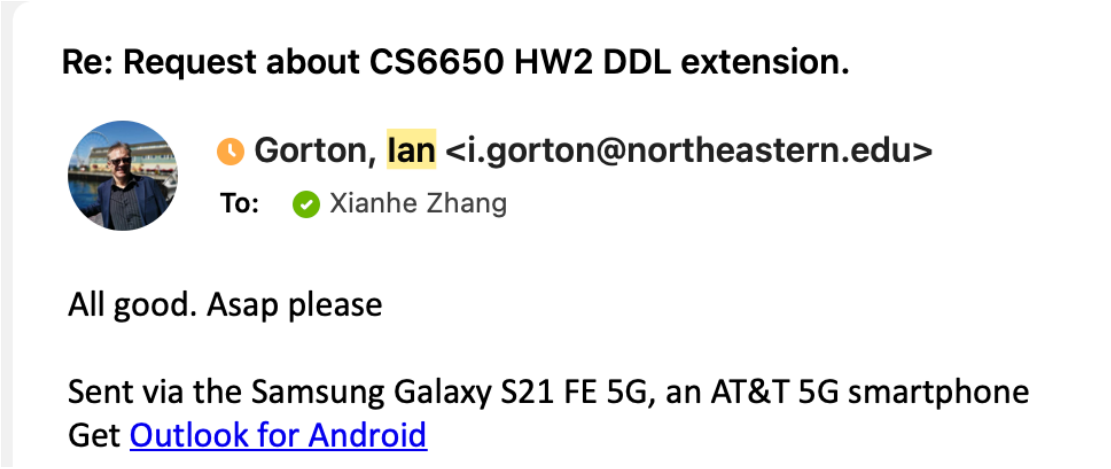
CS6650 Assignment2 Report

Xianhe Zhang

ZHANG.XIANH@

Grading Note

I have talked about late submission with Ian. The below is communication email.



Re: Request about CS6650 HW2 DDL extension.

Gorton, Ian <i.gorton@northeastern.edu>

To: Xianhe Zhang

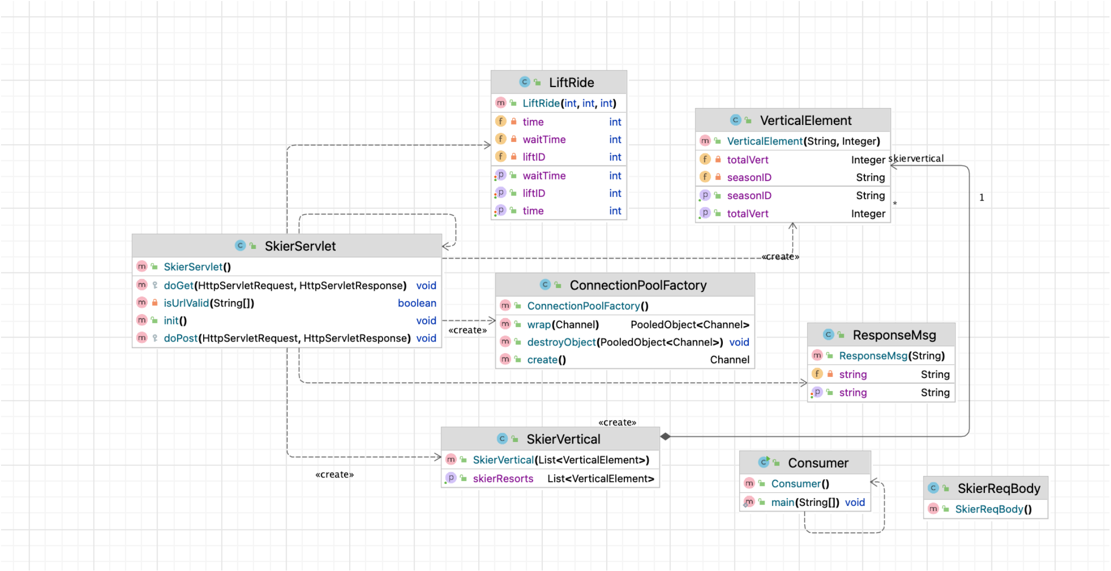
All good. Asap please

Sent via the Samsung Galaxy S21 FE 5G, an AT&T 5G smartphone
Get Outlook for Android

