# **Jmeter-Testing with Docker**

v1.2

[**https://github.com/NavithuSriyananda/Jmeter-Testing**](https://github.com/NavithuSriyananda/Jmeter-Testing)

## **Tools Used**

* apache-jmeter-5.3
* openjdk-8-jdk
* docker

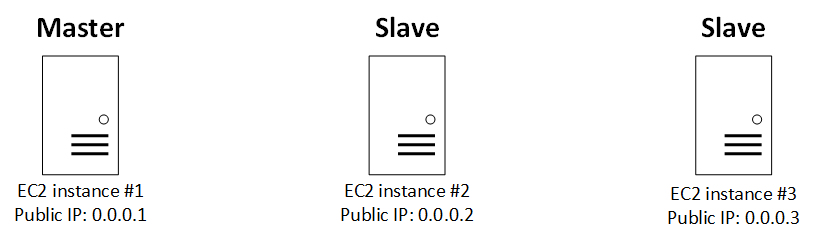
## **File Structure**

|  |  |
| --- | --- |
| **File** | **Description** |
| Jmeter base image | used as the foundation for both master and slave images |
| Master image | used to run master container. It populates with base image. |
| Slave image | used to run slave container. It populates with base image. |
| entrypoint.sh | This script file used to configure JVM. Also configure slaves (server public IP, Ports). This executes when container starts |

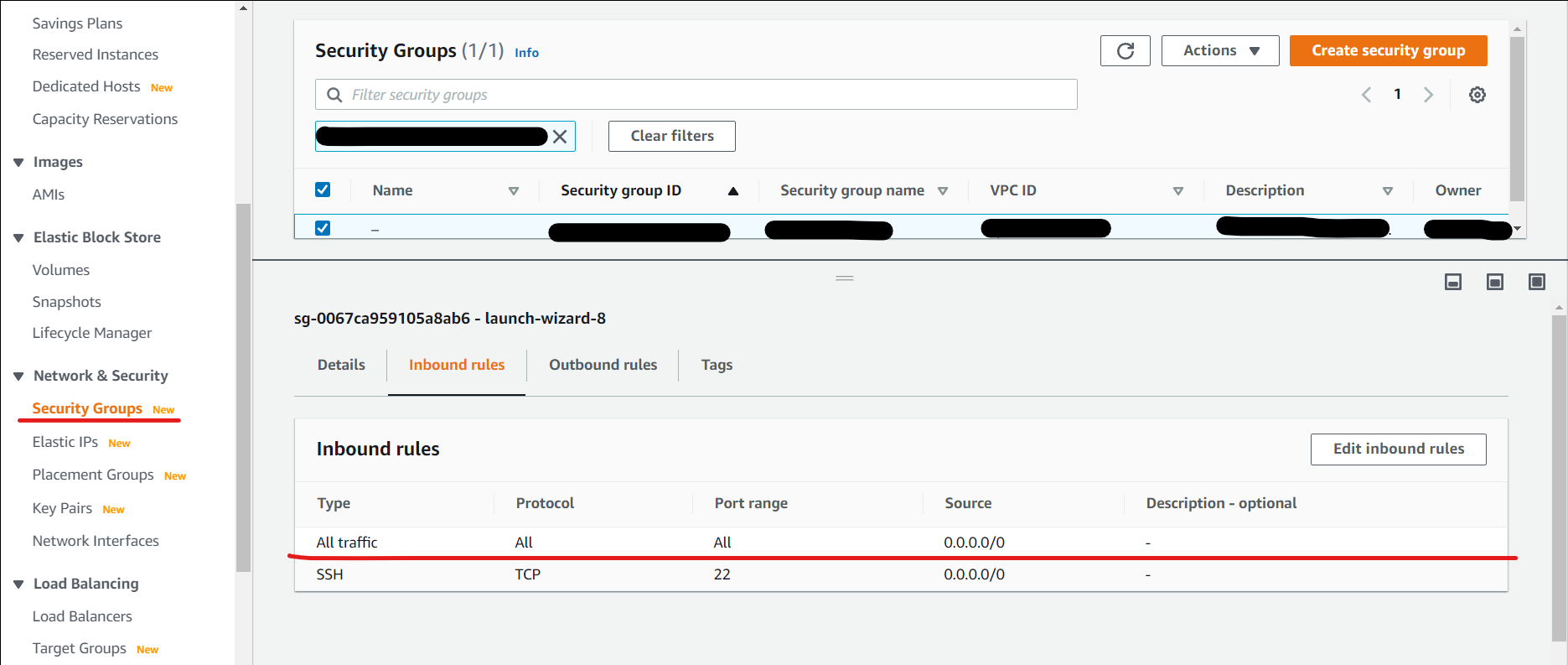
## **Images in Dockerhub**

|  |  |
| --- | --- |
| **Image** | **Description** |
| Navithu/base:1.0 | Base image without plugins |
| Navithu/base:1.1 | Base image with plugins installed |
| Navithu/master | Master image |
| Navithu/slave | Slave image |

# **Getting started**



**Please allow all inbound traffic for the EC2 instances**



**First 2 steps are optional. You can start with step 3. If you want to start from step 1, Try this :** [**https://github.com/NavithuSriyananda/Jmeter-Testing**](https://github.com/NavithuSriyananda/Jmeter-Testing)

**Install docker.io to begin. Refer docker cheat sheet at end of document.**

## **Build images\*OPTIONAL ( base , master , slave)**

* Example

docker build -t navithu/base .

