# Clustering and Performance Testing on Google Cloud Platform (GCP) with Google Compute Engine and Red Hat OpenShift

March 2023

# Table of Contents

# 1. Executive Summary

## 1.1) Introduction

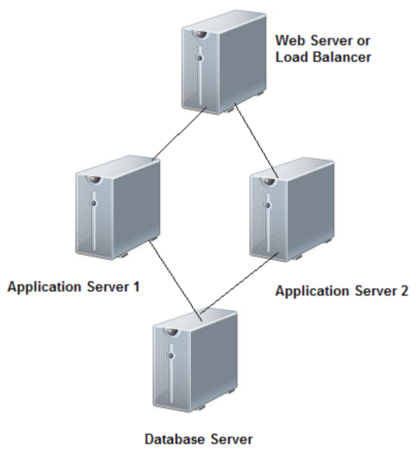
Joget DX 8 is a next generation open source application platform for faster, simpler digital transformation (DX). Joget DX 8 combines the best of business process automation, workflow management and low code application development in a simple, flexible and open platform.

This document is intended to describe and analyze the results of performance tests on a clustered deployment of Joget DX 8 on Google Cloud Platform (GCP).

## 1.2) Test Environment

The tests were conducted on Google Cloud Platform (GCP), specifically using the Google Compute Engine (GCE). GCP offered great flexibility in allowing servers and clients to be created and scaled up as required.

The architecture of the clustered deployment is similar to the following diagram:

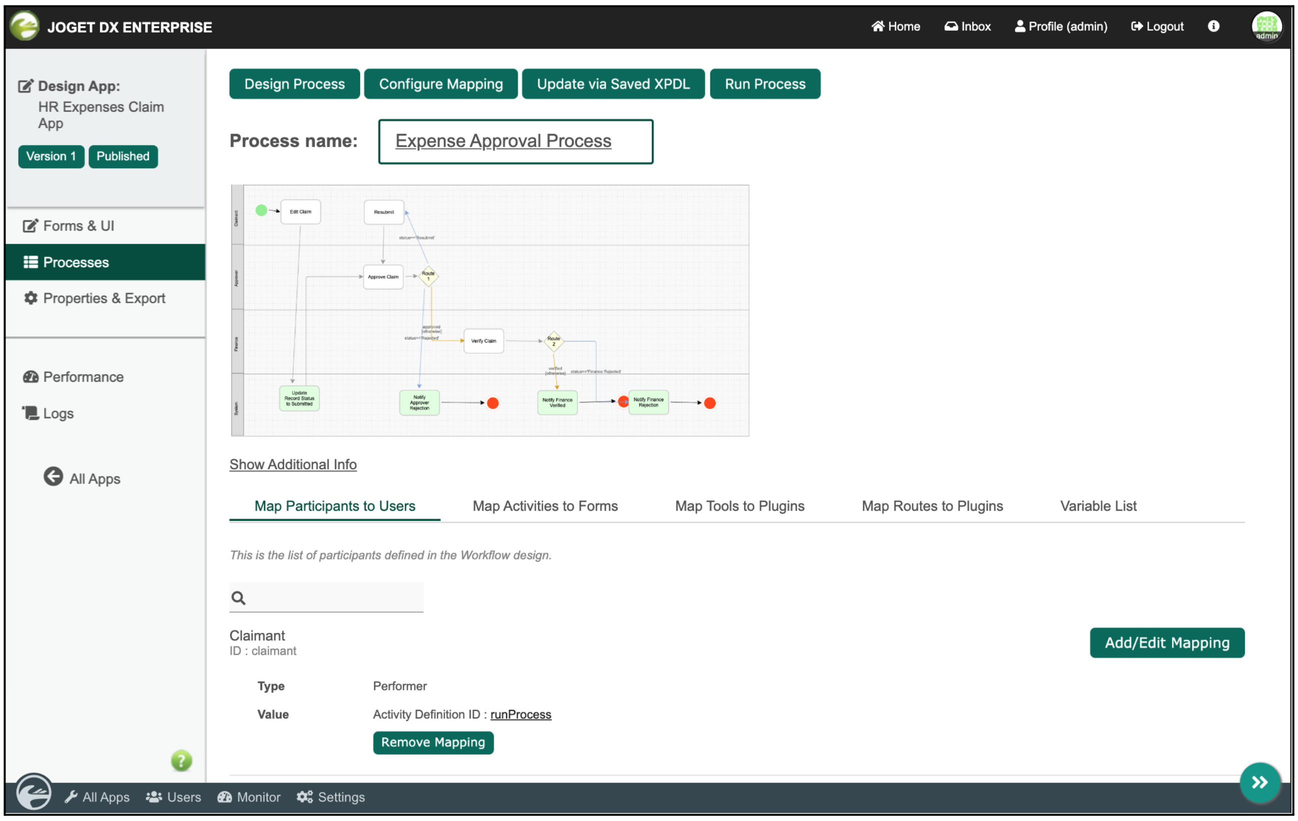


The test was conducted using the following product versions:

**Joget:** Joget DX 8 Cloud Edition 8.0-RC build 917e1b8
**OS:** Ubuntu 22.04 LTS
**Java:** OpenJDK 11.0.17
**Web Application Server:** Apache Tomcat 9.0.71
**Database:** MySQL 8.0.32
**Web Server/Load Balancer:** Nginx Web Server 1.18
**Load Testing Tool:** Apache JMeter 5.5

In addition to the Google Compute Engine, an OpenShift environment was also setup for the load testing:
**Joget Image:** Joget DX 8 on EAP 7 8.0-RC2
**OpenShift Version:** 4.11.27
**Database:** MySQL 8.0.30
**Load Testing Tool:** Apache JMeter 5.5

To establish the baseline performance, a HR Expenses Claim test app was used.



Process name: Expense Approval Process

*This is the list of participants defined in the Workflow design.*

Claimant
ID : claimant

| | |
|---|---|
| Type | Performer |
| Value | Activity Definition ID : runProcess |

Using a think time of 10 seconds with random deviation of 3 seconds, the test script used covers the following app usage:

1. View Login Page
2. Submit Login Form
3. View Expenses Claim Form
4. Get CSRF Token
5. Submit Expenses Claim Form
6. Get CSRF Token
7. Submit Expenses Claim Form to Approver
8. Logout

Tests were carried out for the following (for VM and OpenShift):
VM
1. 100 concurrent users on 1 node (c2d-highcpu-4)
2. 250 concurrent users on 1 node (c2d-highcpu-4)
3. 500 concurrent users on 1 node (c2d-highcpu-4)
4. 750 concurrent users on 1 node (c2d-highcpu-4)
5. 1000 concurrent users on 1 node (c2d-highcpu-4)

6. 1000 concurrent users on 2 nodes (c2d-highcpu-4)
7. 2000 concurrent users on 2 nodes (c2d-highcpu-4)
8. 2000 concurrent users on 3 nodes (c2d-highcpu-4)

<u>OpenShift</u>

1. 100 concurrent users on 2 pods
2. 250 concurrent users on 2 pods
3. 500 concurrent users on 2 pods
4. 750 concurrent users on 2 pods
5. 1000 concurrent users on 2 pods
6. 1000 concurrent users on 4 pods
7. 2000 concurrent users on 4 pods
8. 2000 concurrent users on 6 pods

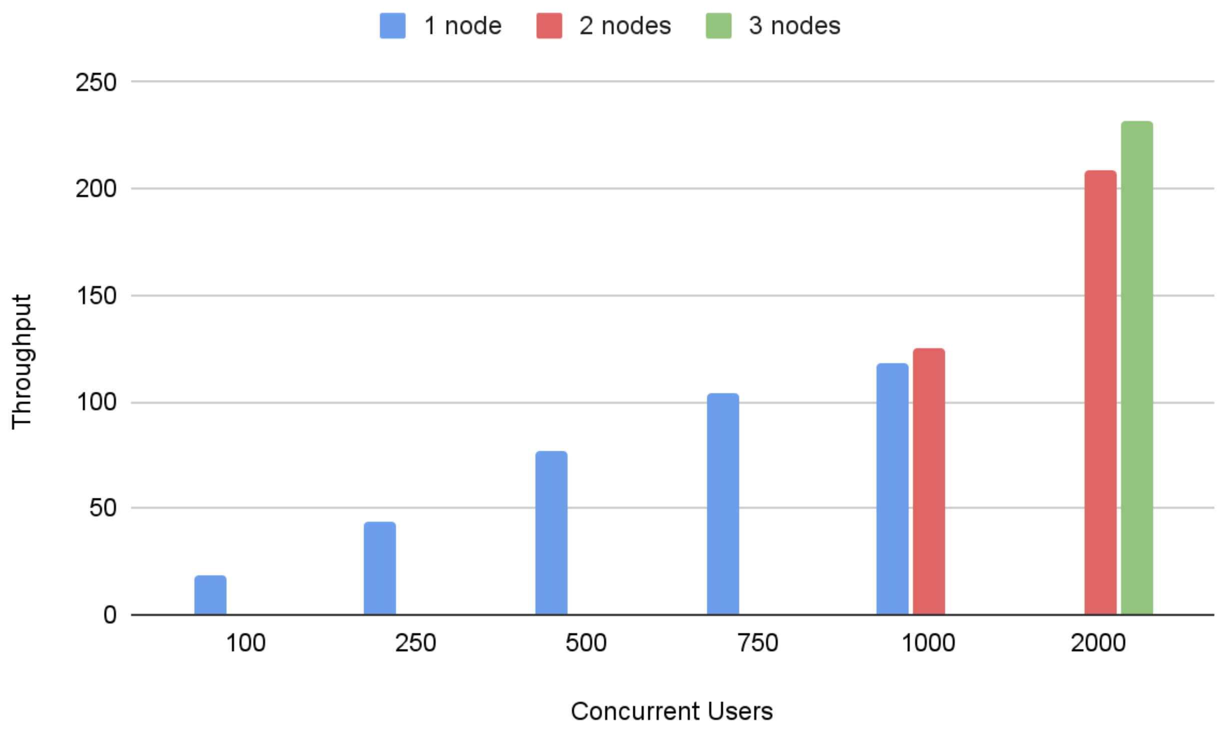
For each test, the JMeter summary results were collected. Once all the results were collected, the throughput (requests per second) and average response times were compared and analyzed.

## 1.3) Summary of Results

### Throughput in GCE VM

The results are summarized in the table and graph below:

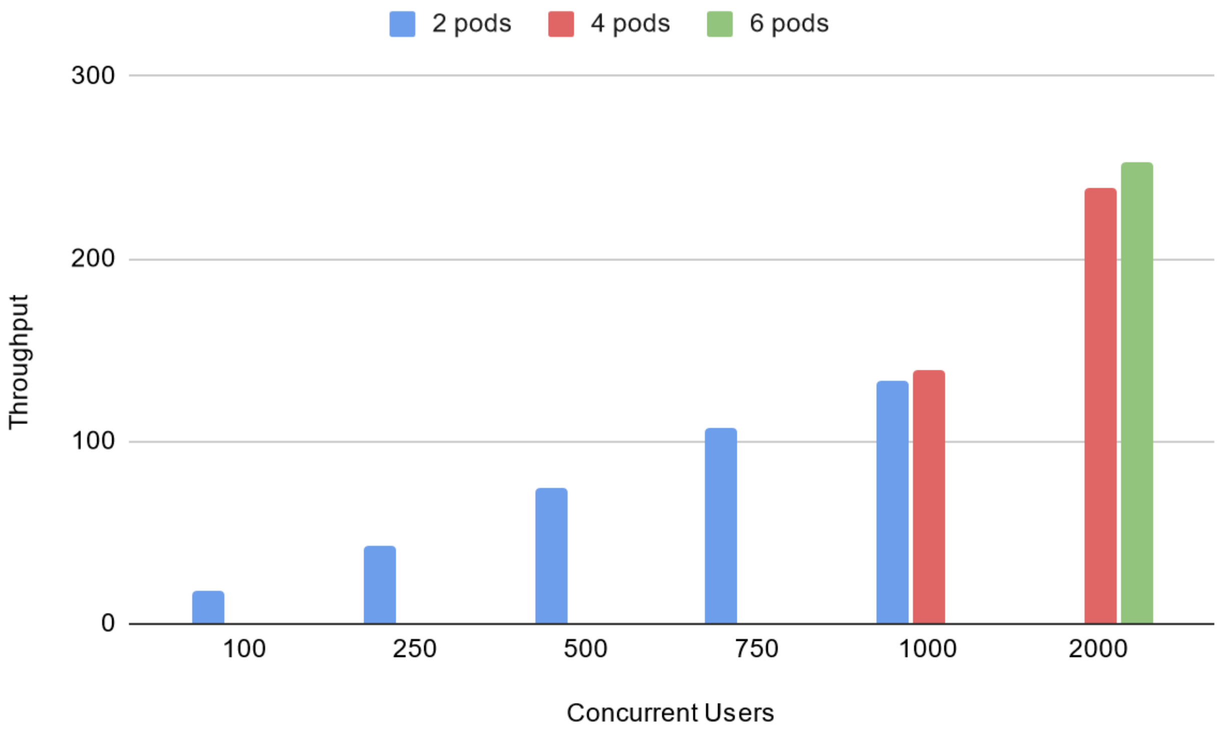
| Throughput (Request/Second) | | | |
|---|---|---|---|
| **Concurrent Users** | **1 node** | **2 nodes** | **3 nodes** |
| 100 | 19.06 | | |
| 250 | 43.46 | | |
| 500 | 77.39 | | |
| 750 | 104.36 | | |
| 1000 | 118.04 | 125.10 | |
| 2000 | | 208.57 | 231.38 |



