

CSE224 HW4

Yawen Zhao A53280596

Provisioning virtual machines

Leader

| | | | |
|-------------------|--|-----------------------|---|
| Instance ID | i-0cfa02e3426be62fa | Public DNS (IPv4) | ec2-34-220-225-90.us-west-2.compute.amazonaws.com |
| Instance state | running | IPv4 Public IP | 34.220.225.90 |
| Instance type | m5.large | IPv6 IPs | - |
| Elastic IPs | | Private DNS | ip-172-31-17-73.us-west-2.compute.internal |
| Availability zone | us-west-2b | Private IPs | 172.31.17.73 |
| Security groups | cse224-fa19 . view inbound rules . view outbound rules | Secondary private IPs | |
| Scheduled events | No scheduled events | VPC ID | vpc-122cbb75 |

Follower

| | | | |
|-------------------|--|-----------------------|--|
| Instance ID | i-0ce45a53da7e01db6 | Public DNS (IPv4) | ec2-52-78-243-169.ap-northeast-2.compute.amazonaws.com |
| Instance state | running | IPv4 Public IP | 52.78.243.169 |
| Instance type | m5.large | IPv6 IPs | - |
| Elastic IPs | | Private DNS | ip-172-31-6-139.ap-northeast-2.compute.internal |
| Availability zone | ap-northeast-2a | Private IPs | 172.31.6.139 |
| Security groups | cse224-fa19 . view inbound rules . view outbound rules | Secondary private IPs | |

Above are the two instance I created. One in Oregon. One in Seoul.

Performing basic benchmarking

```

[ec2-user@ip-172-31-6-139 ~]$ ping 34.220.225.90
PING 34.220.225.90 (34.220.225.90) 56(84) bytes of data.
 64 bytes from 34.220.225.90: icmp_seq=1 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=2 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=3 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=4 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=5 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=6 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=7 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=8 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=9 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=10 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=11 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=12 ttl=232 time=126 ms
 64 bytes from 34.220.225.90: icmp_seq=13 ttl=232 time=126 ms
^C
--- 34.220.225.90 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12013ms
rtt min/avg/max/mdev = 126.197/126.223/126.246/0.311 ms
[ec2-user@ip-172-31-6-139 ~]$ █

```

The RTT I measure is averagely 126.223ms, which is shown as above.

The distance between Seoul and Oregon is 8697km. Given the speed of light in fiber-optic cable which is equal to 2/3 of the speed of light in vacuum, the RTT can be computed as

$$8697 \text{ km} \times 2 / (3.0 \times 10^5 \text{ km/s} \times 2/3) = 86.97 \text{ ms}$$

Benchmarking Kafka

```
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:67
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:74
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:68
turn: cur half rtt:67
```

The mean half RTT calculated by Kafka is 68.3 ms. So the RTT is approximately 136.6ms.

The result is a little bit larger than the result measured by ping method, which may cause by the inconsistent of the clock of the two machine.

Below are the code of producer and consumer:

producer.py

```
from kafka import KafkaConsumer
import time

consumer = KafkaConsumer('testt', bootstrap_servers="52.12.165.225:9092")
sum = 0
i = 0
for msg in consumer:
    i += 1
    cur = int(round(time.time() * 1000)) - int(msg.value)
    sum += cur
    print("%d turn: cur half rtt:%d"%(i,cur))
```

consumer.py

```
from kafka import KafkaConsumer
import time

consumer = KafkaConsumer('testt', bootstrap_servers="52.12.165.225:9092")
sum = 0
i = 0
for msg in consumer:
    i += 1
    cur = int(round(time.time() * 1000)) - int(msg.value)
    sum += cur
    print("%d turn: cur half rtt:%d"%(i,cur))
```