

# Assignment 2 - Yang Chen

Git repo:

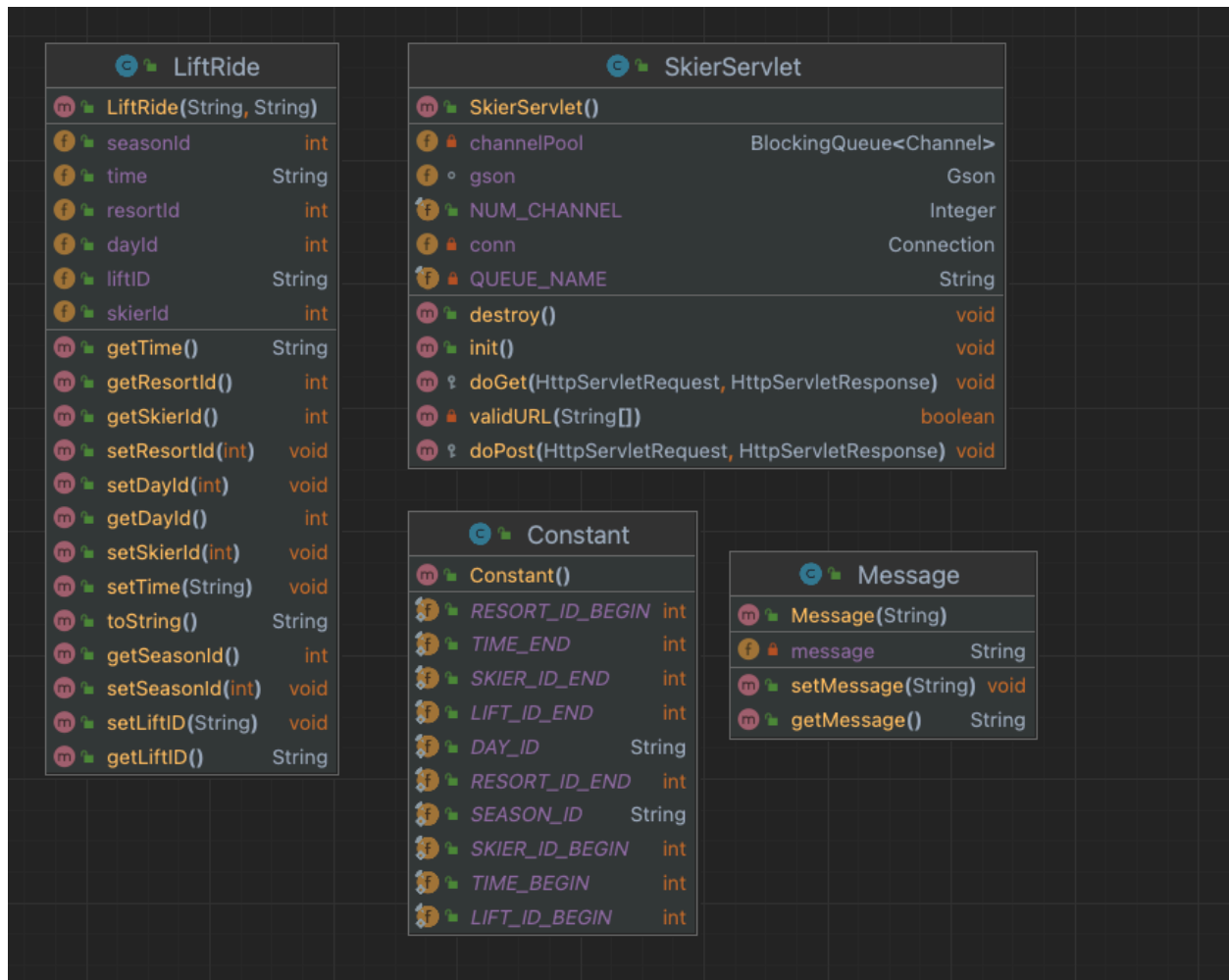
<https://github.com/Yangyanggogo/SkiResortDistributedSystem/tree/hw2>

## A note for this report

The load tests were launched in New Mexico state instead of in Seattle. As a result, the overall test results would be less significant than other students who tested in Seattle.

## Server design description

Here is my Server UML diagram, which presents the classes included in my server.



The server design is quite similar to the server design in my assignment 1, in addition to a complete validation for the URL and JSON payload.

If the server receives valid post requests, it formats the incoming data and sends it as a payload to the RabbitMQ, which is set remotely in a separate EC2 instance. To increase the efficiency of sending payloads to RabbitMQ, a connection pool was designed in my server using "BlockingQueue", with 20 channels initialized in this connection pool. When the server receives a valid post request, it will poll an available channel out of this connection pool; when finishing sending messages to remote RabbitMQ, this channel will be put back into the connection pool.

The correct connection to remote RabbitMQ was ensured in the init() method. All the information needed for this connection was stored in rabbitmq.conf file, which ensured that private information was not exposed in my code. Since my server and the RabbitMQ were located in two different EC2 instances, I used a private IPV4 address for their connections.

The detailed implementation can be found in my code, under the Server file path.

## Consumer design description

The consumer was designed to pull messages off the RabbitMQ with a multi-thread program to consume as soon as the queue receives a message. It receives messages from the RabbitMQ and keeps a record of the individual lift rides for each skier in a hash map "ConcurrentHashMap", which is a good choice for multithread design.

The consumer is also located in a separate EC2 and connects to RabbitMQ using the same methodology as the server.

