
PM2

Realiza o gerenciamento de processos não só de aplicações que rodam em Node.js mas também em outras plataformas, sendo mais indicado para o ambiente de produção

Funcionalidades

- Gestão de processos
- Manter as aplicações rodando em background
- Balanceamento de carga
- Monitoramento local e também remoto
- Inicialização da aplicação ao ligar o servidor
- Reinicialização da aplicação utilizando diversos parâmetros como memória ou horário agendado
- Configuração com quantidade de instâncias, local de armazenamento dos logs, variáveis de ambiente entre outros parâmetros

Instalação

É possível instalar o PM2 utilizando o npm ou yarn

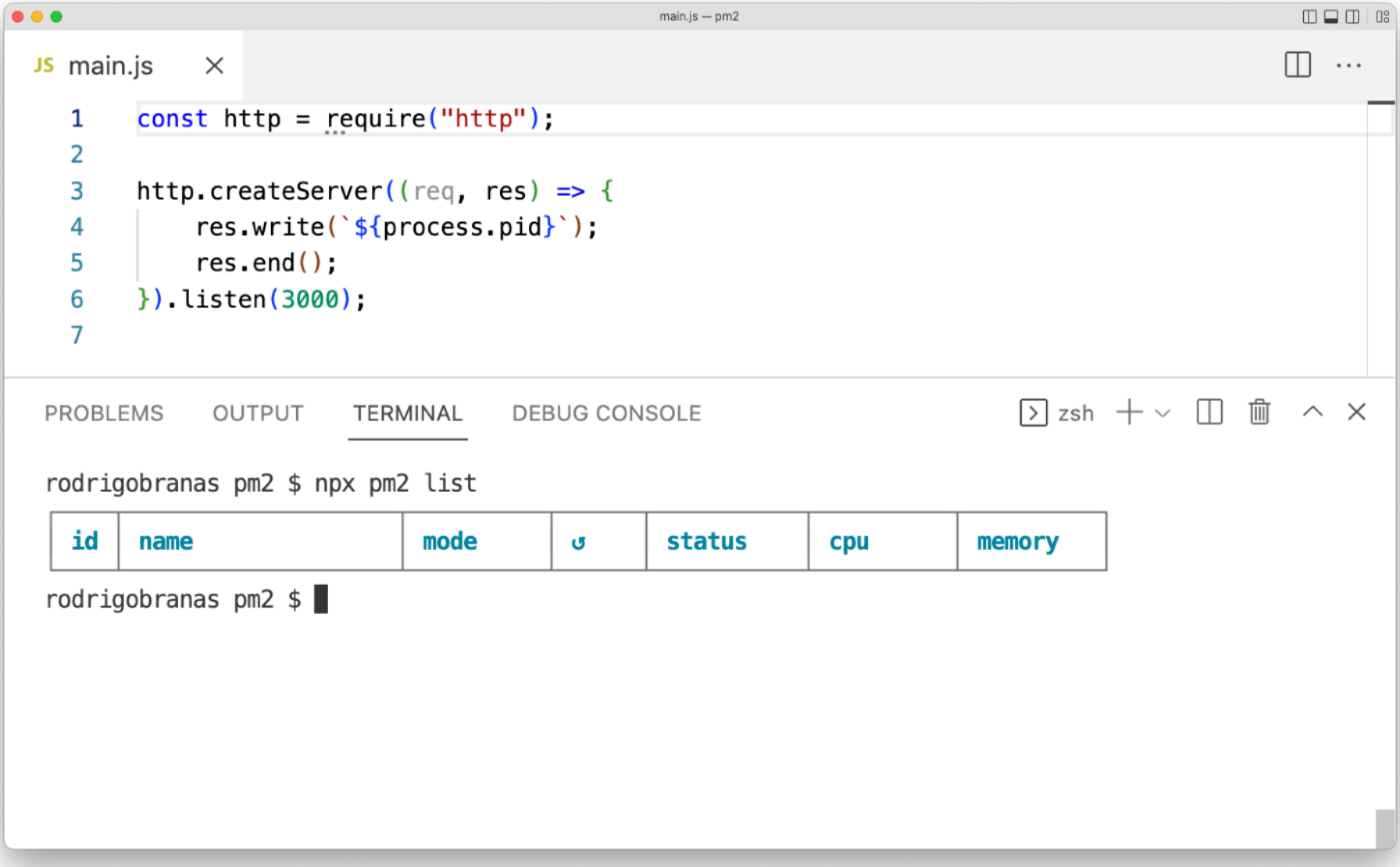
npm install pm2

ou

yarn add pm2

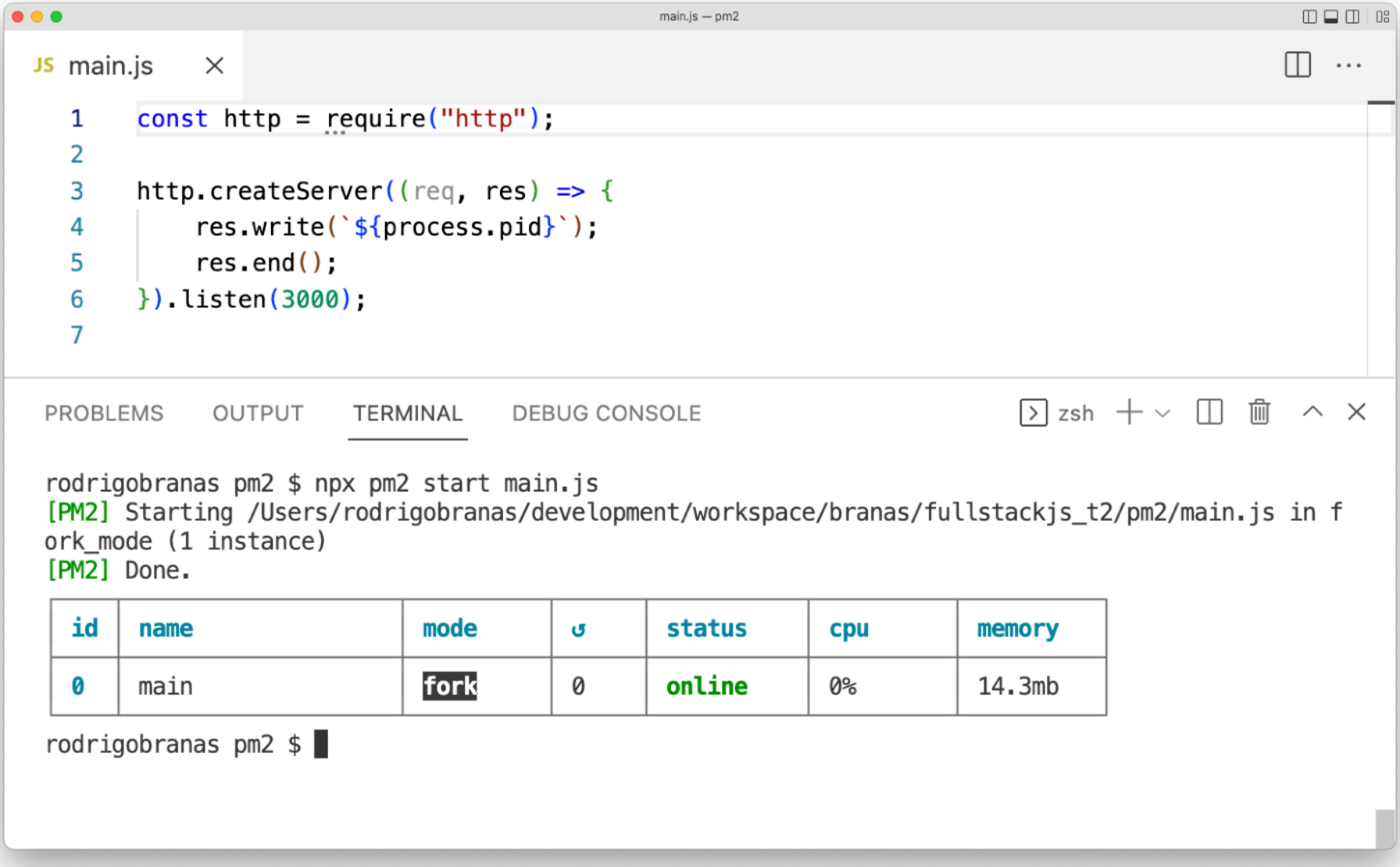
pm2 list

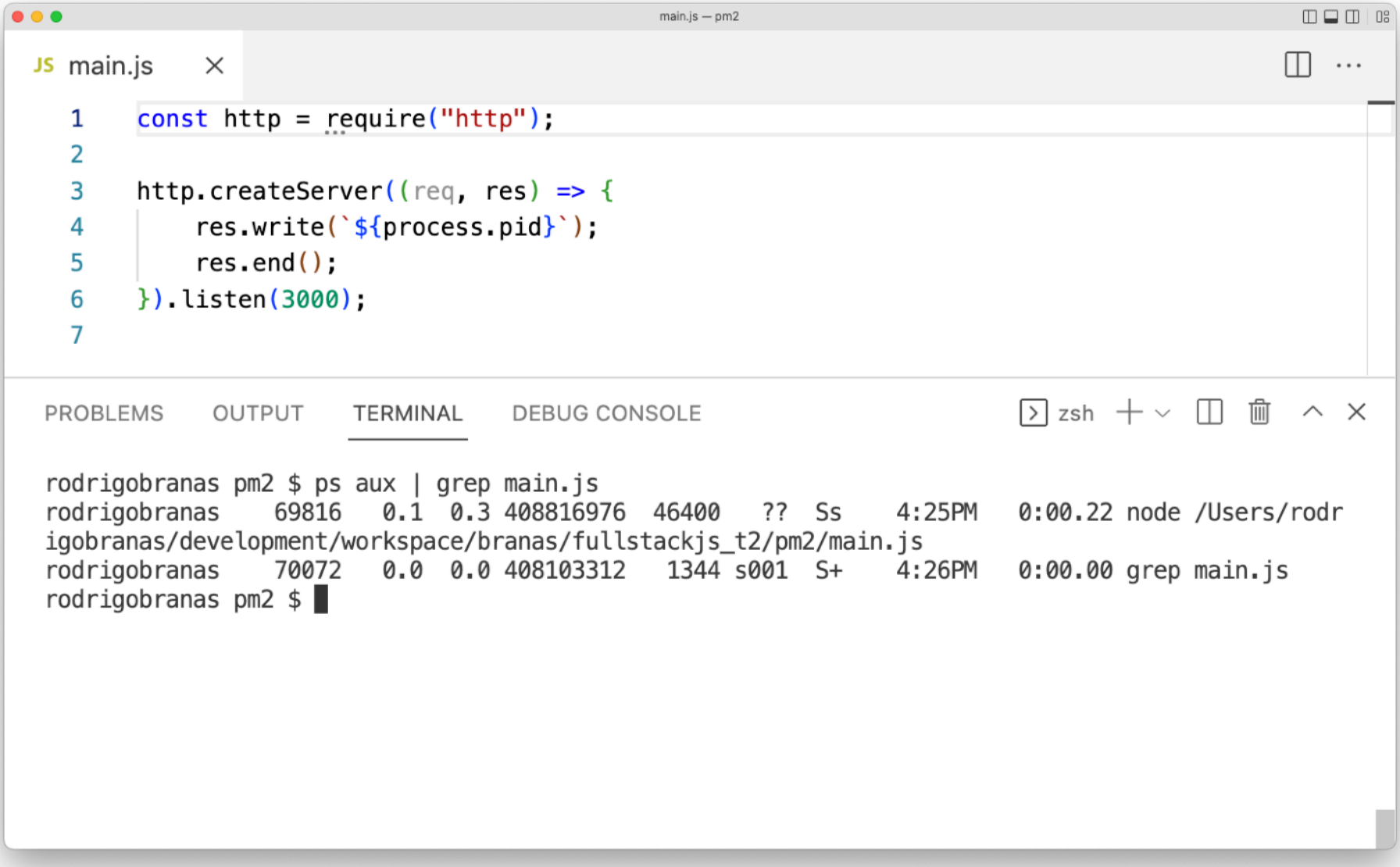
Lista os projetos gerenciados pelo PM2


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 list
id
name
mode
↺
status
cpu
memory
rodrigobranas pm2 $