**\*OPTIONAL**

## **Push images to dockerhub ( base , master, slave)**

* Example

docker push navithu/base

## **Run containers from images**

**Set JVM settings**

-e Xmn=512m –e Xmx=512m –e MaxMetaspaceSize=1024m

**Xmn - Minimum java heap size (default 512m)**

**Xmx - Maximum java heap size (default 512m)**

**MaxMetaspaceSize - Space that manages class information (default 1024m)**

**-**

* EC2 #1 - Master

docker run -dit -e Xmn=512m -e Xmx=512m -e MaxMetaspaceSize=1024m --name master --network host navithu/master /bin/bash

* EC2 #2 – Slave

**Exclude this part of code if you are trying locally.** -e PublicIP='0.0.0.2'

**To run more slaves in one machine, Assign different ports to avoid conflict.**

-e p1=1099 –e p2=50000

**p1 - Dserver\_port (default 1099)**

**p2 -** **Dserver.rmi.localport (default 50000)**

docker run -dit -e PublicIP='0.0.0.2' -e p1=1099 -e p2=50000 -e Xmn=512m -e Xmx=512m -e MaxMetaspaceSize=1024m --name slave --network host navithu/slave /bin/bash

* EC2 #3 - Slave

docker run -dit -e PublicIP='0.0.0.3' -e p1=1099 -e p2=50000 -e Xmn=512m -e Xmx=512m -e MaxMetaspaceSize=1024m --name slave --network host navithu/slave /bin/bash

## **Testing with distributed nodes**

**Create test,jmx file at root directory before execute following command.**

* EC2 #1 – Master

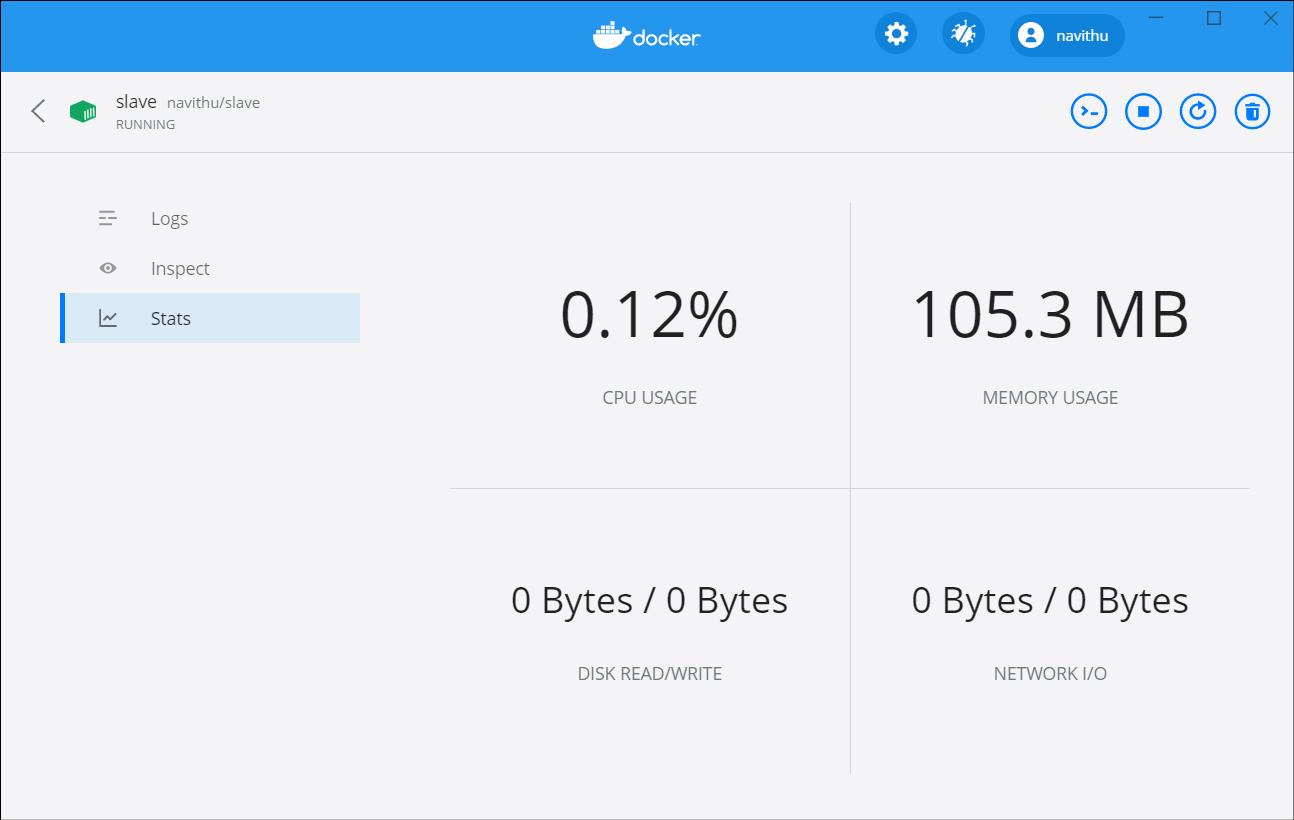
**Execute below command from master container’s terminal.**

