

# Performance Test Tools Introduction to JMeter





### Contents

1. Introduction

- 2. JMeter Advantages
- 3. How JMeter Works
- 4. Best Practice for JMeter Tests
- **5. JMeter Alternatives**
- 6. References





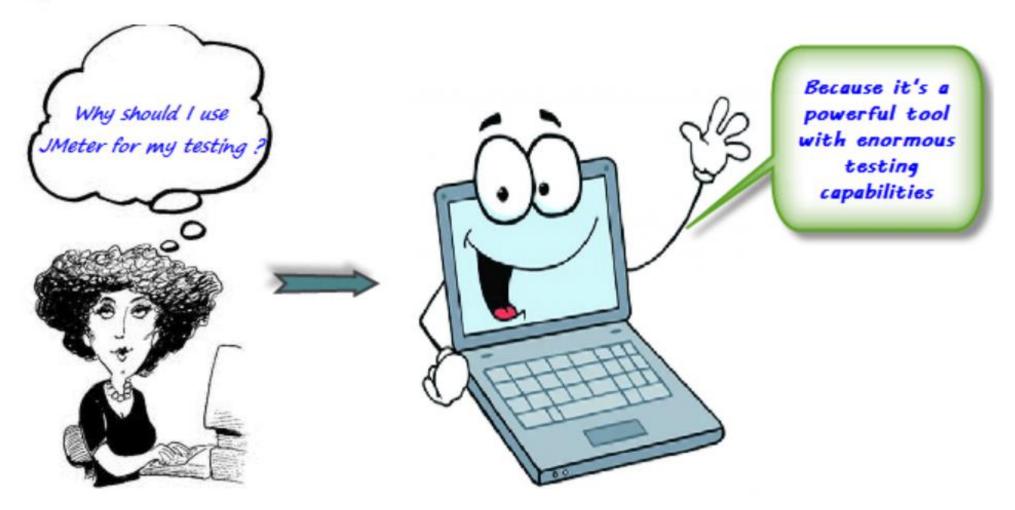
## Introduction

The **Apache JMeter<sup>TM</sup>** is pure <u>Java</u> open source software, which was first developed by Stefano Mazzocchi of the <u>Apache</u> Software Foundation, designed to load test functional behavior and measure performance.

Apache Jmeter is a popular open source performance testing tool.



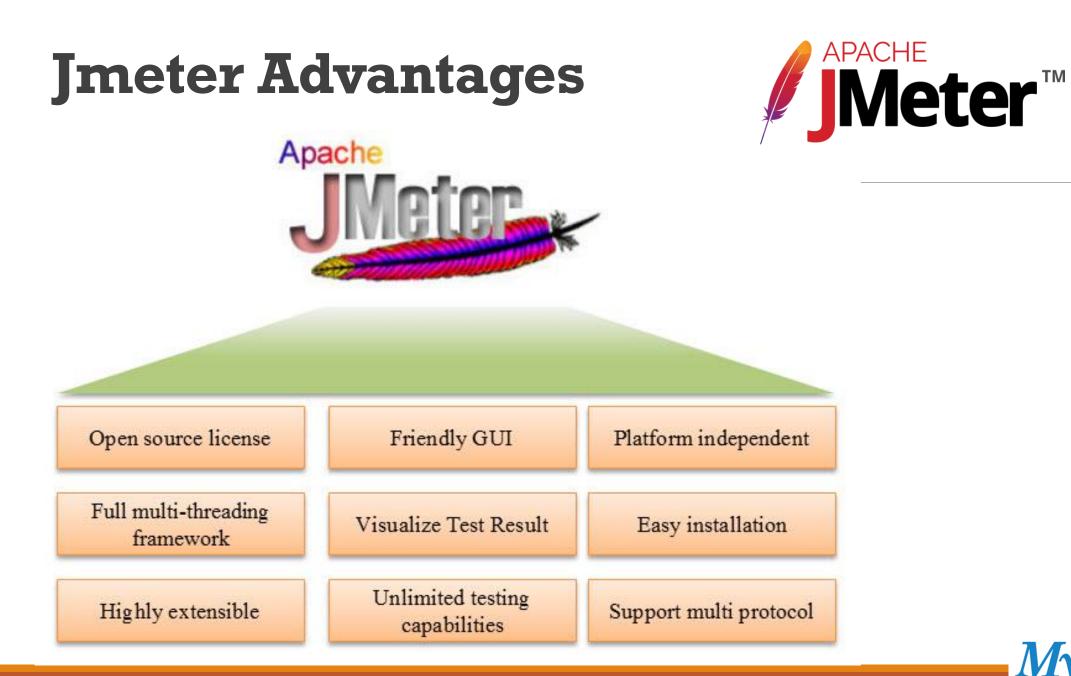
#### Why JMeter?













## **JMeter Advantages**



✓ **Open source license**: JMeter is totally free, allows developer use the source code for the development

✓ Friendly GUI: JMeter is extremely easy to use and doesn't take time to get familiar with it

✓ **Platform independent**: JMeter is 100% pure Java desktop application. So it can run on multiple platforms

 Full multithreading framework. JMeter allows concurrent and simultaneous sampling of different functions by a separate thread group

✓ **Visualize Test Result:** Test result can be displayed in a different format such as chart, table, tree and log file

Easy installation: You just copy and run the \*.bat file to run JMeter. No installation needed.

# JMeter Advantages (continue)



Highly Extensible: You can write your own tests. JMeter also supports visualization plugins allow you to extend your testing

✓ Multiple testing strategy: JMeter supports many testing strategies such as Load Testing, Distributed Testing, and Functional Testing.