The number of threads in the Consumer and the basicQos setting in each channel can be passed as "args" when running Consumer.jar, which is convenient for load tests.

The detailed implementation can be found in my code, under the Consumer file path.

## Load balancers description

I have four instances registered in the registered targets. The setting details can be found in the figures below.

The load balancer listens to port 80, distributes requests to its target groups, and then the target groups route requests to the Tomcat server.

EC2 > Load balancers

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

1

Name

DNS name

State

VPC ID

Availability Zones

Type

Date created

cs6650-LB

cs6650-LB-567700519.us-...

Active

vpc-0de1006cfeda84947

4 Availability Zones

application

February 22, 2024, 18:39 (UTC-07:00)

Load balancer: cs6650-LB

Details

Listeners and rules

Network mapping

Security

Monitoring

Integrations

Attributes

Tags

Details

Load balancer type

Application

Status

Active

VPC

vpc-0de1006cfeda84947

IP address type

IPv4

Scheme

Internet-facing

Hosted zone

Z1H1FL5HABSF5

Availability Zones

subnet-09113303b9229eeec us-west-2d (usw2-az4)

subnet-08da9a3901f1e1164 us-west-2c (usw2-az3)

subnet-021cb5b1eb9c1f5f1 us-west-2b (usw2-az1)

subnet-02caadfa4dd31da2 us-west-2a (usw2-az2)

Date created

February 22, 2024, 18:39 (UTC-07:00)

Load balancer ARN

arn:aws:elasticloadbalancing:us-west-2:851725592368:loadbalancer/app/cs6650-LB/8b783bf72ec9bff1

DNS name

cs6650-LB-567700519.us-west-2.elb.amazonaws.com (A Record)

# Client

There are no changes for the client part, I used Client2 for all the load tests in this assignment.

## Load test with a single servlet.

```
47 - thread received {"time":"348","liftID":"12","skierID":98486}
47 - thread received {"time":"316","liftID":"5","skierID":15547}
47 - thread received {"time":"199","liftID":"27","skierID":61928}
47 - thread received {"time":"11","liftID":"26","skierID":16781}
47 - thread received {"time":"208","liftID":"27","skierID":84733}
47 - thread received {"time":"241","liftID":"20","skierID":42516}
47 - thread received {"time":"163","liftID":"26","skierID":89254}
47 - thread received {"time":"150","liftID":"11","skierID":82488}
47 - thread received {"time":"333","liftID":"30","skierID":72341}
47 - thread received {"time":"90","liftID":"28","skierID":75089}
47 - thread received {"time":"262","liftID":"7","skierID":48230}
47 - thread received {"time":"303","liftID":"2","skierID":50248}
47 - thread received {"time":"123","liftID":"34","skierID":84948}
47 - thread received {"time":"233","liftID":"39","skierID":2248}
47 - thread received {"time":"180","liftID":"25","skierID":78160}
47 - thread received {"time":"2","liftID":"23","skierID":70969}
47 - thread received {"time":"234","liftID":"12","skierID":76252}
47 - thread received {"time":"216","liftID":"26","skierID":74475}
47 - thread received {"time":"199","liftID":"13","skierID":41385}
47 - thread received {"time":"309","liftID":"1","skierID":40722}
47 - thread received {"time":"343","liftID":"4","skierID":22252}
```

Client number of threads in phase 2: 100

Consumer parameters:

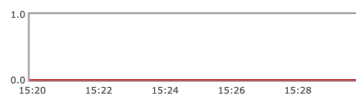
Number of threads: 32

basicQos = 1

### Queue PostMessageQUEUE

Overview

Queued messages [last ten minutes](#) [?](#)



Ready 0  
Unacked 0  
Total 0

Message rates [last ten minutes](#) [?](#)



Publish 0.00/s  
Deliver (manual ack) 0.00/s  
Deliver (auto ack) 0.00/s  
Consumer ack 0.00/s  
Redelivered 0.00/s  
Get (manual ack) 0.00/s  
Get (auto ack) 0.00/s  
Get (empty) 0.00/s

Details

Features	State	Messages	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32	0	0	0	0	0	0
Operator policy	Consumer capacity	100%	0 B	0 B	0 B	0 B	0 B	0 B
Effective policy definition		Process memory	29 KIB					

Consumers (32)

Bindings (1)

```
-----
Multi threads consumer test start
-----

Number of threads in phase 2: 100
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 191298
Mean latency= 71.72291 ms
Median latency= 66.0 ms
P99 latency= 151.0 ms
Throughout: 1047 requests/second
Min latency= 52
Max latency= 638
-----

Multi threads consumer test end
-----
```

Client number of threads in phase 2: 200

Consumer parameters:

Number of threads: 32

basicQos = 1

Queue PostMessageQUEUE

Overview

Queued messages last ten minutes ?

Ready

Unacked

Total

0

0

0

Message rates last ten minutes ?

