Load Testing

Introduction to Locust & Jmeter

Presented by
Md. Khaled Saifullah Sadi
Junior Programmer
WebCrafter Team

Agenda

What Is Load Testing?

Different Tools for Load Testing

Introduction to Locust

Introduction to Apache JMeter

What Is Load Testing?

Definition

Load testing is a type of performance testing that assesses a system's performance under real-life conditions.

Purpose

- ➔ Determine system's load capacity.
- ➔ Confirm system's ability to handle user demand.

Identifying Maximum Capacity

- ➔ Determine maximum operating capacity of an application.
- ➔ Assess if existing infrastructure can support the application.

Concurrency Testing

Determine number of users that can work on an application simultaneously.

Different Tools for Load Testing

Locust

An open source load testing tool. Define user behaviour with Python code, and swarm system with millions of simultaneous users

Apache JMeter

An open-source tool for load testing, performance testing, and functional testing. It can also support database testing.

WebLOAD

An enterprise-grade tool for web applications that supports over 100 technologies.

NeoLoad

A tool that simulates user activity and observes infrastructure performance. This can help identify bottlenecks in web and mobile applications.

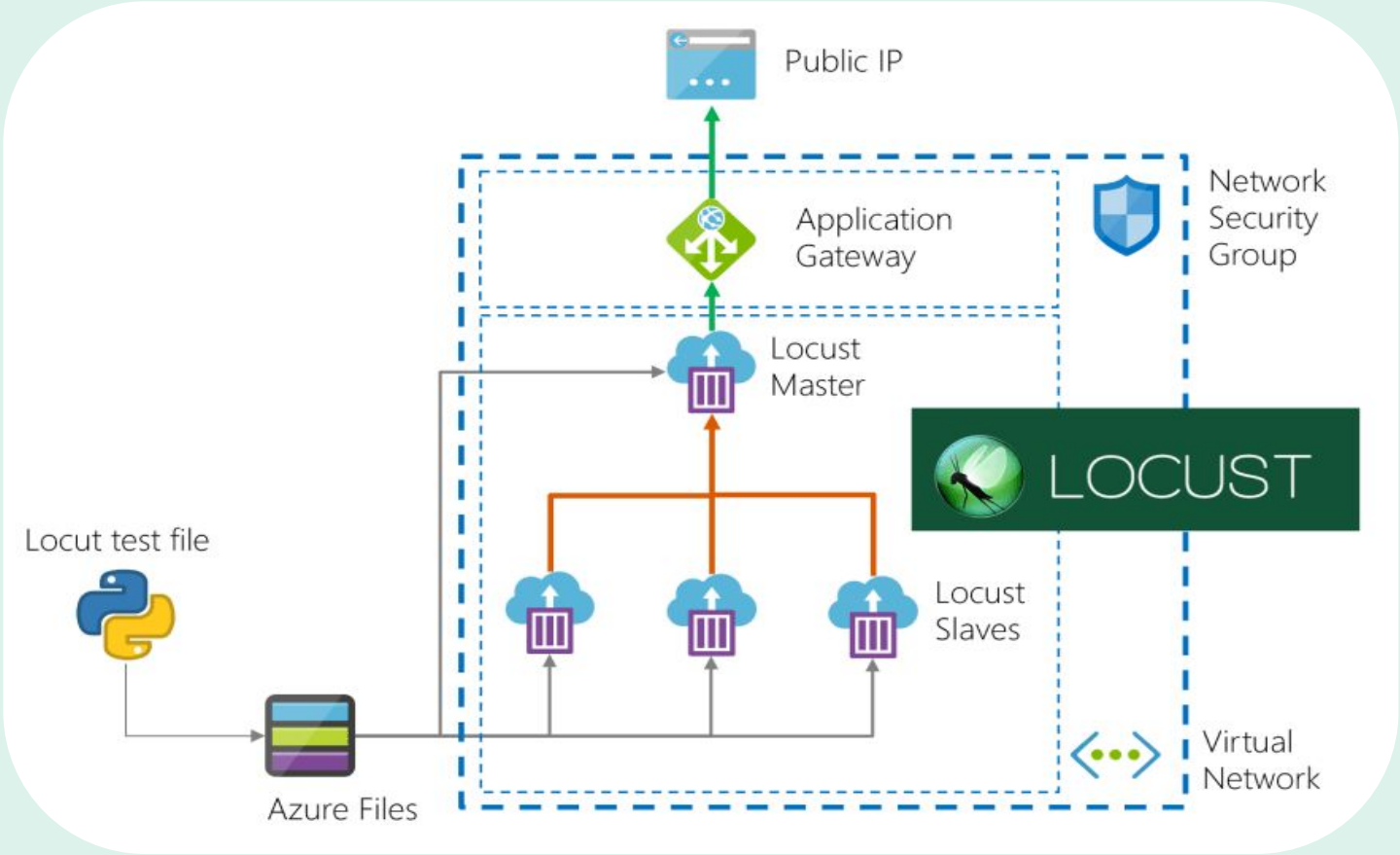
Introduction

Initiating Python Load Testing with Locust

- Python-based open-source load testing tool.
- Allows writing test scenarios in Python code.
- Highly scalable, easy-to-use, and customizable.
- Provides real-time monitoring and reporting.

python™

How Locust works?



