

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

The screenshot displays the IBM Db2 Warehouse on Cloud interface. The top navigation bar includes options like 'Load', 'Explore', 'Run SQL', and 'Analyze'. The main content area is titled 'Database tables' and shows a list of schemas on the left, including 'DASH7173' (selected). The central pane shows the 'TOURISM' table selected. The right pane provides a 'Table definition' for 'TOURISM', indicating it contains 4544 rows (224 KB) and was updated on 2/6/2018 at 6:01:57 PM. Below the definition is a table with columns: COLUMN NAME, DATA TYPE, and NULLABLE.

COLUMN NAME	DATA TYPE	NULLABLE
MODE_OF_ENTRY	VARCHAR	Y
MONTH_YEAR	TIMESTAMP	Y
VISITORS_ORIGIN	VARCHAR	Y
COUNTRY	VARCHAR	Y
NUMBER_OF_VISIT...	INTEGER	Y

At the bottom of the right pane, there is a 'View Data' button.

2. After loading the given dataset (Tourism)

The screenshot shows the IBM Db2 Warehouse on Cloud interface. The top navigation bar includes 'Load', 'Explore', 'Run SQL', and 'Analyze'. The main content area displays a table named 'DASH7173.TOURISM'. The table has six columns: 'MODE_OF_ENTRY', 'MONTH_YEAR', 'VISITORS_ORIGIN', 'COUNTRY', and 'NUMBER_OF_VISITORS'. The data is sorted by 'NUMBER_OF_VISITORS' in descending order. The table contains 16 rows of data, with the first row showing 'Air' mode of entry, '2009-01-09 00:00:00' month/year, 'Quebec' visitors origin, 'Canada' country, and 2800 visitors. The last row shows 'Air' mode of entry, '2009-01-09 00:00:00' month/year, 'Other Overseas' visitors origin, 'Overseas' country, and 2300 visitors. The interface also includes a 'Back' button, a 'Delete Table' button, and an 'Export to CSV' button.

	MODE_OF_ENTRY VARCHAR(4)	MONTH_YEAR TIMESTAMP(6)	VISITORS_ORIGIN VARCHAR(23)	COUNTRY VARCHAR(13)	NUMBER_OF_VISITORS INTEGER
1	Air	2009-01-09 00:00:00	Quebec	Canada	2800
2	Air	2009-01-09 00:00:00	Ontario	Canada	23900
3	Air	2009-01-09 00:00:00	Western Canada	Canada	16800
4	Air	2009-01-09 00:00:00	New England (inc Maine)	United States	1000
5	Air	2009-01-09 00:00:00	Middle Atlantic	United States	1600
6	Air	2009-01-09 00:00:00	East North Central	United States	1200
7	Air	2009-01-09 00:00:00	West North Central	United States	400
8	Air	2009-01-09 00:00:00	South Atlantic	United States	2500
9	Air	2009-01-09 00:00:00	East South Central	United States	300
10	Air	2009-01-09 00:00:00	West South Central	United States	800
11	Air	2009-01-09 00:00:00	Mountain	United States	800
12	Air	2009-01-09 00:00:00	Pacific	United States	1300
13	Air	2009-01-09 00:00:00	United Kingdom	Overseas	3300
14	Air	2009-01-09 00:00:00	Germany	Overseas	2400
15	Air	2009-01-09 00:00:00	Other Europe	Overseas	2600
16	Air	2009-01-09 00:00:00	Other Overseas	Overseas	2300

3. Connecting the Bluemix using command line tool to login

```
Select Command Prompt
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\PRADEESH>cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK

api endpoint: https://api.ng.bluemix.net
api version: 2.92.0

C:\Users\PRADEESH>cf login
API endpoint: https://api.ng.bluemix.net

Email> pradeesh.sivakumar@dal.ca

Password>
Authenticating...
OK

Targeted org pradeesh.sivakumar@dal.ca
Targeted space dev

API endpoint: https://api.ng.bluemix.net (API version: 2.92.0)
User: pradeesh.sivakumar@dal.ca
Org: pradeesh.sivakumar@dal.ca
Space: dev

C:\Users\PRADEESH>cf create-service dashDB Entry assign2
Creating service instance assign2 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Attention: The plan 'Entry' of service 'dashDB' is not free. The instance 'assign2' will incur a cost. Contact your administrator if you think this is in error.

C:\Users\PRADEESH>
```

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

```

C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>cf logs a2pradeesh123
Retrieving logs for app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...

C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>
C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>
C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>
C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>
C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>cf push
Pushing from manifest to org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
Using manifest file C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master\manifest.yml

Deprecation warning: Route component attributes 'domain', 'domains', 'host', 'hosts' and 'no-hostname' are deprecated. Found: host.
Please see http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html#deprecated for the currently supported syntax and other app manifest deprecations. This feature will be removed in the future.

Using manifest file C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master\manifest.yml
Updating app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Using route a2pradeesh123.mybluemix.net
Uploading a2pradeesh123...
Uploading app files from: C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master
Uploading 10.4K, 10 files
Done uploading
OK
Binding service a2pradeesh to app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Stopping app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Starting app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
Downloading sdk-for-nodejs...
Downloaded sdk-for-nodejs
Creating container
Successfully created container
Downloading app package...
Downloaded app package (10.1K)
Downloading build artifacts cache...
Downloaded build artifacts cache (6.8M)
Staging...
-----> IBM SDK for Node.js Buildpack v3.17-20180105-1100
Based on Cloud Foundry Node.js Buildpack v1.5.24
-----> Creating runtime environment
NPM_CONFIG_LOGLEVEL=error
NPM_CONFIG_PRODUCTION=true
NODE_ENV=production

Type here to search
People
ENG US 08:35 PM 2018-02-06

```

Loading...

```

C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master>cf push
Pushing from manifest to org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
Using manifest file C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master\manifest.yml

Deprecation warning: Route component attributes 'domain', 'domains', 'host', 'hosts' and 'no-hostname' are deprecated. Found: host.
Please see http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html#deprecated for the currently supported syntax and other app manifest deprecations. This feature will be removed in the future.

Using manifest file C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master\manifest.yml
Updating app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Using route a2pradeesh123.mybluemix.net
Uploading a2pradeesh123...
Uploading app files from: C:\Users\PRADEESH S\Desktop\New folder\dashdb-nodejs-helloworld-master
Uploading 10.4K, 10 files
Done uploading
OK
Binding service a2pradeesh to app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Stopping app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
OK

Starting app a2pradeesh123 in org pradeesh.sivakumar@dal.ca / space dev as pradeesh.sivakumar@dal.ca...
Downloading sdk-for-nodejs...
Downloaded sdk-for-nodejs
Creating container
Successfully created container
Downloading app package...
Downloaded app package (10.1K)
Downloading build artifacts cache...
Downloaded build artifacts cache (6.8M)
Staging...
-----> IBM SDK for Node.js Buildpack v3.17-20180105-1100
Based on Cloud Foundry Node.js Buildpack v1.5.24
-----> Creating runtime environment
NPM_CONFIG_LOGLEVEL=error
NPM_CONFIG_PRODUCTION=true
NODE_ENV=production
NPM_CONFIG_CACHE=true
NPM_CONFIG_CACHE=true
-----> Installing binaries
engines.node (package.json): ^4
engines.npm (package.json): unspecified (use default)

Resolving node version ^4 via 'node-version-resolver'
Installing IBM SDK for Node.js (4.8.7) from cache
Using default npm version: 2.15.11
-----> Restoring cache
Loading 2 from cacheDirectories (default):
- node_modules
- bower_components (not cached - skipping)
-----> Checking and configuring service extensions before installing dependencies
Installing service extension(s) for: [dashdb]
Extract resources/sqlldb/clidriver.tgz to application directory
Copy resources/sqlldb/set_ibm_db_home.sh to application directory
Set staging env: source set_ibm_db_home.sh
Set boot env: export LD_LIBRARY_PATH="$HOME/clidriver/lib:$LD_LIBRARY_PATH" for application
-----> Building dependencies
Installing node modules (package.json)
-----> Checking and configuring service extensions after installing dependencies
-----> Installing App Management
-----> Caching build
Clearing previous node cache
Saving 2 cacheDirectories (default):
- node_modules
- bower_components (nothing to cache)
-----> Build succeeded!
--- cf-deployment-tracker-client@0.8
--- express@3.21.2
--- ibm_db@2.2.1
--- jade@1.11.0

Exit status 0
Staging complete
Uploading droplet, build artifacts cache...
Uploading build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (6.8M)
Uploaded droplet (41.8M)
Uploading complete
Stopping instance c76ba992-8559-44a0-8046-fdced02e4106
Destroying container
Successfully destroyed container

1 of 1 instances running
App started

Type here to search
People
ENG US 08:35 PM 2018-02-06

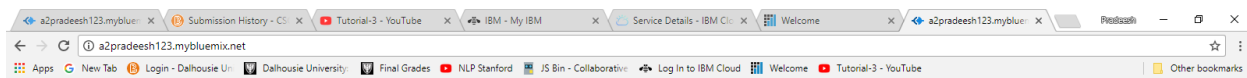
```

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.



Congratulations. Question-1.

10 rows from the tourism table

MODE_OF_ENTRY	MONTH_YEAR	VISITORS_ORIGIN	COUNTRY	NUMBER_OF_VISITORS
Air	2008-01-06 00:00:00.000000	Other Overseas	Overseas	2100
Road	2008-01-06 00:00:00.000000	Atlantic Canada	Canada	92200
Road	2008-01-06 00:00:00.000000	Quebec	Canada	5300
Road	2008-01-06 00:00:00.000000	Ontario	Canada	18200
Road	2008-01-06 00:00:00.000000	Western Canada	Canada	2200
Road	2008-01-06 00:00:00.000000	Maine (by road only)	United States	1700
Road	2008-01-06 00:00:00.000000	Other New England	United States	4700
Road	2008-01-06 00:00:00.000000	Middle Atlantic	United States	2900
Road	2008-01-06 00:00:00.000000	East North Central	United States	1000
Road	2008-01-06 00:00:00.000000	West North Central	United States	400



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.

Congratulations. Question-2.

Visitorsbycountry

TOTAL_VISITORS	YEAR	COUNTRY
177100	2010	United States
72000	2011	Overseas
1758500	2012	Canada
184900	2009	United States
1796600	2006	Canada
82200	2016	Overseas
223700	2007	United States
228800	2017	United States
76500	2006	Overseas
221900	2016	United States
1707100	2014	Canada
193700	2015	United States
1933400	2016	Canada
72200	2007	Overseas
66000	2013	Overseas
64000	2010	Overseas
190600	2008	United States
81300	2015	Overseas
242400	2006	United States
178900	2014	United States
1843100	2007	Canada
84500	2017	Overseas
158100	2013	United States
165800	2012	United States
72900	2014	Overseas
1837300	2009	Canada
1667200	2013	Canada
1815300	2008	Canada
1844100	2017	Canada

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.