**Exclude this part of code if you are trying locally.** -Djava.rmi.server.hostname=0.0.0.1

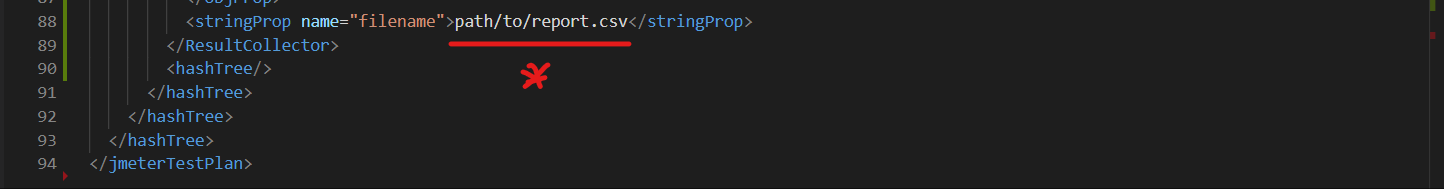
**Specify ports if more slaves are running in one machine.** -R0.0.0.1:1099,0.0.0.1:1100

jmeter -n -t test.jmx -Djava.rmi.server.hostname=0.0.0.1 -Dclient.rmi.localport=60000 -Jserver.rmi.ssl.disable=true -R0.0.0.2,0.0.0.3

**Check slave stats . You can expect high CPU USAGE and MEMORY USAGE while testing.**



* export report - \*\*Add 'Report' from listeners to test.jmx file and define the save location