How Can Use Locust?

1

Install python in our system

Install Python (3.8 or later).Download the latest version Python from python.org

2

Install Locust

To install Locust, Open a terminal or command prompt and execute the following command:

```
$ pip3 install locust
```

3

Create a “locustfile.py” file

To create a locustfile.py file, open a text editor or IDE and save the file with the name "locustfile.py" to start writing Locust test scenarios using Python code.

4

Run “locust”

```
$ locust -f locustfile.py
```

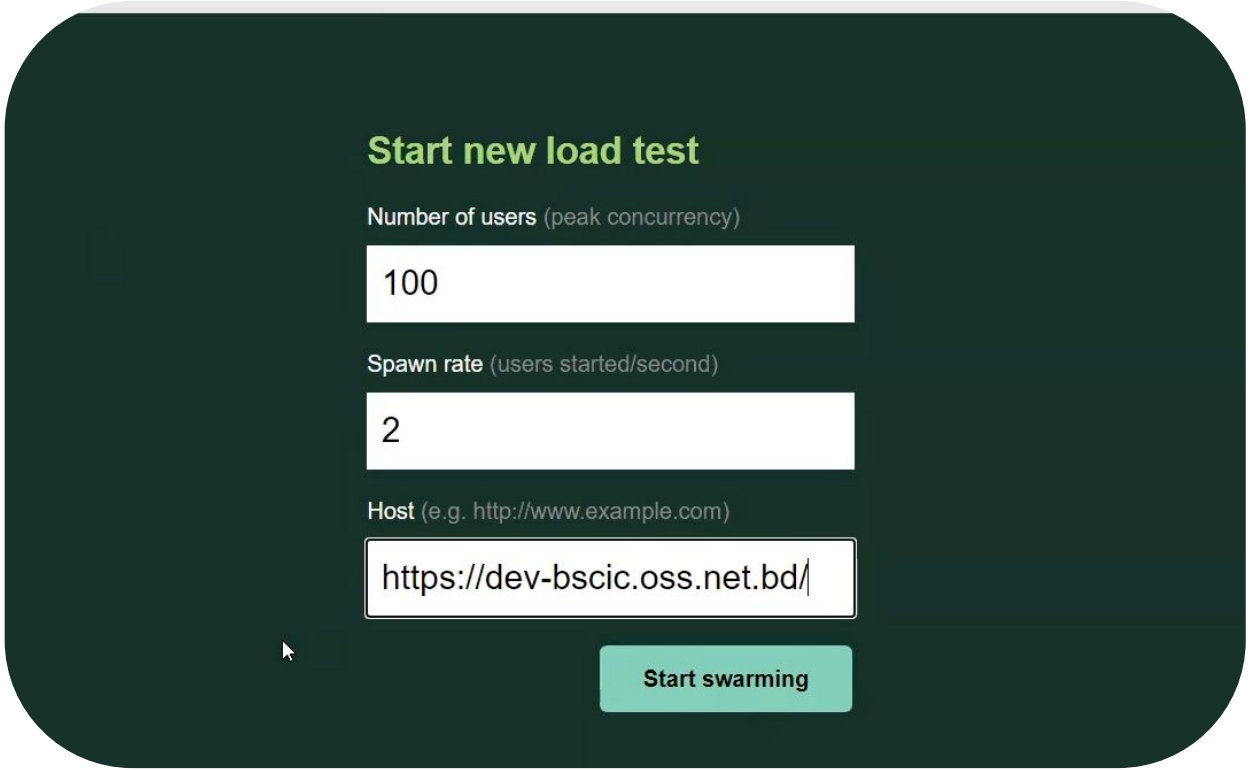
This command runs Locust with the test scenarios defined in the locustfile.py file.

Writing a locustfile test scripts

locustfile.py > ...

```python
from locust import HttpUser, task, between

class WebsiteTestUser(HttpUser):
    # wait_time = between(0.5, 3.0)

    def on_start(self):
        """ on_start is called when a Locust start before any task is scheduled """
        pass

    def on_stop(self):
        """ on_stop is called when the TaskSet is stopping """
        pass

    @task(1)
    def hello_world(self):
        self.client.get("/industrial-advisor")