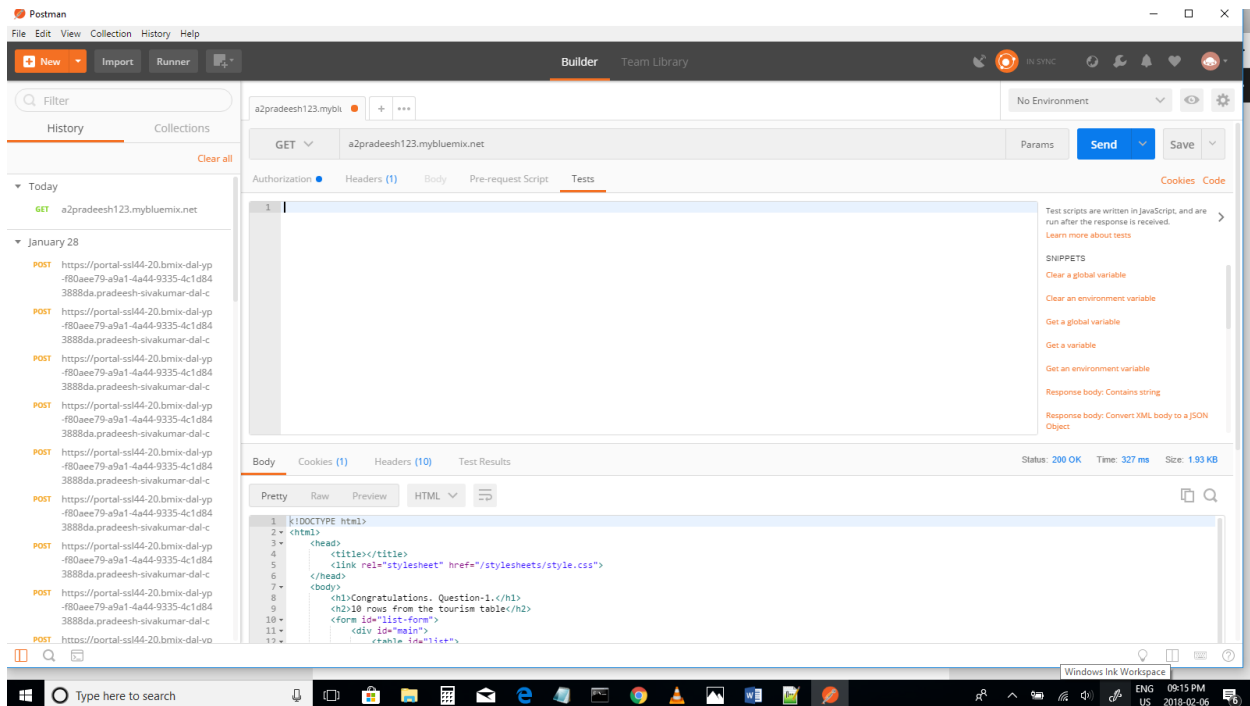
Congratulations. Question-3.

Visitorsbymode

TOTAL_VISITORS	YEAR	MODE_OF_ENTRY
667700	2007	Air
1470100	2017	Road
631100	2015	Air
1437700	2015	Road
701000	2016	Air
1471300	2007	Road
1352400	2010	Road
1313200	2014	Road
1290800	2013	Road
1470100	2006	Road
687600	2017	Air
600500	2013	Air
606700	2009	Air
636900	2012	Air
1297200	2011	Road
1397800	2008	Road
681200	2008	Air
1353800	2012	Road
645400	2006	Air
1536500	2016	Road
655800	2011	Air
602100	2010	Air
1484400	2009	Road
645700	2014	Air

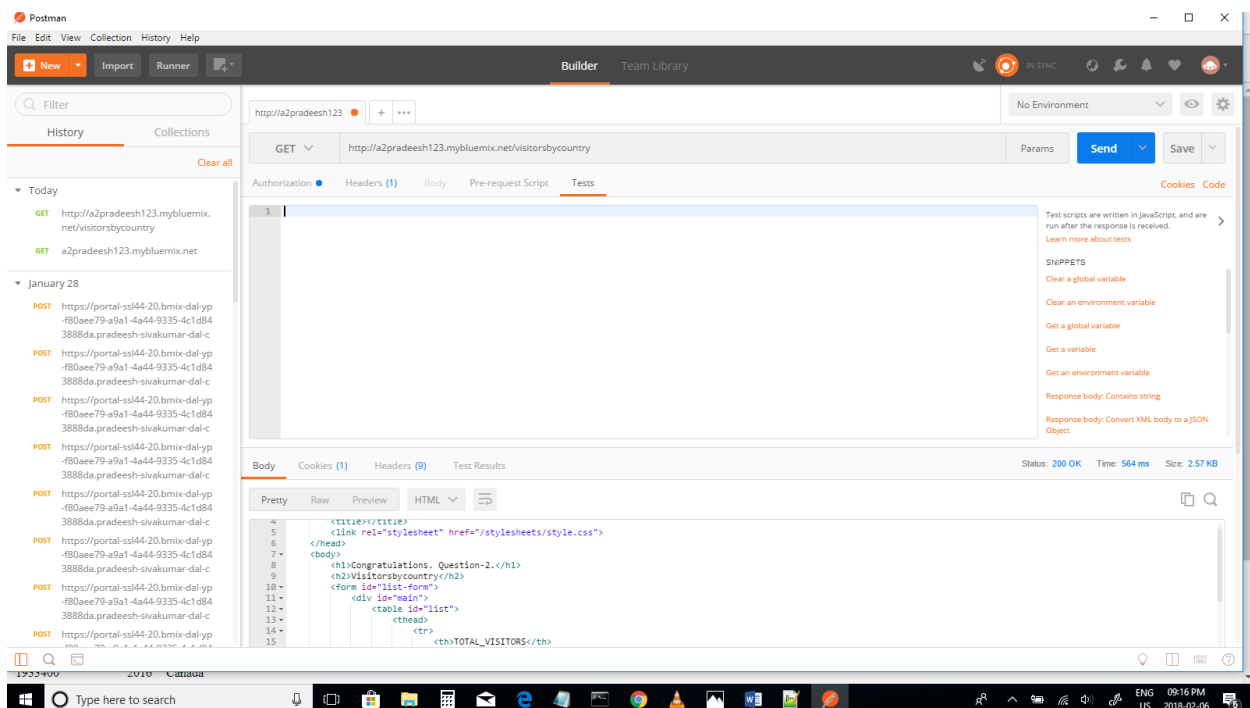
4. Testing Web service through Postman Software

