**1.Task Description**

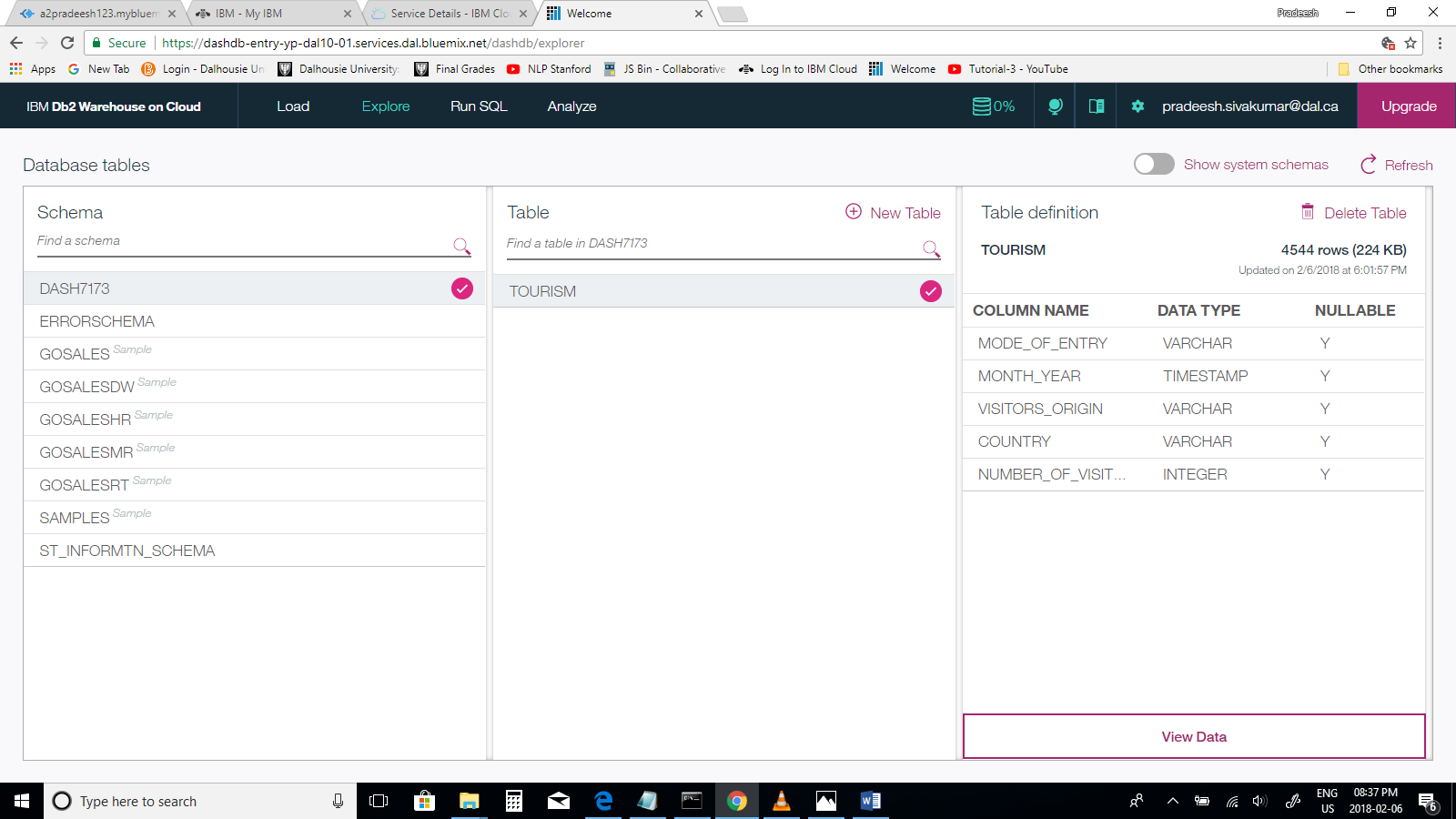
We are supposed to create an account in IBM Bluemix and once logged in we have to use db2 warehouse on cloud to store the given dataset. The dataset given is based on tourism of Nova Scotia from the year starting from 2006 to 2017 and load those dataset in db2 warehouse on cloud. Then we need to write queries which can retrieve first 10 rows from the database, should retrieve number of visitors per year by country, should retrieve number of visitors per year by mode. The next task is to create an application in Node.js or any other application and upload them in Bluemix. Test your web service using Postman Software and comment on the response time that it took for each query. For UI application we need to install Apache JMeter which can send REST service calls to application URL and find the response times. Increase the number of instances to 20,30,40, and 50 and find the five set of requests with the corresponding response time and also draw the graph with number of requests on x axis and response time in y axis. Then we also need to do the horizontal scaling with different instances and find the performance.

**2.Database**

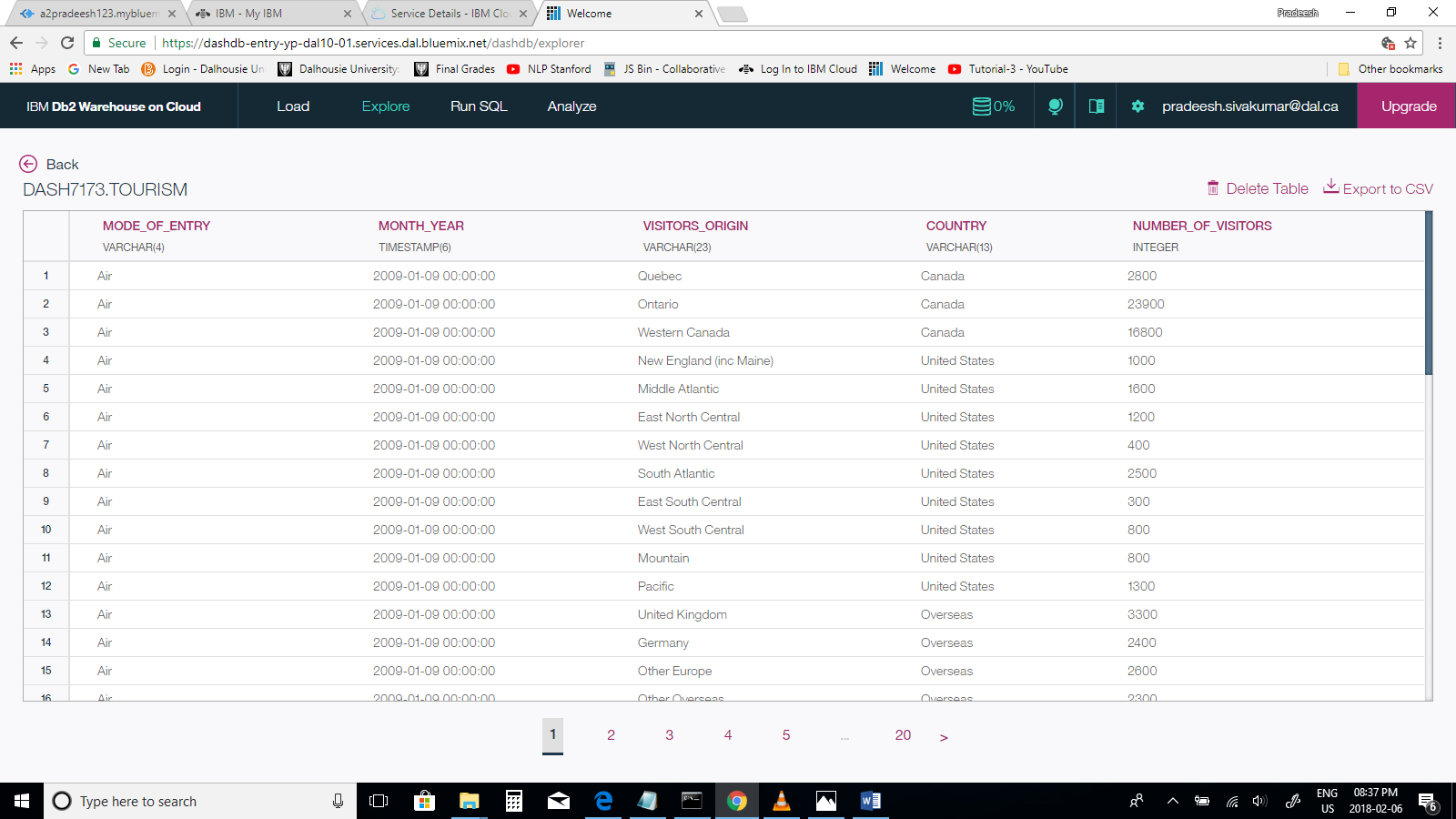
The configuration that are taken while setting up the database.

The screenshots for each steps are given below:

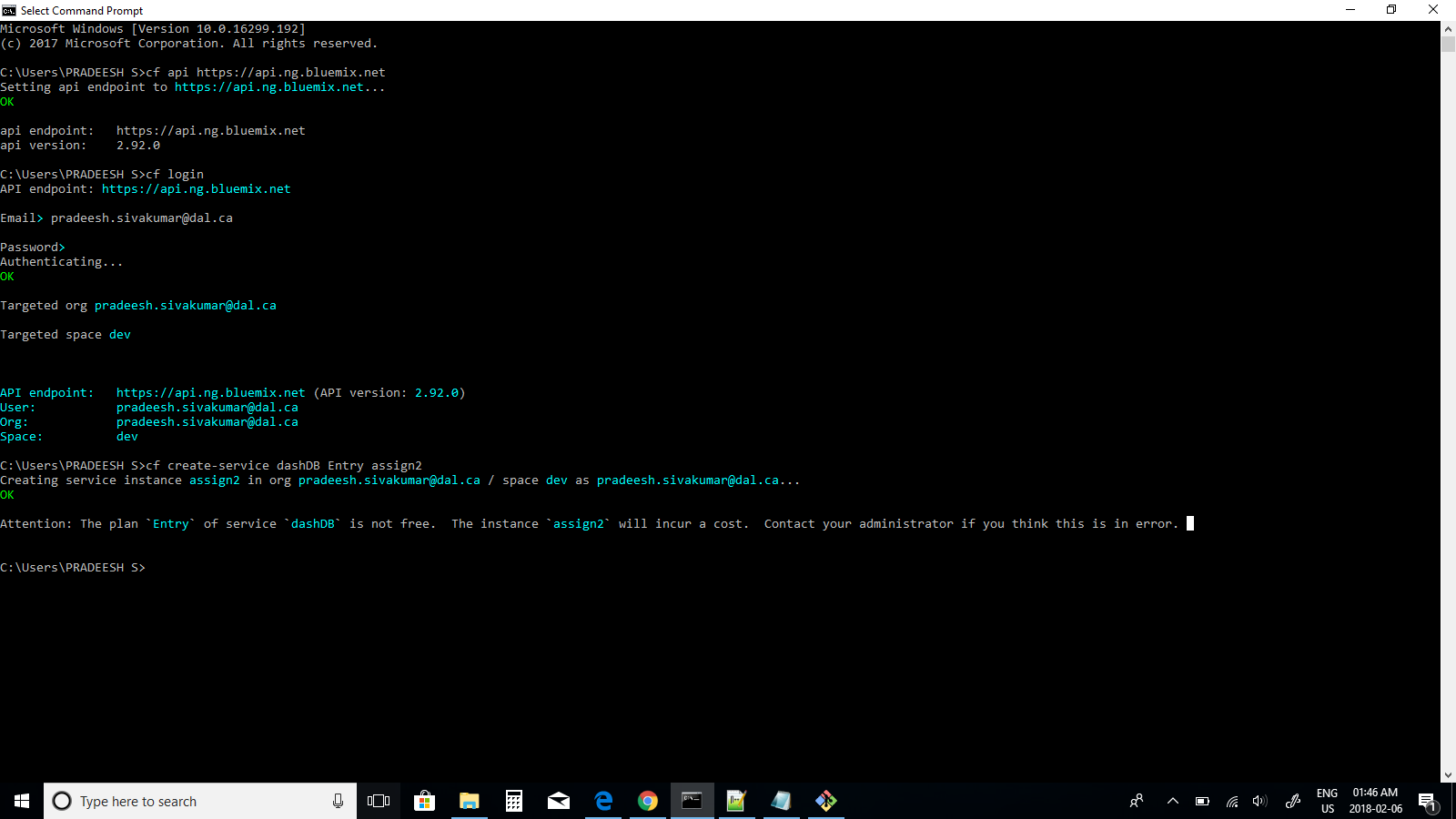
1.Load the dataset



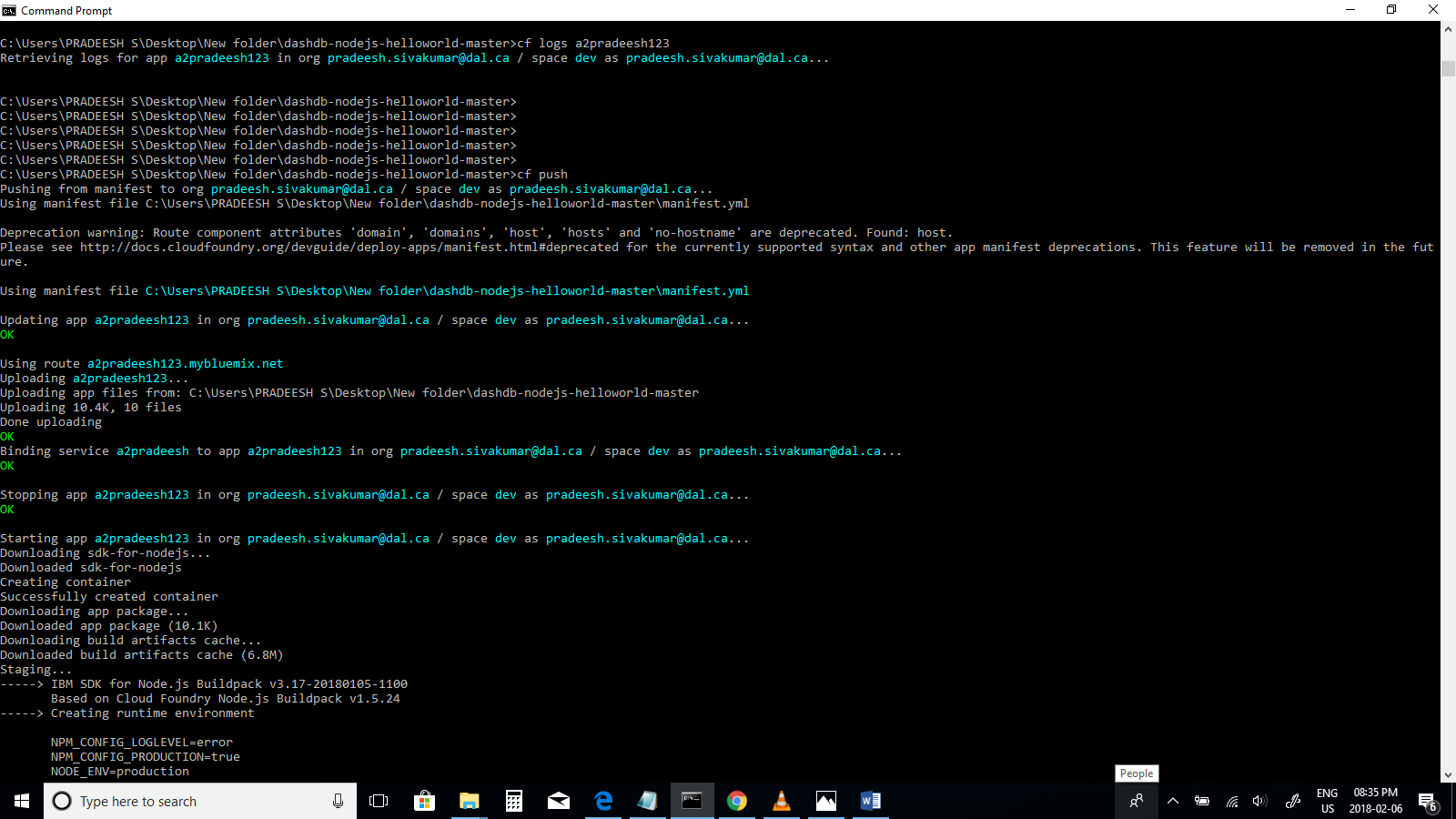
2. After loading the given dataset (Tourism)



