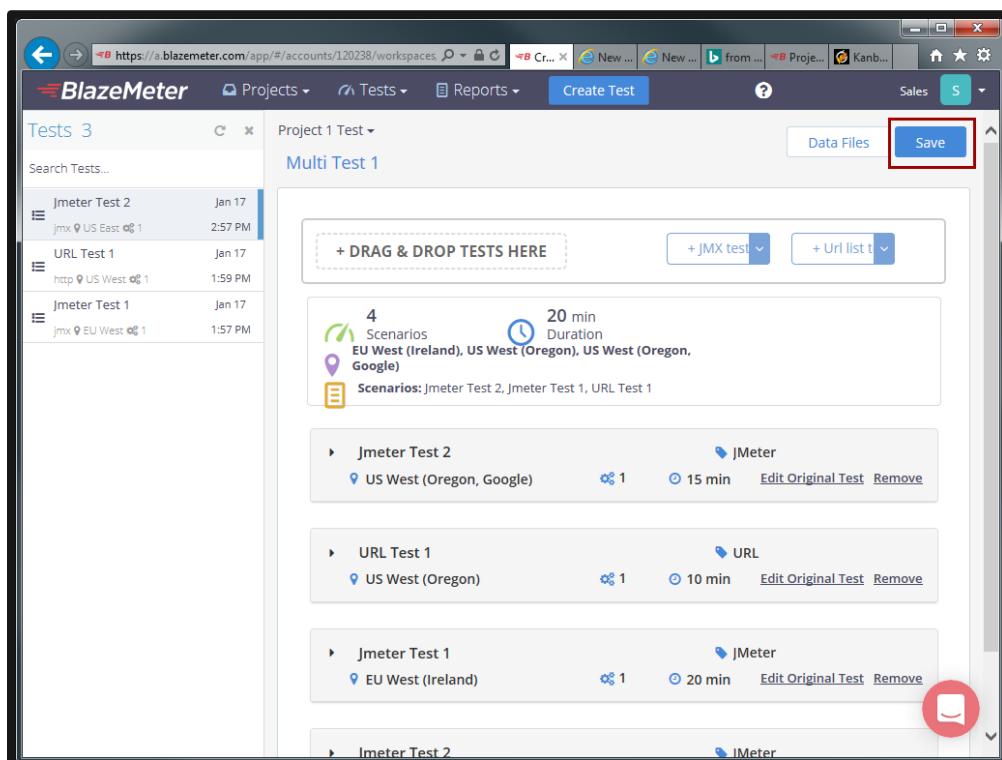
36. Select the required .CSV file and click **Open** to add the file.



37. Once the file is added, click 'X' to close the dialog box. You are provided with an option to 'Spilt CSV' so that the same CSV file is shared across all the JMeter test



38. Click **Save** to save the Multi Test with updated Test Configuration details.



Running a Multi Test.

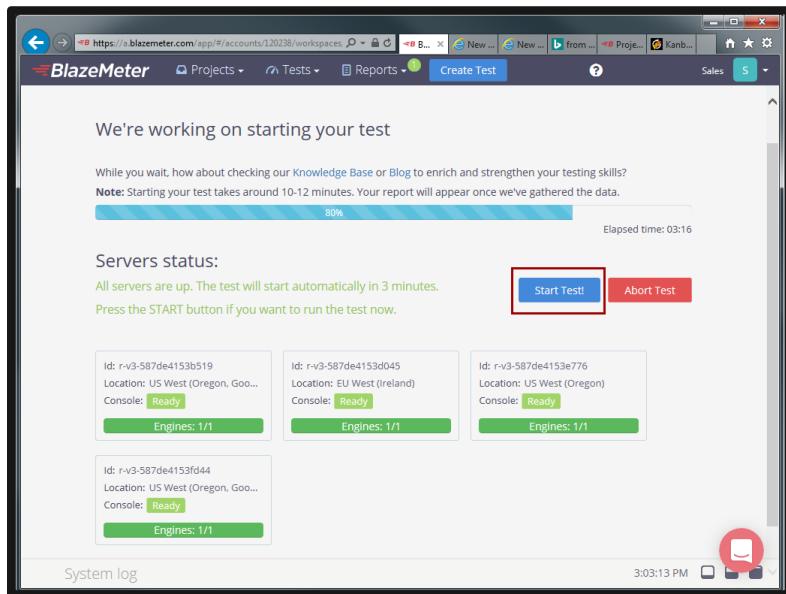
39. Click the **play** button to start the Multi Test. You will be notified that you are about to launch a new load.

The screenshot shows the BlazeMeter web interface. In the center, there's a configuration panel for a 'Multi Test 1'. At the top right of this panel, there's a green rectangular button with a white play icon and the text 'Multi Test 1'. Below this, there are tabs for 'Test History' and 'Test Configuration'. Under 'Test Configuration', there's a section titled '+ DRAG & DROP TESTS HERE' with three items listed: 'Jmeter Test 2', 'URL Test 1', and 'Jmeter Test 1'. Each item has details like location ('US West (Oregon), Google'), duration ('20 min'), and execution count ('1').

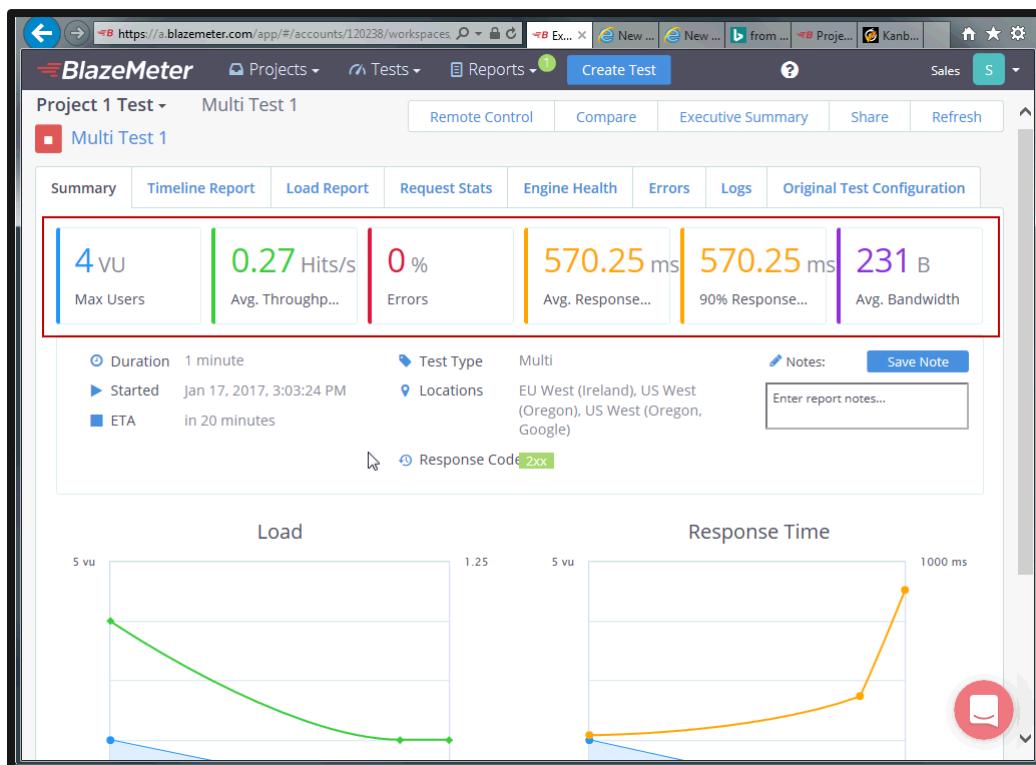
40. Once the test is initiated, a dialog box appears and you must ensure that **Synchronized Start** check box is selected. Selecting 'Synchronized Start' will make sure that all servers are up before starting the test. This option is useful if you have a concern that some servers or locations are significantly slower than others and you wish to synchronize them. Click **Launch Servers** to start the servers and run the Multi test.

This screenshot shows a 'Launch Multi Test' dialog box overlaid on the BlazeMeter interface. The dialog contains instructions about launching a load test and lists the configuration: 4 Consoles, 4 Engines, 55 Total threads, and a Max duration of 20 minutes. It also includes a note about 'Synchronized Start' and two checkboxes: 'Synchronized start' (which is checked) and 'Run test in the background'. At the bottom of the dialog is a large green button labeled 'Launch Servers'.

41. Observe the booting screen while servers are getting ready. Note that each individual JMeter Test and API/URL Test that you added have a unique session ID. Click **Start Test** to initiate the test.



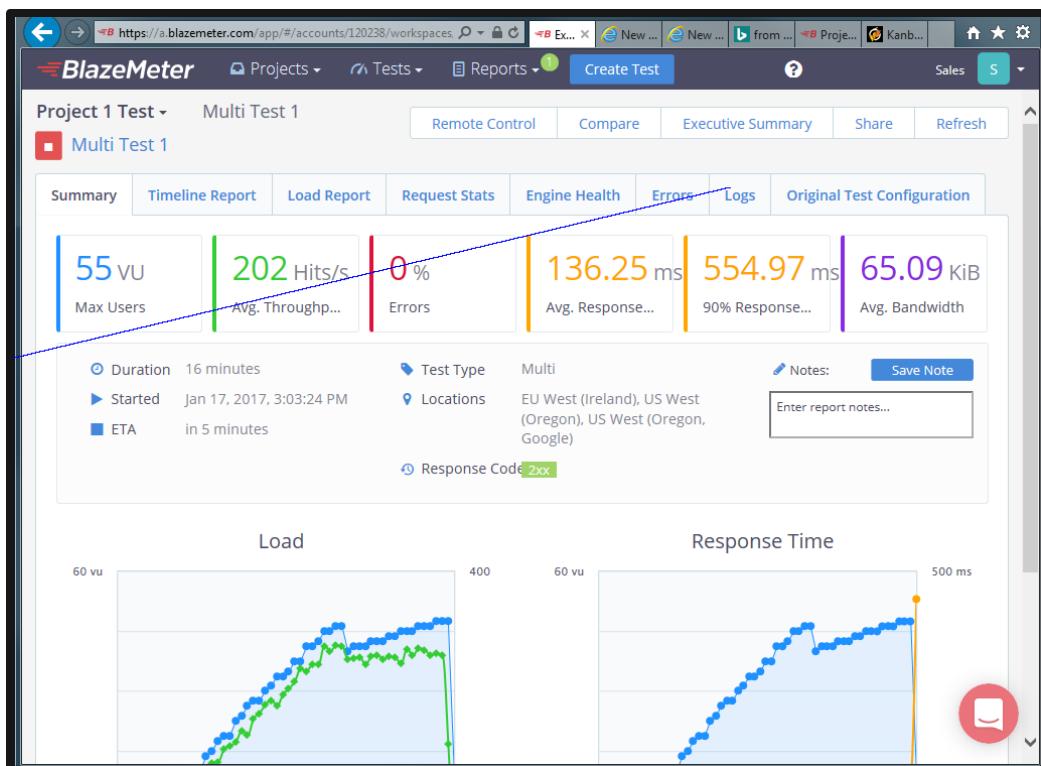
42. Once the Multi Test starts, you can observe different types of reports being displayed for the Multi Test along with the average throughput time, average response time, average bandwidth and maximum virtual users. Even the Load and Response Time graphs keep changing with the number of virtual users.



43. Click **Logs** to view total users configured collectively for all the tests under Multi Test along with the duration to execute the Multi Test successfully. In this scenario, the total users are 55 and the duration is 20 minutes.

The screenshot shows the BlazeMeter interface for a 'Multi Test 1'. The 'Logs' tab is highlighted with a red box. The summary section indicates 4 Scenarios, 55 Total Users, and a Duration of 20 min. Below this, four individual test configurations are listed: Jmeter Test 2 (15 min), Jmeter Test 1 (20 min), URL Test 1 (10 min), and Jmeter Test 2 (15 min).

44. Observe the Load and Response Time graphs once the maximum number of virtual users is reached. An aggregate report that contains complete data and results of the Multi Test is generated and sent to the registered email address. To ensure that Multi Test is executed successfully, observe the errors percentage and make sure that it is 0%.



Lab 4 – Create, Update, and Run JMeter Tests via APIs

Goals

Create, update and run JMeter Tests in BlazeMeter UI through Rest APIs.

Scenario

You can quickly create a JMeter script by using a CURL command with JSON body instead of going to the GUI and following click-by-click procedure. It is more convenient to run tests via an API in the continuous integration environment. Load tests can be easily created and configured online. Once these tests are created, it is possible to run them using the BlazeMeter REST API. This means that instead of manually starting the tests, you can simply create a script that will automatically run a predefined test via the API.

BlazeMeter has a new, improved API for performing various actions that can be done through the UI programmatically. The API is RESTful and works with JSON messages over HTTP. It relies on the standard HTTP verbs including GET, POST, PUT, DELETE, and PATCH.

You will complete this lab by performing the following tasks.

- Identify API key
- Create a JMeter test
- Upload a JMX file
- Update a JMeter test
- Run a JMeter Test
- Observer the results in report

Time

30 minutes

Instructions:**Accessing the BlazeMeter Application**

1. Enter <https://a.blazemeter.com/app/sign-in> in your browser.
2. At the login prompt, enter the following details and click **SIGN IN**:

Field	Value
User Name	Enter your registered user name
Password	Enter a valid password

3. On clicking **SIGN IN**, the BlazeMeter application opens.

Identify API Key

4. From the BlazeMeter UI, you could see the 'User Settings' icon at the top right corner of the screen. Click it and then click **Settings**.