```

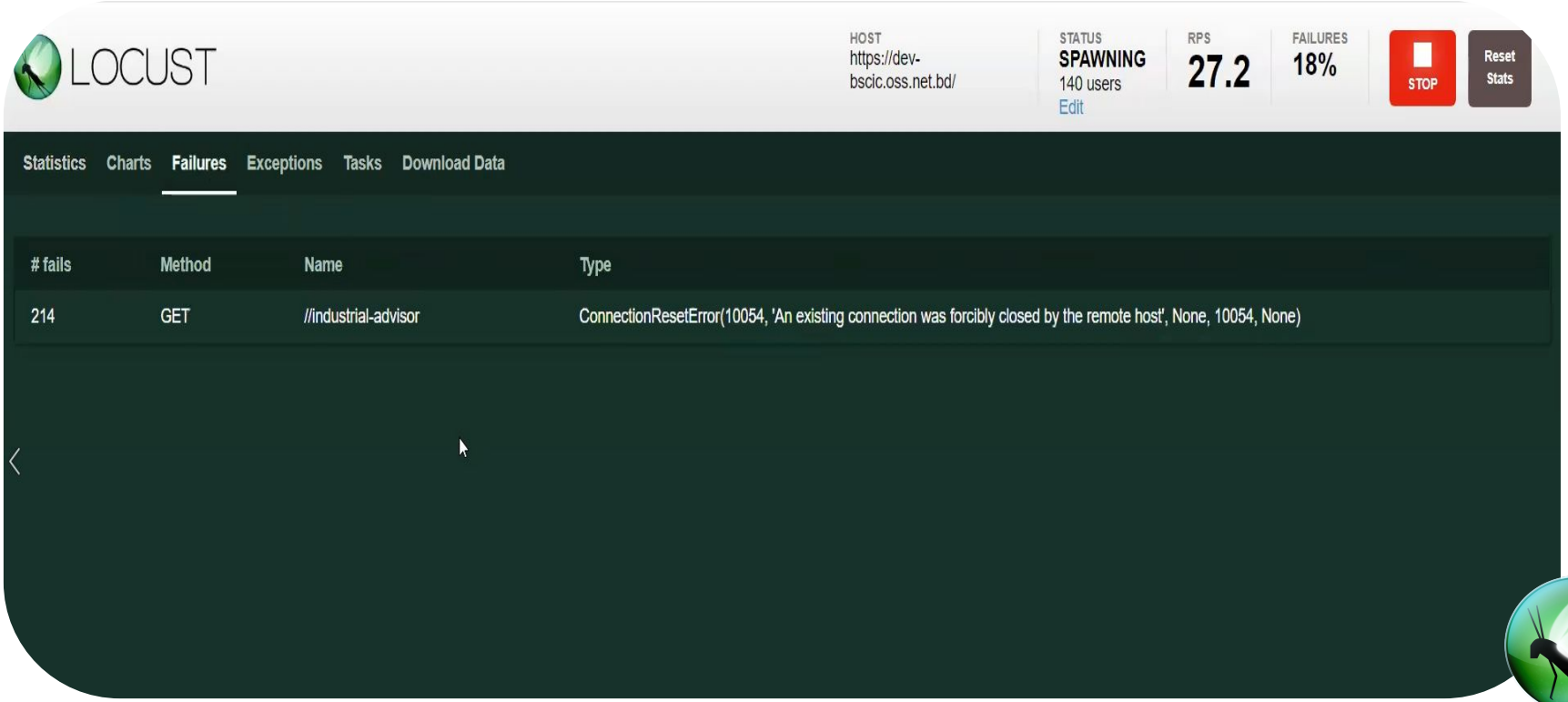
Host and number of users set UI



View result statistics panel

View result chart panel

View result failures panel



Report Download Option

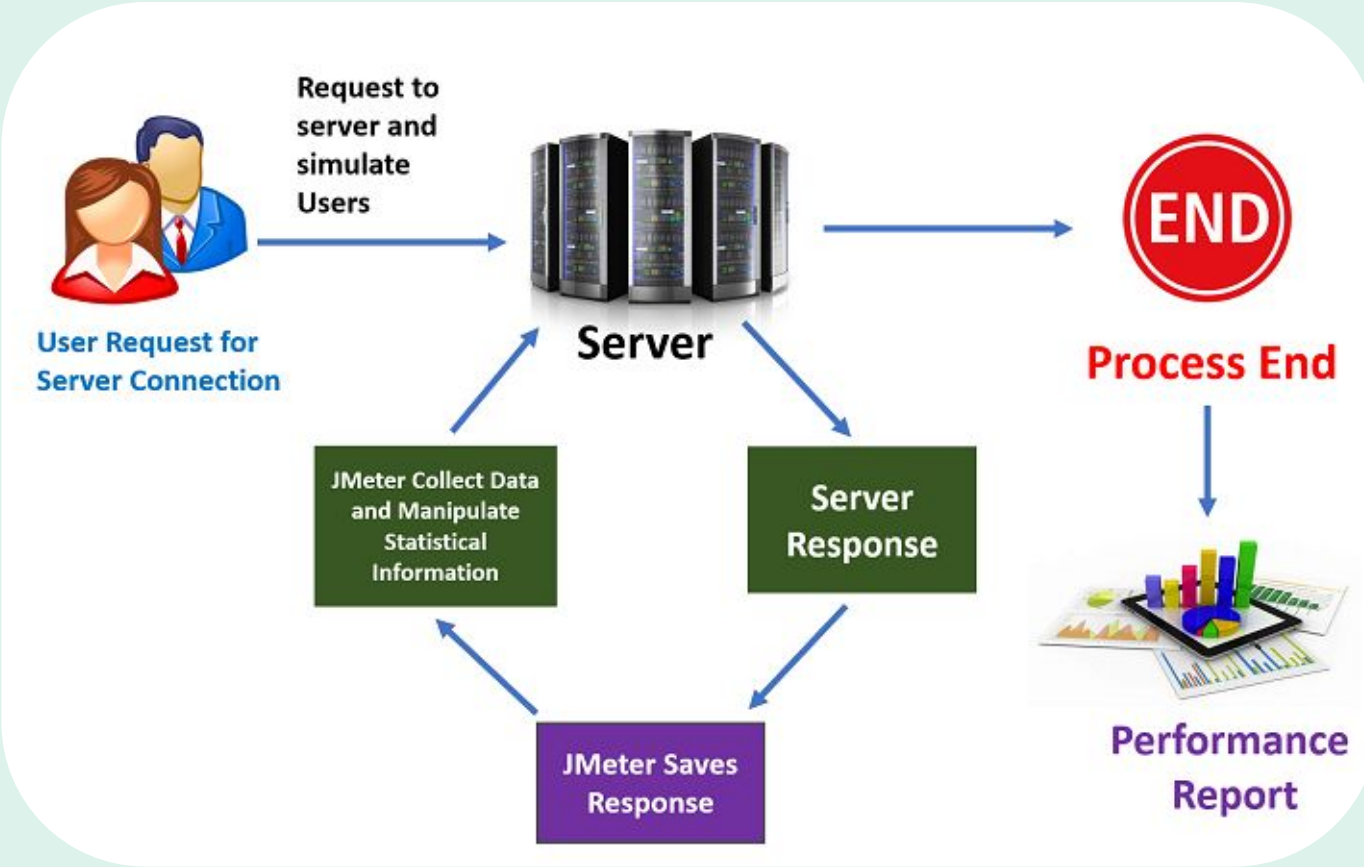
Introduction

Navigating Load Testing with Apache JMeter

- Java-based open-source load testing tool.
- Offers a graphical interface for test scenario creation.
- Supports various protocols including HTTP, HTTPS, FTP, JDBC, etc.
- Extensive plugins for additional functionalities.

Java

How Jmeter works?



How To Create Jmeter Test?

1

Run Jmeter

To run JMeter, navigate to the JMeter bin directory and execute the startup script.

2

Create Test plan and Thread Group(Users)

Create a Thread Group in the Test Plan and configure the number of users, ramp-up period, and loop count.

3

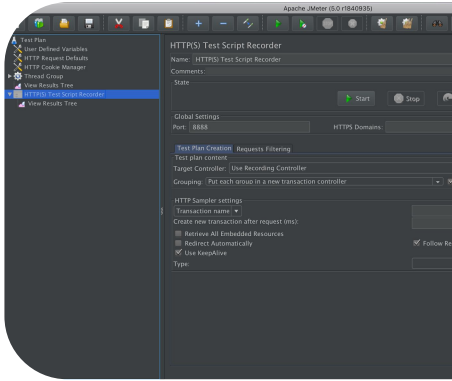
Add Controllers and Listeners

Include Controllers and Listeners in the Test Plan for organizing and analyzing test scenarios and results, respectively.

