

A) INFORMACIÓN DE NROS DE CPU

Se agregó en la ruta /info, con el método cpus del módulo OS, el número de procesadores presentes en la máquina que despliega la app. En mi caso son 4.

```
const {cpus} =require('node:os')
const { Router } = express;
const router = new Router();

const GV = process; // G de Global - V de variable (a diferencia de EV en config que son

router.get("/", (_req, res) => {
  let data = [];
  createMocks(5, data);
  res.render("index", { mocks: data });
});

router.get("/info", (_req, res) => {
  res.json({
    ArgumetosEntrada: GV.argv,
    NombrePlataforma: GV.platform,
    VersionNodeJS: GV.version,
    MemoriaTotalReservada: GV.memoryUsage().rss,
    PathEjecucion: GV.execPath,
    ProcessID: GV.pid,
    CarpetaProyecto: GV.cwd(),
    NumeroDeCPUS: cpus().length,
  });
});
```

B) NUMEROS DE PROCESOS NODEMON

PROBLEMAS

SAIDA

TERMINAL

SQL CONSOLE

CONSOLE DE DEPURACIÓN

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor

=====

node.exe 11488 Console 3 3.536 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor

=====

node.exe 11488 Console 3 3.552 KB

node.exe 2536 Console 3 28.740 KB

node.exe 13224 Console 3 70.636 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ nodemon ./src/app.js -p 8081 -m FORK

[nodemon] 2.0.15

[nodemon] to restart at any time, enter 'rs'

[nodemon] watching path(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting 'node ./src/app.js -p 8081 -m FORK'

Server RUN ON PORT: 8081

MODO FORK

(1)(CONSOLA IZQ) Se chequea cuantos procesos de node se están corriendo antes de iniciar con el comando:

tasklist //fi "imagenname eq node.exe"

Esto devuelve 1 proceso

(2)(CONSOLA DER) Se inicia el servidor en modo FORK con el comando:

Nodemon ./src/app.js -p 8081 -m FORK

(3)(CONSOLA IZQ) Repite acción (1):

Esto devuelve 3 procesos

NOTA:Para iniciar el control como esta en gitbash, se usa la foobar //

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor

=====

node.exe 11488 Console 3 3.552 KB

node.exe 2536 Console 3 28.740 KB

node.exe 13224 Console 3 70.636 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor

=====

node.exe 11488 Console 3 3.820 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor

=====

node.exe 11488 Console 3 3.828 KB

node.exe 15808 Console 3 28.244 KB

node.exe 16100 Console 3 71.280 KB

node.exe 15508 Console 3 71.284 KB

node.exe 14780 Console 3 71.892 KB

node.exe 3784 Console 3 71.384 KB

node.exe 15376 Console 3 71.496 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$

Server RUN ON PORT: 8081

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ nodemon ./src/app.js -p 8081 -m CLUSTER

[nodemon] 2.0.15

[nodemon] to restart at any time, enter `rs`

[nodemon] watching path(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting 'node ./src/app.js -p 8081 -m CLUSTER'

Server iniciado en modo:[CLUSTER] con id de Proceso [14780] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [3784] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [15376] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [15508] en puesto [8081]

MODO CLUSTER

(1) (CONSOLA DER) Se cierra la consola que lleva iniciado el modo FORK con el comando:

c^ o ctrl+c

(2) (CONSOLA IZQ) Se chequea cuantos procesos de node se están corriendo antes de iniciar con el comando:

tasklist //fi "imagenname eq node.exe"

Esto devuelve 1 proceso

(3) (CONSOLA DER) Se inicia el servidor en modo CLUSTER con el comando:

Nodemon ./src/app.js -p 8081 -m CLUSTER

(4) CONSOLA IZQ) Repite acción (1):

Esto devuelve 7 procesos

NOTA:

- el de FORK, ya se encuentra puesto por default con yargs
- el IF que divide a las dos MODOS, posibles: tiene adrede armado un ELSE IF para hacer un código más gráfico y entendible