Repo URL

https://github.com/xianhe-zhang/CS6650-22Fall

Design



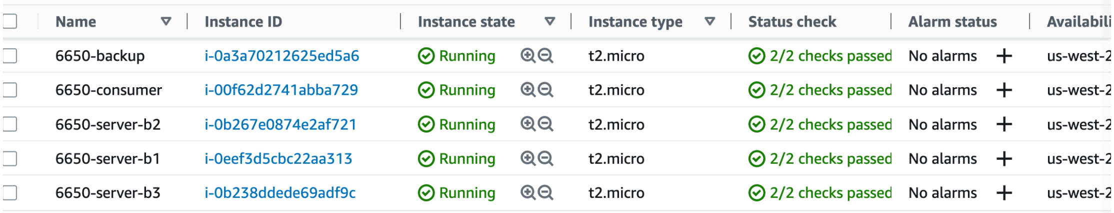
For this assignment, the server design is pretty much like the assignment 1. But I add connection pool to store the connection threads, this will save resources.

For each thread, we need to take one cached thread from our pool and try to connect to our RabbitMQ (the configuration is below). And our consumer is monitoring specific queues.

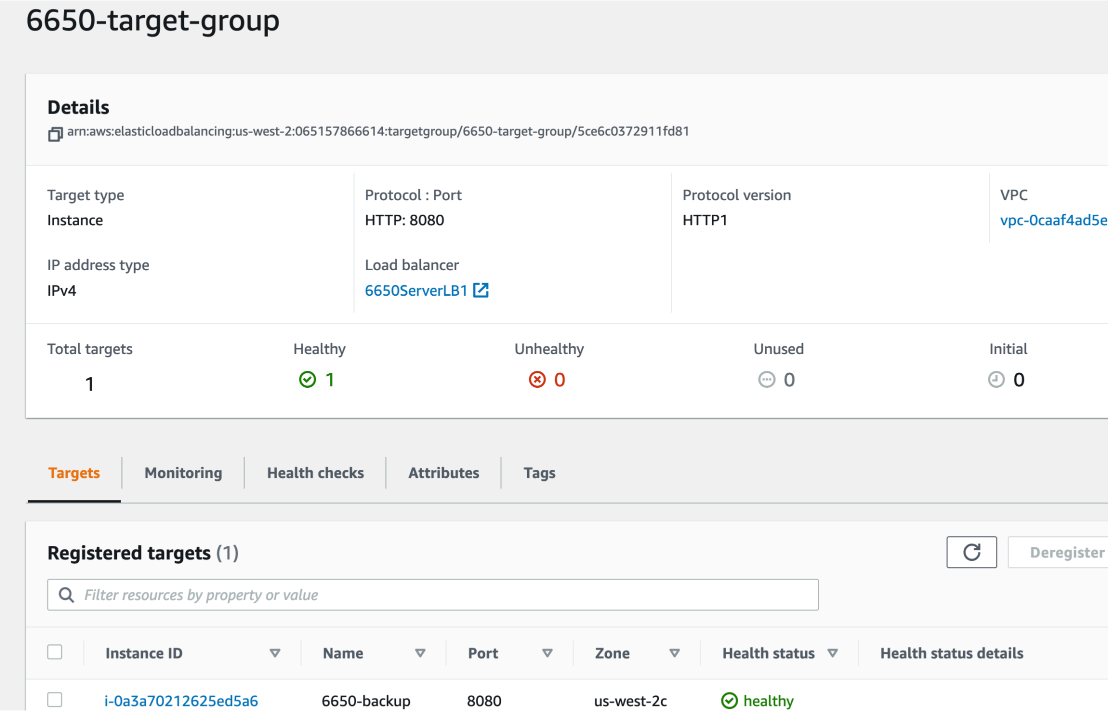
```
        factory.setHost("35.90.118.182");
        factory.setPort(5672);
        factory.setUsername("maXXX");
        factory.setPassword("maXXXXXX");
```

Load Balancer

For this assignment, I tried to set up single-instances, double-instances, four-instances.



