

Serverless Laravel

PHP

Hi!

[@matthieunapoli](#)

Open-source sponsors



I am not a *sysadmin*!

Serverless

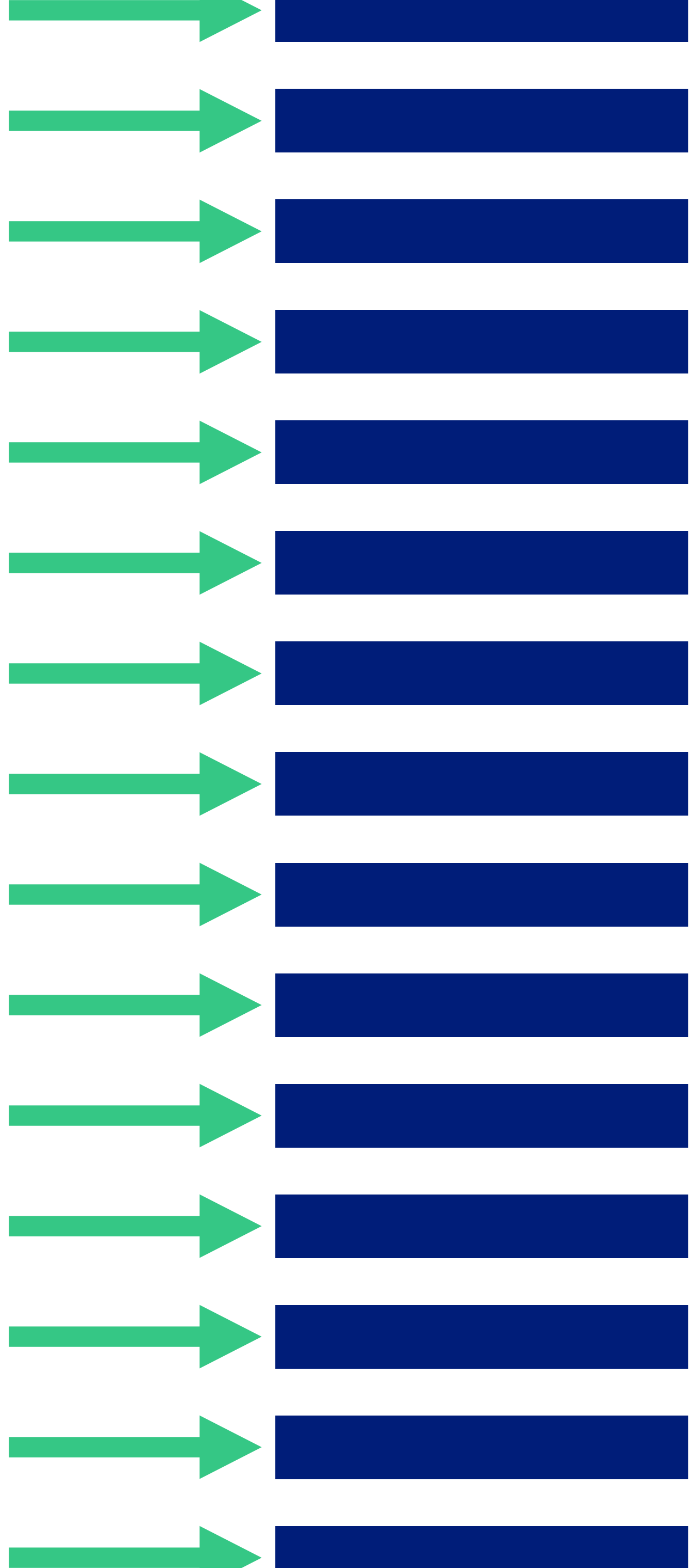
Do less server stuff



Code