## Throughput in OpenShift

The results are summarized in the table and graph below:

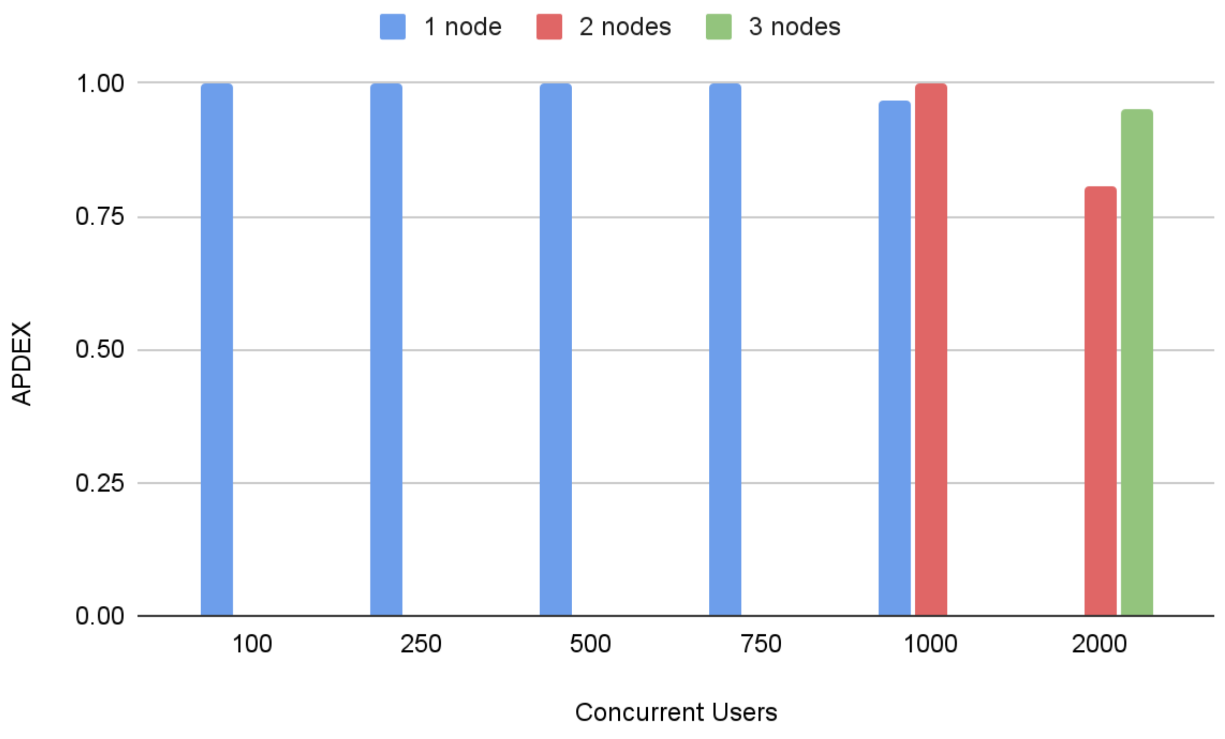
| Throughput (Request/Second) | | | |
|---|---|---|---|
| **Concurrent Users** | **2 pods** | **4 pods** | **6 pods** |
| 100 | 18.69 | | |
| 250 | 42.77 | | |
| 500 | 74.29 | | |
| 750 | 107.02 | | |
| 1000 | 132.82 | 138.91 | |
| 2000 | | 238.56 | 252.92 |



## Application Performance Index (Apdex) in GCE VM

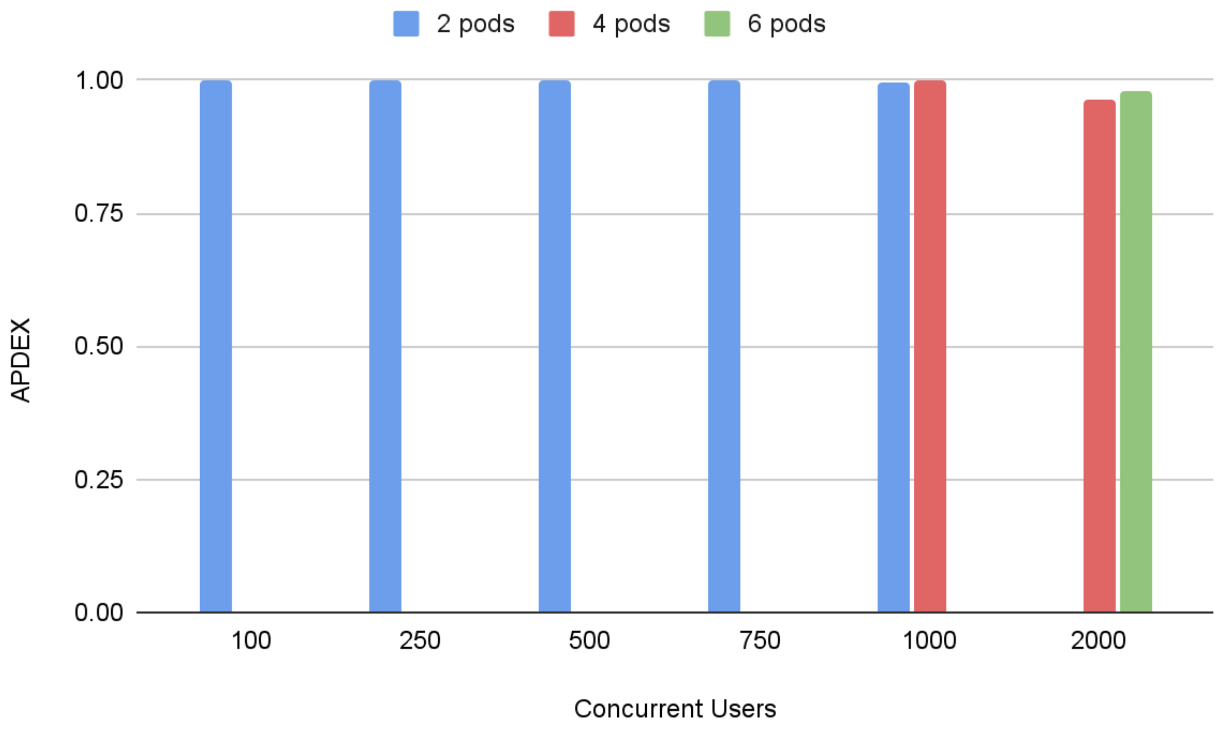
Apdex is an open standard for measuring performance of software applications. The results are summarized in the table and graph below:

| Apdex Score | | | |
|---|---|---|---|
| **Concurrent Users** | **1 node** | **2 nodes** | **3 nodes** |
| 100 | 1.000 | | |
| 250 | 1.000 | | |
| 500 | 1.000 | | |
| 750 | 1.000 | | |
| 1000 | 0.968 | 1.000 | |
| 2000 | | 0.807 | 0.951 |



## Application Performance Index (Apdex) in OpenShift

| Apdex Score | | | |
|---|---|---|---|
| **Concurrent Users** | **2 pods** | **4 pods** | **6 pods** |
| 100 | 1.000 | | |
| 250 | 1.000 | | |
| 500 | 1.000 | | |
| 750 | 0.999 | | |
| 1000 | 0.995 | 0.998 | |
| 2000 | | 0.964 | 0.979 |



## 1.4) Conclusion and Recommendations

From the results it can be seen that for a basic baseline app, a single modestly spec-ed c2d-highcpu-4 server (4 vCPU, 8GB RAM) can handle 500 concurrent users with acceptable response times. The tests also show that scaling out horizontally (adding nodes to a cluster), supports an almost linear increase in concurrent users.

With emphasis on performance optimization at the core platform, Joget DX 8 incurs low overhead when running apps. If there are any specific bottlenecks, it would usually be at the application or plugin level. At the application level, there are various guidelines and best practices that are available in the Performance Optimization and Scalability Tips article in the Joget DX 8 Knowledge Base. Joget DX 8 provides many performance related features such as Application Performance Monitoring and Alerts, Performance Analyzer, Userview Caching, and Governance Health Checks.

For large deployments that support large numbers of concurrent users, it is important that the environment is tuned and optimized e.g. Java VM tuning, app server tuning, database optimization, as per the Deployment Best Practices article.

It is important to note that as Joget is a platform and not directly an end-user app, the scalability and performance would depend on potentially many factors:

1. Total number of users
2. Maximum expected concurrent users
3. Number of apps running on the platform
4. Complexity of each of the apps
5. Amount of data generated in each app
6. Network infrastructure

The recommended deployment architecture would very much depend on the environment and usage. Perhaps some things to be considered:

1. How many total and concurrent users are there? Will this grow in future?
2. In the current environment, is the current infrastructure sufficient for the load? Would it be possible to increase the server resources?
3. If the needs outgrow one server node, it might be time to consider implementing clustering and/or load balancing.
4. Another possible approach could be to partition the apps. Are there specific apps that incur the highest load? Maybe it might be appropriate to separate apps into different servers.
5. Deploy Joget on cloud native platforms like Red Hat OpenShift to take advantage of autoscaling.

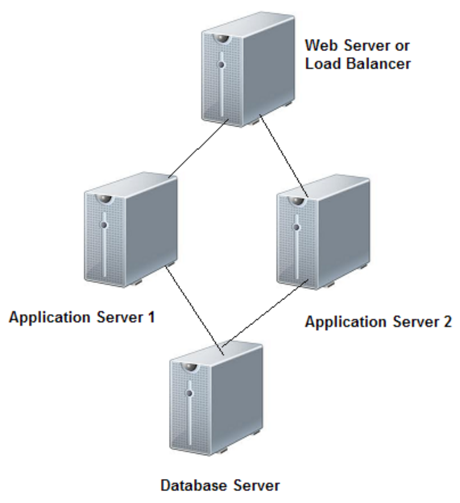
In summary, this report demonstrates the baseline performance of the Joget DX 8 platform for a basic app and shows how horizontal scaling can be used to support larger deployments. Although these results can serve as a base guideline, it is recommended that performance testing and optimisations are performed based on each deployment's unique requirements, environments and usage patterns.

# 2. Test Environment Setup

## 2.1) Test Environment

The tests were conducted on Google Cloud Platform (GCP), specifically using the Google Compute Engine (GCE). GCP offered great flexibility in allowing servers and clients to be created and scaled up as required.

The architecture of the clustered deployment is similar to the following diagram:



### Application Server

**Joget:** Joget DX 8 Cloud Edition 8.0-RC build 917e1b8
**OS:** Ubuntu 22.04 LTS
**Java:** OpenJDK 11.0.17
**Web Application Server:** Apache Tomcat 9.0.71
**GCE Instance:** c2d-highcpu-4

- 4 vCPU (virtual CPUs)
- 8GB RAM
- Java VM Options: -XX:MaxPermSize=256M -Xms4096M -Xmx4096M

### Database Server

**Database:** MySQL 8.0.31
**GCE Instance:** c2d-highcpu-4

- 4 vCPU
- 8GB RAM
- 1500 PIOPS

### Web Server/Load Balancer

**OS:** Ubuntu 22.04 LTS
**Web Server/Load Balancer:** Nginx Web Server 1.18
**GCE Instance:** e2.standard-2:

- 2 vCPU
- 8GB RAM

## OpenShift Configuration

**Joget Image:** Joget DX 8 on EAP 7 8.0-RC2
**OpenShift Version:** 4.11.27
**Master nodes specification:** n2-standard-4
- 4 vCPU
- 16GB RAM