4

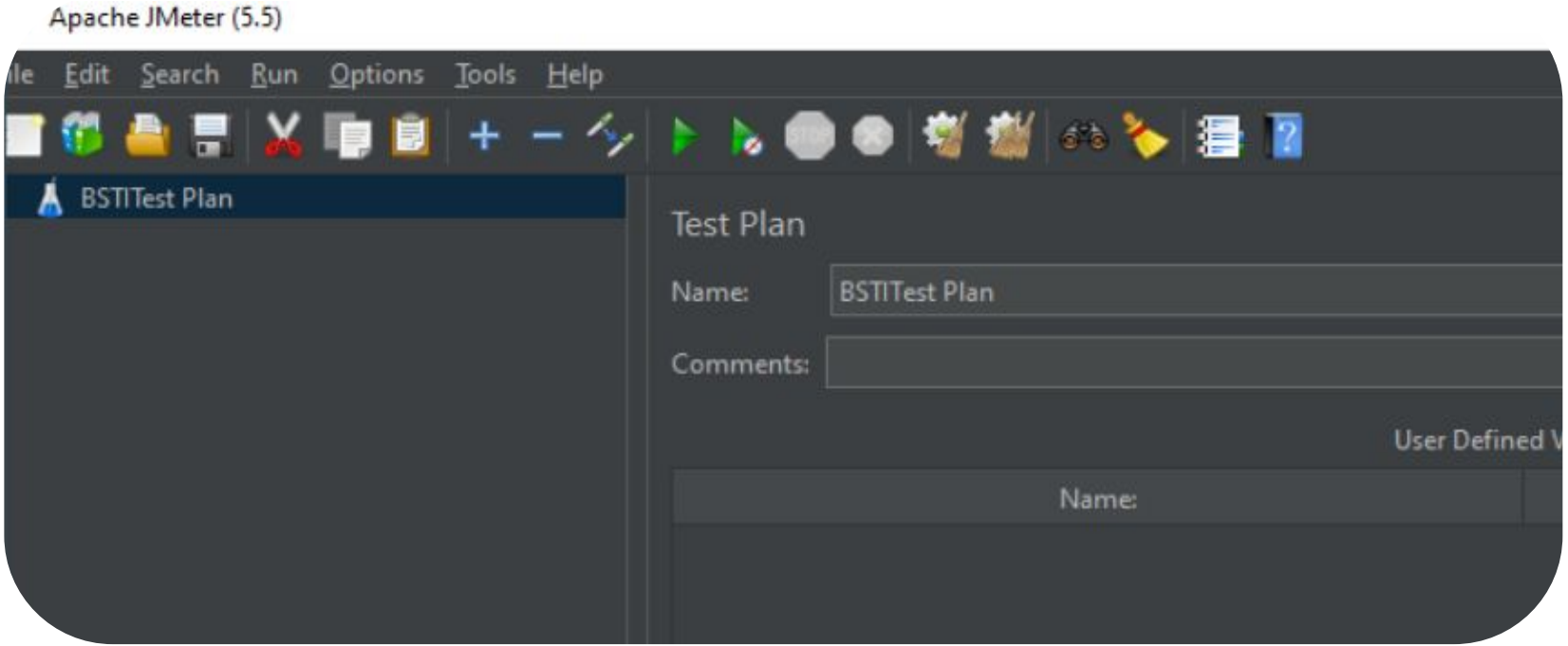
Timers,Assertions,Outcomes

Combine Timers and Assertions to regulate timing and validate outcomes, ensuring effective test execution within the test plan.



JMeter Test Plan

A test plan describes a series of steps JMeter will execute when run. A complete test plan will consist of one or more Thread Groups, logic controllers, sample generating controllers, listeners, timers, assertions, and configuration elements.

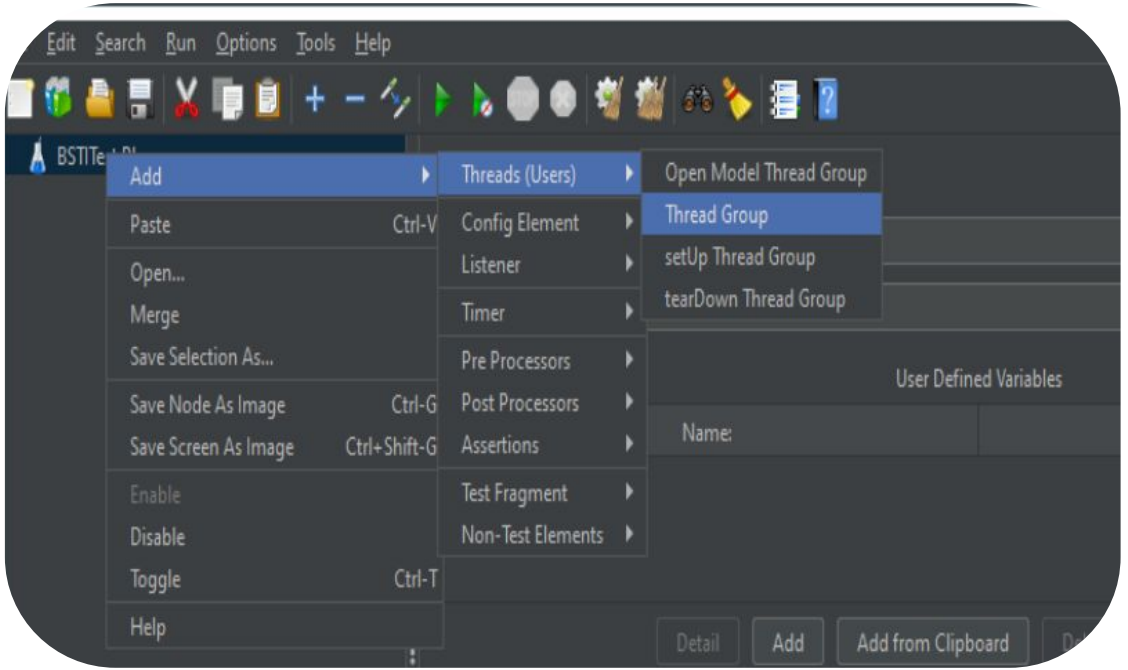


JMeter Thread Group

Thread group elements are the beginning points of any test plan. All controllers and samplers must be under a thread group.

The controls for a thread group allow you to:

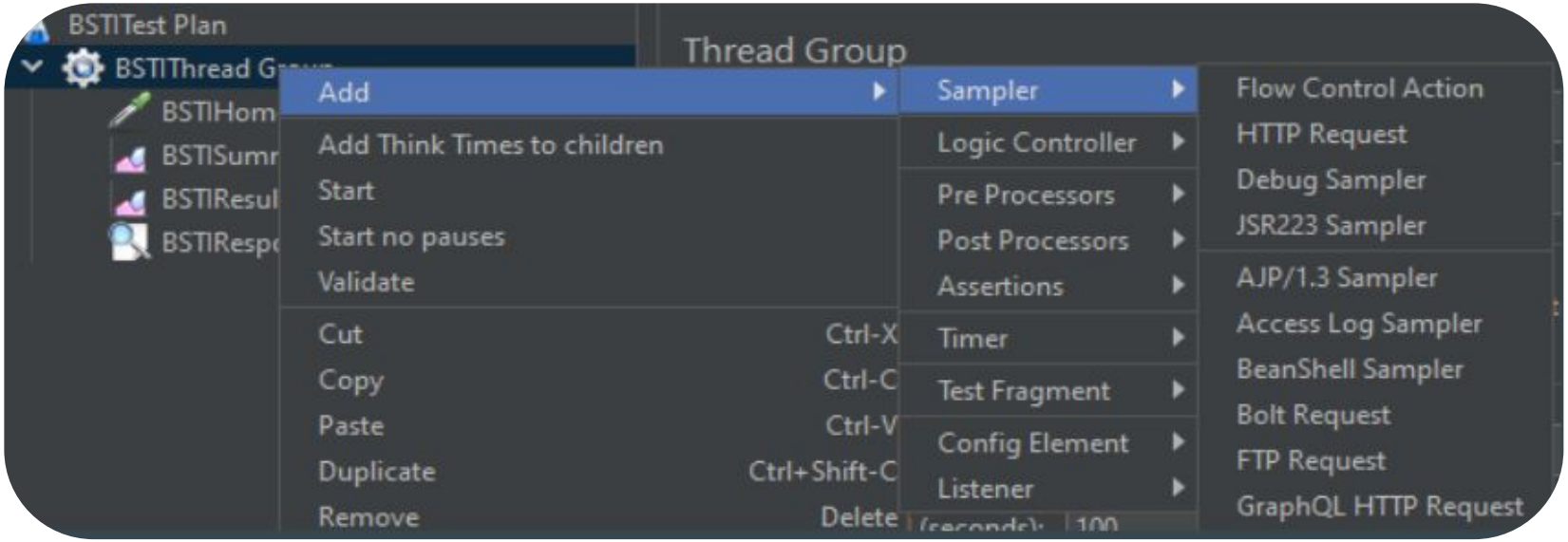
- ➔ Set the number of threads (user).
- ➔ Set the ramp-up period.
- ➔ Set the number of times to execute the test.



JMeter Controllers

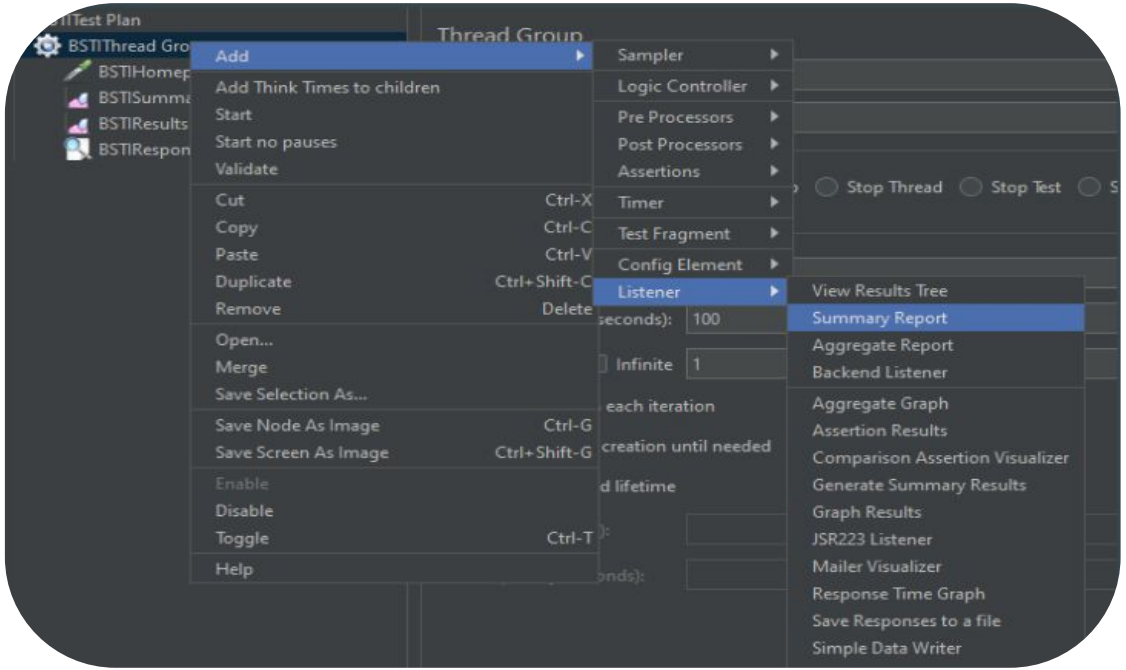
JMeter has two types of Controllers: Samplers and Logical Controllers. These drive the processing of a test.

Samplers, tell JMeter to send requests to a server. For example, add an HTTP Request Sampler if you want JMeter to send an HTTP request. You can also
customize a request by adding one or more Configuration Elements to a Sampler.



JMeter Listeners

Listeners provide access to the information JMeter gathers about the test cases while JMeter runs.
The Graph Results listener plots the response times on a graph.
Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level.



Timers, Assertions

Timers, By default, a JMeter thread sends requests without pausing between each request. We recommend that you specify a delay by adding one of the available timers to your Thread Group.

Assertions, Assertions allow you to assert facts about responses received from the server being tested. Using an assertion, you can essentially "test" that your application is returning the results you expect it to.