pm2 start main.js

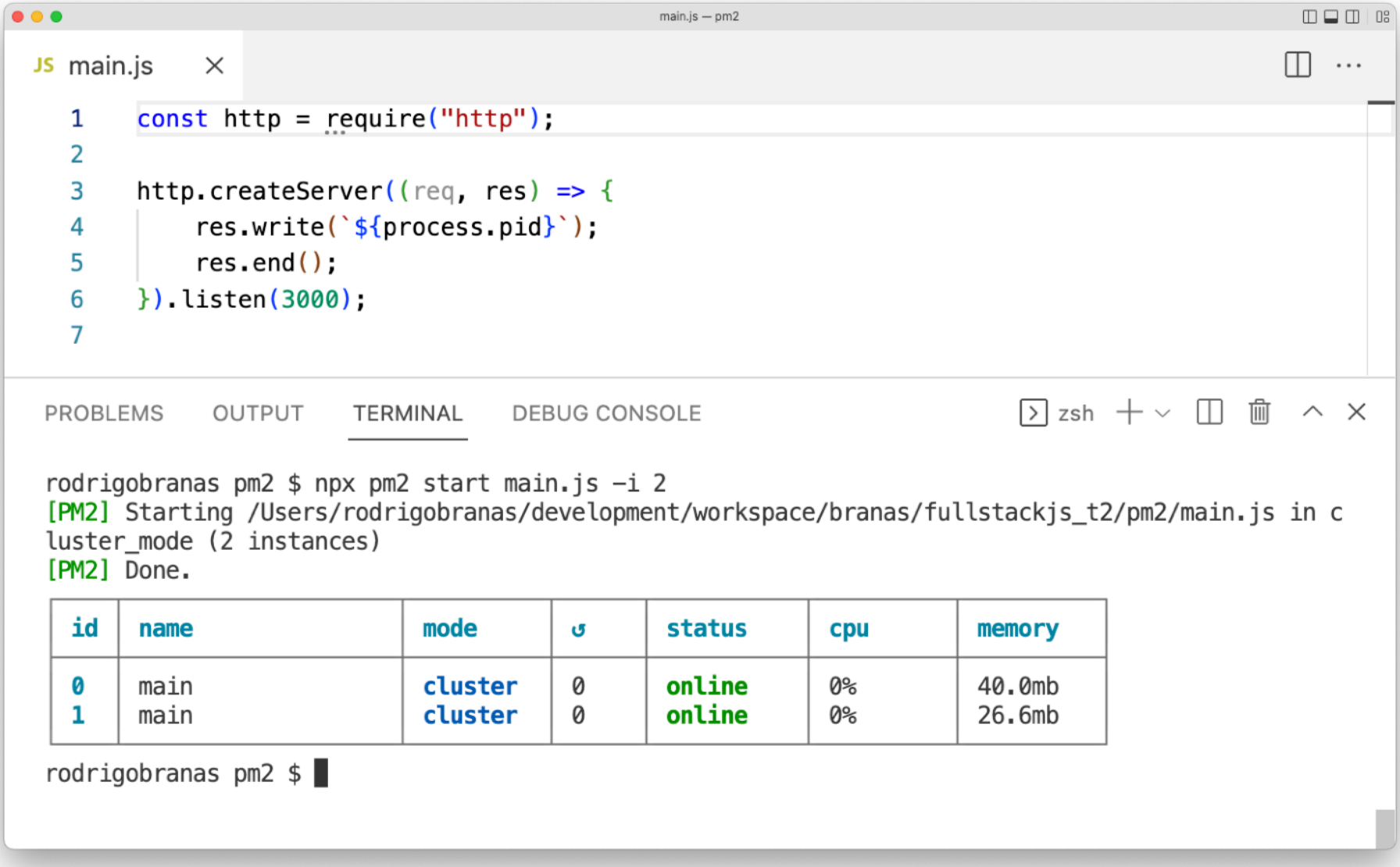
Inicia um projeto, nesse caso main.js


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start main.js
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in fork_mode (1 instance)
[PM2] Done.
id
name
mode
↺
status
cpu
memory
0
main
fork
0
online
0%
14.3mb
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ ps aux | grep main.js
rodrigobranas 69816 0.1 0.3 408816976 46400 ?? Ss 4:25PM 0:00.22 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 70072 0.0 0.0 408103312 1344 s001 S+ 4:26PM 0:00.00 grep main.js
rodrigobranas pm2 $

pm2 start main.js -i 2

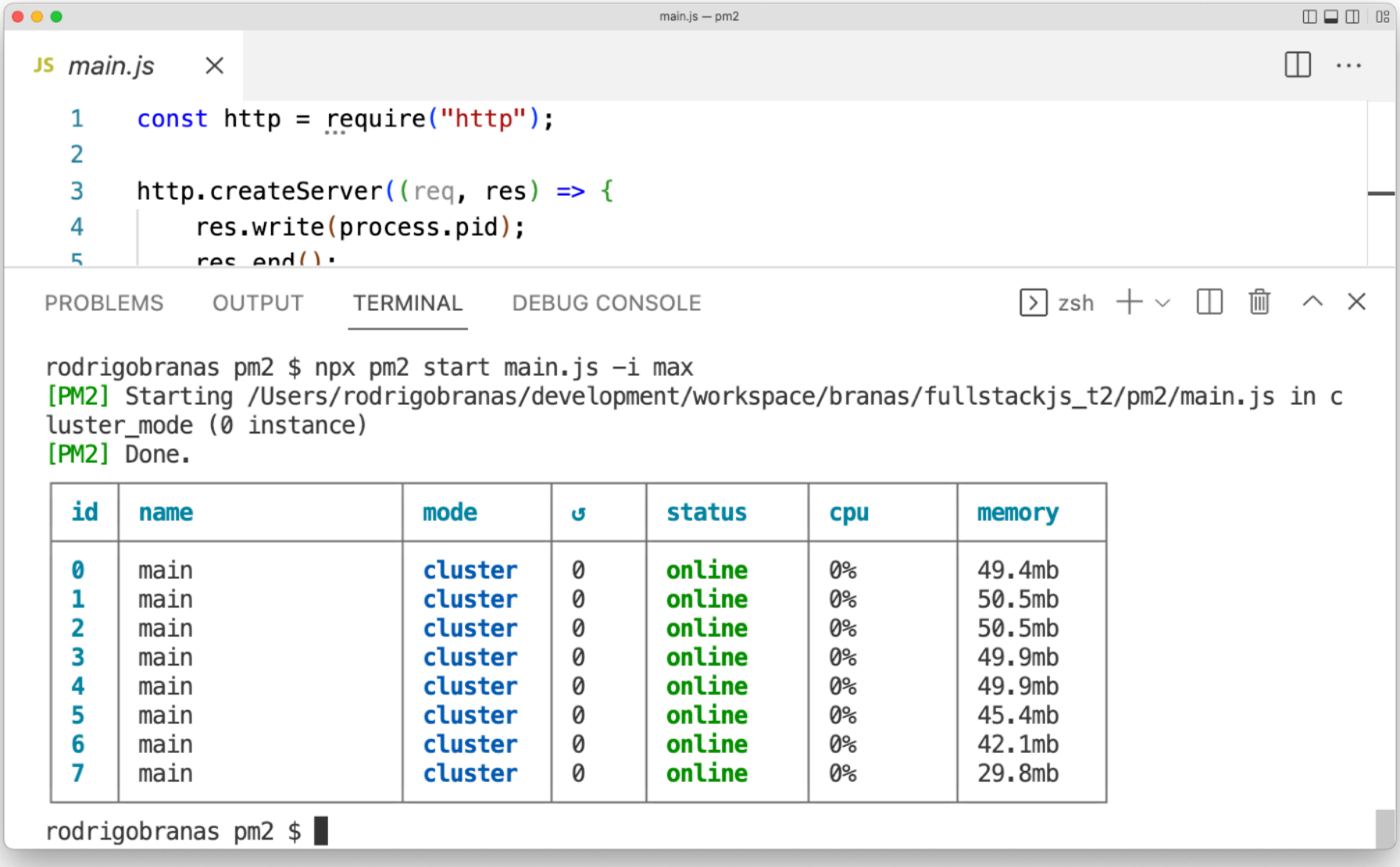
Inicia uma aplicação com múltiplas instâncias