## 5. **Plugin Manager**

* Sample commands

PluginsManagerCMD.sh help

PluginsManagerCMD.sh status

PluginsManagerCMD.sh upgrades

PluginsManagerCMD.sh available

PluginsManagerCMD.sh install jpgc-fifo,jpgc-json=2.2

PluginsManagerCMD.sh install-all-except jpgc-casutg,jpgc-autostop

PluginsManagerCMD.sh uninstall jmeter-tcp,jmeter-ftp,jmeter-jdbc

PluginsManagerCMD.sh install-for-jmx /home/username/jmx/testPlan.jmx

## **Testing plugins**

**Open jmeter GUI from “apache-jmeter-5.3/bin/jmeter.bat”**

**Open example jmx files and click “START” button**

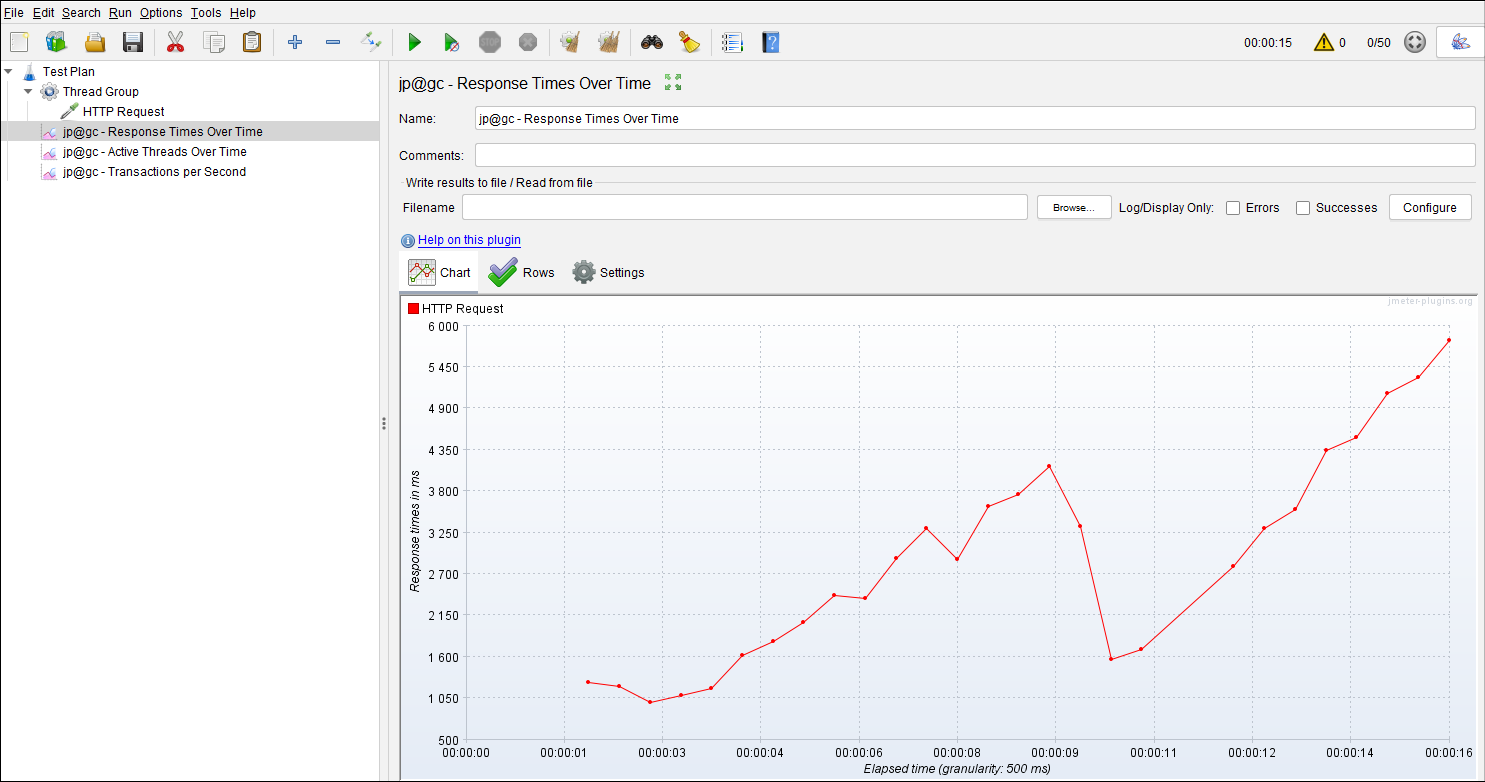


### **3 Basic graphs -** [**https://jmeter-plugins.org/wiki/ResponseTimesOverTime**](https://jmeter-plugins.org/wiki/ResponseTimesOverTime)

**Test Plan**

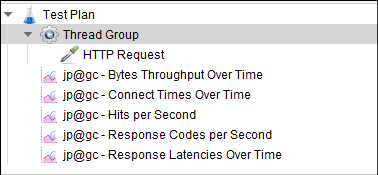


**Example result**



### **5 Additional Graphs -** [**https://jmeter-plugins.org/wiki/ResponseCodesPerSecond**](https://jmeter-plugins.org/wiki/ResponseCodesPerSecond)

**Test Plan**

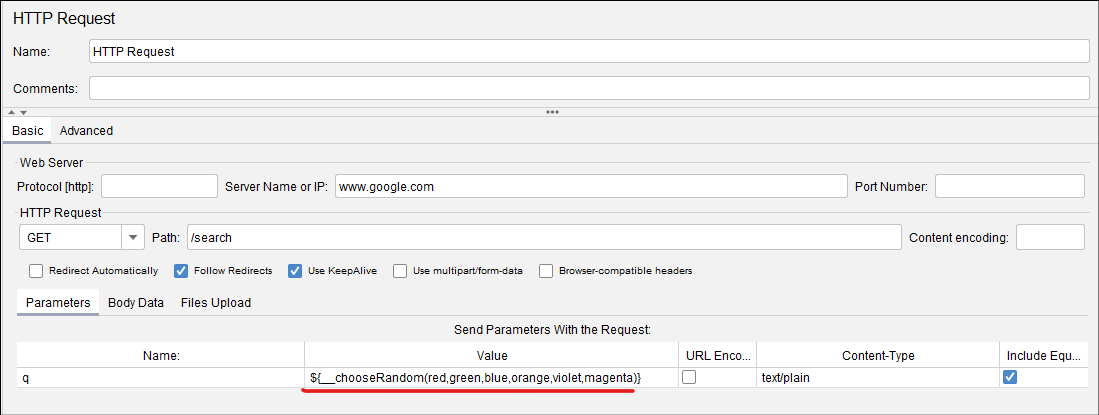


### **Custom JMeter Functions -** [**https://jmeter-plugins.org/wiki/Functions**](https://jmeter-plugins.org/wiki/Functions)

**Test Plan**

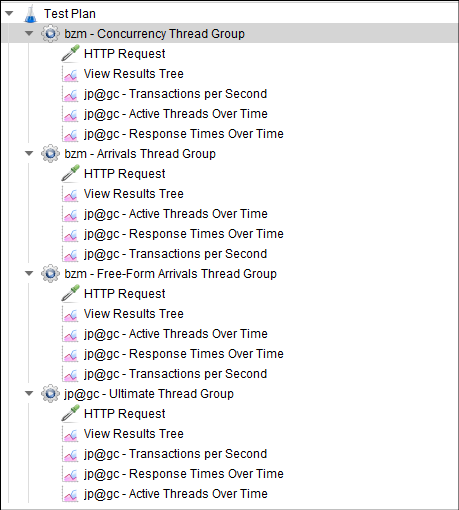


**Example function**



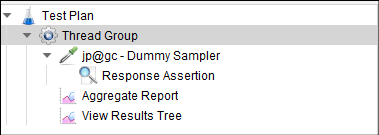
### **Custom Thread Groups -** [**https://jmeter-plugins.org/wiki/ConcurrencyThreadGroup**](https://jmeter-plugins.org/wiki/ConcurrencyThreadGroup)