Generate HTML Dashboard Report From Command Line

```
jmeter -n -t <location of jmeter script> -l <location of result file>
-e -o <location of output folder>
```

```
My cmd “jmeter -n -t "BSTIThread Group.jmx" -l CSV_file/bstitest.csv
-e -o html_file/bstitest.html ”
```

-n specifies the JMeter is to run in cli mode.

-t specifies name of JMX file that contains the Test Plan.

-l specifies name of result file to log sample results to. The result file must not exist or be empty.

-e specifies generate report dashboard after load test

-o specifies output folder where to generate the report dashboard after load test. Folder must not exist or be empty.

Result View (HTML Report)

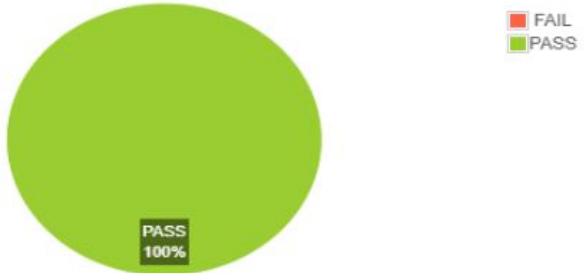
Test and Report information

Source file	"bstitest.csv"
Start Time	"1/31/23 9:57 AM"
End Time	"1/31/23 9:59 AM"
Filter for display	""

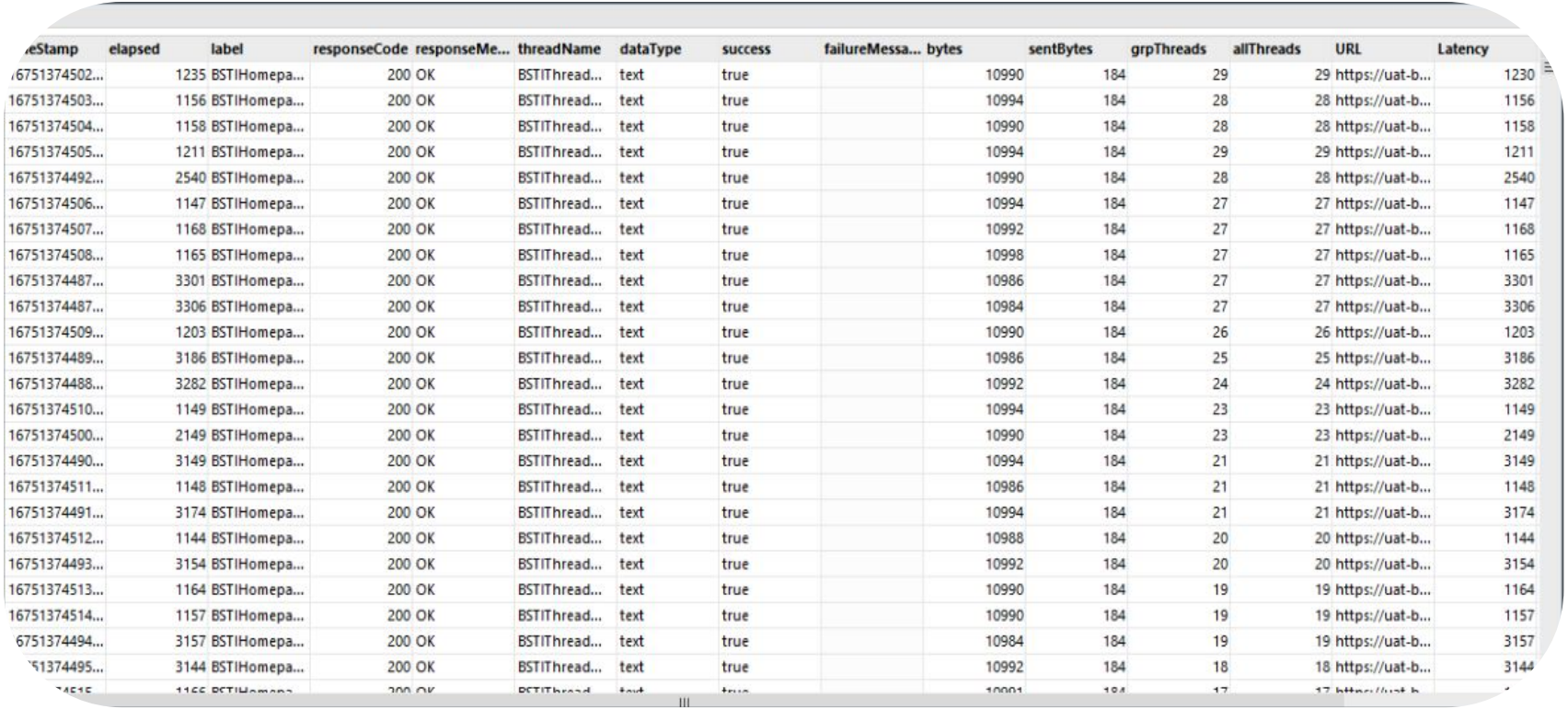
APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.559	**500 ms**	**1 sec 500 ms**	**Total**
0.559	500 ms	1 sec 500 ms	BSTIHomepage Request

