AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

COM202

Drawing the New York City skyline with Amazon Aurora Serverless v2

Renato Losio (he/him)
Principal Cloud Architect, AWS Data Hero
Funambol

Agenda

What is a serverless database?

Before serverless

Amazon Aurora Serverless

Aurora capacity units: v1 versus v2

Can I draw the New York City skyline?

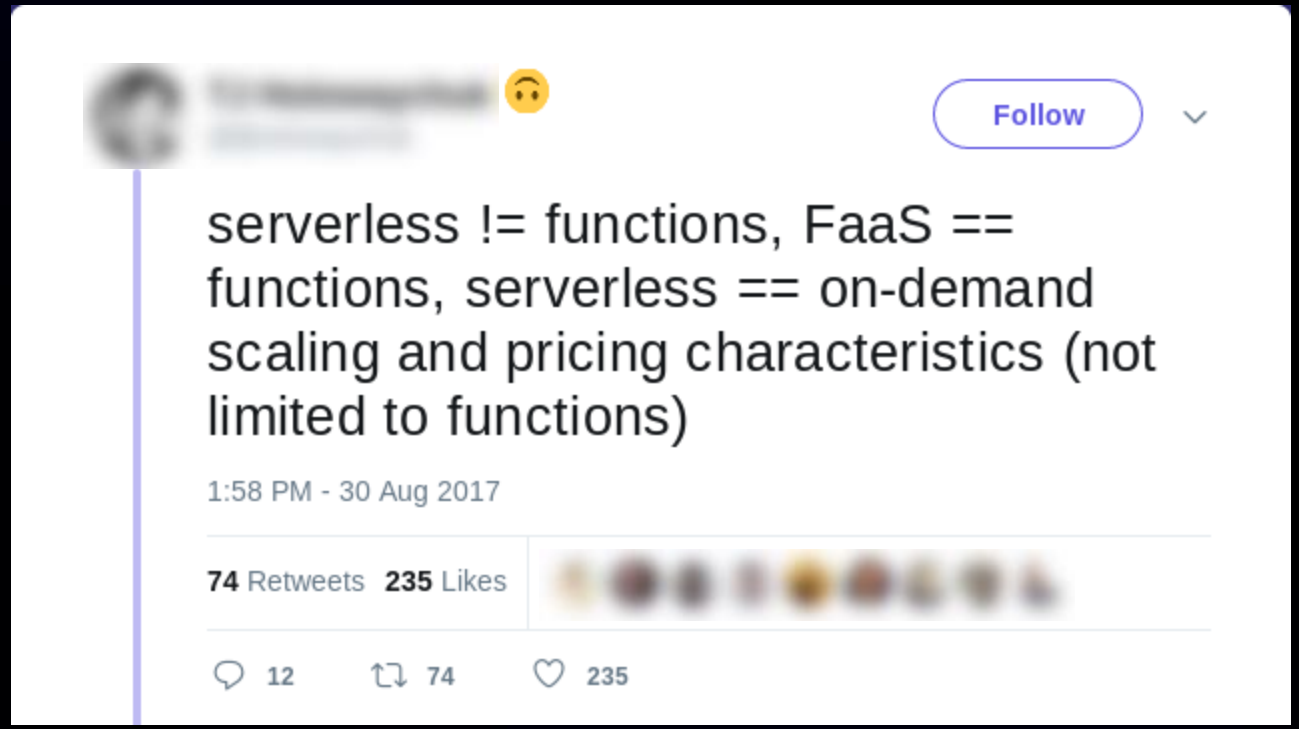
Elasticity and results

What is a serverless database?

Serverless database

- High availability and fault tolerance
- No server maintenance
- Charge for usage
- Continuous scaling

Photo by Andres Siimon on Unsplash



The dream

A relational (MySQL) database cluster that automatically starts up, shuts down, and scales up or down capacity based on my application's needs

Photo by Jeremy Bishop on Unsplash

Main use cases

- Intermittent load
- Development/test environments
- Serverless deployments
- Unpredictable load
- Handling bugs