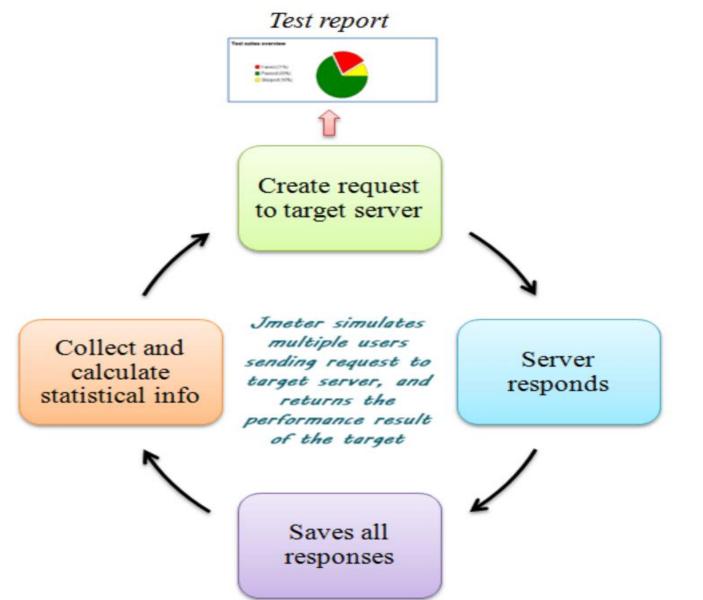
 Simulation: JMeter can simulate multiple users with concurrent threads, create a heavy load against web application under test

✓ **Support multi-protocol**: JMeter does not only support web application testing but also evaluate database server performance. All basic protocols such as HTTP, JDBC, LDAP, SOAP, JMS, and FTP are supported by JMeter

**Record & Playback** - **Record** the user activity on the browser and simulate them in a web application using JMeter

Script Test: Jmeter can be integrated with Bean Shell & <u>Selenium</u> for automated testing

### **How JMeter Works**





# **OS Support For JMeter**

JMeter is a **pure Java** application and should run correctly on any system that has a compatible Java implementation.

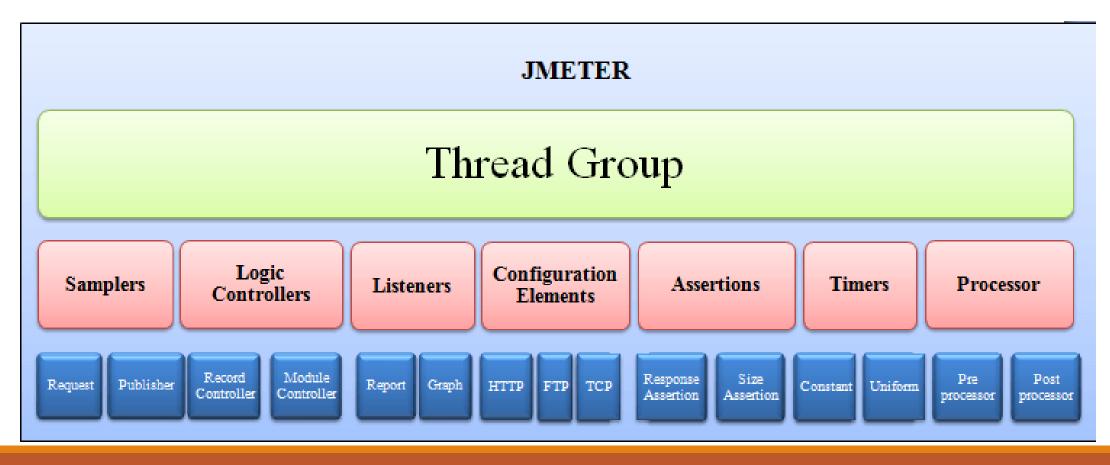
Here is the list of an operating system compatible with JMeter

- Linux
- Windows
- Mac OS
- Ubuntu



# JMeter Elements: Thread Group, Samplers, Listeners, Configuration

What is Element in JMeter?

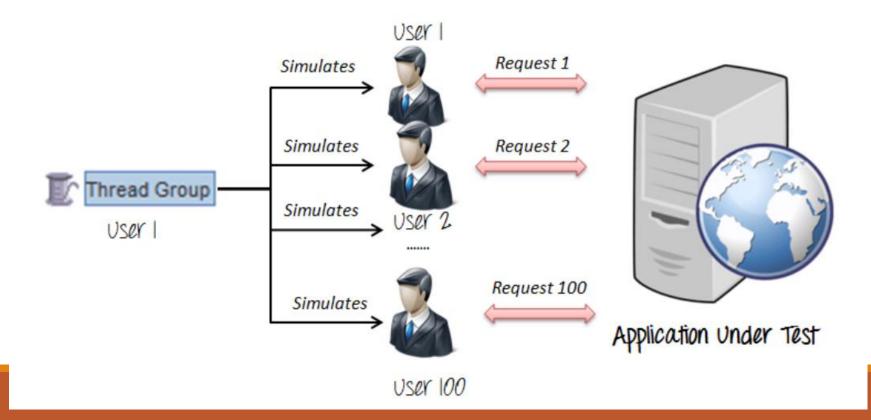


#### **Thread Group**

Thread Groups is a collection of Threads. Each thread represents one user using the application under test. Basically, each Thread simulates one real user request to the server.

The controls for a thread group allow you to Set the number of threads for each group.

For example, if you set the number of threads as 100; JMeter will create and simulate 100 user requests to the server under test



#### Samplers

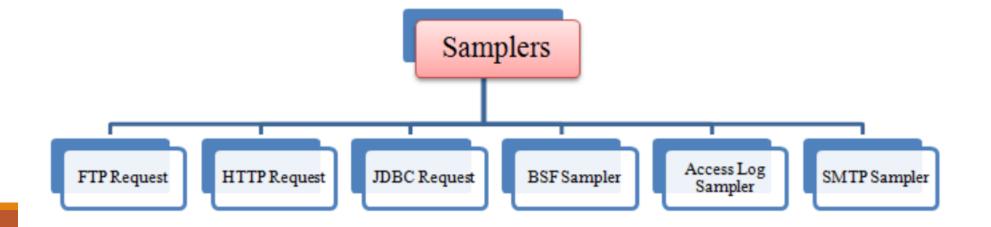
As we know already that JMeter supports testing HTTP, FTP, JDBC and many other protocols.

We already know that Thread Groups simulate user request to the server

But how does a Thread Group know which type of requests (HTTP, FTP etc.) it needs to make?

The answer is Samplers

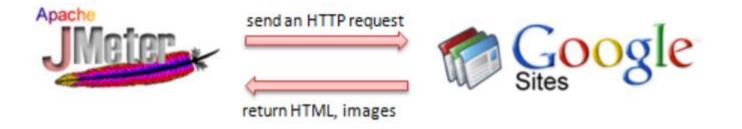
The user request could be FTP Request, HTTP Request, JDBC Request...Etc.



#### **HTTP request:**

This sampler lets you send an HTTP/HTTPS request to a web server.

Consider the example below. JMeter sends an HTTP request to Google website and retrieve HTML files or image from this website.