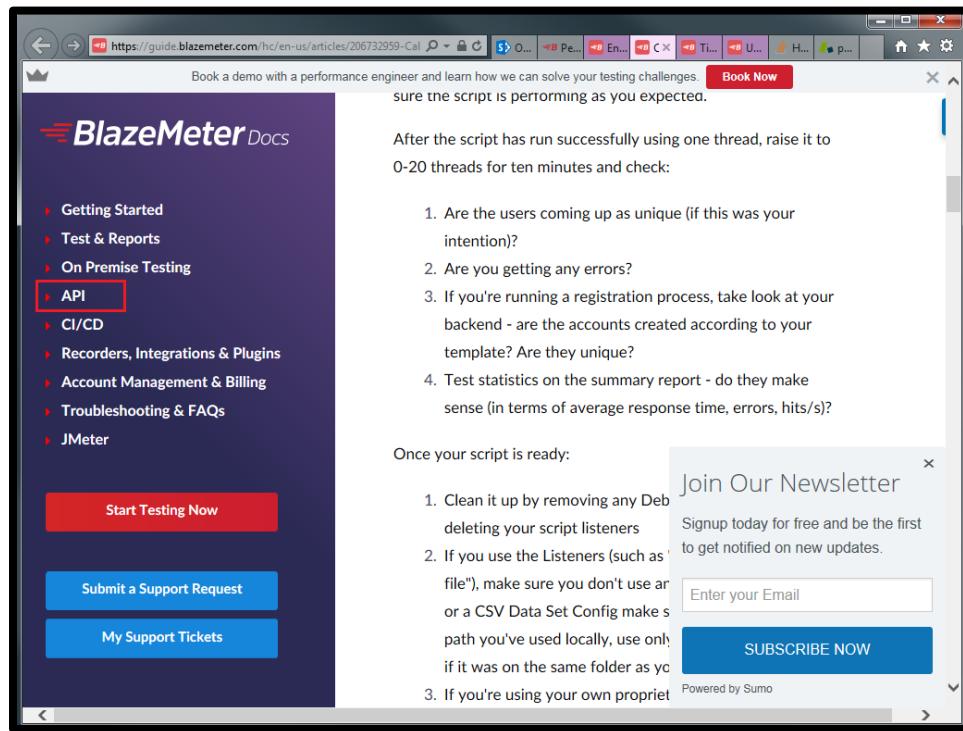
The screenshot shows the BlazeMeter application interface. At the top, there's a navigation bar with links for Projects, Tests, Reports, Create Test, and a user icon. Below the navigation bar is a workspace selector showing 'Sales' is selected. To the right of the workspace selector is a dropdown menu with sections for ACCOUNTS and WORKSPACES. The 'Sales' workspace is listed under WORKSPACES. A red box highlights the 'Settings' link in the dropdown menu. The main content area displays a list of test projects under 'Project 1 Test'. Each project entry includes the name, location, type, and users/engines. A large blue '+' button is at the bottom left, and a red circular icon with a white question mark is at the bottom right.

5. You will see the 'API Key' section in the 'Profile Settings' window. Copy and paste the API Key into a document for later use. This key will be used as a parameter in the API to identify the user.

The screenshot shows the 'Profile Settings' page within the BlazeMeter application. On the left, there's a sidebar with 'CA_SALES_ACCT' and 'Settings' selected. Under 'Personal', there are links for 'Workspace' and 'Personal Settings'. The main content area has a profile picture placeholder 'A' and ID '334563'. It shows 'Profile Settings' with fields for First name ('Araboina'), Last name ('Sal Kumar'), and Email ('kumar12@ca.com'). There's a 'Change Password' link and a 'Save changes' button. Below this is the 'API Key' section, which contains the text 'Your Current Key: o1ln5dcjbls9g0jztjg' with a copy icon, a note about reissuing the key, and a 'Re-issue API Key' button. At the bottom, there's a 'Default Test Location' section with account information.

Creating a JMeter Test Using API

6. To create a JMeter Test by using an API, you must first navigate to the BlazeMeter Guide. Enter <https://guide.blazemeter.com> and navigate to Create a Test section under API.



7. Copy the CURL command syntax from the web page.
 8. Open the command line and paste the CURL command that you have copied from web page.

Syntax

```
curl -X post https://a.blazemeter.com:443/api/latest/tests/ -H "Content-Type: application/json" -H "x-api-key: 65e9f61eb363b2dc1d21" -d '{"projectId":null,"configuration":{"dedicatedIpsEnabled":false,"location":"us-west-1","concurrency":80,"plugins":{"splitCSV":{"enabled":false}, "reportEmail":{"enabled":true}, "jmeter":{"override":{"duration":25,"rampup":150,"threads":null,"iterations":1}, "version":"2.11blazemeter","filename":"file.jmx"}}, "type":"jmeter", "serverCount":null}, "name":"MyJmeterTest"}'
```

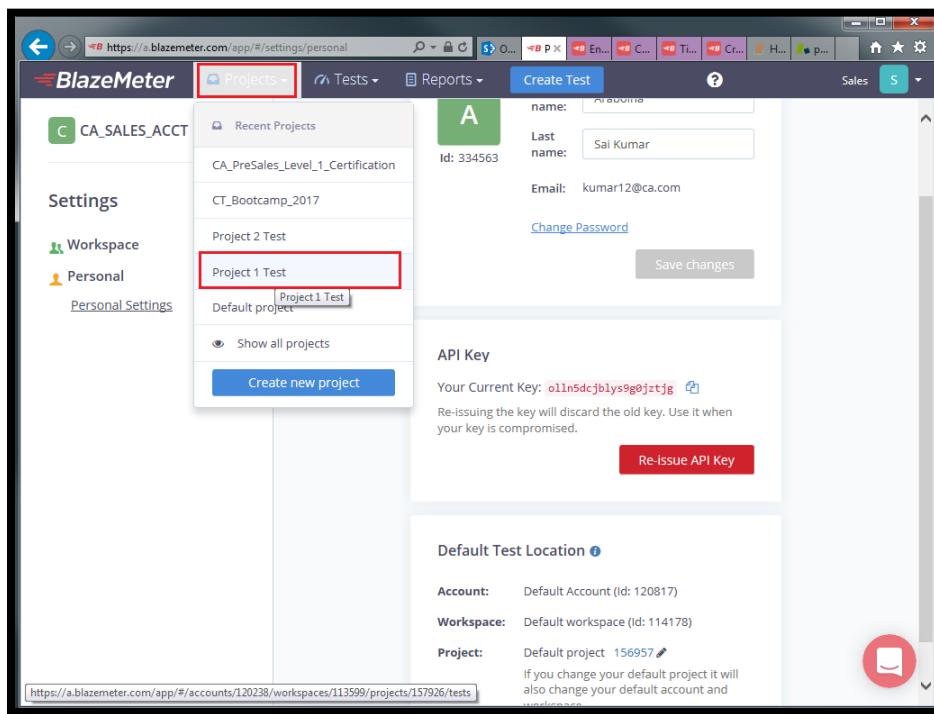
Parameter	Value
name	The verbal identifier of the test.
location	The Geolocation of the load generators.
rampup	The timeframe in which the load will be increased from 0 to the Max concurrent users set.
Iterations	The number of iterations the test will run until finished. Infinite iteration presented as '-1'.
concurrency	Max number of threads active concurrently per engine.
dedicatedIpsEnabled	True will use the Dedicated IPs you've purchased.
splitCSV.enabled	True will split the CSV/s.
reportEmail.enabled	True if you like to receive an Email at test's end.
duration	The period of time for which the test will run until shut down.
jmeter.version	The Jmeter version used to run this test, e.g '2.12 blazemeter'.
type	It can be a Jmeter, URL list or Webdriver Test.
created	The time the test was started at.
updated	The time the test was last updated.
Id	The unique identifier of the test. The 'Test ID'.
filename	the name of the JMX script uploaded.
userId	The unique identifier of the user.

9. Based on the requirement, you can edit the API syntax. Paste the API Key that you have copied initially into the syntax.

```
curl -X post https://a.blazemeter.com:443/api/latest/tests/ -H "Content-Type: application/json" -H "x-api-key: 65e9f61eb363b2dc1d21" -d '{"projectId":null,"configuration":{"dedicatedIpsEnabled":false,"location":"us-west-1","concurrency":80,"plugins":[{"splitCSV":{"enabled":false}}],"reportEmail":{"enabled":true}},"jmeter":{"override":{"duration":25,"rampup":150,"threads":null,"iterations":-1}, "version":"2.12blazemeter","filename":"file.jmx"}}, {"name":"MyJmeterTest"}
```

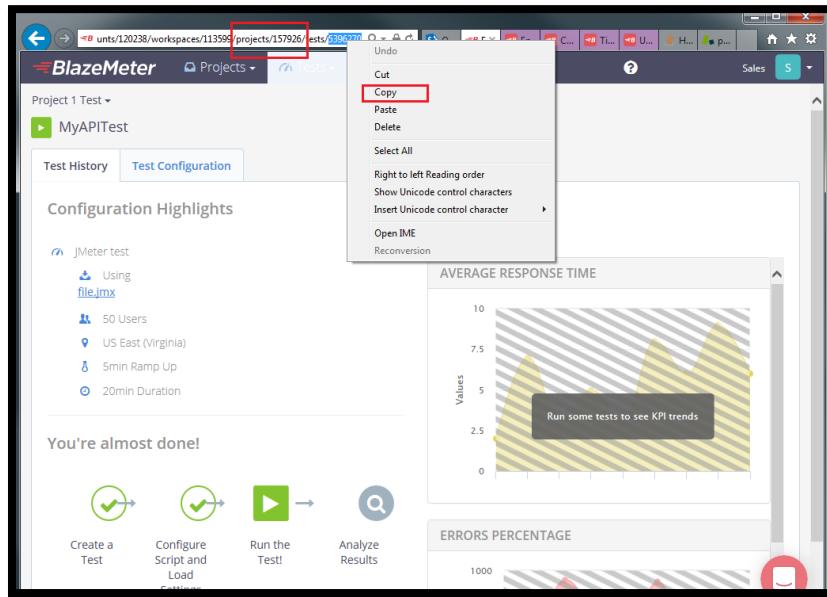
Parameter	Value
location	Us-east-1
concurrency	50
Duration	20
Ramup (sec)	300
JMeter Version	3.1
Name	MyAPITest

10. To identify the Project ID for a specific project, navigate back to the BlazeMeter UI, click **Projects** tab and select the project to which you want to add the test.



The screenshot shows the BlazeMeter UI with the 'Projects' tab selected. On the left sidebar, under 'Recent Projects', there is a list of projects: CA_SALES_ACCT, CA_PreSales_Level_1_Certification, CT_Bootcamp_2017, Project 2 Test, and Project 1 Test. The 'Project 1 Test' item is highlighted with a red box. In the main content area, there is a section for 'Recent Projects' with a list of recent projects: CA_PreSales_Level_1_Certification, CT_Bootcamp_2017, Project 2 Test, and Project 1 Test. The 'Project 1 Test' item is also highlighted with a red box. Below this, there is a 'Create new project' button. Further down, there is a 'Default Test Location' section with account, workspace, and project details. At the bottom of the page, the URL https://a.blazemeter.com/app/#/accounts/120238/workspaces/113599/projects/157926/tests is visible.

11. After selecting the test, go to Address bar and you can find Project ID. It is a numeric value displayed next to "Projects/<ProjectID>/".



- Once you have modified the different request parameters, press Enter, to execute the command and view the response. It is suggested to validate the output response to ensure that the parameters displayed match with the parameters that you have entered in the syntax.

```

MINGW64:/c/Users/kumar12
$ curl -X POST -F "file=@<path_to_file>/file.jmx" https://a.blazemeter.com/api/v4/tests/<test_id>/files -H "x-api-key: 998a2hk7d26be74xxxxx"
{
  "error": null,
  "result": {
    "id": 5396270,
    "name": "MyAPITest",
    "userId": 113599,
    "created": 1487061745,
    "updated": 1487061746,
    "projectId": 157926,
    "lastUpdated": 1487061746,
    "correlation": {
      "location": "us-east-1",
      "duration": 20,
      "type": "jmeter",
      "concurrency": 50,
      "dedicatedIpsEnabled": false,
      "canControlRampup": false,
      "targetThreads": 50,
      "plugins": [
        "splitCSV": {
          "enabled": false
        },
        "reportEmail": {
          "enabled": true
        }
      ],
      "jmeter": {
        "fileName": "file.jmx",
        "version": "3.1",
        "override": {
          "rampup": 300,
          "iterations": -1,
          "duration": 20
        }
      }
    }
  }
}

kumar12@KUMAR12-7510: MINGW64 ~
$ curl -X POST -F "file=@<path_to_file>/file.jmx" https://a.blazemeter.com/api/v4/tests/<test_id>/files -H "x-api-key: 998a2hk7d26be74xxxxx"

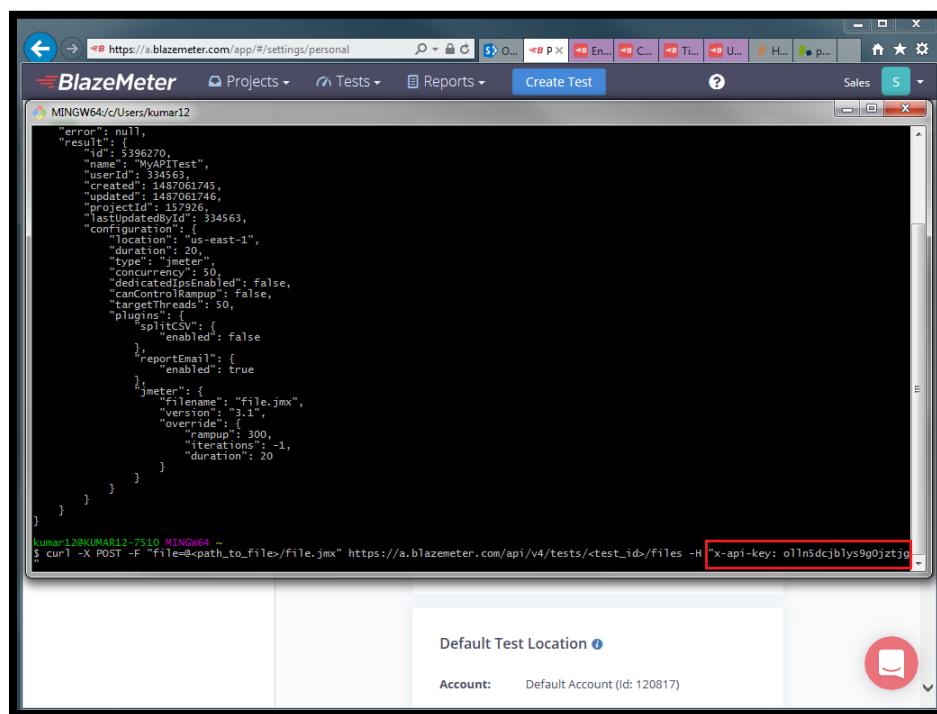
```

