Optimize 25 Testing at Scale: Setup Guide



Prepared by

**Zachery Olson**

Solution Architect

Contributors

**List of Contributors**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  | zacolson | 1.0 | Initial draft for review/discussion |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Table of Contents

[1 Introduction 5](#_Toc130541336)

[1.1 Pre-requisites 5](#_Toc130541337)

[1.2 JMeter 6](#_Toc130541338)

[1.3 Azure Load Testing 7](#_Toc130541339)

[1.4 JMeter Distributed Testing 8](#_Toc130541340)

[1.5 Resources 9](#_Toc130541341)

1. Introduction

This document will provide the pre-requisites and setup required to setup and execute the load test samples provided through this guidance. Our guidance, and the samples provided through this guidance will be based on Apache JMeter. Apache JMeter is an open-source Java based application designed to test functional behavior and measure performance.

Apache JMeter is not a requirement for load testing your application. However, Azure Load Testing uses JMeter scripts to execute tests.

* 1. Pre-requisites

**Java:** Apache JMeter is a Java based desktop application. Download and install the latest Java SE Development Kit (JDK) for your operating system.

* <https://www.oracle.com/java/technologies/downloads/>

You can verify if Java is already installed by running the java -version command in a Command Prompt:

**Text

Description automatically generated**

**Operating System:** Apache JMeter is supported to run on the following Operating systems:

* Windows
* Linux
* Mac OS
* Ubuntu

**Azure Subscription (optional):** An active Microsoft Azure subscription is recommended. This is a requirement if you plan to execute load tests in Microsoft Azure or integrate load testing into your CI/CD pipelines.

* 1. JMeter

A local installation of Apache JMeter is useful for creating new load tests, editing and debugging existing load test scripts.

Follow the steps below to download and extract the Apache JMeter binaries for Windows:

1. Open the Download Apache JMeter page:
   1. <https://jmeter.apache.org/download_jmeter.cgi>
2. Download the appropriate binary package for your operating system:
   1. apache-jmeter-5.x.zip for Windows operating systems
3. Extract the apache-jmeter-5.x.zip folder to a directory on your system:
   1. Example: c:\Tools
4. Navigate the extracted package to the bin folder:
   1. C:\Tools\apache-jmeter-5.5\apache-jmeter-5.5\bin
5. The bin folder contains the script file (jmeter.bat) for launching Apache JMeter
   1. Execute jmeter.bat to launch Apache JMeter in GUI mode.

Apache JMeter has two modes, GUI mode and CLI mode:

* GUI: mode used for test creation, editing, and debugging.
* CLI: mode used to run the load test.

Follow the steps below to load an existing Apache JMeter (.jmx file) script file:

1. In Apache JMeter, click File – Open
2. Use the explorer window to navigate to file or repository containing the .jmx file.
3. Select the .jmx you want to load, and click Open.

IMPORTANT: While the test script is loaded into Apache JMeter using the GUI mode, the test might not be ready to execute against your application. When using the samples provided through this guidance, follow the sample instructions provided for each sample. The instructions for each sample include the steps necessary to prepare the sample for execution against your application.

* 1. Azure Load Testing

An Azure Load Testing resource makes it easy to generate high-scale load with existing an JMeter script. Implement load testing without the need for complex infrastructure and integrate load testing into automated CI/CD pipelines to identify and address performance bottlenecks early in the development lifecycle.

Follow the steps below to create a load testing resource:

1. In Microsoft Azure, click Create a resource.
2. Use the phrase “load test” in the Search services and marketplace box.
3. Click Create on Azure Load Testing.
4. Populate the Basics tab with the:
   1. Appropriate Azure subscription information
   2. Resource group
   3. Instance name
   4. Appropriate Region for the resource
5. On the Encryption tab, select whether to use Microsoft-managed or Customer-managed encryption keys.
6. On the Tag tab, create a tag if required.
7. Once the validation has passed on the Review + create tab, click Create.

Follow the steps below to use an existing JMeter script (.jmx) file to create a load test in Azure Load Testing:

1. In Microsoft Azure, navigate to and open your Azure Load Testing resource.
2. Under Tests, click Tests.
3. Click + Create and choose the option “Upload a JMeter script”
4. On the Basics tab:
   1. Enter a Test name and Test description.
5. On the Test plan tab:
   1. Upload the .jmx file.
   2. Upload any dependency files (i.e., csv configurations)
   3. Uncheck the “Run test after create” box if you do not wish to run the test automatically after the test is created.
6. On the Parameters tab:
   1. Enter any Environment variables to be passed into the test from Azure Load Testing to JMeter user defined variables (System.getenv). This might be a host name, URL, or parameters passed in through CI/CD automation.
   2. Enter any Secrets to be passed into the test from Azure Load Testing. Secrets might be used in a test that executes against an API or function that requires authentication. Azure Load Testing integrates with Azure Key Vault and recommends that Secrets are stored in Azure Key Vault.
   3. Enter any Certificates to be passed into the test from Azure Load Testing. Certificates might be used in a test that executes against a secure endpoint. Azure Load Testing integrates with Azure Key Vault and recommends that your Certificates are stored in Azure Key Vault.
7. On the Load tab:
   1. Configure the test engine instances to a meet the target load for the test. The number of threads specified in the jmx file represents the number of thread (virtual users) execute by one test engine instance.
   2. Network traffic to Power Platform applications will be routed through Public networks.
8. On the Test criteria tab:
   1. Define the criteria to specify the performance expectations of the system under test. Use the defined criteria to determine the failure conditions for the test when the criteria evaluate to true.
9. On the Monitoring tab:
   1. Add any Azure-hosted applications to monitor server-side performance when running a load test in Azure Load Testing.
10. Once the validation has passed on the Review + create tab, click Create. If the “Run test after create” is checked, the test will execute with no further action required.
    1. JMeter Distributed Testing

JMeter distributed testing is a method of executing load test volume across multiple machines simultaneously. Distributed testing introduces the concept of a controller node and a worker node.

* **Controller Node** –the primary system running JMeter (jmeter.bat) and controls the test.
* **Worker Node** - the systems running jmeter-server (jmeter-server.bat) and takes commands from the controller node to send requests to the target application.

Distributed testing should be used in situations where:

* You need to simulate users accessing your application from different locations.
* Local CPU and Memory resources are limited.
* You need to simulate a higher volume of users.

Follow this [step-by-step](https://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.html) to set up and perform distributed testing.

* 1. Resources

Getting Started - <https://jmeter.apache.org/usermanual/get-started.html>

JMeter Best Practices - <https://jmeter.apache.org/usermanual/best-practices.html>

Configure high scale load tests - <https://learn.microsoft.com/en-us/azure/load-testing/how-to-high-scale-load>

Parameterize load tests - <https://learn.microsoft.com/en-us/azure/load-testing/how-to-parameterize-load-tests>

Distributed Testing - <https://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.html>

Java Download - <https://www.oracle.com/java/technologies/downloads/>