#### Listeners

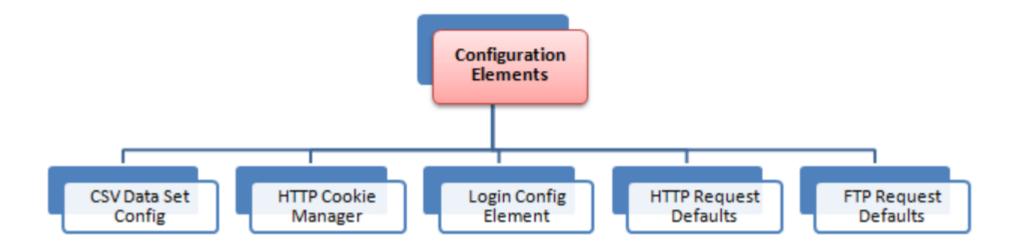
Listeners: shows the results of the test execution. They can show results in a different format such as a tree, table, graph or log file



#### **Configuration Elements**

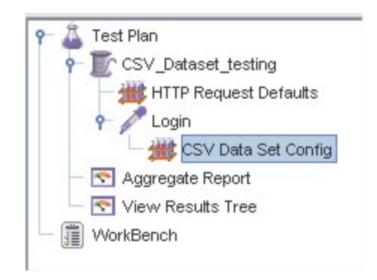
set up defaults and variables for later use by samplers.

The figure below shows some commonly used configuration elements in JMeter

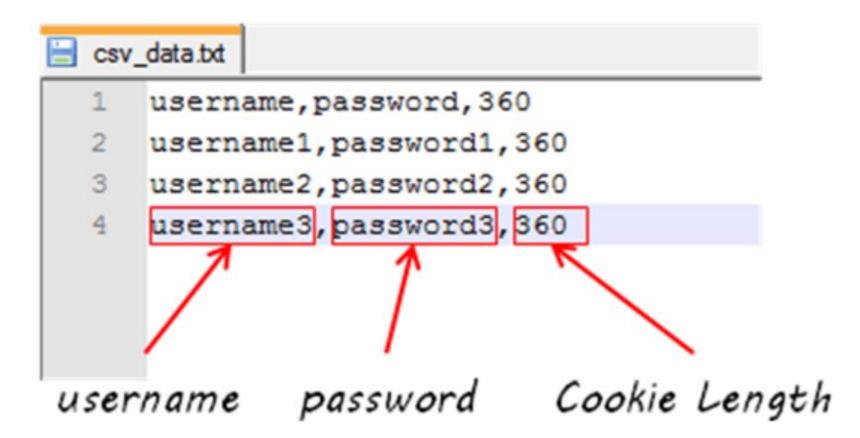


#### **CSV Data Set Config:**

Suppose you want to test a website for 100 users signing-in with different credentials. You do not need to record the script 100 times! You can parameterization the script to enter different login credentials. This login information (e.g. Username, password) could be stored in a text file. JMeter has an element that allows you to read different parameters from that text file. It is "CSV Data Set Config", which is used to read lines from a file, and split them into variables.



This is an example of CSV Data. It's a text file which contains user and password to login your target website



#### **HTTP request default**

This element lets you set default values that your HTTP Request controllers use.

For example,

You are sending 100 HTTP requests to the server google.com

You would have to manually enter server name = google.com for all these 100 requests

Instead, you could add a single HTTP request defaults with the "Server Name or IP" field = google.com

No need to type 100 times!

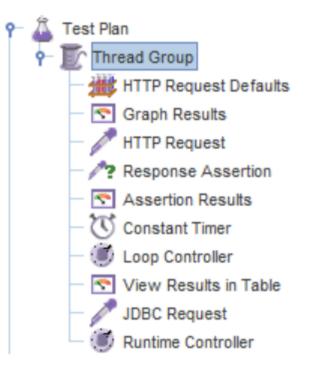
HTTP Request Defaults	
Name: HTTP Request Defaults	
Comments:	
Web Server Server Name or IP: google.com Port Number: 80	Timeouts (milliseconds) Connect: 1000 Response: 2000
HTTP Request Implementation:   Protocol [http]:	Content encoding:

#### What is a Test Plan?

Test Plan is where you add elements required for your JMeter Test.

It stores all the elements (like ThreadGroup, Timers etc) and their corresponding settings required to run your desired Tests.

The following figure shows an example of Test Plan



#### How to add Elements?

Adding Elements is the **essential** step to build a Test Plan because without adding elements, JMeter **cannot** execute your Test Plan

A Test Plan includes many Elements such as Listener, Controller, and Timer

You can add an element to test plan by right-clicking on a **Test Plan** and choose new elements from "**Add**" list.

Suppose, you want to add 2 elements to Test Plan **BeanShell Assertion** and **Java Request Default** 

- Right click Test Plan -> Add -> Assertion -> Bean Shell Assertion
- Right click Test Plan -> Add -> Config Element -> Java Request Default

Let's say, you want to remove element "**HTTP Request Defaults**", select "HTTP Request Default" -> Right click-> choose **Remove** from the context menu -> Click **Yes** to confirm delete this element on message box

P Test Plan	HTTP Request Defaults	
WorkBench 🔏	Name: HTTP Request Defaults	
	Comments:	
	Web Server Server Name or IP:	Port Number:
	HTTP Request Implementation:	I [http]: Content enco
	Path: Parameters	
	Send	d Parameters With the Request:
	Name:	Value

#### How to Save a Test Plan

Before running a test, you should save your Test Plan first. Saving your Test Plan helps you avoid unexpected error when running the test plan. Steps to saving Test plan -

- 1. File -> Save Test Plan as-> a Dialog box display
- 2. Enter a filename of Test Plan ->click Save

	he-jmeter-2.9\bin\BeanShell Assertion.jmx) -
File Edit Search Run Option	ns <u>H</u> elp
	X E E + - 4 Þ
P ▲ Test Plan ₩ Java Request Defaults	Test Plan
BeanShell Assertion	Name: Test Plan
WorkBench	Comments:
	U
	Name:

#### How to Run Test Plan

To run your single or multiple test plans, choose **Start** (Control + R) from the **Run** menu item.

Thread Group	Press Start or Ctrl+R to run a Test
Name: Thread Group	
Comments:	
Action to be taken after a	Sampler error

When JMeter is running, it shows a small green box at the right-hand end of the menu bar.