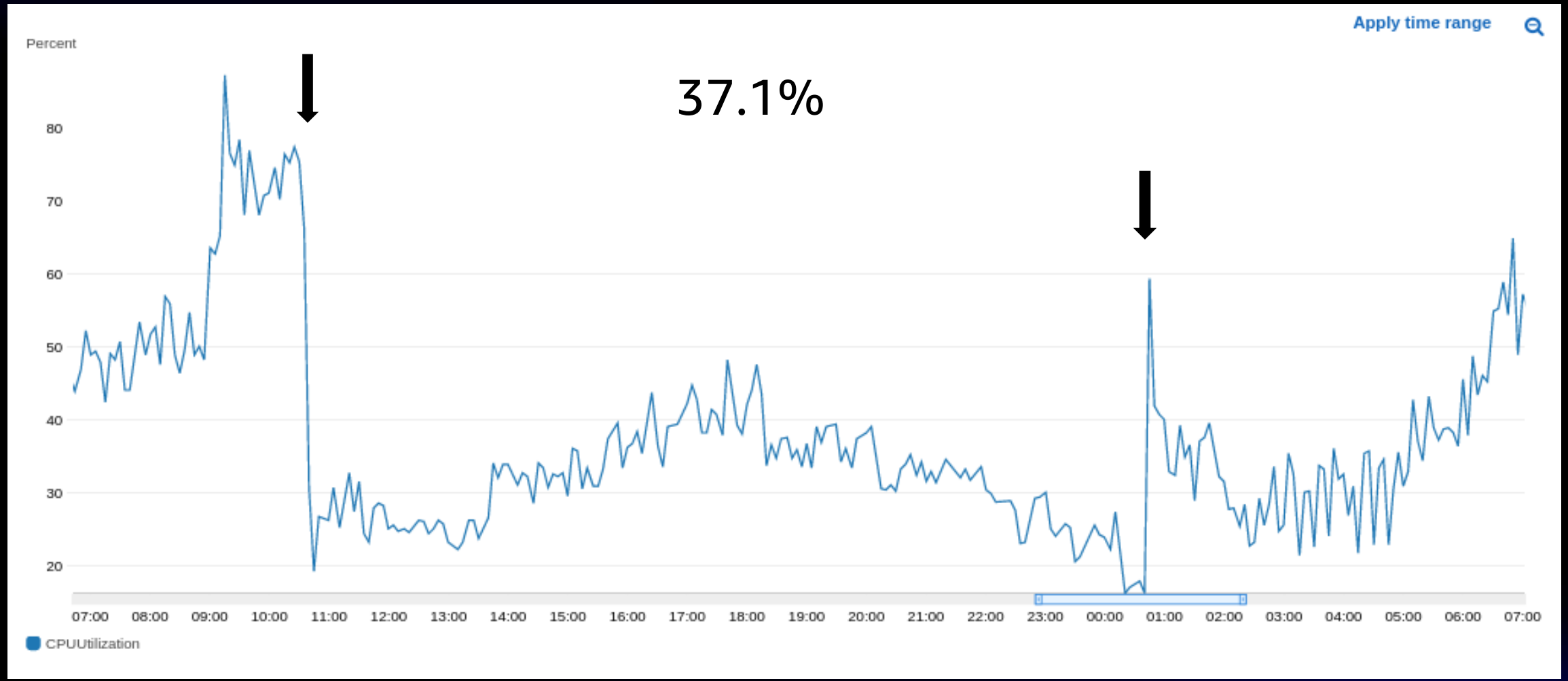
Photo by Ellie Edwards on Unsplash

Before serverless

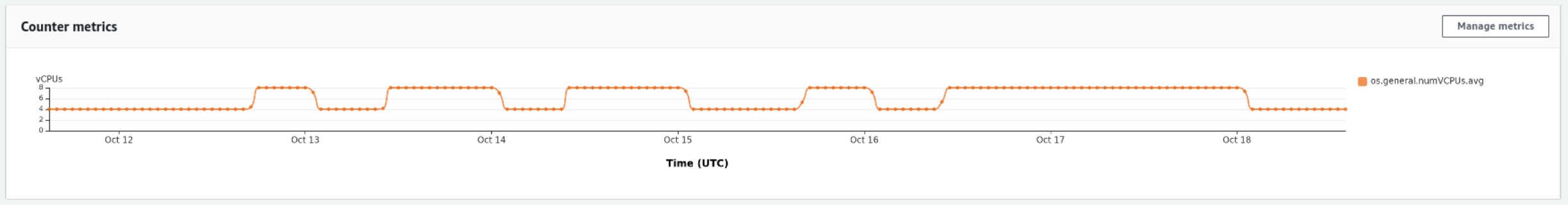
Amazon RDS scaling with AWS CLI

```
cpu_status_alarm=$(aws cloudwatch describe-alarms --alarm-names  "demo-rds01-alarm-
75"  --query "MetricAlarms[].StateValue" | grep -c 'ALARM')
(…)

if [ "$cpu_status_alarm" = "1" ]; then

   if [[ "db.m5.2xlarge" == "$current_instance" ]]; then
      new_instance_type="db.m5.4xlarge"
   elif (...)

   aws rds modify-db-instance --db-instance-identifier \
    "demo-rds01" --db-instance-class "$new_instance_type" \
    --apply-immediately
fi
```

Amazon RDS for MySQL scaling



Amazon RDS for MySQL scaling



Can we be more elastic?

- Can we scale Amazon RDS more often?
- Can we reduce the computational costs?

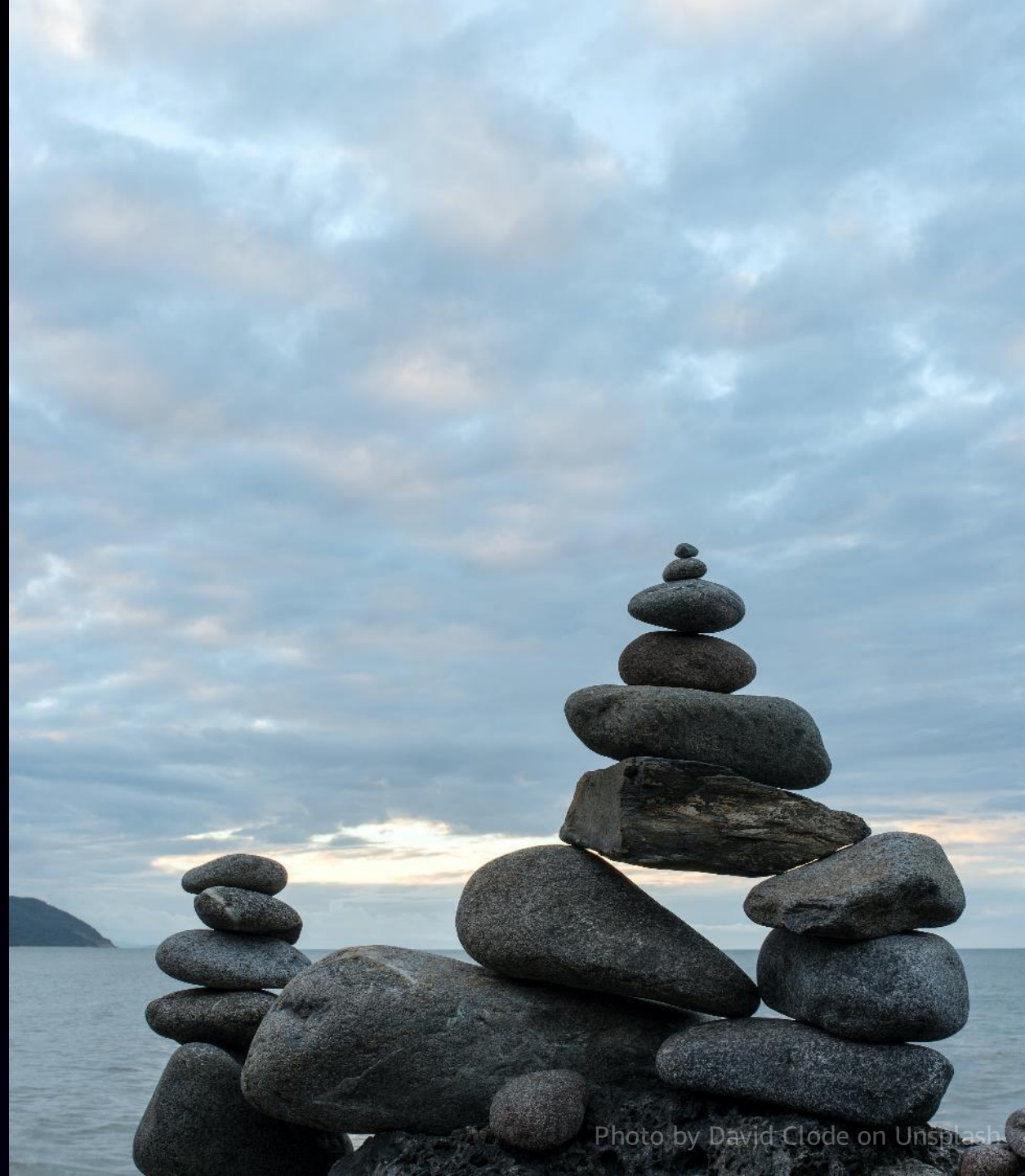
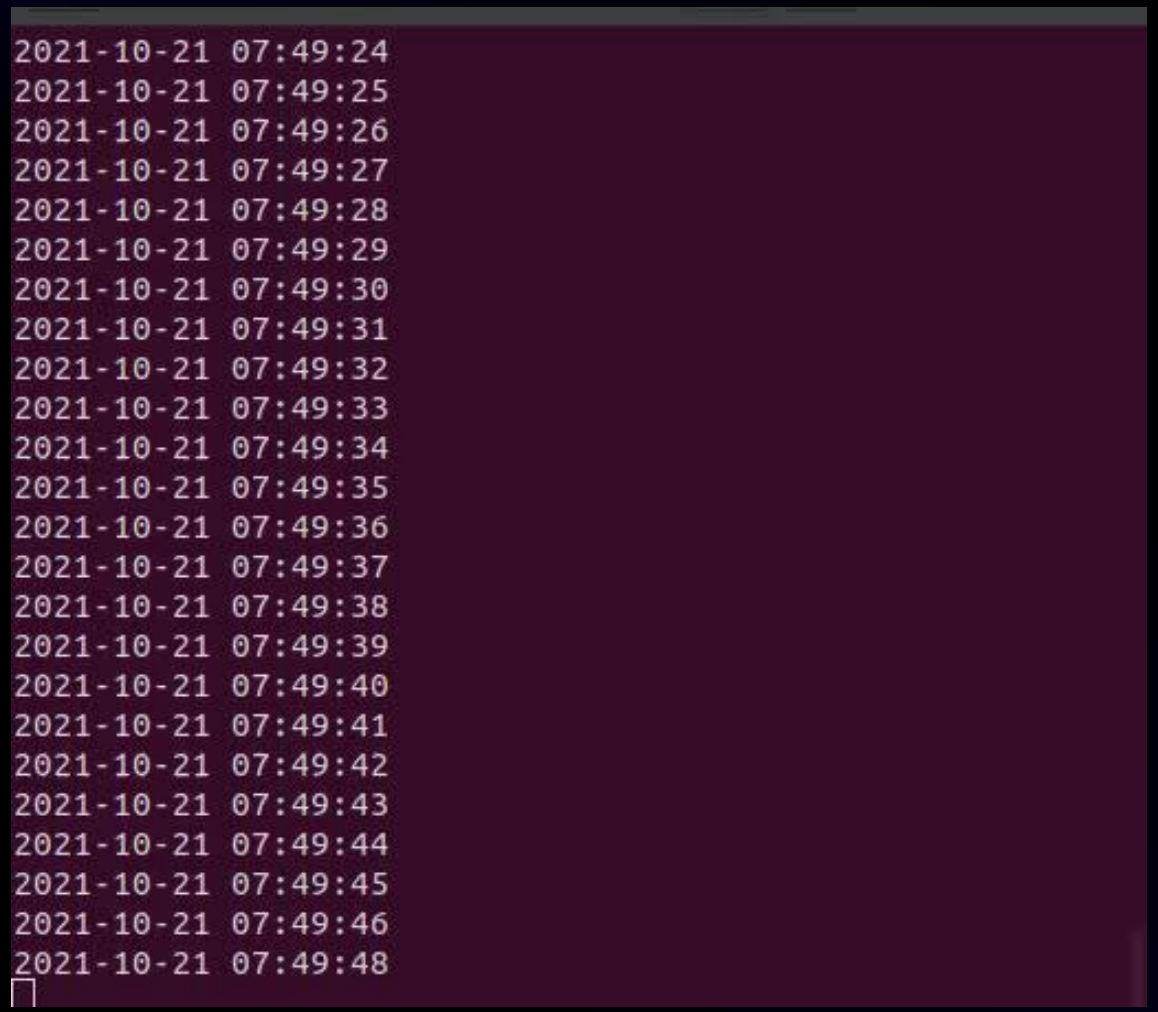