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili
☐	6650-backup	i-0a3a70212625ed5a6	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-west-2
☐	6650-consumer	i-00f62d2741abba729	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-west-2
☐	6650-server-b2	i-0b267e0874e2af721	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-west-2
☐	6650-server-b1	i-0eef3d5cbc22aa313	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-west-2
☐	6650-server-b3	i-0b238ddede69adf9c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-west-2



6650-target-group

Details

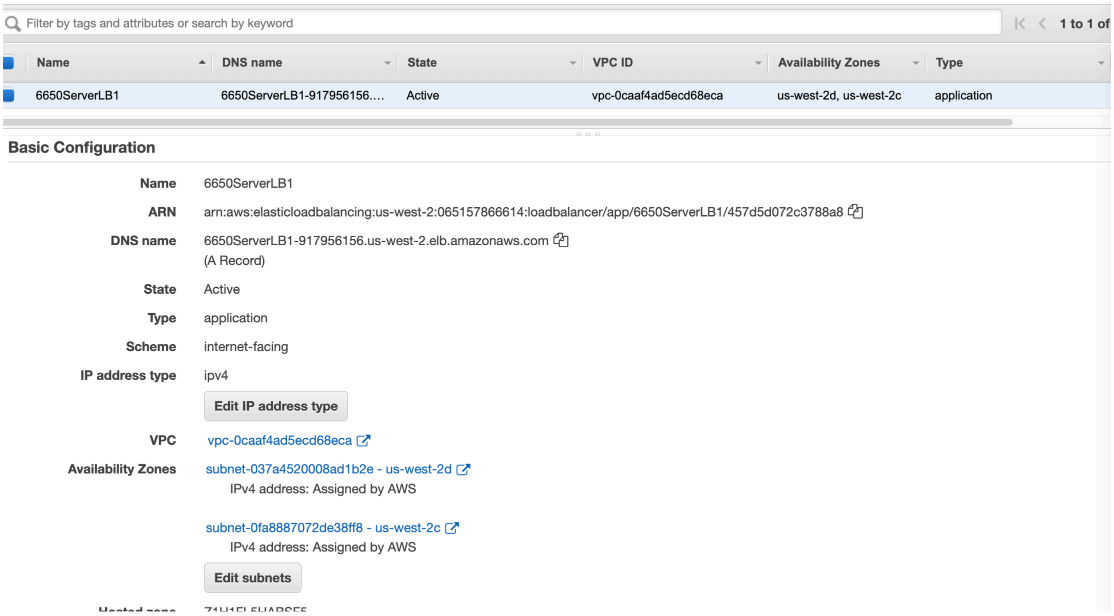
arn:aws:elasticloadbalancing:us-west-2:065157866614:targetgroup/6650-target-group/5ce6c0372911fd81

Target type Instance	Protocol : Port HTTP: 8080	Protocol version HTTP1	VPC vpc-0caaf4ad5e
IP address type IPv4	Load balancer 6650ServerLB1		

Total targets	Healthy	Unhealthy	Unused	Initial
1	✓ 1	⊗ 0	0	0

Registered targets (1)

□	Instance ID	Name	Port	Zone	Health status	Health status details
□	i-0a3a70212625ed5a6	6650-backup	8080	us-west-2c	✓ healthy	



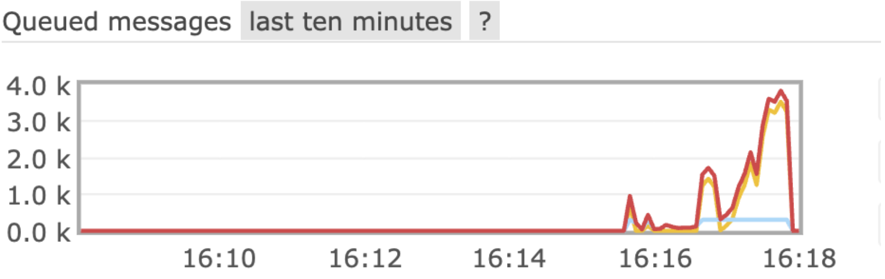
	Name	DNS name	State	VPC ID	Availability Zones	Type
■	6650ServerLB1	6650ServerLB1-917956156....	Active	vpc-0caaf4ad5ecd68eca	us-west-2d, us-west-2c	application