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features	State	Idle	Messages ?	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32	Message body bytes ?	0 B	0 B	0 B	0 B	0 B	0 B
Operator policy	Consumer capacity ?	100%	Process memory ?	29 KIB					
Effective policy definition									

Consumers (32)

Bindings (1)

```
Number of threads in phase 2: 200
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 137421
Mean latency= 79.94868102637948 ms
Median latency= 72.0 ms
P99 latency= 179.0 ms
Throughout: 1459 requests/second
Min latency= 52
Max latency= 10306
-----
Multi threads consumer test end
-----
```

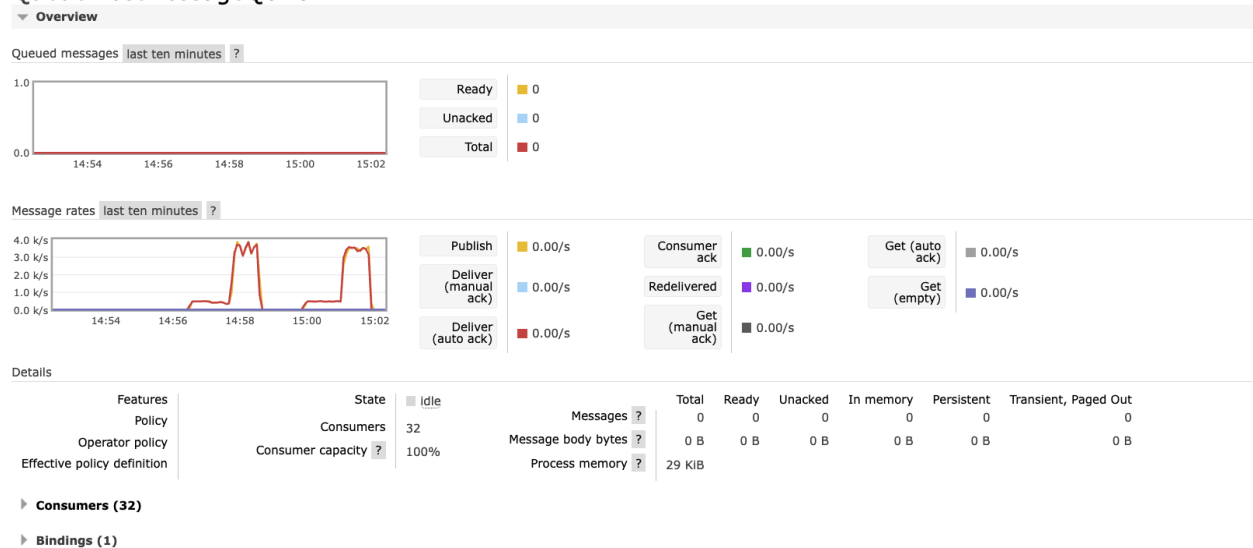
Client number of threads in phase 2: 300

Consumer parameters:

Number of threads: 32

**basicQos = 1**

## Queue PostMessageQUEUE



```
Multi threads consumer test start
-----
Number of threads in phase 2: 300
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 117835
Mean latency= 84.39824 ms
Median latency= 75.0 ms
P99 latency= 196.0 ms
Throughout: 1709 requests/second
Min latency= 51
Max latency= 815
-----
Multi threads consumer test end
-----
```

Client number of threads in phase 2: 400

Consumer parameters:


Number of threads: 32

**basicQos = 1**

Queue PostMessageQUEUE

Overview

Queued messages last ten minutes ?



Ready

Unacked

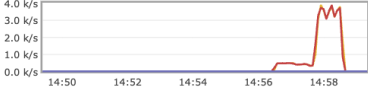
Total

0

0

0

Message rates last ten minutes ?



Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features	State	Idle	Messages ?	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32		0	0	0	0	0	0
Operator policy	Consumer capacity ?	100%	Message body bytes ?	0 B	0 B	0 B	0 B	0 B	0 B
Effective policy definition			Process memory ?	29 KIB					

Consumers (32)

Bindings (1)

```
Multi threads consumer test start
-----
Number of threads in phase 2: 400
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 124165
Mean latency= 109.67791 ms
Median latency= 93.0 ms
P99 latency= 407.0 ms
Throughout: 1612 requests/second
Min latency= 52
Max latency= 1492
-----
Multi threads consumer test end
-----
```

Client number of threads in phase 2: 500

Consumer parameters:

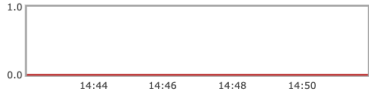
Number of threads: 32

**basicQos = 1**

Queue PostMessageQUEUE

Overview

Queued messages last ten minutes ?



Ready

Unacked

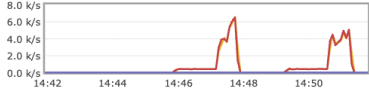
Total

0

0

0

Message rates last ten minutes ?



Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features	State	Idle	Messages ?	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32		0	0	0	0	0	0
Operator policy	Consumer capacity ?	100%	Message body bytes ?	0 B	0 B	0 B	0 B	0 B	0 B
Effective policy definition			Process memory ?	32 KIB					

Consumers (32)