**Master nodes count:** 3 replicas
**Worker nodes specification:** n2-standard-8
- 8 vCPU
- 32GB RAM

**Worker nodes count:** 5 replicas
**Database:** MySQL 8.0.30

## Load Test Configuration

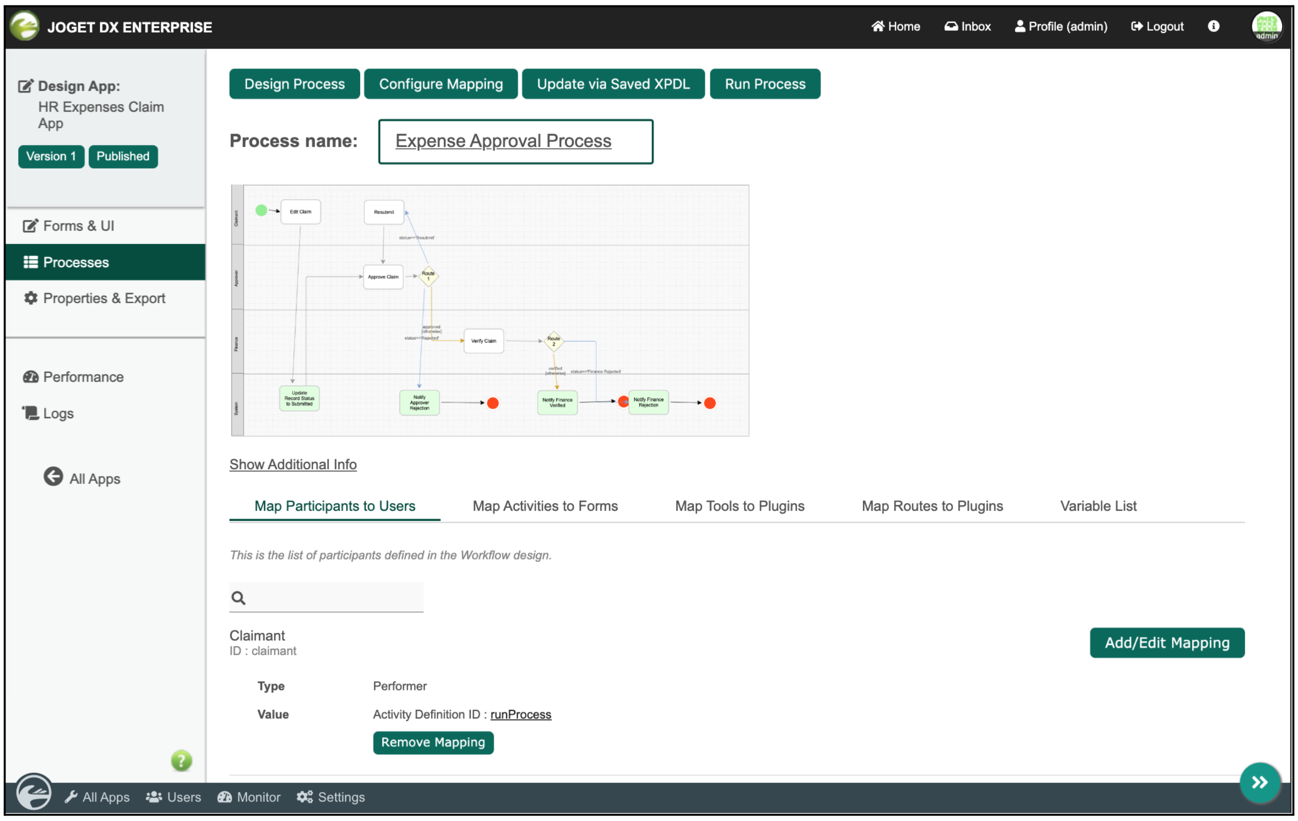
**Load Testing Tool:** Apache JMeter 5.5
**OS:** Ubuntu 22.04 LTS
**GCE Instance:** e2.medium
- 2 vCPU
- 4GB RAM

**Configuration:** 1 master with 2 clients

## Test App

To establish the baseline performance, a HR Expenses Claim test app was used consisting of:

1. 1 process with 4 activities and 4 tools
2. 8 forms
3. 8 datalists
4. 1 userview containing menu pages to run the process and display the datalist and inbox

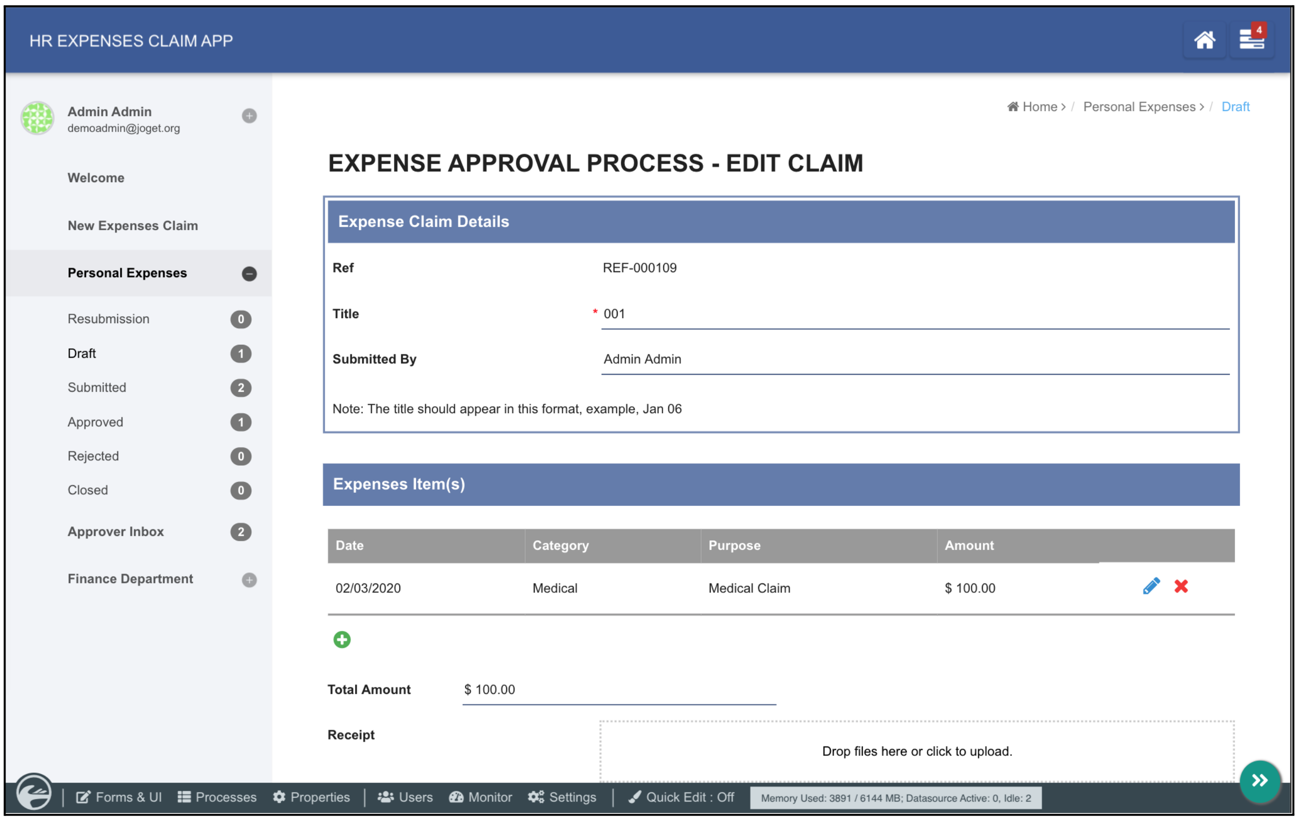


Process name: Expense Approval Process

*This is the list of participants defined in the Workflow design.*

Claimant
ID : claimant

| | |
|---|---|
| Type | Performer |
| Value | Activity Definition ID : runProcess |



EXPENSE APPROVAL PROCESS - EDIT CLAIM

Expense Claim Details

| | |
|---|---|
| Ref | REF-000109 |
| Title | * 001 |
| Submitted By | Admin Admin |

Note: The title should appear in this format, example, Jan 06

Expenses Item(s)

| Date | Category | Purpose | Amount |
|---|---|---|---|
| 02/03/2020 | Medical | Medical Claim | $ 100.00 |

Total Amount $ 100.00

Receipt

## Test Script

The test script used covers the following app usage:

1. View Login Page
2. Submit Login Form
3. View Expenses Claim Form
4. Get CSRF Token
5. Submit Expenses Claim Form
6. Get CSRF Token
7. Submit Expenses Claim Form to Approver
8. Logout

A think time of 10 seconds was used, with random deviation of 3 seconds.

### Test Methodology

The load tests were executed by using the latest Apache JMeter, which provides an automated way of launching, running and collecting JMeter results.

Tests were carried out for the following (for VM and OpenShift):
VM
1. 100 concurrent users on 1 node (c2d-highcpu-4)
2. 250 concurrent users on 1 node (c2d-highcpu-4)
3. 500 concurrent users on 1 node c2d-highcpu-4)
4. 750 concurrent users on 1 node (c2d-highcpu-4)
5. 1000 concurrent users on 1 node (c2d-highcpu-4)
6. 2000 concurrent users on 2 nodes (c2d-highcpu-4)
7. 2000 concurrent users on 3 nodes (c2d-highcpu-4)

OpenShift
9. 100 concurrent users on 2 pods
10. 250 concurrent users on 2 pods
11. 500 concurrent users on 2 pods
12. 750 concurrent users on 2 pods
13. 1000 concurrent users on 2 pods
14. 1000 concurrent users on 4 pods
15. 2000 concurrent users on 4 pods
16. 2000 concurrent users on 6 pods

For each test, the JMeter summary results were collected. Once all the results were collected, the throughput (requests per second) and average response times were compared and analyzed.

## 2.2) Setup the Joget Server Cluster

The following are brief descriptions of the steps used to setup the server instances:

### Launch GCE Instance

From the Google Cloud console, launch the appropriate GCE instance running on Ubuntu 22.04.

### Install Java

```
sudo apt-get install openjdk-11-jdk
```

## Install Joget

Download Linux tar.gz bundle
Extract into /opt/joget
Run setup.sh and configure to the database

## Install Nginx

For the load balancer, install Nginx web server

```
sudo apt-get install nginx
```

## Configure Load Balancer

For the load balancer, another section in /etc/nginx/nginx.conf has been added

```
      underscores_in_headers on;
       upstream joget {
                   hash $remote_addr;
                   server joget-server-1:8080 weight=1;
                   server joget-server-2:8080 weight=1;
}
```

Increase the maximum number of open files by adding

```
      fs.file-max=100000
```

into /etc/sysctl.conf

Increase the limit on the maximum number of open files for worker processes in Nginx by adding

```
      worker_rlimit_nofile 30000;
```

into /etc/nginx/nginx.conf

Create a new file in /etc/nginx/sites-available, named joget

```
sudo vi /etc/nginx/sites-available/joget
```

Add the contents

```
server {
       listen 80;
      server_name 10.128.0.21;
       underscores_in_headers on;
       client_body_buffer_size 10K;
       client_header_buffer_size 1k;
       client_max_body_size 8m;
       large_client_header_buffers 4 16k;
       access_log /var/log/nginx/joget.access.log;

       location /jw/web/applog/ {
             proxy_pass http://joget/jw/web/applog/;
             proxy_set_header Host $http_host;
             proxy_set_header X-Forwarded-Host $host;
             proxy_set_header X-Forwarded-Server $host;
             proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
             proxy_set_header Cookie $http_cookie;
             proxy_http_version 1.1;
             proxy_set_header Upgrade $http_upgrade;
             proxy_set_header Connection "upgrade";
             proxy_buffering off;
      }
       location / {
                   proxy_pass http://joget;
                   proxy_redirect off;
                   proxy_pass_header X-CSRF-TOKEN;
                   proxy_set_header Host $host;
                   proxy_set_header X-Forwarded-Server $host;
                   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                   proxy_set_header X-NginX-Proxy true;
                   proxy_set_header Cookie $http_cookie;
                   proxy_read_timeout 3000;
                   proxy_buffers 32 4m;
                   proxy_busy_buffers_size 25m;
                   proxy_buffer_size 512k;
                   proxy_ignore_headers "Cache-Control" "Expires";
                   proxy_max_temp_file_size 0;
                   client_max_body_size 1024m;
                   client_body_buffer_size 4m;
                   proxy_connect_timeout 3000;
                   proxy_headers_hash_max_size 512;
                   proxy_send_timeout 3000;
                   proxy_intercept_errors off;
                   proxy_http_version 1.1;
                   proxy_set_header   Connection "upgrade";

       }

}
```

Enable the new site and reload Nginx

```
sudo ln -s /etc/nginx/sites-available/joget /etc/nginx/sites-enabled/joget
sudo nginx -t
sudo nginx -s reload
```

## Configure Shared Database

To install a MySQL database

sudo apt-get install mysql-server

Configure database permissions

```
mysql -u root
```

Run the following MySQL commands to create joget user and then grant permissions to user **joget**

```
CREATE USER 'joget'@'%' IDENTIFIED WITH mysql_native_password BY 'joget';
GRANT ALL PRIVILEGES ON jwdb.* TO 'joget'@'%';
flush privileges;
quit
```

Configure MySQL to listen to database connections from remote hosts. Edit the my.cnf file with your favourite editor

```
sudo vim mysqld.conf.d/mysqld.cnf
```

Comment away the bind-address directive by adding a # in front of the line

```
#bind-address = 127.0.0.1
```

Restart MySQL

```
sudo systemctl restart mysql
```

Test remote connections. In the application server, test a remote connection to the database server **database_host**

```
mysql -h database_host -u joget -p
```

## Configure Shared File Directory

Install NFS (for sharing file system)

```
sudo apt-get install portmap nfs-kernel-server nfs-common
```

Create new directory /opt/joget/shared/wflow to mount the shared directory and set the directory permissions

```
sudo mkdir -p /opt/joget/shared/wflow
sudo chmod 777 /opt/joget/shared/wflow
```

Mount the shared directory.

```
sudo mount -t nfs joget-server:/export/wflow /opt/joget/shared/wflow
```

Test read-write permissions to confirm that the directory sharing works.

```
echo test123 > /opt/joget/shared/wflow/test.txt
```

## Optimize Java

Set appropriate Java heap settings e.g.

```
export JAVA_OPTS="-XX:MaxPermSize=256m -Xms4096M -Xmx4096M
-Djoget.home=$JOGET_HOME -Dwflow.home=/opt/joget/shared/wflow
```

```
-javaagent:/opt/joget/shared/wflow/wflow-cluster.jar
-javaagent:$JOGET_HOME/lib/aspectjweaver-1.9.7.jar
-javaagent:/opt/joget/lib/glowroot/glowroot.jar"
```

## Optimize Tomcat

Edit server.xml and add connectors, especially maxThreads

```
<Connector port="8080" protocol="HTTP/1.1"
       connectionTimeout="20000"
       maxThreads="2000"
       compression="on"
       useSendfile="false"
       redirectPort="8443" />
```

Configure Linux ulimit Configuration:

```
ulimit -n 4096
```

## Tomcat Session Persistence

To simulate an actual environment, in the event the load balancer does not support sticky sessions, we can implement Persistent Manager in Tomcat, which has the capability to swap active (but idle) sessions out to a persistent storage mechanism, as well as to save all sessions across a normal restart of Tomcat.