Requests Summary



Result View (CSV)



eStamp	elapsed	label	responseCode	responseMe...	threadName	dataType	success	failureMessa...	bytes	sentBytes	grpThreads	allThreads	URL	Latency
16751374502...	1235	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	29	29	https://uat-b...	1230
16751374503...	1156	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	28	28	https://uat-b...	1156
16751374504...	1158	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	28	28	https://uat-b...	1158
16751374505...	1211	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	29	29	https://uat-b...	1211
16751374492...	2540	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	28	28	https://uat-b...	2540
16751374506...	1147	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	27	27	https://uat-b...	1147
16751374507...	1168	BSTIHomepa...	200	OK	BSTIThread...	text	true		10992	184	27	27	https://uat-b...	1168
16751374508...	1165	BSTIHomepa...	200	OK	BSTIThread...	text	true		10998	184	27	27	https://uat-b...	1165
16751374487...	3301	BSTIHomepa...	200	OK	BSTIThread...	text	true		10986	184	27	27	https://uat-b...	3301
16751374487...	3306	BSTIHomepa...	200	OK	BSTIThread...	text	true		10984	184	27	27	https://uat-b...	3306
16751374509...	1203	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	26	26	https://uat-b...	1203
16751374489...	3186	BSTIHomepa...	200	OK	BSTIThread...	text	true		10986	184	25	25	https://uat-b...	3186
16751374488...	3282	BSTIHomepa...	200	OK	BSTIThread...	text	true		10992	184	24	24	https://uat-b...	3282
16751374510...	1149	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	23	23	https://uat-b...	1149
16751374500...	2149	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	23	23	https://uat-b...	2149
16751374490...	3149	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	21	21	https://uat-b...	3149
16751374511...	1148	BSTIHomepa...	200	OK	BSTIThread...	text	true		10986	184	21	21	https://uat-b...	1148
16751374491...	3174	BSTIHomepa...	200	OK	BSTIThread...	text	true		10994	184	21	21	https://uat-b...	3174
16751374512...	1144	BSTIHomepa...	200	OK	BSTIThread...	text	true		10988	184	20	20	https://uat-b...	1144
16751374493...	3154	BSTIHomepa...	200	OK	BSTIThread...	text	true		10992	184	20	20	https://uat-b...	3154
16751374513...	1164	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	19	19	https://uat-b...	1164
16751374514...	1157	BSTIHomepa...	200	OK	BSTIThread...	text	true		10990	184	19	19	https://uat-b...	1157
6751374494...	3157	BSTIHomepa...	200	OK	BSTIThread...	text	true		10984	184	19	19	https://uat-b...	3157
51374495...	3144	BSTIHomepa...	200	OK	BSTIThread...	text	true		10992	184	18	18	https://uat-b...	3144

Java

Selenium

Automation Testing Using Selenium

Automation Testing is a software testing technique in which automated software tools and scripts are used to perform tests on a software application.

Video 1 : Automation Testing Using Selenium - Part 1

Video 2: Automation Testing Using Selenium - Part2

Reference

https://locust.io

https://dev.to/bedindavide/locust-on-azure-an-end-to-end-experience-48f7

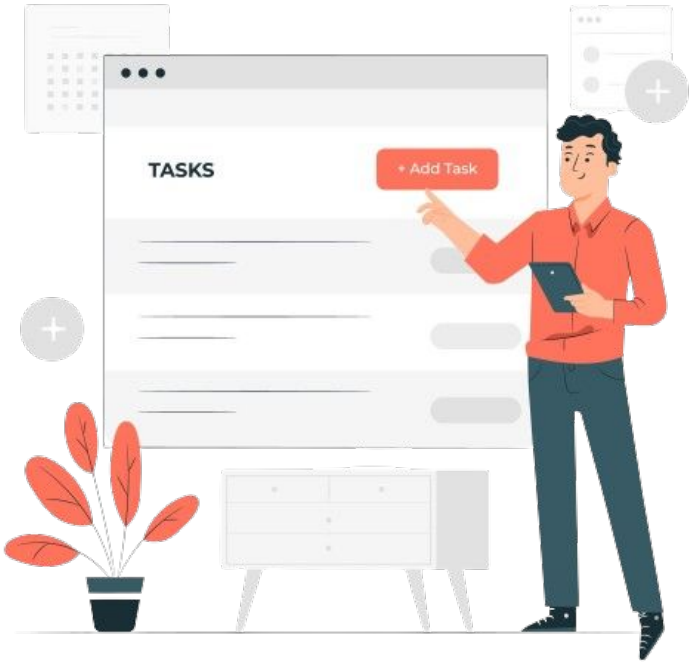
https://jmeter.apache.org

https://learn.microsoft.com/en-us/azure/architecture/example-scenario/...

https://www.numpyninja.com/post/run-jmeter-and-generate-html...

https://intersog.com/blog/load-testing-via-locust-framework/

Assignment



Task 1 :
Submit a 3–4 minute screen recording video with the complete procedure of load testing using Locust.

Task 2 :
Generate a load testing report using locust.

THANKS!

WebCrafter Team , Business Automation Ltd
www.ba-systems.com