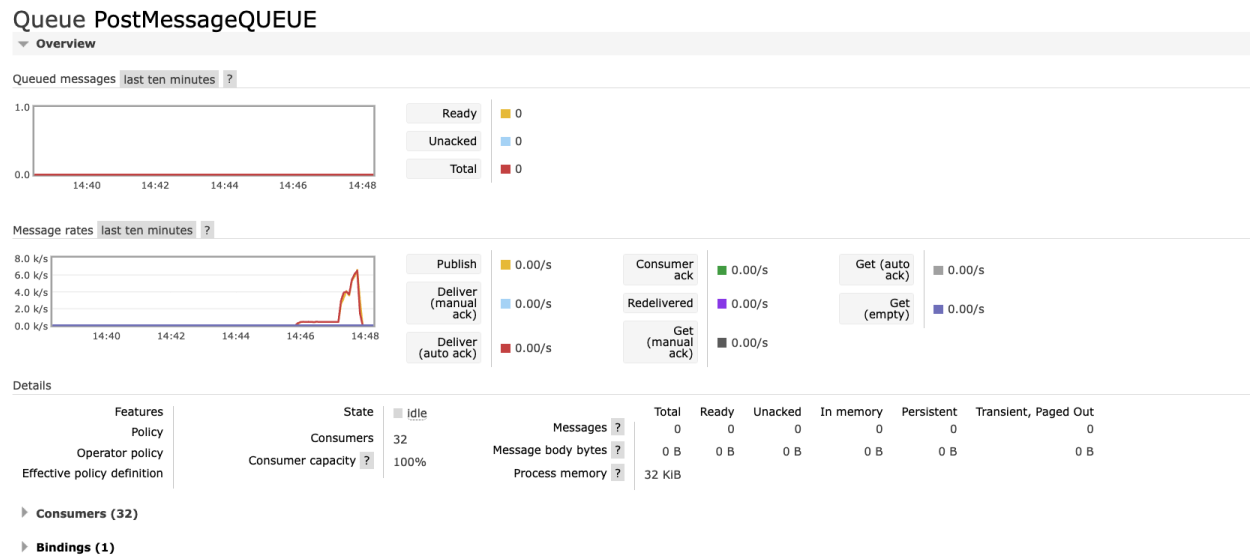
Bindings (1)



```
Number of threads in phase 2: 500
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 119452
Mean latency= 114.52774305798594 ms
Median latency= 92.0 ms
P99 latency= 443.0 ms
Throughout: 1680 requests/second
Min latency= 52
Max latency= 10474
-----
Multi threads consumer test end
```

Client number of threads in phase 2: 800

Consumer parameters:  
Number of threads: 32  
**basicQos = 1**



```
Multi threads consumer test start
-----
Number of threads in phase 2: 800
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 115043|
Mean latency= 152.26968 ms
Median latency= 122.0 ms
P99 latency= 639.0 ms
Throughout: 1739 requests/second
Min latency= 52
Max latency= 1652
-----
Multi threads consumer test end
```

## Load test with a load-balanced servlet

All tests are launched under this condition for Consumer:

Number of threads: 32

**basicQos = 1**

## Load test with a load-balanced servlet - 2 instances

EC2 > Target groups > cs6650-LB-TG

### cs6650-LB-TG

Actions

**Introducing Automatic Target Weights (ATW) to increase application availability**

Automatic Target Weights is achieved by turning on anomaly mitigation, which provides responsive, dynamic distribution of traffic to targets based on anomaly detection results. All HTTP/HTTPS target groups now include anomaly detection by default. [Learn more](#)

#### Details

arn:aws:elasticloadbalancing:us-west-2:851725592368:targetgroup/cs6650-LB-TG/2ce87e8b53c1afd9

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC <a href="#">vpc-0de1006cfeda84947</a>
IP address type IPv4	Load balancer <a href="#">cs6650-LB</a>		

2 Total targets	2 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining
--------------------	-----------------------------	----------------	-------------	--------------	---------------

► **Distribution of targets by Availability Zone (AZ)**

Select values in this table to see corresponding filters applied to the Registered targets table below.

**Targets** | Monitoring | Health checks | Attributes | Tags

#### Registered targets (2)

Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection result
<input type="checkbox"/>	<a href="#">i-07d0a355efb6c6ff6f8</a>		8080	us-west-2c	Healthy	-	February ...	Normal
<input type="checkbox"/>	<a href="#">i-0c51d8e854a4645f3</a>	hw	8080	us-west-2a	Healthy	-	February ...	Normal

## Client number of threads in phase 2: 100

```
Multi threads consumer test start
-----
Number of threads in phase 2: 100
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 200496
Mean latency= 74.163965 ms
Median latency= 67.0 ms
P99 latency= 167.0 ms
Throughout: 1000 requests/second
Min latency= 52
Max latency= 650
-----
Multi threads consumer test end
-----
```

Queue PostMessageQUEUE

Overview

Queued messages

last ten minutes

?

Ready

Unacked

Total

0

0

0

Message rates

last ten minutes

?

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features

Policy

Operator policy

Effective policy definition

State

Consumers

Consumer capacity

idle

32

100%

Messages

Message body bytes

Process memory

0

0 B

34 KIB

Total

Ready

Unacked

In memory

Persistent

Transient, Paged Out

0

0

0

0

0

0

0 B

0 B

0 B

0 B

0 B

0 B

Consumers (32)

Bindings (1)

Client number of threads in phase 2: 200

Overview

Queued messages

last ten minutes

?

Ready

Unacked

Total

0

0

0

Message rates

last ten minutes

?

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features

Policy

Operator policy

Effective policy definition

State

Consumers

Consumer capacity

idle

32

100%

Messages

Message body bytes

Process memory

0

0 B

34 KIB

Total

Ready

Unacked

In memory

Persistent

Transient, Paged Out

0

0

0

0

0

0

0 B

0 B

0 B

0 B

0 B

0 B

Consumers (32)