Uploading JMX Files to JMeter Test

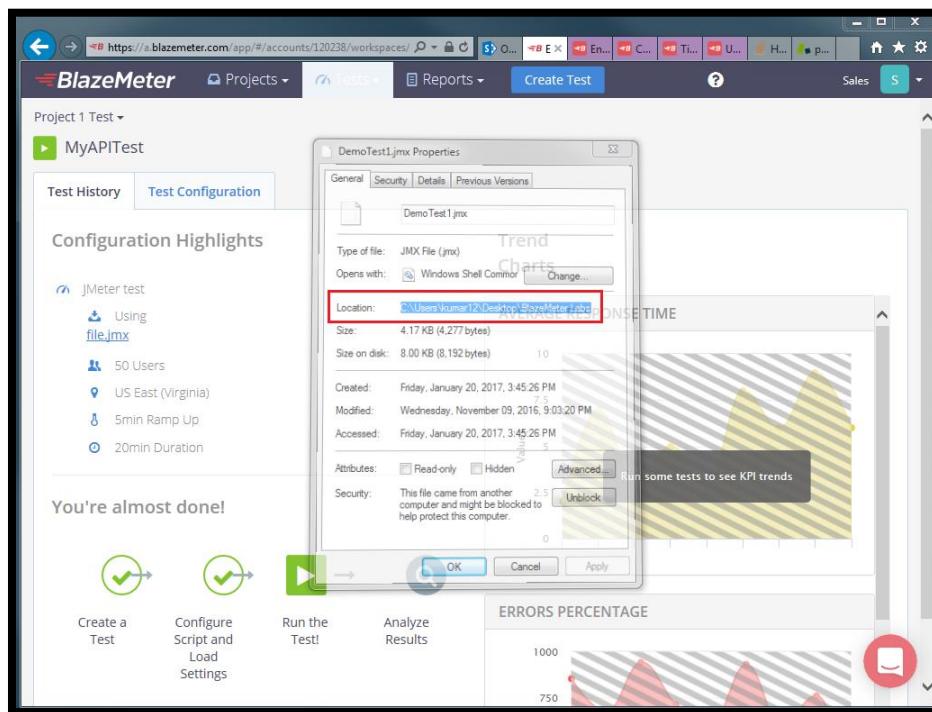
13. To upload a JMX file used for this test, use a Rest API.
14. Return to the BlazeMeter guide and navigate to the [Upload JMX](#) section. Copy the CURL command used for uploading a JMX file. To run this CURL command, open the command line, and paste the CURL command that you have copied.

```
curl -X POST -F "file=@<path_to_file>/file.jmx"
https://a.blazemeter.com/api/v4/tests/<test_id>/files -H "x-api-key: 998a2hk7d26be74xxxxxx"
```

15. Edit the syntax by providing proper file name and the directory where the file is located. Paste the API Key that you have copied earlier to “x-api-key” parameter in the CURL syntax.



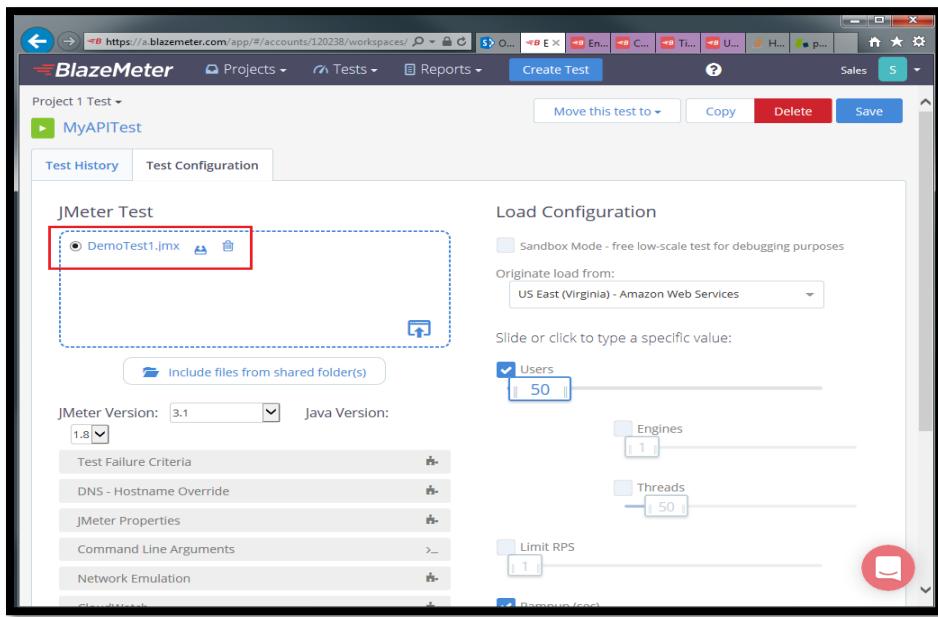
16. Navigate to the folder where the JMX file is saved. Select the filename and paste the same in command line for the respective parameter.
17. Navigate back to the folder where the JMX file is saved. To get the directory path where the file is saved, right-click the file and click **Properties**. From the dialog box, copy the directory path where the file is saved.



18. Open the command line and paste the directory path that you just copied.

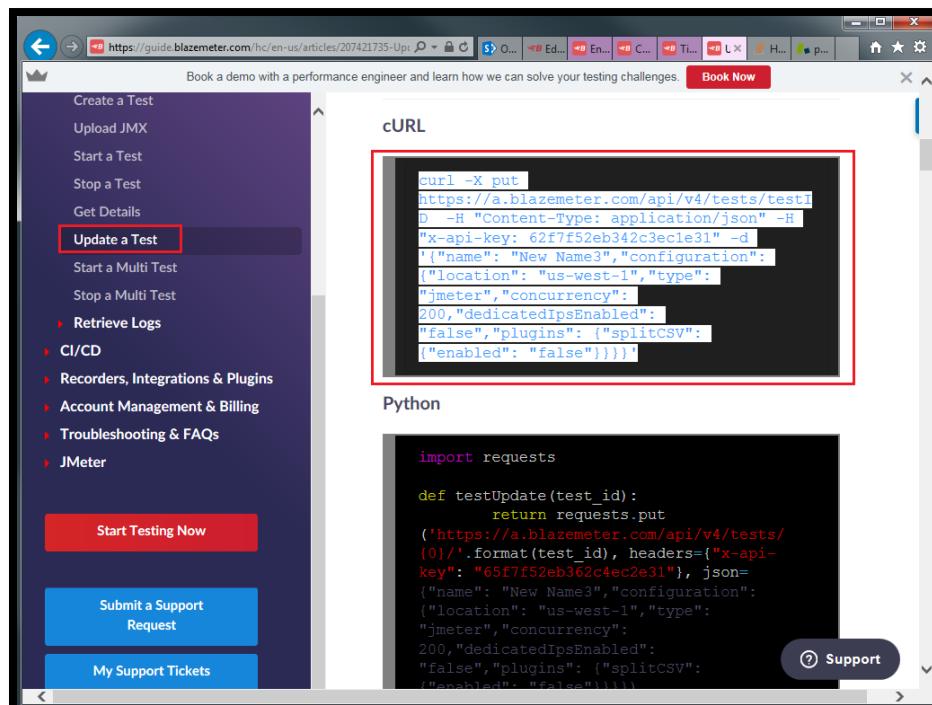
```
MINGW64/c/Users/kumar12
{"error": null,
 "result": [
   {
     "id": 5396270,
     "name": "MyAPITest",
     "userId": 33463,
     "type": "jmeter",
     "concurrency": 50,
     "seed": false,
     "canControlRampup": false,
     "targetThreads": 50,
     "plugins": [
       "splitCSV": {
         "enabled": false
       },
       "reportEmail": {
         "enabled": true
       }
     ],
     "jmeter": {
       "filename": "file.jmx",
       "version": "1",
       "override": {
         "rampup": 300,
         "iterations": 1,
         "duration": 20
       }
     }
   }
]}
kumar12@KUMAR12-7510 MINGW64
$ curl -X POST -F file=@C:/Users/kumar12/Desktop/BlazeMeter Labs/DemoTest1.jmx https://a.blazemeter.com/api/v4/tests/5396270/files -H "x-api-key: oIn5dcjb1ys9gJzrJg"
DemoTest1.jmx Date modified: 11/9/2016 9:03 PM Date created: 1/20/2017 3:45 PM
JMX File Size: 4.17 KB
```

19. Press Enter to execute the command and navigate back to the BlazeMeter UI to confirm that. JMX file is uploaded as expected.



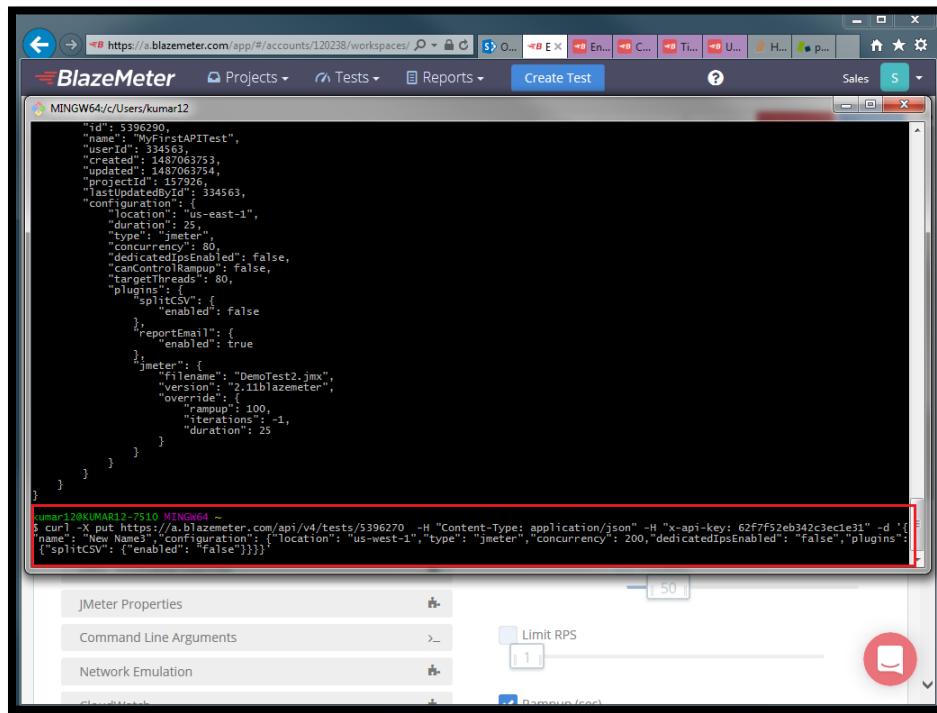
Update the Test Configuration Parameters

20. To update the test configuration details by using APIs, navigate the URL "<https://guide.blazemeter.com/hc/en-us/articles/207421735-Update-a-Test-Update-a-Test>"

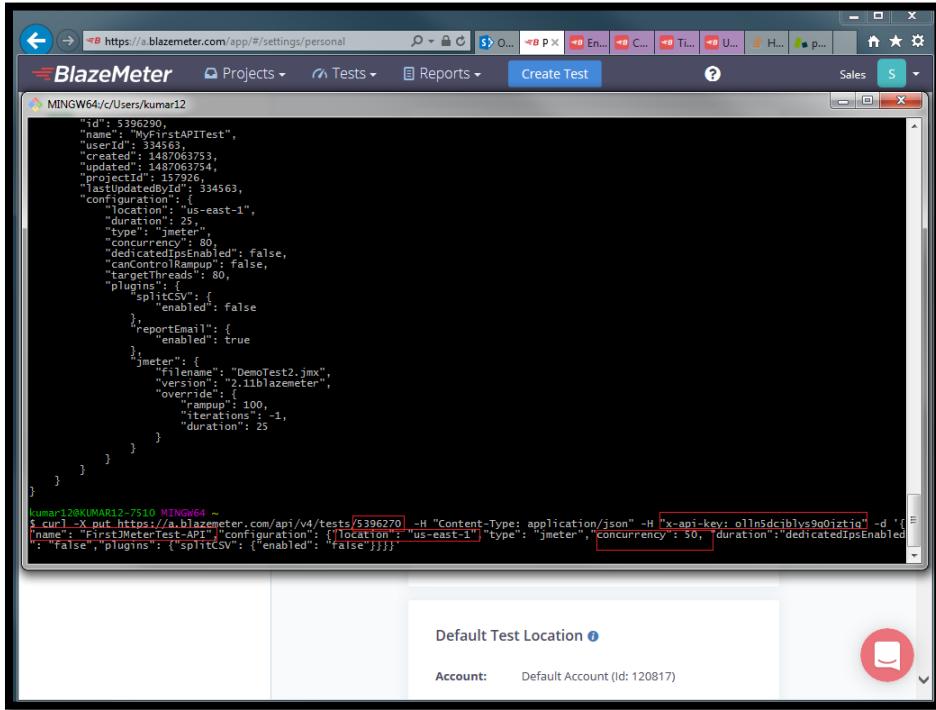


21. From the web page, copy the CURL command used for updating the JMX file. To run this CURL command, open the command line and paste the CURL command.

```
curl -X put https://a.blazemeter.com/api/v4/tests/testID -H "Content-Type: application/json" -H "x-api-key: 62f7f52eb342c3ec1e31" -d '{"name": "New Name3", "configuration": {"location": "us-west-1", "type": "jmeter"}, "concurrency": 200, "dedicatedIpsEnabled": "false", "plugins": {"splitCSV": {"enabled": "false"}}}}'
```



22. Update the parameters with new values that you want to see in the JMeter Test UI. Based on the requirement, you can change the name, location and concurrency details. Navigate back to the document where you have placed your API Key in the earlier step, copy it and paste the same in the syntax. In our scenario, change the name to "FirstJMeterTest-API" and update the concurrency value from 200 to 50 and location to "us-east-1".