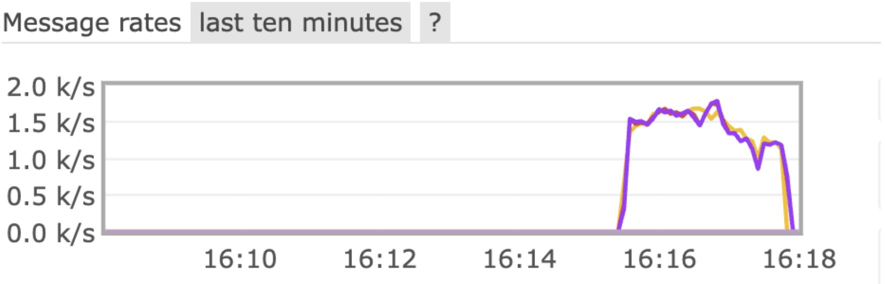
Basic Configuration

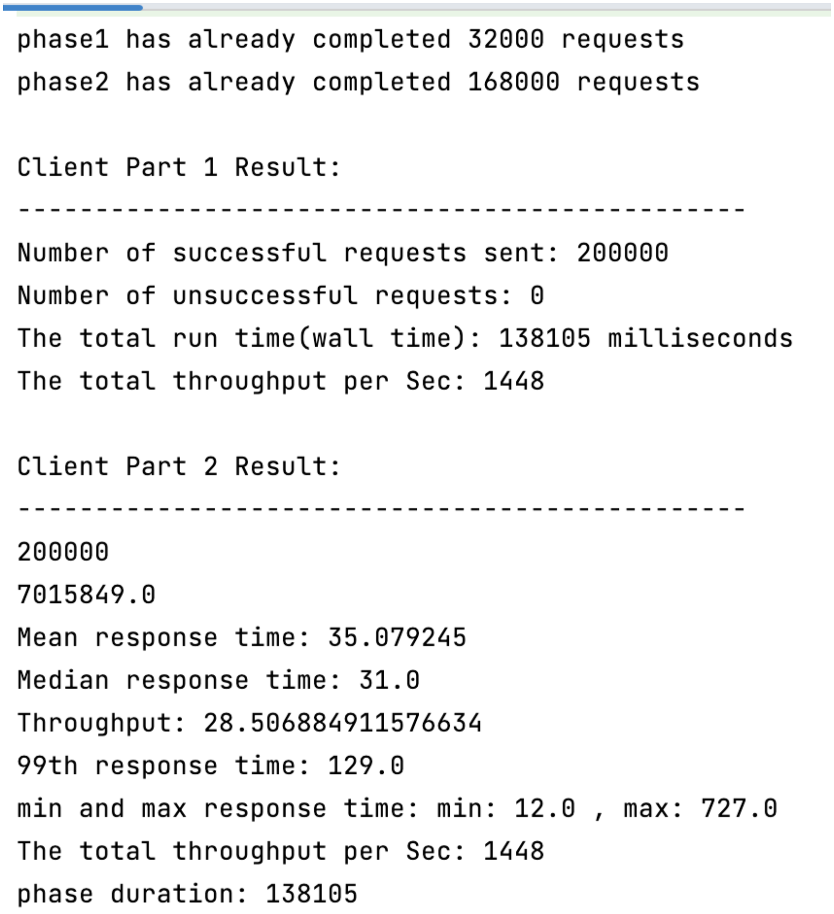
Name	6650ServerLB1
ARN	arn:aws:elasticloadbalancing:us-west-2:065157866614:loadbalancer/app/6650ServerLB1/457d5d072c3788a8
DNS name	6650ServerLB1-917956156.us-west-2.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	ipv4 Edit IP address type
VPC	vpc-0caaf4ad5ecd68eca
Availability Zones	subnet-037a4520008ad1b2e - us-west-2d IPv4 address: Assigned by AWS subnet-0fa8887072de38ff8 - us-west-2c IPv4 address: Assigned by AWS Edit subnets
Hosted zone	Z1H1FL5HABSF5

Result Analysis

SINGLE-INSTANCE





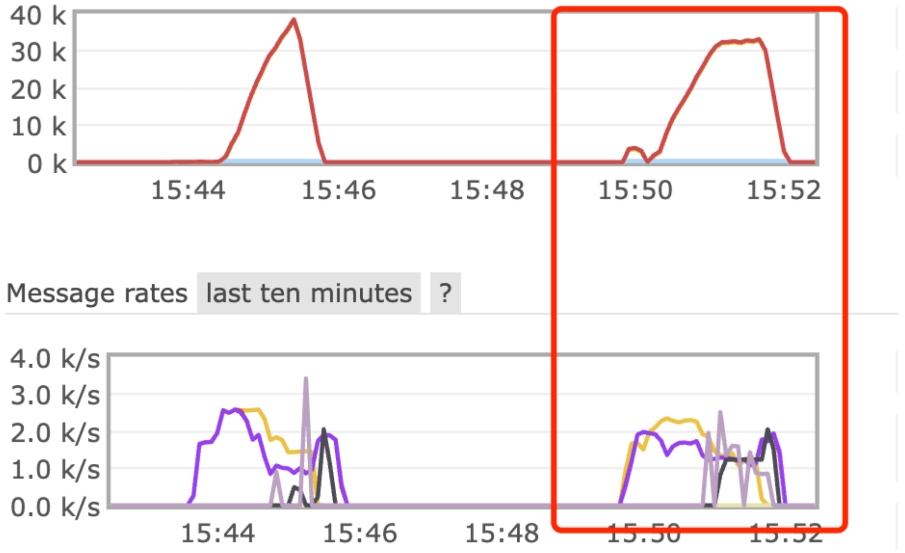


```
phase1 has already completed 32000 requests
phase2 has already completed 168000 requests

Client Part 1 Result:
-----------------------------------------------
Number of successful requests sent: 200000
Number of unsuccessful requests: 0
The total run time(wall time): 138105 milliseconds
The total throughput per Sec: 1448

Client Part 2 Result:
-----------------------------------------------
200000
7015849.0
Mean response time: 35.079245
Median response time: 31.0
Throughput: 28.506884911576634
99th response time: 129.0
min and max response time: min: 12.0 , max: 727.0
The total throughput per Sec: 1448
phase duration: 138105
```