C) FOREVER

MODO FORK

Comando para iniciar forever start -w ./src/app.js -p 8081 -m FORK

NOTA: se usa start -w, porque según la documentación se usa esa flag para que se active el modo escucha


```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever start -w ./src/app.js -p 8081 -m FORK
warn: --minUptime not set. Defaulting to: 1000ms
warn: --spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
info: Forever processing file: ./src/app.js
(node:16192) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:16192) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:12828) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:12828) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info: Forever processes running
data: uid command script forever pid id logfile uptime
data: [0] sJdW "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js -p 8081 -m FORK 4748 10204 C:\Users\Ari\.forever\sJdW.log 0:0:0:4.144
```

El conteo de procesos da 3

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:12828) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:12828) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info: Forever processes running
data: uid command script forever pid id logfile uptime
data: [0] sJdW "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js -p 8081 -m FORK 4748 10204 C:\Users\Ari\.forever\sJdW.log 0:0:0:4.144

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesi3n N3m. de ses Uso de memor
=====
node.exe 11488 Console 3 3.692 KB
node.exe 4748 Console 3 140.248 KB
node.exe 10204 Console 3 55.896 KB
```

MODO CLUSTER

Se inicia el modo cluster con el comando: forever start -w ./src/app.js -p 8081 -m CLUSTER
Se hace solicita un comando: “forever list” para ver cuantas instancias del mismo server hay
Luego se consulta cuantos procesos hay con el comando : tasklist //fi "imagenname eq node.exe"
Se devuelven 7 procesos

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever start -w ./src/app.js -p 8081 -m CLUSTER
warn: --minUptime not set. Defaulting to: 1000ms
warn: --spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
info: Forever processing file: ./src/app.js
(node:796) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:796) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:15568) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:15568) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info: Forever processes running
data: uid command script forever pid id logfile uptime
data: [0] L1ZU "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js -p 8081 -m CLUSTER 492 8280 C:\Users\Ari\.forever\L1ZU.log 0:0:0:11.863

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesi3n N3m. de ses Uso de memor
=====
node.exe 11488 Console 3 3.788 KB
node.exe 492 Console 3 183.864 KB
node.exe 8280 Console 3 56.168 KB
node.exe 14684 Console 3 75.776 KB
node.exe 13016 Console 3 71.456 KB
node.exe 6504 Console 3 71.608 KB
node.exe 15476 Console 3 71.768 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$
```

D) PM2

Con PM2 no es necesario ingresar un argumento por -m para entrar en modo cluster. Porque PM2, ya tiene esa opci3n entre sus funcionalidades,
MODO FORK: pm2 start ./src/app.js --name="Server1" --namespace="Fork" --watch -- 8081

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ pm2 start ./src/app.js --name="Server1" --namespace="Fork" --watch -- 8081
[PM2] Spawning PM2 daemon with pm2_home=C:\Users\Ari\.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js in fork_mode (1 instance)
[PM2] Done.

id name namespace version mode pid uptime 3 status cpu mem user watching
0 Server1 Fork 1.0.0 fork 14000 2s 0 online 56.3% 35.8mb Ari enabled

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ pm2 describe 0
Describing process with id 0 - name Server1