Photo by David Clode on Unsplash

Amazon RDS for MySQL availability



select now();

reboot, scale up or down

Amazon RDS computational costs

	2019	2020	2021
CPU	54%	47%	33%
SSD	31%	21%	24%
Backup	15%	32%	42%

What about NoSQL?

- Knowledge
- Reuse
- Application design
- ACID

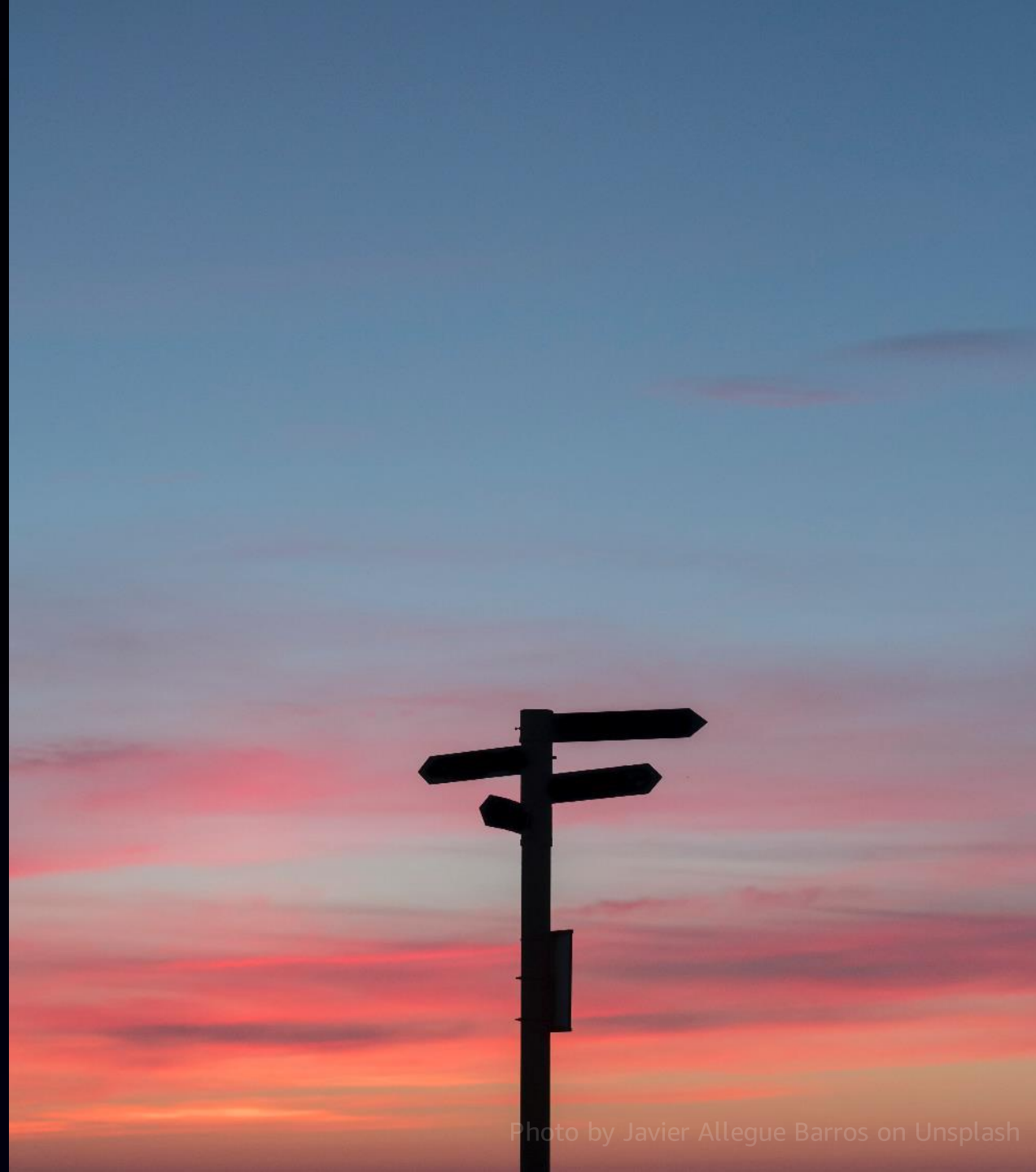


Photo by Javier Allegue Barros on Unsplash

Amazon Aurora Serverless

Amazon Aurora Serverless v1

- On-demand, auto scaling configuration
- Endpoint without size
- Scaling according to CPU and connections
- MySQL GA since 2018, PostgreSQL since 2019
- 1-256 ACU (up to 488 GiB RAM)
- Using TCP connections or data API

