

# Large Scale Computing

## Lab 4

Adrian Madej

### 1. S3 throughput (2p)

Check id

```
PS C:\Users\ltmol> aws sts get-caller-identity
```

GetCallerIdentity	
Account	656247241280
Arn	arn:aws:sts::656247241280:assumed-role/voclabs/user3574227=Adrian_Madej
UserId	AROAZRS3CHJAOXJFIHJ45:user3574227=Adrian_Madej

Create S3 bucket

```
PS C:\Users\ltmol> aws s3api create-bucket --bucket lsc-2025-adrian --region us-east-1
```

CreateBucket	
Location	/lsc-2025-adrian

Local PC

```
PS C:\Users\ltmol> Measure-Command { aws s3 cp large_file.tar.gz s3://lsc-2025-adrian/ }
```

Days	: 0
Hours	: 0
Minutes	: 0
Seconds	: 21
Milliseconds	: 574
Ticks	: 215746642
TotalDays	: 0,000249706761574074
TotalHours	: 0,00599296227777778
TotalMinutes	: 0,3595777366666667
TotalSeconds	: 21,5746642
TotalMilliseconds	: 21574,6642

Upload  $\approx$  4.63 MB/s

## Local PC

```
PS C:\Users\ltmol> Measure-Command { aws s3 cp s3://lsc-2025-adrian/large_file.tar.gz . }

Days           : 0
Hours          : 0
Minutes        : 0
Seconds        : 19
Milliseconds    : 46
Ticks          : 190463360
TotalDays      : 0,000220443703703704
TotalHours     : 0,005290648888888889
TotalMinutes   : 0,3174389333333333
TotalSeconds   : 19,046336
TotalMilliseconds : 19046,336
```

Download  $\approx$  5.25 MB/s

## EC2 instance

```
[ec2-user@ip-172-31-42-138 ~]$ time aws s3 cp large_file.tar.gz s3://lsc-2025-adrian/
upload: ./large_file.tar.gz to s3://lsc-2025-adrian/large_file.tar.gz

real    0m3.648s
user    0m0.863s
sys     0m0.431s
```

Upload  $\approx$  27.4 MB/s

## EC2 instance

```
[ec2-user@ip-172-31-42-138 ~]$ time aws s3 cp s3://lsc-2025-adrian/large_file.tar.gz .
download: s3://lsc-2025-adrian/large_file.tar.gz to ./large_file.tar.gz

real    0m2.785s
user    0m0.985s
sys     0m0.275s
```