We need to set `org.apache.catalina.session.StandardSession.ACTIVITY_CHECK=true` in /opt/joget/apache-tomcat-9.0.71/conf/catalina.properties to ensure the persistent manager works correctly.

In this testing we use a JDBC Based Store to save sessions in individual rows of a preconfigured table in a database that is accessed via a JDBC driver. Create a database named tomcat and table with the following SQL queries:

```
create database tomcat;
grant all privileges on tomcat.* to 'tomcat'@'%' identified by 'tomcat';
use tomcat;
create table tomcat_sessions (
  session_id     varchar(100) not null primary key,
  valid_session  char(1) not null,
  max_inactive   int not null,
  last_access    bigint not null,
  app_name       varchar(255),
  session_data   mediumblob,
  KEY kapp_name(app_name)
);
```

In order for the JDBC Based Store to successfully connect to the database, we need to place the JAR file containing MySQL JDBC driver into /opt/joget/apache-tomcat-9.0.71/lib directory.

Last but not least, add the following content into /opt/joget/apache-tomcat-9.0.71/conf/context.xml

```
<Loader loaderClass="org.apache.catalina.loader.ParallelWebappClassLoader" />
    <Resources cachingAllowed="true" cacheMaxSize="100000" />
```

```
      <Valve className="org.apache.catalina.valves.PersistentValve"/>
    <Manager className="org.apache.catalina.session.PersistentManager"
       maxIdleBackup="0"
       maxIdleSwap="0"
       minIdleSwap="0"
       persistAuthentication='true'
       processExpiresFrequency="6"
       saveOnRestart='true'>
    <Store className="org.apache.catalina.session.JDBCStore"
connectionURL="jdbc:mysql://joget-db-server-ip/tomcat?user=tomcat&amp;password=tom
cat"
       driverName="com.mysql.jdbc.Driver"
       sessionAppCol="app_name"
       sessionDataCol="session_data"
       sessionIdCol="session_id"
       sessionLastAccessedCol="last_access"
       sessionMaxInactiveCol="max_inactive"
       sessionTable="tomcat_sessions"
       sessionValidCol="valid_session"/>  </Manager>
```

### Optimize MySQL

Configure /etc/mysql/mysqld.conf.d/mysqld.cnf containing the following and restart MySQL

```
character-set-server=utf8
collation-server=utf8_unicode_ci

# Add innodb buffer pool size config
innodb_buffer_pool_size = 6000M
```

## 2.3) Add a New Joget Node

When adding a new node to the server cluster, the following steps are taken (in this sample the new node hostname will be joget-server3):

### Launch New Joget Node

Launch new instance of GCE and follow the steps to configure Joget as above

### Configure New Joget Node

SSH into node

Edit /etc/hosts to add node hostname, and modify joget-server IP if necessary e.g.

```
127.0.0.1 joget-server3
172.31.30.203 joget-server
```

Edit /etc/hostname to modify node hostname e.g.

```
joget-server3
```

Modify hostname e.g.

```
sudo hostname joget-server3
```

Remount NFS share (if joget-server shared directory IP was modified)

Configure Tomcat for clustering by editing server.xml. Add **jvmRoute="node03"** to the **Engine** tag.

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="node03">
```

Restart Tomcat.

### Add to Load Balancer

In the load balancer, edit /etc/nginx/nginx.conf to add the BalancerMember node e.g.

```
      underscores_in_headers on;
        upstream joget {
                    hash $remote_addr;
                    server joget-server-1:8080 weight=1;
                    server joget-server-2:8080 weight=1;
                    server joget-server-3:8080 weight=1;
}
```

then reload/restart Nginx.

## 2.4) Setup the Joget OpenShift environment

### Joget Deployment

The deployment yaml;

```
---
kind: PersistentVolume
apiVersion: v1
metadata:
  name: joget-pv-dx8loadtest
spec:
  storageClassName: openshift-nfs
  capacity:
    storage: 20Gi
  accessModes:
    - ReadWriteMany
  nfs:
    server : {NFS-Server-IP}
    path: /wflow
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: joget-pv-dx8loadtest-claim
spec:
  storageClassName: openshift-nfs
  accessModes:
    - ReadWriteMany
  Resources:
    requests:
    storage: 20Gi
---
apiVersion: v1
kind: Service
metadata:
  name: joget
  labels:
    app: joget
spec:
  ports:
  - port: 8080
  selector:
    app: joget
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: joget-ping
  labels:
    app: joget
spec:
  ports:
  - name: joget-ping
    port: 8888
  selector:
    app: joget
  clusterIP: None
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: joget
spec:
  selector:
    matchLabels:
      app: joget
  replicas: 4
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: joget
    spec:
      containers:
      - image: quay.io/joget/joget-dx8-eap7:8.0-RC2
```

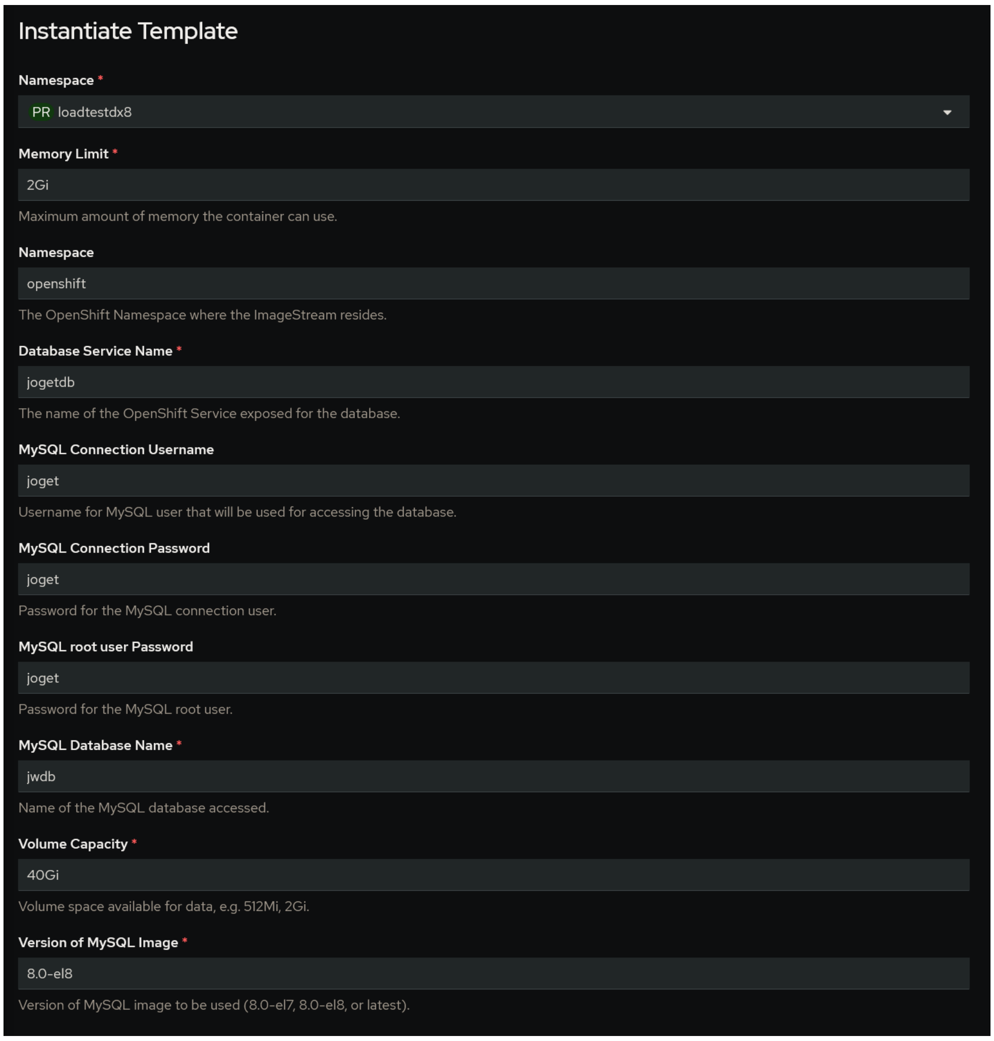
```yaml
          name: joget
        env:
        - name: JGROUPS_PING_PROTOCOL
          value: openshift.DNS_PING
        - name: OPENSHIFT_DNS_PING_SERVICE_NAME
          value: joget-ping
        - name: OPENSHIFT_DNS_PING_SERVICE_PORT
          value: "8888"
        - name: CACHE_NAME
          value: http-session-cache
        ports:
        - containerPort: 8080
          name: joget
        volumeMounts:
        - name: joget-persistent-storage
          mountPath: /home/jboss/wflow
        startupProbe:
          httpGet:
            path: /jw/web/console
            port: 8080
            scheme: HTTP
          periodSeconds: 5
          timeoutSeconds: 1
          failureThreshold: 120
        livenessProbe:
          httpGet:
            path: /jw/web/console
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 300
          timeoutSeconds: 5
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 20
        readinessProbe:
          httpGet:
            path: /jw/web/console
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 20
        terminationGracePeriodSeconds: 120
        volumes:
        - name: joget-persistent-storage
          persistentVolumeClaim:
            claimName: joget-pv-dx8loadtest-claim
---
apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRoleBinding
metadata:
  name: joget-dx8loadtest-clusterrolebinding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: view
subjects:
  - kind: ServiceAccount
    name: default
    namespace: loadtestdx8
```

Configured the *GC_MAX_METASPACE_SIZE* environment variable to 1000.

## Database Configuration

Using Instantiate Template feature from OpenShift Console ;



Instantiate Template

| | |
|---|---|
| Namespace * | PR loadtestdx8 |
| Memory Limit * | 2Gi<br>Maximum amount of memory the container can use. |
| Namespace | openshift<br>The OpenShift Namespace where the ImageStream resides. |
| Database Service Name * | jogetdb<br>The name of the OpenShift Service exposed for the database. |
| MySQL Connection Username | joget<br>Username for MySQL user that will be used for accessing the database. |
| MySQL Connection Password | joget<br>Password for the MySQL connection user. |
| MySQL root user Password | joget<br>Password for the MySQL root user. |
| MySQL Database Name * | jwdb<br>Name of the MySQL database accessed. |
| Volume Capacity * | 40Gi<br>Volume space available for data, e.g. 512Mi, 2Gi. |
| Version of MySQL Image * | 8.0-el8<br>Version of MySQL image to be used (8.0-el7, 8.0-el8, or latest). |

Configured the *DB MYSQL_MAX_CONNECTIONS* environment variable value to 5000.

### Route Configuration

Added annotation for *haproxy.router.openshift.io/balance* and *haproxy.router.openshift.io/timeout.*

The yaml for the route;

```
kind: Route
apiVersion: route.openshift.io/v1
metadata:
  name: jogetloadtest
  namespace: loadtestdx8
…
  labels:
    app: joget
  annotations:
    haproxy.router.openshift.io/balance: leastconn
    haproxy.router.openshift.io/timeout: 60s
    openshift.io/host.generated: 'true'
  managedFields:
…{}
spec:
  host: jogetloadtest-loadtestdx8.apps.openshift.joget.ai
  path: /jw
  to:
    kind: Service
    name: joget
    weight: 100
  port:
    targetPort: 8080
  wildcardPolicy: None
  tls: null
…
```

## 2.5) Setup Load Testing Clients

### Create a folder to store JMeter test file, results and reports

mkdir -p ~/load_tests/reports

### Download & Configure JMeter master

Download JMeter from from https://jmeter.apache.org/

Extract the installer and edit user.properties file

```
vi apache-jmeter-5.5/bin/user.properties
```

change the value of APDEX satisfied and tolerated threshold.

