Large Scale Computing

Lab 4 – AWS Cloud (EC2/S3)

Author: Adrian Żerebiec

Assignment 1 - S3 throughput (2p)

Task 1: Results of the data transfer throughput measurements to and from S3.

For this task I used 100 MB file generated with command:

fsutil file createnew large_file.tar.gz 104857600

Measured times

- **PC – S3**
  - ***Upload***: 12,467s
  - ***Download***: 17,663s
- **EC2 – S3**
  - ***Upload***: 2,472s
  - ***Download***: 2,563s

Throughput mathematical formula:

$$\boldsymbol{Throughput} = \frac{\boldsymbol{100MB}}{\boldsymbol{time}\ [\boldsymbol{s}]}\ [\boldsymbol{MB/s}]$$

Throughput values

- **PC – S3**
  - ***Upload***: 8,021 MB/s
  - ***Download***: 5,662 MB/s
- **EC2 – S3**
  - ***Upload***: 40,453 MB/s
  - ***Download***: 39,016 MB/s

Conclusions

Throughput values are significantly higher for data transfer between EC2 and S3. In the case of PC to S3, the upload results are five times worse, and the download results are eight times worse.

Assignment 2 - EC2 auto-scaling (5p)

Task 1: Find or draw an architecture diagram of EC2 autoscaling that shows different resources involved: Autoscaling Group, Launch Template, Load Balancer, Security Groups, EC2 instances, CloudWatch, and Scaling policy. Explain in 1-2 sentences the role of each of these components.

Task 2: CloudWatch charts illustrating the autoscaling (load and count of instances over time).

Diagram

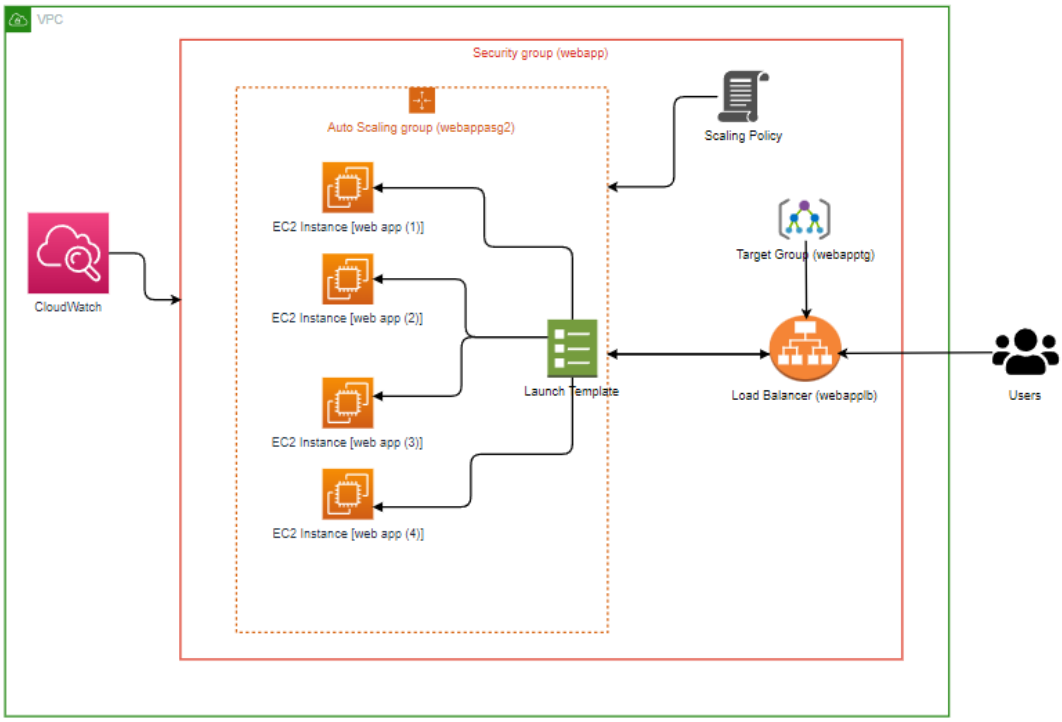

Photo 1: Diagram of created web app

Description

EC2 Instance – A virtual server in AWS that runs applications and services.
Launch Template – A configuration template for EC2 instances, including settings like instance type, AMI, and security groups.
Load Balancer – Distributes incoming network traffic across multiple EC2 instances to improve availability and scalability.
Scaling Policy – Rules that automatically add or remove EC2 instances based on metrics like CPU usage.
Auto Scaling Group – A group of EC2 instances managed automatically to maintain desired performance and availability.
Security Groups – A set of rules that control inbound and outbound traffic to EC2 instances.
Target Group – A collection of EC2 instances or other resources to which the Load Balancer directs traffic.
Users – Users of application.
CloudWatch - Monitoring utility, used to check the telemetry of our services.