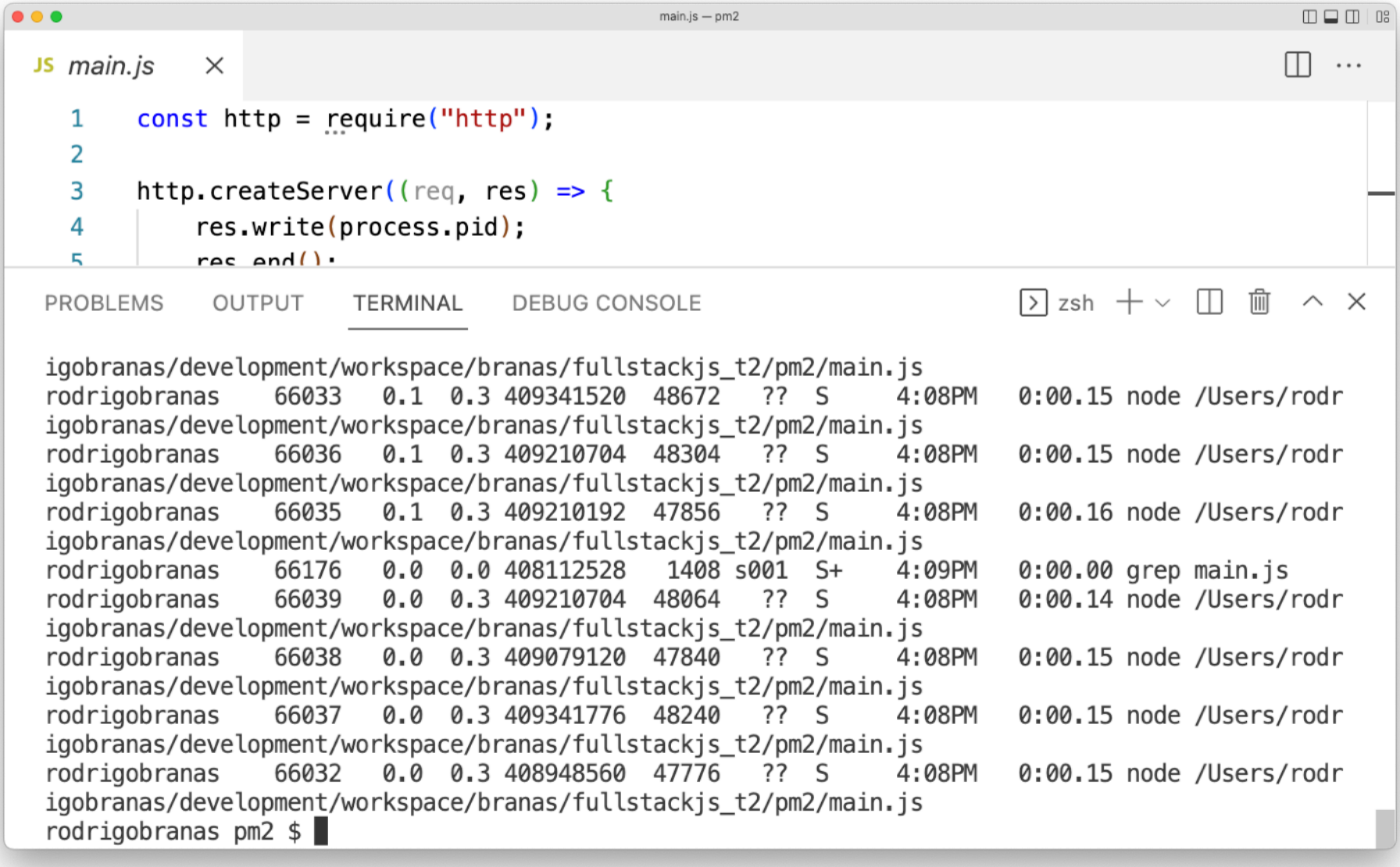

main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start main.js -i 2
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in cluster_mode (2 instances)
[PM2] Done.
id name mode ↺ status cpu memory
0 main cluster 0 online 0% 40.0mb
1 main cluster 0 online 0% 26.6mb
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ ps aux | grep main.js
rodrigobranas 70299 0.1 0.3 408948560 50800 ?? S 4:27PM 0:00.10 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 70300 0.1 0.3 408677968 50064 ?? S 4:27PM 0:00.09 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 70343 0.0 0.0 408103312 1344 s001 S+ 4:27PM 0:00.00 grep main.js
rodrigobranas pm2 $