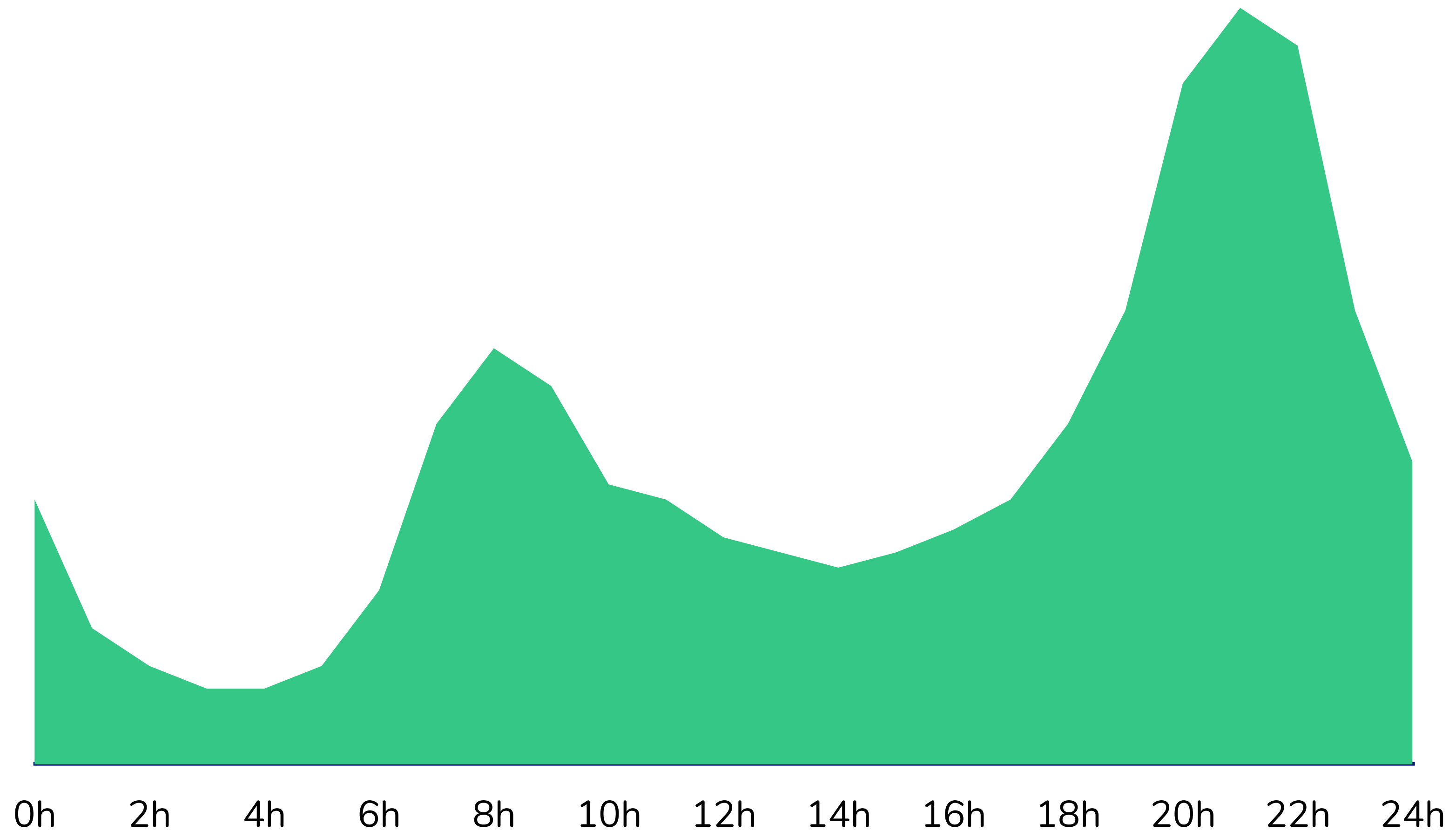


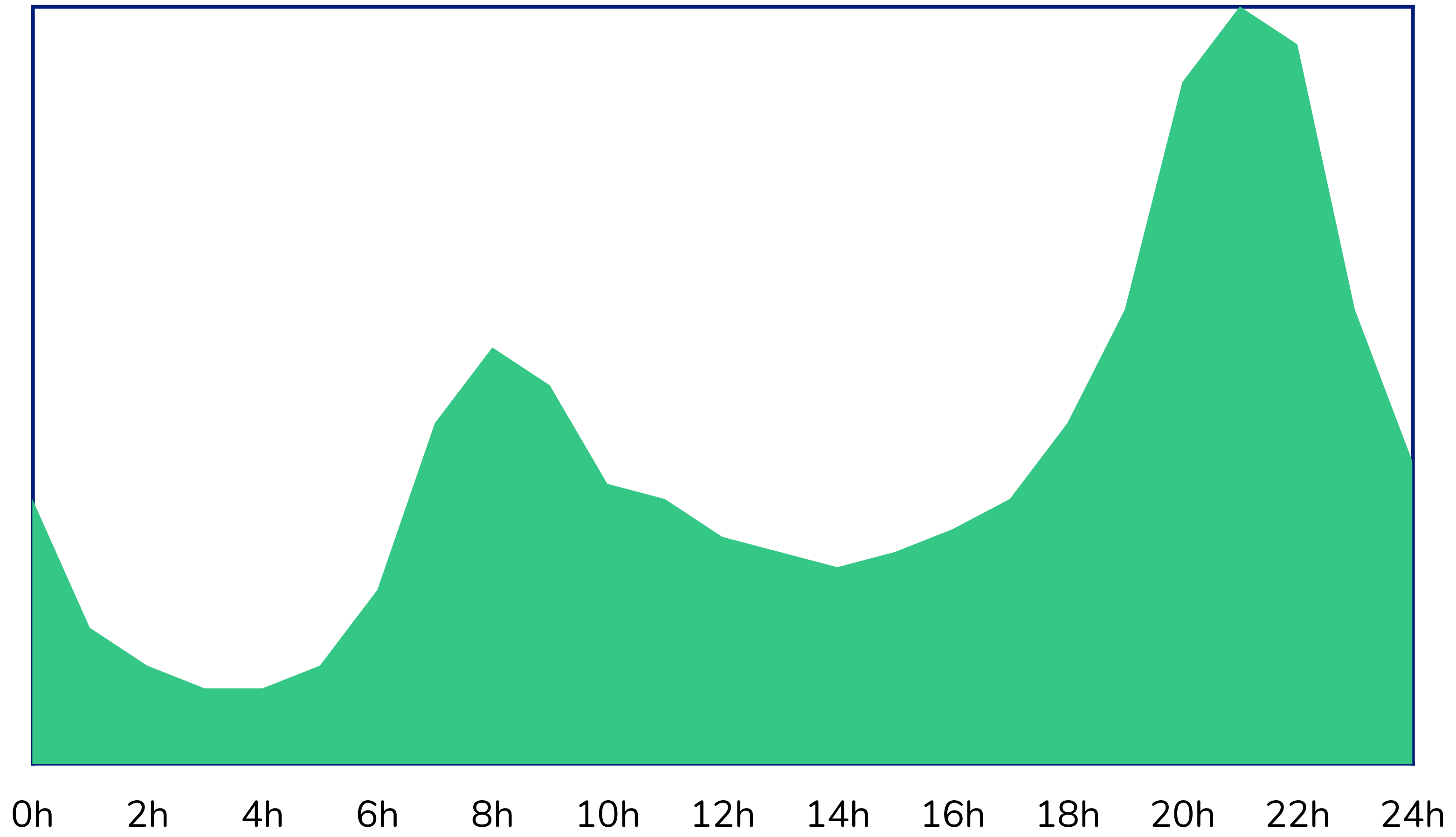


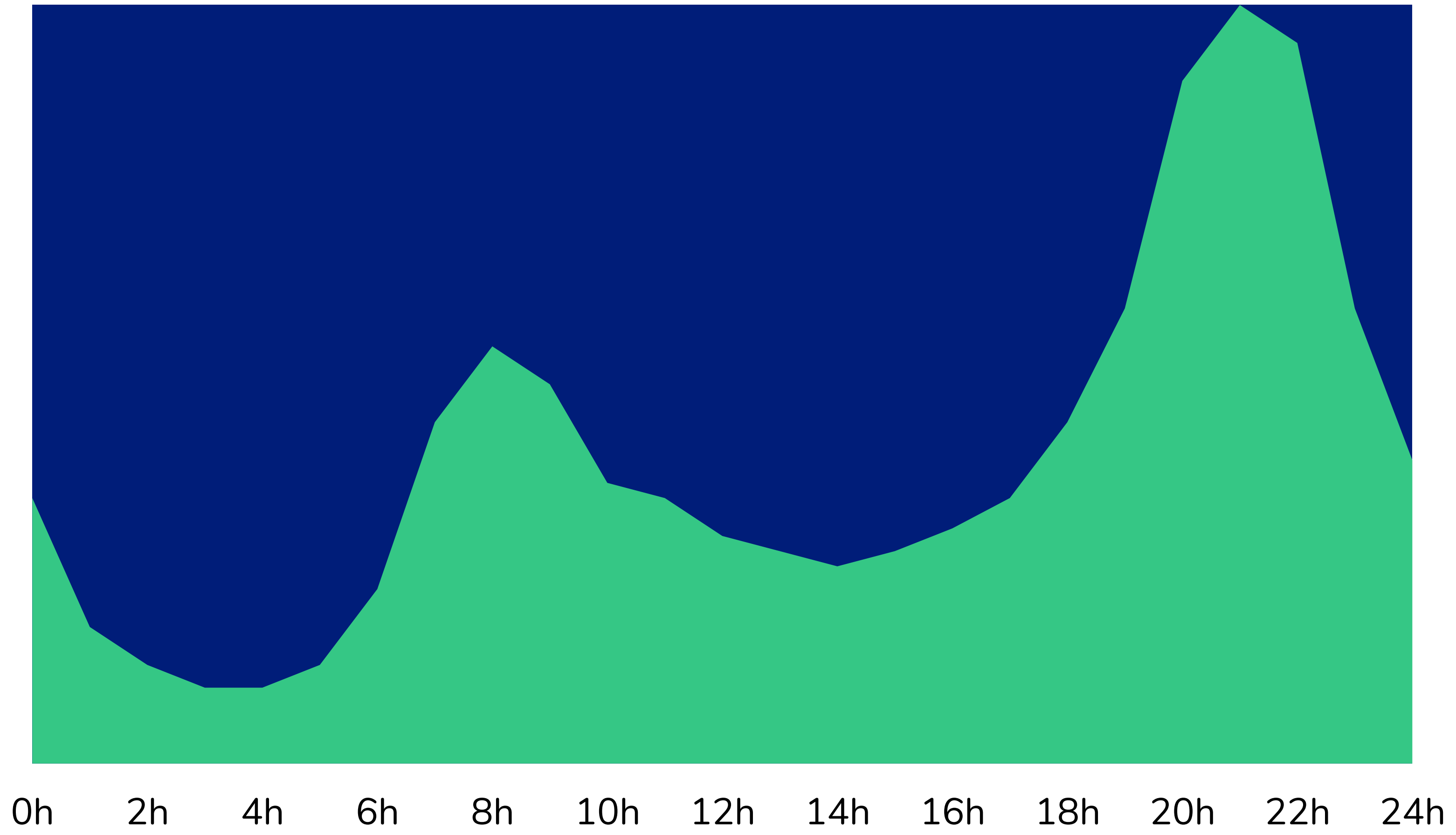


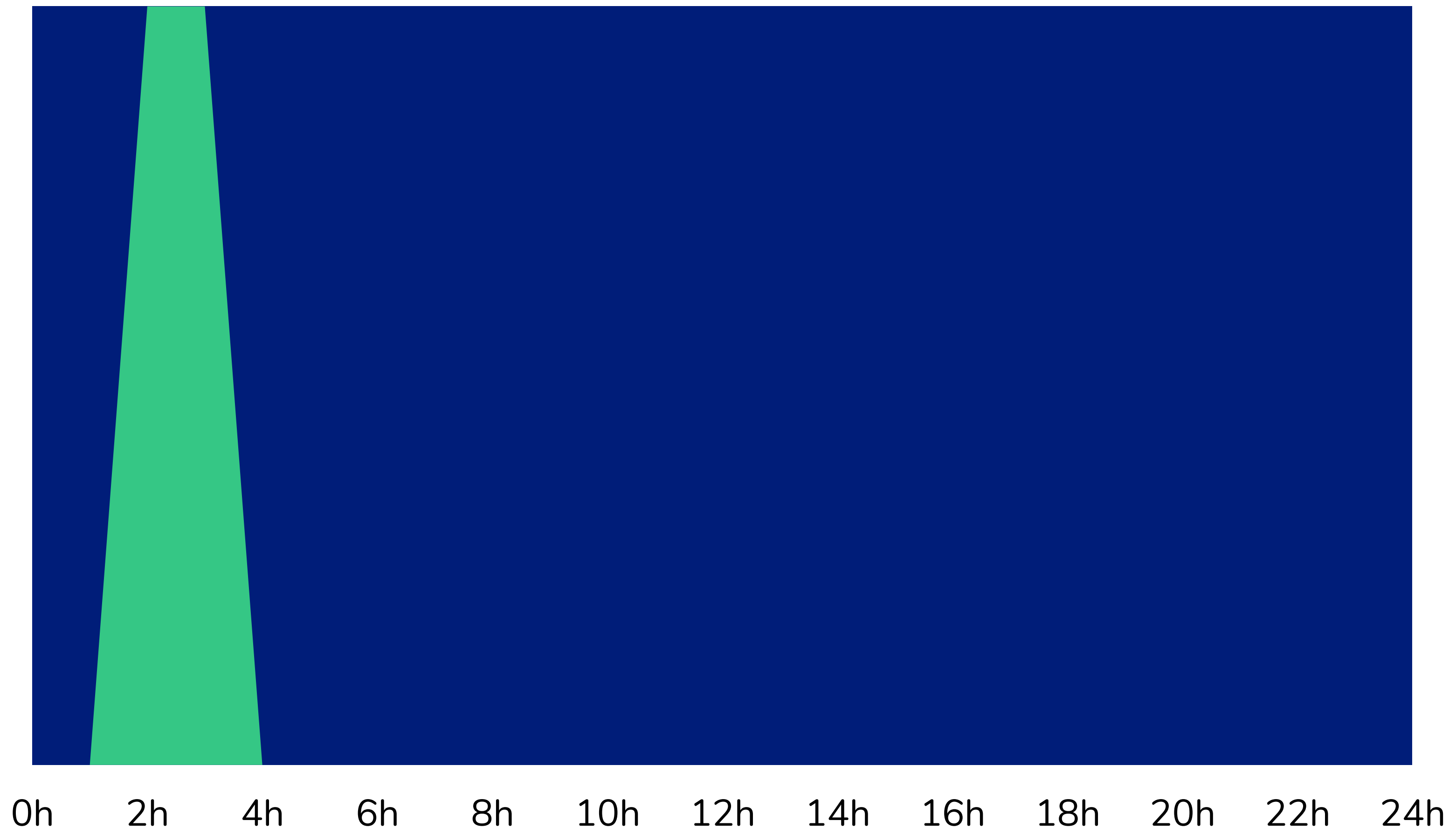
300ms

$$\$0.0000002 + 300 \times \$0.00000000208$$









Manage less

Scale more

Pay per use

- AWS Lambda
- Microsoft Azure Functions
- Google Cloud Functions
- CloudFlare Workers
- IBM OpenWhisk
- Alibaba Cloud Function
- ...

