Practica 4 - Benchmarks

Phoronix

1. Instalamos phoronix con el comando:

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
juanka1995@juanka1995-Laptop ~ $ sudo apt-get install phoronix-test-suite
```

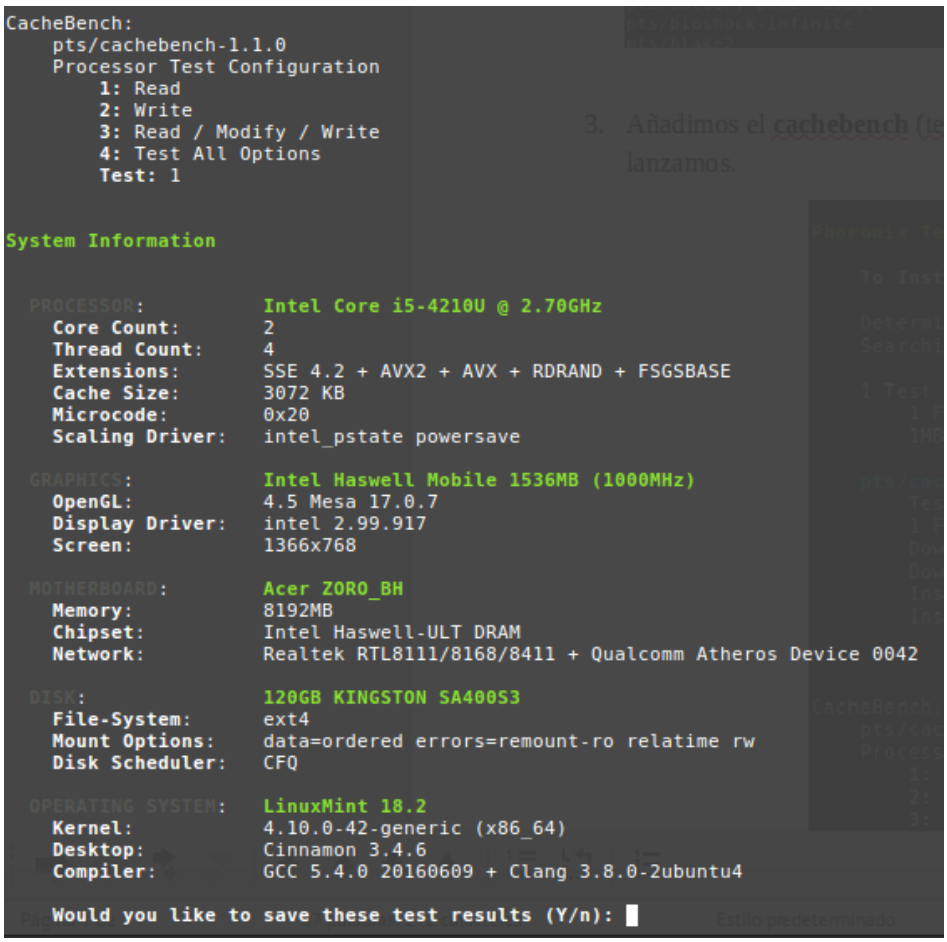
2. Listamos los test disponibles e instalamos 2 de ellos.

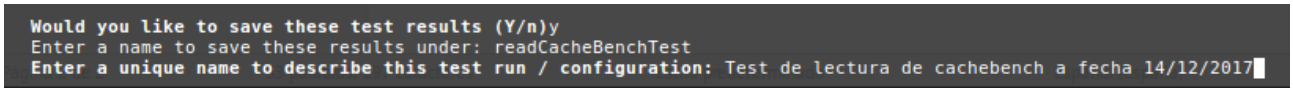
```
juanka1995@juanka1995-Laptop ~ $ phoronix-test-suite list-available-tests