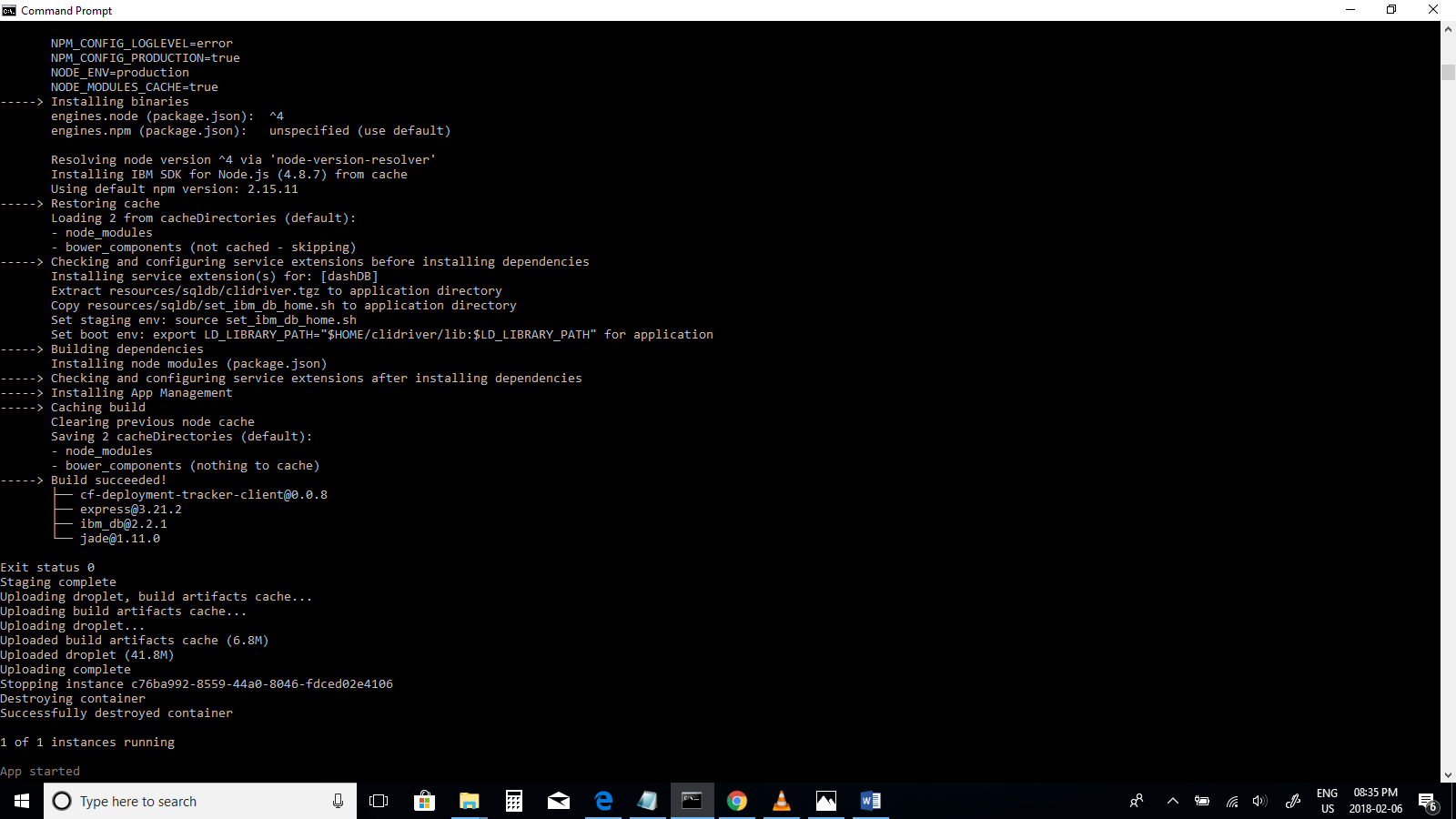
3. Connecting the Bluemix using command line tool to login



4. Create dash DB instance using in Bluemix and push the app in Bluemix



Loading…

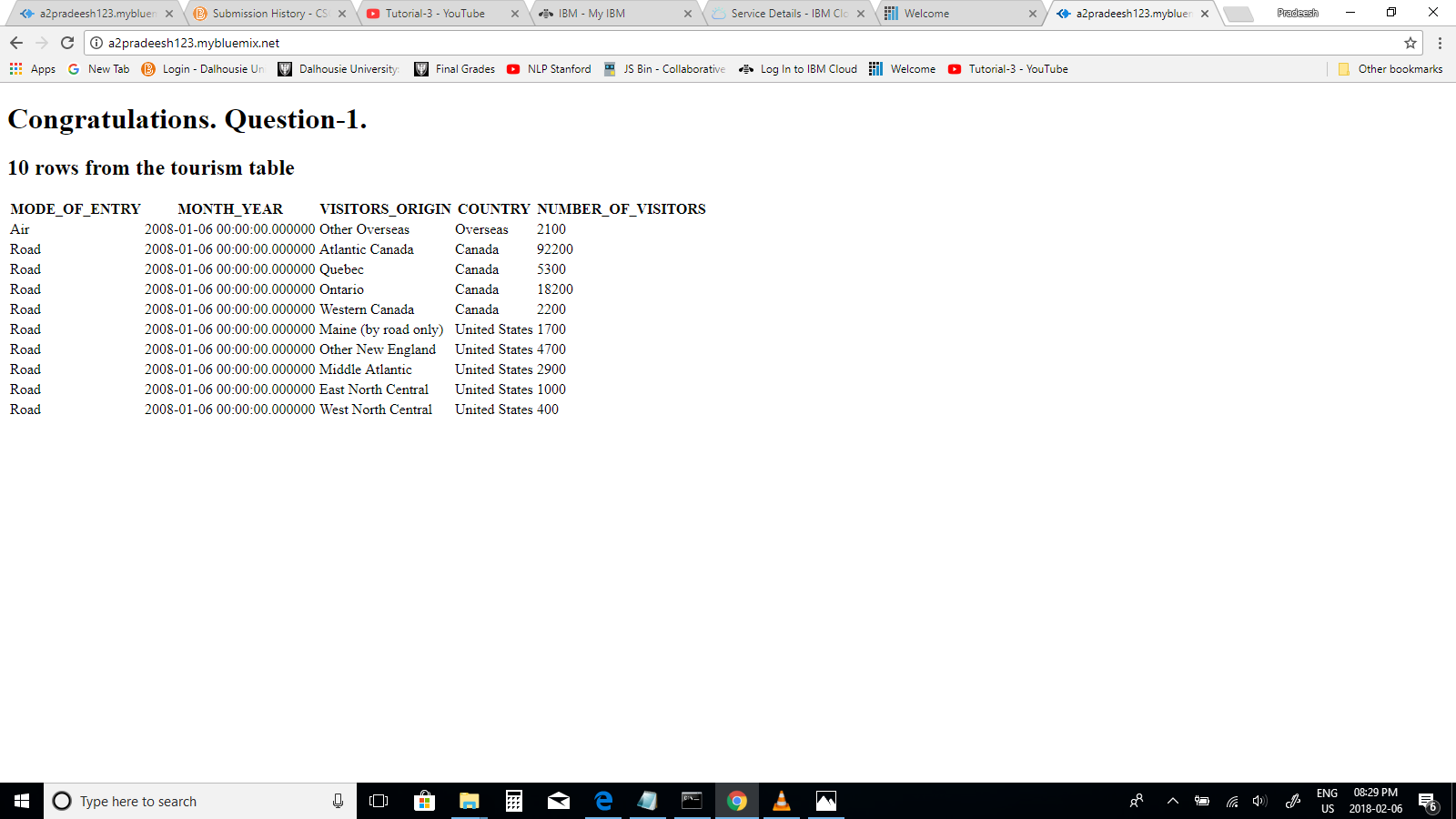


**3c) Application Queries**

**Question-1**

SELECT \* from tourism FETCH FIRST 10 ROWS ONLY

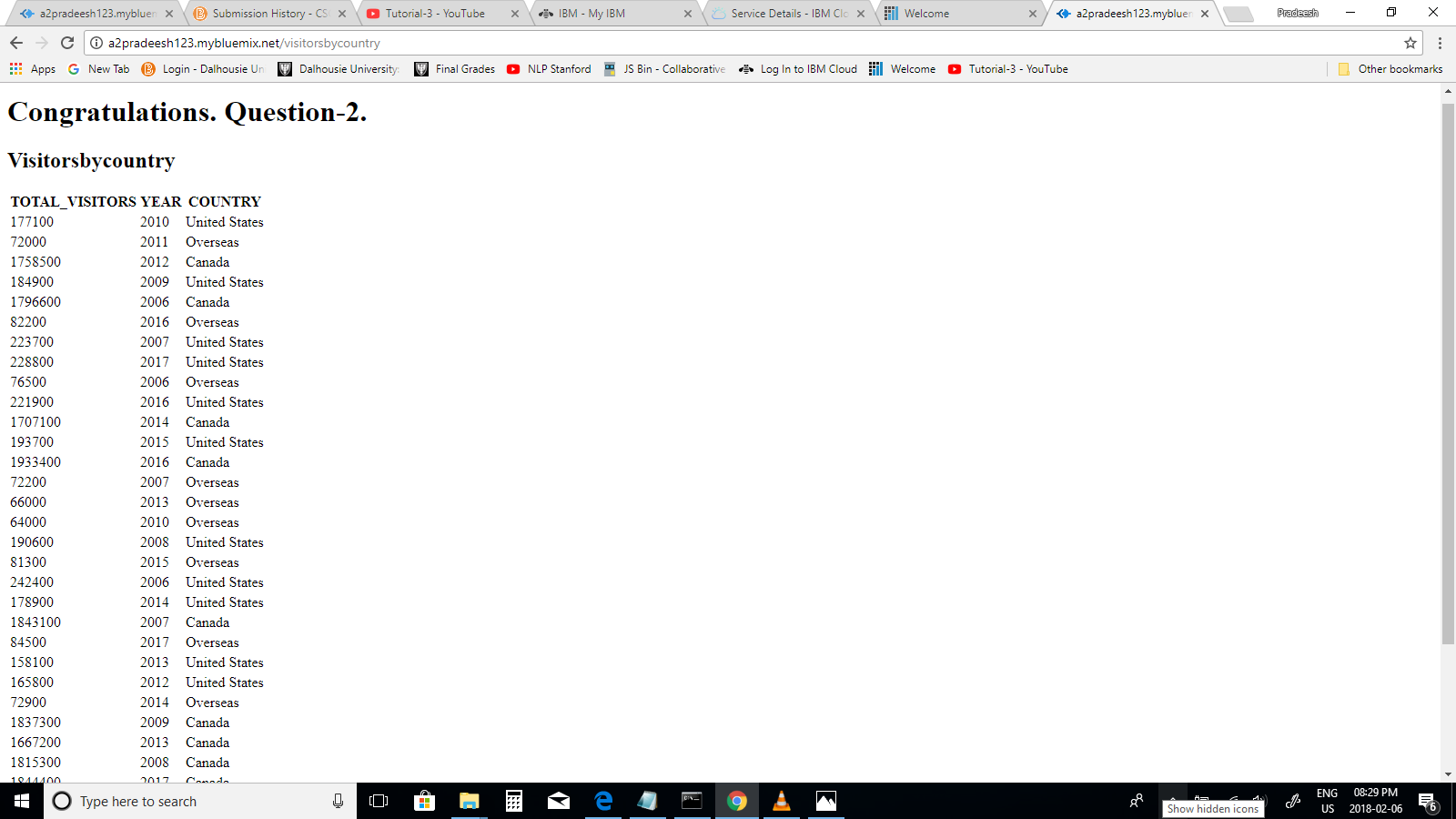
SELECT \* from tourism FETCH FIRST 10 ROWS ONLY displays first 10 rows present in tourism dataset.



**Question-2**

SELECT SUM(NUMBER\_OF\_VISITORS) as total\_visitors, YEAR(month\_year) as year, country FROM TOURISM GROUP BY COUNTRY, YEAR(MONTH\_YEAR)

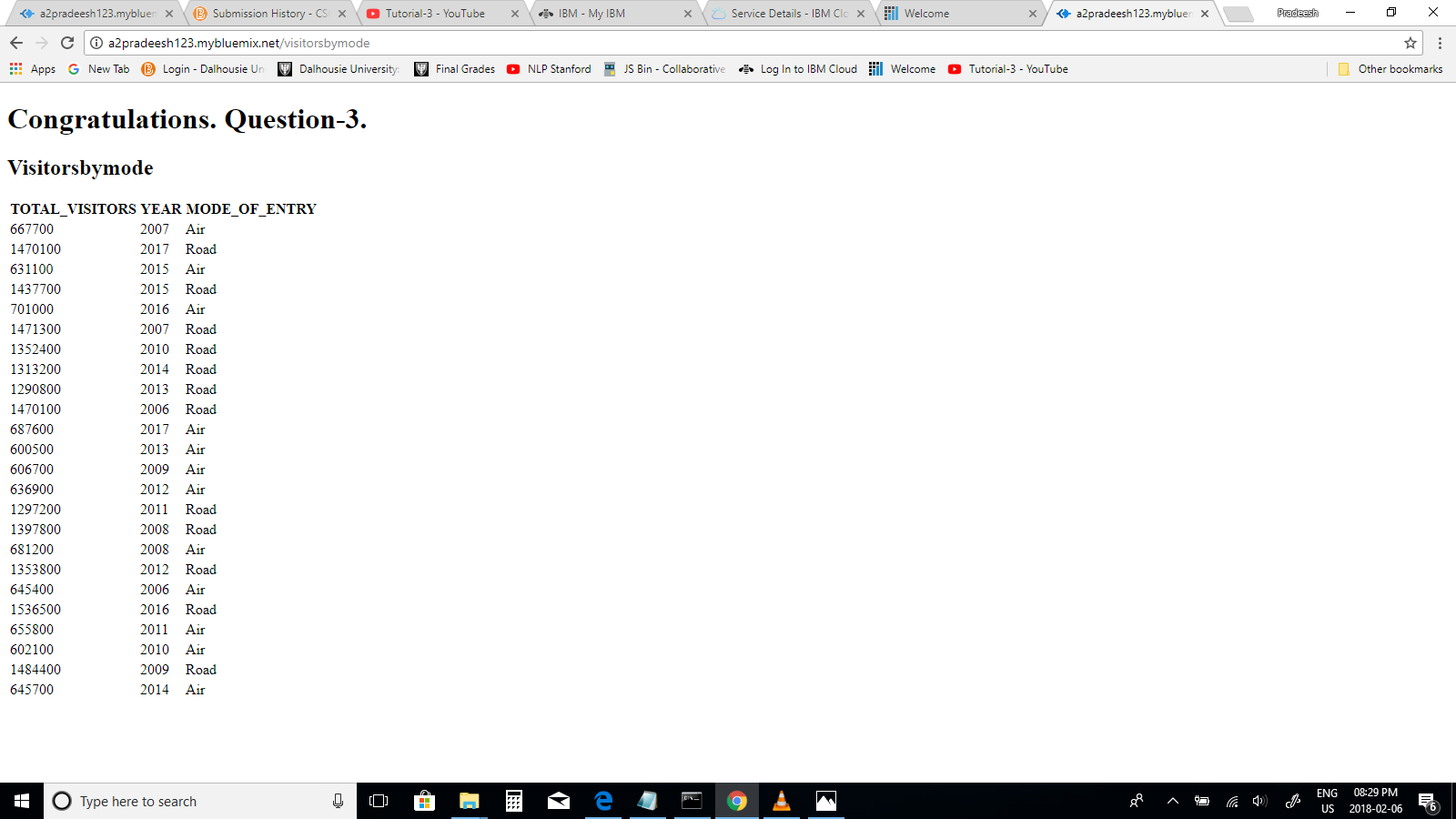
This query retrieves the total number of visitors information which is displayed as total visitors, per year displayed as year, by country from tourism dataset.



**Question-3**

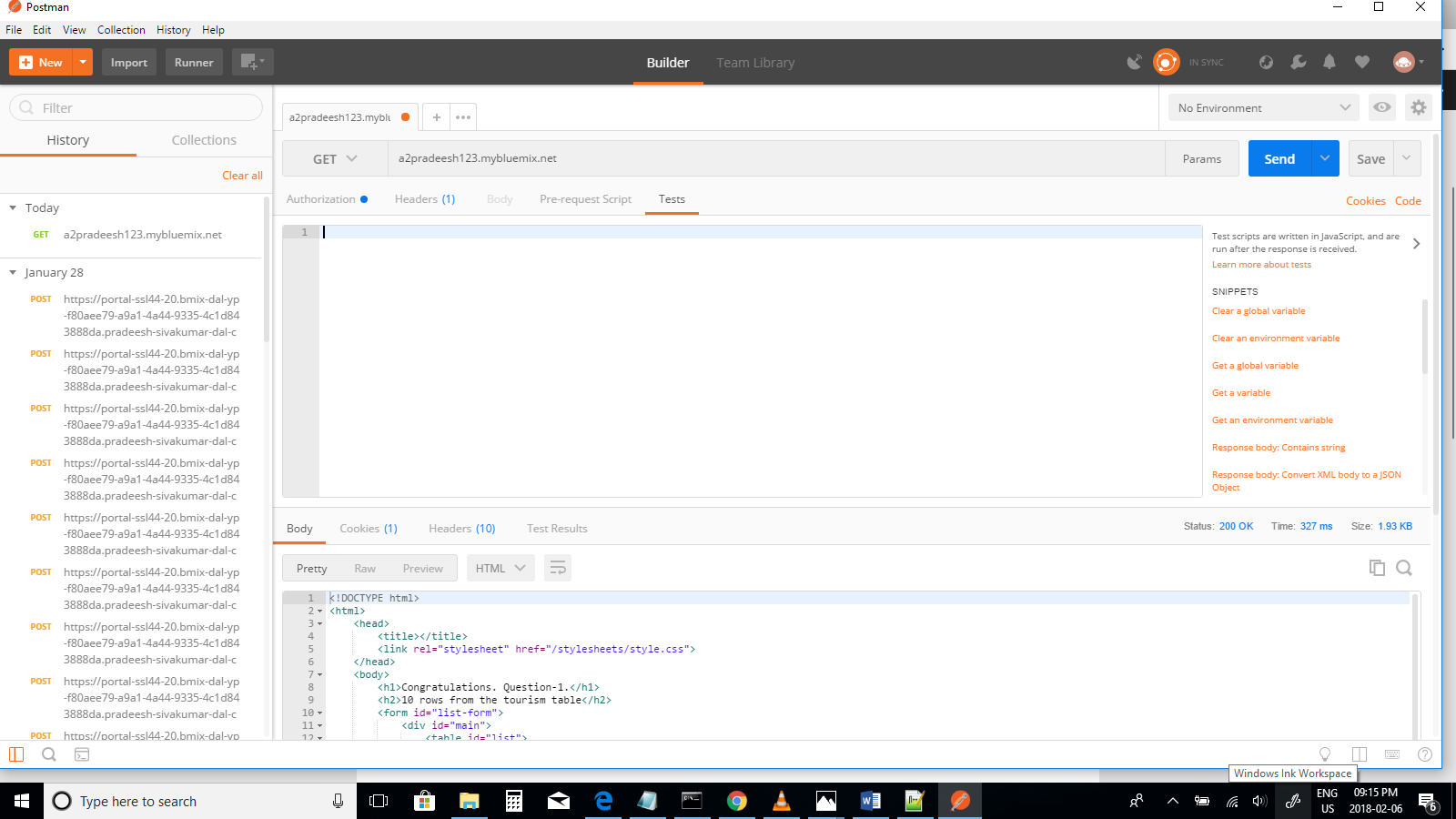
SELECT SUM (NUMBER\_OF\_VISITORS) as total\_visitors, YEAR(month\_year) as year , MODE\_OF\_ENTRY FROM TOURISM GROUP BY MODE\_OF\_ENTRY, YEAR(MONTH\_YEAR).