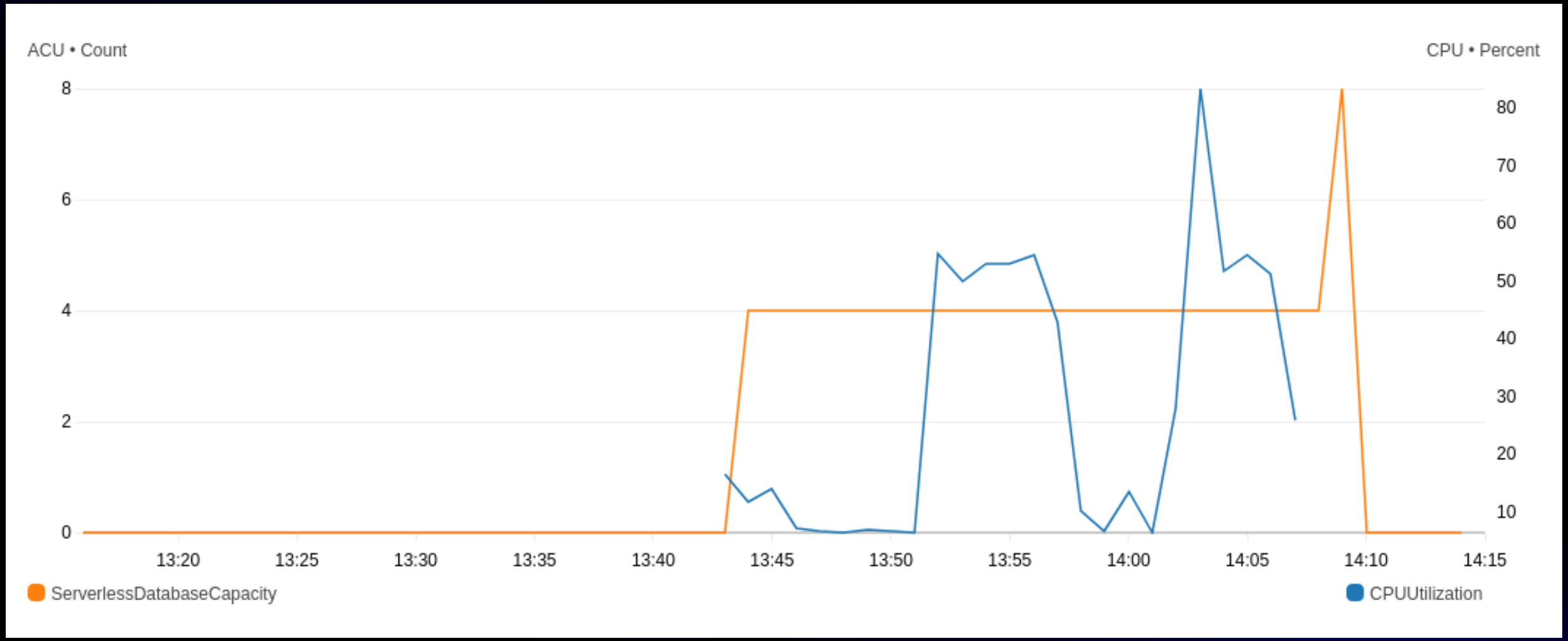
Test procedure

```
DELIMITER $$
CREATE PROCEDURE generate_load()
BEGIN
  DECLARE i INT DEFAULT 0;
  WHILE i < 5000000 DO
    INSERT INTO `serverless_data` (`datetime`,`value`) VALUES (
      FROM_UNIXTIME(UNIX_TIMESTAMP('2021-10-01 01:00:00')+FLOOR(RAND()*31536000)),
      ROUND(RAND()*100,2));
    SET i = i + 1;
  END WHILE;
END$$
DELIMITER ;
```

Test load

```
--
-- Nothing more that 2 runs
-- with a 5 minute break
--

CALL generate_load();
SELECT sleep(300);
CALL generate_load();
```

Elasticity: Aurora Serverless v1



Aurora Serverless v2 (and reinvented)

Aurora capacity units (ACU)