Q-1 URL : a2pradeesh123.mybluemix.net



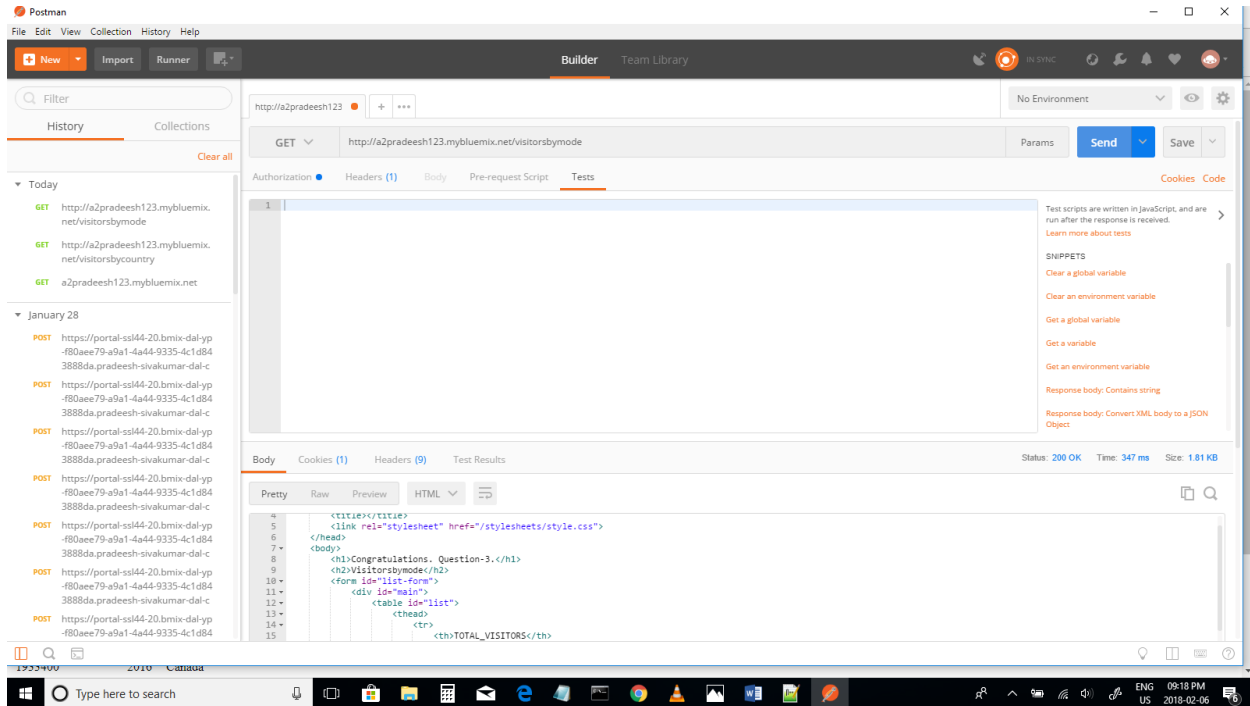
Response Time for the web URL – 327 ms

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



Response Time for the web URL – 564 ms

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



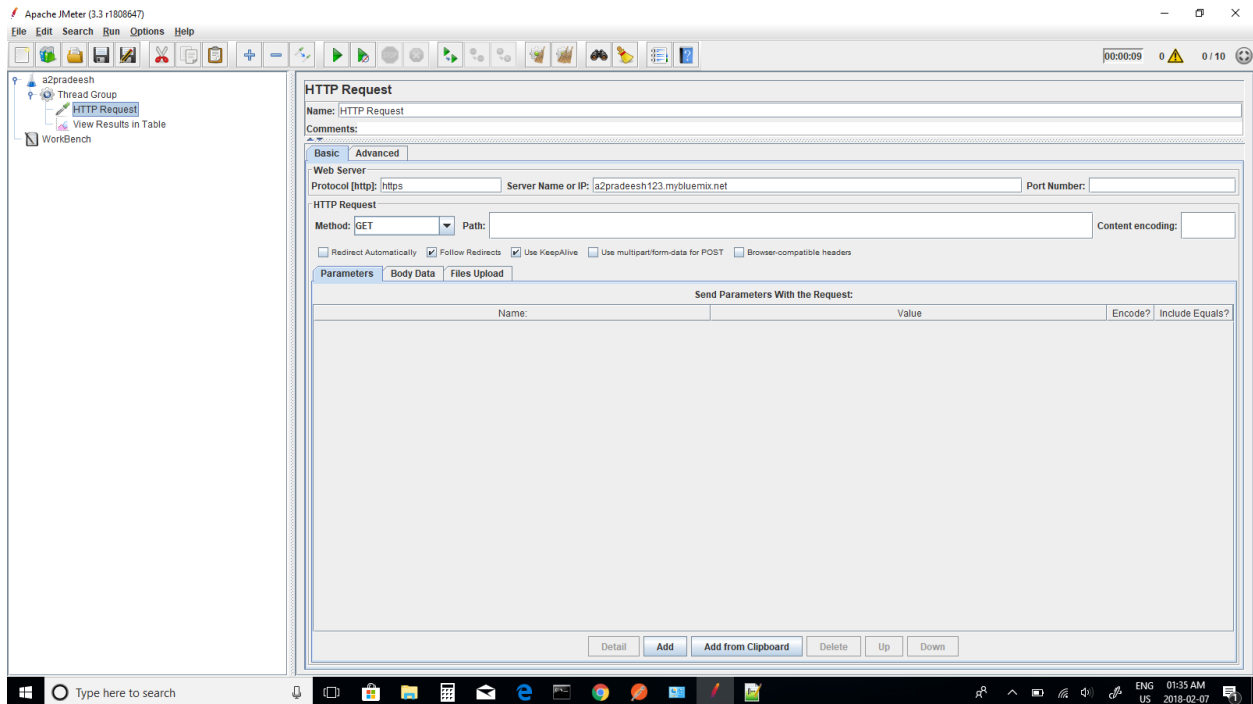
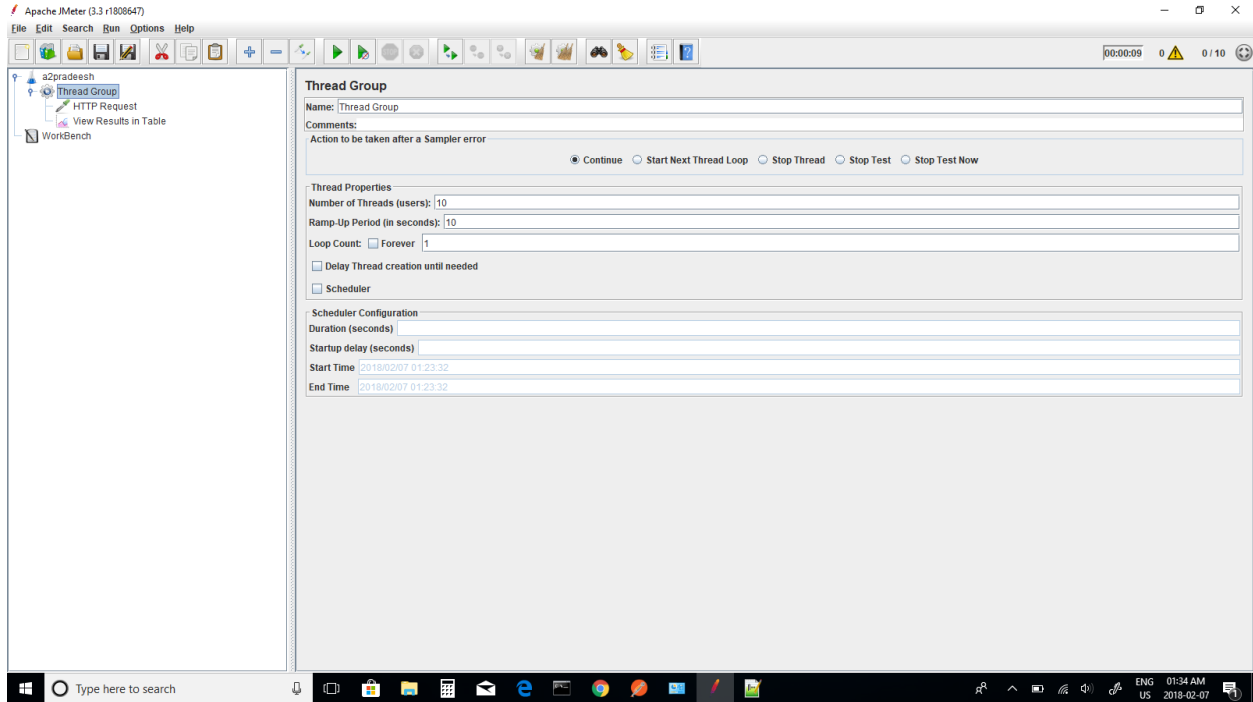
Response Time for the web URL – 347 ms

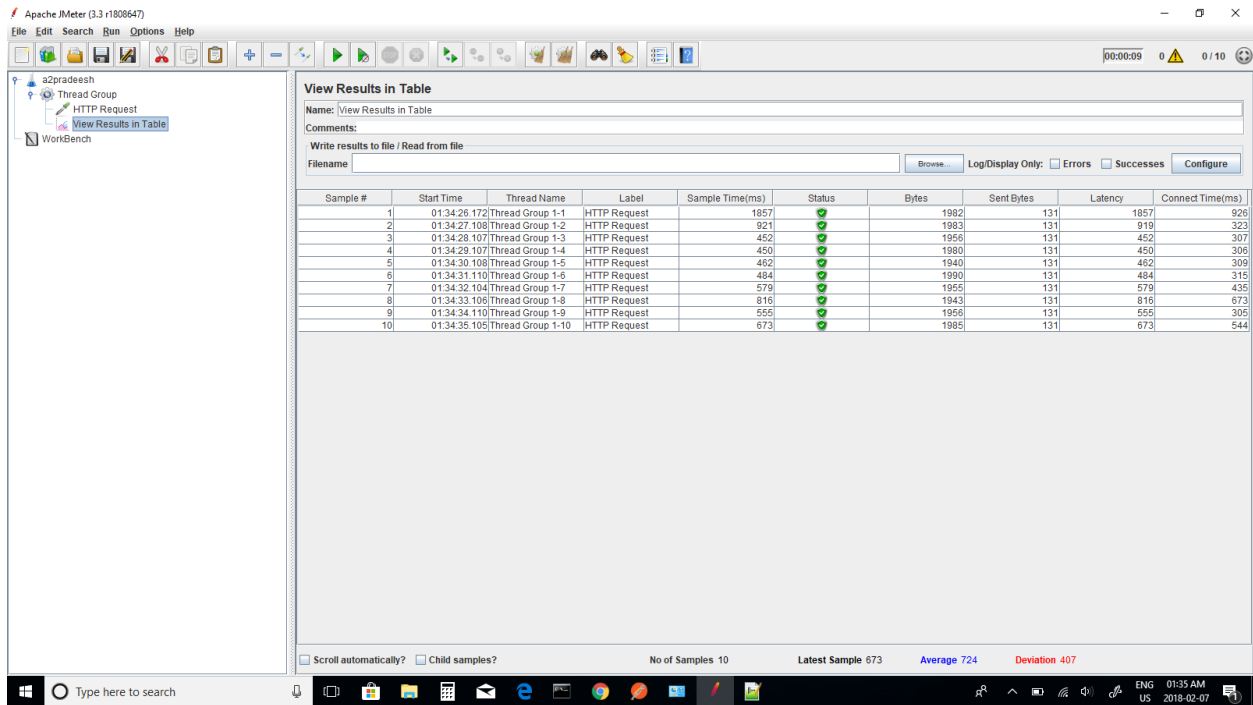
The above results shows that the web service for three URLs using postman software. It has been observed that for all URL the response time is different. The time taken for the first web URL is 327 ms, The time taken for second web URL is 564 ms and the time taken for third web URL is 564ms. This shows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

5. UI Application through JMeter

i) URL : a2pradeesh123.mybluemix.net

Number of Threads-10





Apache JMeter (3.3 r1808647)

File Edit Search Run Options Help

00:00:09 0 / 10

View Results in Table

Name: View Results in Table

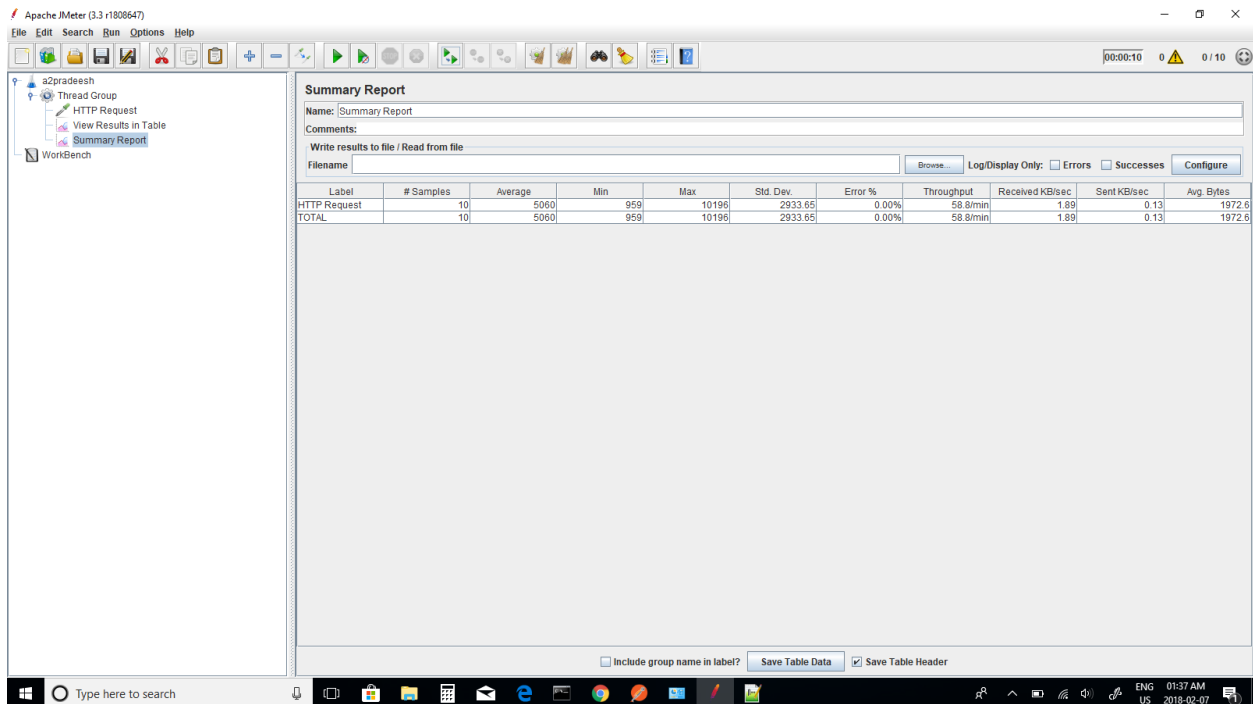
Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	01:34:26.172	Thread Group 1-1	HTTP Request	1857	✓	1982	131	1857	926
2	01:34:27.108	Thread Group 1-2	HTTP Request	921	✓	1983	131	919	323
3	01:34:28.107	Thread Group 1-3	HTTP Request	452	✓	1956	131	452	307
4	01:34:29.107	Thread Group 1-4	HTTP Request	450	✓	1980	131	450	305
5	01:34:30.108	Thread Group 1-5	HTTP Request	462	✓	1940	131	462	309
6	01:34:31.110	Thread Group 1-6	HTTP Request	484	✓	1990	131	484	315
7	01:34:32.104	Thread Group 1-7	HTTP Request	579	✓	1955	131	579	435
8	01:34:33.106	Thread Group 1-8	HTTP Request	816	✓	1943	131	816	673
9	01:34:34.110	Thread Group 1-9	HTTP Request	555	✓	1956	131	555	305
10	01:34:35.105	Thread Group 1-10	HTTP Request	673	✓	1985	131	673	544