TWO-INSTANCES

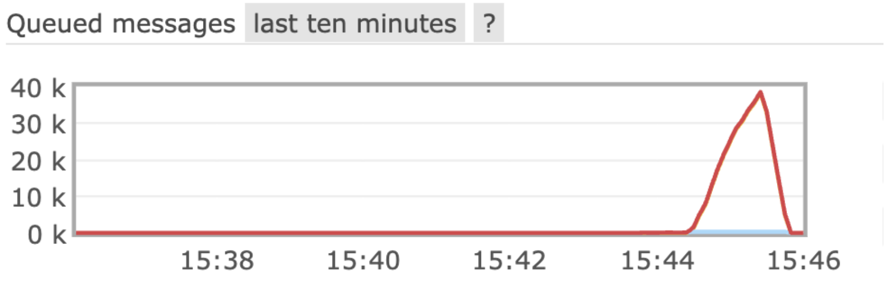


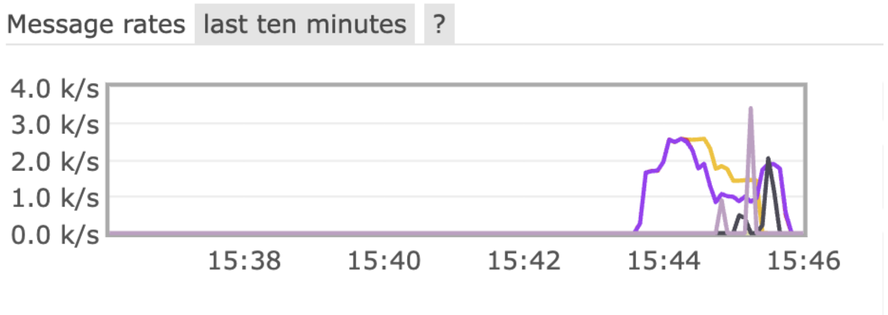
```
phase1 has already completed 32000 requests
phase2 has already completed 168000 requests

Client Part 1 Result:
-----------------------------------------------
Number of successful requests sent: 200000
Number of unsuccessful requests: 0
The total run time(wall time): 113993 milliseconds
The total throughput per Sec: 1754

Client Part 2 Result:
-----------------------------------------------
200000
5726710.0
Mean response time: 28.63355
Median response time: 23.0
Throughput: 34.92406634874125
99th response time: 158.0
min and max response time: min: 12.0 , max: 986.0
The total throughput per Sec: 1754
phase duration: 113993
```

FOUR-INSTANCES


Queued messages last ten minutes ?


Message rates last ten minutes ?

```
phase1 has already completed 32000 requests
phase2 has already completed 168000 requests

Client Part 1 Result:
-----------------------------------------------
Number of successful requests sent: 200000
Number of unsuccessful requests: 0
The total run time(wall time): 102000 milliseconds
The total throughput per Sec: 1960

Client Part 2 Result:
-----------------------------------------------
200000
4999586.0
Mean response time: 24.99793
Median response time: 20.0
Throughput: 40.00331227425631
99th response time: 141.0
min and max response time: min: 11.0 , max: 931.0
The total throughput per Sec: 1960
phase duration: 102000
```

For SERVER, I still have 2 phases like Assignment1. For phase 2, I use **54** threads to send the rest requests.
For CONSUMER, **basicQos == 30,** because publisher rate is way higher than cosumer rate
From the performance screenshot above, we can tell load balancer is good for our throughput, given more instances are backing up, throughput improves accordingly.