**Test Plan**



### **Dummy Sampler -** [**https://jmeter-plugins.org/wiki/DummySampler**](https://jmeter-plugins.org/wiki/DummySampler)

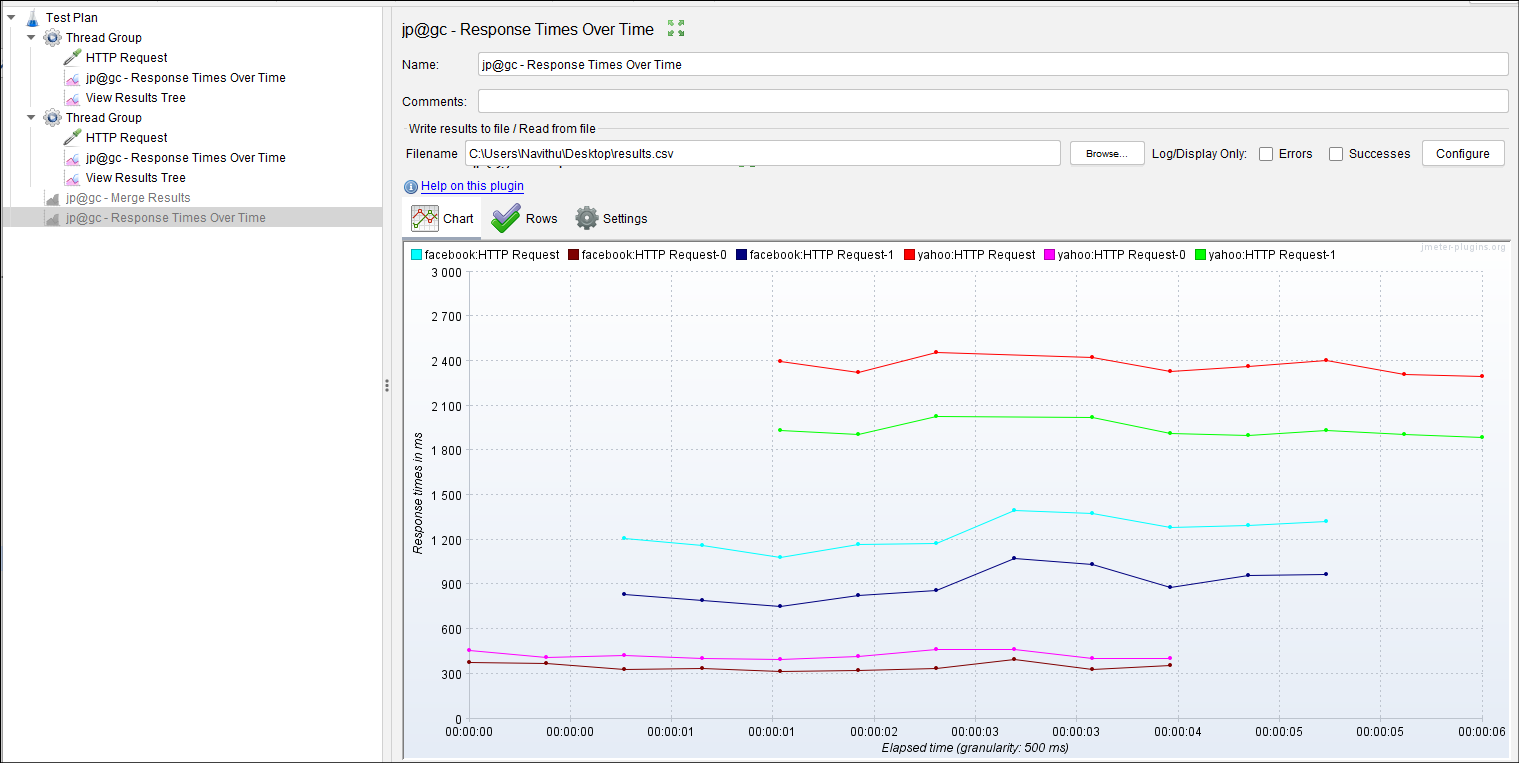
**Test Plan**



### **Merge Results -** [**https://jmeter-plugins.org/wiki/MergeResults**](https://jmeter-plugins.org/wiki/MergeResults)

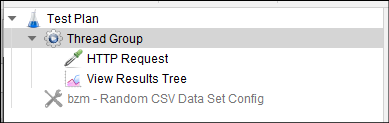
**After test stopped, Merge the reports from “jp@gc - Merge Results”.**

**Example result**



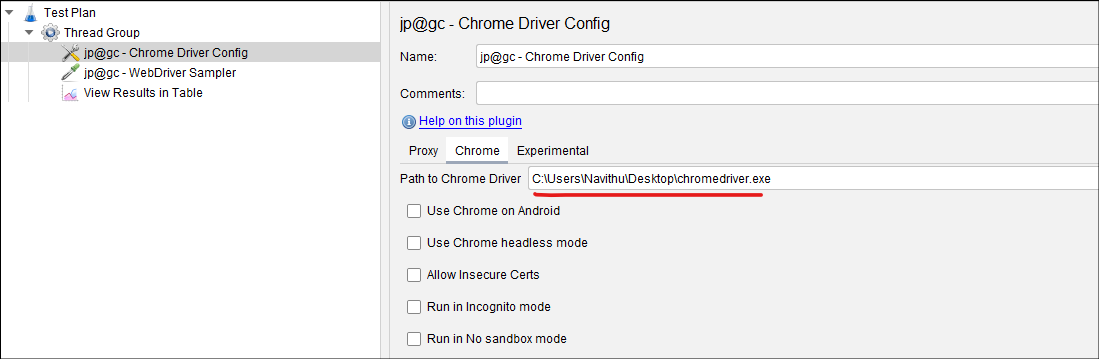
### **Random CSV Data Set -** [**https://github.com/Blazemeter/jmeter-bzm-plugins/blob/master/random-csv-data-set/RandomCSVDataSetConfig.md**](https://github.com/Blazemeter/jmeter-bzm-plugins/blob/master/random-csv-data-set/RandomCSVDataSetConfig.md)