```
# Change this parameter if you want to override the APDEX satisfaction threshold.
jmeter.reportgenerator.apdex_satisfied_threshold=5000
```

```
# Change this parameter if you want to override the APDEX tolerance threshold.
jmeter.reportgenerator.apdex_tolerated_threshold=10000
```

edit the jmeter.properties file to add the IP of the clients into the remote hosts eg.

```
remote_hosts=10.128.0.26,10.128.0.27
```

## Running the multi Jmeter testing (distributed load testing)

On the 2 jmeter clients system, run the jmeter-server

```
cd apache-jmeter-5.5/bin

./jmeter-server
```

## Run JMeter load test

copy the jmeter test file and run jmeter

```
apache-jmeter-5.5/bin/jmeter.sh -n -t loadtest-expenses.jmx -l ~/tests/result.csv
-e -o ~/load_tests/reports/ -R 10.128.0.26,10.128.0.27
```

# 3. Performance Test Results

## GCE Virtual Machine

### 3.1) 100 users 1 node

Application Server: 1 c2d.highcpu-4 node

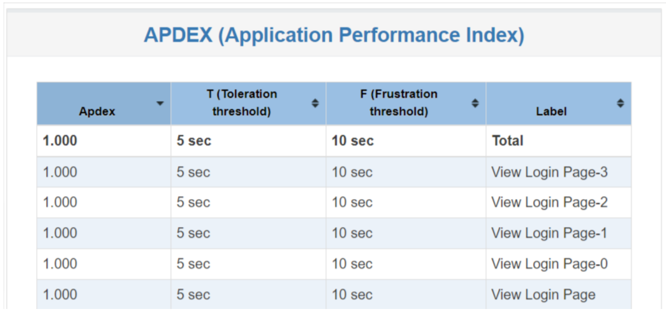
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 50 users in 2 e2.medium instances

Concurrent Users: 100 users

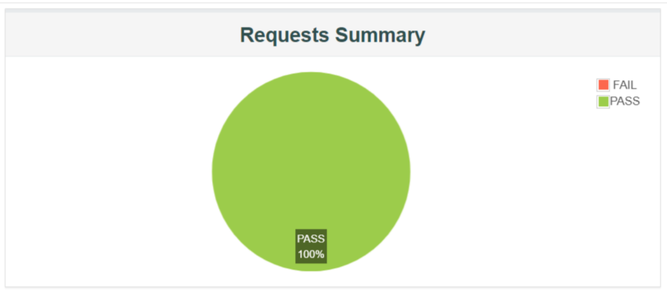
Ramp-up Time: 5s ramp-up time for each Jmeter client

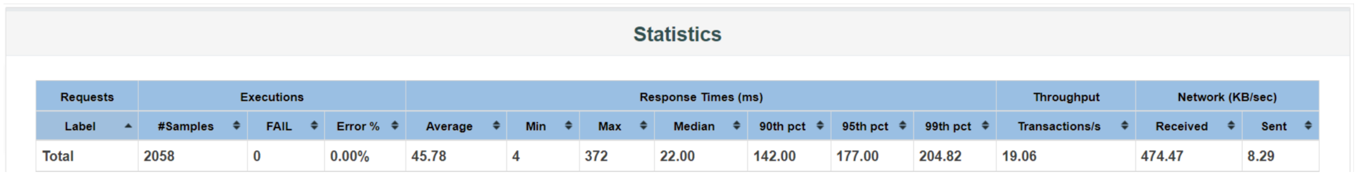
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 2058 | 0 | 0.00% | 45.78 | 4 | 372 | 22.00 | 142.00 | 177.00 | 204.82 | 19.06 | 474.47 | 8.29 |

## 3.2) 250 users 1 node

Application Server: 1 c2d.highcpu-4 node

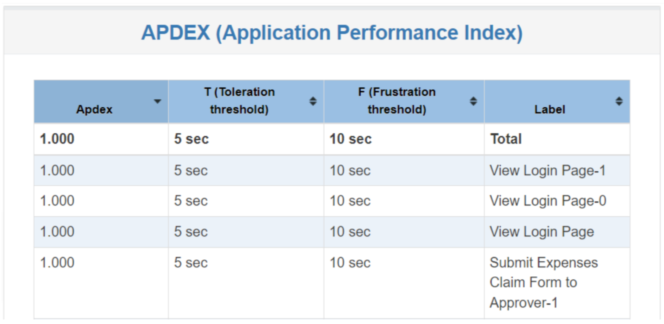
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 125 users in 2 e2.medium instances

Concurrent Users: 250 users

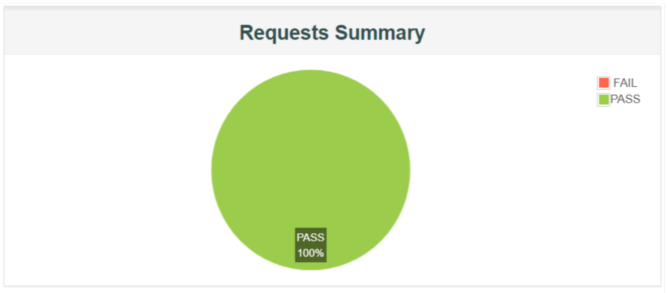
Ramp-up Time: 10s ramp-up time for each Jmeter client

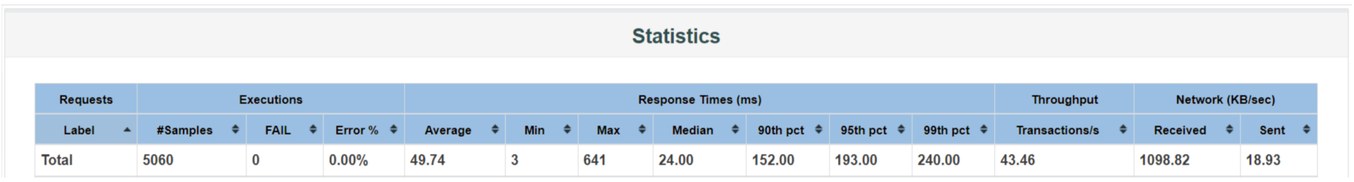
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 5060 | 0 | 0.00% | 49.74 | 3 | 641 | 24.00 | 152.00 | 193.00 | 240.00 | 43.46 | 1098.82 | 18.93 |

## 3.3) 500 users 1 node

Application Server: 1 c2d.highcpu-4 node

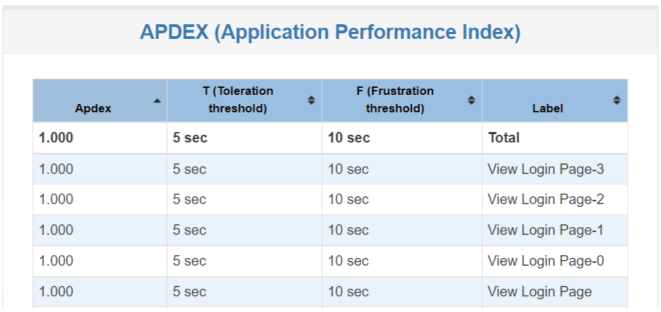
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 250 users in 2 e2.medium instances

Concurrent Users: 500 users

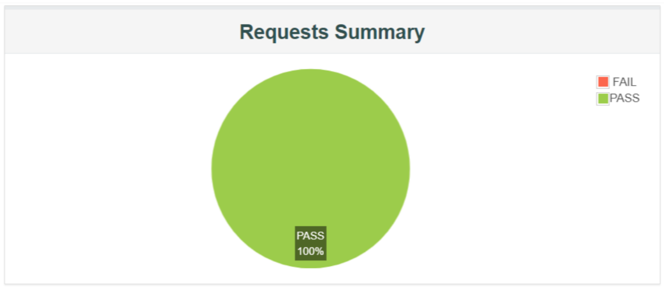
Ramp-up Time: 25s ramp-up time for each Jmeter client

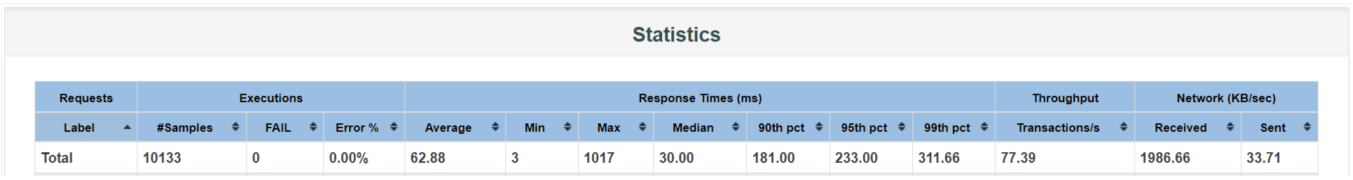
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 10133 | 0 | 0.00% | 62.88 | 3 | 1017 | 30.00 | 181.00 | 233.00 | 311.66 | 77.39 | 1986.66 | 33.71 |

## 3.4) 750 users 1 node

Application Server: 1 c2d.highcpu-4 node

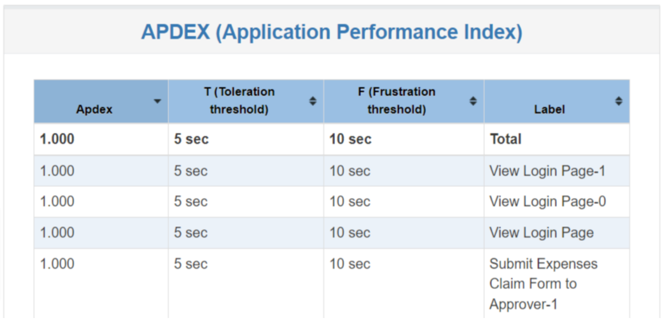
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 375 users in 2 e2.medium instances

Concurrent Users: 750 users

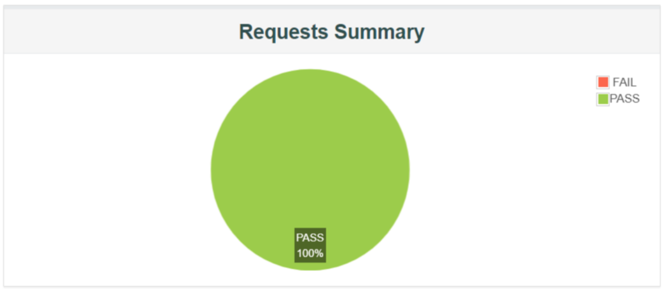
Ramp-up Time: 35s ramp-up time for each Jmeter client

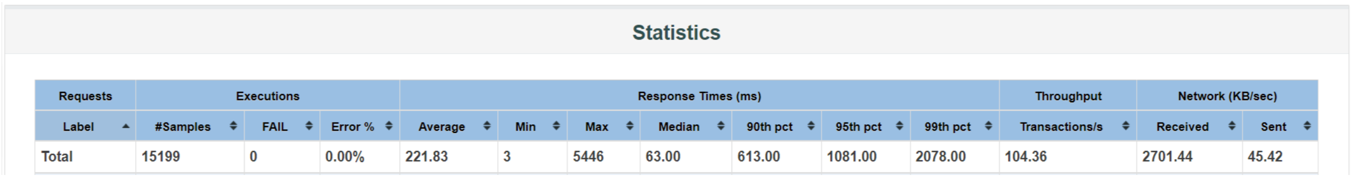
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |


Requests Summary



Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 15199 | 0 | 0.00% | 221.83 | 3 | 5446 | 63.00 | 613.00 | 1081.00 | 2078.00 | 104.36 | 2701.44 | 45.42 |

## 3.5) 1000 users 1 node

Application Server: 1 c2d.highcpu-4 node

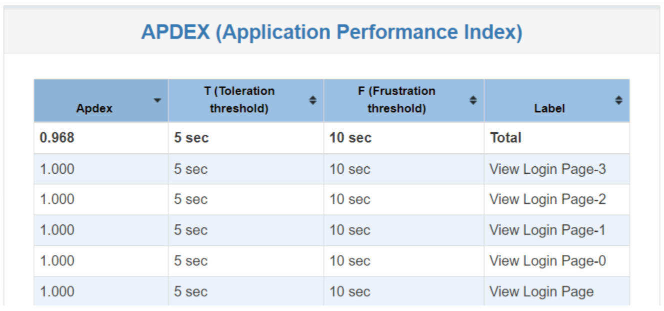
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 500 users in 2 e2.medium instances

Concurrent Users: 1000 users

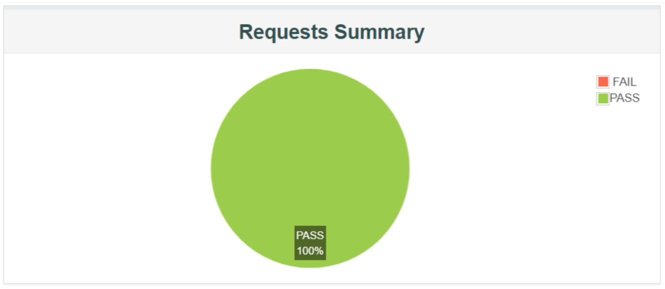
Ramp-up Time: 50s ramp-up time for each Jmeter client

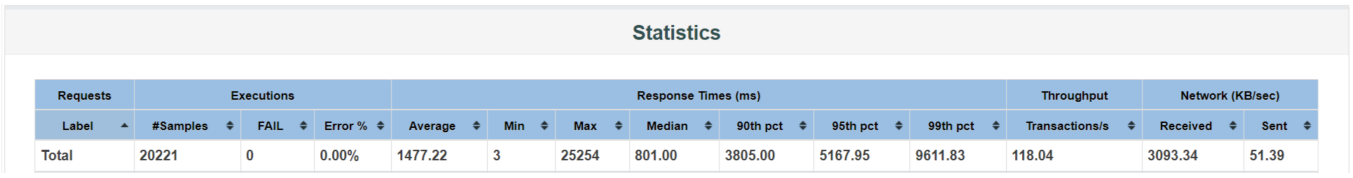
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.968** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 20221 | 0 | 0.00% | 1477.22 | 3 | 25254 | 801.00 | 3805.00 | 5167.95 | 9611.83 | 118.04 | 3093.34 | 51.39 |

## 3.6) 1000 users 2 node cluster

Load Balancer: Nginx web server e2.standard-2

Application Server: 2 c2d.highcpu-4

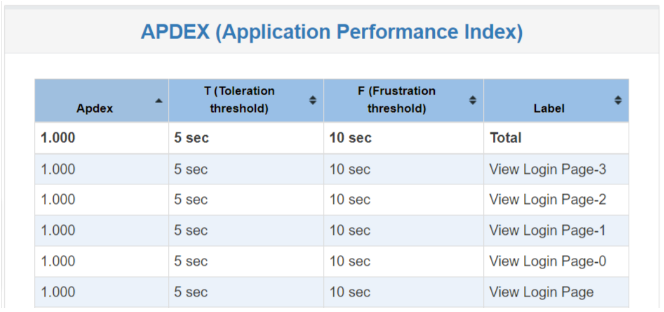
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 500 users in 2 e2.medium instances

Concurrent Users: 1000 users

Ramp-up Time: 50s ramp-up time for each Jmeter client

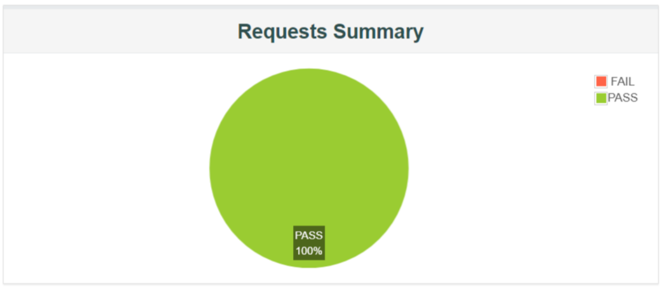
Think Time: 10s random delay 3s deviation

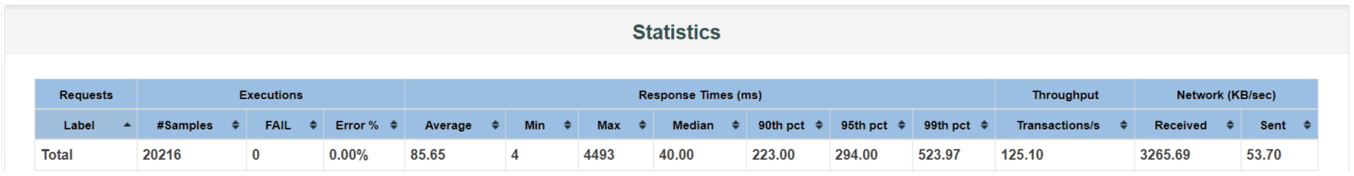


APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |

Requests Summary





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 20216 | 0 | 0.00% | 85.65 | 4 | 4493 | 40.00 | 223.00 | 294.00 | 523.97 | 125.10 | 3265.69 | 53.70 |

## 3.7) 2000 users 2 node cluster

Load Balancer: Nginx web server e2.standard-2

Application Server: 2 c2d.highcpu-4

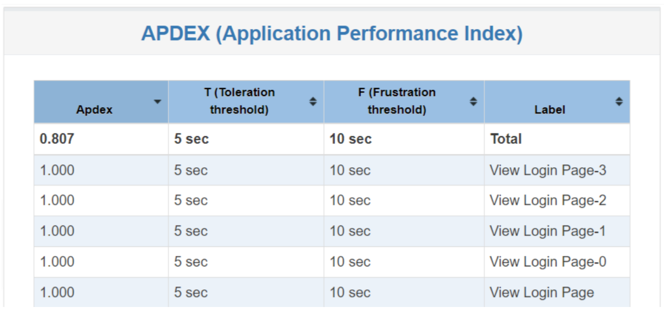
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 1000 users in 2 e2.medium instances

Concurrent Users: 2000 users

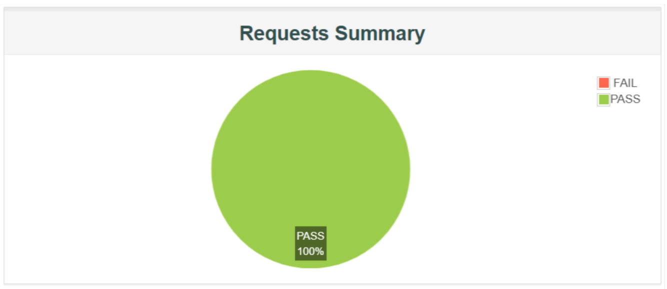
Ramp-up Time: 50s ramp-up time for each Jmeter client

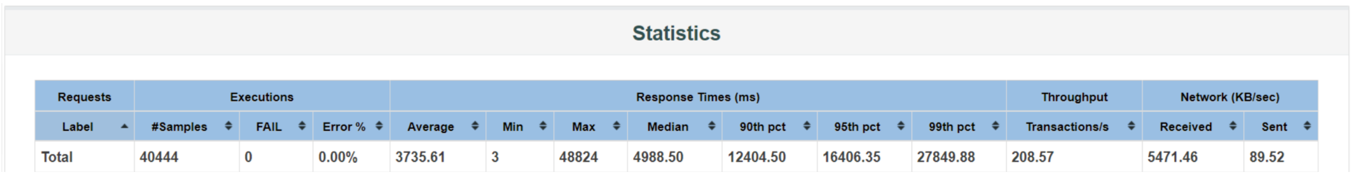
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.807** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 40444 | 0 | 0.00% | 3735.61 | 3 | 48824 | 4988.50 | 12404.50 | 16406.35 | 27849.88 | 208.57 | 5471.46 | 89.52 |

## 3.8) 2000 users 3 node cluster

Load Balancer: Nginx web server e2.standard-2

Application Server: 3 c2d.highcpu-4

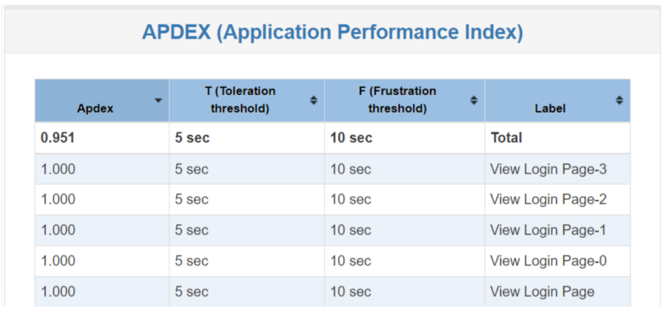
Database Server: 1 c2d.highcpu-4 node with 1500 IOPS

Client: 667 users in 3 e2.medium instances

Concurrent Users: 2001 users

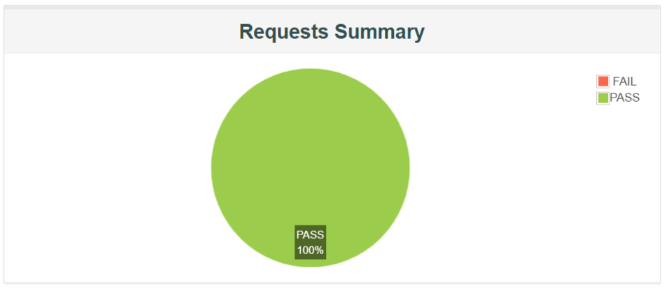
Ramp-up Time: 50s ramp-up time for each Jmeter client

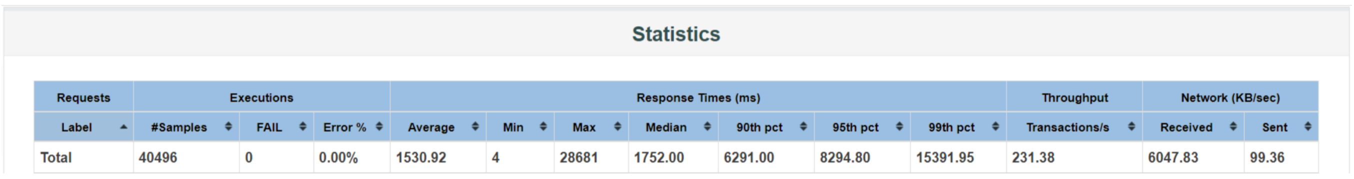
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.951** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-3 |