Aurora Serverless v1

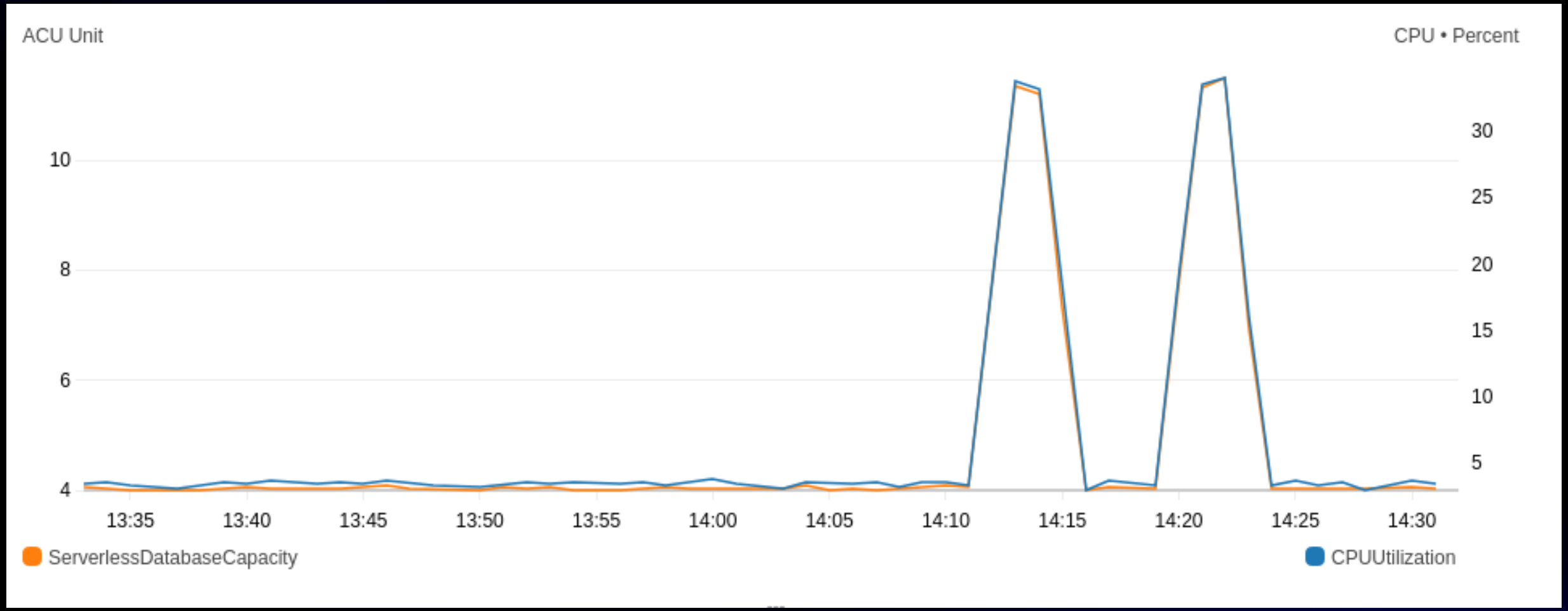
1 2 4 8 16 32 64 128 256

Aurora Serverless v2

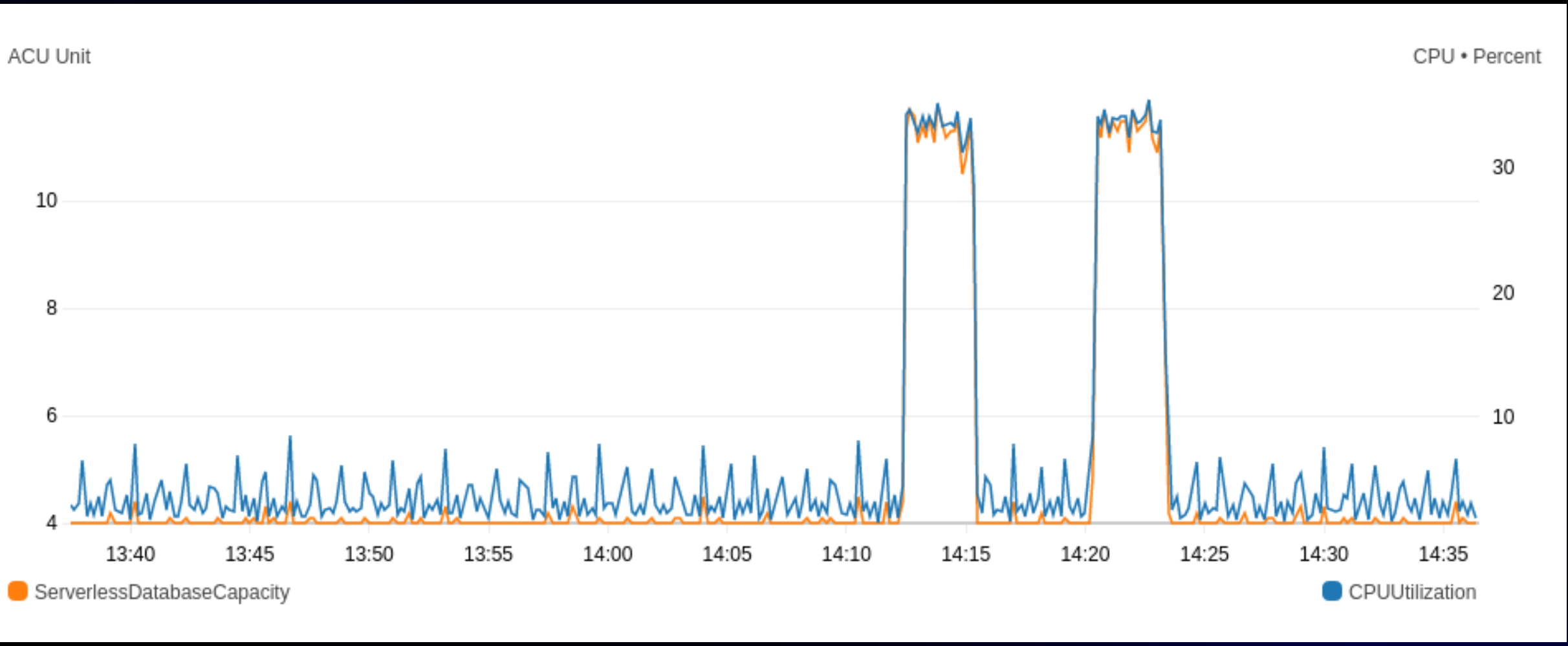
0.5 1 1.5 2 … 256

Photo by Mak on Unsplash

Elasticity: Aurora Serverless v2



Elasticity: Aurora Serverless v2



Execution time

Aurora Serverless v1 (4 ACU)

```
real    15m48.517s
user    0m0.007s
sys     0m0.007s
```

10 min 48 sec

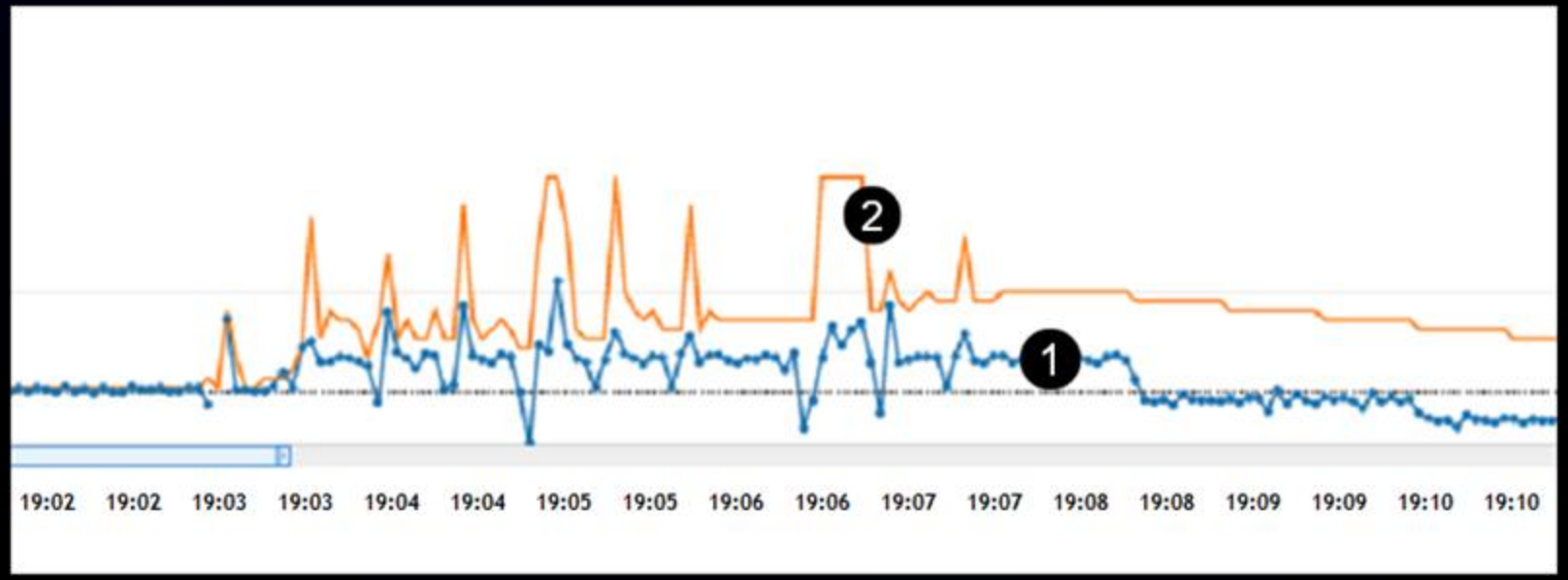
Aurora Serverless v2 in preview (4 ACU)

```
real    10m55.151s
user    0m0.015s
sys     0m0.000s
```