**Test Plan**



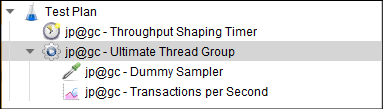
### **Selenium/WebDriver Support -** [**https://github.com/undera/jmeter-plugins-webdriver**](https://github.com/undera/jmeter-plugins-webdriver)

**Select chrome web driver path before testing.**



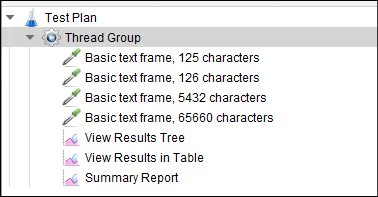
### **Throughput Shaping Timer -** [**https://jmeter-plugins.org/wiki/ThroughputShapingTimer**](https://jmeter-plugins.org/wiki/ThroughputShapingTimer)

**Test Plan**



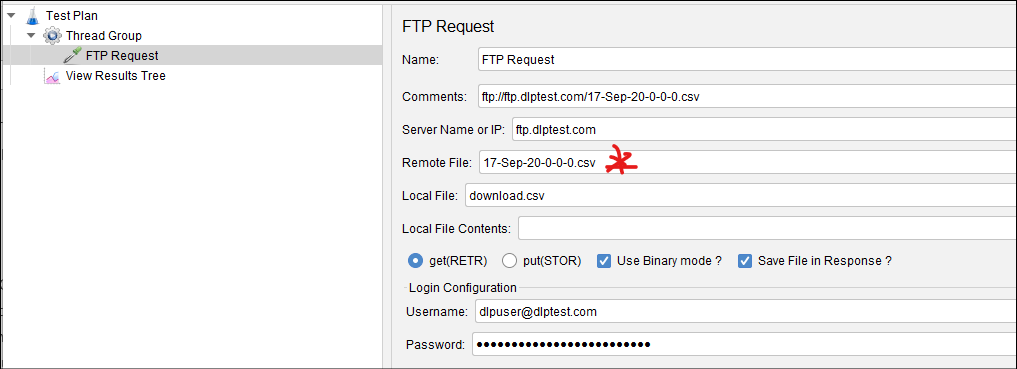
### **WebSocket Samplers by Peter Doornbosch -** [**https://bitbucket.org/pjtr/jmeter-websocket-samplers/src/master**](https://bitbucket.org/pjtr/jmeter-websocket-samplers/src/master)

**Test Plan**



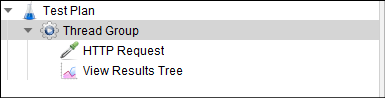
### **FTP Protocol Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#FTP\_Request**](https://jmeter.apache.org/usermanual/component_reference.html#FTP_Request)

**Get FTP test data from here. Select a valid file in server -** [**https://dlptest.com/ftp-test**](https://dlptest.com/ftp-test)



### **HTTP Protocol Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#HTTP\_Request**](https://jmeter.apache.org/usermanual/component_reference.html#HTTP_Request)

**Test Plan**

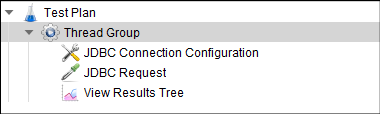


### **JDBC Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#JDBC\_Request**](https://jmeter.apache.org/usermanual/component_reference.html#JDBC_Request)

**Tested with mysql server.**

**Copy the mysql-connector-java.jar to “apache-jmeter-5.3/lib” and restart jmeter**

**Test Plan**

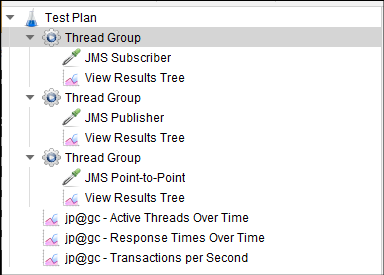


### **JMS Support -**

[**https://jmeter.apache.org/usermanual/component\_reference.html#JMS\_Subscriber**](https://jmeter.apache.org/usermanual/component_reference.html#JMS_Subscriber)[**https://jmeter.apache.org/usermanual/component\_reference.html#JMS\_Publisher**](https://jmeter.apache.org/usermanual/component_reference.html#JMS_Publisher)[**https://jmeter.apache.org/usermanual/component\_reference.html#JMS\_Point-to-Point**](https://jmeter.apache.org/usermanual/component_reference.html#JMS_Point-to-Point)