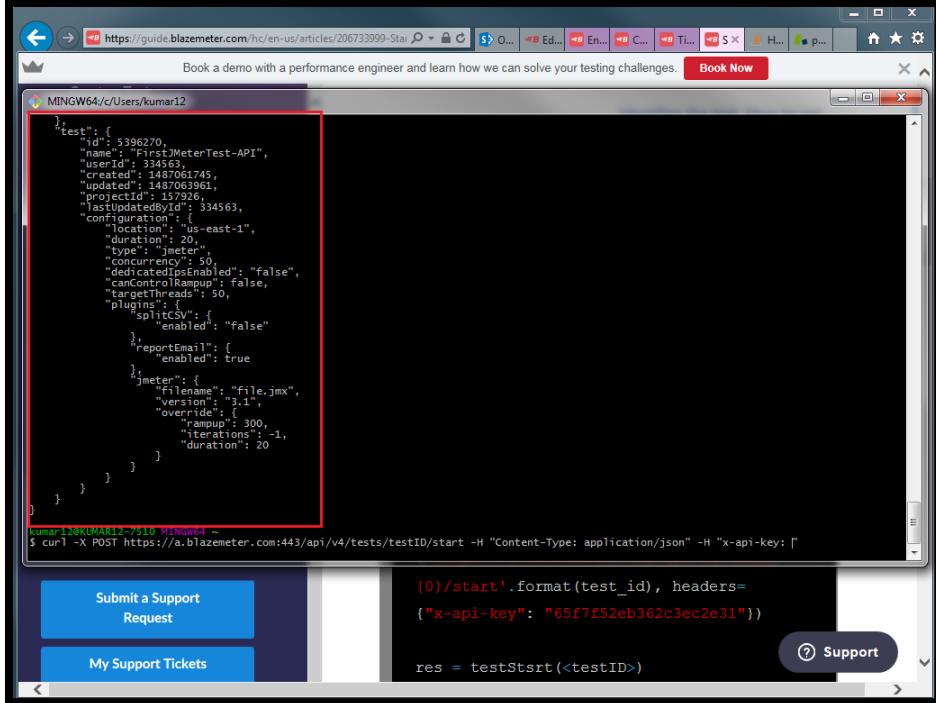
```

{
  "id": 5396250,
  "name": "FirstAPItest",
  "userId": 334563,
  "created": 1487063753,
  "updated": 1487063754,
  "lastUpdatedById": 334563,
  "configuration": {
    "location": "us-east-1",
    "duration": 25,
    "reportEmail": {
      "enabled": true
    },
    "concurrency": 80,
    "dedicatedIpsEnabled": false,
    "canControlRampup": false,
    "totalThreads": 80,
    "plugins": {
      "splitCSV": {
        "enabled": false
      },
      "reportEmail": {
        "enabled": true
      }
    },
    "jmeter": {
      "filename": "DemoTest2.jmx",
      "version": "2.11blazemeter",
      "override": {
        "rampup": 100,
        "iterations": -1,
        "duration": 25
      }
    }
  }
}

kumar1@KIMARI1-7510 MINGW64 ~
$ curl -X put https://a.blazemeter.com/api/v4/tests/5396270 -H "Content-Type: application/json" -H "x-api-key: oln5dcjblvs9o0iztig" -d '{ "name": "FirstMeterTest-API", "configuration": { "location": "us-east-1" } type: "jmeter", "concurrency": 50, "dedicatedIpsEnabled": "false", "plugins": { "splitCSV": { "enabled": "false" } } }'

```

23. Press Enter to see the response below to confirm that parameters modified are reflected in the response.



```

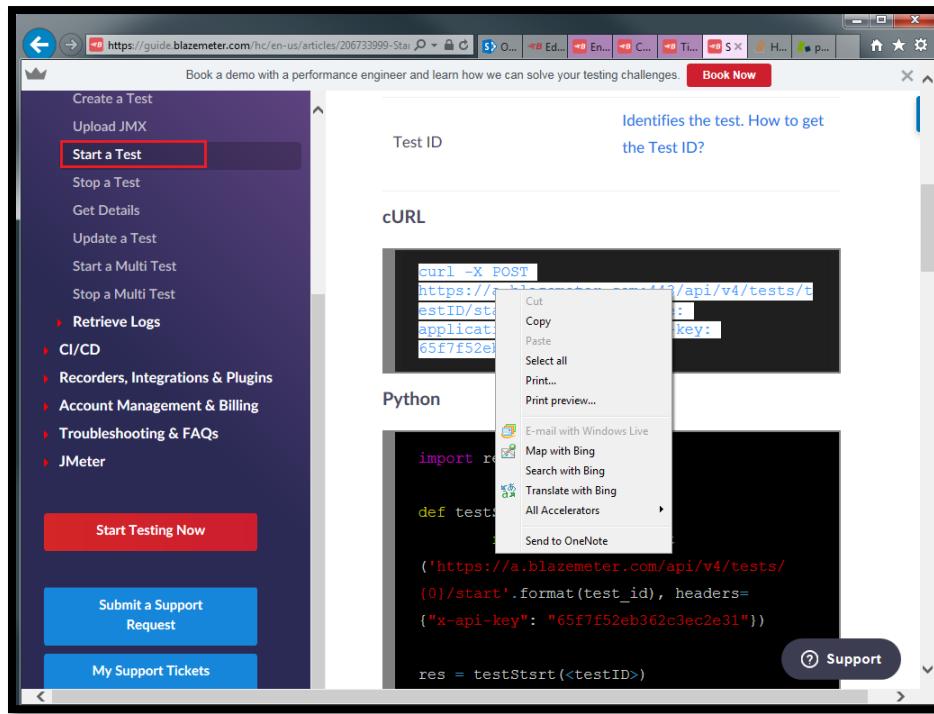
{
  "test": {
    "id": 5396270,
    "name": "FirstJMeterTest-API",
    "userId": 334563,
    "created": 1487061745,
    "updated": 1487063961,
    "lastUpdatedById": 334563,
    "configuration": {
      "location": "us-east-1",
      "duration": 20,
      "reportEmail": {
        "enabled": true
      },
      "concurrency": 50,
      "dedicatedIpsEnabled": "false",
      "canControlRampup": false,
      "totalThreads": 50,
      "plugins": {
        "splitCSV": {
          "enabled": "false"
        },
        "reportEmail": {
          "enabled": true
        }
      },
      "jmeter": {
        "filename": "file.jmx",
        "version": "3.1",
        "override": {
          "rampup": 300,
          "iterations": -1,
          "duration": 20
        }
      }
    }
  }
}

kumar1@KIMARI1-7510 MINGW64 ~
$ curl -X POST https://a.blazemeter.com:443/api/v4/tests/testID/start -H "Content-Type: application/json" -H "x-api-key: [REDACTED]"

```

Start the JMeter Test

24. To start the test, navigate to the <https://guide.blazemeter.com/hc/en-us/articles/206733999-Start-a-Test-Start-a-Test> URL.



25. From the web page, copy the CURL command that is used for starting the preconfigured load test. To run this CURL command, open the command line and paste the CURL command that you just copied.

```
curl -X POST https://a.blazemeter.com:443/api/v4/tests/testID/start
-H "Content-Type: application/json" -H "x-api-key:
65f7f52eb362c3ec2e31"
```

```

{
  "test": {
    "id": 5396270,
    "name": "FirstJMeterTest-API",
    "userId": 334563,
    "created": 1487061745,
    "updated": 1487063961,
    "projectId": 157926,
    "lastUpdatedById": 334563,
    "configuration": {
      "location": "us-east-1",
      "duration": 20,
      "type": "jmeter",
      "concurrency": 50,
      "dedicatedIpsEnabled": "false",
      "canControlRampup": false,
      "targetThreads": 50,
      "plugins": {
        "splitCSV": {
          "enabled": "false"
        },
        "reportEmail": {
          "enabled": true
        }
      },
      "jmeter": {
        "filename": "file.jmx",
        "version": "3.1",
        "override": {
          "rampup": 300,
          "iterations": -1,
          "duration": 20
        }
      }
    }
  }
}

kumar12@KUMAR12-7510 MINGW64 ~
$ curl -X POST https://a.blazemeter.com:443/api/v4/tests/testID/start -H "Content-Type: application/json" -H "x-api-key: f"

```

Submit a Support Request
My Support Tickets

(0)/start'.format(test_id), headers= {"x-api-key": "65f7f52eb362c3ec2e31"})
res = testStsrt(<testID>)

Support

26. In the CURL command, paste the API key that you have copied earlier. To identify and run the exact load test that you want to run, you must use the appropriate Test ID.

```

{
  "test": {
    "id": 5396270,
    "name": "FirstJMeterTest-API",
    "userId": 334563,
    "created": 1487061745,
    "updated": 1487063961,
    "projectId": 157926,
    "lastUpdatedById": 334563,
    "configuration": {
      "location": "us-east-1",
      "duration": 20,
      "type": "jmeter",
      "concurrency": 50,
      "dedicatedIpsEnabled": "false",
      "canControlRampup": false,
      "targetThreads": 50,
      "plugins": {
        "splitCSV": {
          "enabled": "false"
        },
        "reportEmail": {
          "enabled": true
        }
      },
      "jmeter": {
        "filename": "file.jmx",
        "version": "3.1",
        "override": {
          "rampup": 300,
          "iterations": -1,
          "duration": 20
        }
      }
    }
  }
}

kumar12@KUMAR12-7510 MINGW64 ~
$ curl -X POST https://a.blazemeter.com:443/api/v4/tests//start -H "Content-Type: application/json" -H "x-api-key: o1ln5dcjb1ys9g0jztjg"

```

27. To identify the Test ID for a specific test, navigate back to the BlazeMeter UI, click Test tab and select the test from the drop-down list. On selecting the test, go to Address bar and you can find test ID. It is a numeric value displayed next to "tests/<testID>".

The screenshot shows the BlazeMeter web interface. In the top navigation bar, the 'Tests' tab is selected. Below it, a list of 'Recently Active Tests' is displayed. The first item, 'FirstJMeterTest-API', is highlighted with a red box. The address bar at the top contains the URL <https://a.blazemeter.com/app/#/settings/personal>. On the left side, there's a sidebar with 'Settings' and 'Personal' sections.

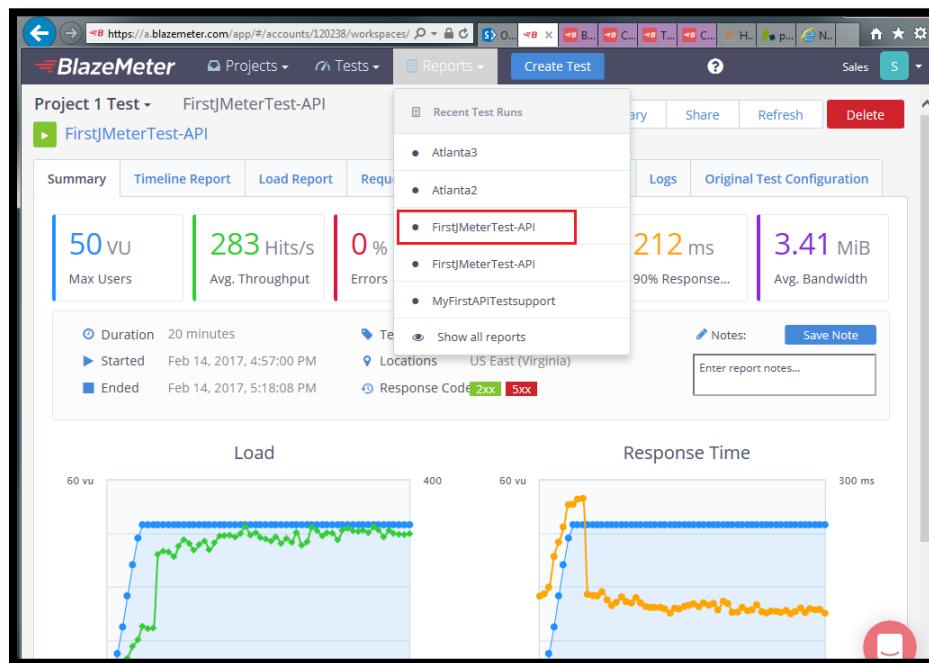
This screenshot shows the 'Test Configuration' tab for the 'FirstJMeterTest-API' test. A context menu is open over the test name, with 'Copy' highlighted. The address bar shows the URL <https://a.blazemeter.com/units/120238/workspaces/113599/projects/157926/tests/157926/test-configurations/157926-test-configuration>. The interface includes a 'Configuration Highlights' section, a 'JMeter test' configuration panel, and two charts: 'AVERAGE RESPONSE TIME' and 'ERRORS PERCENTAGE'.