PHP



Laravel

brief

running PHP made simple

Everything you need to easily
deploy and run serverless PHP applications.



PHP functions

PHP-FPM

Choices

Documentation

Tools

Serverless framework

serverless.com

serverless.yml

```
service: my-app

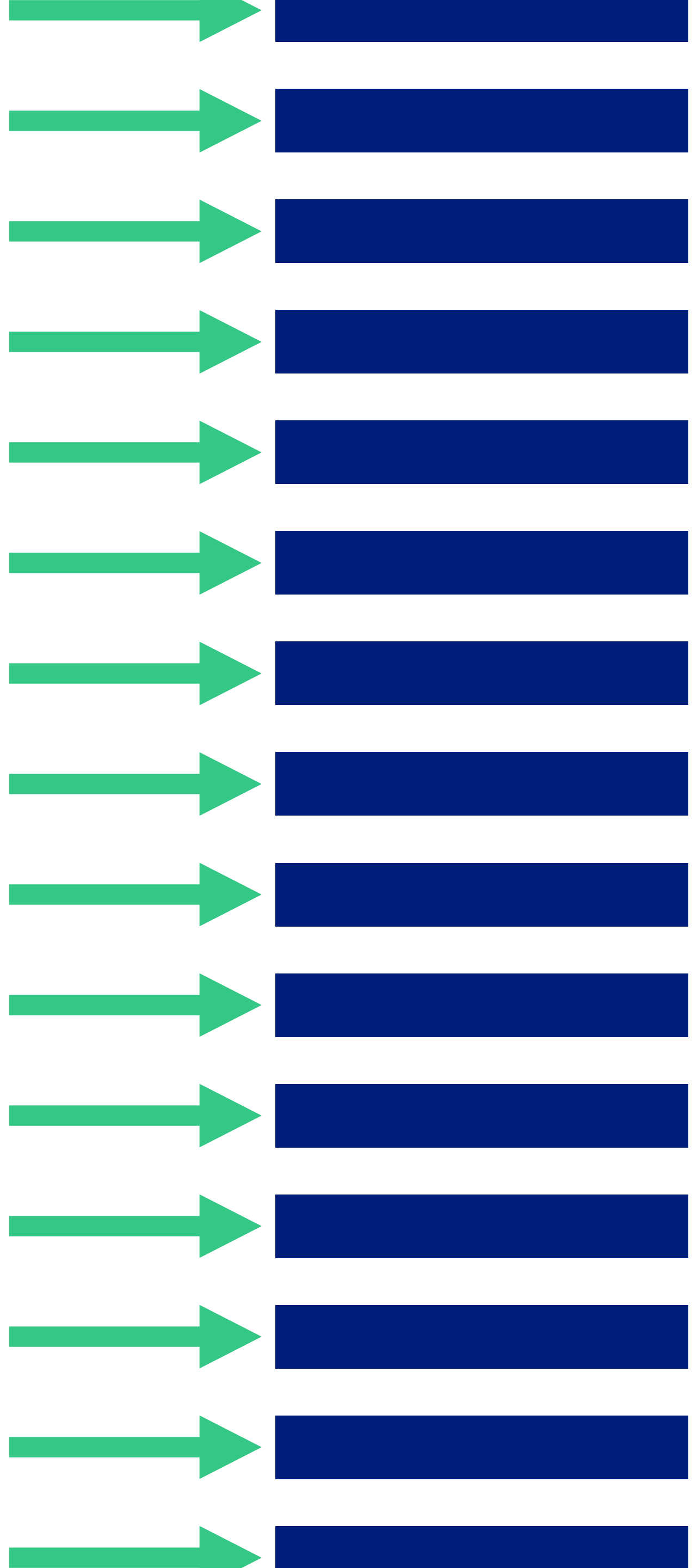
provider:
  name: aws

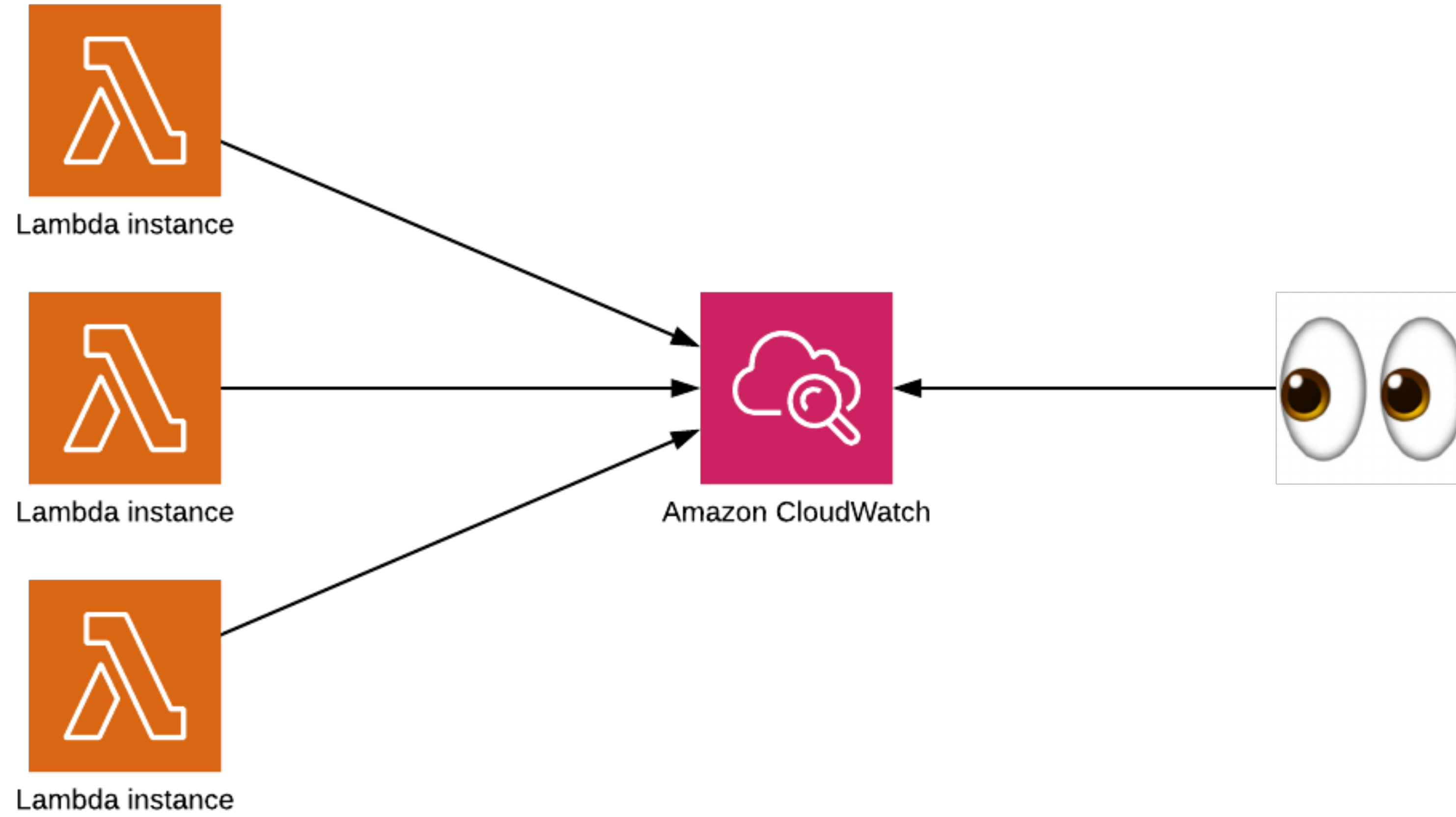
functions:
  web:
    handler: public/index.php
    runtime: provided
    layers:
      - ${bref:layer.php-74-fpm}
    events:
      - httpApi: '*'

plugins:
  - ./vendor/bref/bref
```