☐ Scroll automatically? ☐ Child samples? No of Samples 10 Latest Sample 673 Average 724 Deviation 407



Apache JMeter (3.3 r1808647)

File Edit Search Run Options Help

00:00:10 0 / 10

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

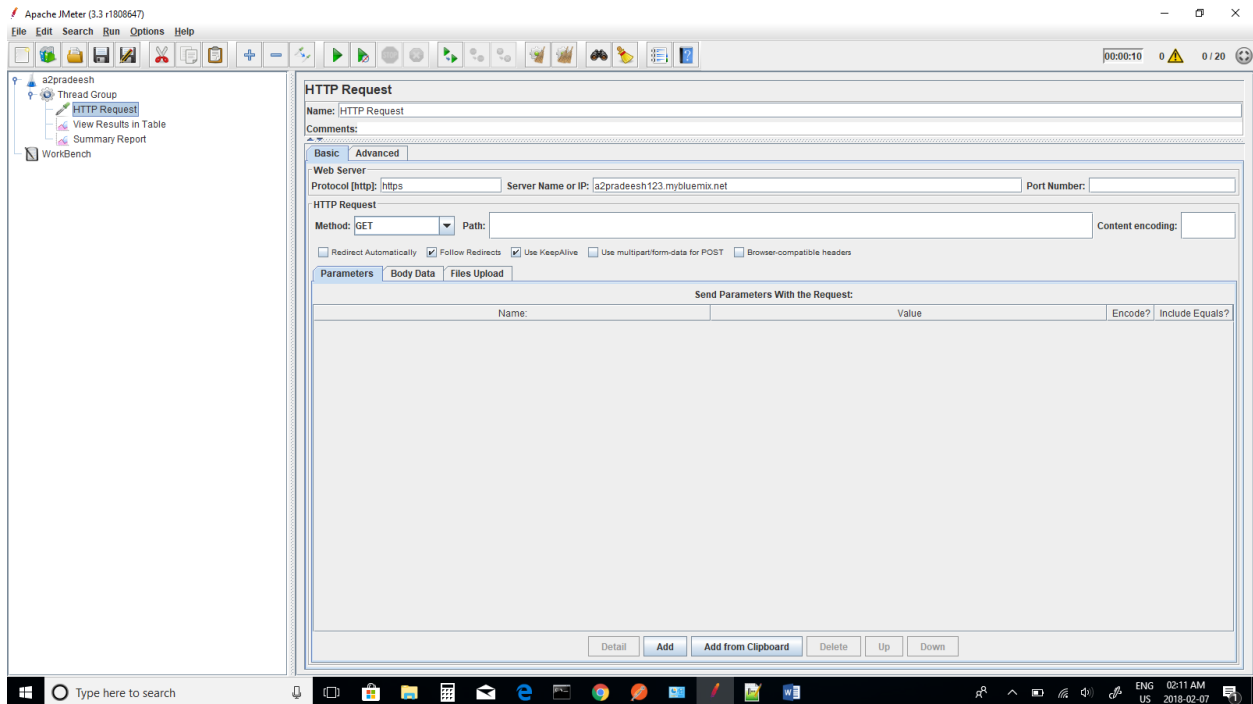
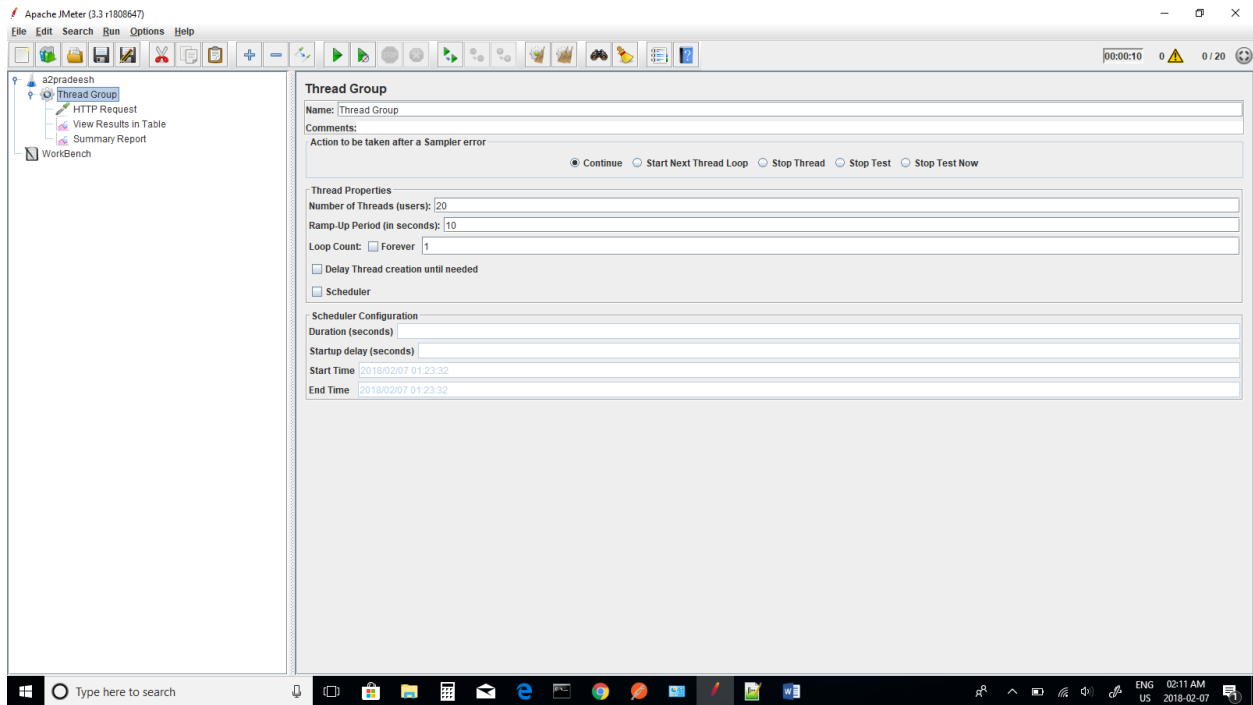
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	10	5060	959	10196	2933.65	0.00%	58.8/min	1.89	0.13	1972.6
TOTAL	10	5060	959	10196	2933.65	0.00%	58.8/min	1.89	0.13	1972.6

☐ Include group name in label? ☒ Save Table Header

Average Response Time for 10 Threads - 5060ms

Changing number of threads – 20



Apache JMeter (3.3 r180647)

File Edit Search Run Options Help

00:00:10 0 / 20

a2pradeesh

- Thread Group
 - HTTP Request
 - View Results in Table
 - Summary Report
- WorkBench

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	02:10:35.825	Thread Group 1-1	HTTP Request	664	✓	2042	131	664	361
2	02:10:36.332	Thread Group 1-2	HTTP Request	448	✓	2038	131	448	304
3	02:10:36.831	Thread Group 1-3	HTTP Request	423	✓	1954	131	423	298
4	02:10:37.328	Thread Group 1-4	HTTP Request	497	✓	1956	131	496	320
5	02:10:37.829	Thread Group 1-5	HTTP Request	429	✓	1943	131	429	291
6	02:10:38.328	Thread Group 1-6	HTTP Request	507	✓	2010	131	507	305
7	02:10:38.829	Thread Group 1-7	HTTP Request	438	✓	1998	131	438	309
8	02:10:39.330	Thread Group 1-8	HTTP Request	442	✓	1942	131	440	284
9	02:10:39.831	Thread Group 1-9	HTTP Request	412	✓	1988	131	411	289
10	02:10:40.330	Thread Group 1-10	HTTP Request	420	✓	2038	131	420	292
11	02:10:40.830	Thread Group 1-11	HTTP Request	425	✓	2017	131	424	303
12	02:10:41.331	Thread Group 1-12	HTTP Request	452	✓	1991	131	461	324
13	02:10:41.831	Thread Group 1-13	HTTP Request	417	✓	2055	131	416	295
14	02:10:42.330	Thread Group 1-14	HTTP Request	508	✓	1985	131	508	318
15	02:10:42.831	Thread Group 1-15	HTTP Request	432	✓	1985	131	432	304
16	02:10:43.332	Thread Group 1-16	HTTP Request	507	✓	1954	131	507	323
17	02:10:43.832	Thread Group 1-17	HTTP Request	423	✓	1943	131	422	291
18	02:10:44.332	Thread Group 1-18	HTTP Request	462	✓	1945	131	461	324
19	02:10:44.831	Thread Group 1-19	HTTP Request	459	✓	1989	131	459	306
20	02:10:45.332	Thread Group 1-20	HTTP Request	487	✓	2046	131	487	357

☐ Scroll automatically? ☐ Child samples? No of Samples 20 Latest Sample 487 Average 463 Deviation 55

Apache JMeter (3.3 r180647)

File Edit Search Run Options Help

00:00:10 0 / 20

a2pradeesh

- Thread Group
 - HTTP Request
 - View Results in Table
 - Summary Report
- WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