This retrieves the total number of visitors information which is displayed as total visitors, per year displayed as year, by mode of entry information from tourism dataset.



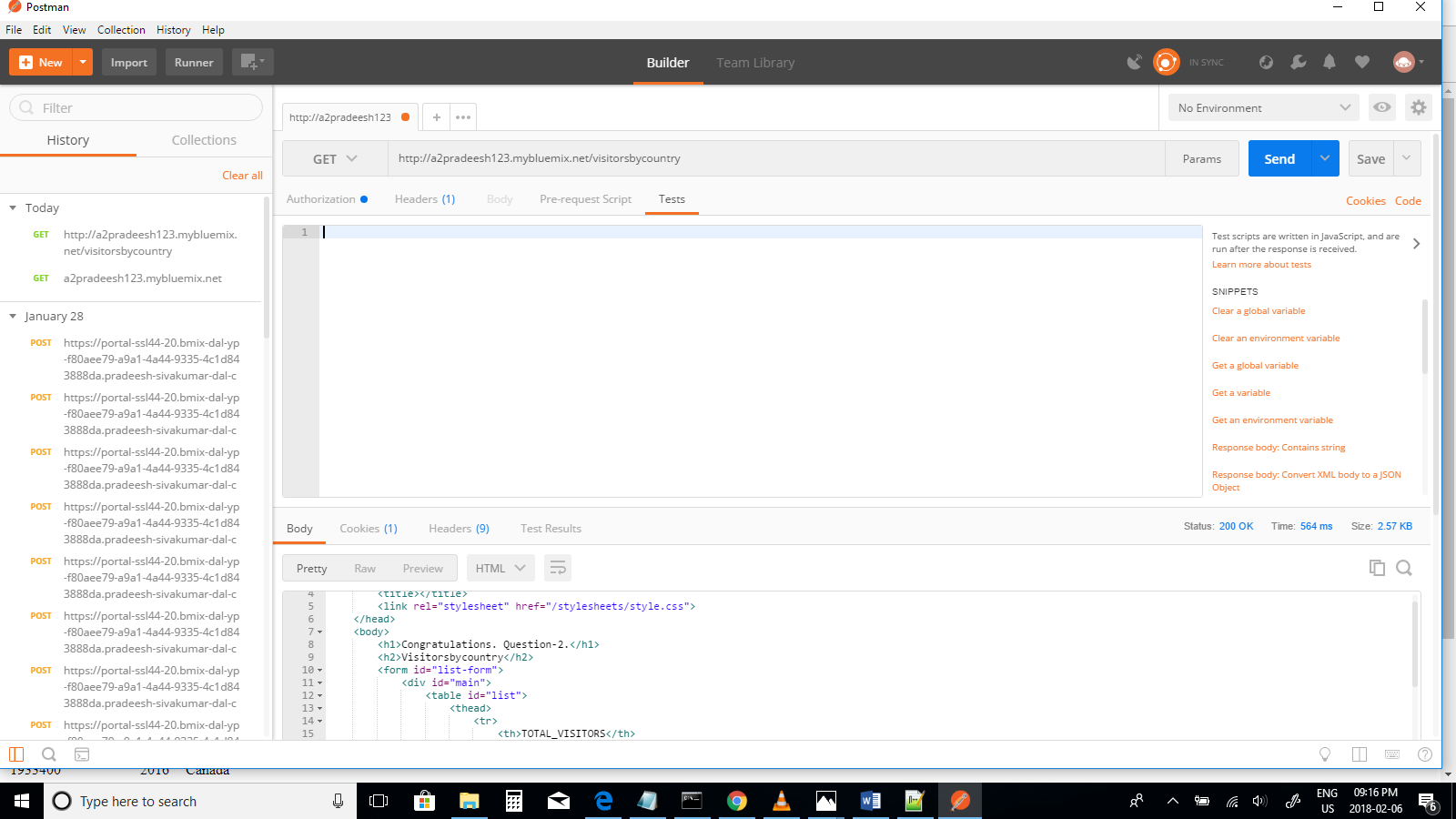
**4. Testing Web service through Postman Software**

**Q-1 URL : a2pradeesh123.mybluemix.net**



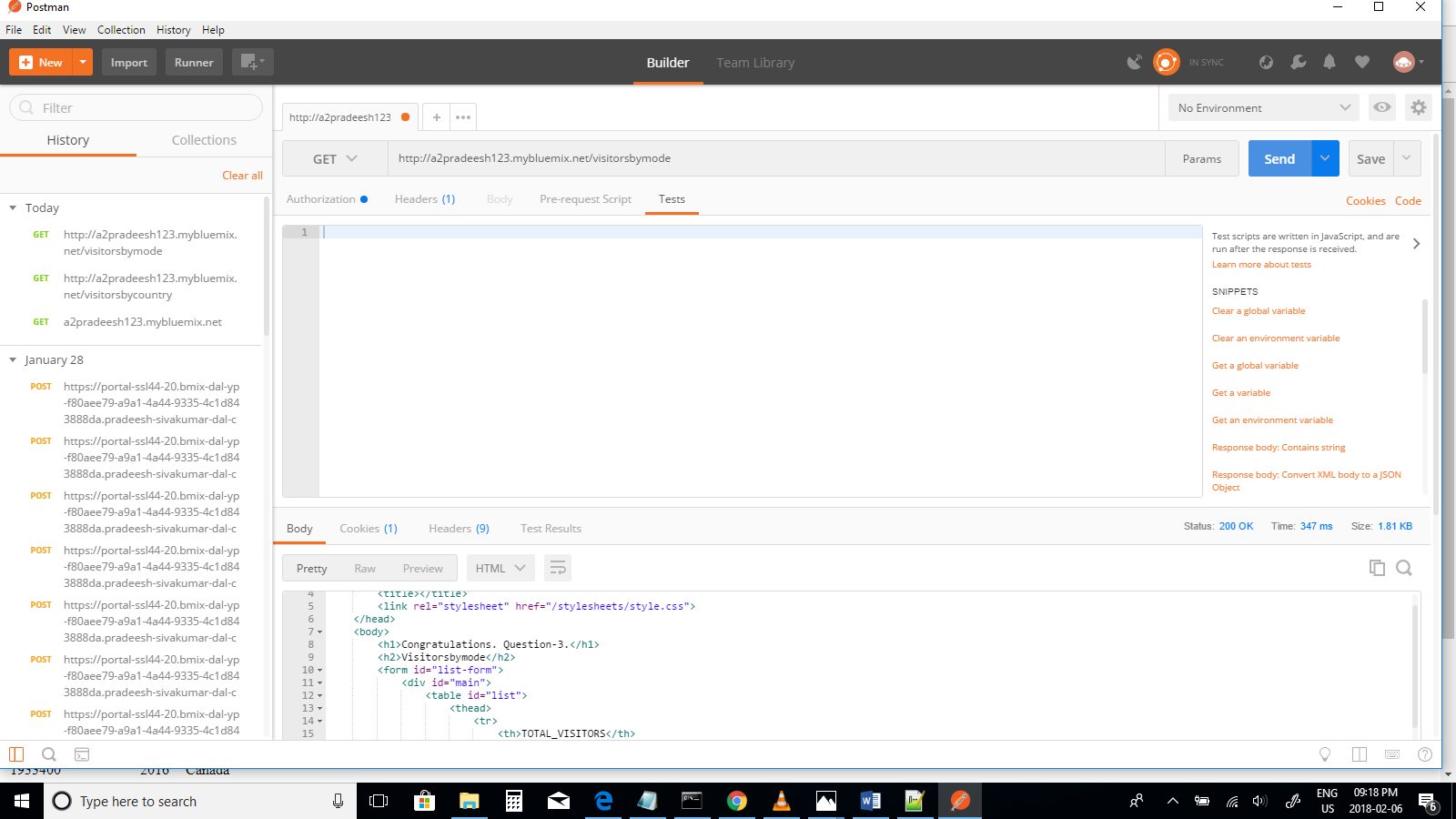
Response Time for the web URL – 327 ms

**Q-2 URL : a2pradeesh123.mybluemix.net/visitorsbycountry**



Response Time for the web URL – 564 ms

**Q-3 URL : a2pradeesh123.mybluemix.net/visitorsbymode**



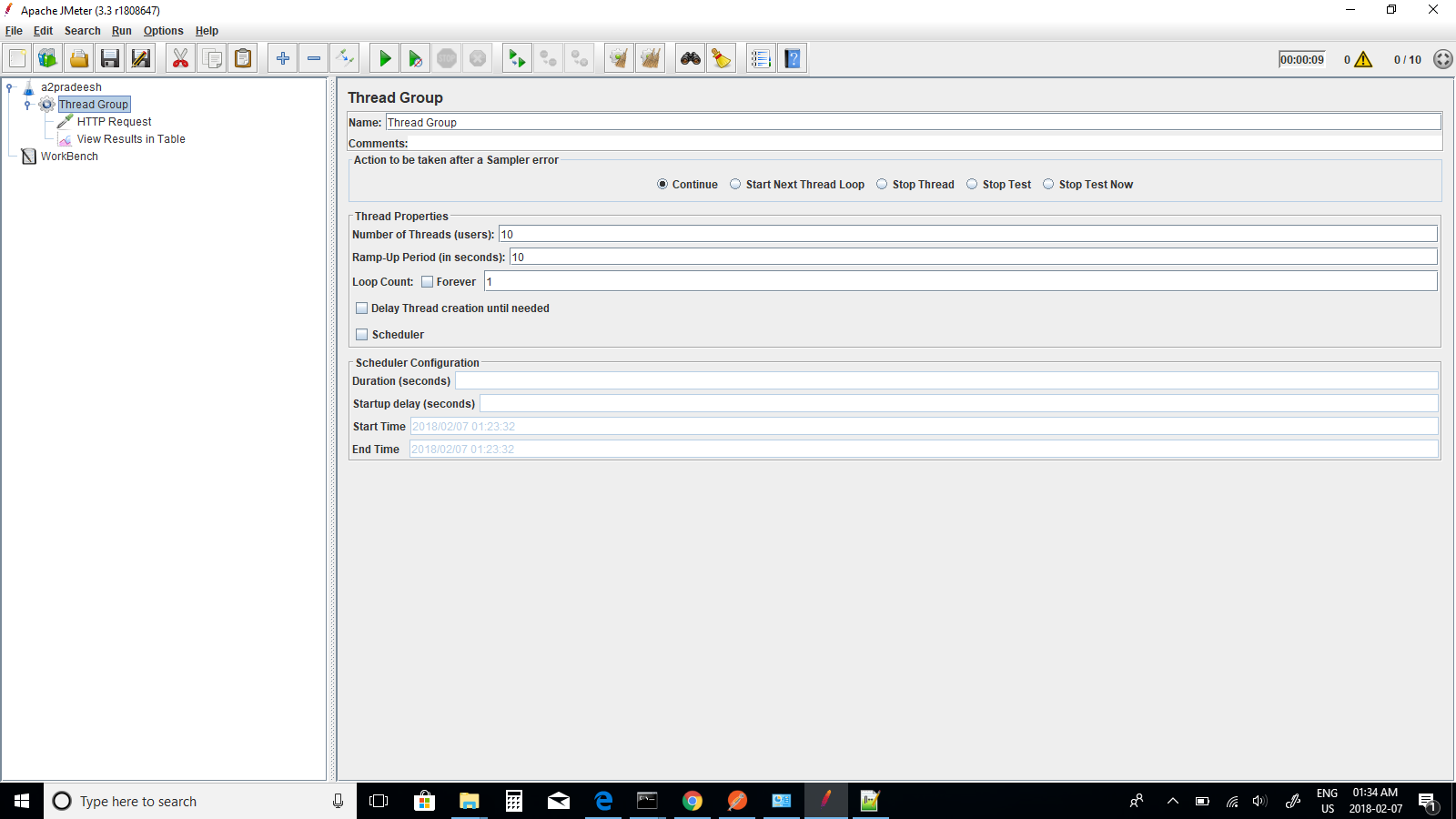
Response Time for the web URL – 347 ms

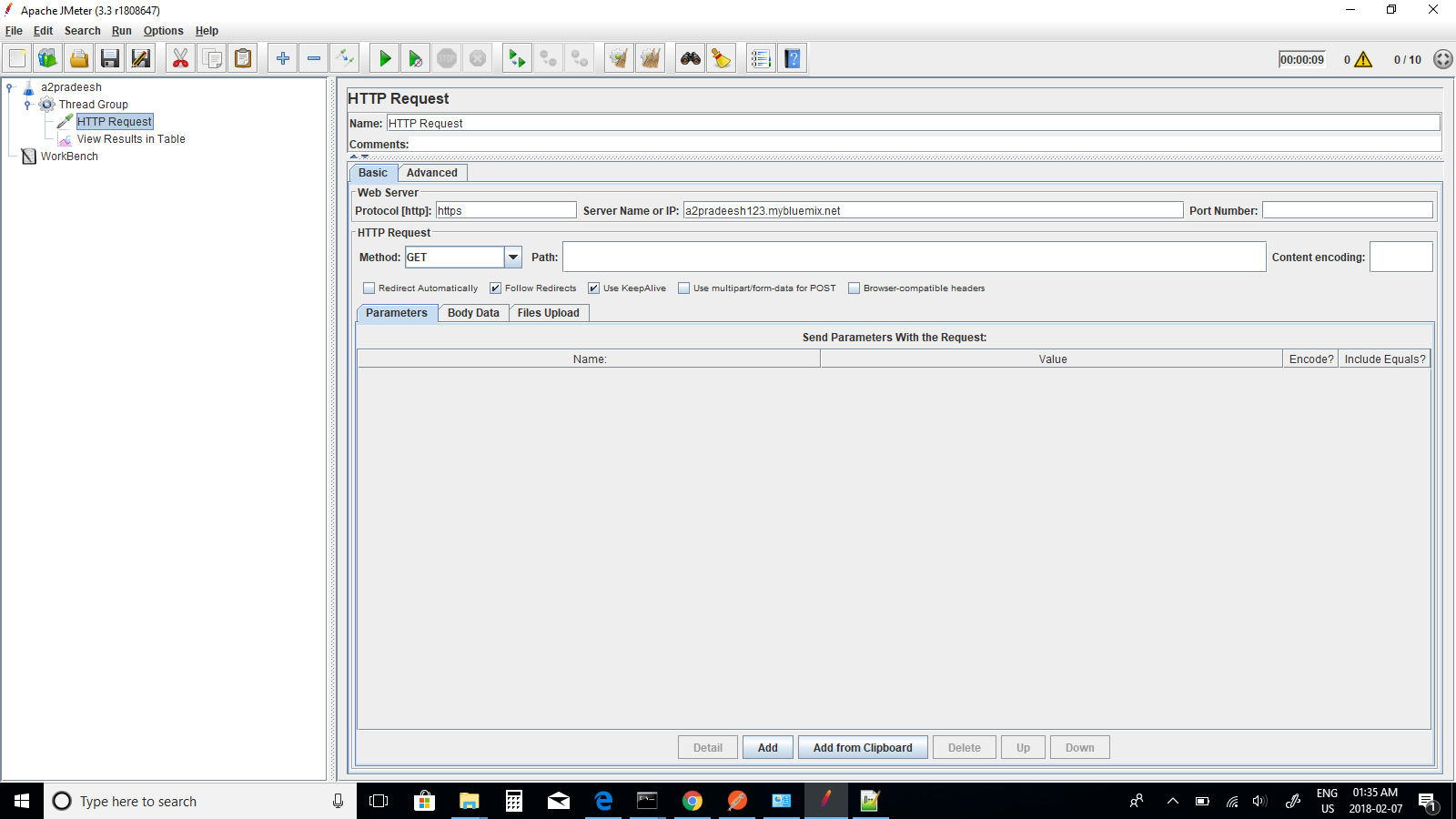
The above results shows that the web service for three URLs using postman software. It has been observed that for all URL the response time is different. The time taken for the first web URL is 327 ms, The time taken for second web URL is 564 ms and the time taken for third web URL is 564ms. This shows the response time for each URL constantly changes.