Adapting Laravel to AWS Lambda

- Logs





.env

```
LOG_CHANNEL=stderr
```

- Logs

- Sessions

.env

```
LOG_CHANNEL=stderr
```

```
SESSION_DRIVER=cookies
```

```
# or
```

```
SESSION_DRIVER=database
```

```
# or
```

```
SESSION_DRIVER=redis
```

```
# or
```

```
SESSION_DRIVER=array
```


- Logs
- Sessions
- The codebase is read-only

.env

```
LOG_CHANNEL=stderr
```

```
SESSION_DRIVER=cookies
```

```
VIEW_COMPILED_PATH=/tmp/compiled-views
```

```
$ laravel new my-project
```

```
[...]
```

```
$ composer require bref/laravel-bridge
```

```
[...]
```

```
$ php artisan vendor:publish --tag=serverless-config
```

```
Created file serverless.yml
```

```
$ serverless deploy
```

```
...
```

```
Service deployed
```

```
https://wyn0qrz.execute-api.us-east-1.amazonaws.com/
```

Demo!

File storage

```
/var/task # read-only
```

```
/tmp # read/write
```

```
/ # read-only
```

```
/var/task # read-only
```

```
/tmp # read/write
```

```
/var/t
```

```
/ # read-only
```

```
/tmp # read/write
```

```
/var/t
```

```
/ # read-only
```

```
/tmp # read/write
```

```
/ # read-only
```


AWS EFS

```
/var/task # read-only
```

```
/tmp # read/write
```

```
/var/t
```

```
/ # read-only
```

```
/tmp # read/write
```

```
/var/t
```

```
/ # read-only
```

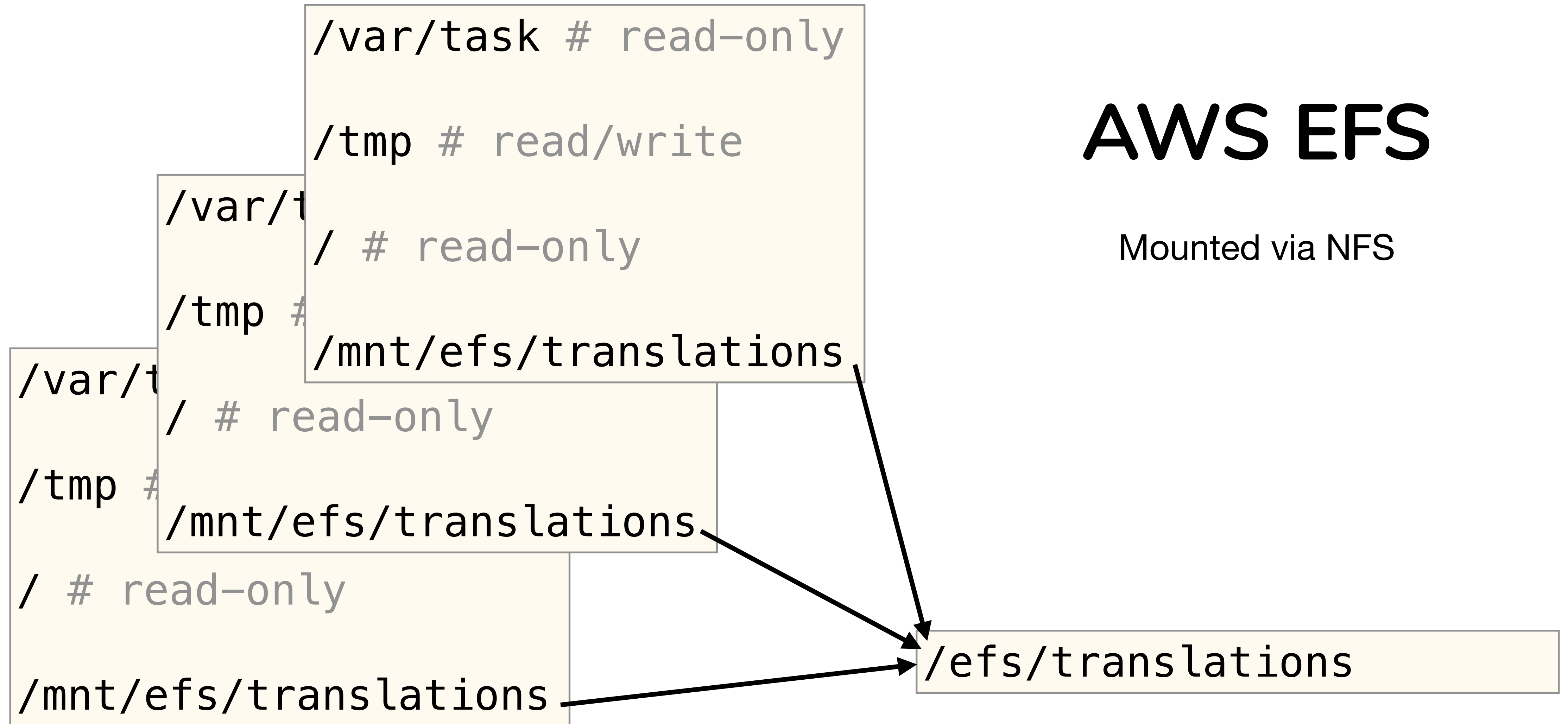
```
/tmp # read/write
```

```
/ # read-only
```