5 min 55 sec

Simulated “flash sale” scenario

AWS DOCUMENTATION

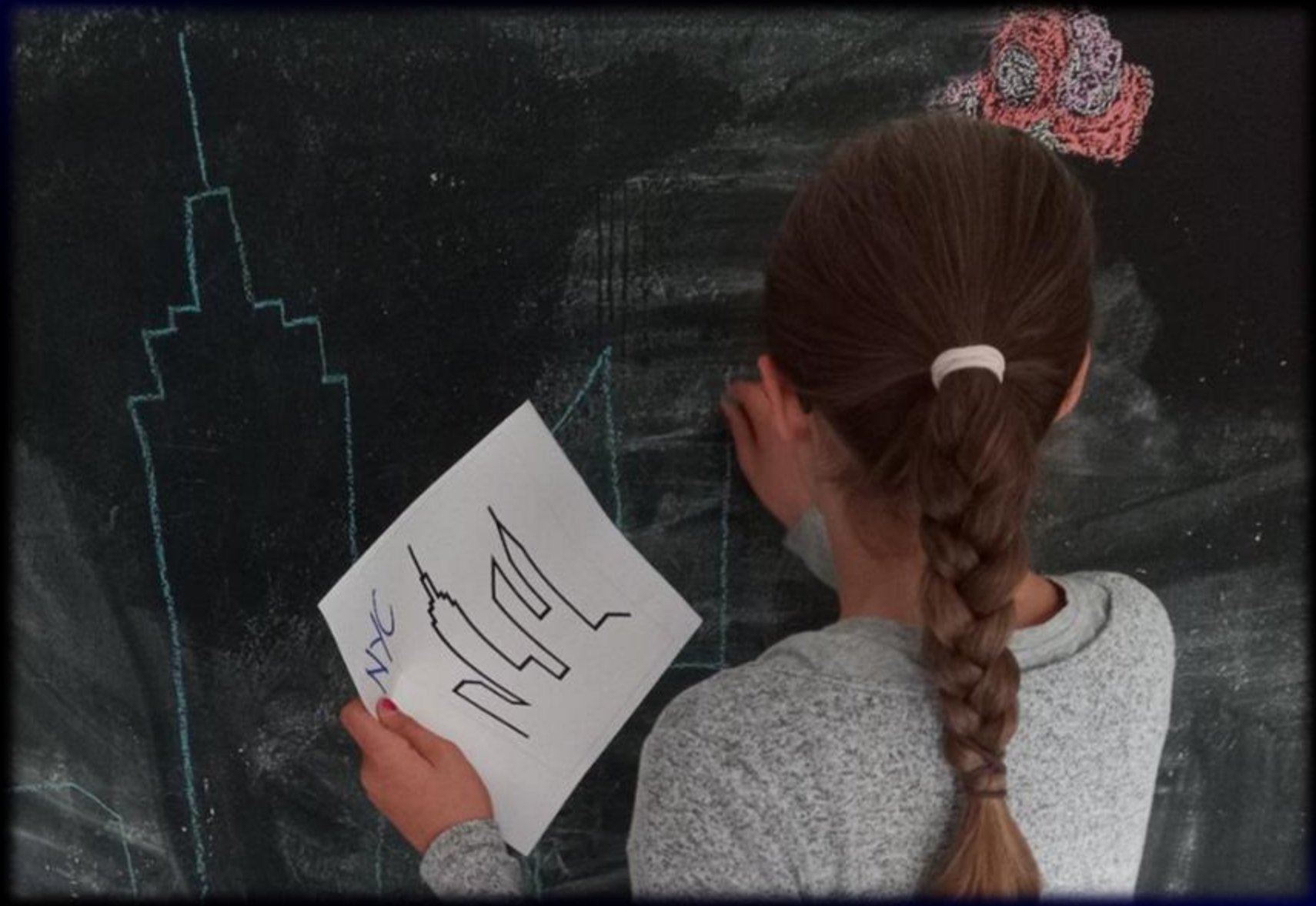


1) Orders processed each second
2) Aurora capacity units (ACUs)

Creative benchmarking

New York City skyline

Photo by Rohit Tandon on Unsplash



The rules

- Simple skyline
- SQL only, no other languages
- No tools
- Better than my daughter?

How can I draw?

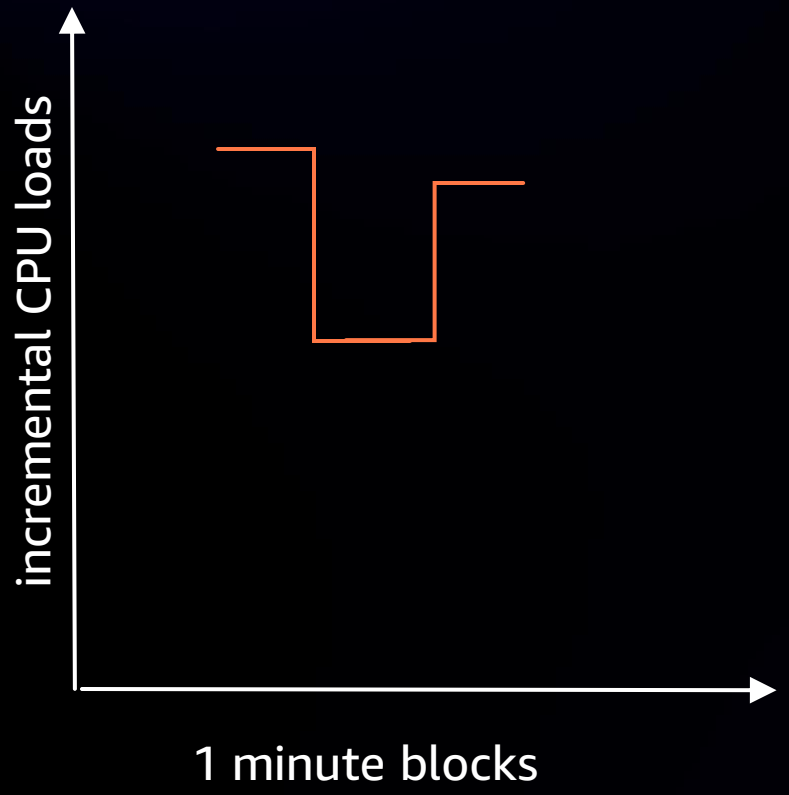
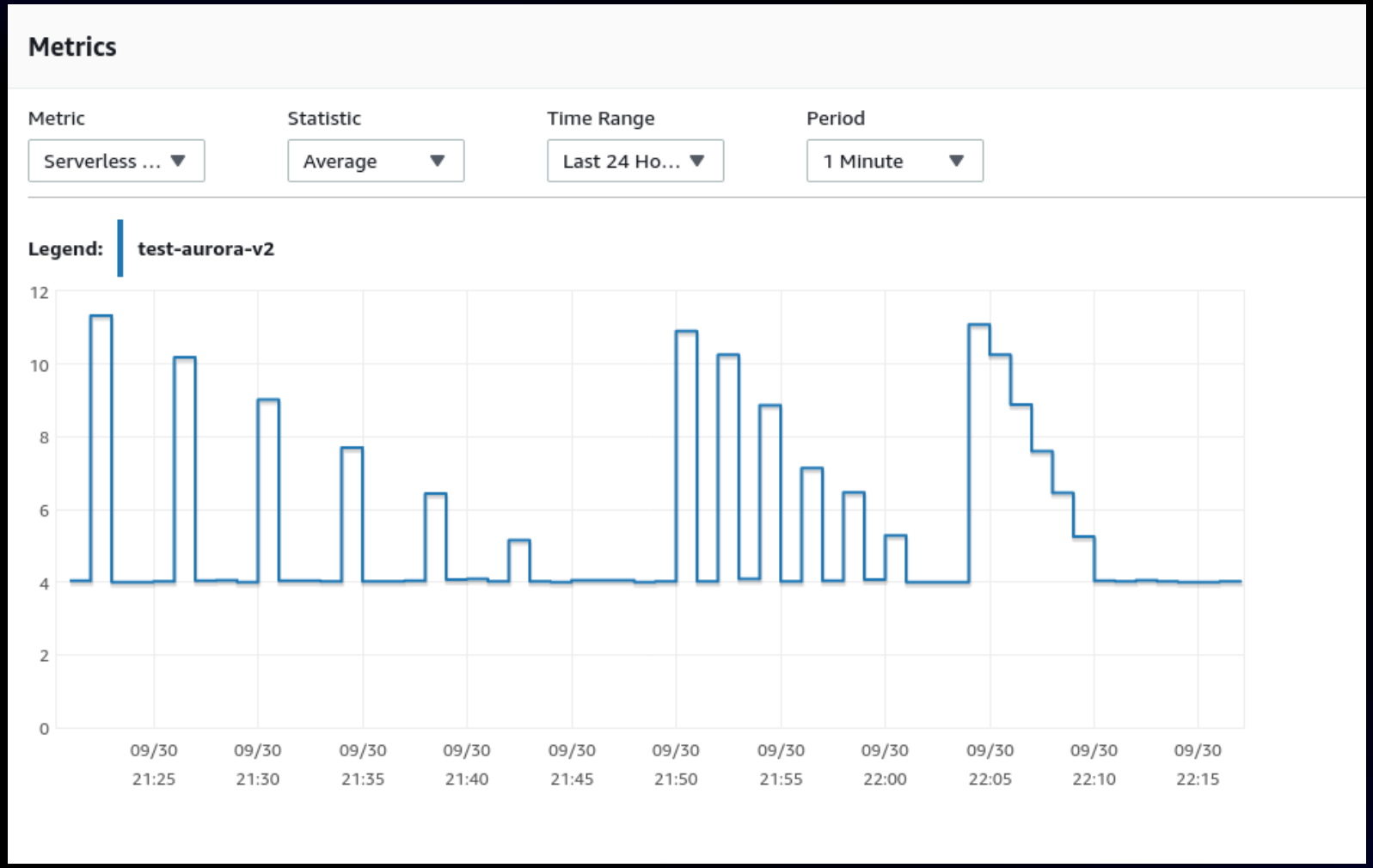