The numbers to the left of the green box are the number of **active threads** / **total number** of threads.

To Stop the Test, press **Stop** button or use short key Ctrl + '.'



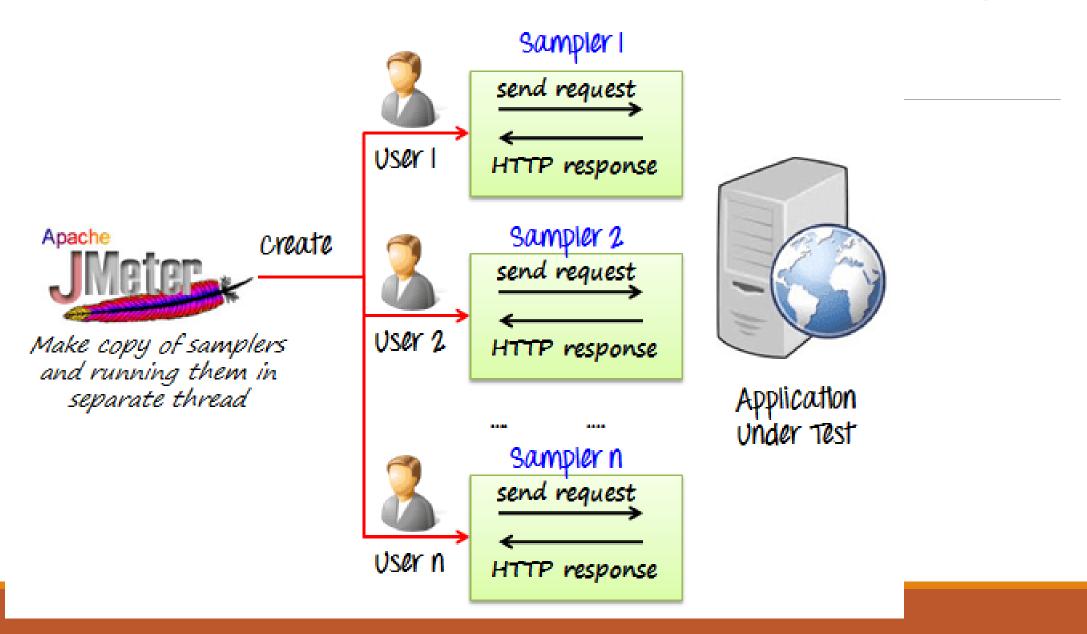
#### **Test Report**

When test execution is done, you can get the test report. The test report includes the error log file, which is saved in jmeter.log, and the test results summary. Here is a sample log file of JMeter

- 2013/08/18 08:41:12 INFO jmeter.JMeter: Copyright (c) 1998-2013 The Apache Software Foundation
- 2013/08/18 08:41:12 INFO jmeter.JMeter: Version 2.9 r1437961
- 2013/08/18 08:41:12 INFO jmeter.JMeter: java.version=1.7.0\_25
- 2013/08/18 08:41:12 INFO jmeter.JMeter: java.vm.name=Java HotSpot(TM) Client VM
- 2013/08/18 08:41:12 INFO jmeter.JMeter: os.name=Windows 7
- 2013/08/18 08:41:12 INFO jmeter.JMeter: os.arch=x86
- 2013/08/18 08:41:12 INFO jmeter.JMeter: os.version=6.1
- 2013/08/18 08:41:12 INFO jmeter.JMeter: file.encoding=Cp1252
- 2013/08/18 08:41:12 INFO jmeter.JMeter: Default Locale=English (United States)
- 2013/08/18 08:41:12 INFO jmeter.JMeter: JMeter Locale=English (United States)
- 2013/08/18 08:41:12 INFO jmeter.JMeter:

JMeterHome=C:\Nguyen\Source\_code\apache-jmeter-2.9

### How to Use JMeter for Performance & Load Testing

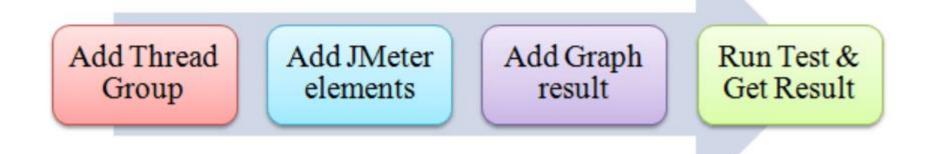


#### Create a Performance Test Plan in JMeter

In this tutorial, we are doing a performance analysis of Google.com for 1000 users

Before testing the performance of target web application, we should determine-

- Normal Load: Average number of users visit your website
- Heavy Load: The maximum number of users visit your website
- What is your **target** in this test?



#### Step 1) Add Thread Group

- 1. Start JMeter
- 2. Select Test Plan on the tree
- 3. Add Thread Group

Right click on the "Test Plan" and add a new thread group: Add -> Threads (Users) -> Thread

Group

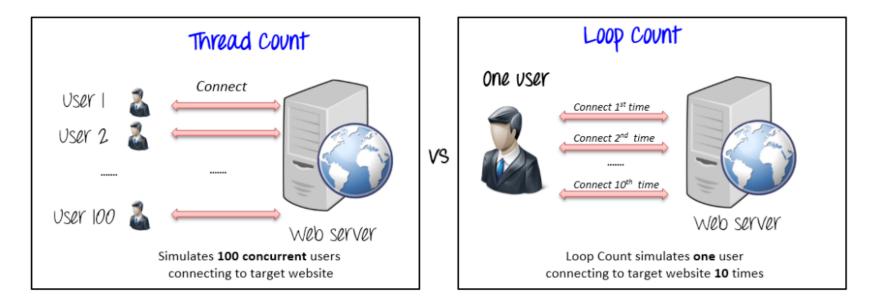
<u>ile E</u> dit Sea	rch <u>R</u> un <u>O</u> ptions <u>H</u> elp		2
l 🔒	? 🛛 📈 🛣 🗊 🗊	우 - 🛷	Þ 🗞 🔍 🗳
— 🚨 Test Plan			
- 🗊 WorkBen	Add	Threads (Users) 🕨	Thread Group
_	Paste Ctrl-V	Test Fragment	setUp Thread Group
	Reset Gui	Config Element 🔸	tearDown Thread Group
	Open	Timer 🕨	
		Pre Processors 🔸	Us
	Merge	Post Processors >	Name:
	Save Selection As	Assertions	
	Save Node As Image Ctrl-G	Listener	
	Save Screen As Image Ctrl+Shift-G		]
	Enable		
	Disable		
	Toggle Ctrl-T		
	Help		1
·			