pm2 start main.js -i max

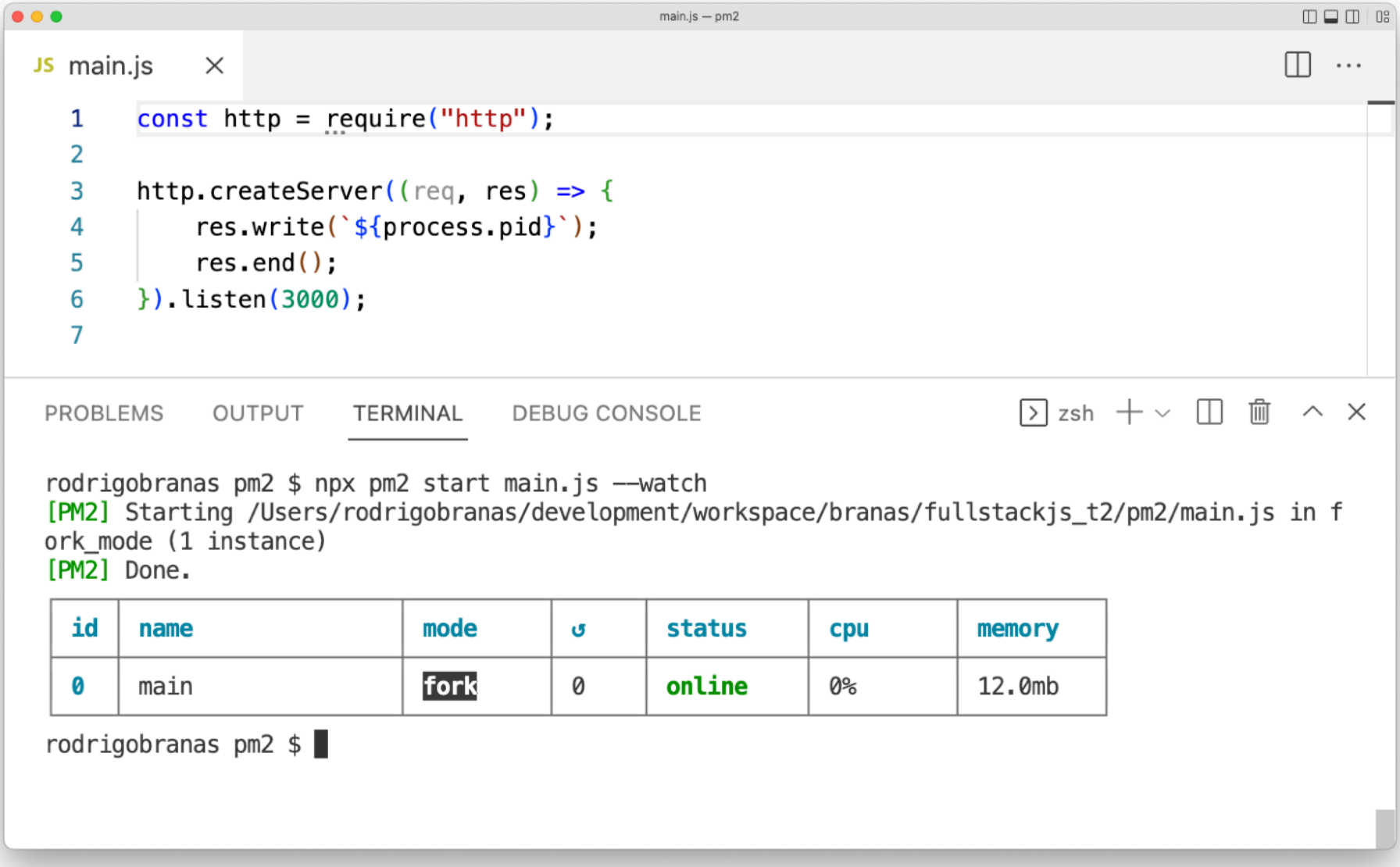
Inicia uma aplicação com múltiplas instâncias


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(process.pid);
5 res.end();
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start main.js -i max
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in cluster_mode (0 instance)
[PM2] Done.
id name mode ↺ status cpu memory
0 main cluster 0 online 0% 49.4mb
1 main cluster 0 online 0% 50.5mb
2 main cluster 0 online 0% 50.5mb
3 main cluster 0 online 0% 49.9mb
4 main cluster 0 online 0% 49.9mb
5 main cluster 0 online 0% 45.4mb
6 main cluster 0 online 0% 42.1mb
7 main cluster 0 online 0% 29.8mb
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(process.pid);
5 res.end();
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66033 0.1 0.3 409341520 48672 ?? S 4:08PM 0:00.15 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66036 0.1 0.3 409210704 48304 ?? S 4:08PM 0:00.15 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66035 0.1 0.3 409210192 47856 ?? S 4:08PM 0:00.16 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66176 0.0 0.0 408112528 1408 s001 S+ 4:09PM 0:00.00 grep main.js
rodrigobranas 66039 0.0 0.3 409210704 48064 ?? S 4:08PM 0:00.14 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66038 0.0 0.3 409079120 47840 ?? S 4:08PM 0:00.15 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66037 0.0 0.3 409341776 48240 ?? S 4:08PM 0:00.15 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas 66032 0.0 0.3 408948560 47776 ?? S 4:08PM 0:00.15 node /Users/rodr
igobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js
rodrigobranas pm2 $