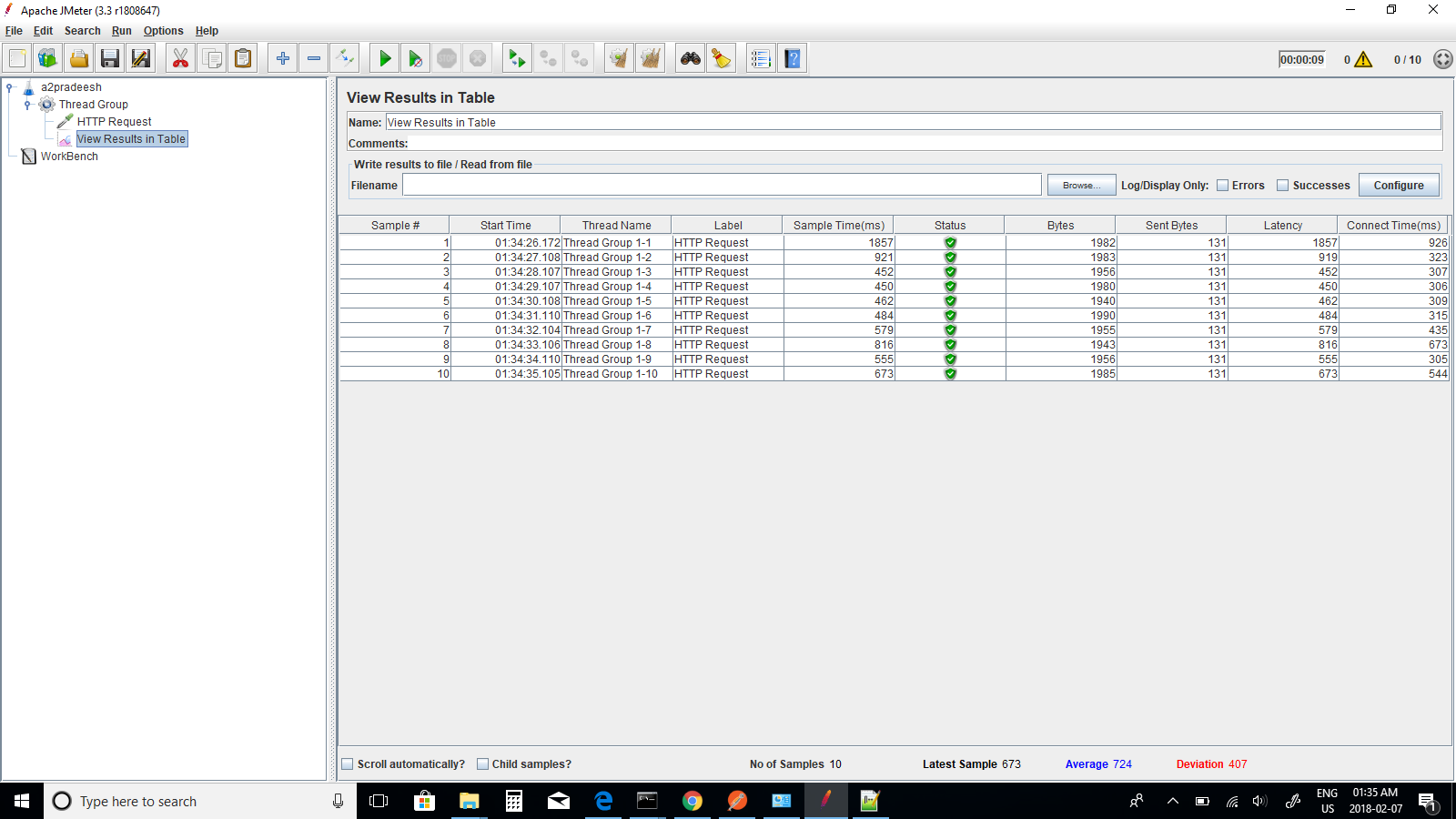
**5. UI Application through JMeter**

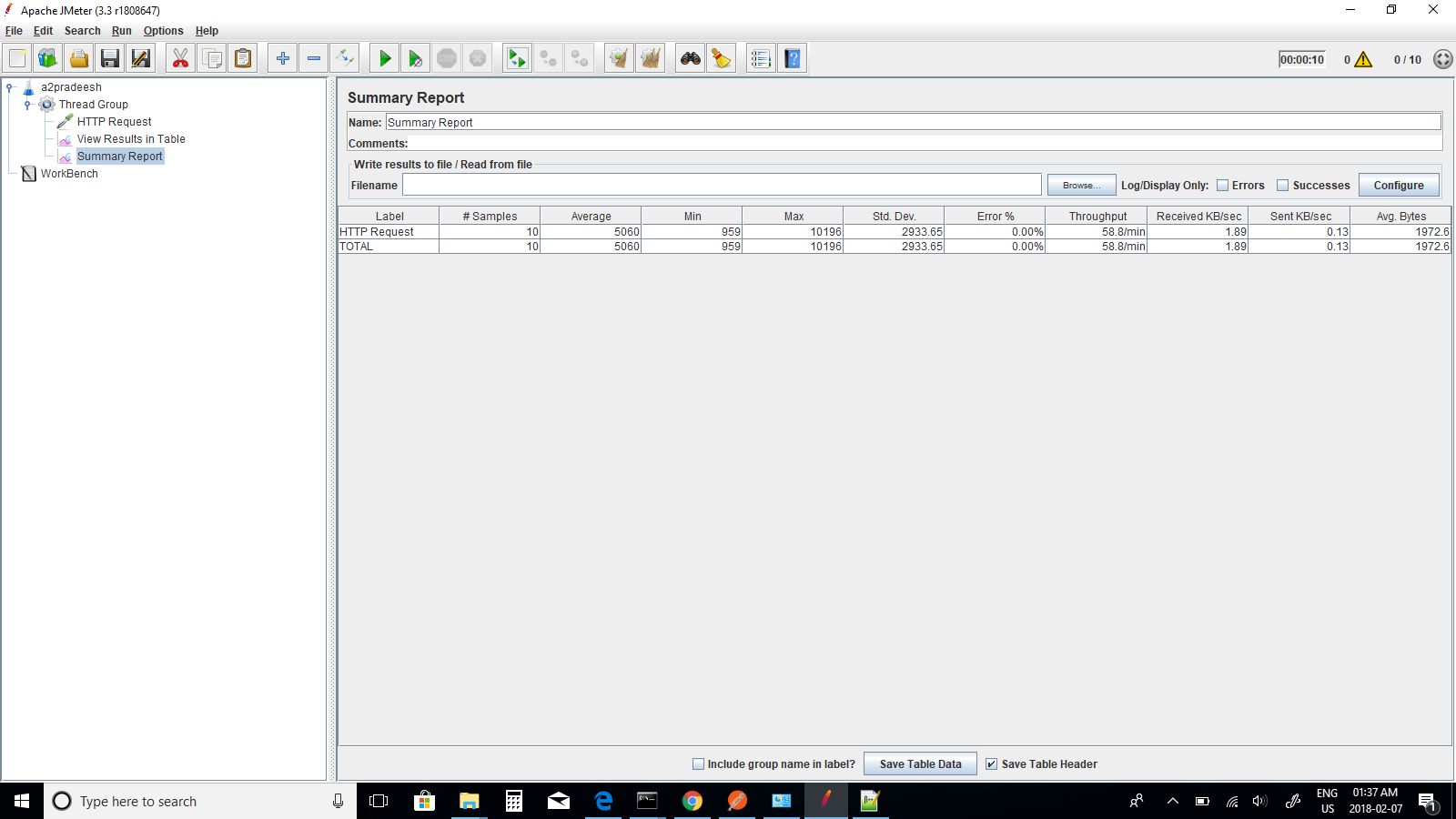
**i) URL : a2pradeesh123.mybluemix.net**