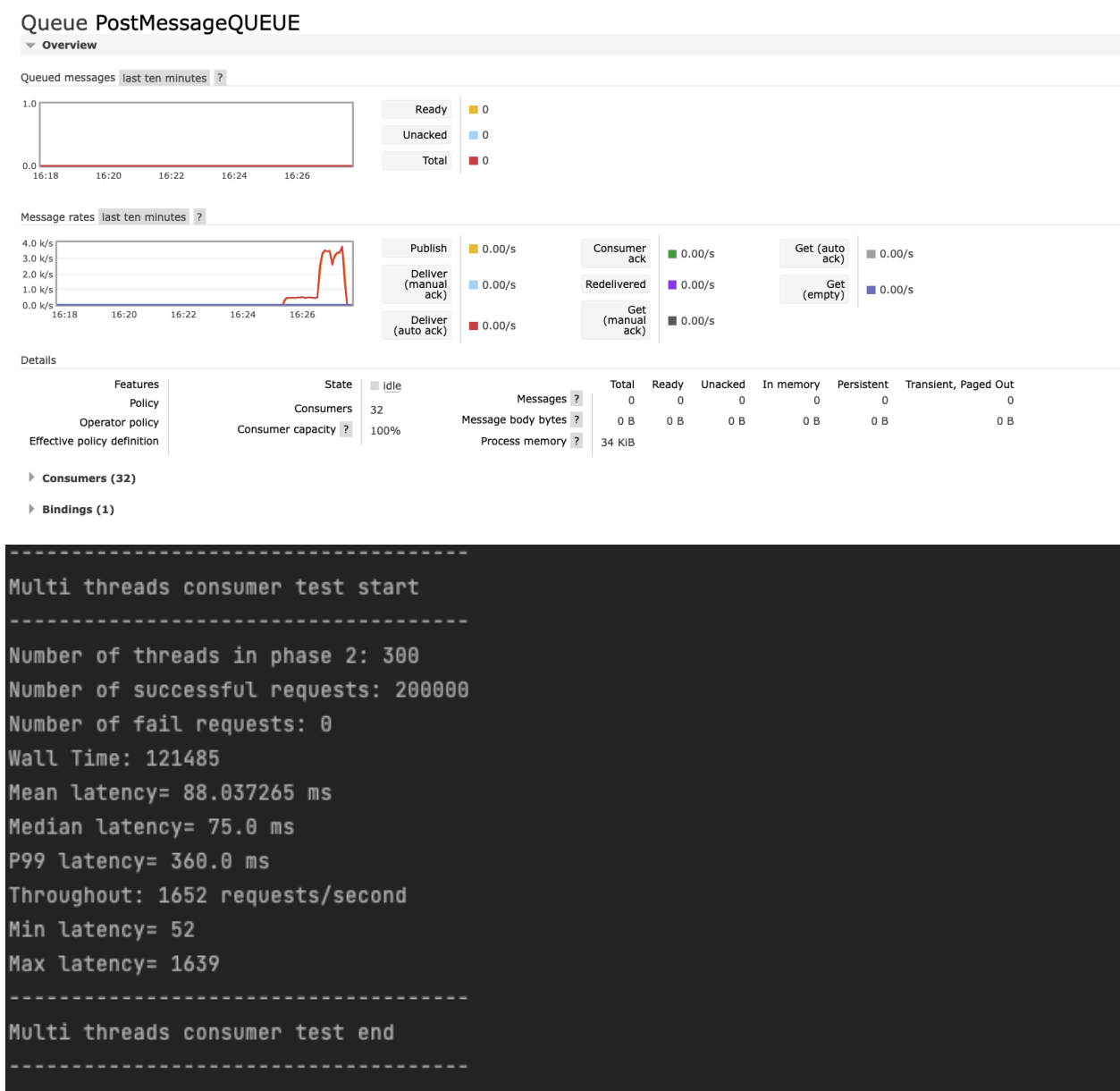
Bindings (1)

```
-----
Multi threads consumer test start
-----

Number of threads in phase 2: 200
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 142182
Mean latency= 80.760925 ms
Median latency= 72.0 ms
P99 latency= 195.0 ms
Throughout: 1408 requests/second
Min latency= 52
Max latency= 1162
-----

Multi threads consumer test end
-----
```

Client number of threads in phase 2: 300



Client number of threads in phase 2: 400

Queue PostMessageQUEUE

Overview

Queued messages

last ten minutes

?

1.0

0.0

16:22

16:24

16:26

16:28

16:30

Ready

Unacked

Total

0

0

0

Message rates

last ten minutes

?

6.0 k/s

4.0 k/s

2.0 k/s

0.0 k/s

16:22

16:24

16:26

16:28

16:30

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features

Policy

Operator policy

Effective policy definition

State

Consumers

Consumer capacity

idle

32

100%

Messages

Message body bytes

Process memory

0

0 B

34 KiB

Total

Ready

Unacked

In memory

Persistent

Transient, Paged Out

0

0

0

0

0

0

Consumers (32)

Bindings (1)

```
Multi threads consumer test start
-----
Number of threads in phase 2: 400
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 115863
Mean latency= 90.453235 ms
Median latency= 79.0 ms
P99 latency= 349.0 ms
Throughout: 1739 requests/second
Min latency= 52
Max latency= 2348
-----|
Multi threads consumer test end
-----
```

Client number of threads in phase 2: 500

Queue PostMessageQUEUE

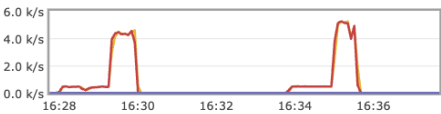
Overview

Queued messages last ten minutes ?



Ready	0
Unacked	0
Total	0

Message rates last ten minutes ?