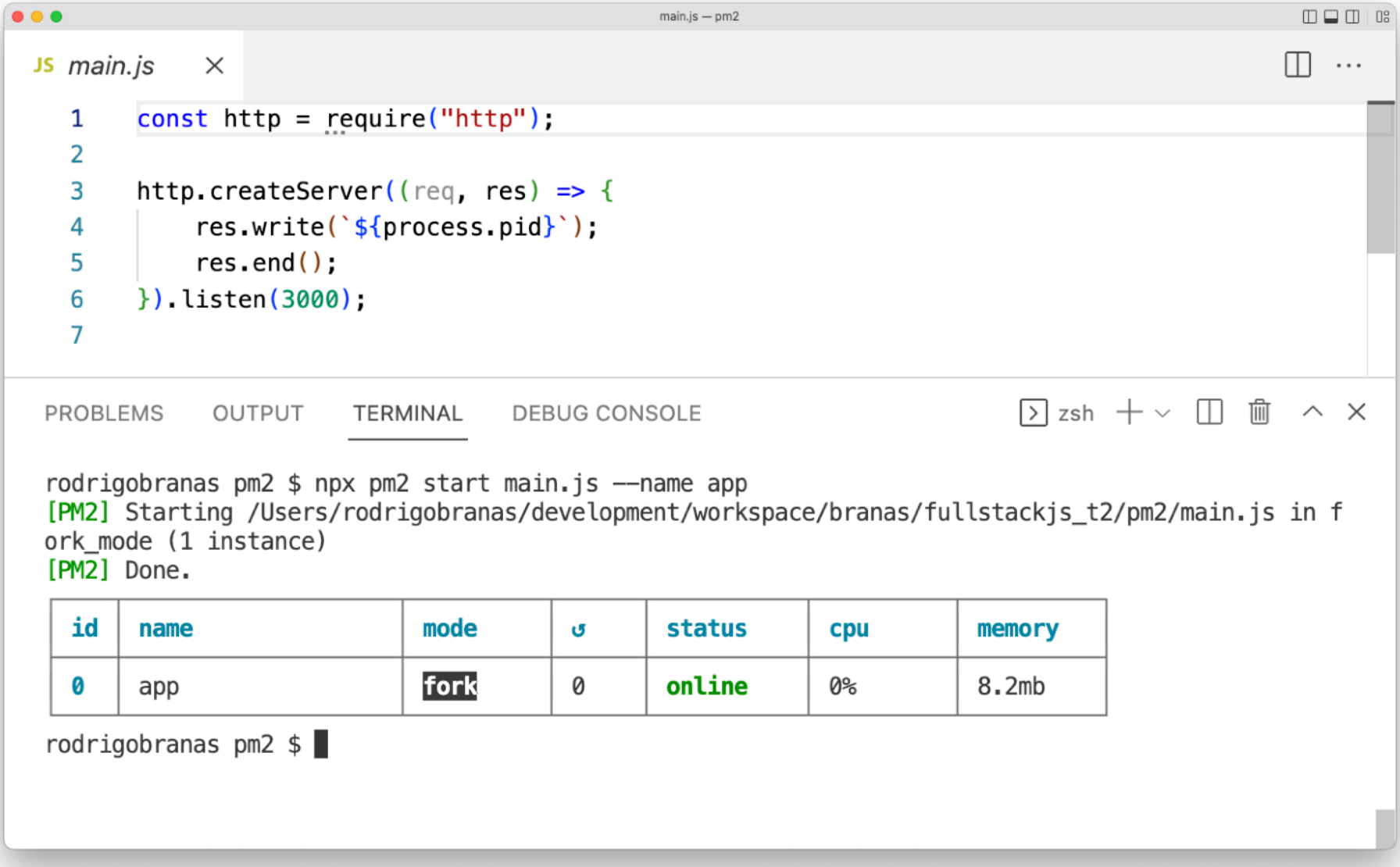
pm2 start main.js --watch

Inicia uma aplicação e monitora mudanças no código-fonte, reiniciando de forma automática


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start main.js --watch
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in fork_mode (1 instance)
[PM2] Done.
id
name
mode
↺
status
cpu
memory
0
main
fork
0
online
0%
12.0mb
rodrigobranas pm2 $