**Run Apache-ActiveMQ from “apache-activemq-5.4.0/bin/activemq.bat” before testing**

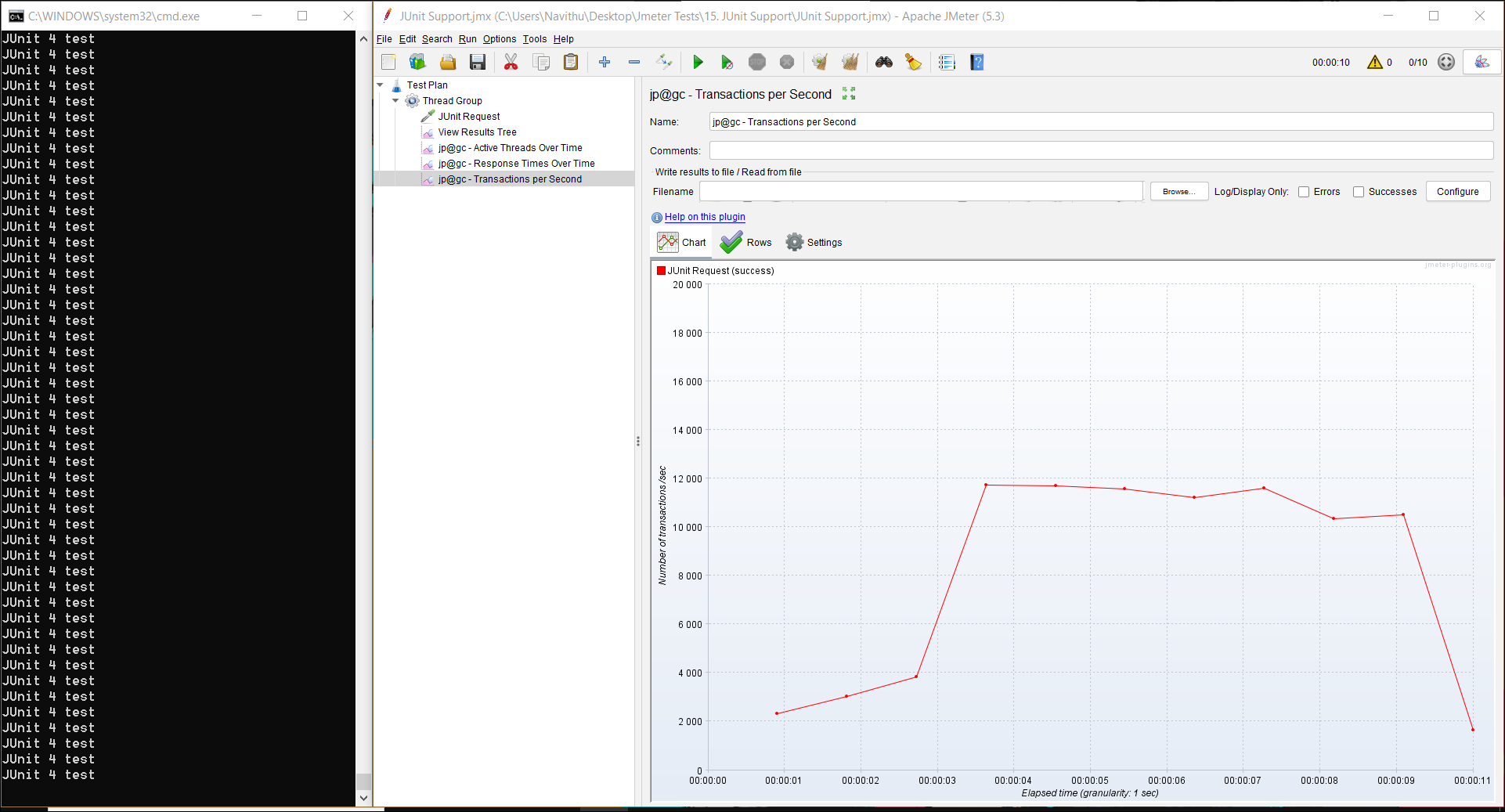
**Test Plan**



### **JUnit Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#JUnit\_Request**](https://jmeter.apache.org/usermanual/component_reference.html#JUnit_Request)

**Copy the JUnitTest.jar to “apache-jmeter-5.3/lib/junit” and restart jmeter**

**Example result**



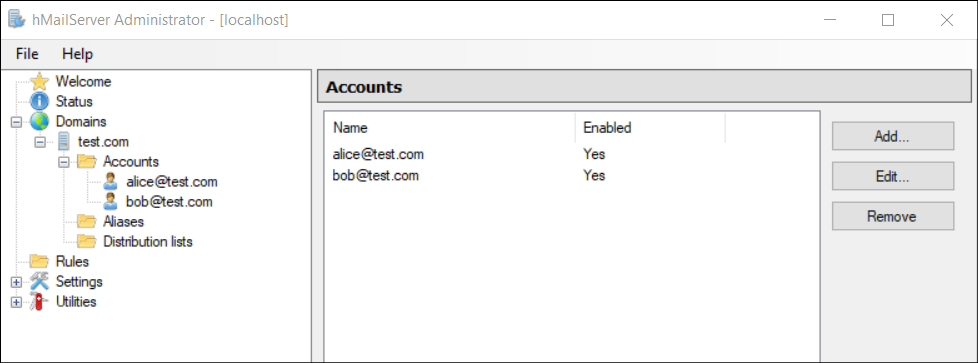
### **Java Components ( Not tested )**

### **LDAP Protocol Support ( Not tested ) - https://jmeter.apache.org/usermanual/component\_reference.html#LDAP\_Request**