28. Press Enter and see the response below to validate that parameters are displayed with zero errors. To observe the results, navigate back to the BlazeMeter UI and click Reports. Select the report from the drop-down list.

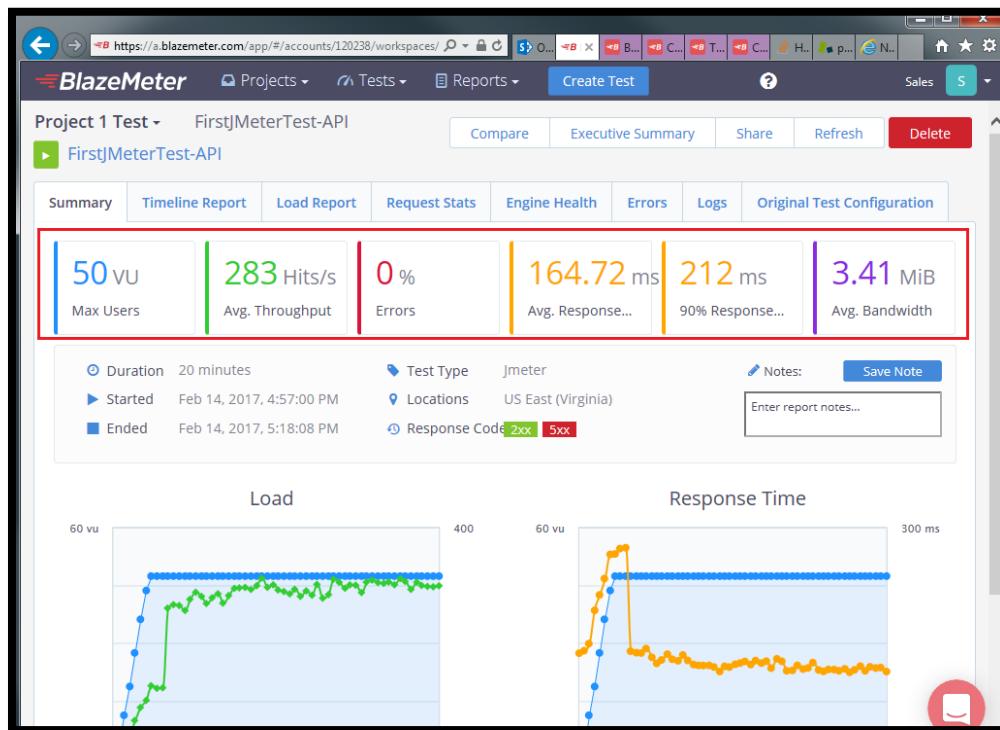
```

kumar12@KUMAR12-7510 MINGW64 ~
$ curl -X POST https://a.blazemeter.com:443/api/v4/tests/5396270/start -H "Content-Type: application/json" -H "x-api-key: o1n5dcjblys9g0jztjt9"
% Total % Received % Xferd Average Speed Time Time Current
100 170 100 170 0 0 107 0 0:00:01 0:00:01 --:--:-- 108[{"error": {"code": 400, "message": "Bad Request: Not enough dedicated IPs, needed: 1 available: 0"}, "api_version": 4, "result": null}
kumar12@KUMAR12-7510 MINGW64 ~
$ curl -X POST https://a.blazemeter.com:443/api/v4/tests/5396270/start -H "Content-Type: application/json" -H "x-api-key: o1n5dcjblys9g0jztjt9"
% Total % Received % Xferd Average Speed Time Time Current
100 436 100 436 0 0 266 0 0:00:01 0:00:01 --:--:-- 268[{"api_version": 4, "error": null, "result": {"id": 15955093, "name": "FirstJMeterTest-API", "userId": "o1n5dcjblys9g0jztjt9", "ownerUserId": 334563, "canControlRampup": false, "targetThreads": 30, "created": "2017-02-14T20:00:24Z", "updated": "2017-02-14T20:00:24Z", "testId": 5396270, "projectId": 157926, "sessionsId": [{"r-v3-58a2d09aab736"}]}]
kumar12@KUMAR12-7510 MINGW64 ~
$ |

```



29. On selecting the report, a window opens where you can see the engines getting ready and once the engines are ready, the test starts automatically. Once the JMeter test starts, you can observe different types of reports being displayed for the Multi Test along with the average throughput time, average response time, average bandwidth and maximum virtual users. Even the Load and Response Time graphs keep changing with number of virtual users.



Lab 5 - Record and Run a JMeter Test using Chrome Browser Plugin

Goals This document provides brief instructions for the attendees to record and run a JMeter Test using Chrome browser plugin in BlazeMeter.

Scenario You will accomplish this by completing the following tasks:

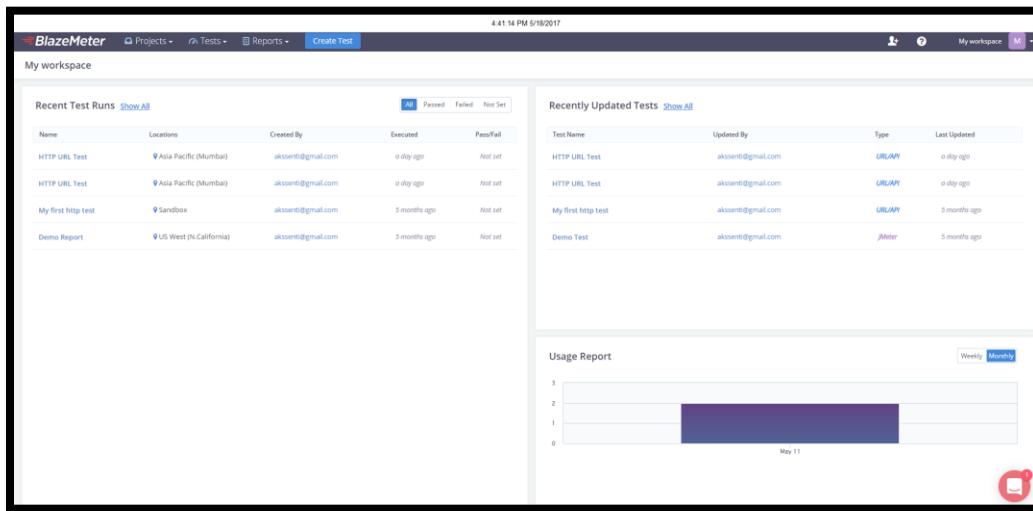
- Add the BlazeMeter Chrome extension to your browser
- Transfer and run the JMeter script to BlazeMeter UI
- Generate an Executive Summary report

Time 20 minutes

Instructions:

Run a simple URL/API Test and Create a Report

1. Open an instance of Google Chrome browser and go to <http://www.blazemeter.com>. If you have already created a BlazeMeter account, use your BlazeMeter account email and password to log into BlazeMeter.

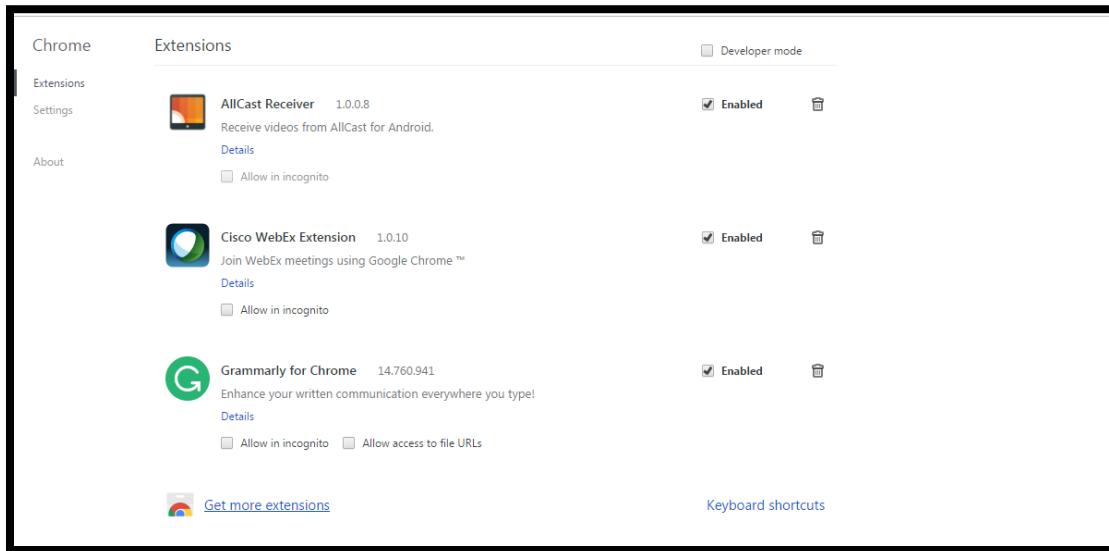


The screenshot shows the BlazeMeter dashboard. On the left, there's a sidebar with 'My workspace' and a 'Recent Test Runs' section containing five entries. The first four are 'HTTP URL Test' with locations 'Asia Pacific (Mumbai)', 'Sandbox', and 'US West (N California)'. The fifth entry is 'Demo Report'. The right side features a 'Recently Updated Tests' section with three entries: 'HTTP URL Test' (uploaded by akssent@gmail.com), 'My first http test' (uploaded by akssent@gmail.com), and 'Demo Test' (uploaded by akssent@gmail.com). Below these sections is a 'Usage Report' chart showing activity over time, with a blue bar extending from May 11. A red notification badge with the number '1' is visible in the bottom right corner of the dashboard area.

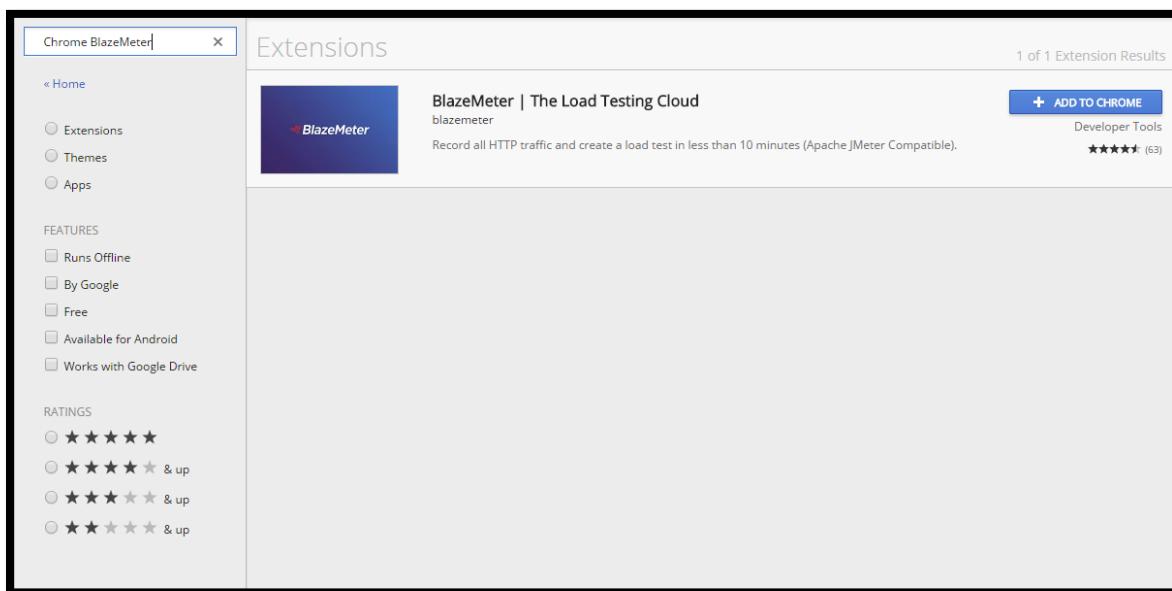
Let's now see how to add a BlazeMeter's chrome extension to run the JMeter tests. This extension records all the HTTP/S requests that your browser makes, creates a JMeter script, and automatically uploads it to BlazeMeter, where you can execute it with a single click

2. Open a new tab in the chrome browser.

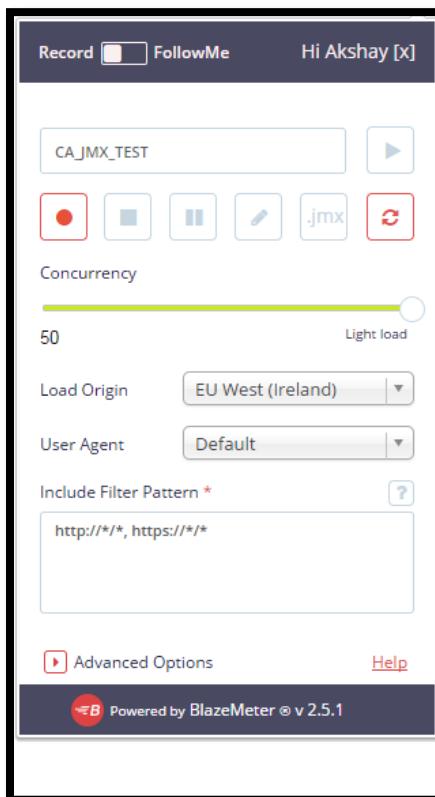
3. Now, click '**More Options**' icon available at top-right corner of the browser, and then select **Settings** from the drop-down menu. Then, select **Extensions**.