CloudWatch charts illustrating the autoscaling

Main instance CPU utilization, Network in/out and Network packets in

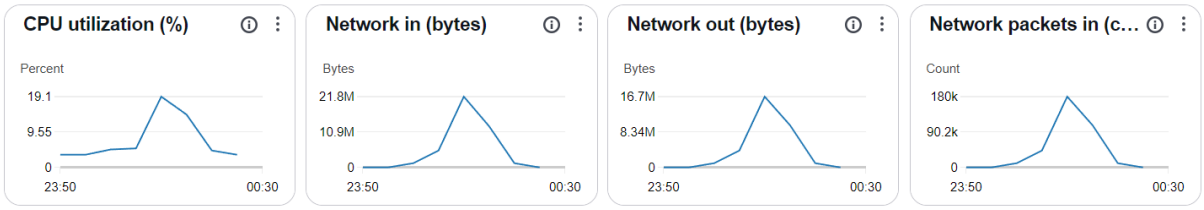

Photo 2: Usage diagrams of main instance web app

CloudWatch GroupDesiredCapacity Diagram

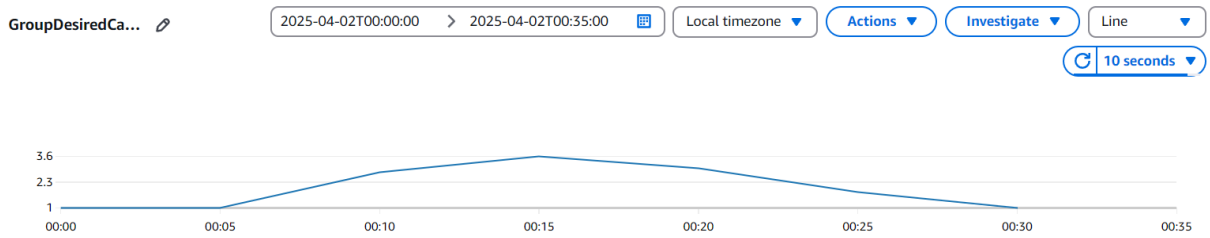

Photo 3: Diagram of GroupDesiredCapacity

CloudWatch GroupDesiredCapacity Peak Diagram

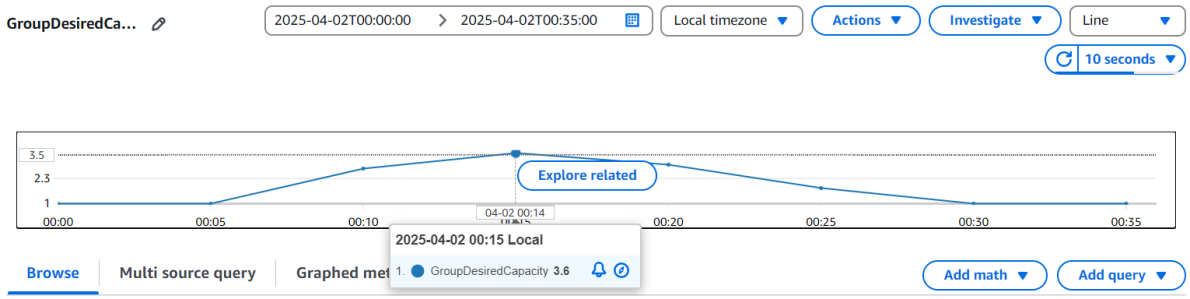

Photo 4: Diagram of GroupDesiredCapacity *Peak (3.6)*

In Service Capacity

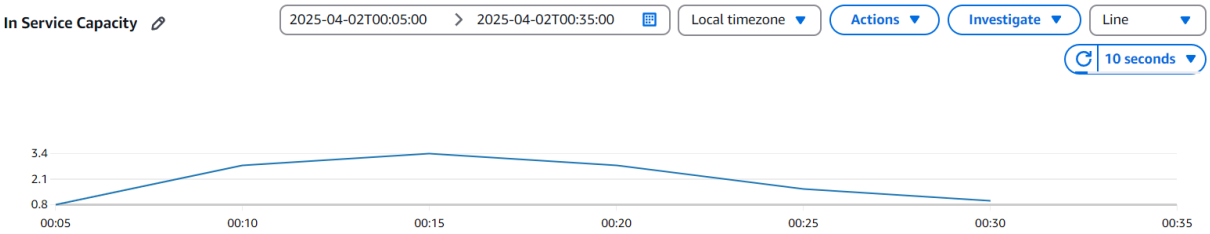

Photo 5: Diagram of In Service Capacity

Conclusions

As we can see, when the load increases, the scaling policy activates new instances to balance the load generated by users. At the peak, the desired capacity was 3.6. In Service Capacity was about 3.4. After the load generation ended, only one instance was in use again. At the critical moment, the CPU load was about 19 percent, and network in and out were above 15M bytes.

Logs form load generator

```
[ec2-user@ip-172-31-82-48 ~]$ ab -c 10 -n 100000 http://webapplb-1721776426.us-east-1.elb.amazonaws.com/phpinfo.php
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking webapplb-1721776426.us-east-1.elb.amazonaws.com (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests


Server Software:        awselb/2.0
Server Hostname:        webapplb-1721776426.us-east-1.elb.amazonaws.com
Server Port:            80

Document Path:          /phpinfo.php
Document Length:        122 bytes

Concurrency Level:      10
Time taken for tests:   213.464 seconds
Complete requests:      100000
Failed requests:        0
Non-2xx responses:      100000
Total transferred:      27200000 bytes
HTML transferred:       12200000 bytes
Requests per second:    468.46 [#/sec] (mean)
Time per request:       21.346 [ms] (mean)
Time per request:       2.135 [ms] (mean, across all concurrent requests)
Transfer rate:          124.44 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    9  10.9      1    1099
Processing:     1   12  73.4     16    7289
Waiting:        0   12  73.4     16    7289
Total:          1   21  74.4     20    7309

Percentage of the requests served within a certain time (ms)
  50%     20
  66%     22
  75%     38
  80%     39
  90%     40
  95%     40
  98%     40
  99%     40
 100%   7309 (longest request)
```

Photo 6: Logs of load generator