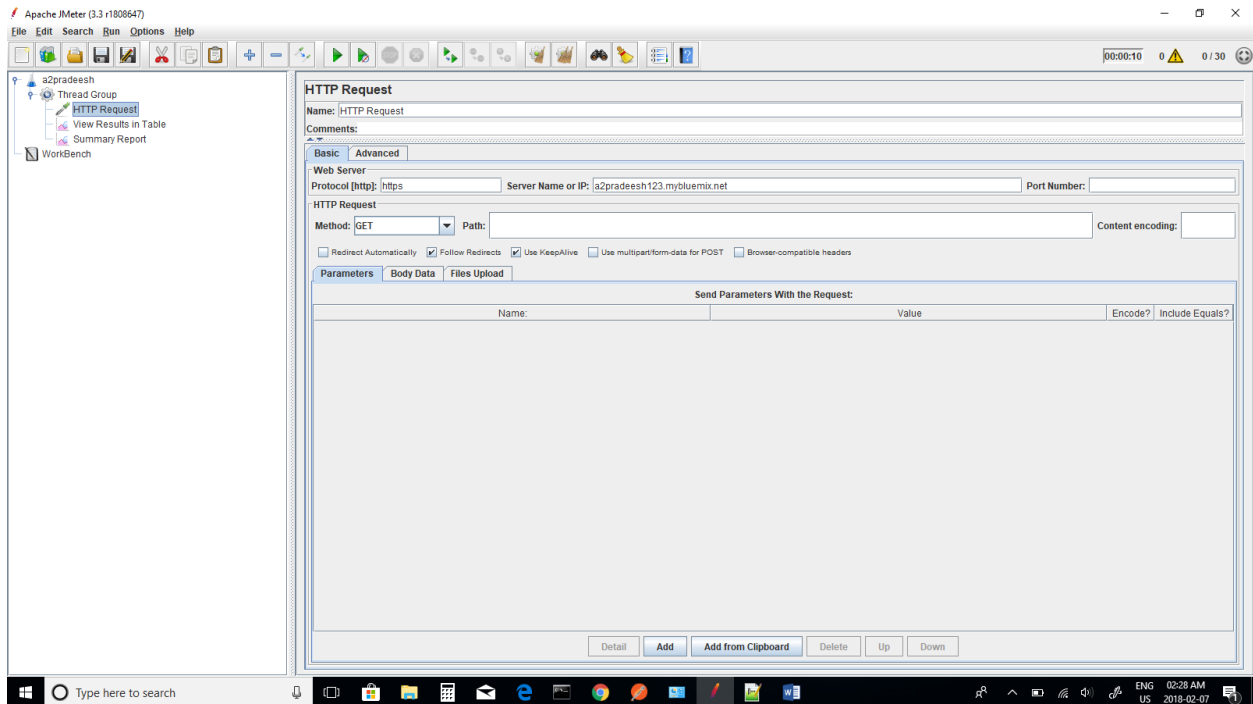
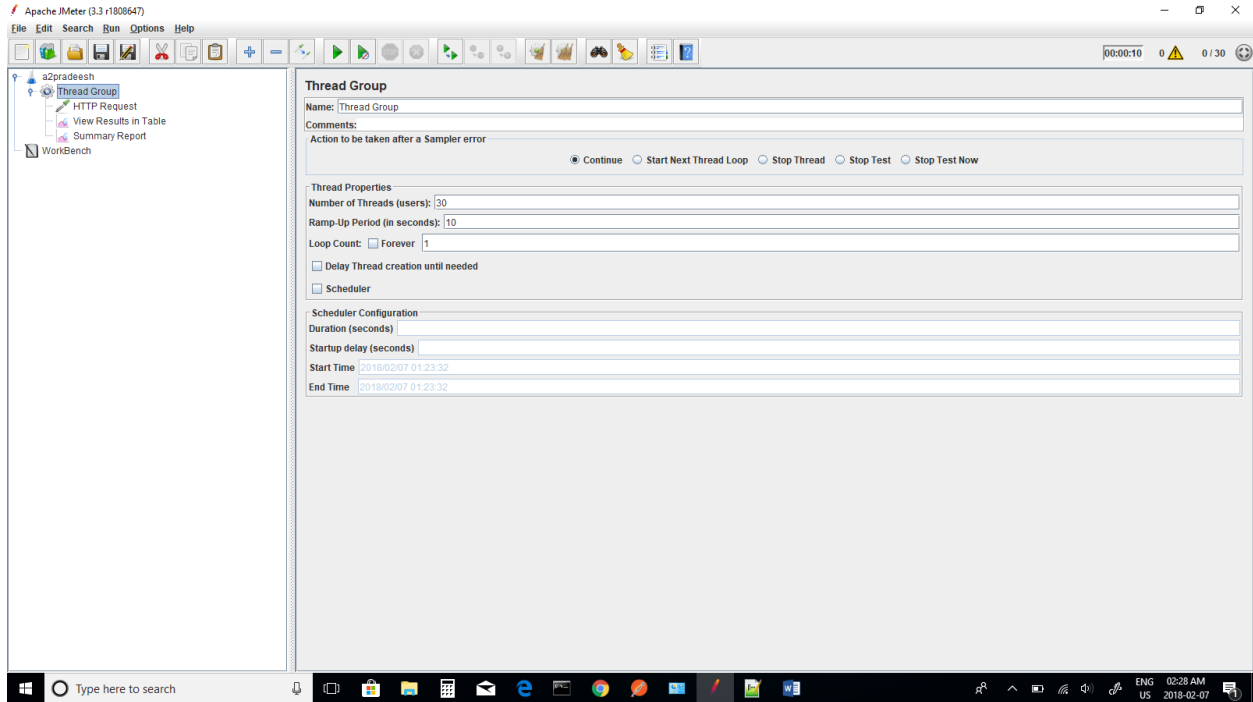
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	463	412	664	55.98	0.00%	2.0/sec	3.89	0.26	1990.9
TOTAL	20	463	412	664	55.98	0.00%	2.0/sec	3.89	0.26	1990.9

☐ Include group name in label? ☒ Save Table Header

Average Response Time for 20 Threads - 463ms

Changing number of threads – 30



Apache JMeter (3.3 r1806647)

File Edit Search Run Options Help

00:00:10 0 0/30

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	02:27:54.993	Thread Group 1-1	HTTP Request	1066	✓	1954	131	1066	364
2	02:27:55.328	Thread Group 1-2	HTTP Request	732	✓	1942	131	732	340
3	02:27:55.661	Thread Group 1-3	HTTP Request	427	✓	2044	131	426	298
4	02:27:56.002	Thread Group 1-4	HTTP Request	493	✓	1990	131	490	339
5	02:27:56.329	Thread Group 1-5	HTTP Request	462	✓	1945	131	461	327
6	02:27:56.662	Thread Group 1-6	HTTP Request	458	✓	2037	131	457	326
7	02:27:56.997	Thread Group 1-7	HTTP Request	472	✓	2053	131	470	302
8	02:27:57.331	Thread Group 1-8	HTTP Request	451	✓	1986	131	451	328
9	02:27:57.665	Thread Group 1-9	HTTP Request	444	✓	2038	131	444	321
10	02:27:57.998	Thread Group 1-10	HTTP Request	439	✓	1951	131	438	318
11	02:27:58.331	Thread Group 1-11	HTTP Request	472	✓	2000	131	471	326
12	02:27:58.663	Thread Group 1-12	HTTP Request	456	✓	1988	131	456	327
13	02:27:58.997	Thread Group 1-13	HTTP Request	476	✓	2046	131	476	340
14	02:27:59.332	Thread Group 1-14	HTTP Request	456	✓	1998	131	455	317
15	02:27:59.665	Thread Group 1-15	HTTP Request	444	✓	1993	131	444	321
16	02:27:59.999	Thread Group 1-16	HTTP Request	444	✓	1993	131	443	319
17	02:28:00.331	Thread Group 1-17	HTTP Request	460	✓	1981	131	459	313
18	02:28:00.665	Thread Group 1-18	HTTP Request	451	✓	1988	131	450	319
19	02:28:00.998	Thread Group 1-19	HTTP Request	490	✓	1993	131	490	355
20	02:28:01.331	Thread Group 1-20	HTTP Request	487	✓	2044	131	486	318
21	02:28:01.664	Thread Group 1-21	HTTP Request	451	✓	1987	131	450	312
22	02:28:01.998	Thread Group 1-22	HTTP Request	437	✓	1993	131	436	310
23	02:28:02.331	Thread Group 1-23	HTTP Request	437	✓	1941	131	437	316
24	02:28:02.665	Thread Group 1-24	HTTP Request	470	✓	1943	131	470	323
25	02:28:03.000	Thread Group 1-25	HTTP Request	467	✓	1943	131	466	347
26	02:28:03.333	Thread Group 1-26	HTTP Request	429	✓	1989	131	429	303
27	02:28:03.665	Thread Group 1-27	HTTP Request	448	✓	2048	131	447	315
28	02:28:04.000	Thread Group 1-28	HTTP Request	450	✓	1945	131	450	319
29	02:28:04.333	Thread Group 1-29	HTTP Request	441	✓	2049	131	441	315
30	02:28:04.666	Thread Group 1-30	HTTP Request	523	✓	1993	131	523	304

☐ Scroll automatically? ☐ Child samples? No of Samples 30 Latest Sample 523 Average 487 Deviation 119

Apache JMeter (3.3 r1806647)

File Edit Search Run Options Help

00:00:10 0 0/30

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

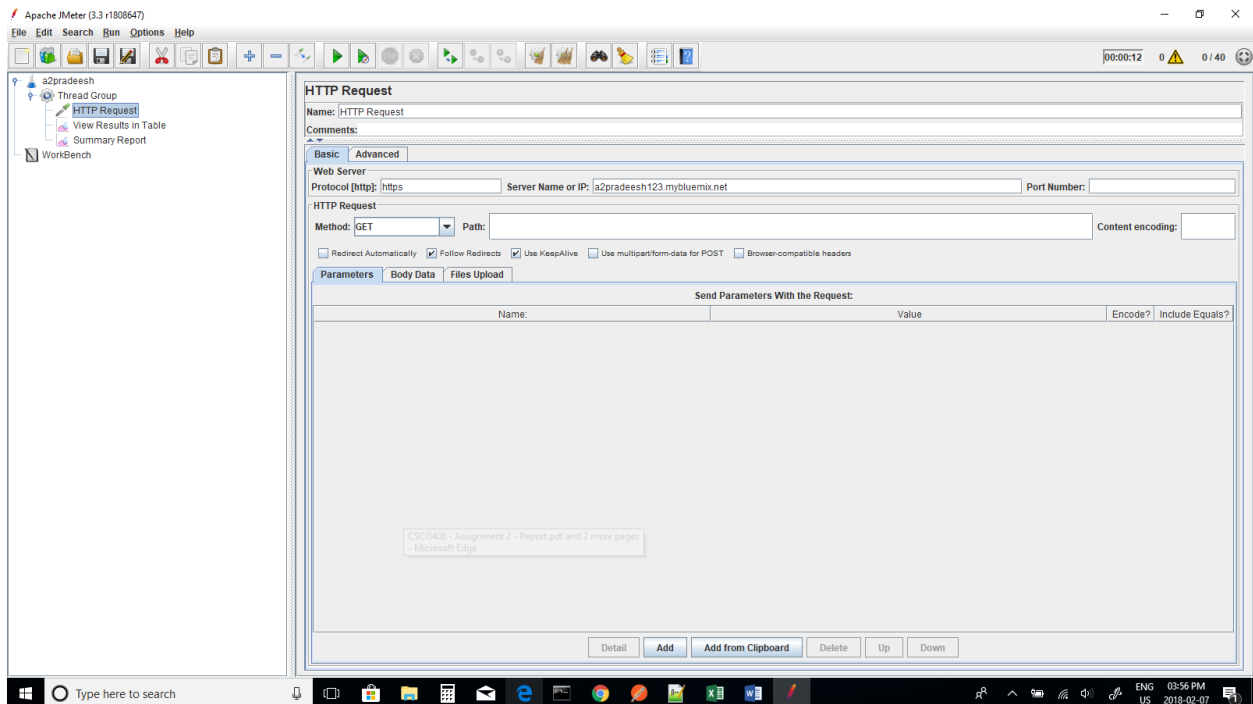
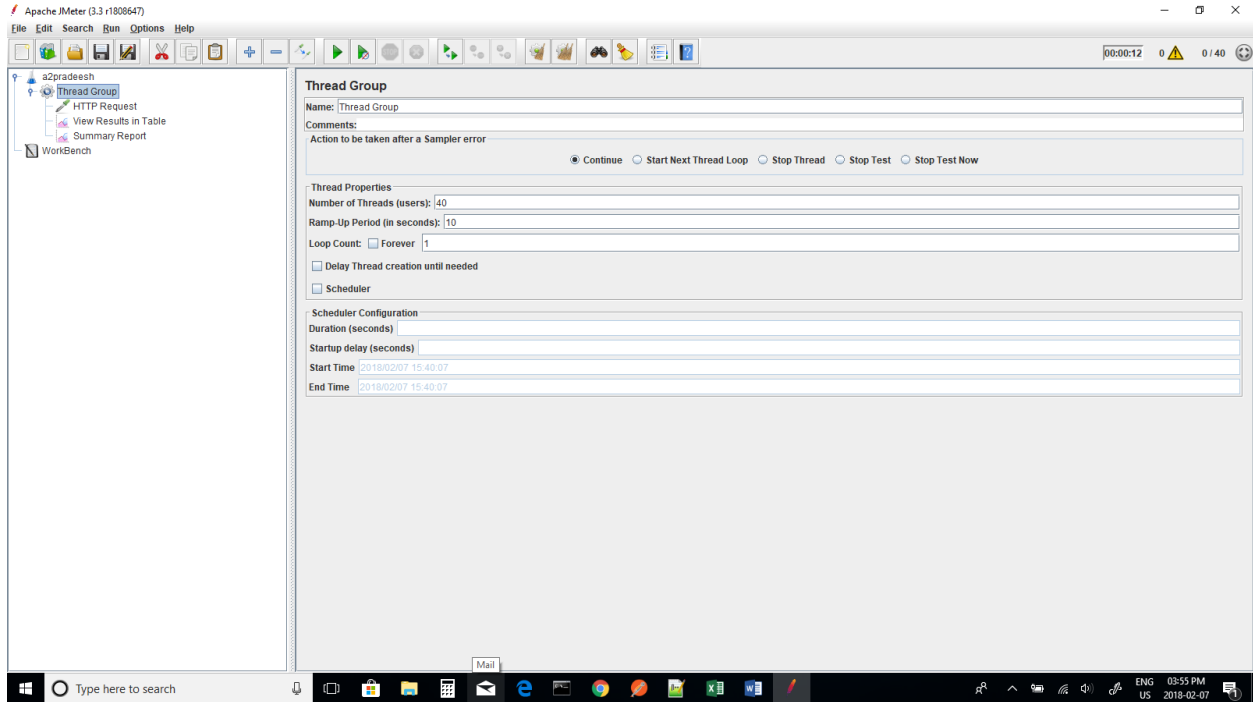
Filename: Browse Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Av. Bytes
HTTP Request	30	487	427	1056	119.82	0.00%	2.9/sec	5.73	0.38	1993.2
TOTAL	30	487	427	1056	119.82	0.00%	2.9/sec	5.73	0.38	1993.2