Phoronix Test Suite v7.6.0
Available Tests

pts/aio-stress                - AIO-Stress                        Disk
pts/apache                    - Apache Benchmark                  System
pts/apitest                   - APITest                           Graphics
pts/apitrace                  - APITrace                          Graphics
pts/arrayfire                 - ArrayFire                         Processor
pts/askap                     - ASKAP tConvolveCuda               Graphics
pts/asmfish                   - asmFish                           Processor
pts/battery-power-usage       - Battery Power Usage               System
pts/bioshock-infinite         - BioShock Infinite                 Graphics
pts/blake2                    - BLAKE2                            Processor
```

3. Añadimos el **cachebench** (testea el rendimiento de la memoria cache del procesador y del ancho de banda) y lo lanzamos.

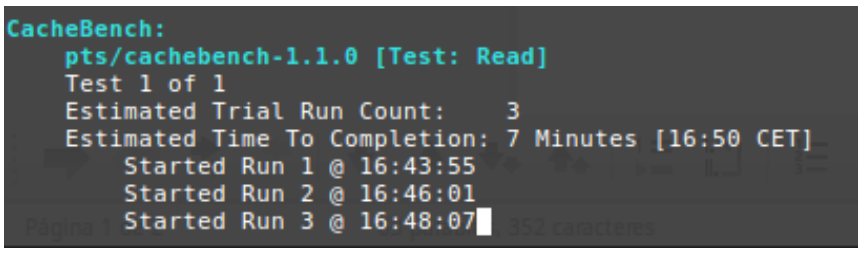
```
Phoronix Test Suite v7.6.0

    To Install:    pts/cachebench-1.1.0

    Determining File Requirements .....
    Searching Download Caches .....

    1 Test To Install
        1 File To Download [0.08MB]
        1MB Of Disk Space Is Needed

    pts/cachebench-1.1.0:
        Test Installation 1 of 1
        1 File Needed [0.08 MB]
        Downloading: llcbench-20170104.tar.gz
        Downloading .....
        Installation Size: 0.5 MB
        Installing Test @ 16:37:38


CacheBench:
    pts/cachebench-1.1.0
    Processor Test Configuration
        1: Read
        2: Write
        3: Read / Modify / Write
        4: Test All Options
        Test:
```

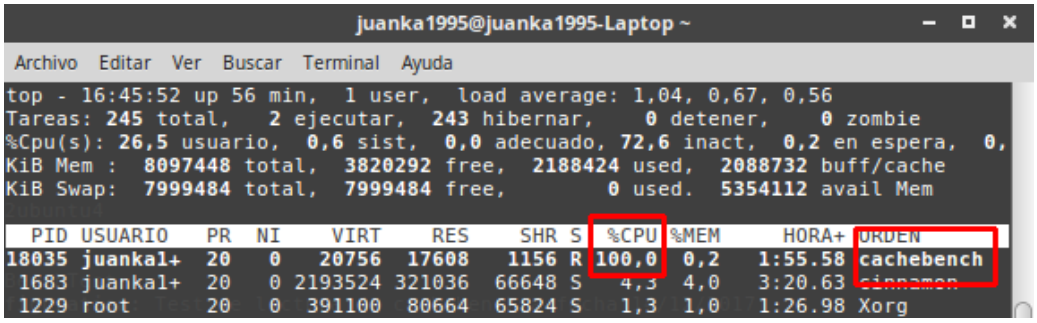
4. Elegimos el test de **READ** y le decimos que queremos guardar los resultados del test.





5. Esperamos a que realice el test.

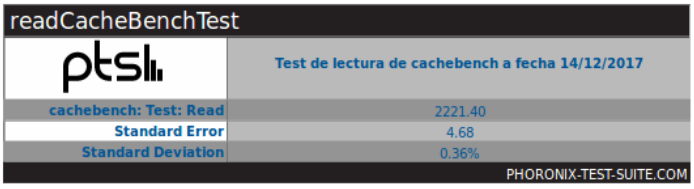




6. Mostramos los resultados obtenidos en nuestro navegador.

```
CacheBench:
    pts/cachebench-1.1.0 [Test: Read]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 7 Minutes [16:50 CET]
        Started Run 1 @ 16:43:55
        Started Run 2 @ 16:46:01
        Started Run 3 @ 16:48:07