In the Thread Group control panel, enter Thread Properties as follows:

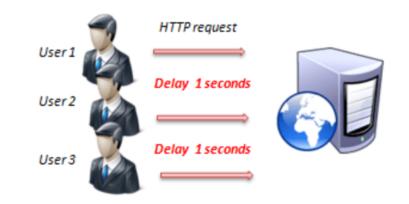
Thread Group
Name: Thread Group
Comments:
Action to be taken after a Sampler error
Continu
Thread Properties
Number of Threads (users): 100
Ramp-Up Period (in seconds): 100
Loop Count: E Forever 10
Delay Thread creation until needed
Scheduler

- Number of Threads: 100 (Number of users connects to the target website: 100)
- Loop Count: 10 (Number of time to execute testing)
- Ramp-Up Period: 100

#### The Thread Count and The Loop Counts are different.



Ramp-Up Period tells JMeter how long to **delay** before starting the next user. For example, if we have 100 users and a 100-second Ramp-Up period, then the delay between starting users would be 1 second (100 seconds /100 users)



#### Step 2) Adding JMeter elements

Now we determine what JMeter elements in this test. The elements are

#### • HTTP request Default

This element can be added by right-clicking on the Thread Group and selecting: Add - > Config Element -> HTTP Request Defaults.

Thread Group		Thread Gr	oup	
HTTP Req		•	Logic Controller	
- 🔄 Graph Re		Ctrl-X	Config Element	Counter
- 🎤 HTTP Req	Сору	Ctrl-C	Timer I	CSV Data Set Config
- A? Response		Ctrl-V	Pre Processors	FTP Request Defaults
- 🛜 Assertion	Duplicate	Ctrl+Shift-C	Sampler 0	HTTP Authorization Manager
- O Constant	Reset Gui		Post Processors	HTTP Cache Manager
- 💽 View Res	Remove	Delete	Assertions 0	HTTP Cookie Manager
View Res. Providence Result State WorkBench			Listener	HTTP Header Manager
				HTTP Request Defaults
			Forever 100	Java Request Defaults
	Save Node As Image	Ctrl-G	ad creation until n	JDBC Connection Configuration     Keystore Configuration

#### **HTTP Request Defaults**

Name: HTTP Request Defaults	
Comments:	
Web Server	
Server Name or IP: www.google.com	Port Number: 80

#### • HTTP Request

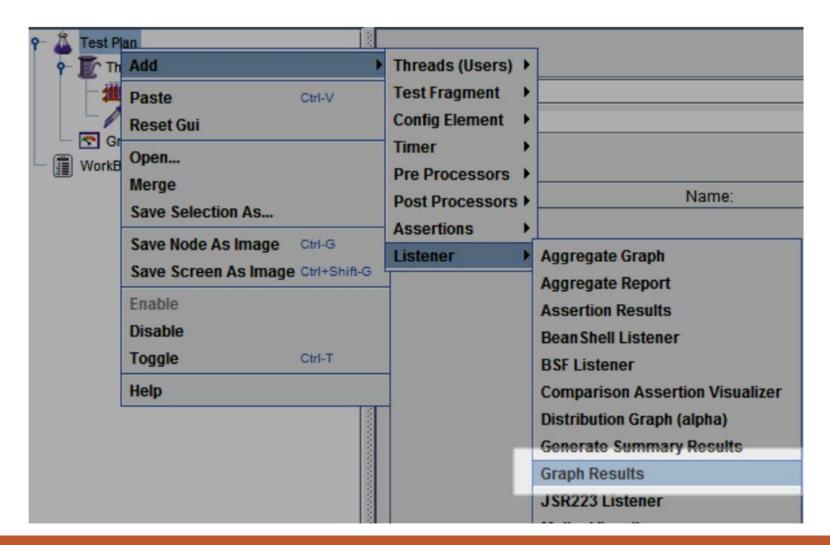
Right-click on Thread Group and select: **Add** -> **Sampler** -> **HTTP Request**.

♀- Â Test Plan ♀- Ĩ hread Group		Thread Gr	oup		
HTTP Rec		•	Logic Controller	Þ	
- 🔄 Graph Re		Ctrl-X	Config Element	۲	
- 🎤 HTTP Rec	Сору	Ctrl-C	Timer	۲	r error
- ARESPONS		Ctrl-V	Pre Processors	۲	
- 💽 Assertion	Dublicate	Ctrl+Shift-C	Sampler	Þ	Access Log Sampler
- 🕚 Constant	Reset Gui		Post Processors	۲	AJP/1.3 Sampler
- 💽 View Res	Remove	Delete	Assertions	۲	BeanShell Sampler
─ S View Res			Listener	۲	BSF Sampler
WorkBench	Merge			=	Debug Sampler
SE WORDERCH	Save Selection As		Forever 100	_	FTP Request
			ead creation until I	n	HTTP Request
	Save Node As Image	Ctrl-G	t	1	Java Request

#### Step 3) Adding Graph result

JMeter can show the test result in Graph format.

Right click Test Plan, Add -> Listener -> Graph Results



#### Step 4) Run Test and get the test result

Press **the Run** button (Ctrl + R) on the Toolbar to start the software testing process. You will see the test result display on Graph in the real time.

0 +		2 🔍 🔍 🐍				0 🔔	0/100
Graph Resu							
Name: Graph F Comments:	Results						
	to file / Read from file						
Filename			Browse Log/D	splay Only: 🔲 Erro	ors 🔲 Successes	Conf	ligure
	Graphs to Di	splay 🖌 Data 🖌 Ave			_		
		spiny 🕑 Data 🕐 Are	ruge 👔 meuran	P Denation P	moughput		
m	5						
				N .			
				3			
ms	4	1					_
		Latest Sample					

### Jmeter Timers: Constant, Gaussian Random, Uniform [Example]

#### What are Timers?

By default, JMeter sends the request **without pausing** between each request. In that case, JMeter could **overwhelm** your test server by making too many requests in a short amount of times.

#### **Constant Timer:**

Constant timer delays each user request for the **same** amount of time.