**Number of Threads-10**



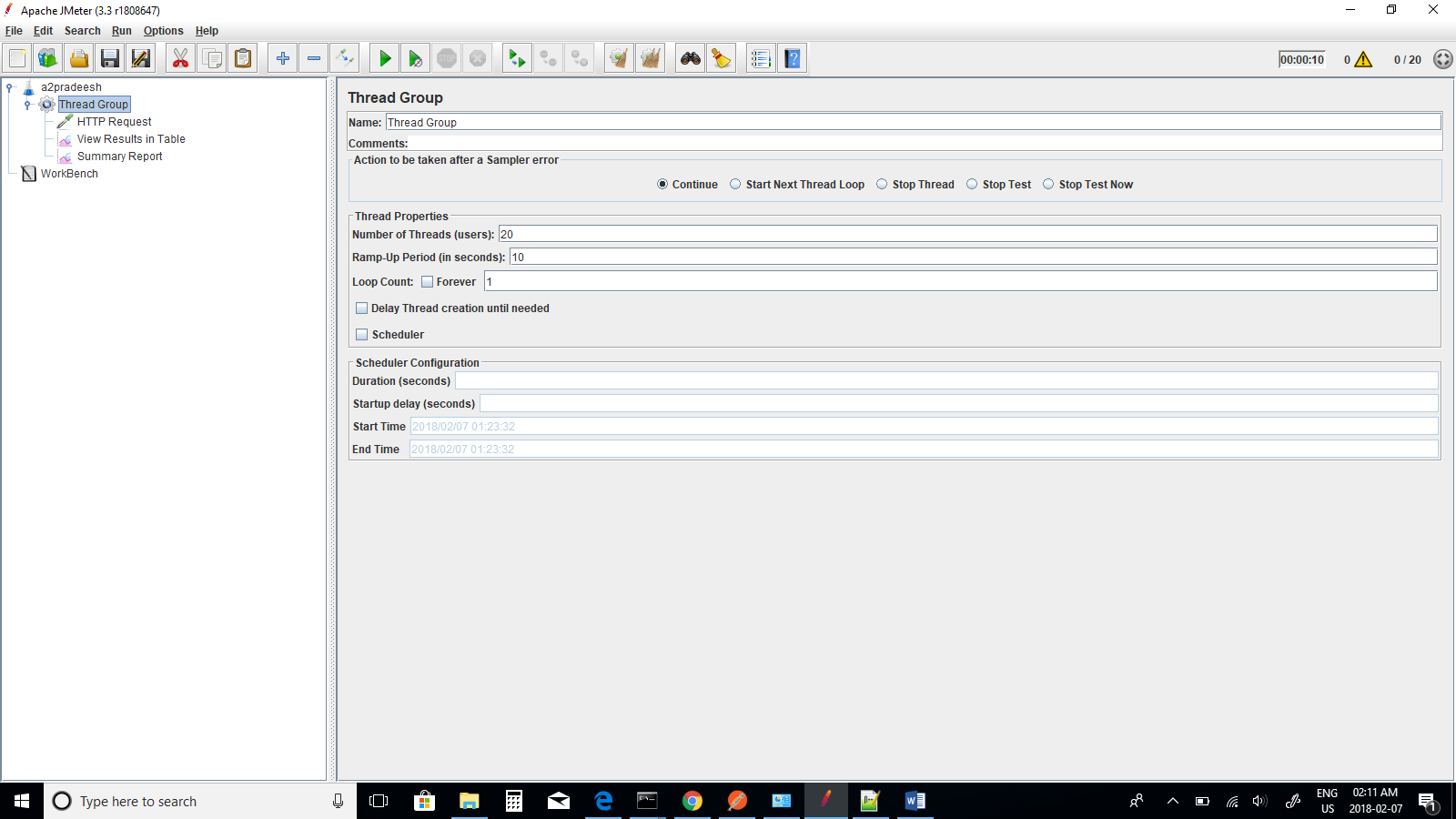


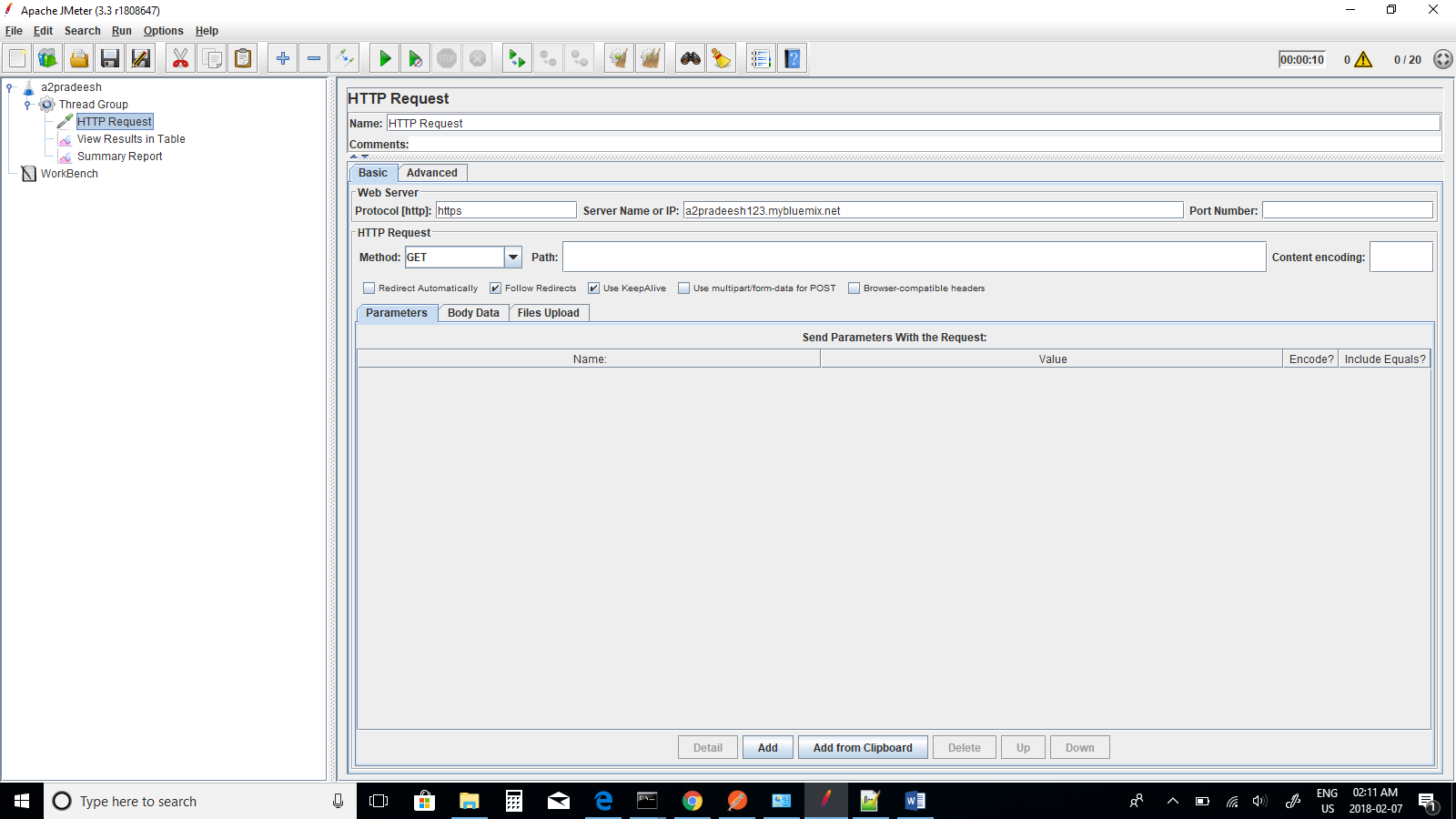


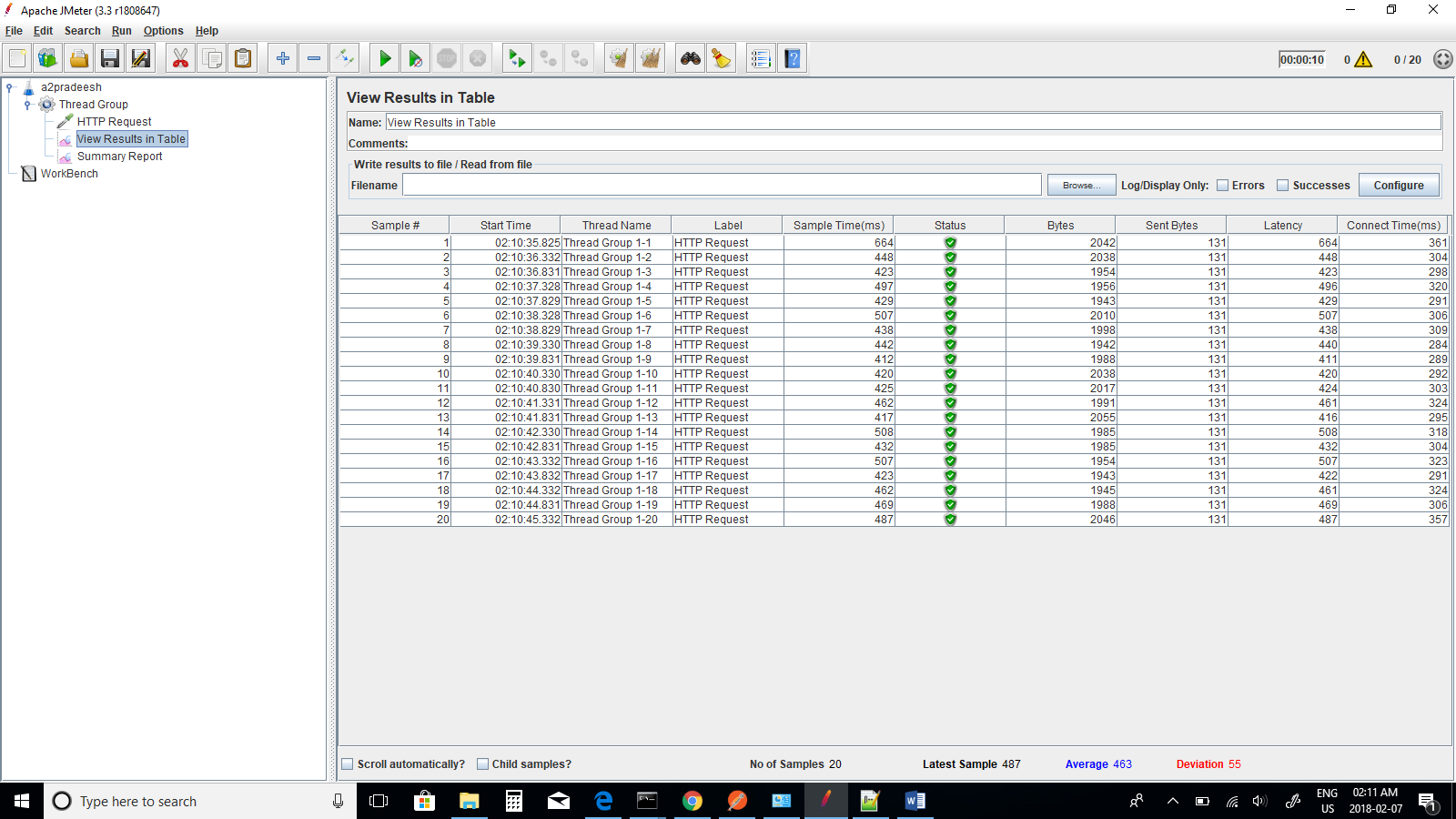


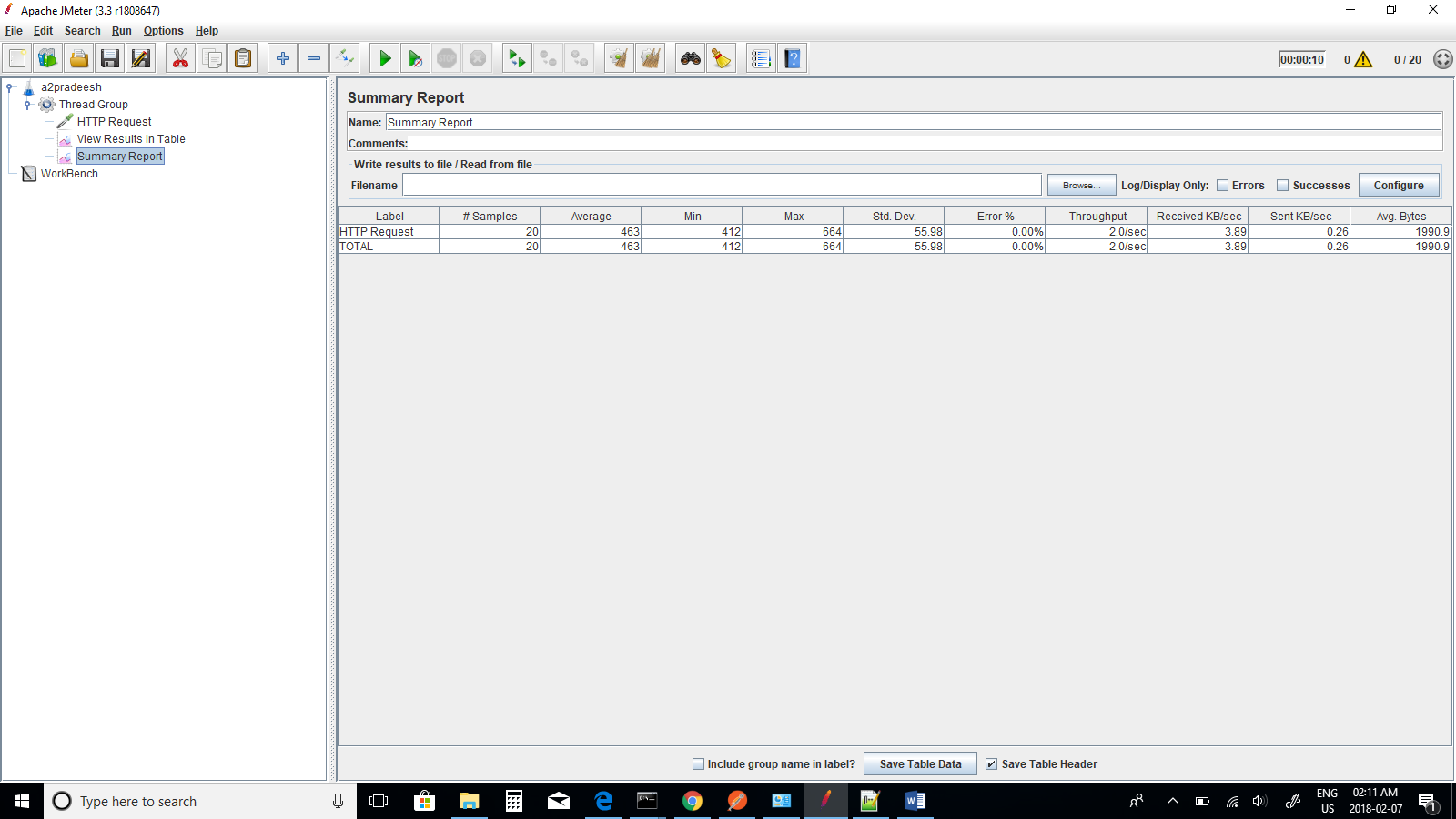
Average Response Time for 10 Threads - 5060ms

**Changing number of threads – 20**



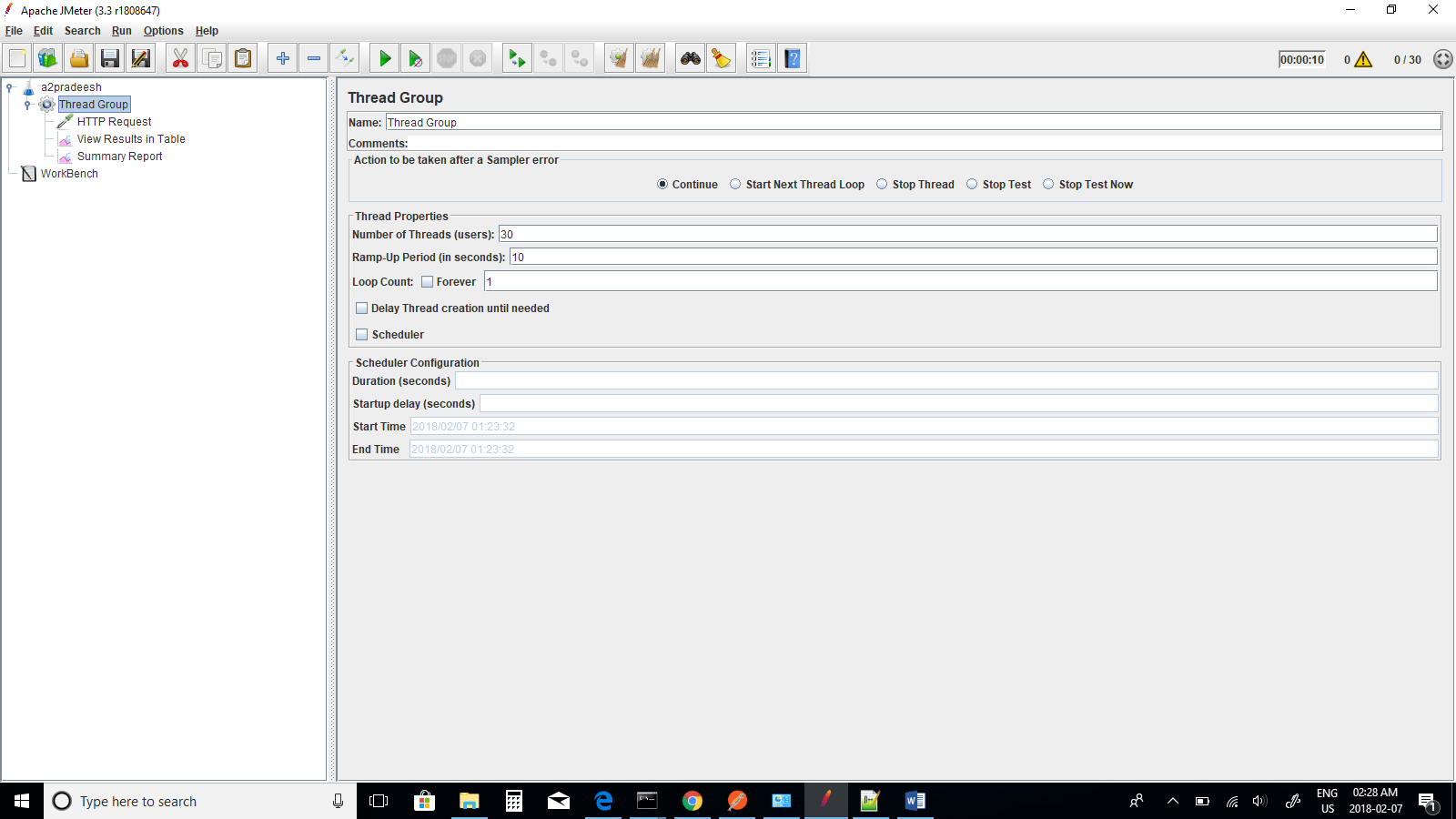


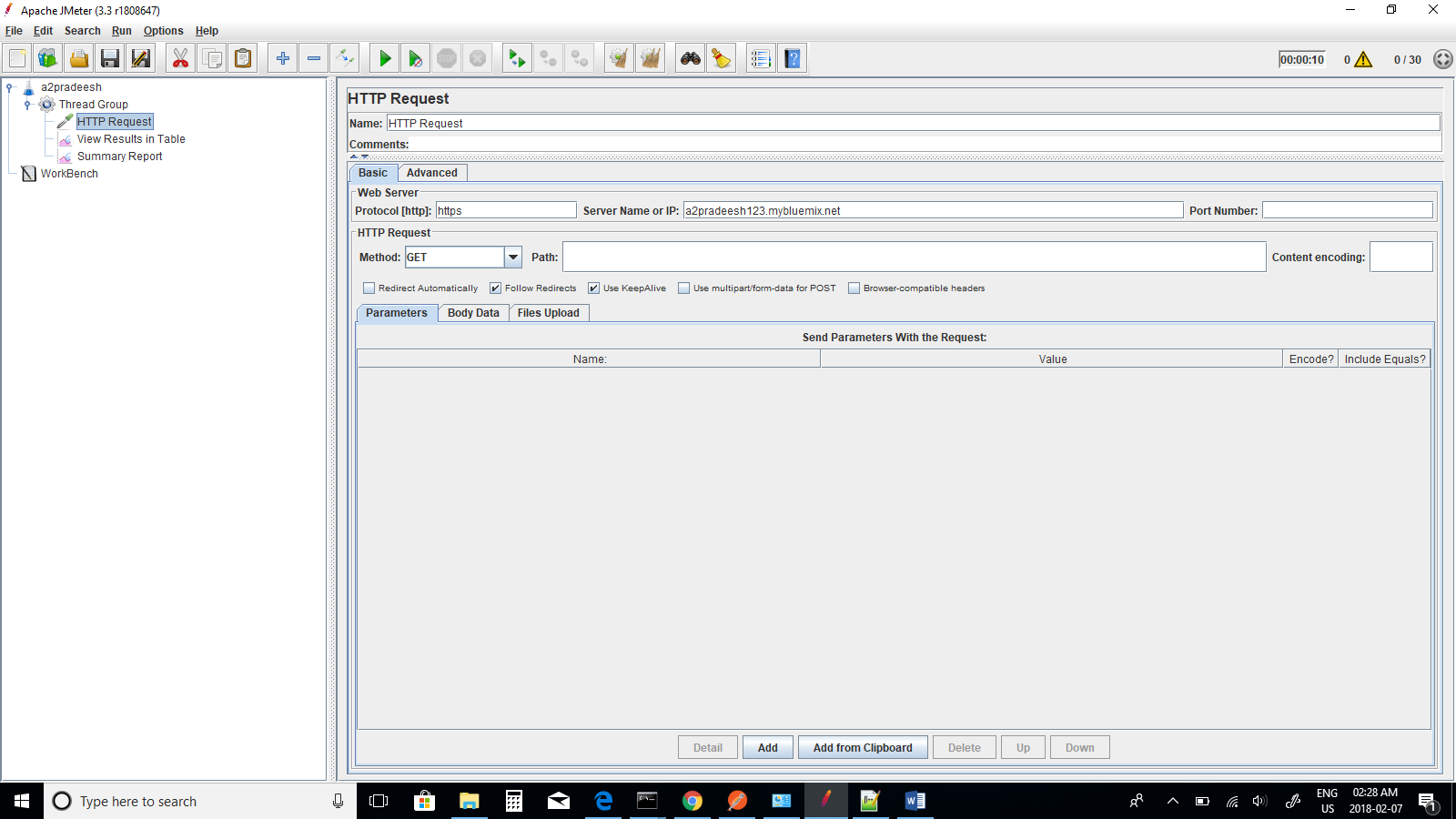


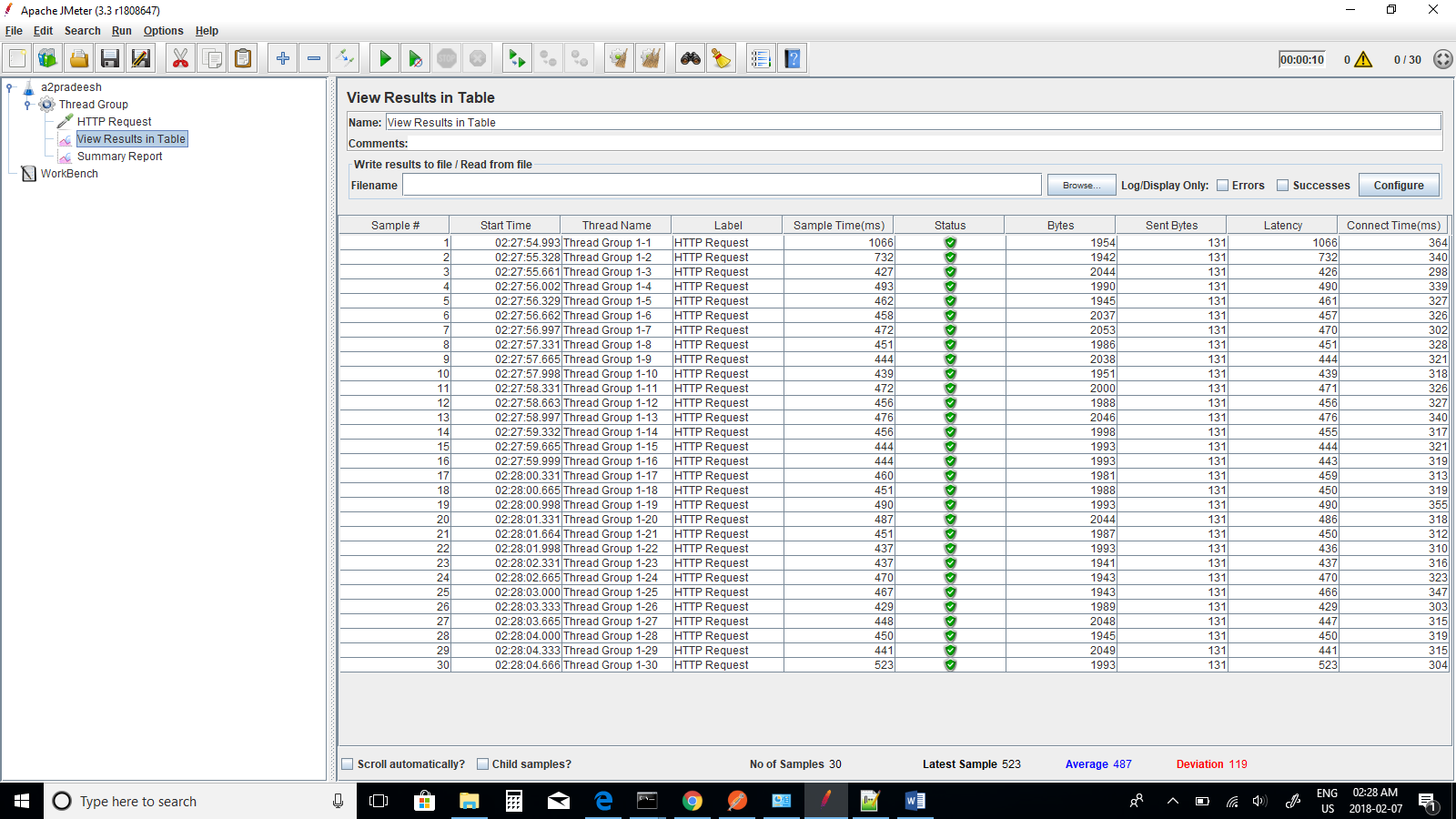


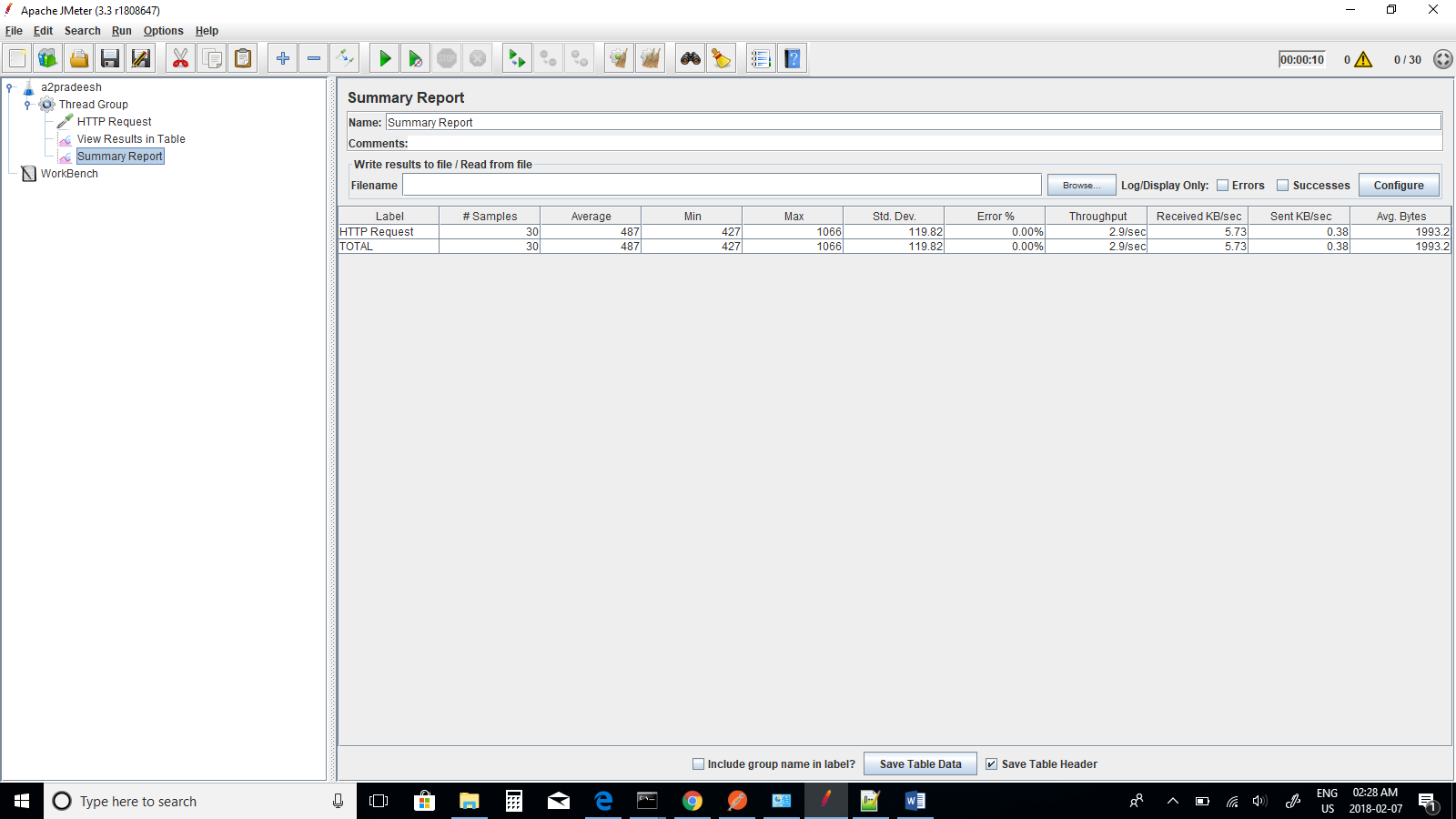
Average Response Time for 20 Threads - 463ms