    Test: Read:
        2212.3406872857
        2223.8947532857
        2227.968774

    Average: 2221.40 MB/s
    Deviation: 0.36%

    OpenBenchmarking.org Dynamic Comparison:
    MB/s > Higher Is Better
    Test de lectura de cachebench a fecha 14/12/2017 .. 2221.40 |====
    pts/processor ..................................... 2295.98 |====
    Result Perspective: https://openbenchmarking.org/result/1705261-S
```

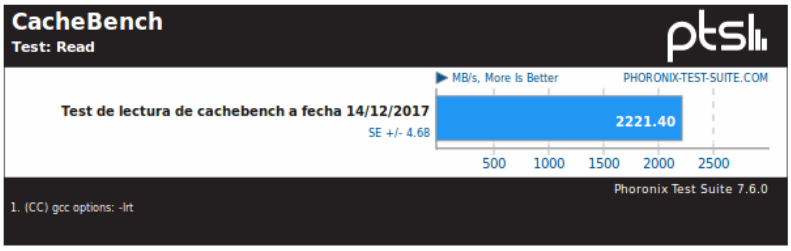
Results Overview



Test Results

CacheBench

Test: Read



7. Añadimos el **sudokut** (testea el rendimiento del procesador para resolver 100 sudokus) y lo lanzamos.

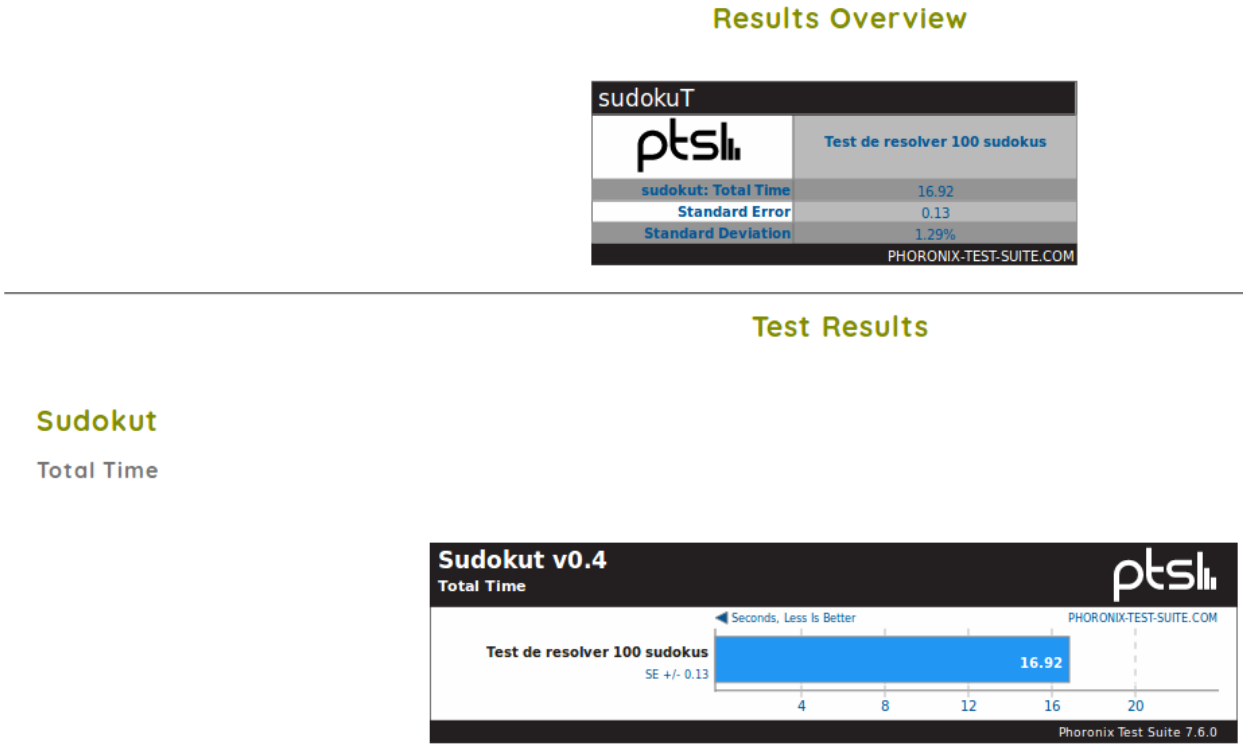
```
juanka1995@juanka1995-Laptop ~ $ phoronix-test-suite benchmark pts/sudokut
```

8. Elegimos el nombre que le pondremos al resultado del test

```
    Would you like to save these test results (Y/n): y

Recently Saved Test Results:
- readcachebenchtest    [Today]

    Enter a name to save these results under: sudokuT
    Enter a unique name to describe this test run / configuration: Test de resolver 100 sudokus
```

9. Esperamos a que termine y mostramos el resultado en nuestro navegador

```
Sudokut 0.4:
    pts/sudokut-1.0.0
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 5 Minutes [17:08 CET]
        Started Run 1 @ 17:03:45
        Started Run 2 @ 17:04:02
        Started Run 3 @ 17:04:20

    Total Time:
        16.767526865005
        16.814805984497
        17.167697906494

    Average: 16.92 Seconds
    Deviation: 1.29%

    Do you want to view the results in your web browser (Y/n):
```



Apache Benchmark

1. Instalamos **apache2** en el anfitrion.