status online
name Server1
namespace Fork
version 1.0.0
restarts 0
uptime 50s
script path C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js
script args 8081
error log path C:\Users\Ari\.pm2\logs\Server1-error.log
out log path C:\Users\Ari\.pm2\logs\Server1-out.log
pid path C:\Users\Ari\.pm2\pids\Server1-0.pid
interpreter node
interpreter args N/A
script id 0
exec cwd C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30
exec mode fork_mode
node.js version 16.13.1
node env N/A
watch & reload 3
unstable restarts 0
created at 2022-08-08T04:59:27.153Z
```

MODO CLUSTER: pm2 start ./src/app.js --name="Server2"--namespace="Cluster" --watch -i -max -- 8081

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ pm2 start ./src/app.js --name="Server1" --namespace="Fork" --watch -i max -- 8081
[PM2] Starting C:\Users\Ari\Documents\repobackend\curso-ch-backend\desafioClase30\src\app.js in cluster_mode (0 instance)
[PM2] Done.

id	name	namespace	version	mode	pid	uptime	U	status	cpu	mem	user	watching
0	Server1	Fork	1.0.0	cluster	15664	7s	0	online	82.8%	64.7mb	Ari	enabled
1	Server1	Fork	1.0.0	cluster	804	7s	0	online	81.3%	64.5mb	Ari	enabled
2	Server1	Fork	1.0.0	cluster	8404	7s	0	online	89%	59.9mb	Ari	enabled
3	Server1	Fork	1.0.0	cluster	6276	6s	0	online	84.4%	46.9mb	Ari	enabled

E) Eliminar procesos:

NODEMON

PROBLEMAS SALIDA **TERMINAL** SQL CONSOLE CONSOLA DE DEPURACIÓN

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor
===== =====
node.exe 11488 Console 3 3.860 KB
node.exe 3172 Console 3 32.324 KB
node.exe 60 Console 3 71.092 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ TASKKILL //PID:3172 //F
Correcto: se terminó el proceso con PID 3172.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor
===== =====
node.exe 11488 Console 3 3.772 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ █

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ nodemon ./src/app.js -p 8081 -m FORK
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./src/app.js -p 8081 -m FORK`
Server RUN ON PORT: 8081

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ █

MODO FORK. Se inicia (en consola de Derecha) la app.js en modo FORK. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 3 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar).Se observa en consola derecha que se vuelve a habilitar para ingresar comandos y al consultar con tasklist (en consola izq) se encuentra con un solo proceso.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor
===== =====
node.exe 11488 Console 3 3.748 KB
node.exe 10244 Console 3 28.068 KB
node.exe 4548 Console 3 71.376 KB
node.exe 12668 Console 3 71.808 KB
node.exe 15148 Console 3 71.464 KB
node.exe 6020 Console 3 71.292 KB
node.exe 8500 Console 3 71.512 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ TASKKILL //PID:10244 //F
Correcto: se terminó el proceso con PID 10244.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen PID Nombre de sesión Núm. de ses Uso de memor
===== =====
node.exe 11488 Console 3 3.752 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ █

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ nodemon ./src/app.js -p 8081 -m CLUSTER
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./src/app.js -p 8081 -m CLUSTER`
Server iniciado en modo:[CLUSTER] con id de Proceso [6020] en puesto [8081]
Server iniciado en modo:[CLUSTER] con id de Proceso [15148] en puesto [8081]
Server iniciado en modo:[CLUSTER] con id de Proceso [12668] en puesto [8081]
Server iniciado en modo:[CLUSTER] con id de Proceso [8500] en puesto [8081]

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ █

Por otro lado, en caso de no matar al primario arroja un CRASHED:

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ nodemon ./src/app.js -p 8081
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./src/app.js -p 8081`
Server RUN ON PORT: 8081
[nodemon] app crashed - waiting for file changes before starting...

Luego si se mata al primario, habilita nuevamente los comandos:

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ nodemon ./src/app.js -p 8081
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./src/app.js -p 8081`
Server RUN ON PORT: 8081
[nodemon] app crashed - waiting for file changes before starting...

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
\$ █

NOTA: Diferencia con forever que en estos casos al no matar al Primario reinicia al worker.

MODO CLUSTER.
ELIMINAR EL CLUSTER.ISPRIMARY

Se inicia (en consola de Derecha) la app.js en modo CLUSTER. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 7 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar).Se observa en consola derecha que se vuelve a habilitar para ingresar comandos y al consultar con tasklist (en consola izq) se encuentra con un solo proceso.

PROBLEMAS

SAUDA

TERMINAL

CONSOLE

CONSOLA DE DEPURACIÓN

node.exe12668Console371.808 KB

node.exe15148Console371.464 KB

node.exe6020Console371.292 KB

node.exe8500Console371.512 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ TASKKILL //PID:10244 //F

Correcto: se terminó el proceso con PID 10244.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen	PID	Nombre de sesión	Núm. de ses	Uso de memor
node.exe	11488	Console	3	3.752 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen	PID	Nombre de sesión	Núm. de ses	Uso de memor
node.exe	11488	Console	3	3.752 KB
node.exe	1488	Console	3	28.888 KB
node.exe	916	Console	3	71.420 KB
node.exe	6064	Console	3	71.232 KB
node.exe	8676	Console	3	71.716 KB
node.exe	11028	Console	3	71.356 KB
node.exe	12752	Console	3	71.336 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ TASKKILL //PID:12752 //F

Correcto: se terminó el proceso con PID 12752.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen	PID	Nombre de sesión	Núm. de ses	Uso de memor
node.exe	11488	Console	3	3.760 KB
node.exe	1488	Console	3	27.606 KB
node.exe	916	Console	3	56.320 KB
node.exe	6064	Console	3	55.940 KB
node.exe	8676	Console	3	71.936 KB
node.exe	11028	Console	3	56.264 KB
node.exe	1032	Console	3	71.144 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ nodemon ./src/app.js -p 8081 -m CLUSTER

[nodemon] 2.0.15

[nodemon] to restart at any time, enter `rs`

[nodemon] watching path(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node ./src/app.js -p 8081 -m CLUSTER`

Server iniciado en modo:[CLUSTER] con id de Proceso [6020] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [15148] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [12668] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [8500] en puesto [8081]

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)

\$ nodemon ./src/app.js -p 8081 -m CLUSTER

[nodemon] 2.0.15

[nodemon] to restart at any time, enter `rs`

[nodemon] watching path(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node ./src/app.js -p 8081 -m CLUSTER`

Server iniciado en modo:[CLUSTER] con id de Proceso [11028] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [8676] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [6064] en puesto [8081]

Server iniciado en modo:[CLUSTER] con id de Proceso [12752] en puesto [8081]

worker con id de proceso 12752 eliminado

Server iniciado en modo:[CLUSTER] con id de Proceso [1032] en puesto [8081]

ELIMINAR EL CLUSTER.ISWORKER

Se inicia (en consola de Derecha) la app.js en modo CLUSTER. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 7 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar). Se observa en consola derecha que se despide por consola un mensaje que indica que el proceso con id 12752 fue eliminado, y de inmediato se genera un nuevo worker con pid 1032, y al consultar con tasklist (en consola izq) se encuentra con 7 procesos nuevamente.

NOTA: QUE SURJA UN NUEVO CLUSTER.FORK DEL PRIMARIO SE DEBE A LA SIGUIENTE PIEZA DE CÓDIGO:

```
if(MODO == "FORK"){
  server.listen(PORT, (err) => {
    !err
      ? console.log("Server RUN ON PORT: ", PORT)
      : console.log("Server NOT RUN");
  });
} else {
  if(cluster.isMaster){
    for (let i = 0; i < numCPUS; i++) {
      cluster.fork()
    }
    cluster.on('exit', (worker,code,signal)=>{
      console.log(`worker con id de proceso ${worker.process.pid} eliminado`)
      //para mantener una cantidad de cluster activos acá se puede poner un
      cluster.fork()
      // sino al eliminar el último cluster las request al sitio por parte del cliente no tendran un cpu que lo procese
    })
  } else if( MODO == "CLUSTER") {
    server.listen(PORT, (err) => {
      !err
        ? console.log(`Server iniciado en modo:[${MODO}] con id de Proceso [${process.pid}] en puesto [${PORT}]`)
        : console.log("Server NOT RUN");
    });
  }
}
```

FOREVER: MODO FORK

- Se inicia con un tasklist //fi "imagenname eq node.exe", para ver cuantos procesos de node hay abiertos. Figura uno
- Se ejecuta y da inicio a una instancia del servidor en modo FORK.
- Se hace otro tasklist se ven 3 procesos
- Se hace un forever list, para visualizar el servidor iniciado.
- Se hace un forever stop por uid. Se verifica que el proceso se cierra con otro