**Changing number of threads – 30**



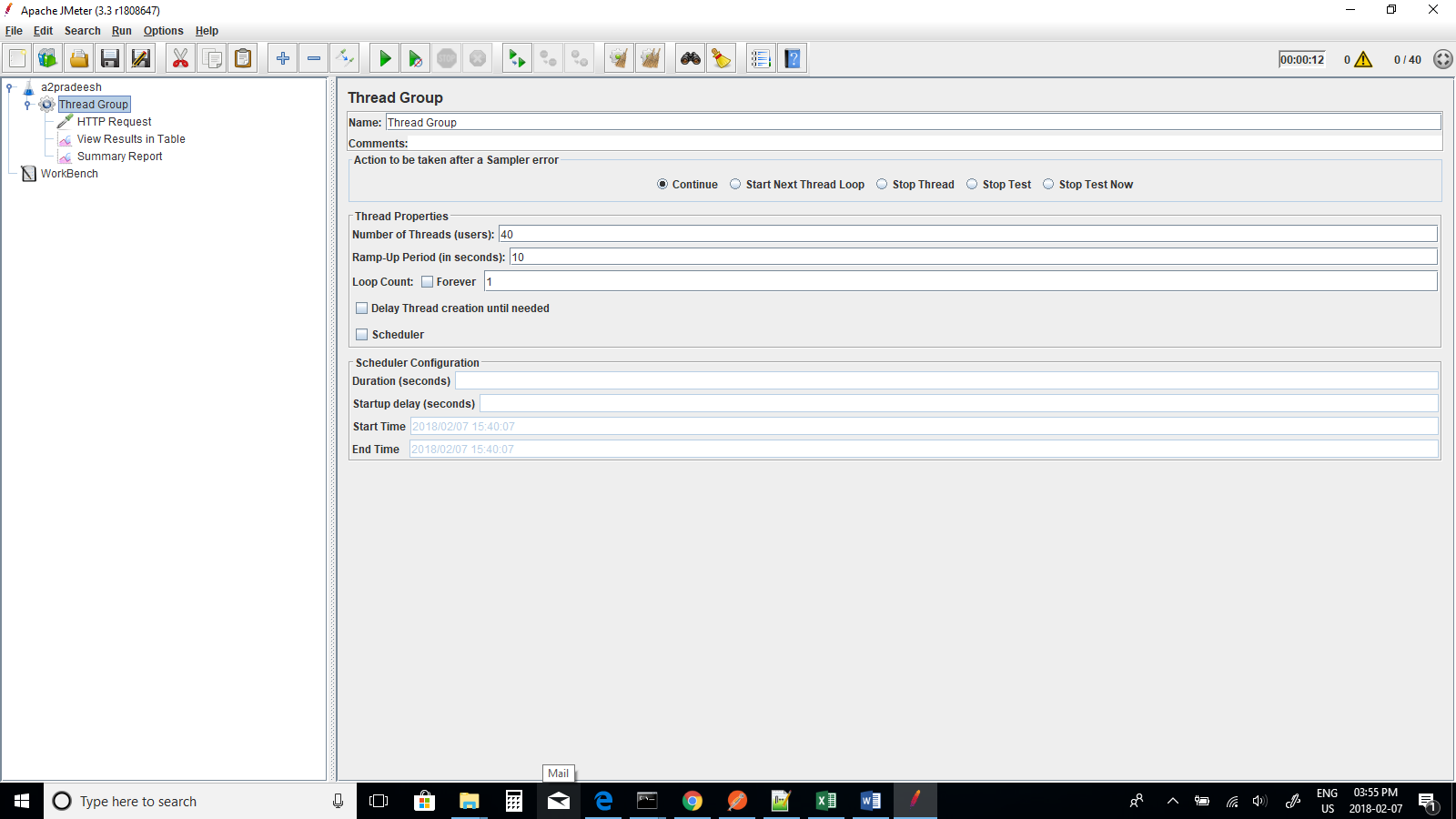


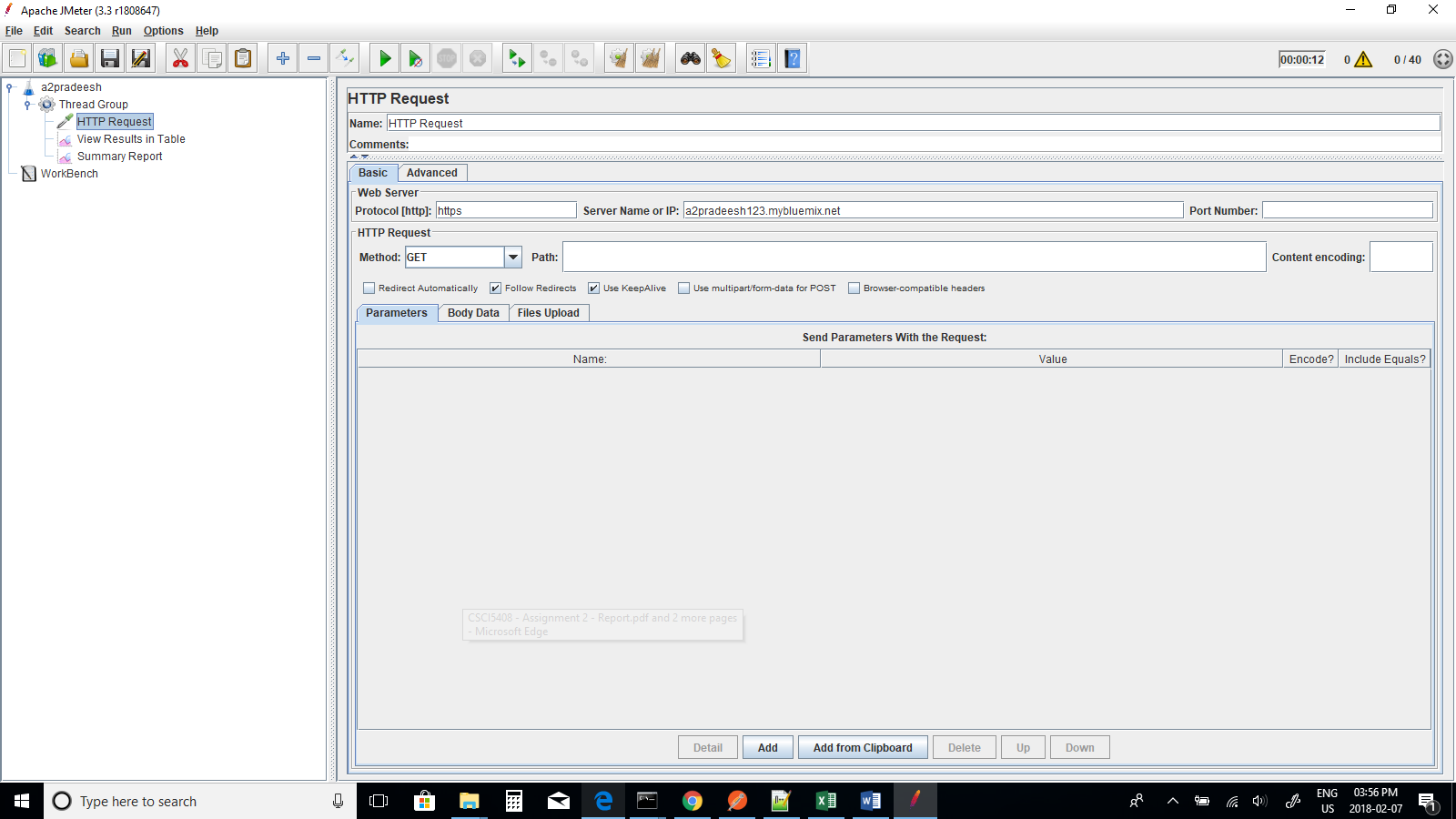


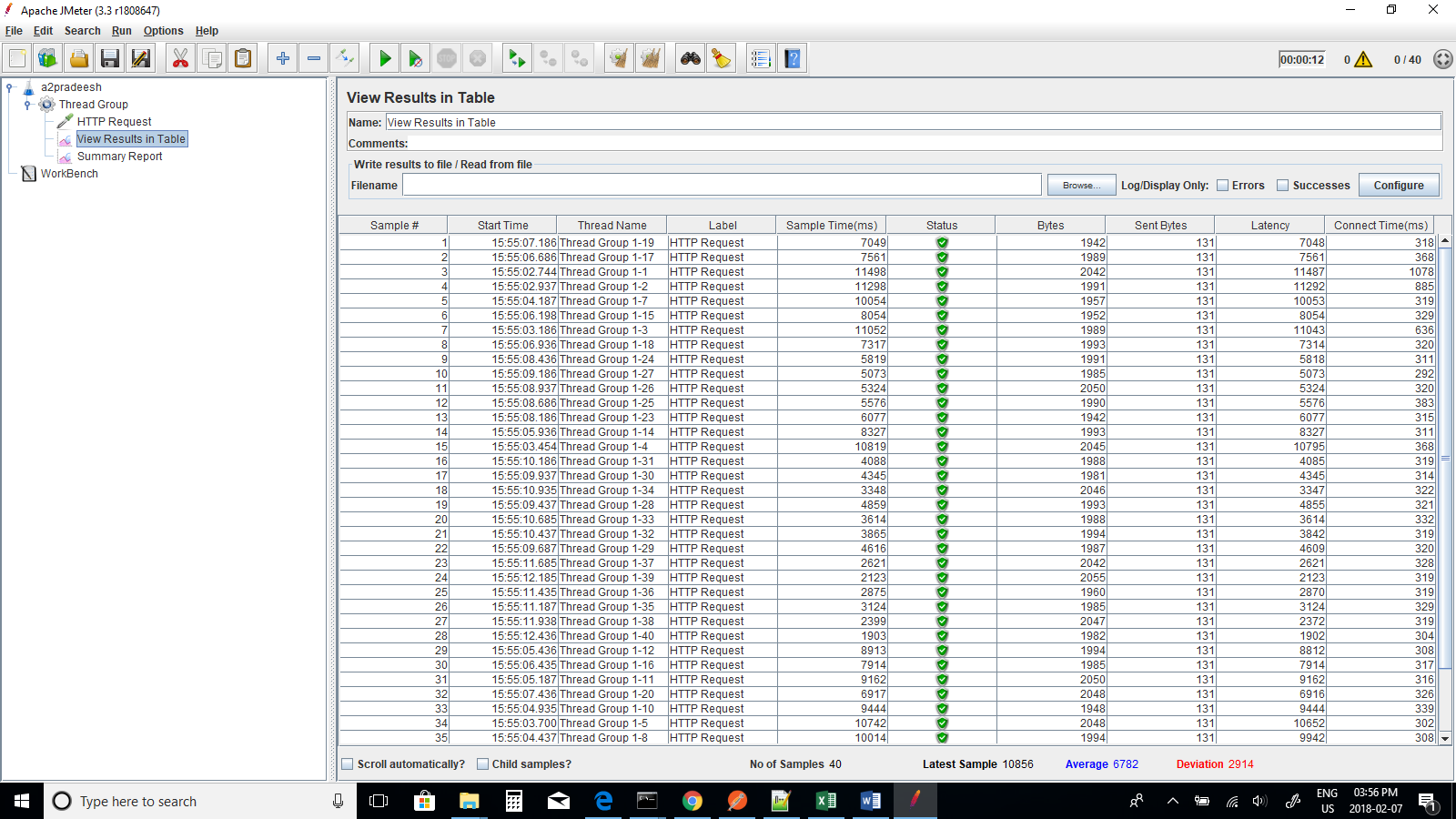


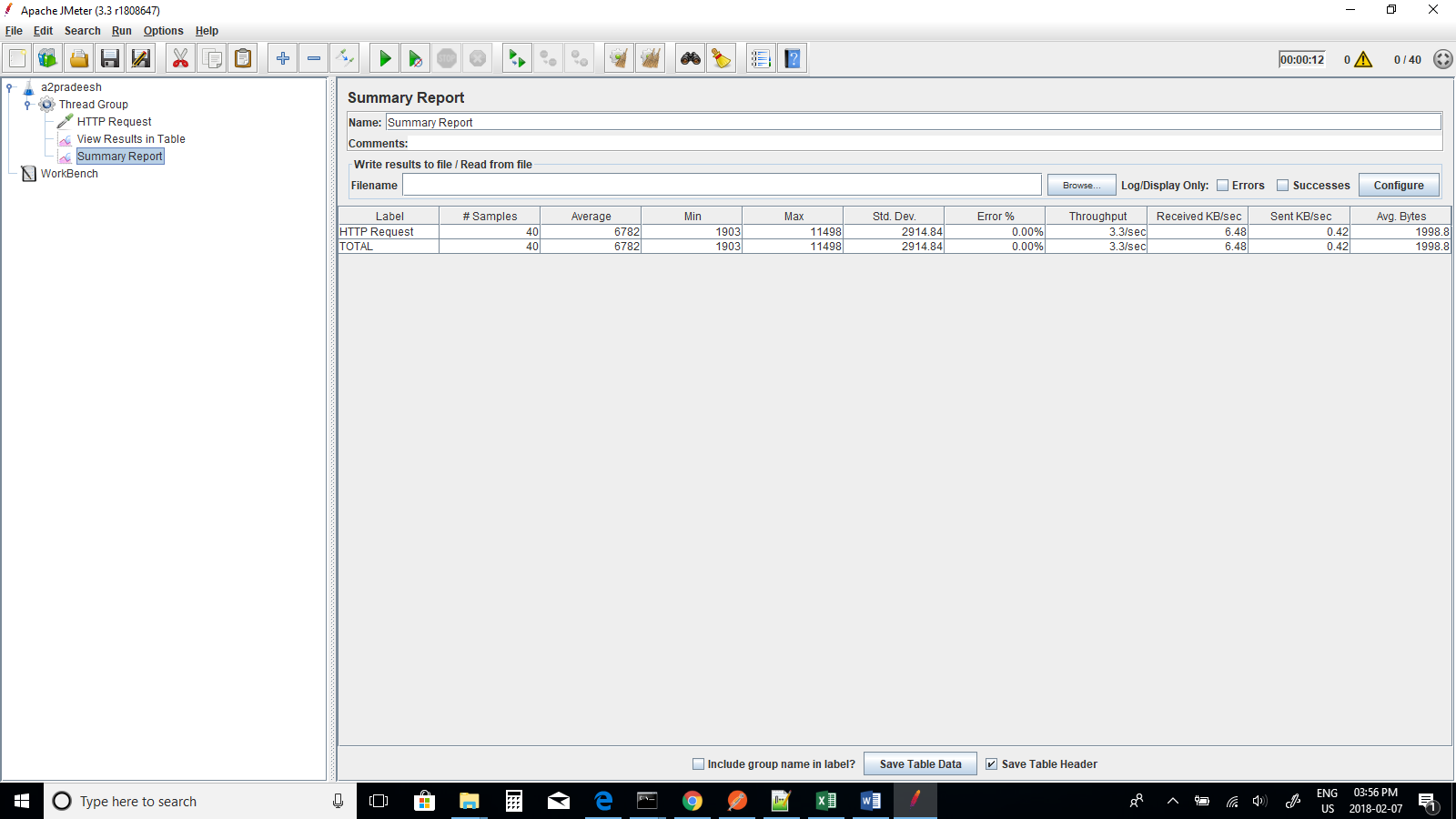
Average Response Time for 30 Threads - 487ms