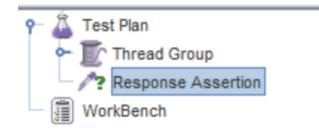
Constant Timer				
Name:	Constant Timer			
Comm	ents:			
Thread	Delay (in milliseconds): 300			

## How to use Assertions in JMeter (Response Example)

## What is an Assertion?

Assertion help verifies that your server under test returns the **expected** results.

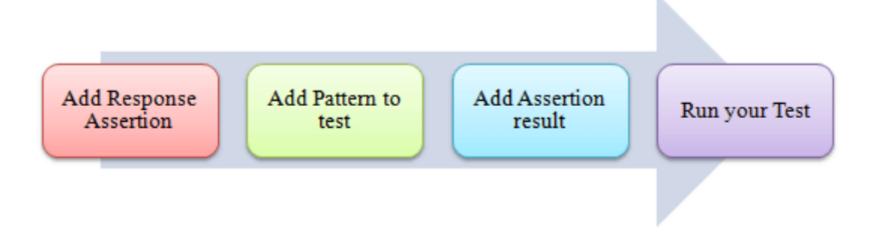
### **Response Assertion**



The response assertion lets you add pattern strings to be compared against various fields of the server response.

For example, you send a user request to the website <a href="http://www.google.com">http://www.google.com</a> and get the server response. You can use Response Assertion to verify if the server response **contains** expected pattern string (e.g. "OK").

#### **Steps to use Response Assertion**



#### Step 1) Add Response Assertion

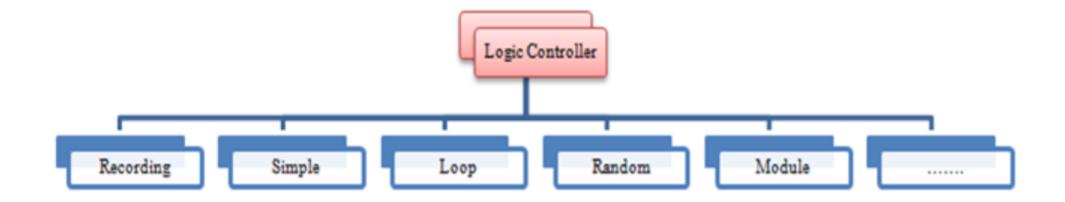
#### Right-Click Thread Group -> Add -> Assertions -> Response Assertion

Test Plan Thread Group		Thre	ad Group	
HTTP Req	Add	•	Logic Controller >	
— 🥕 HTTP Req	Cut	Ctrl-X	Config Element	
- A? Response	Сору	Ctrl-C	Timer 🕨	Sampler error
Assertion	Paste	Ctrl-V	Pre Processors >	Start Next Thread Loop
WorkBench	Duplicate	Ctrl+Shift-C	Sampler •	
	Reset Gui		Post Processors >	
	Remove	Delete	Assertions	Bean Shell Assertion
	Open		Listener	BSF Assertion
	Merge		Count: 🔲 Forever	Compare Assertion
	Save Selection As			Duration Assertion
	Save Node As Image	Ctrl-G	elay Thread creation	HTML Assertion
	Save Screen As Image		cheduler	JSR223 Assertion
		our onneo	ſ	MD5Hex Assertion
	Enable			Response Assertion
	Disable			Size Assertion
L	Toggle	Ctrl-T		SMIMEAssertion

# Controllers in JMeter: Loop, Simple, Transaction, Module, Random

## What is the Logic Controller?

Logic Controllers let you define the order of processing request in a Thread. It lets you control "when" to send a user request to a web server. For example, you can use Random Controllers to send HTTP requests to the server randomly



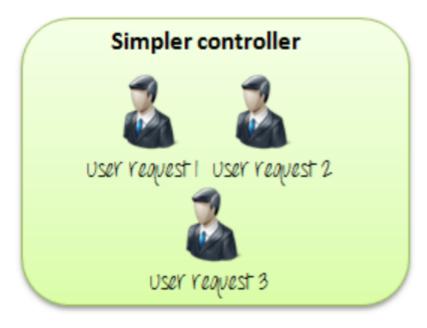
## **Recording Controller:**

JMeter can **record** your **Testing** steps; a recording controller is a **placeholder** to store these recording steps.



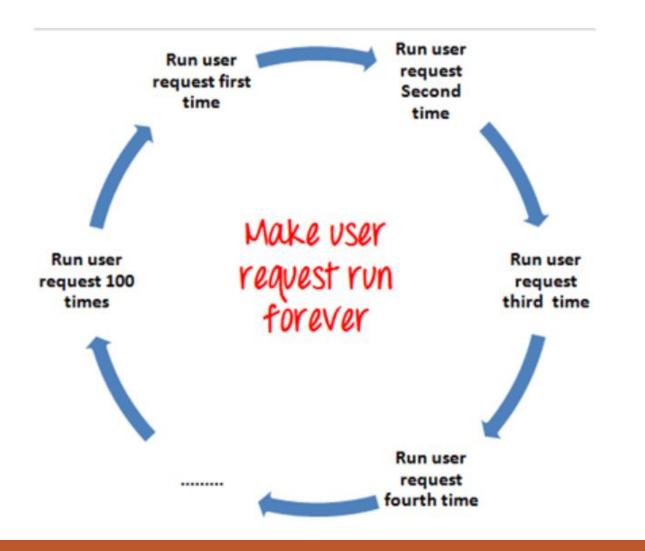
### Simple Controller:

Simple Controller is just a container for user request.



### Loop Controller:

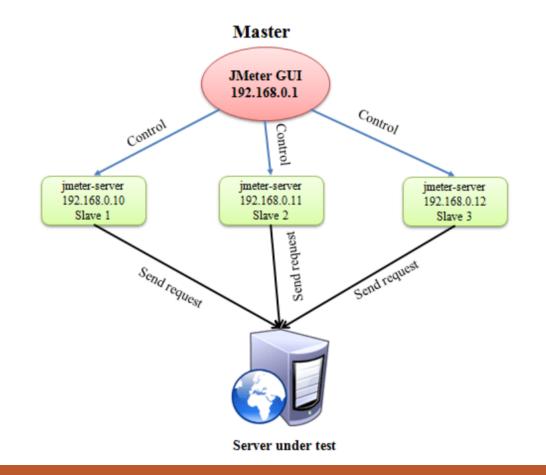
Loop Controller makes the user request run **a specified number of times** or run **forever** as shown in figure:



## **Advanced in JMeter**

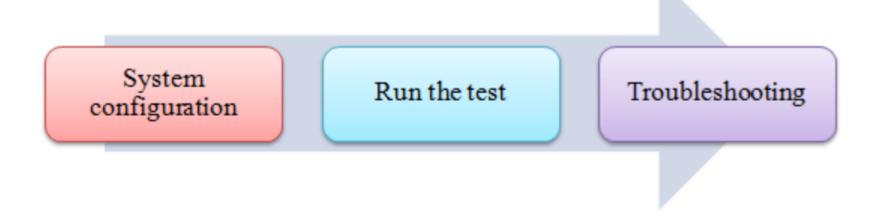
## **Jmeter Distributed (Remote) Testing: Master Slave** Configuration

Distributes testing uses client-server model as the figure below:



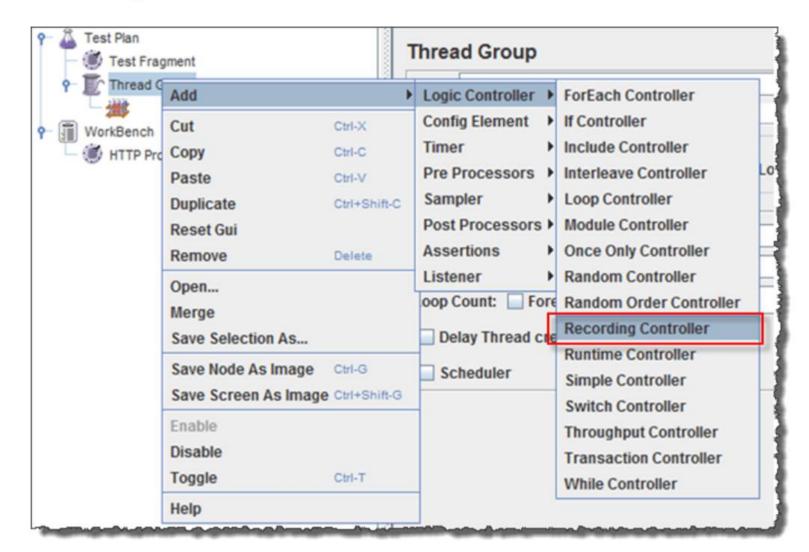
## Jmeter Distributed (Remote) Testing: Master Slave Configuration

- Master: the system running JMeter GUI, control each slave.
- Slave: the system running JMeter-server, receive a command from the master and send a request to a server under test.
- **Target**: the web server under test, get a request from slaves.



## HTTP Proxy Server in JMeter: Record Example Script

#### **Recording Controller**



## HTTP Proxy Server in JMeter: Record Example Script

#### Set Target Controller where your recorded scripts will be added

HTTP Proxy Server	
Name: HTTP Proxy Server	
Comments:	
Port: 8080 Attempt HTTPS Spoofing Optional URL match string:	
Test plan content Target Controller: Test Plan > Thread Group Grouping: Do not group samplers Capture HTTP Headers Add Assertions Regex matching	
HTTP Sampler settings Type: HTTP Request 🔹 Redirect Automatically 🖉 Follow Redirects 🖉 Use KeepAlive 📄 Retrieve All Embedd	led Res
Content-type filter	
Include: Exclude:	

. Start Proxy Server

Return to HTTP Proxy Server, and click the **Start** button at the bottom. Now your JMeter

proxy server start

HTTP Proxy Server	
Name: HTTP Proxy Server	
Comments:	
Global Settings       Port:     8080	
Test plan content	
Target Controller:       Test Plan > Thread Group <ul> <li>Grouping:</li> <li>Do not g</li> </ul>	
Capture HTTP Headers Add Assertions Regex matching	
HTTP Sampler settings         Type: <ul> <li>Redirect Automatically</li> <li>Follow Redirects</li> <li>Use KeepAlive</li> <li>Image: Setting Seting Setting Setting Setting Setting Setting Setting S</li></ul>	
Content-type filter	
Include: Exclude:	
URL Patterns to Include	
URL Patterns to Include	
Add Delete Add from Clipboard	

Start your Browser (I used Firefox), choose **Tool** => **Option** => **Advanced** => **Network** =>

**Setting =>** Enter HTTP proxy as figure below

Options							23
		页		90		$\bigcirc$	iç.
General	Tabs	Content	Applications	Privacy	Security	Sync	Advanced
General Da	ata Choices	Network	Update Encry	ption			
Connec	tion						
Configu	ure how Fire	efox connec	ts to the Interne	et		S <u>e</u> ttir	ngs
-	1000 C 1000					1	x
Connec	tion Setting	S					
Con	figure Proxi	es to Acces	s the Internet				
	No prox <u>y</u>						
0	Auto-detec	t proxy sett	ings for this net	<u>w</u> ork			
0	<u>U</u> se system	proxy settin	ngs				
•	<u>M</u> anual pro	xy configui	ration:	4			
	HTTP Pro	xy: localh	ost		<u>P</u> ort:	8080 🌲	
		🔲 U <u>s</u>	e this proxy serv	er for all pr	otocols		
	SS <u>L</u> Pro	xy:			P <u>o</u> rt:	0	
	ETP Pro	xy:			Po <u>r</u> t:	0	
	SO <u>C</u> KS Ho	st:			Por <u>t</u> :	0	
		SO	C <u>K</u> S v4 ⊚ SC	OCKS <u>v</u> 5			<u> </u>
	No Proxy fo	or:					

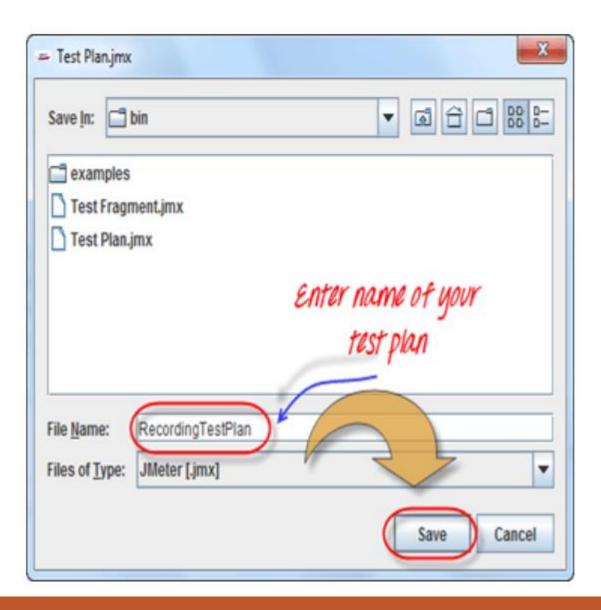
After finishing recording, you will see JMeter automatically created a new HTTP request as the figure below