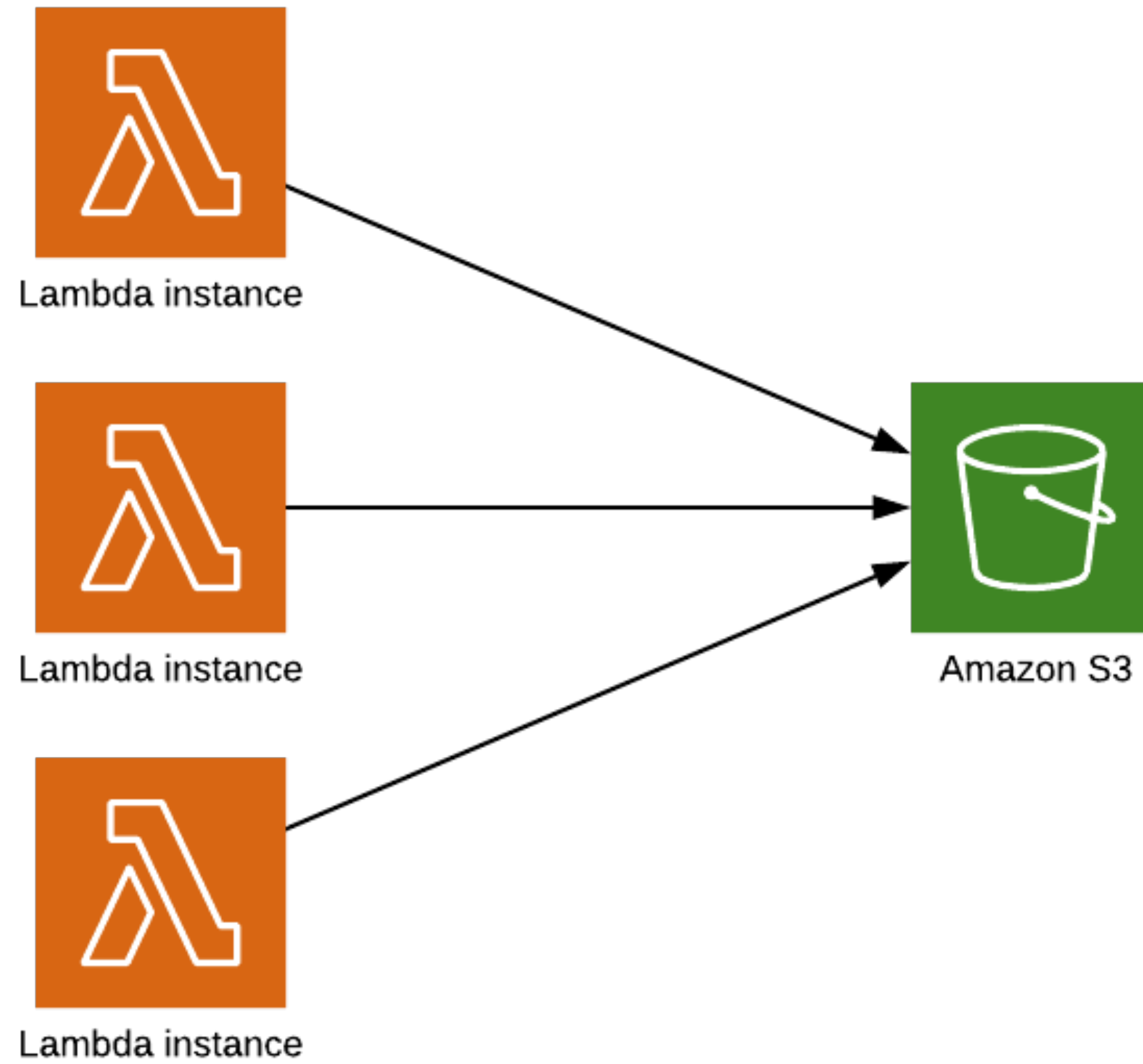
```
/efs/translations
```

AWS EFS

Mounted via NFS



Amazon S3



Laravel filesystems

```
'default' => env('FILESYSTEM_DRIVER', 'local'),

'disks' => [

    // ...

    's3' => [
        'driver' => 's3',
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'region' => env('AWS_DEFAULT_REGION'),
        'bucket' => env('AWS_BUCKET'),
        'url' => env('AWS_URL'),
        'endpoint' => env('AWS_ENDPOINT'),
    ],

],
```

Laravel filesystems

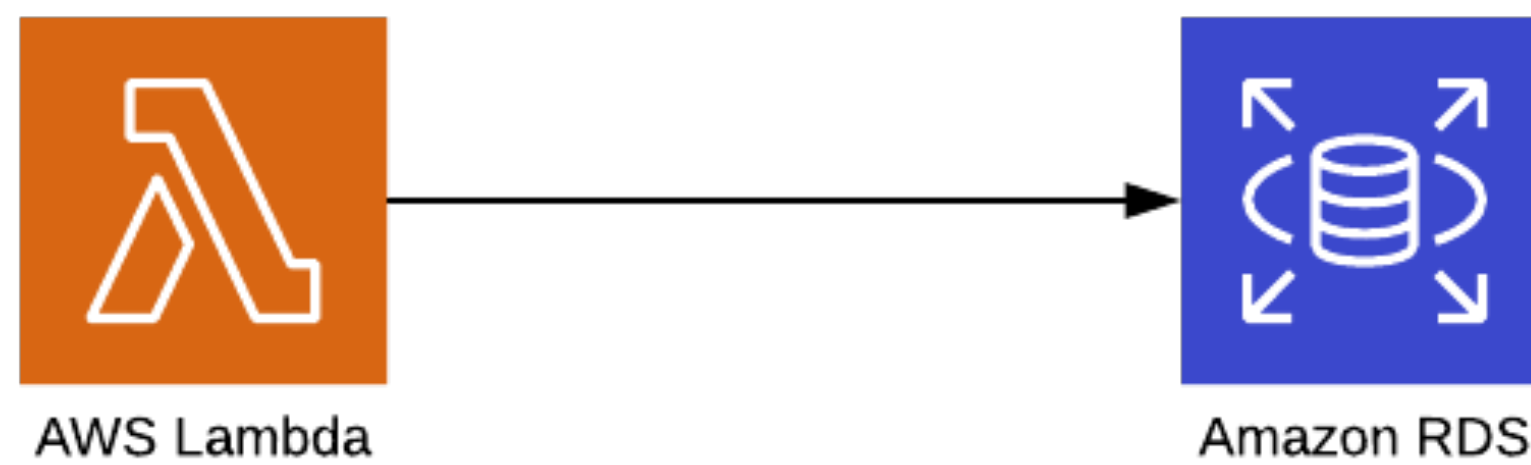
```
FILESYSTEM_DRIVER=s3
```

```
AWS_BUCKET=my-bucket-name
```