**Changing number of threads – 40**



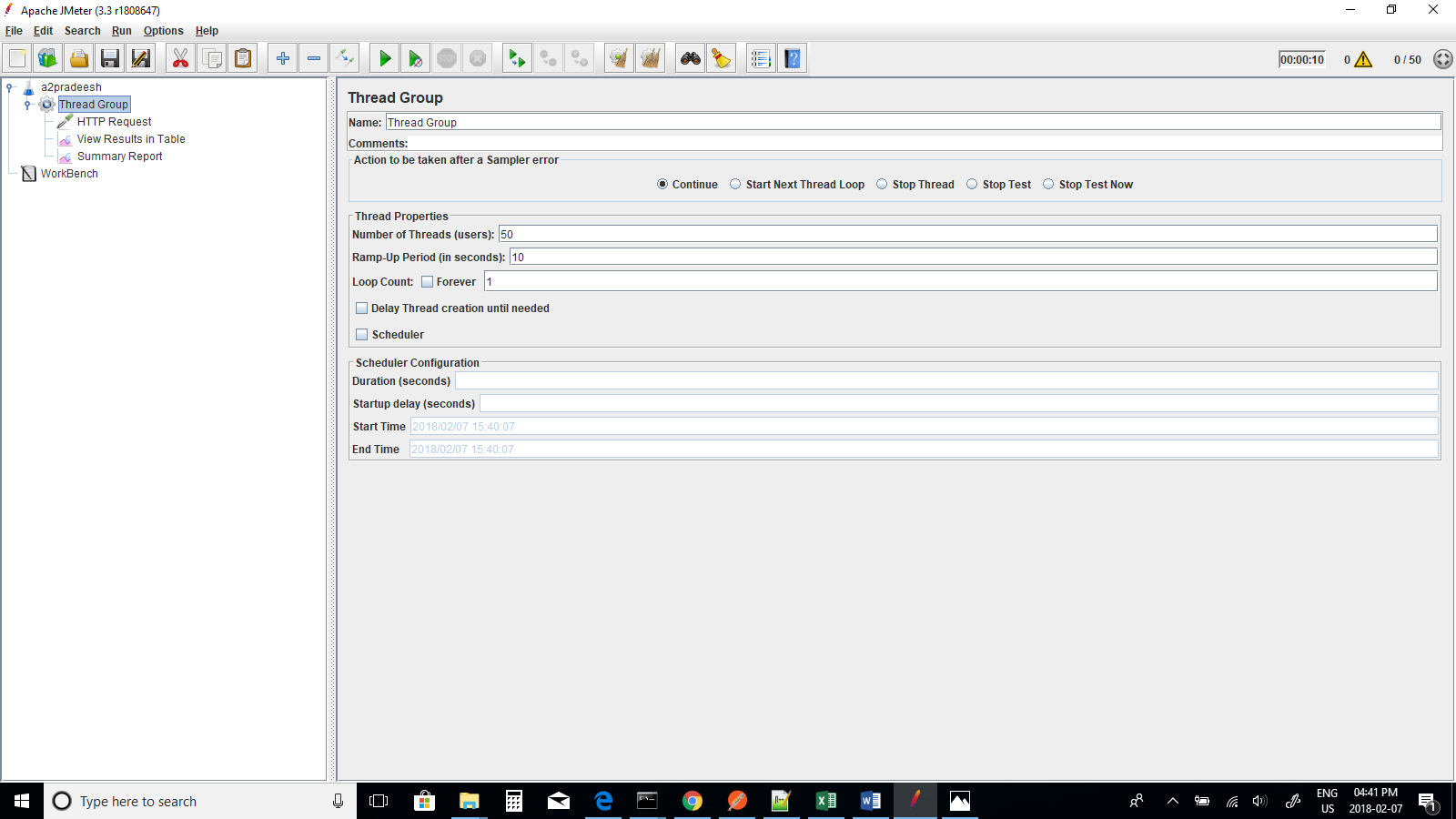


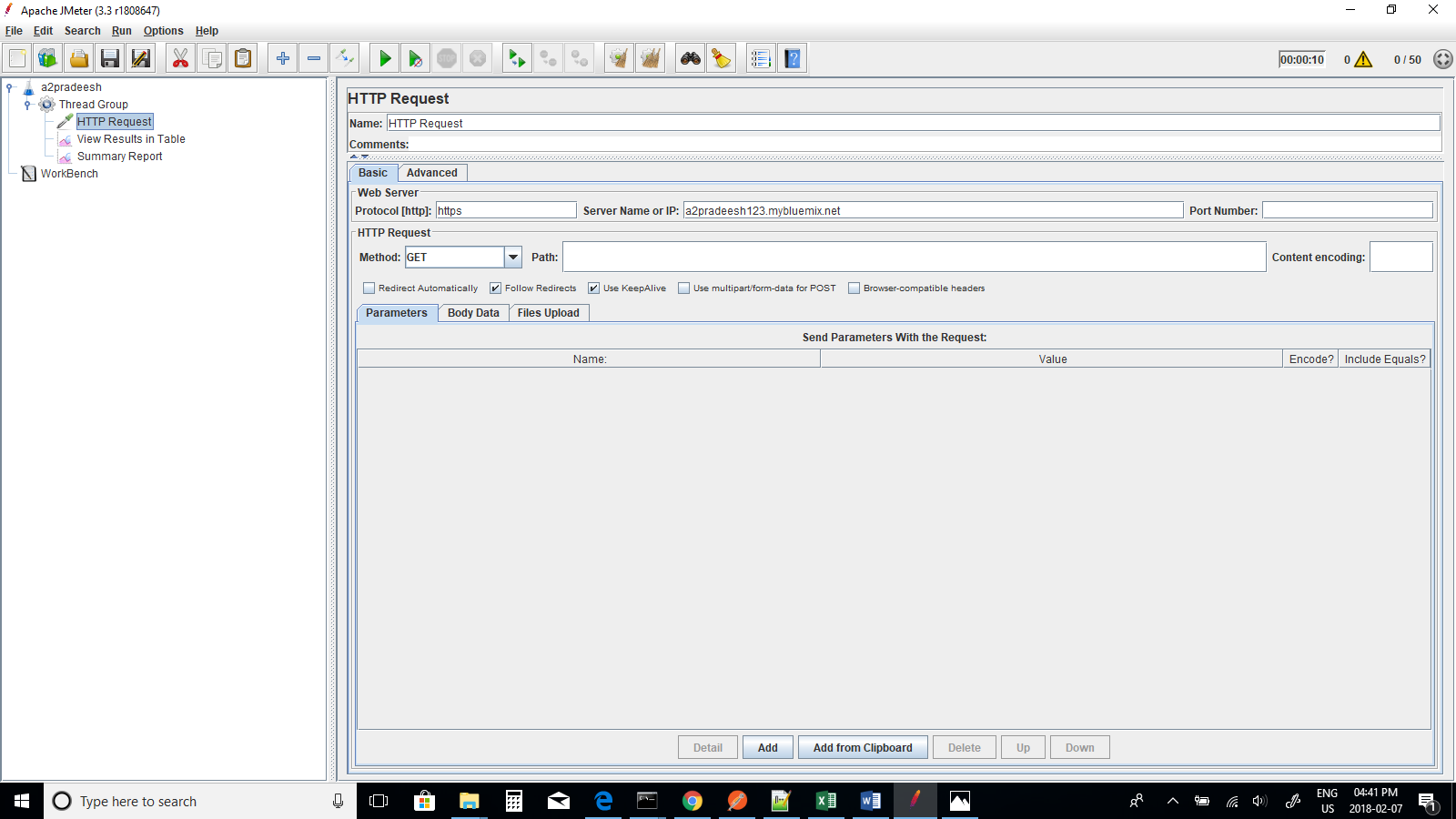


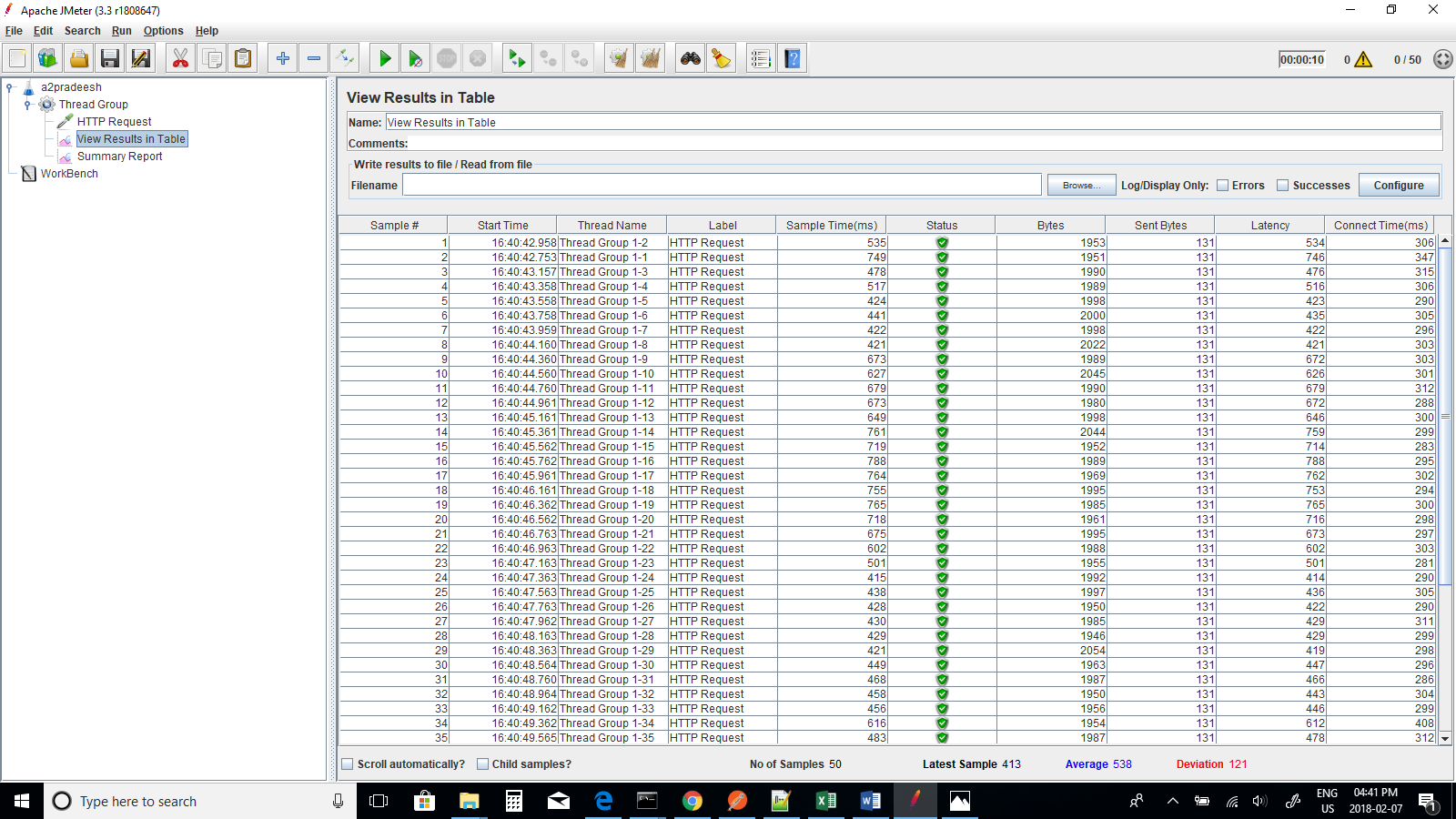


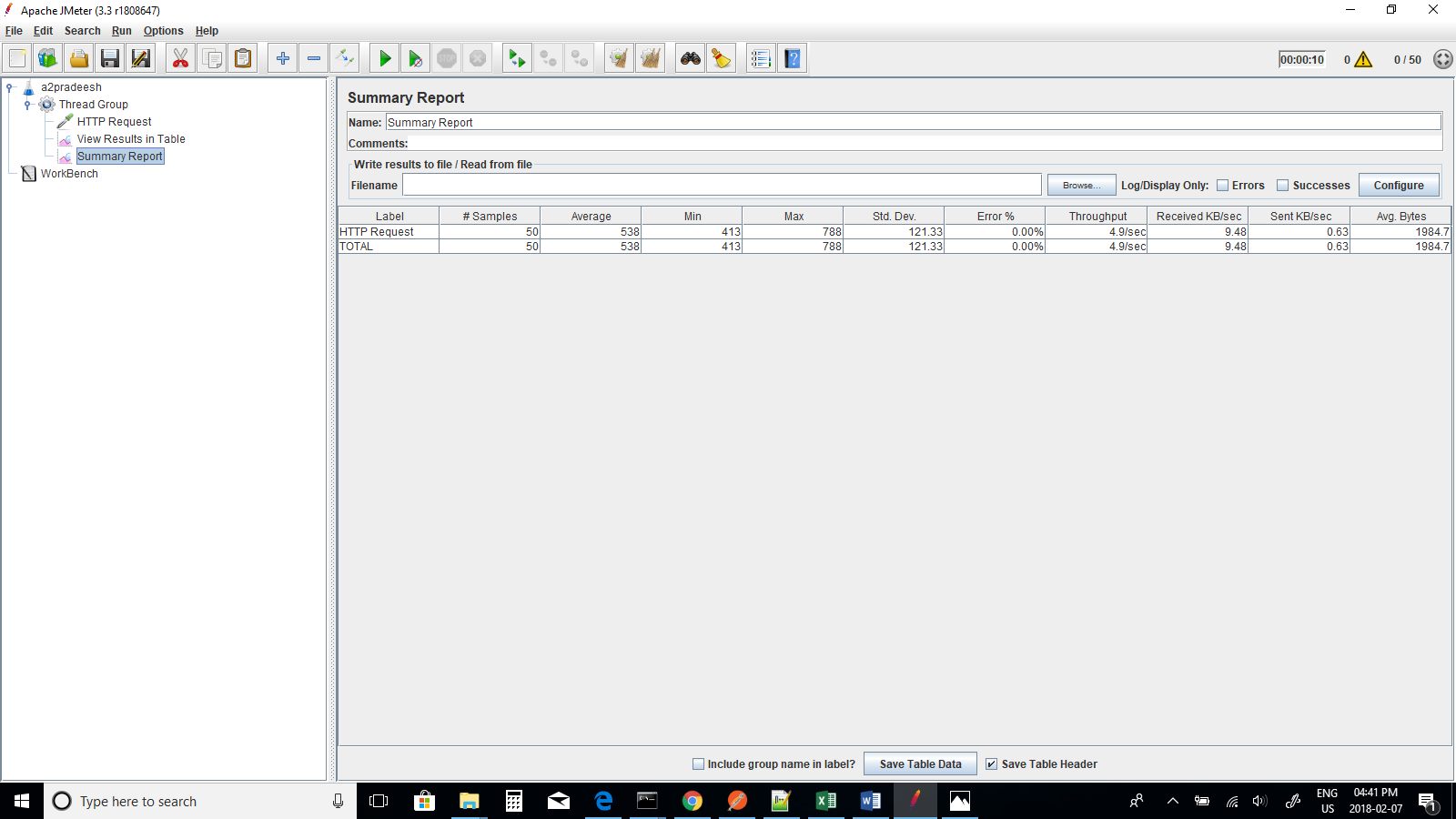
Average Response Time for 40 Threads - 6782ms

**Changing number of threads – 50**









Average Response Time for 50 Threads - 538ms

**i) URL : a2pradeesh123.mybluemix.net**

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms )** |
| 10 | 5060 |
| 20 | 463 |
| 30 | 487 |
| 40 | 6782 |
| 50 | 538 |

**ii) URL : a2pradeesh123.mybluemix.net/visitorsbycountry**

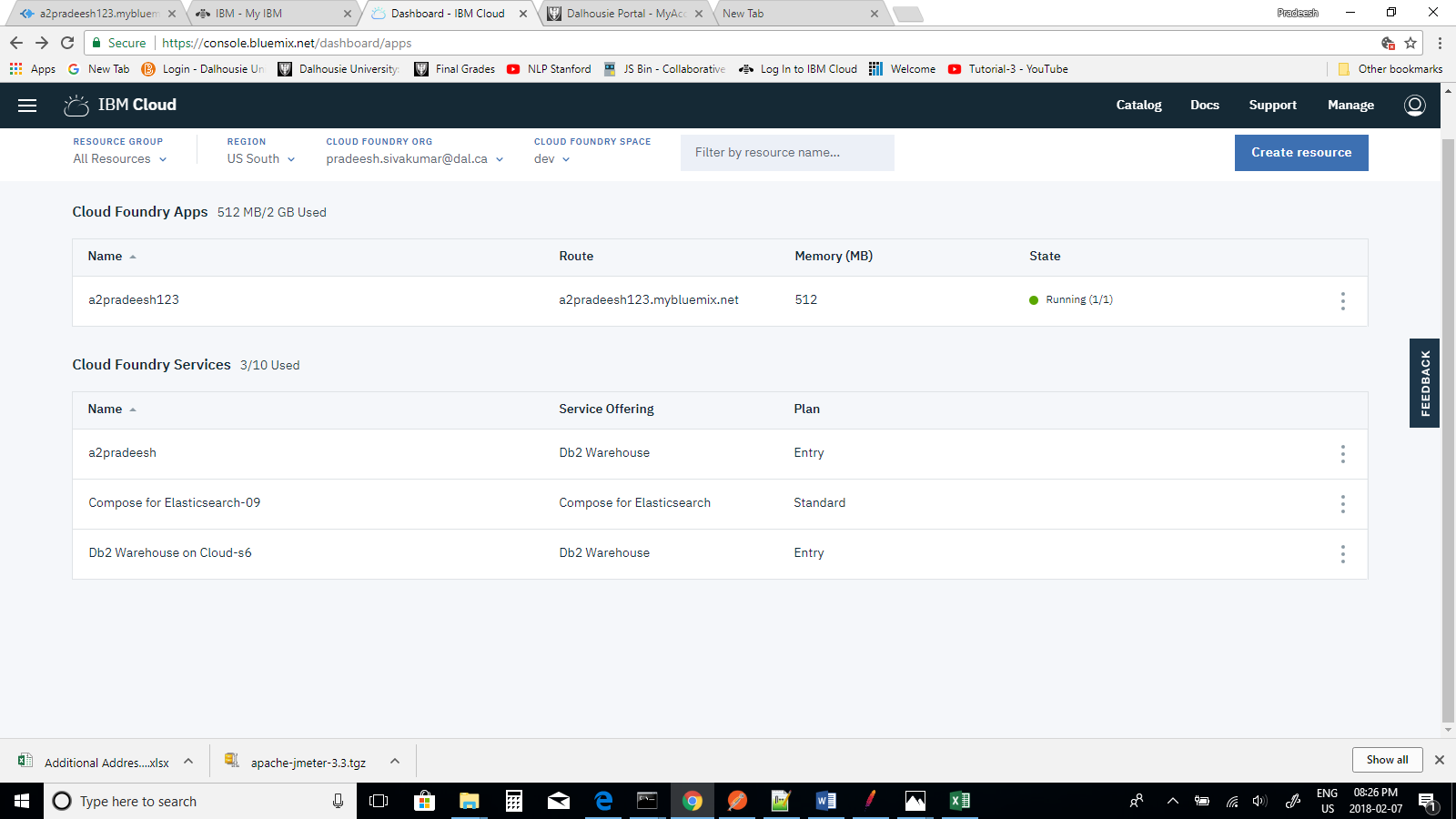
|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms )** |
| 10 | 1990 |
| 20 | 509 |
| 30 | 499 |
| 40 | 1220 |
| 50 | 2284 |