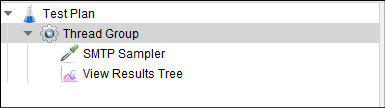
### **Mail/SMTP Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#SMTP\_Sampler**](https://jmeter.apache.org/usermanual/component_reference.html#SMTP_Sampler)

**Install and Run hMailServer.**

**Add domain and accounts.**



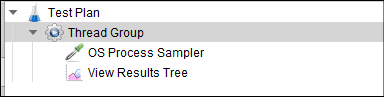
**Test Plan**



### **MongoDB Support (DEPRECATED) -** [**https://jmeter.apache.org/usermanual/component\_reference.html#MongoDB\_Script\_(DEPRECATED)**](https://jmeter.apache.org/usermanual/component_reference.html#MongoDB_Script_(DEPRECATED))

### **OS Process Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#OS\_Process\_Sampler**](https://jmeter.apache.org/usermanual/component_reference.html#OS_Process_Sampler)

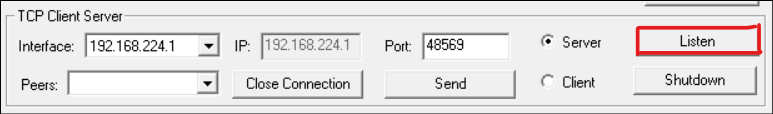
**Test Plan**



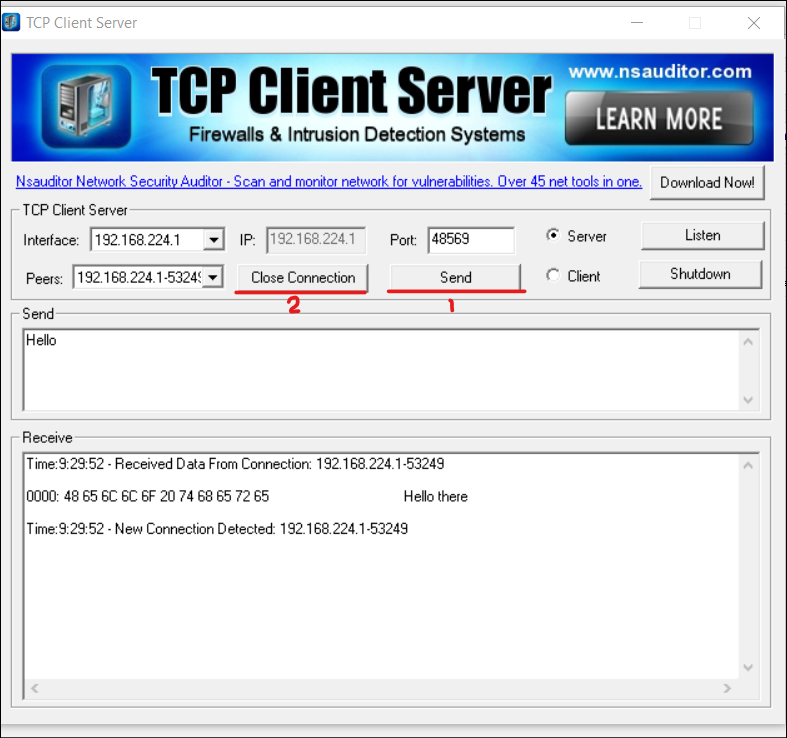
### **OS TCP Protocol Support -** [**https://jmeter.apache.org/usermanual/component\_reference.html#TCP\_Sampler**](https://jmeter.apache.org/usermanual/component_reference.html#TCP_Sampler)

**Install and Run TcpClientServer.**

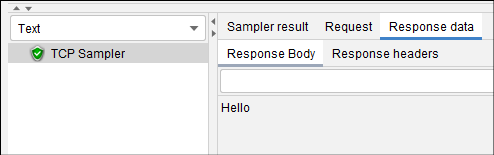
**Start listen before testing.**



**Send reply and close the connection.**



**Example result**



### **Various Core Components ( Not tested )**

# **-Docker Cheat Sheet-**

**-Global-**

|  |  |
| --- | --- |
| **Description** | **Command** |
| Install docker in linux | sudo apt-get update && sudo apt install docker.io |
| Login to dockerhub | docker login |
| Set docker socket permission | sudo chmod 777 /var/run/docker.sock |
| System reset | docker system prune -a |
| Create test file and save | cat > test.jmx > sample text |

**-Images-**

|  |  |
| --- | --- |
| **Description** | **Command** |
| **Build** image and tag | docker build -t navithu/MyImage . |
| **Push** to dockerhub | docker push navithu/MyImage . |
| **Pull** from docker hub | docker pull navithu/MyImage . |
| View all | docker images |
| Tag | docker tag 123456 navithu/MyImage |
| Remove | docker rmi navithu/MyImage |
| Remove all | docker image prune -a |

**-Containers -**

|  |  |
| --- | --- |
| **Description** | **Command** |
| Run | docker run -dit --name MyContainer navithu/MyImage /bin/bash |
| Start | docker start MyContainer |
| Stop | docker stop MyContainer |
| Access container | docker exec -it MyContainer /bin/bash |
| View all | docker ps -a |
| Remove | docker rm MyContainer |
| Remove all | docker container prune |