```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  11488 Console                3      3.820 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever start -w ./src/app.js -p 8081 -m FORK
warn:  --minUptime not set. Defaulting to: 1000ms
warn:  --spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
info:   Forever processing file: ./src/app.js
(node:7796) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:7796) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  11488 Console                3      3.820 KB
node.exe                  8232 Console                3     87.660 KB
node.exe                  13720 Console               3     70.724 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:7068) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:7068) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   Forever processes running
data:    uid  command          script                                forever pid  id logfile                                uptime
data: [0] Gr_j "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\d
(node:7480) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:7480) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   Forever stopped process:
uid  command          script                                forever pid  id logfile                                uptime
[0] Gr_j "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\d
(node:9480) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:9480) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   Forever processes running
uid  command          script                                forever pid  id logfile                                uptime
[0] Gr_j "C:\Program Files\nodejs\node.exe" C:\Users\Ari\Documents\repobackend\curso-ch-backend\d
(node:7272) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:7272) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   No forever processes running

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  11488 Console                3      3.808 KB
```

Funciona del mismo modo si se usa un taskkill (siempre que se apunte al primario, en caso de apuntar al worker FOREVER reinicia la instancia y se mantiene el proceso primario del server)

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  11488 Console                3      5.244 KB
node.exe                  15672 Console               3     192.324 KB
node.exe                  6468 Console                3      55.844 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ TASKKILL //PID:15672 //F
Correcto: se terminó el proceso con PID 15672.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:15904) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:15904) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   No forever processes running

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$
```

MODO CLUSTER
TASKKILL isPRIMARY

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever start -w ./src/app.js -p 8081 -m CLUSTER
warn:  --minUptime not set. Defaulting to: 1000ms
warn:  --spinSleepTime not set. Your script will exit if it does not stay up for at least 1000ms
info:   Forever processing file: ./src/app.js
(node:8452) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:8452) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  12588 Console                3     127.728 KB
node.exe                  15476 Console                3      72.712 KB
node.exe                  15740 Console                3      71.640 KB
node.exe                  11252 Console                3      71.640 KB
node.exe                  3844 Console                3      71.796 KB
node.exe                  9604 Console                3      71.448 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ TASKKILL //PID:12588 //F
Correcto: se terminó el proceso con PID 12588.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ forever list
(node:16296) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:16296) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
info:   No forever processes running
```

Forever no reconoce que quede un server corriendo. Pero node, tiene procesos que quedan pendientes

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagenname eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  15476 Console                3     56.532 KB
node.exe                  15740 Console                3     71.852 KB
node.exe                  11252 Console                3     56.852 KB
node.exe                  3844 Console                3     56.820 KB
node.exe                  9604 Console                3     56.512 KB
```

Pero al matar a uno de ellos se cierran todos


```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ TASKKILL //PID:3844 //F
Correcto: se termino el proceso con PID 3844.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagename eq node.exe"
INFORMACIÓN: no hay tareas ejecutándose que coincidan con los
criterios especificados.
```

PM2:

NOTA: En PM2, los taskkill no sirven, PM2 entiende que es un cierre inesperado y lo reinicia.

MODO FORK

Cuando se mata al proceso principal, se reinicia uno nuevo.

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ TASKKILL //PID:14000 //F
Correcto: se termino el proceso con PID 14000.

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ tasklist //fi "imagename eq node.exe"

Nombre de imagen          PID Nombre de sesión Núm. de ses Uso de memor
=====
node.exe                  2024 Console          3    34.036 KB
node.exe                  6508 Console          3    59.672 KB

Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$
```

Para matar al proceso, se usa el comando pm2 delete id:

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: [ '0' ])
[PM2] [Server1](0) ✓
```

id	name	namespace	version	mode	pid	uptime	🔄	status	cpu	mem	user	watching
----	------	-----------	---------	------	-----	--------	---	--------	-----	-----	------	----------

MODO CLUSTER

Cuando se mata un proceso, aún siendo el primario pm2, lo reemplaza con otro.