Download  $\approx$  35.9 MB/s

## Throughput

	Upload [MB/s]	Download [MB/s]
Local PC	4.63	5.25
EC2 instance	27.4	35.9

## 2. EC2 auto-scaling (5p)

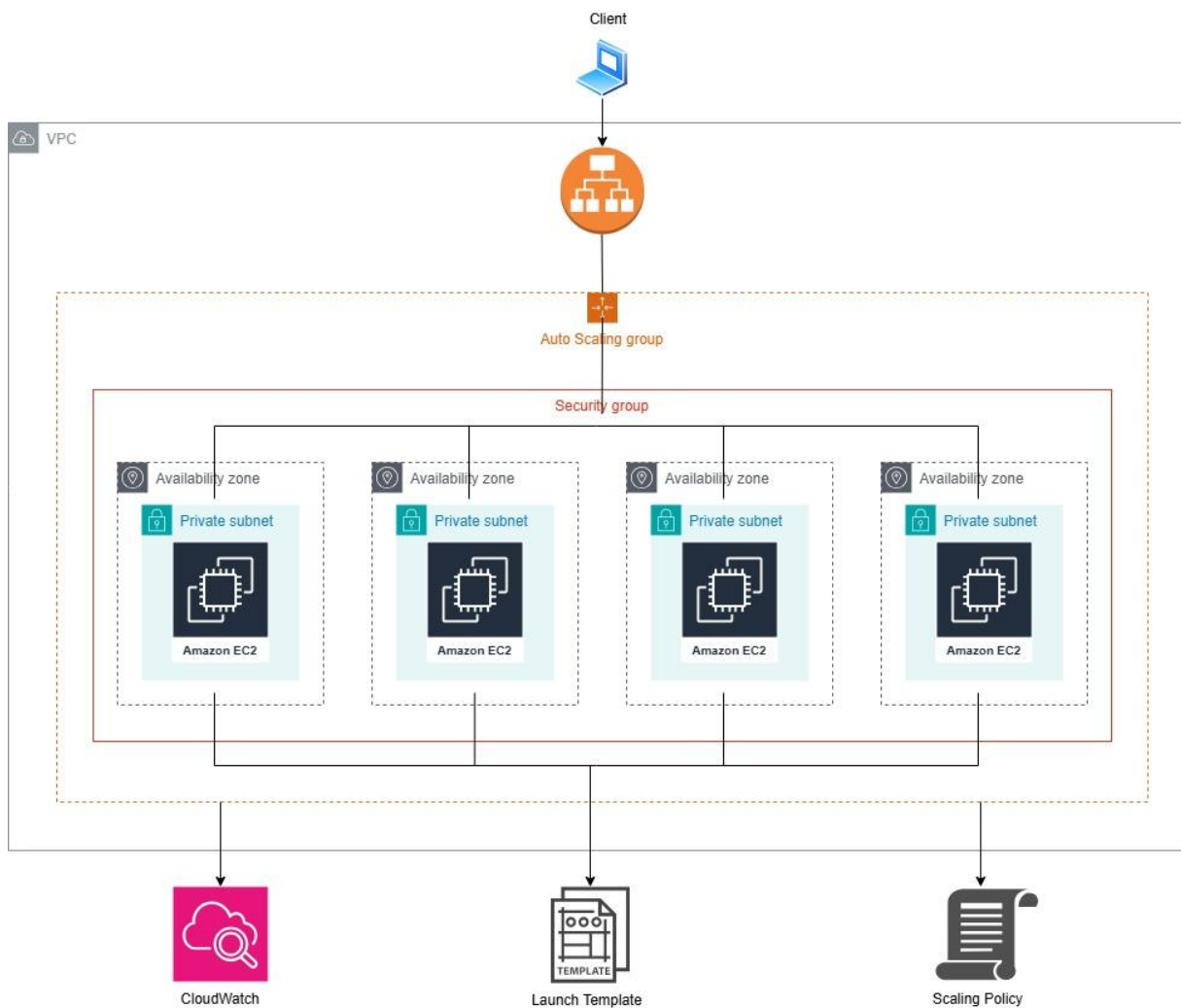
After opening: <http://<machine-ip>/phpinfo.php>



PHP Version 8.2.27

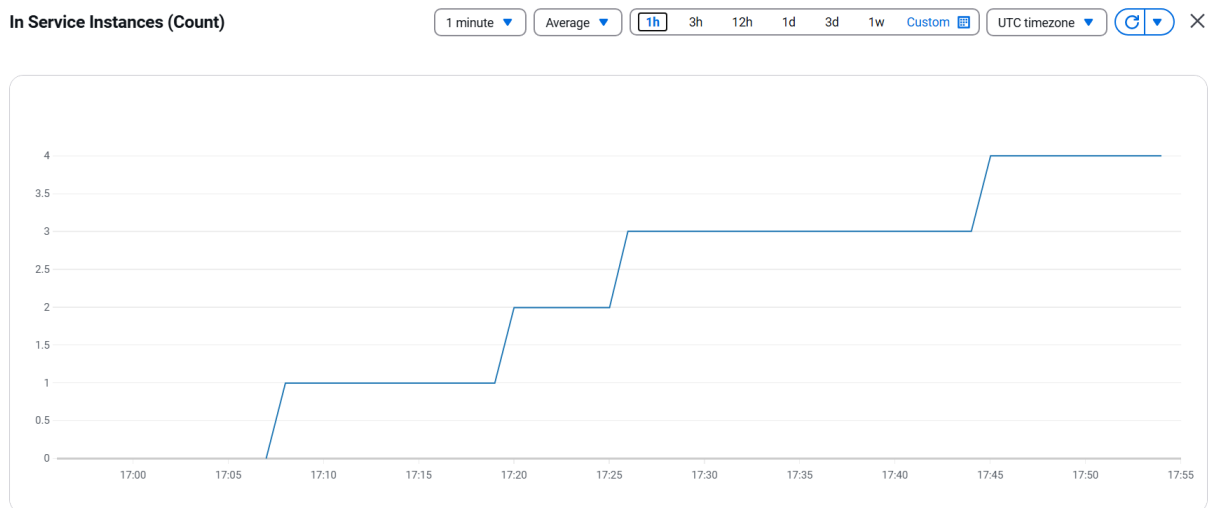
System	Linux ip-172-31-88-39.ec2.internal 6.1.130-139.222.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Mar 11 01:10:58 UTC 2025 x86_64
Build Date	Dec 17 2024 11:39:23
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.4.1 20230605 (Red Hat 11.4.1-2)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bc2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-ffi.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-imagick.ini, /etc/php.d/20-ldap.ini, /etc/php.d/20-ldap_sasl.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mcrypt.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-sysmsg.ini, /etc/php.d/20-syssem.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlrpc.ini, /etc/php.d/20-zlib.ini