| 1.000 | 5 sec | 10 sec | View Login Page-2 |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 40496 | 0 | 0.00% | 1530.92 | 4 | 28681 | 1752.00 | 6291.00 | 8294.80 | 15391.95 | 231.38 | 6047.83 | 99.36 |

## OpenShift Joget EAP

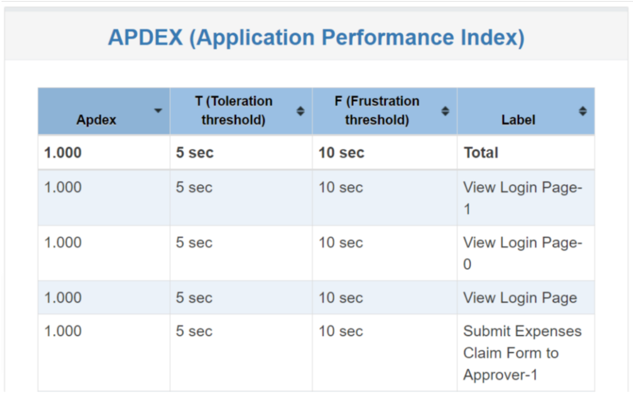
### 3.9) 100 users 2 pods

Client: 50 users in 2 e2.medium instances

Concurrent Users: 100 users

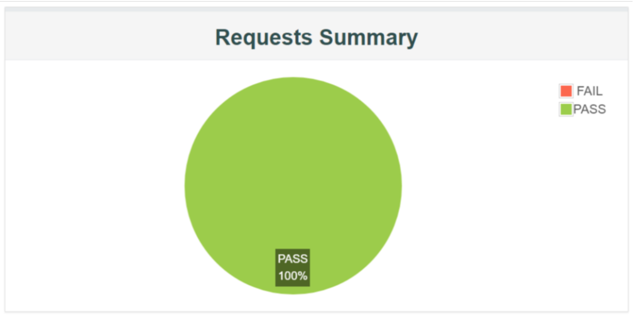
Ramp-up Time: 5s ramp-up time for each Jmeter client

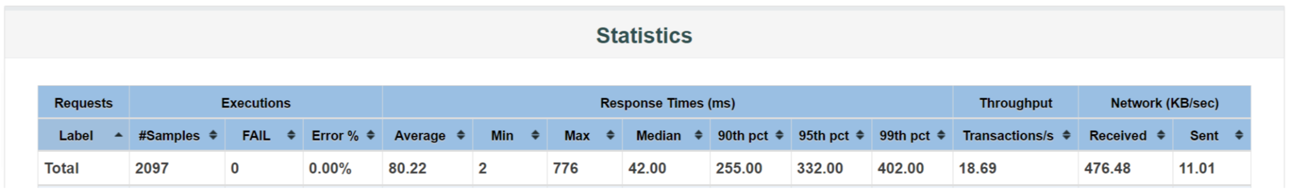
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 2097 | 0 | 0.00% | 80.22 | 2 | 776 | 42.00 | 255.00 | 332.00 | 402.00 | 18.69 | 476.48 | 11.01 |

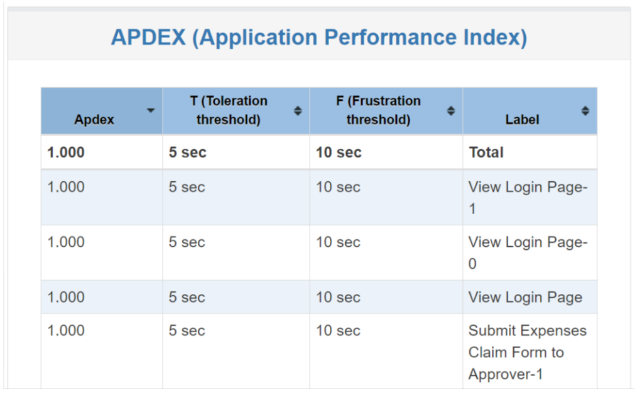
## 3.10) 250 users 2 pods

Client: 125 users in 2 e2.medium instances

Concurrent Users: 250 users

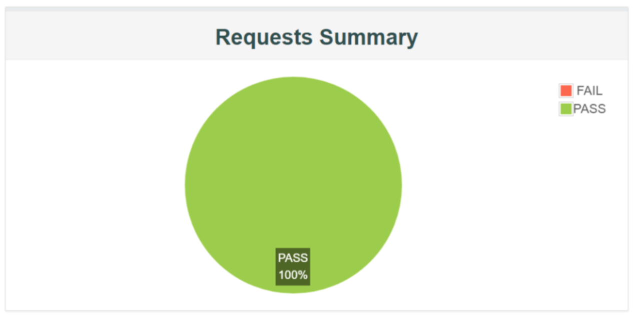
Ramp-up Time: 10s ramp-up time for each Jmeter client

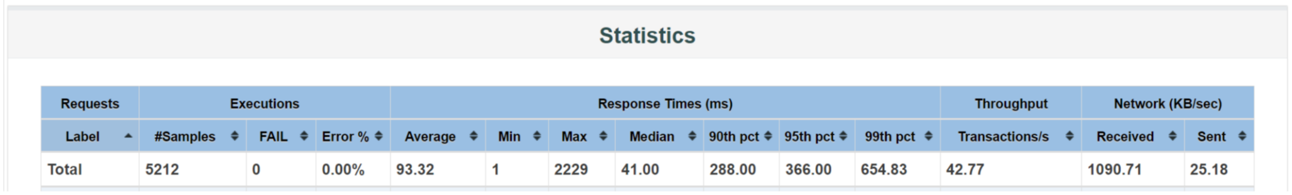
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 5212 | 0 | 0.00% | 93.32 | 1 | 2229 | 41.00 | 288.00 | 366.00 | 654.83 | 42.77 | 1090.71 | 25.18 |

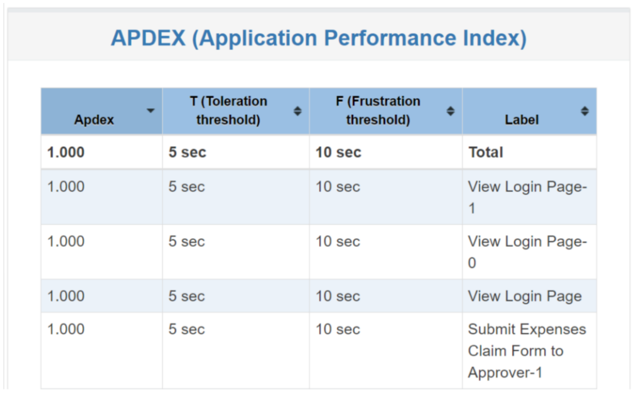
## 3.11) 500 users 2 pods

Client: 250 users in 2 e2.medium instances

Concurrent Users: 500 users

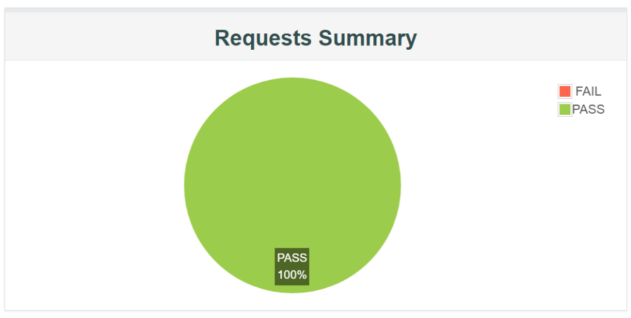
Ramp-up Time: 25s ramp-up time for each Jmeter client

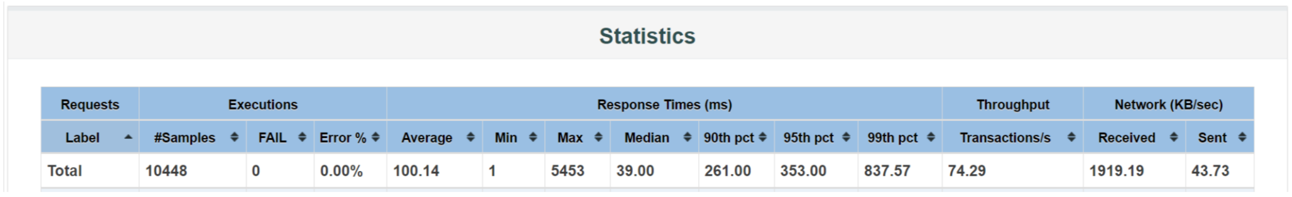
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **1.000** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |





Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| **Total** | **10448** | **0** | **0.00%** | **100.14** | **1** | **5453** | **39.00** | **261.00** | **353.00** | **837.57** | **74.29** | **1919.19** | **43.73** |

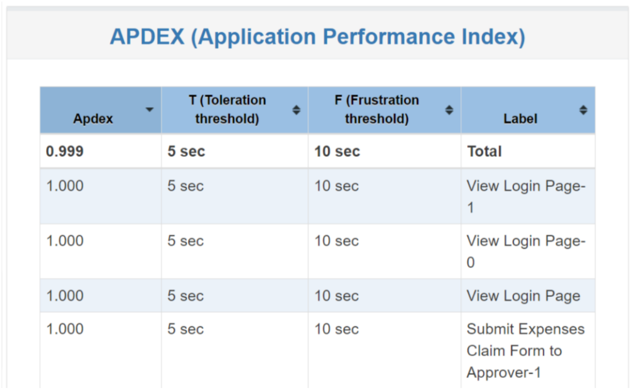
## 3.12) 750 users 2 pods

Client: 375 users in 2 e2.medium instances

Concurrent Users: 750 users

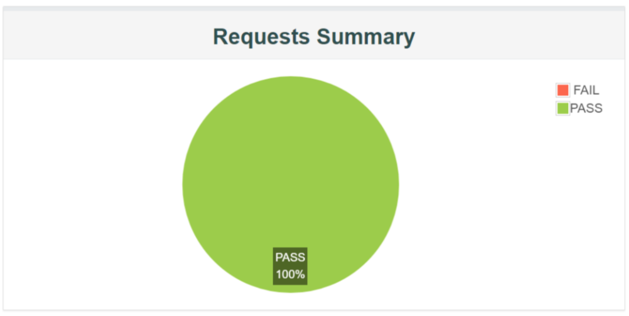
Ramp-up Time: 35s ramp-up time for each Jmeter client

Think Time: 10s random delay 3s deviation

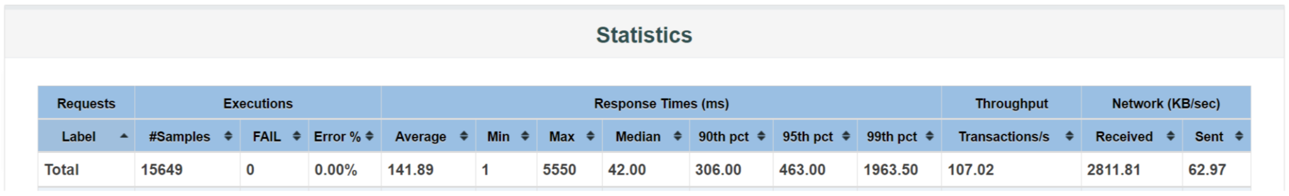


APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.999** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |
| 1.000 | 5 sec | 10 sec | Submit Expenses Claim Form to Approver-1 |



Requests Summary



Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 15649 | 0 | 0.00% | 141.89 | 1 | 5550 | 42.00 | 306.00 | 463.00 | 1963.50 | 107.02 | 2811.81 | 62.97 |

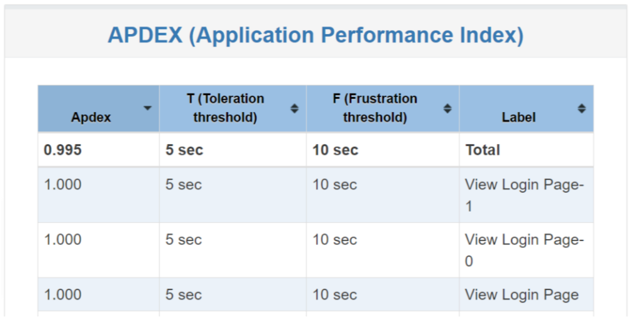
## 3.13) 1000 users 2 pods

Client: 500 users in 2 e2.medium instances

Concurrent Users: 1000 users

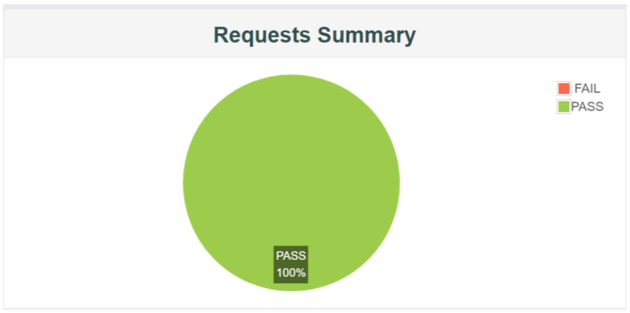
Ramp-up Time: 50s ramp-up time for each Jmeter client

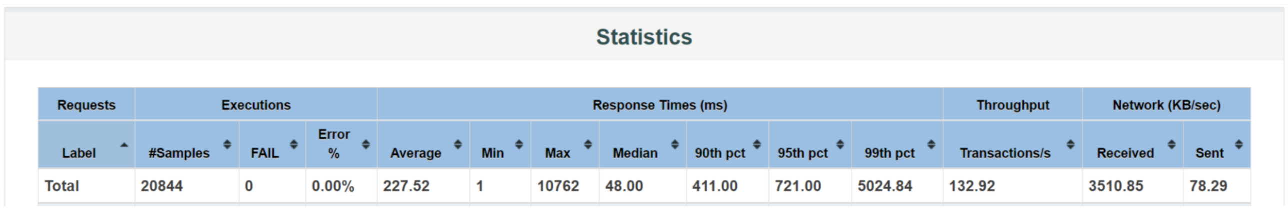
Think Time: 10s random delay 3s deviation



APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.995** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |


Requests Summary



Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| **Total** | **20844** | **0** | **0.00%** | **227.52** | **1** | **10762** | **48.00** | **411.00** | **721.00** | **5024.84** | **132.92** | **3510.85** | **78.29** |

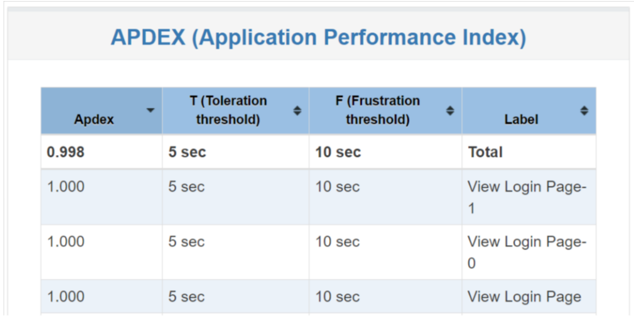
## 3.14) 1000 users 4 pods

Client: 500 users in 2 e2.medium instances

Concurrent Users: 1000 users

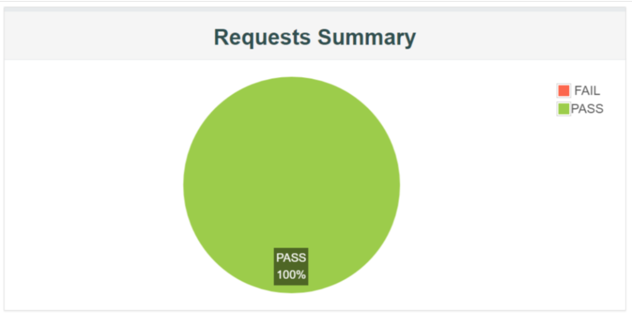
Ramp-up Time: 50s ramp-up time for each Jmeter client

Think Time: 10s random delay 3s deviation

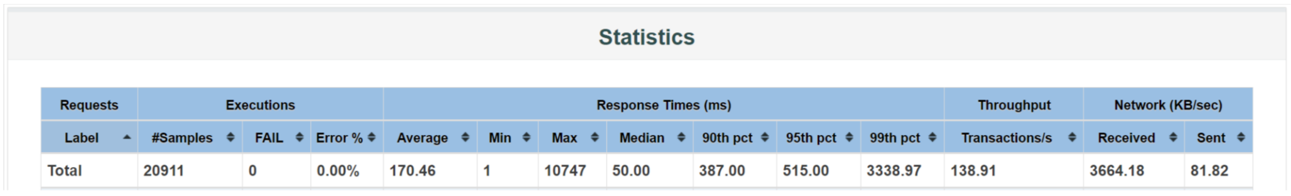


APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.998** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |



Requests Summary



Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 20911 | 0 | 0.00% | 170.46 | 1 | 10747 | 50.00 | 387.00 | 515.00 | 3338.97 | 138.91 | 3664.18 | 81.82 |

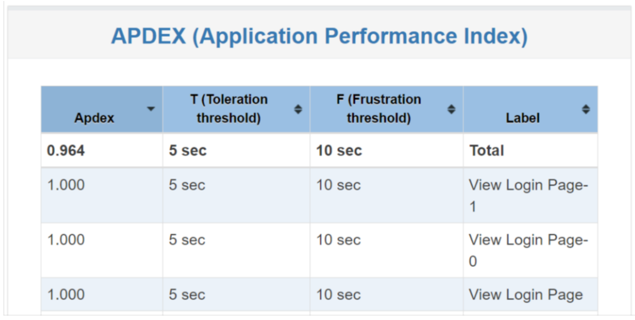
## 3.15) 2000 users 4 pods

Client: 1000 users in 2 e2.medium instances

Concurrent Users: 2000 users

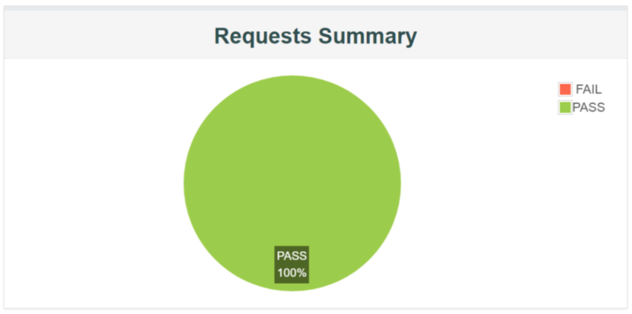
Ramp-up Time: 50s ramp-up time for each Jmeter client

Think Time: 10s random delay 3s deviation

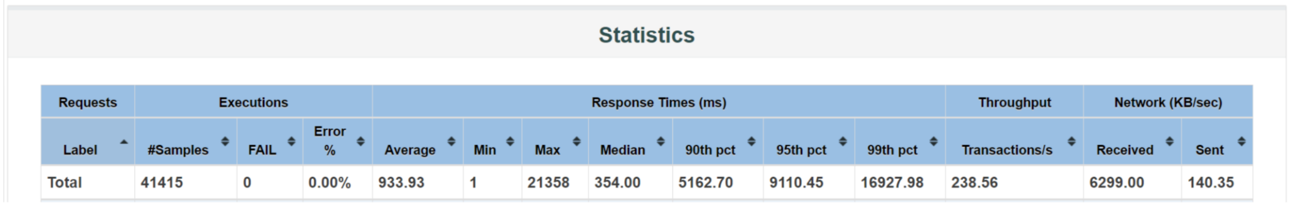


**APDEX (Application Performance Index)**

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.964** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |



**Requests Summary**



**Statistics**

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 41415 | 0 | 0.00% | 933.93 | 1 | 21358 | 354.00 | 5162.70 | 9110.45 | 16927.98 | 238.56 | 6299.00 | 140.35 |

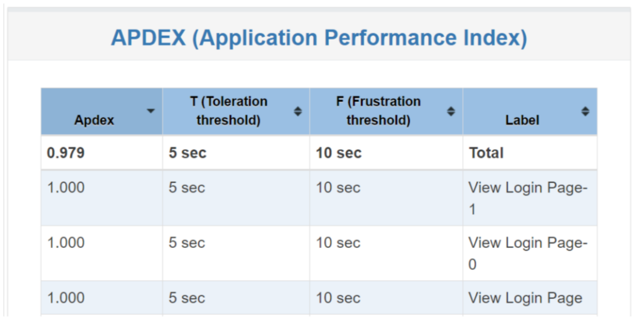
## 3.16) 2000 users 6 pods

Client: 1000 users in 2 e2.medium instances

Concurrent Users: 2000 users

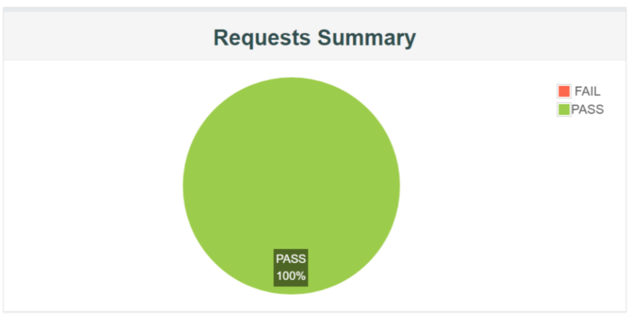
Ramp-up Time: 50s ramp-up time for each Jmeter client

Think Time: 10s random delay 3s deviation

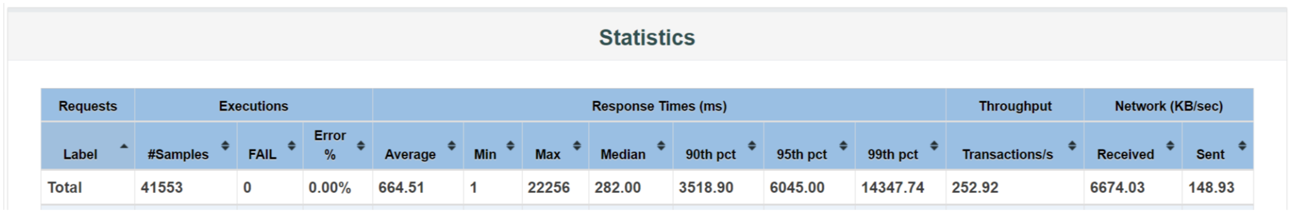


APDEX (Application Performance Index)

| Apdex | T (Toleration threshold) | F (Frustration threshold) | Label |
|---|---|---|---|
| **0.979** | **5 sec** | **10 sec** | **Total** |
| 1.000 | 5 sec | 10 sec | View Login Page-1 |
| 1.000 | 5 sec | 10 sec | View Login Page-0 |
| 1.000 | 5 sec | 10 sec | View Login Page |



Requests Summary

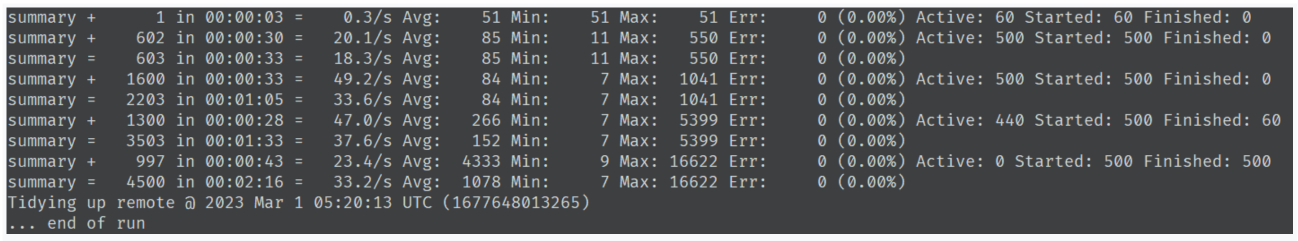


Statistics

| Requests | Executions | | | Response Times (ms) | | | | | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct | Transactions/s | Received | Sent |
| Total | 41553 | 0 | 0.00% | 664.51 | 1 | 22256 | 282.00 | 3518.90 | 6045.00 | 14347.74 | 252.92 | 6674.03 | 148.93 |

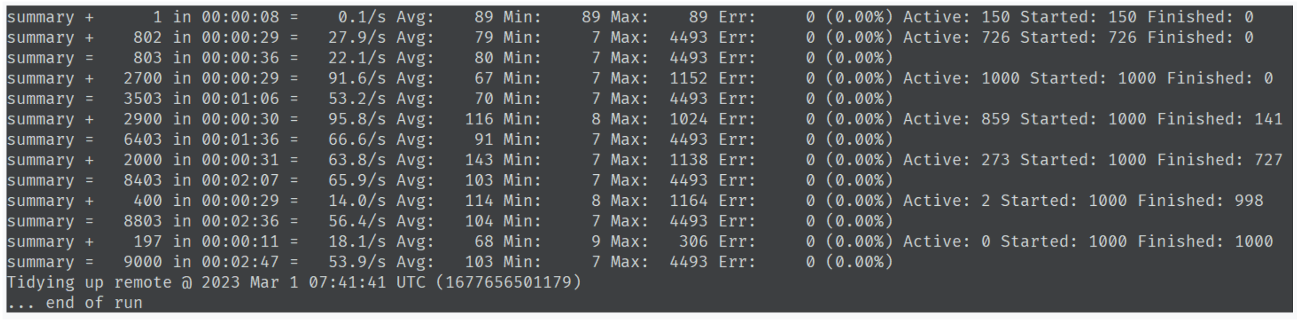
# Appendix: Sample Test Output

## 500 users 1 node JMeter output