☐ Include group name in label? ☒ Save Table Header

Average Response Time for 30 Threads - 487ms

Changing number of threads – 40



The screenshot displays the Apache JMeter 3.3 r1808647 interface. The left sidebar shows a tree view with 'a2pradeesh' as the root, containing 'Thread Group', 'HTTP Request', 'View Results in Table', 'Summary Report', and 'WorkBench'. The main window is split into two panes. The top pane, titled 'View Results in Table', shows a table of 40 samples. The bottom pane, titled 'Summary Report', shows a summary table for the 'HTTP Request' label.

View Results in Table

Name: View Results in Table
Comments:
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes ☐ Configures

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connected Time(ms)
1	15:55:07.186	Thread Group 1-19	HTTP Request	7049	✓	1942	131	7048	318
2	15:55:06.686	Thread Group 1-17	HTTP Request	7561	✓	1989	131	7561	368
3	15:55:02.744	Thread Group 1-1	HTTP Request	11498	✓	2042	131	11487	1078
4	15:55:02.937	Thread Group 1-2	HTTP Request	11298	✓	1991	131	11292	885
5	15:55:04.187	Thread Group 1-7	HTTP Request	10054	✓	1957	131	10053	319
6	15:55:06.198	Thread Group 1-15	HTTP Request	8954	✓	1952	131	8954	329
7	15:55:03.186	Thread Group 1-3	HTTP Request	11052	✓	1989	131	11043	636
8	15:55:06.936	Thread Group 1-18	HTTP Request	7317	✓	1993	131	7314	320
9	15:55:08.436	Thread Group 1-24	HTTP Request	5819	✓	1991	131	5818	311
10	15:55:09.186	Thread Group 1-27	HTTP Request	5073	✓	1985	131	5073	292
11	15:55:08.937	Thread Group 1-25	HTTP Request	5324	✓	2050	131	5324	320
12	15:55:08.686	Thread Group 1-25	HTTP Request	5576	✓	1990	131	5576	383
13	15:55:08.186	Thread Group 1-23	HTTP Request	6077	✓	1942	131	6077	315
14	15:55:05.936	Thread Group 1-14	HTTP Request	8327	✓	1993	131	8327	311
15	15:55:03.454	Thread Group 1-4	HTTP Request	10819	✓	2045	131	10795	368
16	15:55:10.186	Thread Group 1-31	HTTP Request	4088	✓	1988	131	4085	319
17	15:55:09.937	Thread Group 1-30	HTTP Request	4345	✓	1981	131	4345	314
18	15:55:10.935	Thread Group 1-34	HTTP Request	3348	✓	2046	131	3347	322
19	15:55:09.437	Thread Group 1-28	HTTP Request	4859	✓	1993	131	4855	321
20	15:55:10.685	Thread Group 1-33	HTTP Request	3614	✓	1988	131	3614	332
21	15:55:10.437	Thread Group 1-32	HTTP Request	3865	✓	1994	131	3842	319
22	15:55:09.687	Thread Group 1-29	HTTP Request	4616	✓	1987	131	4609	320
23	15:55:11.685	Thread Group 1-37	HTTP Request	2621	✓	2042	131	2621	328
24	15:55:12.185	Thread Group 1-39	HTTP Request	2123	✓	2055	131	2123	319
25	15:55:11.435	Thread Group 1-36	HTTP Request	2875	✓	1960	131	2870	319
26	15:55:11.187	Thread Group 1-35	HTTP Request	3124	✓	1985	131	3124	329
27	15:55:11.938	Thread Group 1-38	HTTP Request	2399	✓	2047	131	2372	319
28	15:55:12.436	Thread Group 1-40	HTTP Request	1903	✓	1982	131	1902	304
29	15:55:05.436	Thread Group 1-12	HTTP Request	8913	✓	1994	131	8912	308
30	15:55:06.435	Thread Group 1-16	HTTP Request	7914	✓	1985	131	7914	317
31	15:55:05.187	Thread Group 1-11	HTTP Request	9162	✓	2050	131	9162	316
32	15:55:07.436	Thread Group 1-20	HTTP Request	6917	✓	2048	131	6916	326
33	15:55:04.935	Thread Group 1-10	HTTP Request	9444	✓	1948	131	9444	339
34	15:55:03.700	Thread Group 1-5	HTTP Request	10742	✓	2048	131	10652	302
35	15:55:04.437	Thread Group 1-8	HTTP Request	10014	✓	1994	131	9942	308

☐ Scroll automatically? ☐ Child samples? No of Samples 40 Latest Sample 10856 Average 6782 Deviation 2914

Summary Report

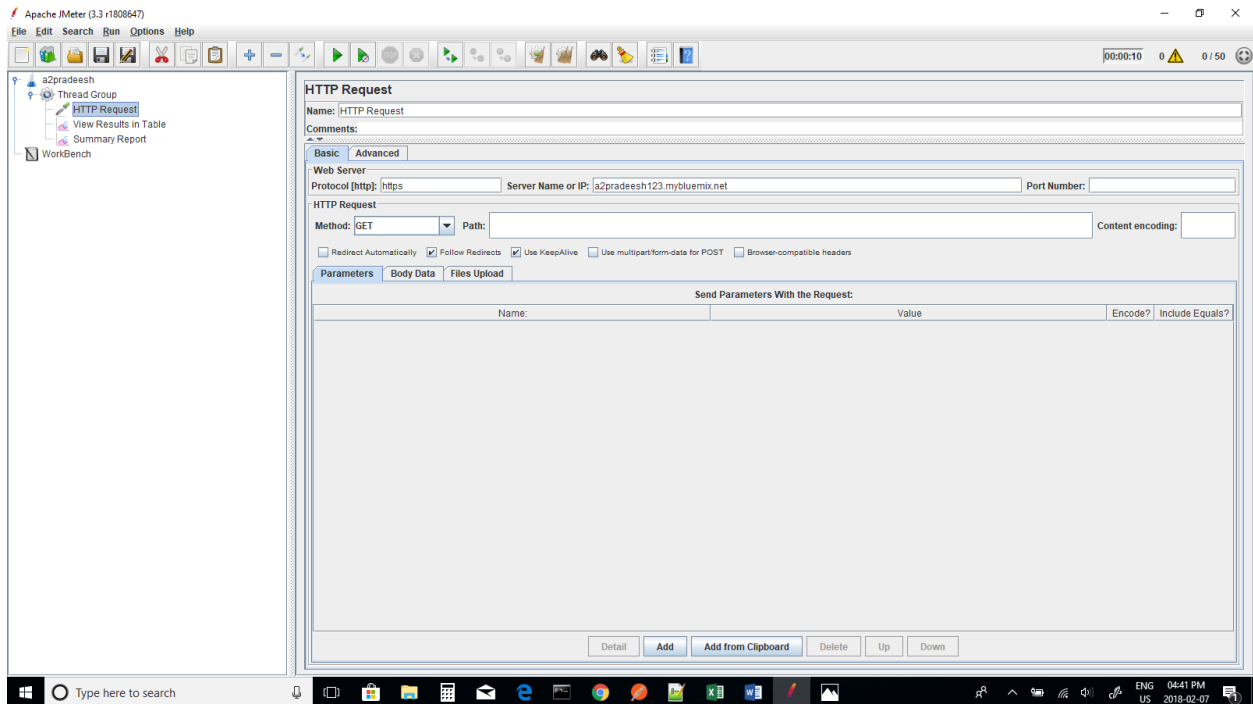
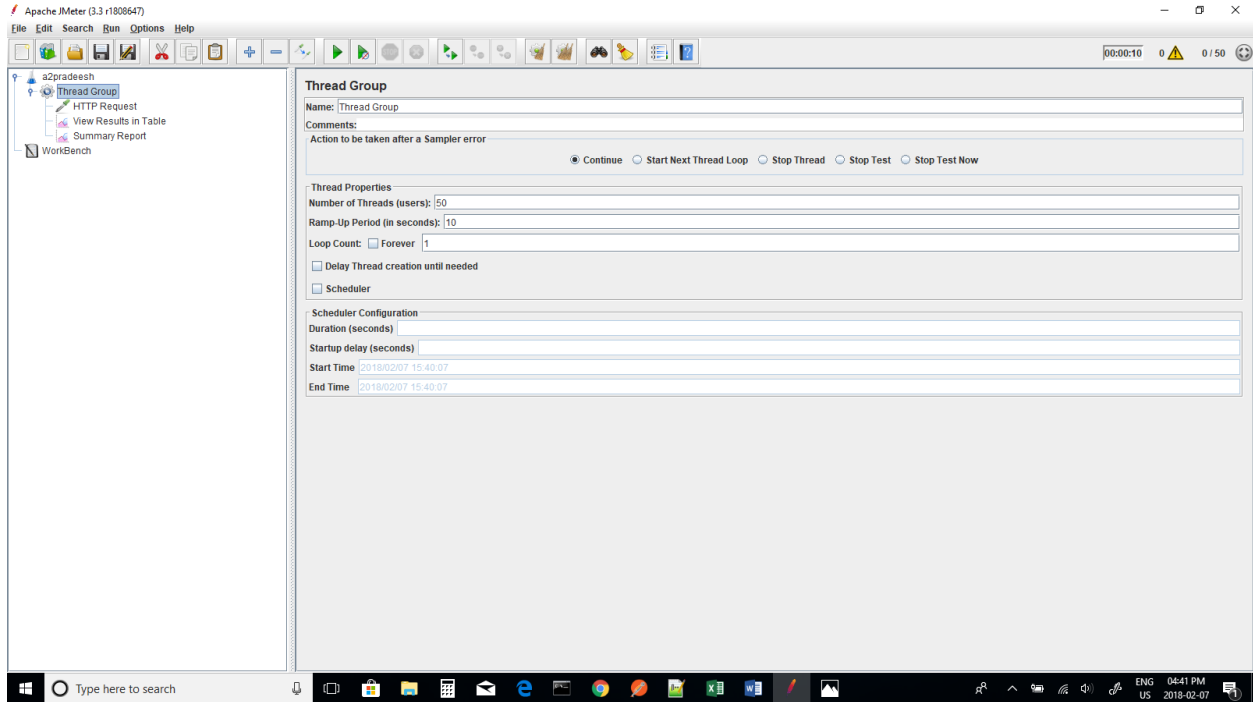
Name: Summary Report
Comments:
Write results to file / Read from file
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes ☐ Configures