### Architecture diagram of EC2



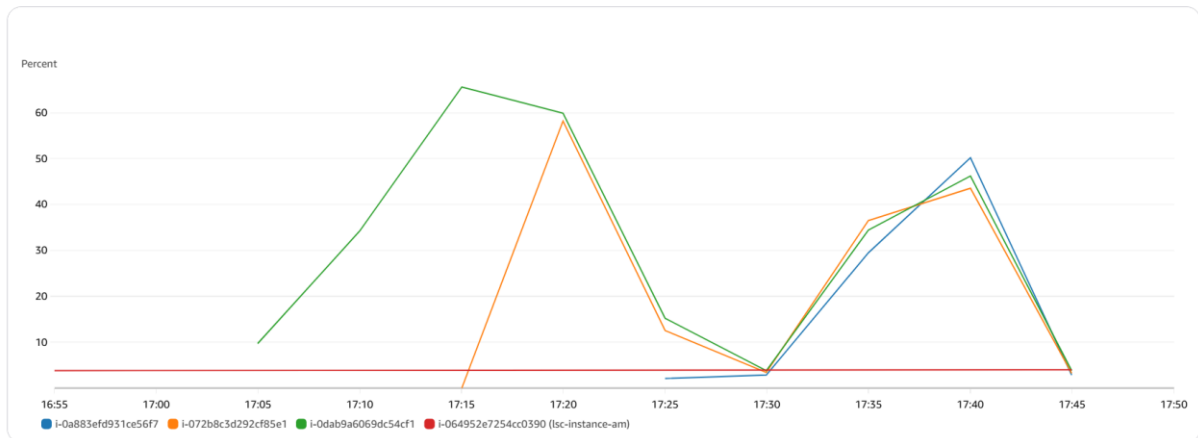
- **Autoscaling Group:** Manages a group of EC2 instances, automatically scaling them up or down based on demand to maintain availability and performance.
- **Launch Template:** Defines the configuration (AMI, instance type, key pair, etc.) for EC2 instances launched within an ASG.
- **Load Balancer:** Distributes incoming traffic across multiple EC2 instances to ensure high availability and reliability.
- **Security Groups:** Act as virtual firewalls, controlling inbound and outbound traffic to EC2 instances.
- **EC2 Instances:** Virtual servers that run applications and services in AWS.
- **CloudWatch:** Monitors metrics, logs, and alarms for AWS resources to provide observability and automated responses to changes.
- **Scaling Policy:** Defines the conditions and rules for automatically adjusting the number of instances in an ASG based on CloudWatch metrics.

## CloudWatch charts



## CPU Utilization: Average

5 minutes ▾ Average ▾ 1h 3h 12h 1d 3d 1w Custom [table icon] UTC timezone [refresh icon] [close icon]



## Logs

```
ltmollo@DESKTOP-2QCT8V6:~$ ab -c 100 -n 10000 http://lsc-load-balancer-920270590.us-east-1.elb.amazonaws.com/phpinfo.php
```

This is ApacheBench, Version 2.3 <\$Revision: 1879490 \$>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, <http://www.zeustech.net/>

Licensed to The Apache Software Foundation, <http://www.apache.org/>

Benchmarking lsc-load-balancer-920270590.us-east-1.elb.amazonaws.com (be patient)

Completed 1000 requests

Completed 2000 requests

Completed 3000 requests

Completed 4000 requests

Completed 5000 requests

Completed 6000 requests

Completed 7000 requests

Completed 8000 requests

Completed 9000 requests

Completed 10000 requests

Finished 10000 requests

Server Software: Apache/2.4.62  
Server Hostname: lsc-load-balancer-920270590.us-east-1.elb.amazonaws.com  
Server Port: 80

Document Path: /phpinfo.php  
Document Length: 76660 bytes

Concurrency Level: 100  
Time taken for tests: 309.256 seconds  
Complete requests: 10000  
Failed requests: 9982  
(Connect: 0, Receive: 0, Length: 9982, Exceptions: 0)  
Total transferred: 768170814 bytes  
HTML transferred: 766640814 bytes  
Requests per second: 32.34 [#/sec] (mean)  
Time per request: 3092.564 [ms] (mean)  
Time per request: 30.926 [ms] (mean, across all concurrent requests)  
Transfer rate: 2425.71 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	110	174 103.0	152	1842
Processing:	451	2905 647.2	3041	9899
Waiting:	182	2238 442.7	2369	5465
Total:	563	3080 653.0	3211	10309

Percentage of the requests served within a certain time (ms)

50% 3211

66% 3301

75% 3359

80% 3398

90% 3521

95% 3677

98% 4049

99% 4498

100% 10309 (longest request)