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ TASKKILL //PID:15664 //F
Correcto: se termino el proceso con PID 15664.
```

Resultado de matar un proceso(se siguen manteniendo los 4):

Process List

[0] Server1	Mem: 58 MB	CPU: 1 %	online
[1] Server1	Mem: 58 MB	CPU: 3 %	online
[2] Server1	Mem: 59 MB	CPU: 0 %	online
[3] Server1	Mem: 58 MB	CPU: 0 %	online

Server1 Logs

Custom Metrics

Used Heap Size	26.40 MiB
Heap Usage	88.76 %
Heap Size	29.74 MiB

Metadata

App Name	Server1
Namespace	Fork
Version	1.0.0

left/right: switch boards | up/down/mouse: scroll | Ctrl-C: exit

Para eliminar se puede hacer de a uno con pm2 delete id

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: [ '0' ])
[PM2] [Server1](0) ✓
```

id	name	namespace	version	mode	pid	uptime	🔄	status	cpu	mem	user	watching
1	Server1	Fork	1.0.0	cluster	804	3m	0	online	0%	58.2mb	Ari	enabled
2	Server1	Fork	1.0.0	cluster	8404	3m	0	online	0%	58.8mb	Ari	enabled
3	Server1	Fork	1.0.0	cluster	6276	3m	0	online	0%	59.2mb	Ari	enabled

o pm2 delete

```
Ari@DESKTOP-PPRBKQD MINGW64 ~/Documents/repobackend/curso-ch-backend/desafioClase30 (main)
$ pm2 delete all
[PM2] Applying action deleteProcessId on app [all](ids: [ 1, 2, 3 ])
[PM2] [Server1](3) ✓
[PM2] [Server1](1) ✓
[PM2] [Server1](2) ✓
```

id	name	namespace	version	mode	pid	uptime	🔄	status	cpu	mem	user	watching
----	------	-----------	---------	------	-----	--------	---	--------	-----	-----	------	----------

F) Nginx:

- i. Crear dos server. Uno en modo fork apuntando a puerto 8080 y otro en modo cluster apuntando a puerto 8081. Configurar en Nginx que las consultas tengan el proxy_pass adecuado a la petición del cliente.

```
nginx.conf X
conf > nginx.conf
1 worker_processes 1;
2
3 events {
4     worker_connections 1024;
5 }
6
7 http {
8     include mime.types;
9     default_type application/octet-stream;
10
11     sendfile on;
12
13     keepalive_timeout 65;
14
15     upstream app_cluster {
16         server 127.0.0.1:8081;
17     }
18     upstream app_fork {
19         server 127.0.0.1:8080;
20     }
21     #gzip on;
22     server {
23         listen 80;
24         server_name localhost;
25
26         location /api/randoms/ {
27             proxy_pass http://app_cluster;
28         }
29
30         location / {
31             proxy_pass http://app_fork;
32         }
33
34         error_page 500 502 503 504 /50x.html;
35         location = /50x.html {
36             root html;
37         }
38     }
39 }
```

ii. Crear en modo cluster, 4 instancias de server, que escuchen en puertos 8081, 8082, 8083 y 8084 equitativamente (es decir que weight este por default en 1), las peticiones que haga el cliente en la ruta /api/randoms

```
nginx.conf X
conf > nginx.conf
1 worker_processes 1;
2
3 events {
4     worker_connections 1024;
5 }
6
7 http {
8     include mime.types;
9     default_type application/octet-stream;
10
11     sendfile on;
12
13     keepalive_timeout 65;
14
15     upstream app_cluster {
16         server 127.0.0.1:8081;
17         server 127.0.0.1:8082;
18         server 127.0.0.1:8083;
19         server 127.0.0.1:8084;
20     }
21     upstream app_fork {
22         server 127.0.0.1:8080;
23     }
24     #gzip on;
25     server {
26         listen 80;
27         server_name localhost;
28
29         location /api/randoms/ {
30             proxy_pass http://app_cluster;
31         }
32
33         location / {
34             proxy_pass http://app_fork;
35         }
36
37         error_page 500 502 503 504 /50x.html;
38         location = /50x.html {
39             root html;
40         }
41     }
42 }
```