Publish	0.00/s
Deliver (manual ack)	0.00/s
Deliver (auto ack)	0.00/s

Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s

Get (auto ack)	0.00/s
Get (empty)	0.00/s

Details

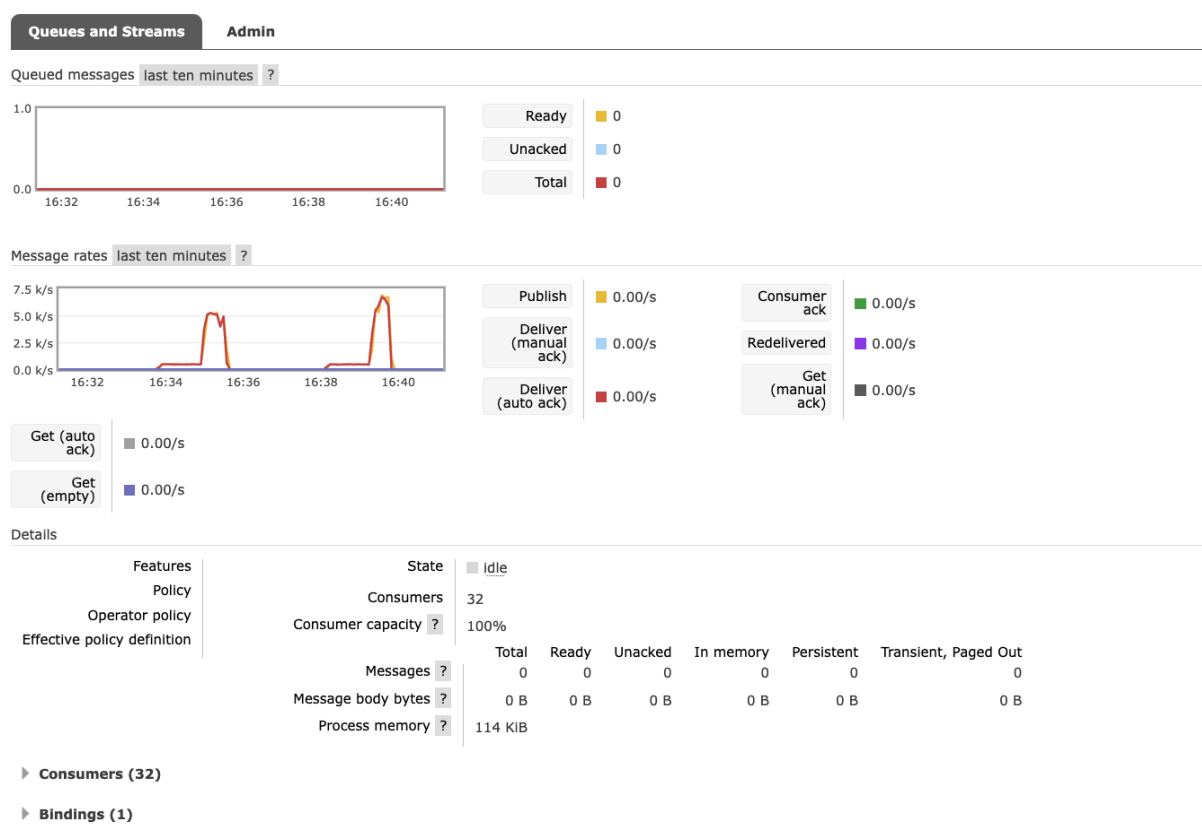
Features	State	Idle
Policy	Consumers	32
Operator policy	Consumer capacity ?	100%
Effective policy definition	Messages ?	Total 0 Ready 0 Unacked 0 In memory 0 Persistent 0 Transient, Paged Out 0
	Message body bytes ?	0 B 0 B 0 B 0 B 0 B 0 B
	Process memory ?	34 KiB

Consumers (32)

Bindings (1)

```
Multi threads consumer test start
-----
Number of threads in phase 2: 500
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 103245
Mean latency= 98.39125 ms
Median latency= 84.0 ms
P99 latency= 370.0 ms
Throughout: 1941 requests/second
Min latency= 53
Max latency= 1270
-----
Multi threads consumer test end
-----
```

Client number of threads in phase 2: 800



```
-----
Multi threads consumer test start
-----
Number of threads in phase 2: 800
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 97766
Mean latency= 123.22004 ms
Median latency= 100.0 ms
P99 latency= 569.0 ms
Throughout: 2061 requests/second
Min latency= 52
Max latency= 2927
-----
Multi threads consumer test end
-----
```

Client number of threads in phase 2: 1000



Queue PostMessageQUEUE

Overview

Queued messages

last ten minutes

?

1.0

0.0

16:4816:5016:5216:5416:56

Ready

Unacked

Total

0

0

0

Message rates

last ten minutes

?

7.5 k/s

5.0 k/s

2.5 k/s

0.0 k/s

16:4816:5016:5216:5416:56

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features	State	idle	Messages	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32	?	0	0	0	0	0	0
Operator policy	Consumer capacity	100%	?	0 B	0 B	0 B	0 B	0 B	0 B
Effective policy definition			?	34 KIB					

Consumers (32)

Bindings (1)

Multi threads consumer test start

Number of threads in phase 2: 1000

Number of successful requests: 200000

Number of fail requests: 0

Wall Time: 103384

Mean latency= 157.938165 ms

Median latency= 122.0 ms

P99 latency= 794.0 ms

Throughout: 1941 requests/second

Min latency= 53

Max latency= 2973

Multi threads consumer test end

## Load test with a load-balanced servlet - 4 instances

EC2 > Target groups

Target groups (1/1) [Info](#)

☒ Name ARN Port Protocol Target type Load b

<input checked="" type="checkbox"/>	<a href="#">cs6650-LB-TG</a>	<a href="#">arn:aws:elasticloadbalanci...</a>	80	HTTP	Instance	cs6650
-------------------------------------	------------------------------	---	----	------	----------	--------

### Target group: cs6650-LB-TG

Details **Targets** Monitoring Health checks Attributes Tags

Registered targets (4) [Info](#) [Anomaly mitigation: Not applicable](#) [Deregister](#) [Register targets](#)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.