```
juanka1995@juanka1995-Laptop ~ $ sudo apt-get install apache2
```

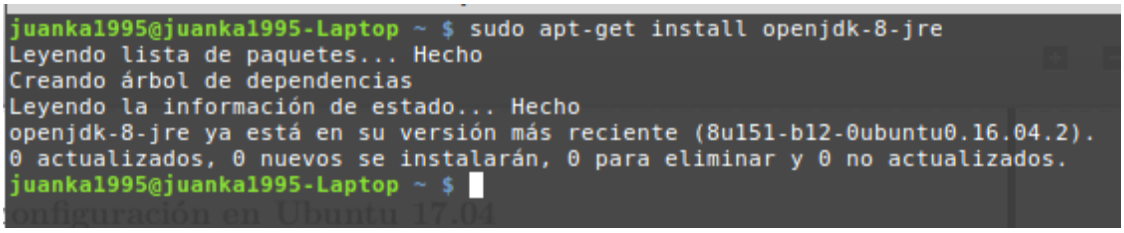
2. Comprobamos que **apache2.service** esta corriendo en el anfitrion y en la maquina virtual de Ubuntu Server.

3. Ejecutamos el ab para hacer un test de peticiones a un servidor HTTP, desde la maquina anfitrión hacia la maquina virtual.

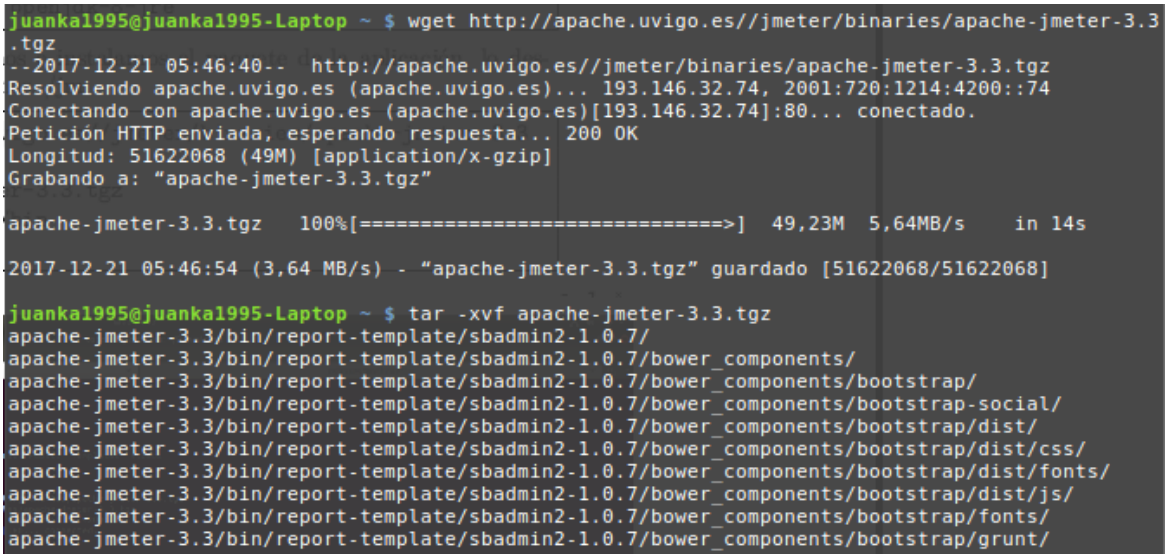
```
juanka1995@juanka1995-Laptop ~ $ ab -n 10000 -c 3 http://192.168.56.105/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.105 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:        Apache/2.4.18
Server Hostname:        192.168.56.105
Server Port:            80

Document Path:          /
Document Length:        11321 bytes    --> tamaño de web devuelta

Concurrency Level:      3
Time taken for tests:   4.337 seconds
Complete requests:      10000
Failed requests:        0
Total transferred:      115950000 bytes
HTML transferred:       113210000 bytes
Requests per second:    2305.56 [#/sec] (mean)
Time per request:       1.301 [ms] (mean)
Time per request:       0.434 [ms] (mean, across all concurrent requests)
Transfer rate:          26106.37 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.4      0       7
Processing:     0    1   0.9      1      20
Waiting:        0    1   0.5      1      17
Total:          1    1   1.0      1      20