```
summary +      1 in 00:00:03 =    0.3/s Avg:    51 Min:    51 Max:    51 Err:     0 (0.00%) Active: 60 Started: 60 Finished: 0
summary +    602 in 00:00:30 =   20.1/s Avg:    85 Min:    11 Max:   550 Err:     0 (0.00%) Active: 500 Started: 500 Finished: 0
summary =    603 in 00:00:33 =   18.3/s Avg:    85 Min:    11 Max:   550 Err:     0 (0.00%)
summary +   1600 in 00:00:33 =   49.2/s Avg:    84 Min:     7 Max:  1041 Err:     0 (0.00%) Active: 500 Started: 500 Finished: 0
summary =   2203 in 00:01:05 =   33.6/s Avg:    84 Min:     7 Max:  1041 Err:     0 (0.00%)
summary +   1300 in 00:00:28 =   47.0/s Avg:   266 Min:     7 Max:  5399 Err:     0 (0.00%) Active: 440 Started: 500 Finished: 60
summary =   3503 in 00:01:33 =   37.6/s Avg:   152 Min:     7 Max:  5399 Err:     0 (0.00%)
summary +    997 in 00:00:43 =   23.4/s Avg:  4333 Min:     9 Max: 16622 Err:     0 (0.00%) Active: 0 Started: 500 Finished: 500
summary =   4500 in 00:02:16 =   33.2/s Avg:  1078 Min:     7 Max: 16622 Err:     0 (0.00%)
Tidying up remote @ 2023 Mar 1 05:20:13 UTC (1677648013265)
... end of run
```

## 1000 users 2 node cluster JMeter output