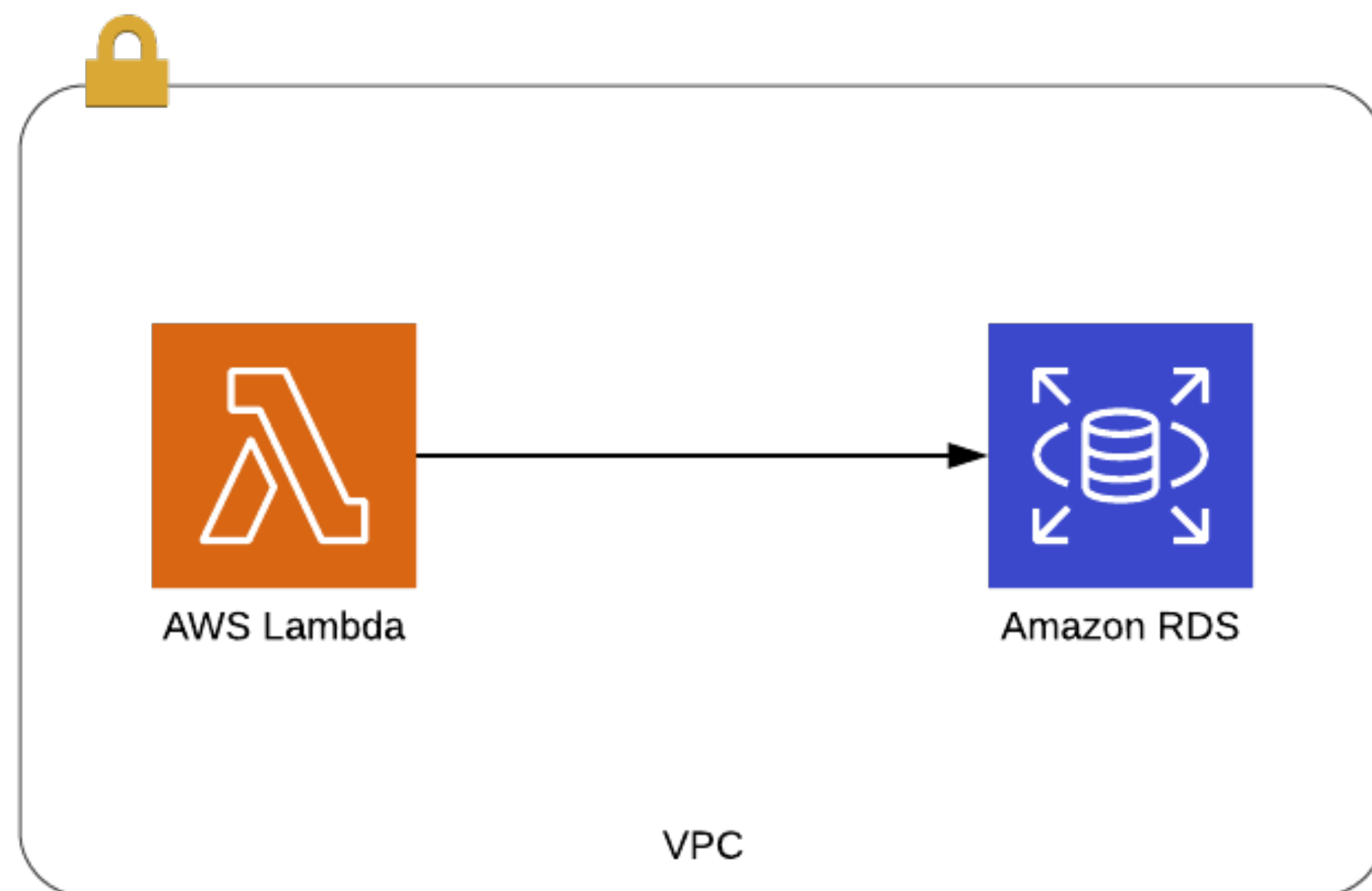
bref.sh/docs/frameworks/laravel.html#file-storage-on-s3