Percentage of the requests served within a certain time (ms)
  50%      1
  66%      1
  75%      1
  80%      1
  90%      2                 porcentaje
  95%      3
  98%      5
  99%      6
```

JMeter

1. Lo primero que debemos hacer es instalar Java 8 ya que JMeter es una aplicación implementada en Java.

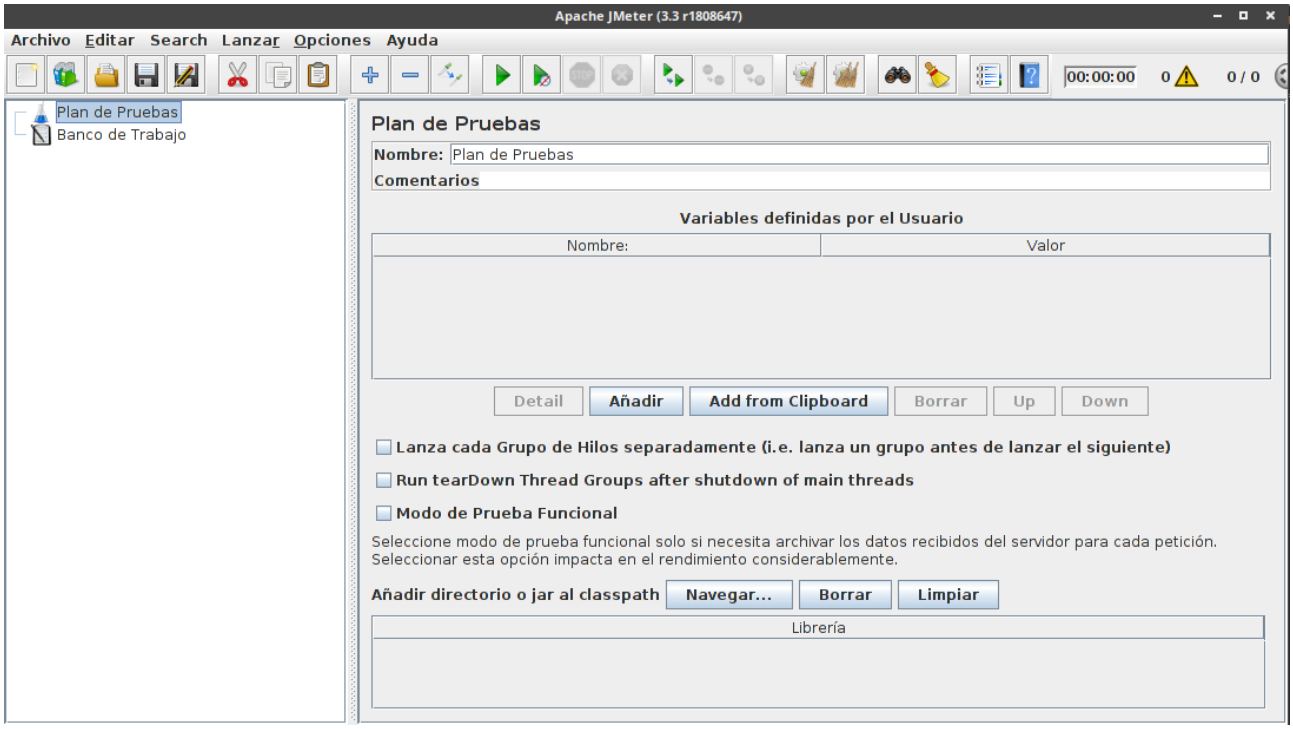


2. Después descargamos el paquete de instalación de la aplicación, lo descomprimimos y seguidamente lo ejecutamos.

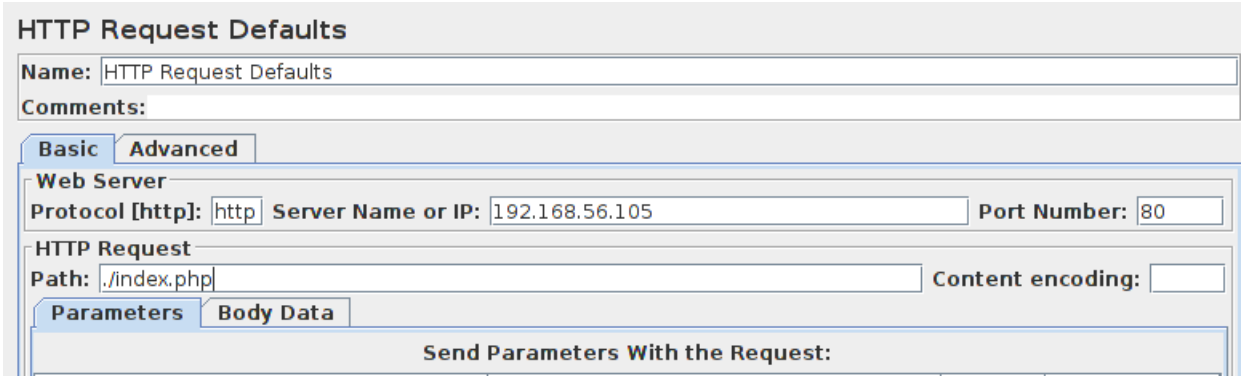


3. Y ahora ya podemos ejecutarlo. Vamos a la carpeta y ejecutamos jmeter.

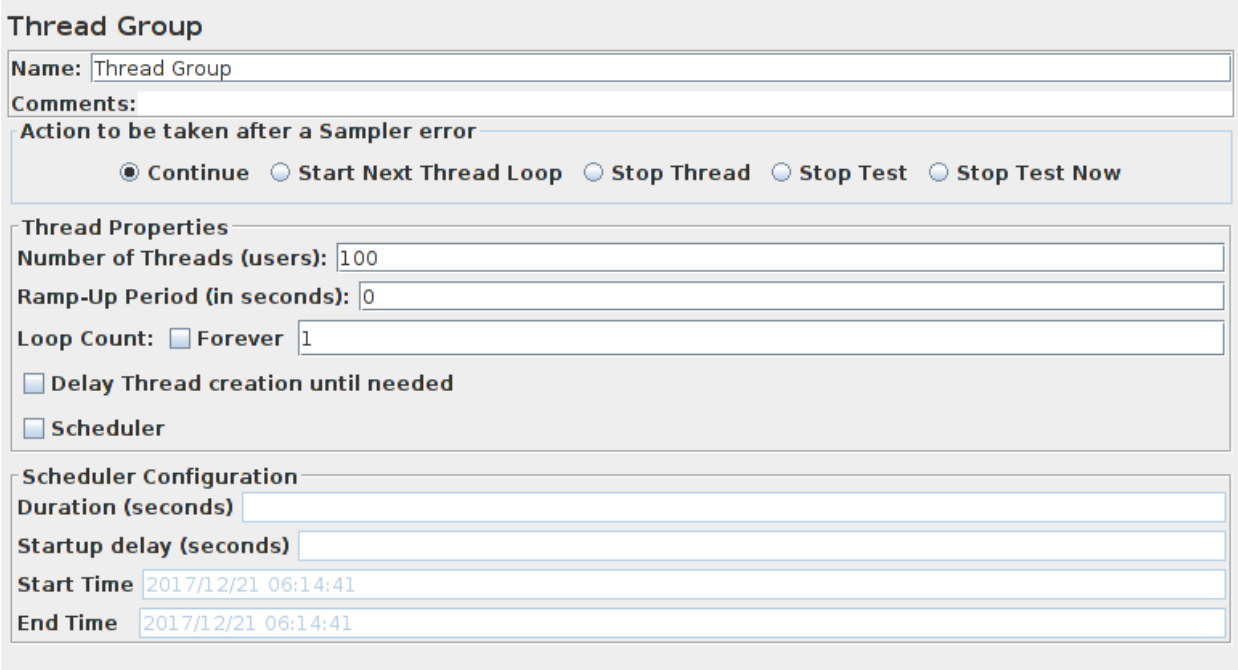
```
juanka1995@juanka1995-Laptop ~ $ cd apache-jmeter-3.3/bin/
juanka1995@juanka1995-Laptop ~/apache-jmeter-3.3/bin $ ./jmeter
```



4. Ahora vamos a crear un test desde 0, para ello click derecho sobre **Test Plan/Add/Config Element/HTTP Request Defaults.** Ahora definimos, el protocolo, la ip, la ruta y el puerto.



5. Después añadimos el gestor de hebras **Test Plan/Add/Threads (Users)/Thread Group**. Lo configuramos de la siguiente forma.



6. Sobre este último, hacemos click derecho en **Thread Group/Add/Sampler/HTTP Request.**

7. Por ultimo añadimos **Test plan/Add/Listener/Aggregate Graph**. Y ya podemos pulsar el botón de play y esperar a que se ejecute el test