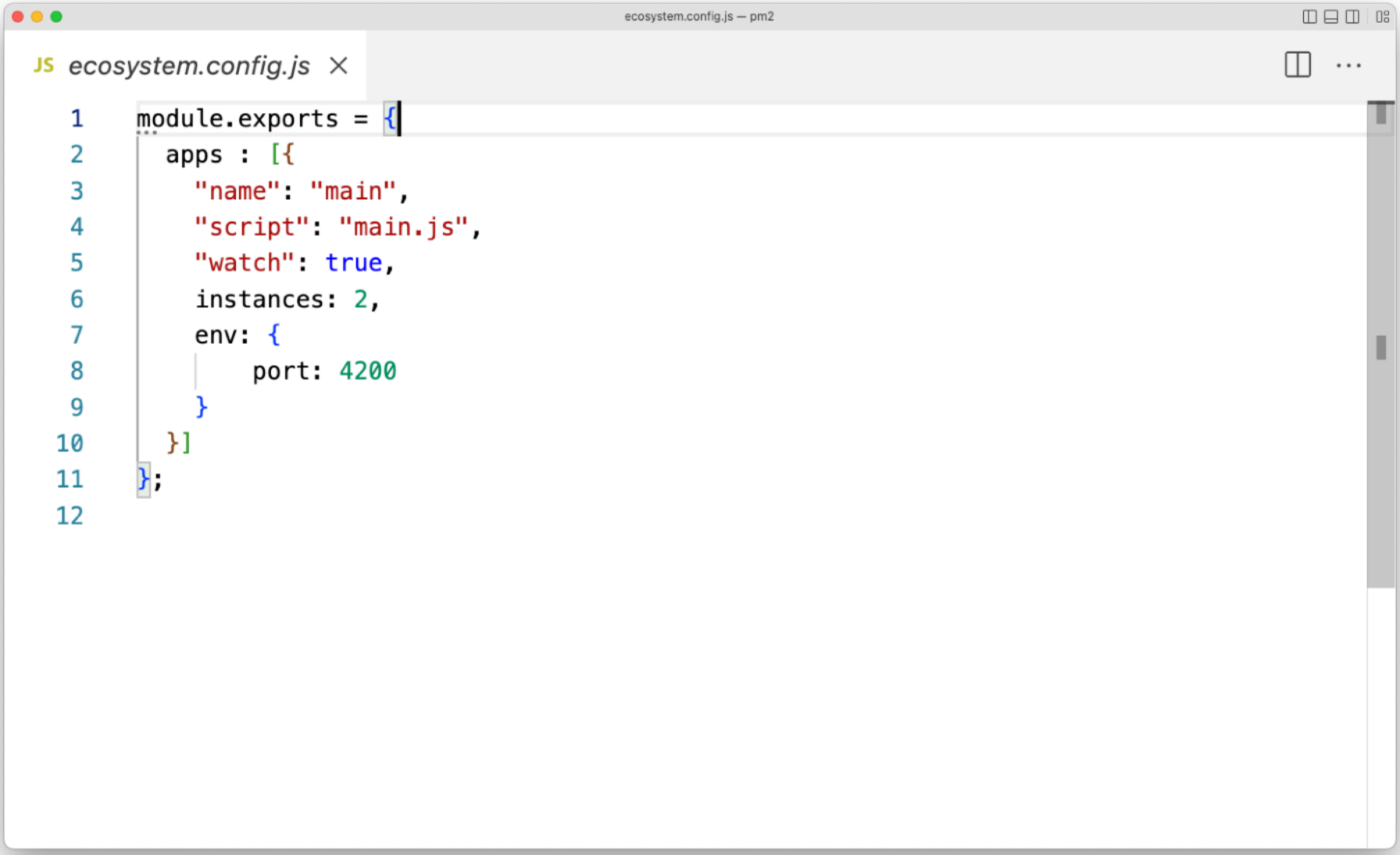
pm2 start main.js --name app

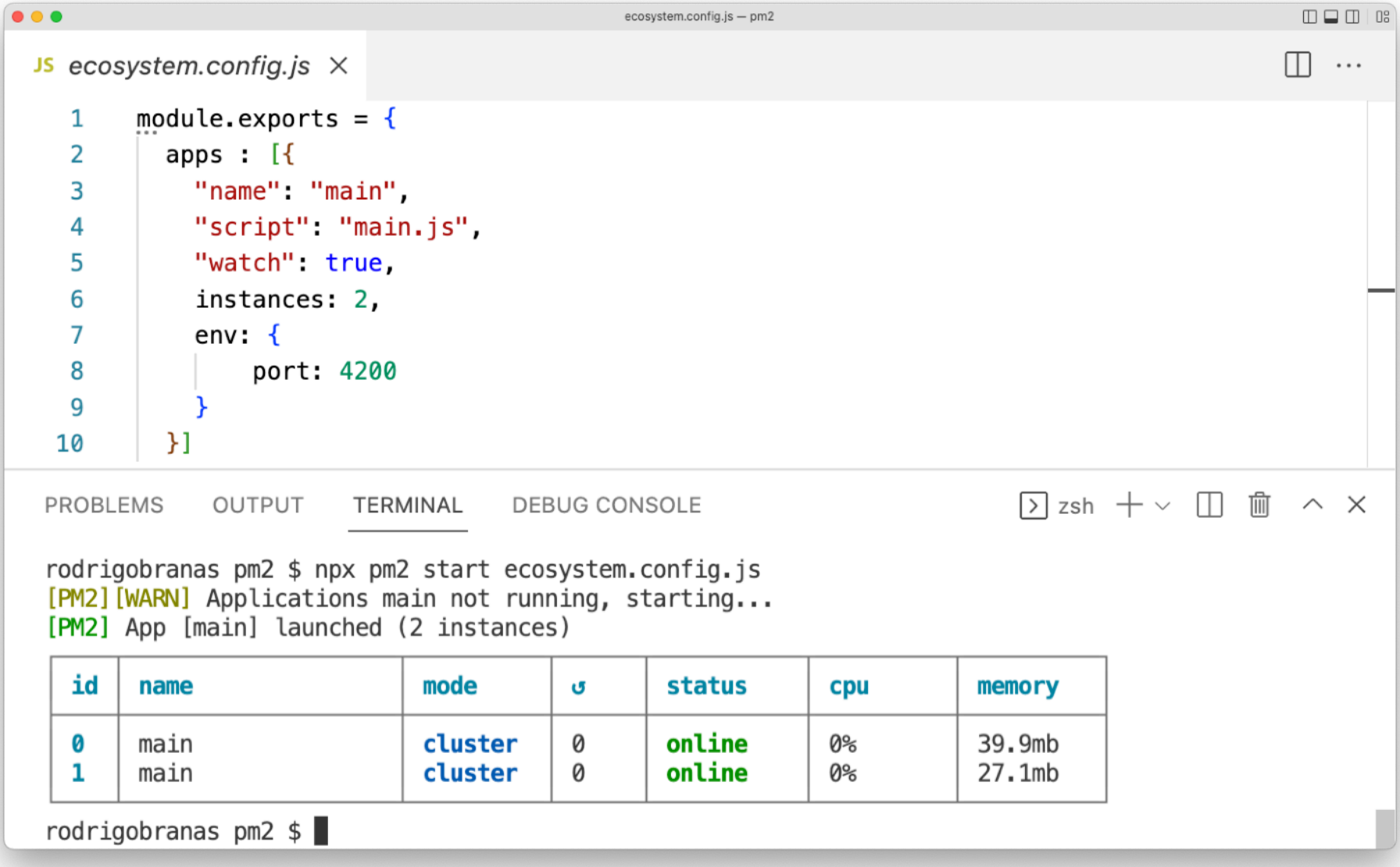
Inicia uma aplicação e atribui um nome


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start main.js --name app
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in fork_mode (1 instance)
[PM2] Done.
id
name
mode
↺
status
cpu
memory
0
app
fork
0
online
0%
8.2mb
rodrigobranas pm2 $