Test Plan	HTTP Request
HTTP Request Defaults Recording Controller //nrpc/p //nrpc/n //js_f.php Summary Report WorkBench	Name: / Comments: -Web Server Server Name or IP: -HTTP Request Implementation:  Protocol [http]: http Method Path: / Redirect Automatically  Follow Redirects  Use KeepAlive Parameters Post Body
	Send Parameters
	Name:

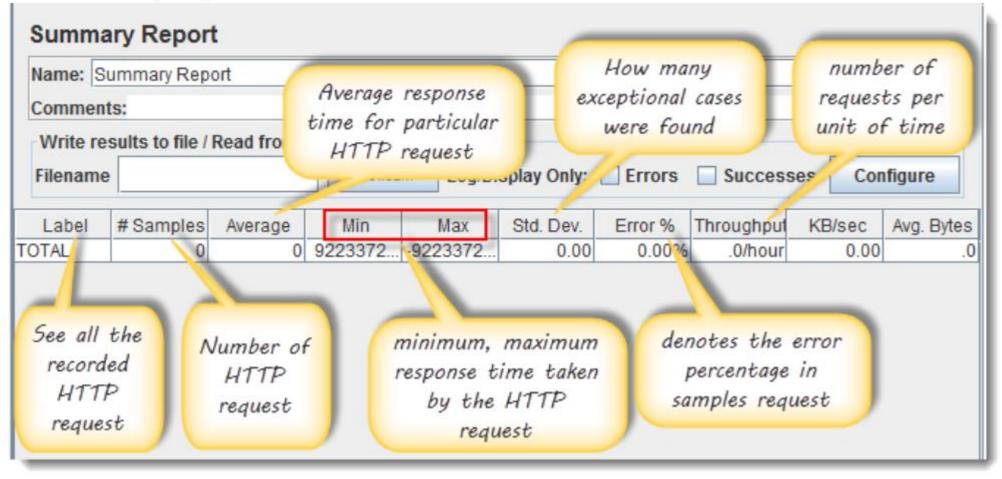
#### Click File => Save your Test Plan as

- Apache JMeter (2.9 r1437	7961)		
<u>File Edit Search Run</u>	Options	Help	
Close	Ctrl-L	1	1 + - 4
Open	Ctrl-O		
Merge		1	est Plan
Save	Ctrl-S		
Save Test Plan as	Ctrl+Shi	ft-S	lame: Test Plan
Save Selection As			comments:
<u>R</u> evert			
1 TestPlan.jmx			Nam
2 TestPlan_v1.jmx			
3 AggregateGraph.jmx			
<u>4</u> RecordingTestPlan.jmx			
5 Test Fragment.jmx			
6 PerfomanaceTestPlan.jr	nx		
7 Test Plan.jmx			
8 Test Fragment.jmx			
E <u>x</u> it	Ctrl-Q		
isel bim		_hh	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

A Dialog box display => enter a name of your test plan at File Name field => Click Save Now your Test Plan is saved under name RecordingTestPlan.jmx



#### The Summary Report will show some basic statics



### Save your test result

#### 1. Click Save Table Data to save test result to file

Name: Su	mmary Repor	t							
Comments	s:								
Write res	Write results to file / Read from file								
Filename			B	rowse	og/Display On	ly: 🗌 Error	s Succes	sses Co	nfigure
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
	100	535	356	782	97.34	100.00%	1.9/sec	6.37	3506.9
OTAL	100	535	356	782	97.34	100.00%	1.9/sec	6.37	3506.9

 Enter the name of the test result and click Save. Test Result in JMeter is saved in \*.csv format as default

summary.csv	x
Save In: 🗂 bin 🔻 🖬 🔂 🔂 🔀	0-
i examples	
summary.csv	
File Name: RecordingTestResult.csv	
Files of Type: JMeter [.csv]	•
Save Cance	

## **Best Practice for JMeter Tests**

- 1) Limit the Number of Threads
- 2) Using a proxy server
- 3) Using variables
- 4) Reduce resource requirement
- 5) Check the JMeter logs
- 6) Erase the local path from CSV Data Set Config
- 7) Follow file naming convention



JNeter™

# Best Practice for JMeter Tests **/ Meter**™



#### **1)** Limit the Number of Threads

The maximum number of threads you can effectively run with JMeter is 300. This limit is because of hardware's capabilities. If JMeter is made to run with more number of threads, the accuracy of timing information will decrease.

#### 2) Using a proxy server

The Proxy server helps you to abstract out certain common elements from the recorded samples. Moreover, it is useful features to record your testing.

#### **3) Using variables**

Some test plans need to use different values for different users/threads. For example, you might want to test a sequence that requires a unique login for each user. This is easy to achieve using variables.

#### 4) **Reduce resource requirement**

> The GUI mode consumes a lot of computer memory under heavy load. It causes performance issues

# **Best Practice for JMeter Tests / Meter**<sup>™</sup>



#### **Reduce resource requirement**

The GUI mode consumes a lot of computer memory under heavy load. It causes performance issues.

There're some tips to reduce resource requirement:

- Use non-GUI mode
- Disable the "View Result Tree" listener during the Load test. Because it consumes more memory and causes JMeter running to run out of memory.
- Disable all JMeter graphs results
- Use the CSV test result format.
- Only save the needed test result. JMeter could take a long time to save very detailed test results.

# Best Practice for JMeter Tests



Any errors in the <u>Test Plan</u> or test execution will be recorded in the log files. Monitoring the log file help you to find the error early

#### 6) Erase the local path from CSV Data Set Config

If you are using an existing CSV data file that you created on your local computer, you should delete the existing local path (Current path of CSV file). If you don't delete the local path, JMeter cannot find the CSV data file on your local PC.

#### 7) Follow file naming convention

Don't save test plan under complex file name, use only alphanumeric characters

## **Jmeter Alternatives 2020**













Locust

## References

- **1. JMeter Website** 
  - <u>https://jmeter.apache.org/</u>
- 2. JMeter Tutorial for Beginners: Learn in 7 Days
  - <u>https://www.guru99.com/jmeter-tutorials.html</u>

## **THANK YOU**