Photo by Luca Nicoletti on Unsplash

Draw block procedure

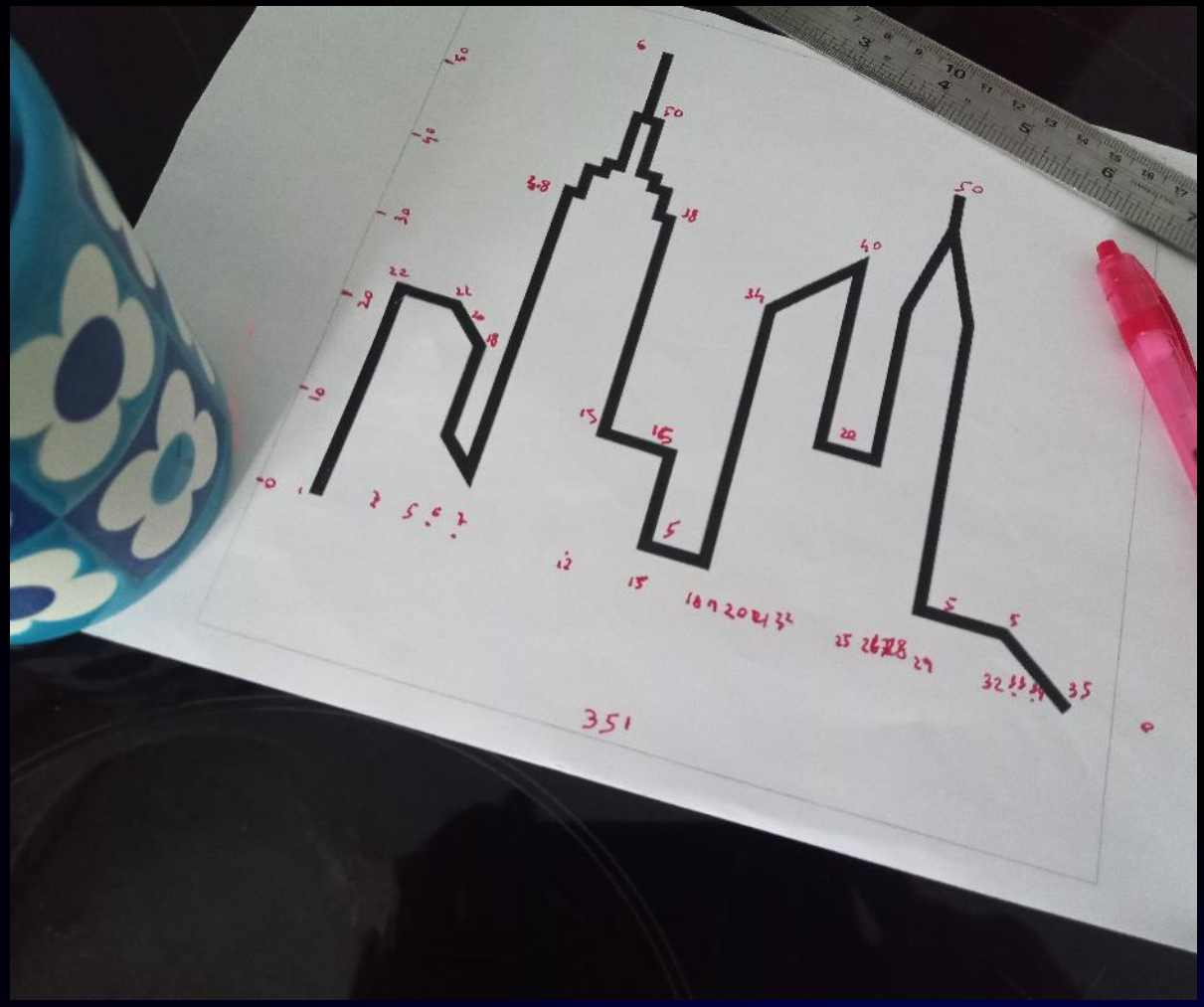
```
CREATE PROCEDURE draw_block(IN x INT)
BEGIN

    WHILE MOD(UNIX_TIMESTAMP(),60) <= x DO
          INSERT INTO `nyc` (`datetime`,`value`) VALUES (
            FROM_UNIXTIME(UNIX_TIMESTAMP('2021-10-01
01:00:00')+FLOOR(RAND()*31536000)),
            ROUND(RAND()*100,2));
    END WHILE;

    -- padding to one minute
    SELECT SLEEP(60-x);

END$$
```