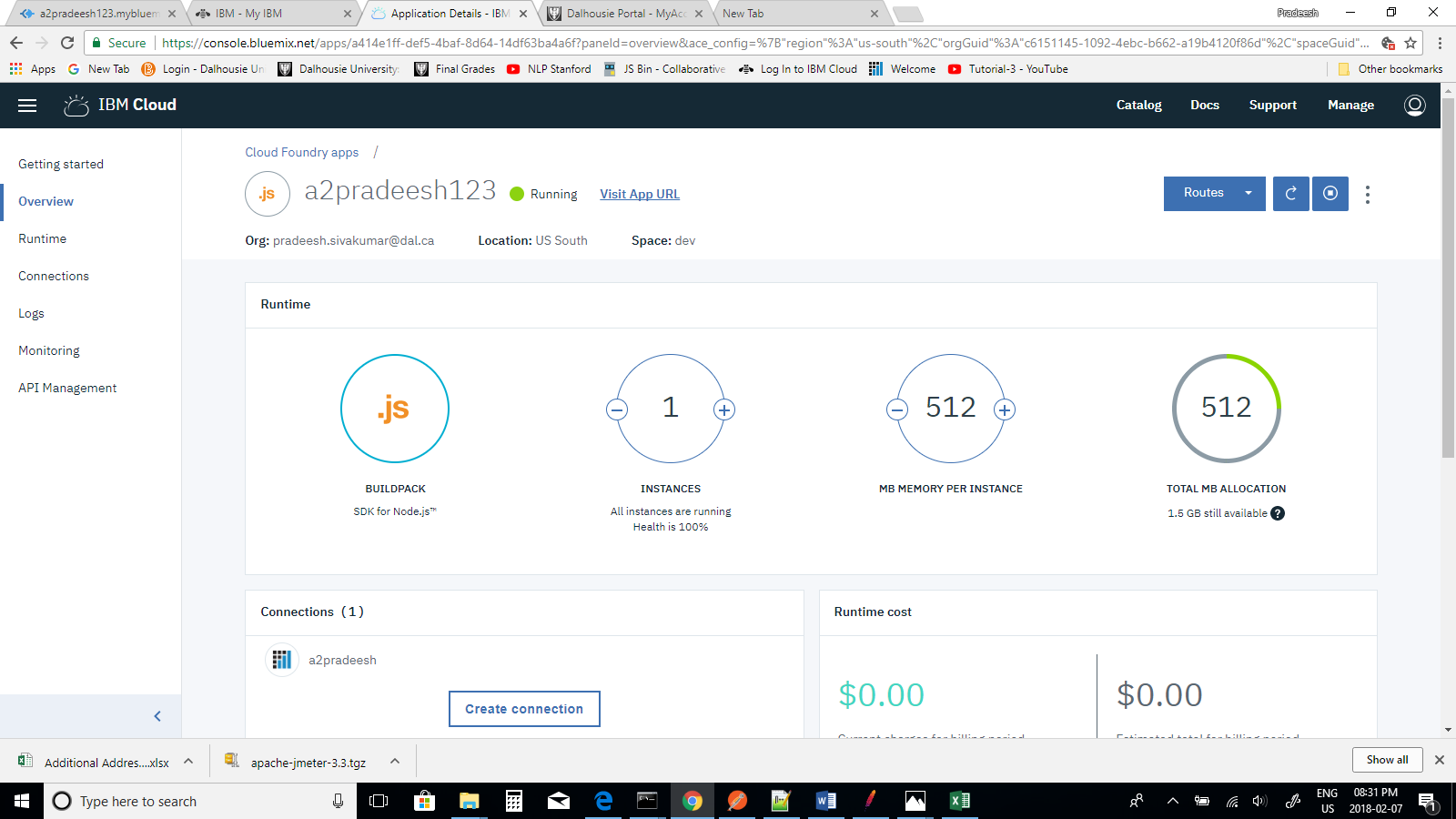
**iii) a2pradeesh123.mybluemix.net/visitorsbymode**

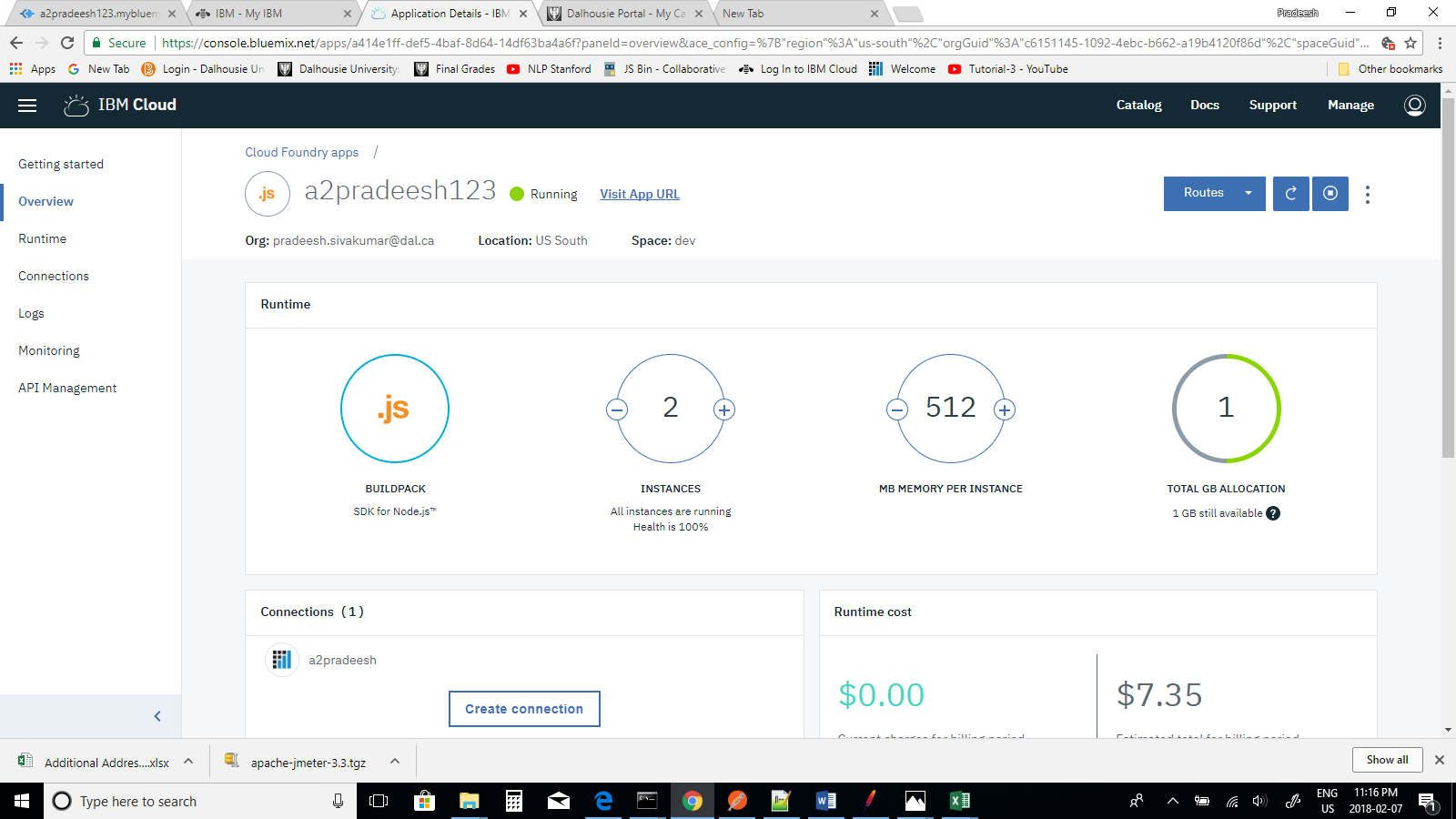
|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms )** |
| 10 | 8581 |
| 20 | 710 |
| 30 | 11875 |
| 40 | 465 |
| 50 | 499 |

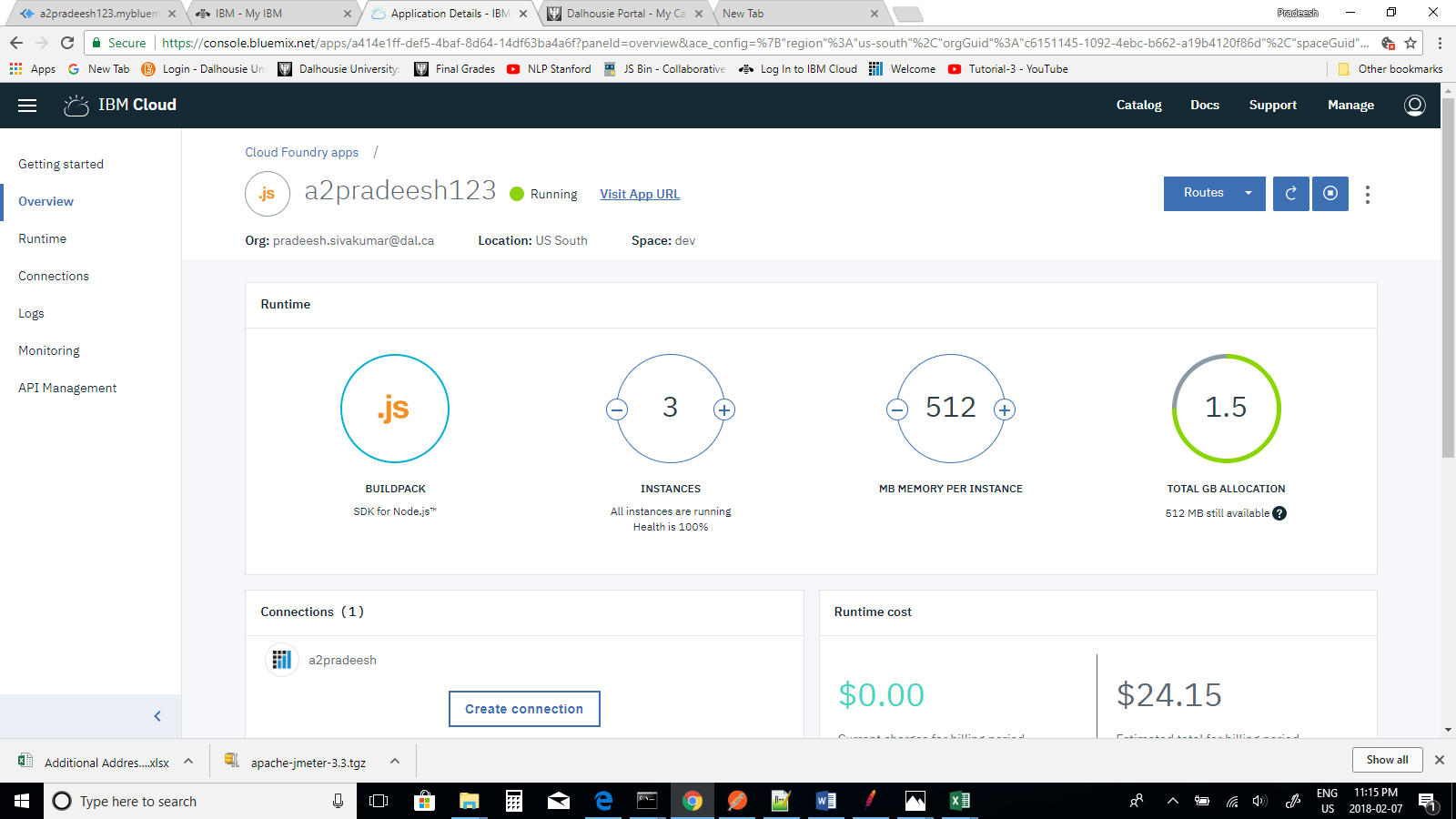
The above graphs displays number of threads with average response time in milli seconds (ms) for three URLs used in the experiment. The performance for all URLs varies in graph for each URL used. The average response time in URL-3 took more time compared to URL-1 and URL-2 and keeps on changing on each instance.

**6. Scaling**









**Q-1 a2pradeesh123.mybluemix.net**

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 1 instance** |
| 10 | 5060 |
| 20 | 473 |
| 30 | 488 |
| 40 | 6782 |
| 50 | 538 |

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 2 instance** |
| 10 | 22309 |
| 20 | 7373 |
| 30 | 61834 |
| 40 | 4060 |
| 50 | 39595 |

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 3 instance** |
| 10 | 755 |
| 20 | 319 |
| 30 | 497 |
| 40 | 47108 |
| 50 | 34142 |

**Scaling : Q-1 No of Threads Vs Average Response Time for 1, 2, and 3 instance**

**Q-2 a2pradeesh123.mybluemix.net/visitorsbycountry**

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 1 instance** |
| 10 | 1990 |
| 20 | 509 |
| 30 | 499 |
| 40 | 1220 |
| 50 | 2284 |
|  |  |
| **No of Threads** | **Average Response Time (in ms ) for 2 instance** |
| 10 | 606 |
| 20 | 966 |
| 30 | 656 |
| 40 | 10288 |
| 50 | 22439 |

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 3 instance** |
| 10 | 543 |
| 20 | 5743 |
| 30 | 1293 |
| 40 | 34229 |
| 50 | 326 |

**Scaling: Q-2 No of Threads Vs Average Response Time for 1, 2, and 3 instance**

**Q-3 a2pradeesh123.mybluemix.net/visitorsbymode**

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 1 instance** |
| 10 | 8581 |
| 20 | 710 |
| 30 | 11875 |
| 40 | 465 |
| 50 | 499 |

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 2 instance** |
| 10 | 520 |
| 20 | 1244 |
| 30 | 669 |
| 40 | 2988 |
| 50 | 519 |

|  |  |
| --- | --- |
| **No of Threads** | **Average Response Time (in ms ) for 3 instance** |
| 10 | 20165 |
| 20 | 30959 |
| 30 | 406 |
| 40 | 444 |
| 50 | 19118 |

**Scaling: Q-3 No of Threads Vs Average Response Time for 1, 2, and 3 instance**

The above graphs displays number of threads and average response time for 1, 2, and 3 instances for each web URL used in the dataset using JMeter. In the graph it has been analyzed that the average response time for instance 3 is more compared to instance 1, and instance 2 for each URL analyzed in the experiment because when more load is given the time gradually increases and keeps on changing.

**Summary**

We have been given with the visitors dataset of Nova Scotia and we are supposed to use db2 warehouse on cloud to store the dataset. Then we have to retrieve the number of visitors information from the database given for the task. For testing the web service I have used Apache JMeter to test each application given in the assignment. The application is created using NodeJS and have been uploaded in the Bluemix. The performance of multiple threads using UI application (JMeter) is calculated by changing the number of threads to 10,20,30,40, and 50 and measuring average response time in milliseconds. Similarly, for scaling previous steps are repeated by changing the number of instances to 1,2, and 3 and average response for each web URL based on the assignment is measured and the graphs are illustrated. Finally, for each requests on x axis and response on y axis is observed with response times by changing the number of instances to 1,2, and 3.

**Software tools Experience**

Based on the software tools used such as Jmeter, Postman and Overall it was a good experience. Jmeter is very useful to create multiple threads from UI application to observe response times and mostly for load and performance testing, whereas software like postman is used for testing the web service and mostly for faster and easier API development.

**References**

[**https://www.youtube.com/watch?v=4fbli5lbiAs&feature=youtu.be**](https://www.youtube.com/watch?v=4fbli5lbiAs&feature=youtu.be)

[**https://www.youtube.com/watch?v=PDEejbdQVeo**](https://www.youtube.com/watch?v=PDEejbdQVeo)

[**http://blog.getpostman.com/2017/11/18/postman-makes-soap-requests-too/**](http://blog.getpostman.com/2017/11/18/postman-makes-soap-requests-too/)

[**http://toolsqa.com/jmeter/thread-group-in-jmeter-test-plan/**](http://toolsqa.com/jmeter/thread-group-in-jmeter-test-plan/)

[**https://www.tutorialspoint.com/jmeter/jmeter\_test\_plan\_elements.htm**](https://www.tutorialspoint.com/jmeter/jmeter_test_plan_elements.htm)

[**https://console.bluemix.net/docs/tutorials/sql-database.html#sql-database-for-cloud-data**](https://console.bluemix.net/docs/tutorials/sql-database.html#sql-database-for-cloud-data)