Databases

AWS RDS



AWS RDS



bref.sh/docs/environment/database.html

Laravel Queues

AWS SQS

Simulator


Demo

Use cases

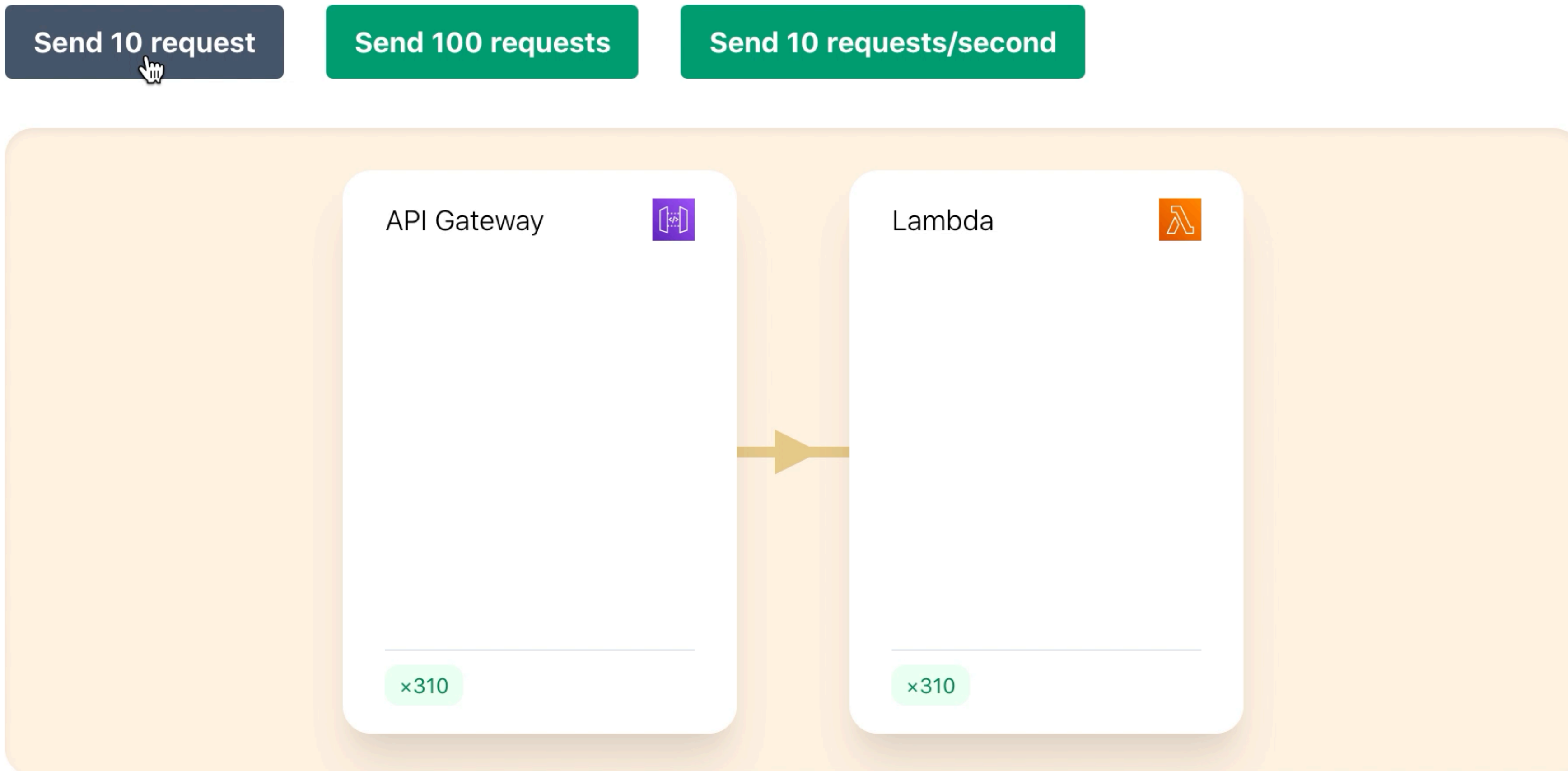
- Small websites/APIs -> cheap, effortless hosting
- Large websites/APIs -> scalable, lower TCO
- Back-offices -> better performances for lower costs
- Many websites (agencies, multi-tenancy) -> costs, effort
- Queue workers with SQS -> scalable, cost-efficient
- Micro-services messaging -> decoupled, scalable architecture
- Micro-functions: webhook, image resizing, PDF generation...

brief.sh

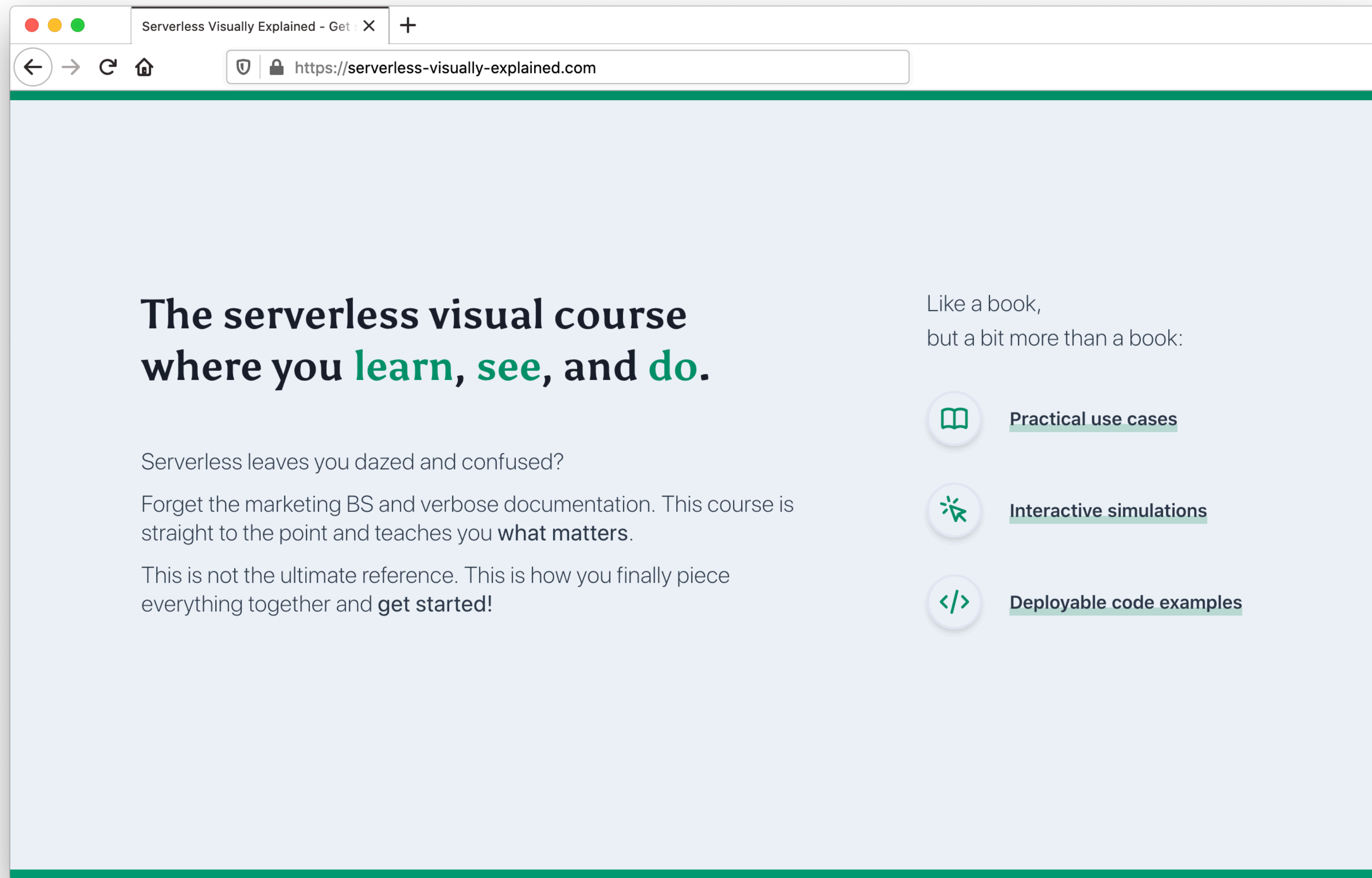
@matthieunapoli

 *New Lambda instances will be booted instantly when the running instances are already busy handling requests.*

Let's illustrate that: try sending 10 requests, then 100, then 10 requests per second.



As we can see, no matter the HTTP traffic, our application will scale instantly to handle it.



serverless-visually-explained.com