

- Develop a foundational understanding of JMeter
- Determine strategies for recording a test
- Discover how to monitor bottlenecks
- Understand meaningful reporting strategies
- Learn how to Scale tests in the cloud cost effectively

Load and performance testing is complicated due to the holistic nature of finding performance bottlenecks. Rather than complicate matters by introducing the complexities of a proprietary load testing software licensing scheme, with some added engineering effort, you can use an open source framework to manage, record, and execute your tests.

This hands-on class will engage students in running tests against a sample application to help solve issues that include basic test recording, dealing with SSL, and parameterizing tests. It includes theoretical components that discuss how to find bottlenecks, what and how to monitor tests, the importance of results aggregation and reporting, and how to implement it. Lastly, we will discuss the creation of a cost-effective load and performance infrastructure in the cloud so you can scale tests and not break the bank.

## Who Should Attend

This intermediate class contains technical content that teaches JMeter load testing techniques and is appropriate for Software Developer, Software Tester, Test Engineer, Performance Tester, Performance Engineer, Test Lead, DevOps Engineer, and Operational Support roles. Anyone interested in making an elastic load testing infrastructure that can fit into their CI/CD pipeline should also attend.

## Prerequisites

Participants should have some programming knowledge, some knowledge of the http protocol, and a basic understanding of how web servers work.

## Laptop Required

Attendees are required to bring their own laptop with permissions to install the following:

- Java 8 or 9 installed.
- [JMeter software](#) [1]
- A text editor. The text editor will need to be more advanced than Notepad. If you are a developer, your existing IDE will do; if not we recommend [Notepad++](#) [2].
- The ability to ssh; we recommend [putty](#) [3].
- A git client. You can choose any client but [we recommend this one](#) [4].

At a minimum you must be able to install [putty](#) [3] and have an [html5 capable browser](#) [5]. If you can only meet the minimum requirements, please contact Client Support prior to class as this will require additional preparation by the course instructor.

## Course Outline

---

### Intro to JMeter

Discussion of the tool

### Picking Test Scenarios

Theory section that identifies some rules of thumb for selecting tests

Overview of the UI  
Discussion of Basic Performance testing concepts

### **Recording a Test**

Step by step instruction for recording a simple test against a sample application  
Examples of how to locally execute that test  
Reading the results to a simple test

### **Dealing with SSL**

High level discussion of SSL concepts  
How JMeter handles SSL  
Hands on run through of recording a test against an SSL sample site

### **Building Tests from a Recorded Script**

Discussion on generalizing tests  
Walk through of Jmeter UI tools that help parameterize tests

### **Running Tests**

Why you should use Jenkins to control test execution  
SCM and managing your tests  
Exercises with Jenkins

What to do when you have to create test scenarios from scratch

How to decide what to test on a brand new system

### **Running Tests in Parallel**

Introduction of thread groups  
Overview of data dependancies  
Discussion of when and how to seed data

### **What to Monitor**

Where to look for bottlenecks  
Common free tools to use  
Discussion of the usual suspects

### **Results Aggregation and Reporting**

How to collect data  
How to normalize the data for a coherent report  
The importance of time

### **Scaling in the Cloud**

Demo of a simple, scaleable, cost effective performance testing architecture  
Discussion of design decisions and limitations

**Price:** \$1545