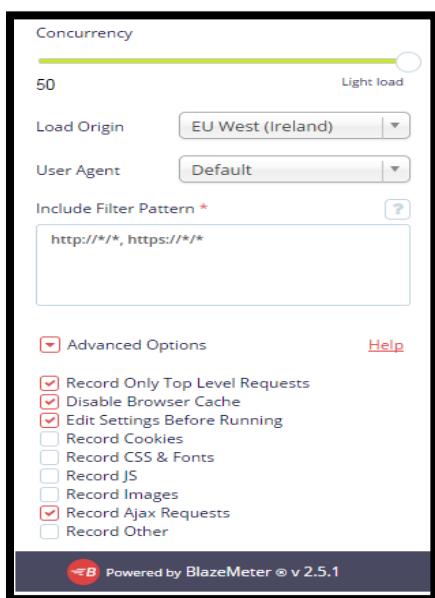
4. Click **more extensions** to find new extensions.
5. Type **Chrome BlazeMeter** in the search bar and press 'Enter'. You can now see the BlazeMeter extension. Click '**Add To Chrome**' to add the extension.



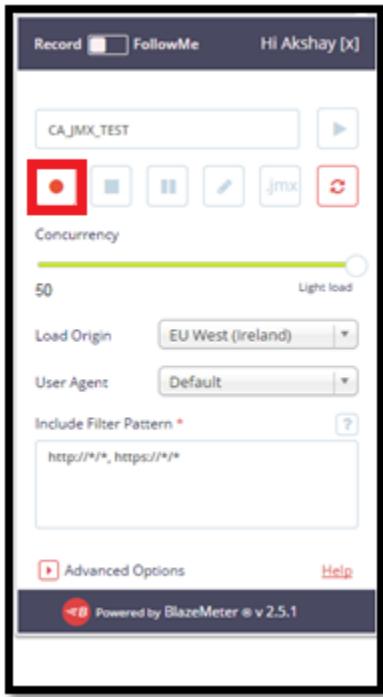
6. Now, click BlazeMeter icon available in browser's address bar, a dialog box opens in the same window.
7. Let's call the test "CA_JMX_TEST_DEMO".



8. Now, click **Advanced Options** to view the additional settings available.
9. Scroll down and select the '**Edit Settings Before Running**' checkbox.

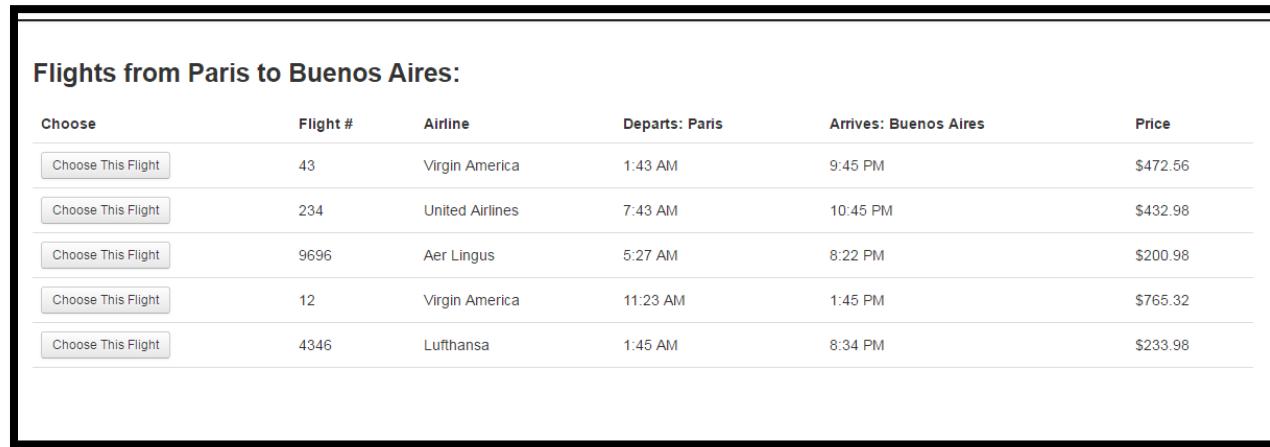


10. Click the red circle button to start recording the traffic and open a new tab to enter the website URL.



- Let's enter the Blaze Demo URL (or your customer's website URL) to load the traffic. The Blazdemo Simple Travel Agency website opens.

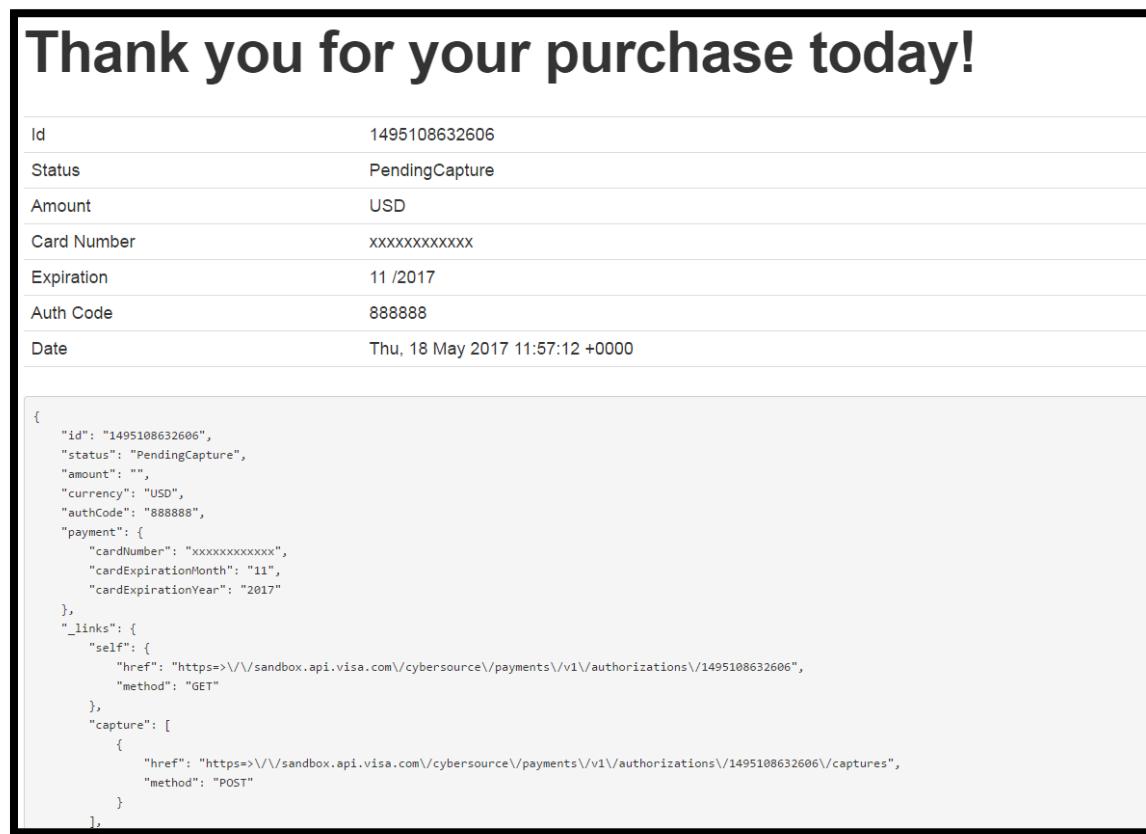
- Now click 'Find Flights'. and click on 'Choose This Flight' button from the list.



Flights from Paris to Buenos Aires:

Choose	Flight #	Airline	Departs: Paris	Arrives: Buenos Aires	Price
Choose This Flight	43	Virgin America	1:43 AM	9:45 PM	\$472.56
Choose This Flight	234	United Airlines	7:43 AM	10:45 PM	\$432.98
Choose This Flight	9696	Aer Lingus	5:27 AM	8:22 PM	\$200.98
Choose This Flight	12	Virgin America	11:23 AM	1:45 PM	\$765.32
Choose This Flight	4346	Lufthansa	1:45 AM	8:34 PM	\$233.98

13. In the next page, click '**Purchase Flight**'.
14. You are directed to flight purchase confirmation page. Now, click the '**BlazeMeter Chrome extension**' icon.



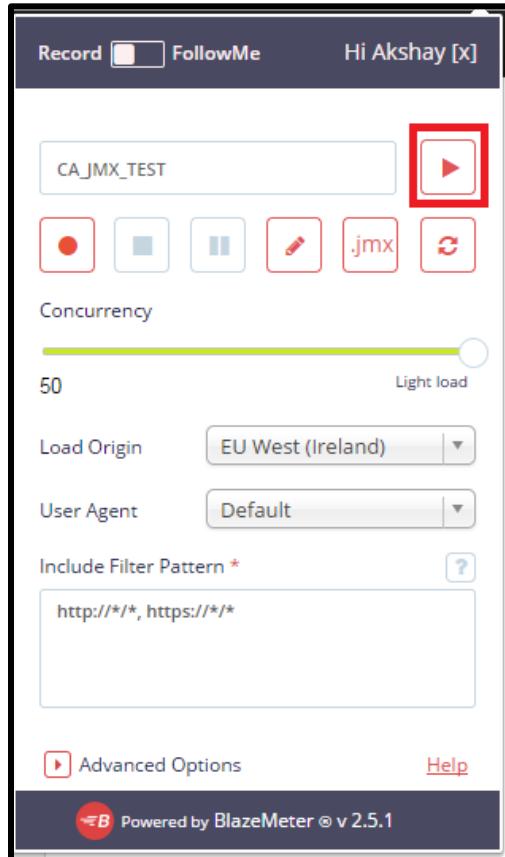
Thank you for your purchase today!

Id	1495108632606
Status	PendingCapture
Amount	USD
Card Number	XXXXXXXXXXXX
Expiration	11 /2017
Auth Code	888888
Date	Thu, 18 May 2017 11:57:12 +0000

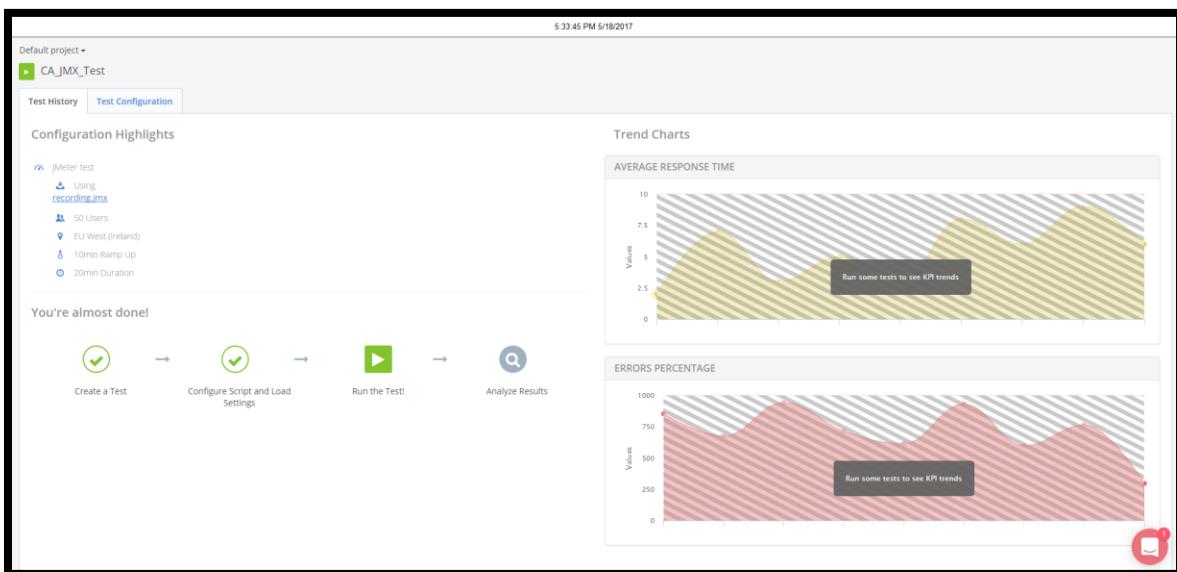
```
{
  "id": "1495108632606",
  "status": "PendingCapture",
  "amount": "",
  "currency": "USD",
  "authCode": "888888",
  "payment": {
    "cardNumber": "XXXXXXXXXXXX",
    "cardExpirationMonth": "11",
    "cardExpirationYear": "2017"
  },
  "_links": {
    "self": {
      "href": "https://sandbox.api.visa.com/cybersource/payments/v1/authorizations/1495108632606",
      "method": "GET"
    },
    "capture": [
      {
        "href": "https://sandbox.api.visa.com/cybersource/payments/v1/authorizations/1495108632606/captures",
        "method": "POST"
      }
    ]
  }
}
```

15. A pop-up dialog box opens. Then, click the red square button to stop the recording.

16. Did you notice that a message 'Recording Stopped' is displayed at the top and it confirms that the recording has stopped successfully?
17. Now, you must transfer the recorded JMeter script to BlazeMeter UI. To transfer the script, click 'Red Arrow' button. This starts a BlazeMeter session in a new browser tab.



18. From the BlazeMeter UI, click the '**Test Configuration**' tab to modify the test configuration details.



19. To reconfigure the test case details, click the '**JMeter version**' and select '**3.1**' from the drop-down list.

The screenshot shows the 'Test Configuration' tab for a project named 'CA_JMX_Test'. The 'JMeter Test' section contains a recording file named 'recording.jmx'. In the 'Load Configuration' section, the 'JMeter Version' dropdown is circled in red and set to '3.1'. Other settings include 'Java Version: 1.8', 'Sandbox Mode - free low-scale test for debugging purposes' checked, 'Originate load from: EU West (Ireland) - Amazon Web Services' selected, and 'Users' set to 50. Below these are sections for 'Test Failure Criteria', 'DNS - Hostname Override', and 'JMeter Properties'.