<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	<a href="#">i-03da4893b60a09bce</a>		8080	us-west-2b	Healthy	-
<input type="checkbox"/>	<a href="#">i-096401af540c3a832</a>		8080	us-west-2b	Healthy	-
<input type="checkbox"/>	<a href="#">i-07d0a35efb6cff6f8</a>	elb1	8080	us-west-2c	Healthy	-
<input type="checkbox"/>	<a href="#">i-0c51d8e854a4645f3</a>	hw	8080	us-west-2a	Healthy	-

## Client number of threads in phase 2: 300

### Queue PostMessageQUEUE


Overview

Queued messages [last ten minutes](#) [?](#)



Ready 0  
Unacked 0  
Total 0

Message rates [last ten minutes](#) [?](#)



Publish 0.00/s  
Deliver (manual ack) 0.00/s  
Deliver (auto ack) 0.00/s

Consumer ack 0.00/s  
Redelivered 0.00/s  
Get (manual ack) 0.00/s

Get (auto ack) 0.00/s  
Get (empty) 0.00/s

Details

Features	State	idle	Messages	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32	<a href="#">?</a>	0	0	0	0	0	0
Operator policy	Consumer capacity	100%	<a href="#">?</a>	0 B	0 B	0 B	0 B	0 B	0 B
Effective policy definition			<a href="#">?</a>	43 KIB					

Consumers (32)

Bindings (1)

```
Multi threads consumer test start
-----
Number of threads in phase 2: 300
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 121547
Mean latency= 82.717205 ms
Median latency= 74.0 ms
P99 latency= 219.0 ms
Throughout: 1652 requests/second
Min latency= 53
Max latency= 1210
-----
Multi threads consumer test end
```

Client number of threads in phase 2: 400


Queue PostMessageQUEUE

Overview

Queued messages

last ten minutes

?



Ready

0

Unacked

0

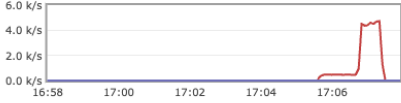
Total

0

Message rates

last ten minutes

?



Publish

0.00/s

Consumer ack

0.00/s

Get (auto ack)

0.00/s

Deliver (manual ack)

0.00/s

Redelivered

0.00/s

Get (empty)

0.00/s

Deliver (auto ack)

0.00/s

Get (manual ack)

0.00/s

Details

Features	State	idle	Messages	?	Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
Policy	Consumers	32	Message body bytes	?	0 B	0 B	0 B	0 B	0 B	0 B
Operator policy	Consumer capacity	100%	Process memory	?	43 KiB					
Effective policy definition										

Consumers (32)

Bindings (1)

```
Multi threads consumer test start
-----
Number of threads in phase 2: 400
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 105496
Mean latency= 84.971395 ms
Median latency= 76.0 ms
P99 latency= 231.0 ms
Throughout: 1904 requests/second
Min latency= 52
Max latency= 1173
-----
Multi threads consumer test end
```

Client number of threads in phase 2: 500

## Queue PostMessageQUEUE

Overview

Queued messages

last ten minutes

?

Ready

Unacked

Total

0

0

0

Message rates

last ten minutes

?

Publish

Deliver (manual ack)

Deliver (auto ack)

0.00/s

0.00/s

0.00/s

Consumer ack

Redelivered

Get (manual ack)

0.00/s

0.00/s

0.00/s

Get (auto ack)

Get (empty)

0.00/s

0.00/s

Details

Features

Policy

Operator policy

Effective policy definition

State

Consumers

Consumer capacity

Idle

0

0%

Messages

Message body bytes

Process memory

0

0 B

26 KiB

Total

Ready

Unacked

In memory

Persistent

Transient, Paged Out

0

0

0

0 B

0 B

0 B


```
Multi threads consumer test start
-----
Number of threads in phase 2: 500
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 106436
Mean latency= 103.18645 ms
Median latency= 88.0 ms
P99 latency= 418.0 ms
Throughout: 1886 requests/second
Min latency= 52
Max latency= 1488
-----
Multi threads consumer test end
```

Client number of threads in phase 2: 800

Queue PostMessageQUEUE

Overview

Queued messages last ten minutes ?



Ready

0

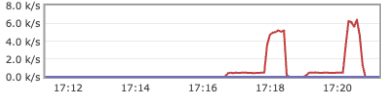
Unacked

0

Total

0

Message rates last ten minutes ?



Publish

0.00/s

Consumer ack

0.00/s

Get (auto ack)

0.00/s

Deliver (manual ack)

0.00/s

Redelivered

0.00/s

Get (empty)

0.00/s

Deliver (auto ack)

0.00/s

Get (manual ack)

0.00/s

Details

Features

Policy

Operator policy

Effective policy definition

State

idle

Consumers

32

Consumer capacity ?

100%

Messages ?

Message body bytes ?

Process memory ?

43 KiB

Consumers (32)

Bindings (1)

```

Multi threads consumer test start
-----
Number of threads in phase 2: 800
Number of successful requests: 200000
Number of fail requests: 0
Wall Time: 100383
Mean latency= 129.456845 ms
Median latency= 106.0 ms
P99 latency= 533.0 ms
Throughout: 2000 requests/second
Min latency= 52
Max latency= 4040
-----
Multi threads consumer test end
-----

```

All the load tests done made sure that the messages in rabbitMQ were less than 1000.