Building blocks

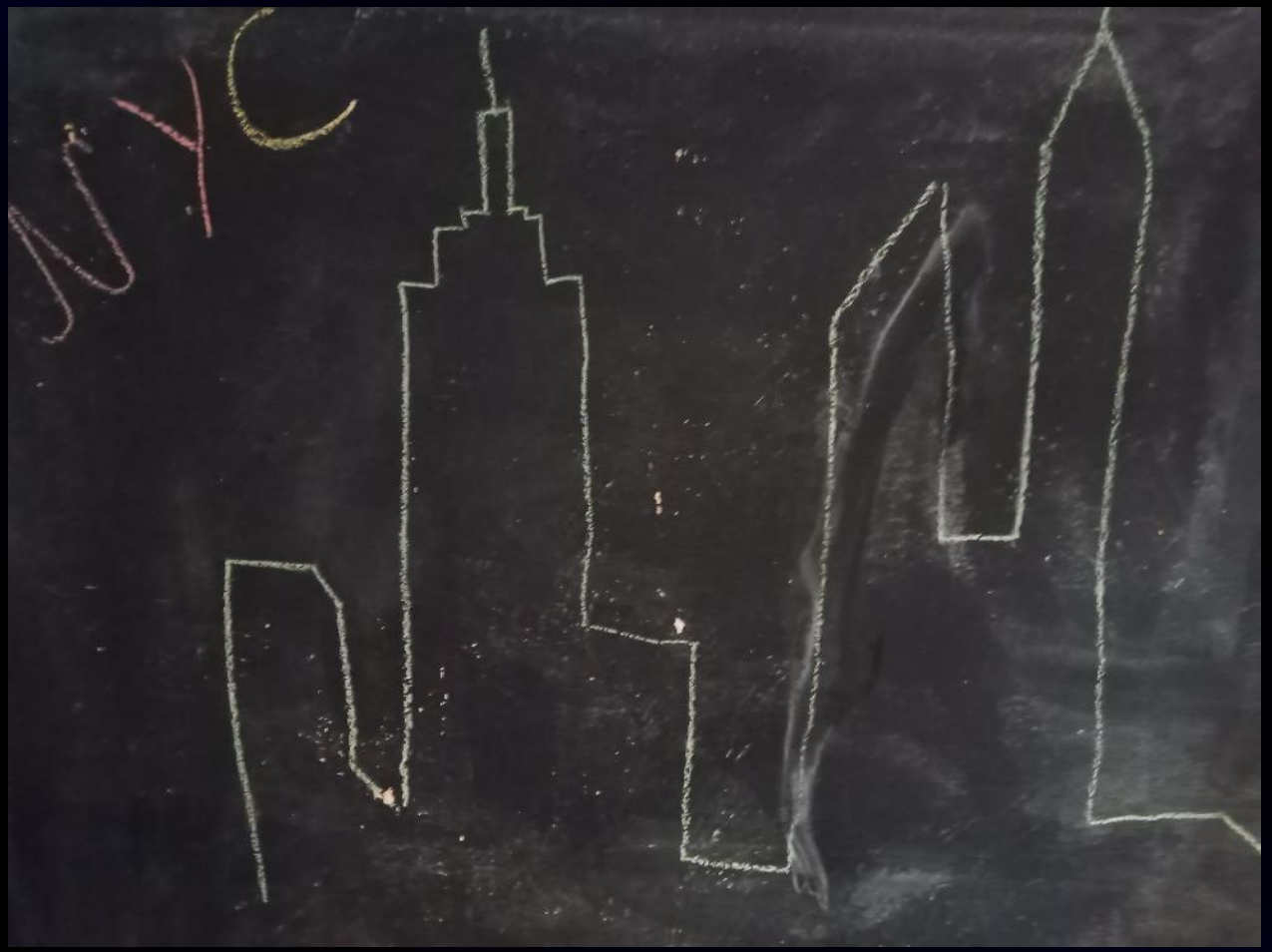


Drawing

```
(...)
call draw_block(7);
call draw_block(5);
call draw_block(38);
call draw_block(40);
call draw_block(42);
call draw_block(50);
(...)
```

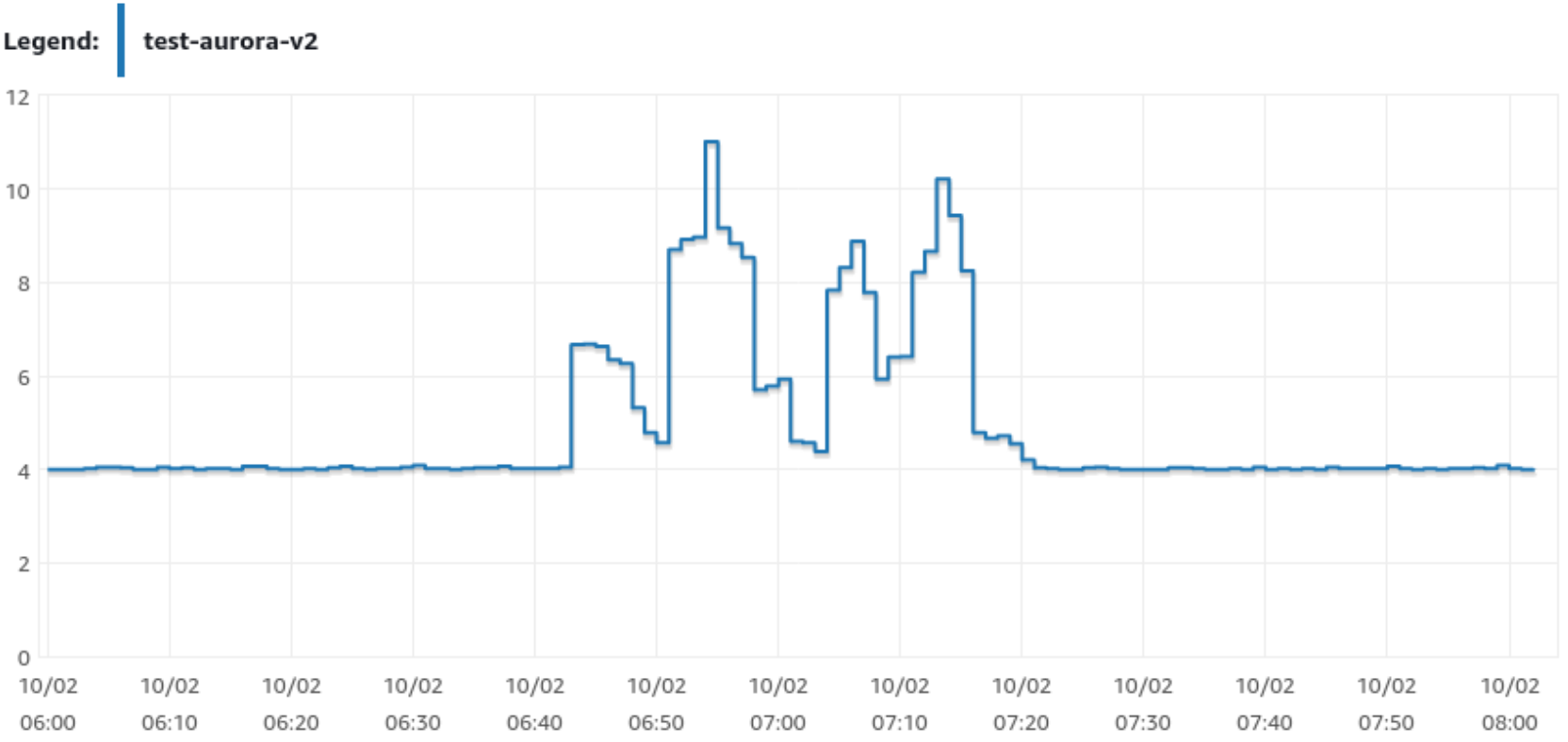


NYC results



Metrics

Metric	Statistic	Time Range	Period
Serverless ...	Average	Last 24 Ho...	1 Minute



Buffer size procedure

```
DELIMITER $$
CREATE PROCEDURE innodb_buffer_pool_size()
BEGIN
  DECLARE i INT DEFAULT 0;
  WHILE i < 100 DO
    SELECT time(now()), @@innodb_buffer_pool_size, SLEEP(1);
    SET i = i + 1;
  END WHILE;
END$$
DELIMITER ;

CALL innodb_buffer_pool_size();
```

InnoDB buffer pool

```
09:40:23        8808038400      0
09:40:24        8808038400      0
09:40:25        8808038400      0
09:40:26        8808038400      0
09:40:27        8808038400      0
09:40:28        8808038400      0
09:40:29        8808038400      0
09:40:30        8808038400      0
09:40:31        8808038400      0
09:40:32        8808038400      0
09:40:33        8808038400      0
09:40:34        7549747200      0
09:40:35        7549747200      0
09:40:36        7549747200      0
```

memory space

Photo by Andres Siimon on Unsplash

“Auto Scaling can convert some performance bugs driven downtime to large bills. This does not always apply to databases where bad query is often so bad no auto scaling can make it tolerable.”

Peter Zaitsev

Co-founder and CEO of Percona

aws

Resources

Learn more

RE:INVENT SESSIONS

DAT302
Instant and fine-grained scaling with Amazon Aurora Serverless v2
Tuesday, November 30, 2:00 PM – 3:00 PM

Learn more

ARTICLES AND VIDEOS ABOUT RELATIONAL DATABASES AND ELASTICITY

- Auto scaling changes performance bugs from an outage into a cost problem, Evan Jones, Datadog
- Aurora Serverless v2: The Good, the Better, and the Possibly Amazing, Jeremy Daly, Serverless
- A first look at Aurora Serverless v2 (video), Renato Losio, Funambol
- How Aurora Serverless v2 (preview) works, AWS
- Managing auto-scaling of RDS on AWS using the AWS CLI and Bash, Renato Losio, Funambol

Thank you!

Renato Losio

cloudiamo.com

Please complete the session survey