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Aug. Bytes
HTTP Request	40	6782	1903	11498	2914.84	0.00%	3.3/sec	6.48	0.42	1998.8
TOTAL	40	6782	1903	11498	2914.84	0.00%	3.3/sec	6.48	0.42	1998.8

☐ Include group name in label? ☒ Save Table Header

Average Response Time for 40 Threads - 6782ms

Changing number of threads – 50



Apache JMeter (3.3 r1806547)

File Edit Search Run Options Help

00:00:10 0 0/50

a2pradeesh

- Thread Group
 - HTTP Request
 - View Results in Table
 - Summary Report
- WorkBench

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Conned Time(ms)
1	16:40:42.958	Thread Group 1-2	HTTP Request	535	✓	1953	131	534	308
2	16:40:42.753	Thread Group 1-1	HTTP Request	749	✓	1951	131	746	347
3	16:40:43.157	Thread Group 1-3	HTTP Request	478	✓	1990	131	476	315
4	16:40:43.358	Thread Group 1-4	HTTP Request	517	✓	1989	131	516	306
5	16:40:43.558	Thread Group 1-5	HTTP Request	424	✓	1998	131	423	290
6	16:40:43.758	Thread Group 1-6	HTTP Request	441	✓	2000	131	435	305
7	16:40:43.959	Thread Group 1-7	HTTP Request	422	✓	1998	131	422	296
8	16:40:44.160	Thread Group 1-8	HTTP Request	421	✓	2022	131	421	303
9	16:40:44.360	Thread Group 1-9	HTTP Request	673	✓	1989	131	672	303
10	16:40:44.560	Thread Group 1-10	HTTP Request	627	✓	2045	131	626	301
11	16:40:44.760	Thread Group 1-11	HTTP Request	679	✓	1990	131	679	312
12	16:40:44.961	Thread Group 1-12	HTTP Request	673	✓	1980	131	672	288
13	16:40:45.161	Thread Group 1-13	HTTP Request	649	✓	1998	131	646	300
14	16:40:45.361	Thread Group 1-14	HTTP Request	761	✓	2044	131	759	299
15	16:40:45.562	Thread Group 1-15	HTTP Request	719	✓	1992	131	714	283
16	16:40:45.762	Thread Group 1-16	HTTP Request	788	✓	1989	131	788	295
17	16:40:45.961	Thread Group 1-17	HTTP Request	764	✓	1969	131	762	302
18	16:40:46.161	Thread Group 1-18	HTTP Request	755	✓	1995	131	753	294
19	16:40:46.362	Thread Group 1-19	HTTP Request	765	✓	1985	131	765	300
20	16:40:46.562	Thread Group 1-20	HTTP Request	718	✓	1961	131	716	298
21	16:40:46.763	Thread Group 1-21	HTTP Request	675	✓	1995	131	673	297
22	16:40:46.963	Thread Group 1-22	HTTP Request	602	✓	1988	131	602	303
23	16:40:47.163	Thread Group 1-23	HTTP Request	501	✓	1955	131	501	281
24	16:40:47.363	Thread Group 1-24	HTTP Request	415	✓	1992	131	414	290
25	16:40:47.563	Thread Group 1-25	HTTP Request	438	✓	1997	131	436	305
26	16:40:47.763	Thread Group 1-26	HTTP Request	428	✓	1950	131	422	290
27	16:40:47.962	Thread Group 1-27	HTTP Request	430	✓	1985	131	429	311
28	16:40:48.163	Thread Group 1-28	HTTP Request	429	✓	1948	131	429	299
29	16:40:48.363	Thread Group 1-29	HTTP Request	421	✓	2054	131	419	298
30	16:40:48.564	Thread Group 1-30	HTTP Request	449	✓	1963	131	447	296
31	16:40:48.760	Thread Group 1-31	HTTP Request	468	✓	1987	131	466	286
32	16:40:48.964	Thread Group 1-32	HTTP Request	458	✓	1950	131	443	304
33	16:40:49.162	Thread Group 1-33	HTTP Request	456	✓	1955	131	446	299
34	16:40:49.362	Thread Group 1-34	HTTP Request	616	✓	1954	131	612	408
35	16:40:49.565	Thread Group 1-35	HTTP Request	483	✓	1987	131	478	312

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 413 Average 538 Deviation 121

Apache JMeter (3.3 r1806547)

File Edit Search Run Options Help

00:00:10 0 0/50

a2pradeesh

- Thread Group
 - HTTP Request
 - View Results in Table
 - Summary Report
- WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	50	538	413	788	121.33	0.00%	4.9/sec	9.48	0.63	1984.7
TOTAL	50	538	413	788	121.33	0.00%	4.9/sec	9.48	0.63	1984.7

☐ Include group name in label? ☒ Save Table Header

Average Response Time for 50 Threads - 538ms

i) URL : a2pradeesh123.mybluemix.net

No of Threads	Average Response Time (in ms)
10	5060

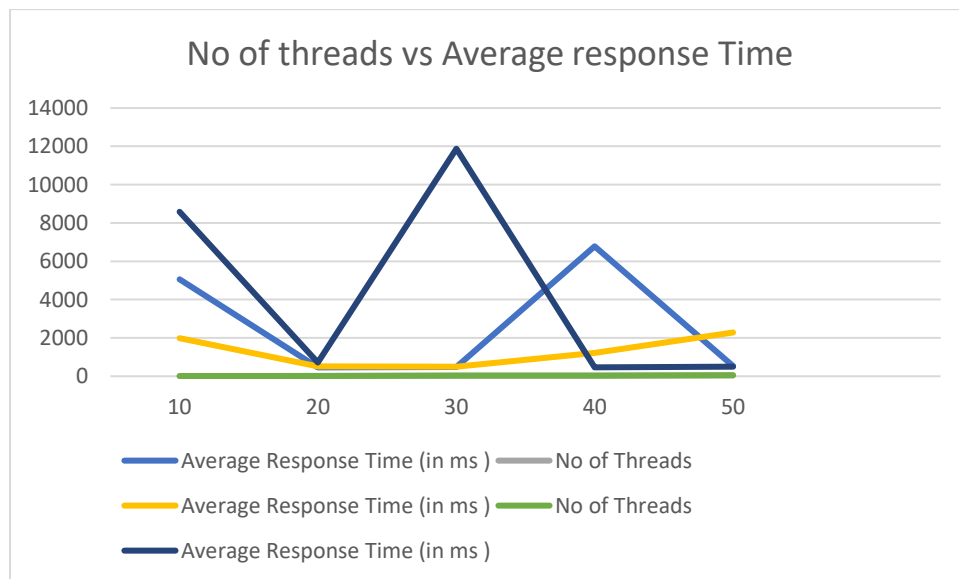
20	463
30	487
40	6782
50	538

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

The screenshot displays the IBM Cloud console interface. At the top, the navigation bar shows the IBM Cloud logo and various links like Catalog, Docs, Support, and Manage. Below this, the main content area is titled 'Cloud Foundry Apps' and shows a table of running applications. The application 'a2pradeesh123' is listed with a route of 'a2pradeesh123.mybluemix.net', 512 MB of memory, and a state of 'Running (1/1)'. Below the apps table, there is a section for 'Cloud Foundry Services' showing three services: 'a2pradeesh' (Db2 Warehouse, Entry plan), 'Compose for Elasticsearch-09' (Compose for Elasticsearch, Standard plan), and 'Db2 Warehouse on Cloud-s6' (Db2 Warehouse, Entry plan). The bottom part of the screenshot shows the 'Application Details' page for 'a2pradeesh123'. This page includes a sidebar with navigation links like Getting started, Overview, Runtime, Connections, Logs, Monitoring, and API Management. The main content area shows the app's runtime details, including the buildpack (Node.js), the number of instances (1), the memory per instance (512 MB), and the total memory allocation (512 MB). It also shows the runtime cost as \$0.00.

IBM Cloud

RESOURCE GROUP: All Resources | REGION: US South | CLOUD FOUNDRY ORG: pradeesh.sivakumar@dal.ca | CLOUD FOUNDRY SPACE: dev

Cloud Foundry Apps 512 MB/2 GB Used

Name	Route	Memory (MB)	State
a2pradeesh123	a2pradeesh123.mybluemix.net	512	Running (1/1)

Cloud Foundry Services 3/10 Used

Name	Service Offering	Plan
a2pradeesh	Db2 Warehouse	Entry
Compose for Elasticsearch-09	Compose for Elasticsearch	Standard
Db2 Warehouse on Cloud-s6	Db2 Warehouse	Entry

Application Details - IBM Cloud

Cloud Foundry apps / a2pradeesh123 Running Visit App URL

Org: pradeesh.sivakumar@dal.ca Location: US South Space: dev

Runtime

- BUILDPACK: SDK for Node.js™
- INSTANCES: 1 (All instances are running, Health is 100%)
- MB MEMORY PER INSTANCE: 512
- TOTAL MB ALLOCATION: 512 (1.6 GB still available)