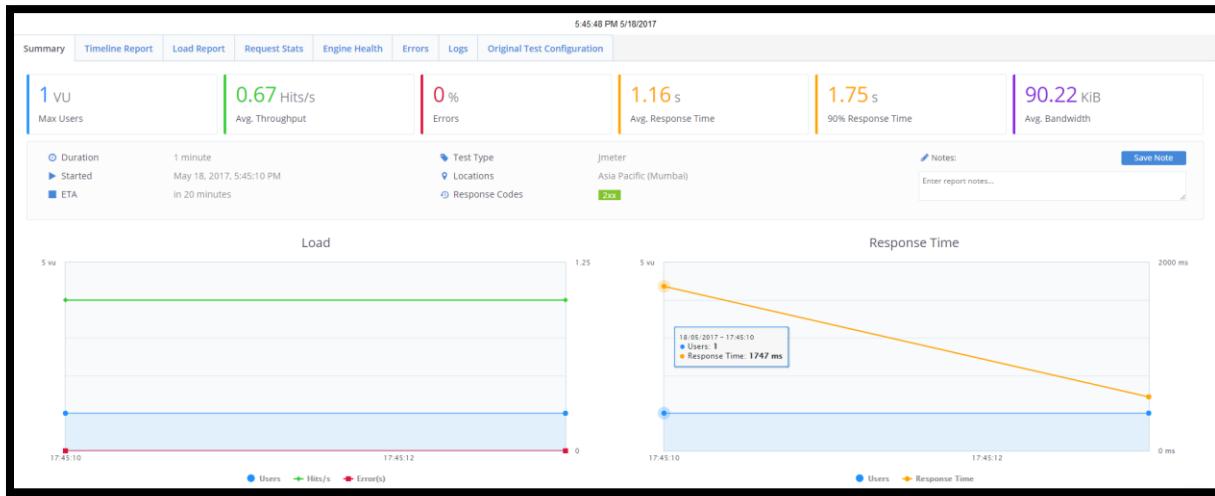
20. You must further change the server to point to the location from where the load is generated. To change it, click originate load from and select the '**Asia Pacific (Mumbai)**' from the drop-down menu.

21. You can change threads by selecting 'Users' and entering the appropriate value. Based on your account privileges, you can add more engines. If you have a free-tier account, you can run the test with default users.

The screenshot shows the 'Test Configuration' tab for the same project and test case. The 'JMeter Version' dropdown is set to '3.1'. The 'Load Configuration' section shows 'Originate load from: Asia Pacific (Mumbai) - Amazon Web Services' selected. Other settings include 'Users' set to 50, 'Engines' set to 0, 'Threads' set to 50, and 'Limit RPS' set to 1. A 'Bumpers (sec)' setting is also present at 600.

22. Click **Save** to finalize the changes to your test case.

23. Then, click the '**Play**' button to start the performance test of the website that you just recorded by using the BlazeMeter's Chrome extension. This JMeter test uses the JMeter script that you captured using the plugin's recording feature.



24. Once the test starts, you can observe different types of reports, being displayed for the test along with the average throughput time, average response time, average bandwidth, and maximum virtual users. Even the Load and Response Time graphs, keep changing with number of virtual users.
25. Once, the test is complete, in the reports window, click '**Executive Summary**' report.
26. The report opens in new tab and you can observe the top 5 slow responses, top 5 errors, test setup details, summarized aggregate report, summarized Error report, graphs presenting users, Response times and hits per second.

Lab 6 - BlazeMeter – Jenkins Integration Setup

Goals Setup Jenkins integration with BlazeMeter

Scenario Integrating BlazeMeter with a Continuous Integration (CI) solution like Jenkins enables shifting Performance Test *Left* to execute performance tests much earlier in the SDLC. This helps detect and remediate performance issues while the component is being developed. BlazeMeter supports other CI tools like TeamCity and Bamboo. This lab will be focused around integration with Jenkins that will help execute BlazeMeter performance test as part of a code build. The actual execution of performance tests will be done during subsequent labs.

You will accomplish this by completing the following tasks:

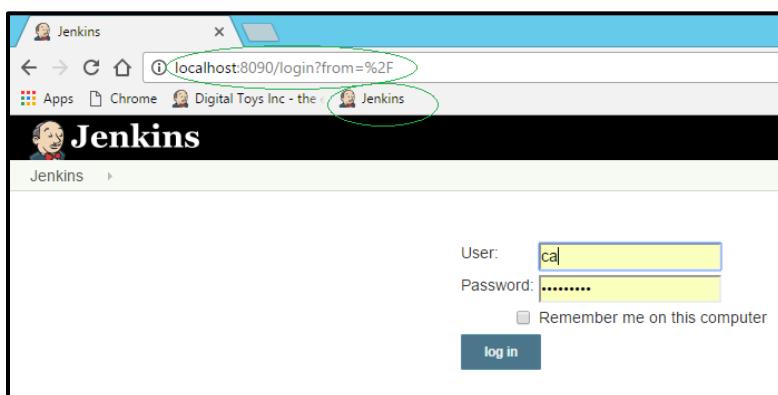
- Install BlazeMeter Plugins for Jenkins
- Setup Jenkins Credentials for login to BlazeMeter
- Accessing BlazeMeter Application
- Complete Setup Jenkins Credentials for login to BlazeMeter

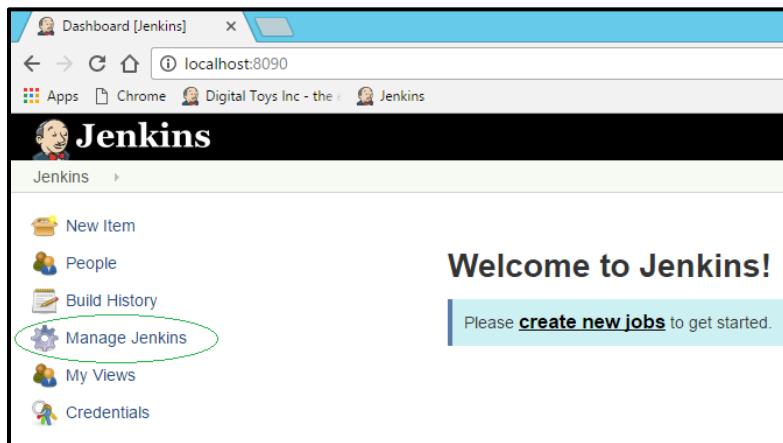
Time 20 minutes

Instructions:

Install BlazeMeter Plugins for Jenkins

1. You must download the latest version of Jenkins Setup file and install Jenkins your local machine. If you have Jenkins readily installed, then enter your valid login credentials and click **Login**.

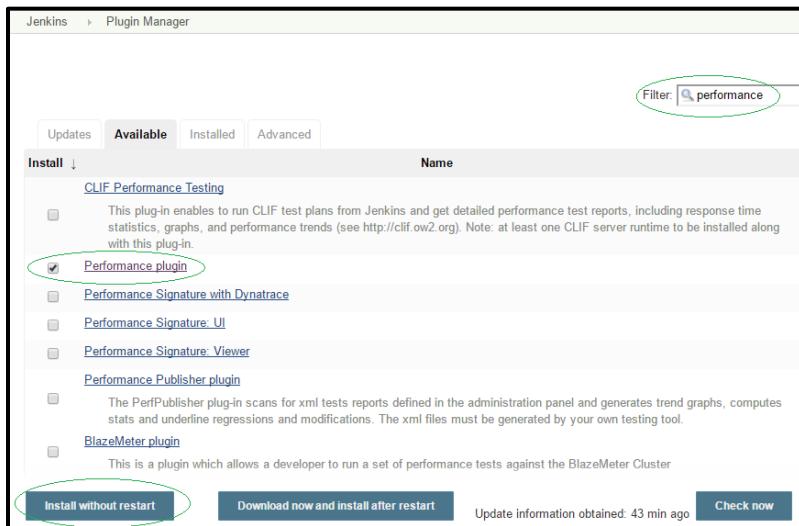


2. Click on **Manage Jenkins**.3. Click on **Manage Plugins** and then **Available Tab**.

A screenshot of the 'Manage Jenkins' page. The title bar says 'Jenkins'. The left sidebar includes 'New Item', 'People', 'Build History', 'Manage Jenkins' (underlined), 'My Views', and 'Credentials'. Below the sidebar are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area is titled 'Manage Jenkins' and lists several configuration options: 'Configure System', 'Configure Global Security', 'Configure Credentials', 'Global Tool Configuration', 'Reload Configuration from Disk', and 'Manage Plugins' (which is circled in green). A note below says 'Add, remove, disable or enable plugins that can extend Jenkins'.

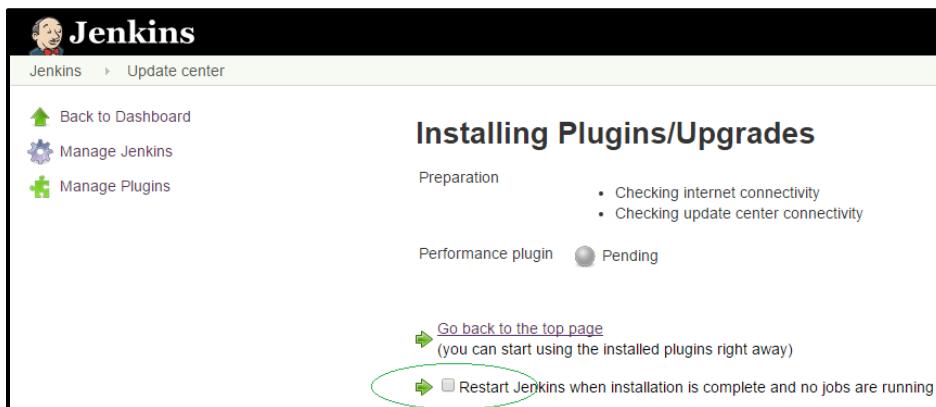
A screenshot of the 'Plugin Manager' page. The title bar says 'Jenkins > Plugin Manager'. The left sidebar has 'Back to Dashboard' and 'Manage Jenkins'. At the top right is a search bar with a magnifying glass icon. Below the sidebar are tabs: 'Updates' (circled in green), 'Available' (underlined and circled in green), 'Installed', and 'Advanced'. A 'Filter' input field with a magnifying glass icon is also present. The main content area shows a table of available plugins under '.NET Development': 'CCM Plug-in' (version 3.1), 'FxCop Runner plugin' (version 1.1), and 'MSBuild Plugin' (version 1.27).

4. In the **Filter** text box, enter **performance** to locate the BlazeMeter performance reporting plugin for Jenkins.



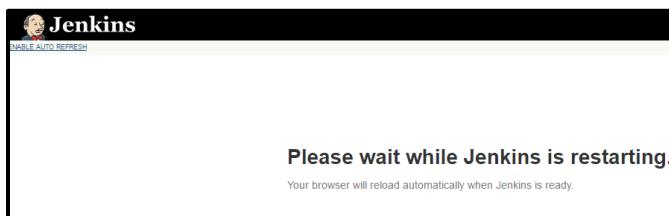
The screenshot shows the Jenkins Plugin Manager interface. The 'Available' tab is selected. A search bar at the top right contains the text 'performance'. Below the tabs, there's a table with columns 'Install' and 'Name'. The 'Performance plugin' is listed and has a checked checkbox next to it. Other listed plugins include 'CLIF Performance Testing', 'Performance Signature with Dynatrace', 'Performance Signature UI', 'Performance Signature Viewer', 'Performance Publisher plugin', and 'BlazeMeter plugin'. At the bottom, there are three buttons: 'Install without restart' (highlighted with a green oval), 'Download now and install after restart', and 'Check now'.

5. Select on the checkbox next to Performance plugin and click on Install without restart.



The screenshot shows the Jenkins Update center under the 'Installing Plugins/Upgrades' section. On the left, there are links for 'Back to Dashboard', 'Manage Jenkins', and 'Manage Plugins'. On the right, there's a 'Preparation' section with two bullet points: 'Checking internet connectivity' and 'Checking update center connectivity'. Below that is a 'Performance plugin' entry with a status of 'Pending'. At the bottom, there are two links: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running' (which is highlighted with a green oval).

6. Select the checkbox next to Restart. This will restart Jenkins after plugin installation.



The screenshot shows the Jenkins dashboard with a central message box. The message reads 'Please wait while Jenkins is restarting.' and includes the sub-instruction 'Your browser will reload automatically when Jenkins is ready.'

5. Repeat steps 1-6 for **BlazeMeter plugin** to install the BlazeMeter plugin for Jenkins.

Jenkins Plugin Manager

Available

BlazeMeter plugin

This is a plugin which allows a developer to run a set of performance tests against the BlazeMeter Cluster

Install without restart

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

BlazeMeter plugin Installing

Restarting Jenkins Pending

[Go back to the top page](#)
(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

- This may take few minutes to complete. If the screen doesn't refresh, and you do not see the Jenkins login-screen, click on Go back to the top page. This will take you to the Jenkins login screen. Go to the Installed plugins tab and make sure BlazeMeter and Performance plugins are installed successfully under Installed Tab.