## Throughputs Comparison

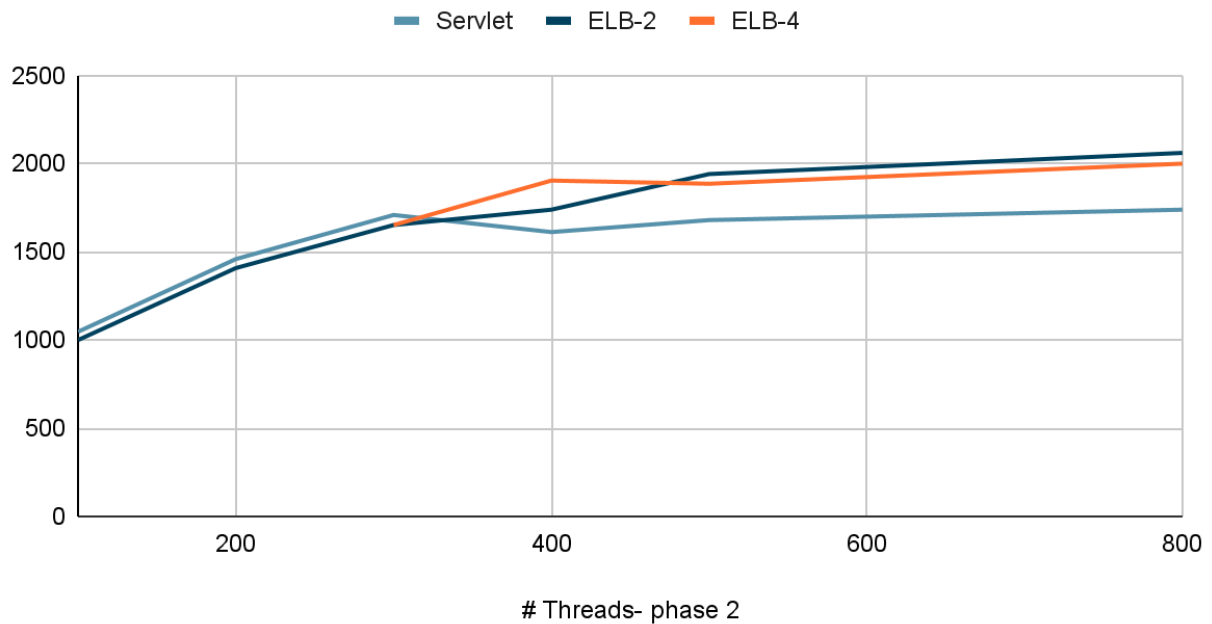
All tests are launched under this condition for the Consumer:

Number of threads: 32

**basicQos = 1**

# Threads- phase 2	Servlet	ELB-2	ELB-4
100	1047	1000	-
200	1459	1408	-
300	1709	1652	1652
400	1612	1739	1904
500	1680	1941	1886
800	1739	2061	2000

## Servlet, ELB-2 and ELB-4



We can conclude from the figure above that using elastic load balance to distribute requests gives us greater throughputs, hence improving the performance of our system. This improvement is particularly more significant when the number of threads on client side gets bigger.

The best throughputs without ELB is around 1700; the best throughputs with ELB is around 2060.

The improvement can be calculated as  $2060/1700 = 1.2$ .

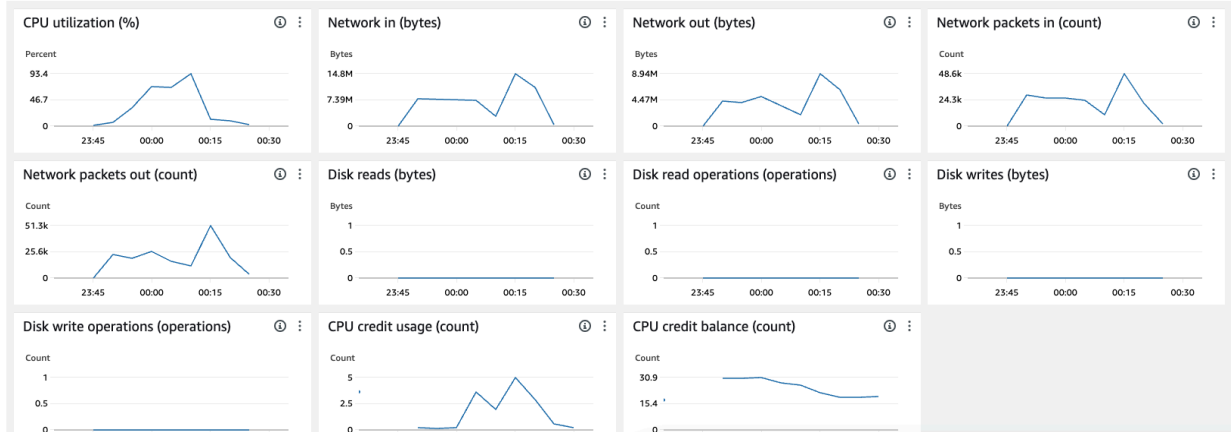
# CPU usage for the Consumer

Instance: i-0172b2201ad97d043 (cs6650Consumer)

Details | Status and alarms **New** | **Monitoring** | Security | Networking | Storage | Tags

Alarm recommendations

1h 3h 12h 1d 3d 1w Custom UTC timezone Add to dashboard



Here we can see that the peak CPU usage of the consumer reached 93.4%, which is way beyond the 60% (normal utilization rate).