```
summary +      1 in 00:00:08 =    0.1/s Avg:    89 Min:    89 Max:    89 Err:     0 (0.00%) Active: 150 Started: 150 Finished: 0
summary +    802 in 00:00:29 =   27.9/s Avg:    79 Min:     7 Max:  4493 Err:     0 (0.00%) Active: 726 Started: 726 Finished: 0
summary =    803 in 00:00:36 =   22.1/s Avg:    80 Min:     7 Max:  4493 Err:     0 (0.00%)
summary +   2700 in 00:00:29 =   91.6/s Avg:    67 Min:     7 Max:  1152 Err:     0 (0.00%) Active: 1000 Started: 1000 Finished: 0
summary =   3503 in 00:01:06 =   53.2/s Avg:    70 Min:     7 Max:  4493 Err:     0 (0.00%)
summary +   2900 in 00:00:30 =   95.8/s Avg:   116 Min:     8 Max:  1024 Err:     0 (0.00%) Active: 859 Started: 1000 Finished: 141
summary =   6403 in 00:01:36 =   66.6/s Avg:    91 Min:     7 Max:  4493 Err:     0 (0.00%)
summary +   2000 in 00:00:31 =   63.8/s Avg:   143 Min:     7 Max:  1138 Err:     0 (0.00%) Active: 273 Started: 1000 Finished: 727
summary =   8403 in 00:02:07 =   65.9/s Avg:   103 Min:     7 Max:  4493 Err:     0 (0.00%)
summary +    400 in 00:00:29 =   14.0/s Avg:   114 Min:     8 Max:  1164 Err:     0 (0.00%) Active: 2 Started: 1000 Finished: 998
summary =   8803 in 00:02:36 =   56.4/s Avg:   104 Min:     7 Max:  4493 Err:     0 (0.00%)
summary +    197 in 00:00:11 =   18.1/s Avg:    68 Min:     9 Max:   306 Err:     0 (0.00%) Active: 0 Started: 1000 Finished: 1000
summary =   9000 in 00:02:47 =   53.9/s Avg:   103 Min:     7 Max:  4493 Err:     0 (0.00%)
Tidying up remote @ 2023 Mar 1 07:41:41 UTC (1677656501179)
... end of run
```