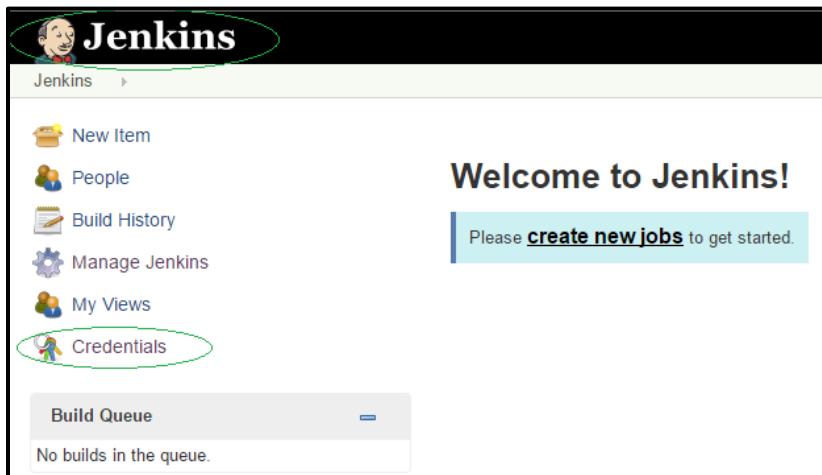
Plugin Manager

Installed

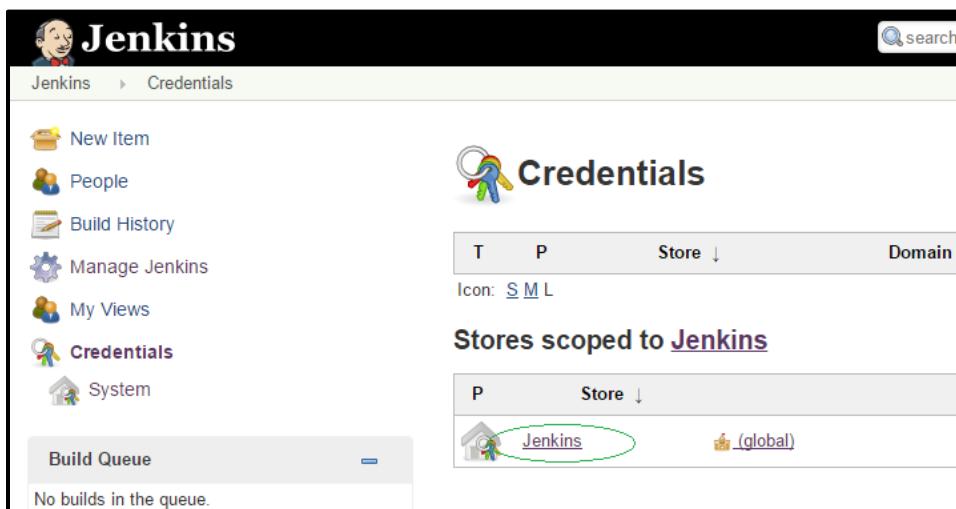
Enabled	Name	Version	Previously installed version	Pinned	Uninstall
<input checked="" type="checkbox"/>	Ant Plugin	1.4			Uninstall
<input checked="" type="checkbox"/>	Authentication Tokens API Plugin	1.3			Uninstall
<input checked="" type="checkbox"/>	BlazeMeter plugin	2.7			Uninstall

Setup Jenkins Credentials for login to BlazeMeter

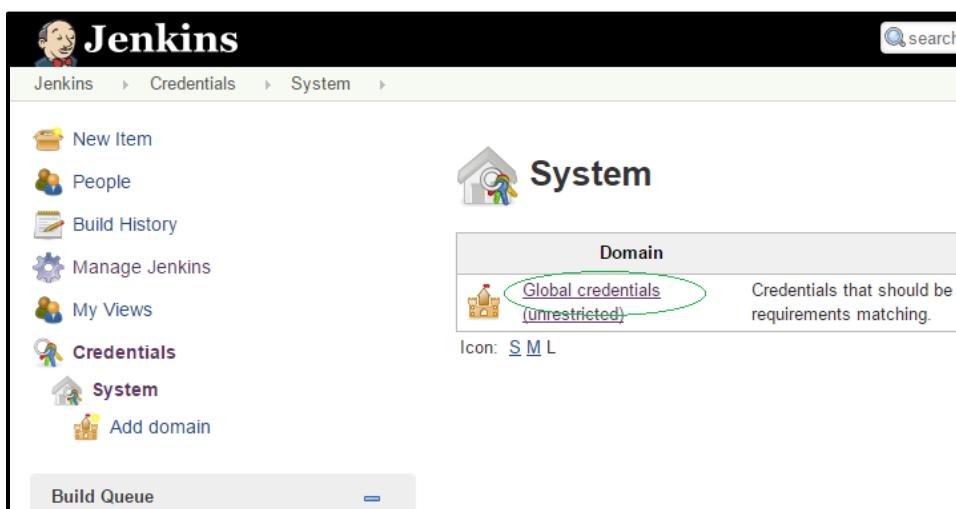
- Click on Jenkins image on top left corner of the Jenkins webpage on your browser to go to Jenkins home page or Login to Jenkins, using your login credentials. Then click on **Credentials > Jenkins > Global Credentials > Add Credentials**.



The screenshot shows the Jenkins home page. On the left sidebar, there is a 'Credentials' link, which is highlighted with a green oval. The main content area displays a 'Welcome to Jenkins!' message and a 'Please [create new jobs](#) to get started.' button.



The screenshot shows the 'Credentials' page under the 'Jenkins' scope. The 'Stores scoped to Jenkins' section lists two entries: 'Jenkins' (highlighted with a green oval) and '(global)'.



The screenshot shows the 'System' page. In the 'Domain' section, the 'Global credentials (unrestricted)' link is highlighted with a green oval. A tooltip explains that these are 'Credentials that should be available to all jobs matching requirements.'

The screenshot shows the Jenkins Global credentials (unrestricted) page. At the top left is a 'Back to credential domains' link and an 'Add Credentials' button, which is highlighted with a green oval. The main title is 'Global credentials (unrestricted)' with a castle icon. Below it is a table header with columns 'Name' and 'Kind'. A message at the bottom says 'This credential domain is empty. How about adding some?' with icons for 'S' (Small), 'M' (Medium), and 'L' (Large).

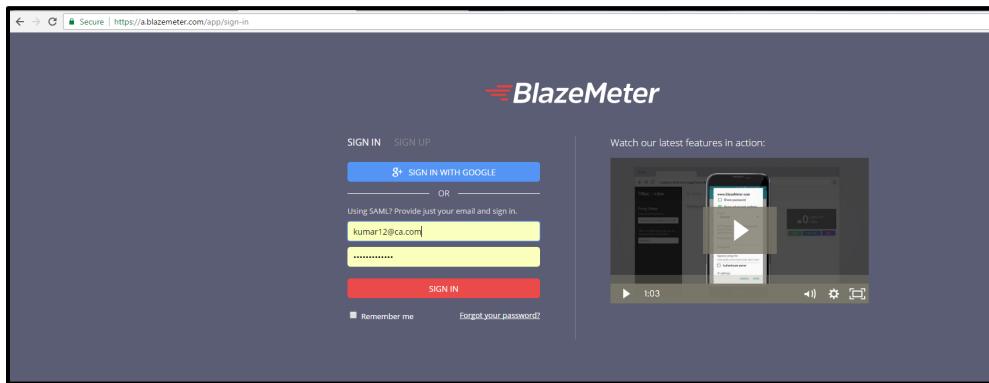
9. In the Add Credentials window, select BlazeMeter API Key in Kind drop-down menu. Obtain your BlazeMeter account API key from BlazeMeter application. Copy and paste this key in the empty text box and click on Test BlazeMeter API Key.

The screenshot shows the Jenkins Add Credentials window. The 'Kind' dropdown is set to 'BlazeMeter API Key', which is highlighted with a green oval. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. At the bottom right is an 'OK' button and a large empty text input field. To its right is a 'Test BlazeMeter API Key' button, also highlighted with a green oval.

10. In the Add Credentials screen, select BlazeMeter API Key in Kind drop-down menu. Obtain your BlazeMeter account API key from BlazeMeter application using step 10. Copy and paste this key in the empty text box and click on Test BlazeMeter API Key.

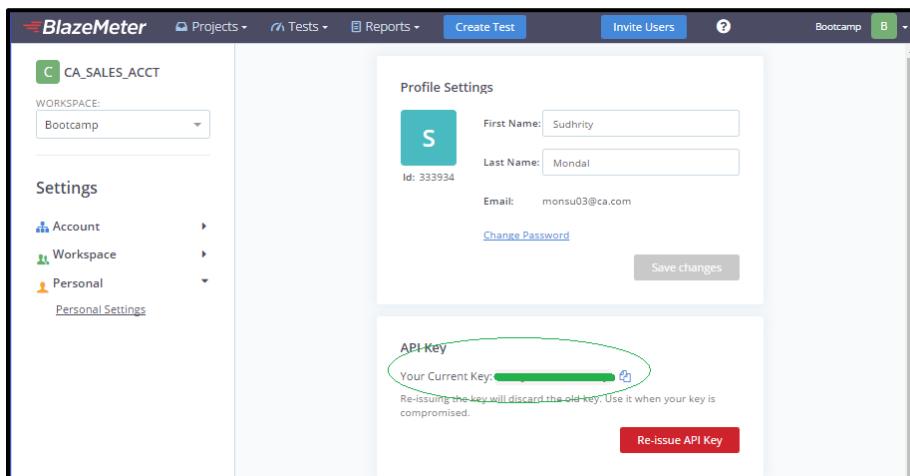
Accessing BlazeMeter Application

11. Enter <https://a.blazemeter.com/app/sign-in> in your browser.
12. At the login prompt, enter the following details (If you are an existing user):



Field	Value
User Name	Enter your registered user name
Password	Enter a valid password

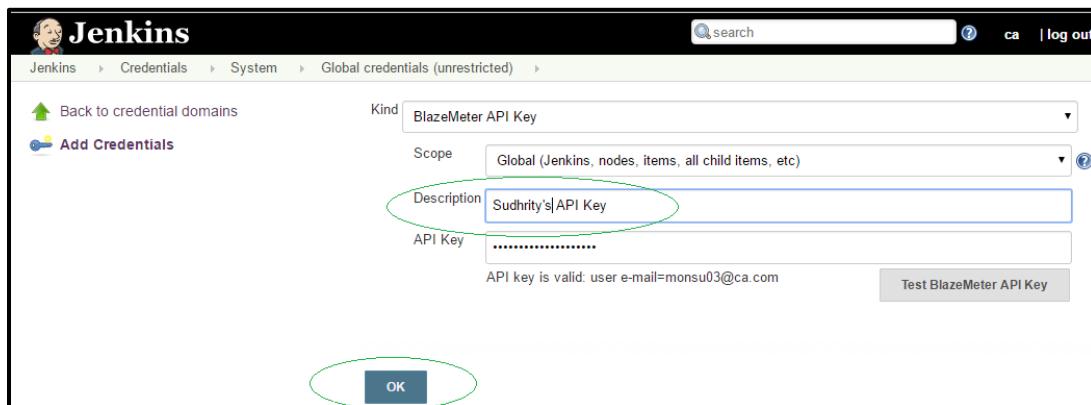
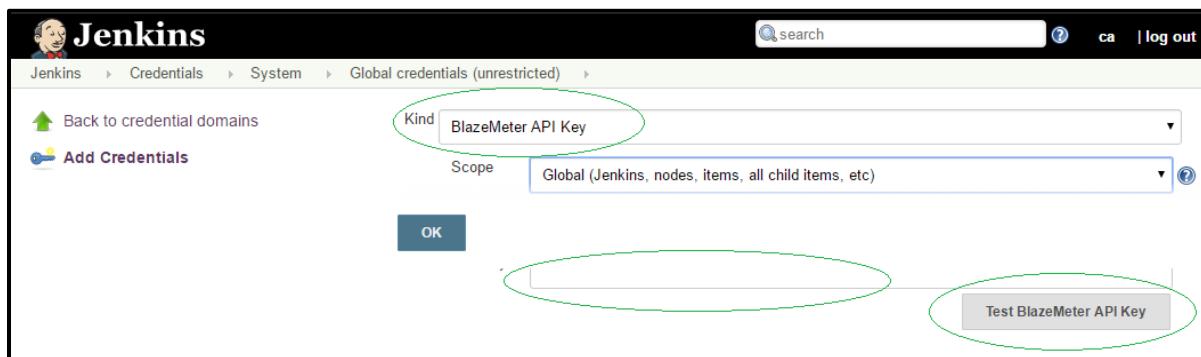
13. On clicking Sign In, the BlazeMeter application opens. Click on the My Workspace (or whatever you may have) dropdown arrow on top-right corner.
14. In a similar fashion, click on Settings.



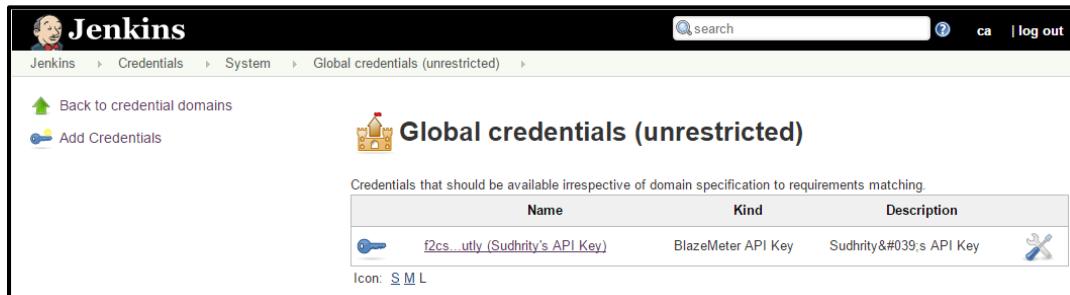
15. Copy the text next to Your Current Key into Clipboard. You may also use the icon next to it, to copy the key to Clipboard.

Complete Setup Jenkins Credentials for login to BlazeMeter

16. In Jenkins Add Credentials window, paste the API Key from Clipboard to the empty text field and click on Test BlazeMeter API Key. Your machine must be connected to the internet to perform this test.



17. Once the API Key is verified and valid, enter a Description for the key, and click on OK to save the key.



18. This completes the BlazeMeter – Jenkins integration setup lab.