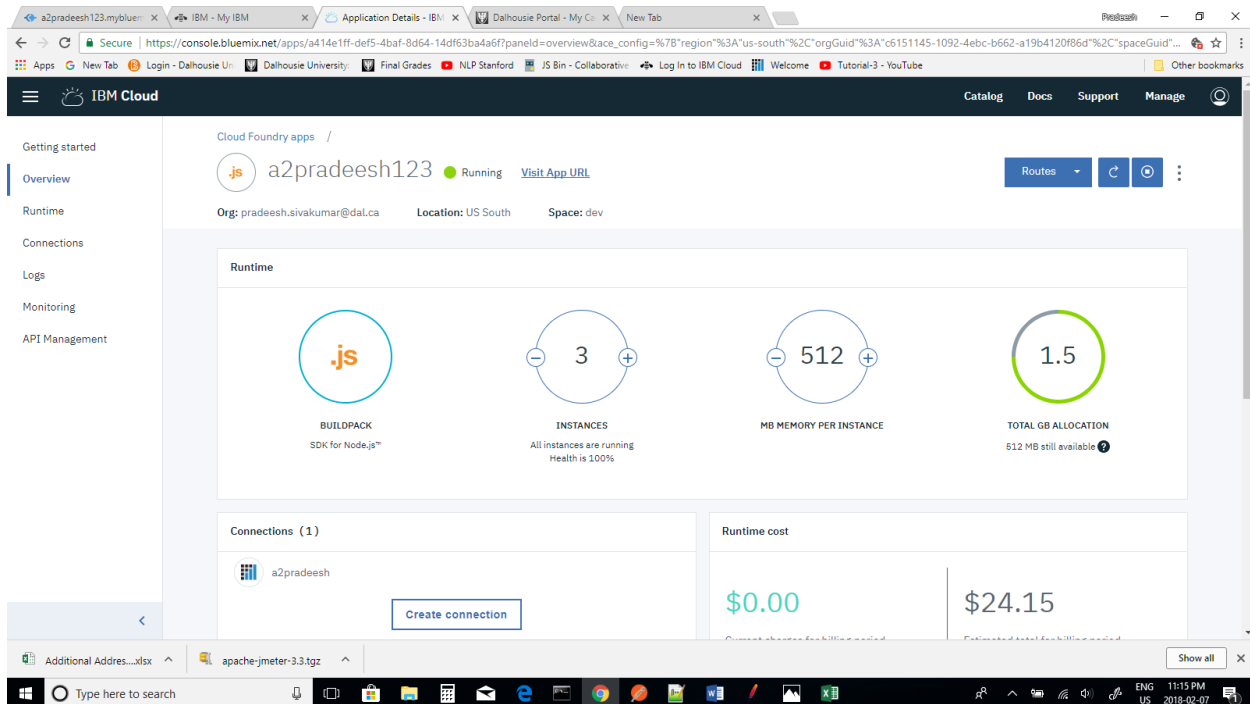
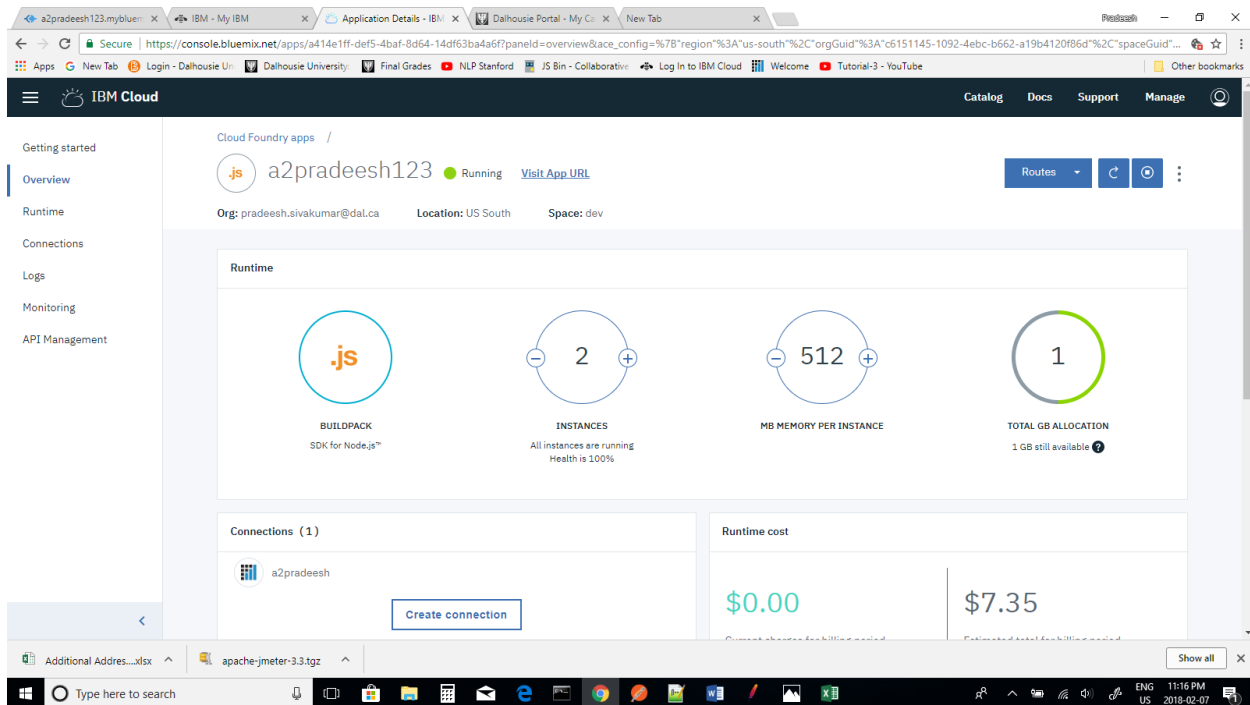
Connections (1)

- a2pradeesh

Runtime cost

\$0.00

\$0.00



Q-1 a2pradeesh123.mybluemix.net

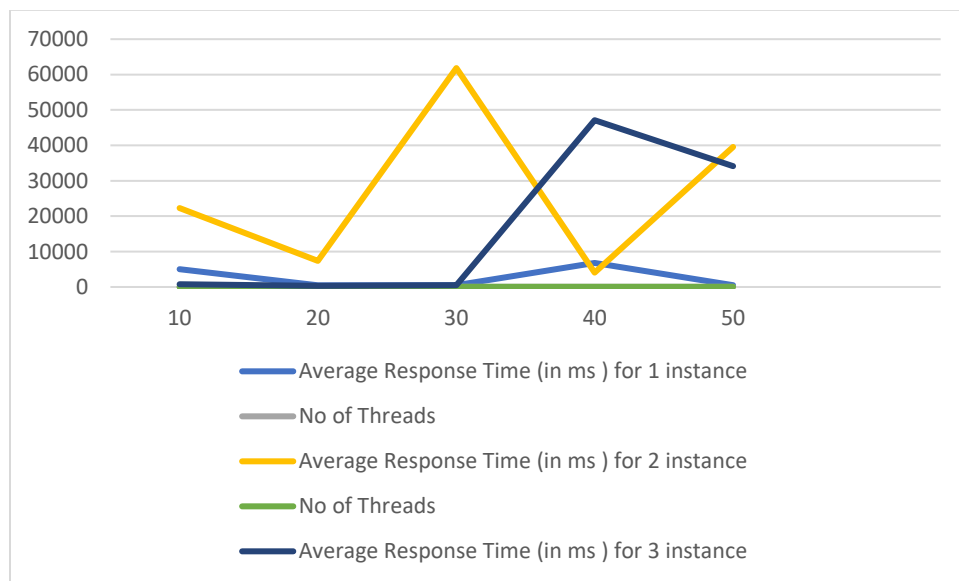
No of Threads	Average Response Time (in ms) for 1 instance
10	5060
20	473
30	488

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



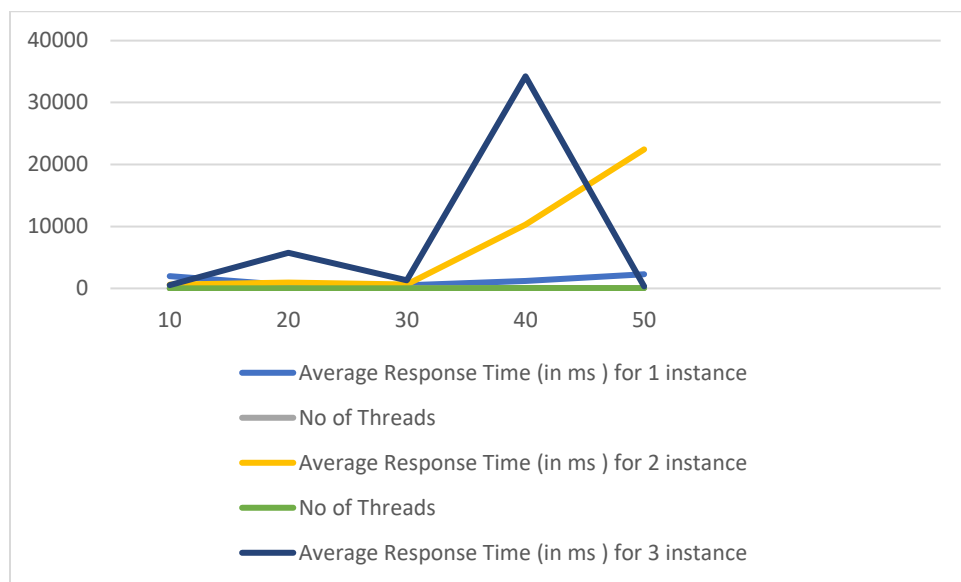
Q-2 a2pradeesh123.mybluemix.net/visitorsbycountry

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

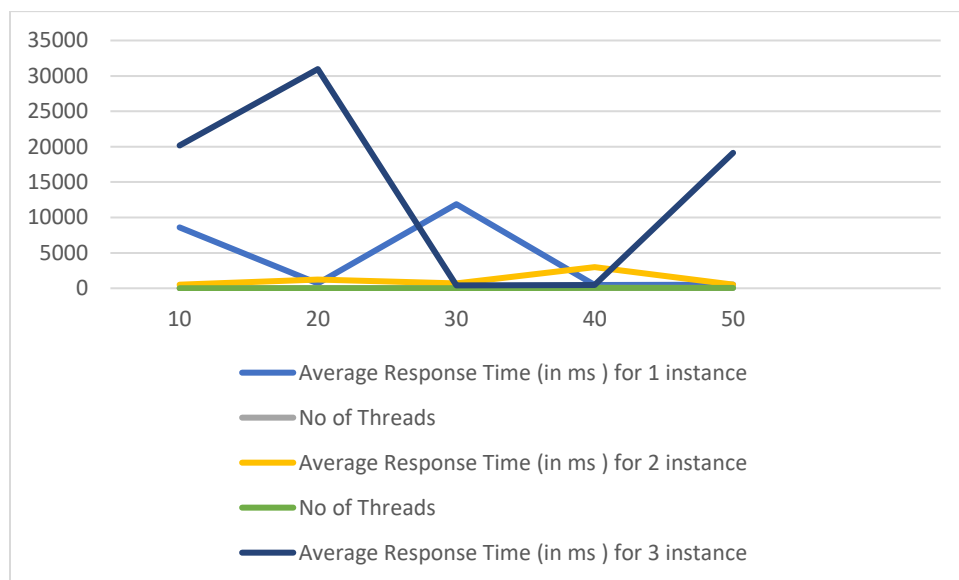
No of Threads	Average Response Time (in ms) for 1 instance
10	8581

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=PDEejbdQVeo>

<http://blog.getpostman.com/2017/11/18/postman-makes-soap-requests-too/>

<http://toolsqa.com/jmeter/thread-group-in-jmeter-test-plan/>

https://www.tutorialspoint.com/jmeter/jmeter_test_plan_elements.htm

<https://console.bluemix.net/docs/tutorials/sql-database.html#sql-database-for-cloud-data>