pm2 start config_file

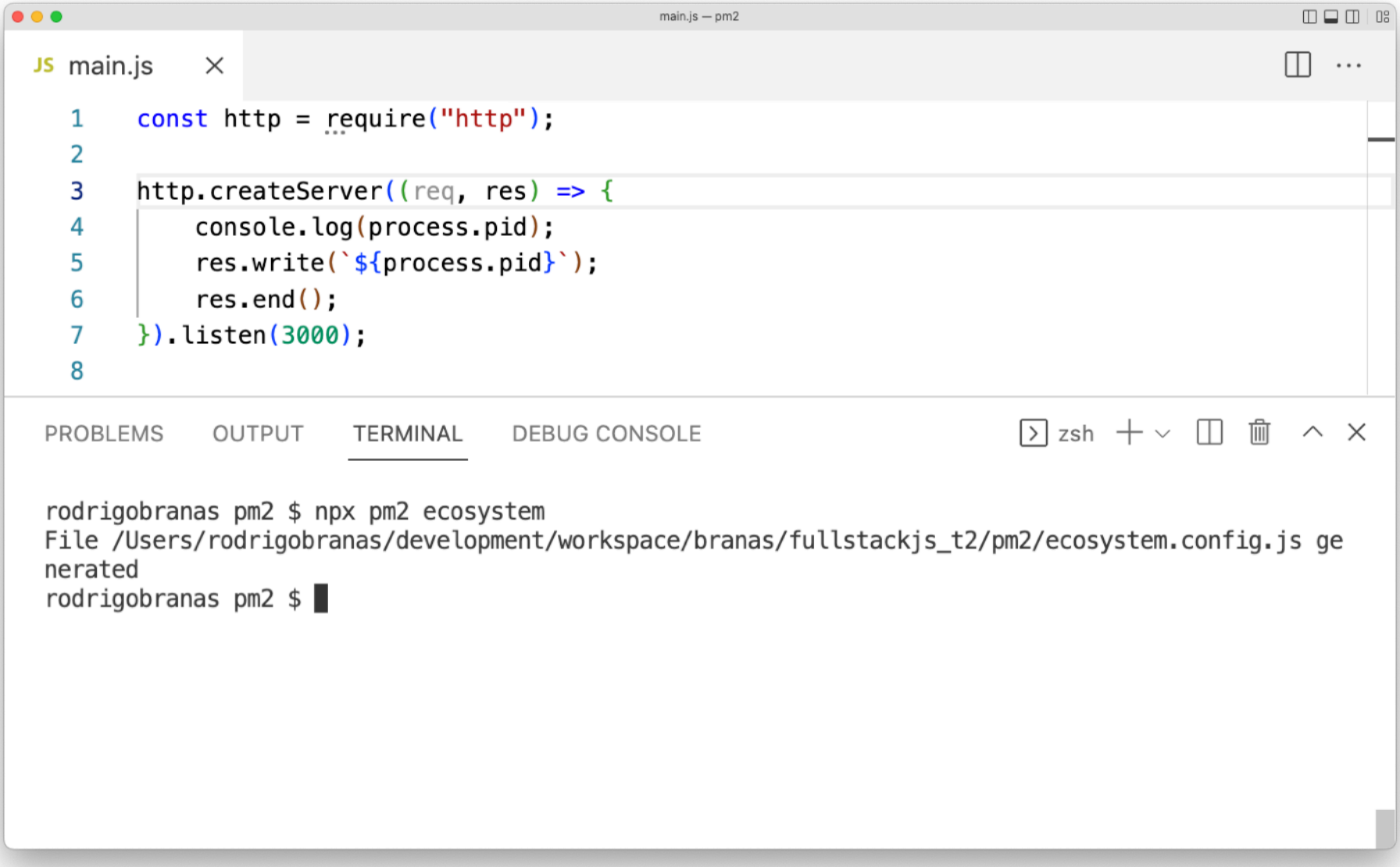
Inicia uma aplicação com base em um arquivo de configurações com a quantidade de instâncias, local de armazenamento dos logs, variáveis de ambiente entre outros parâmetros

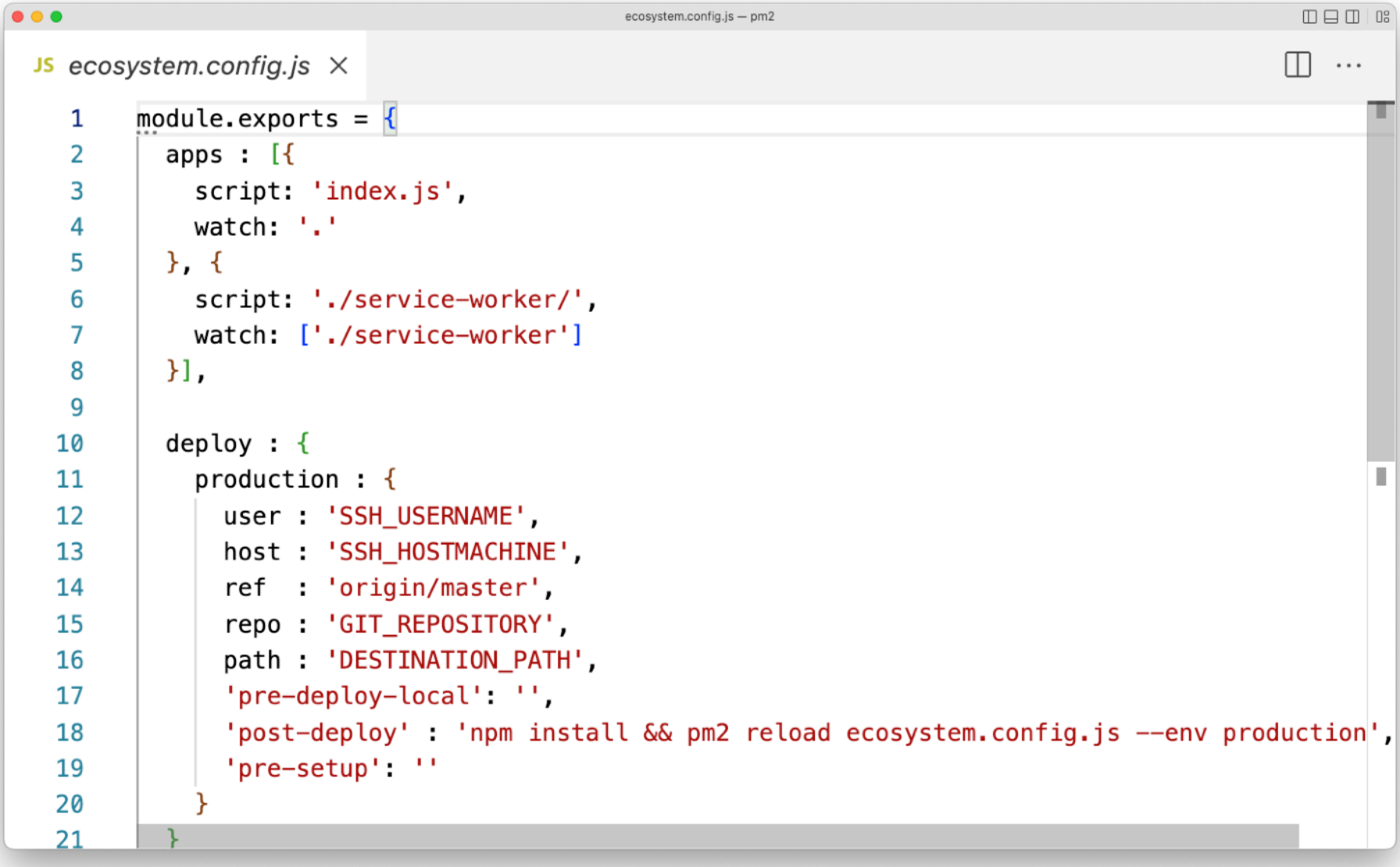

ecosystem.config.js — pm2
ecosystem.config.js
module.exports = {
apps : [{
"name": "main",
"script": "main.js",
"watch": true,
instances: 2,
env: {
port: 4200
}
}]
};

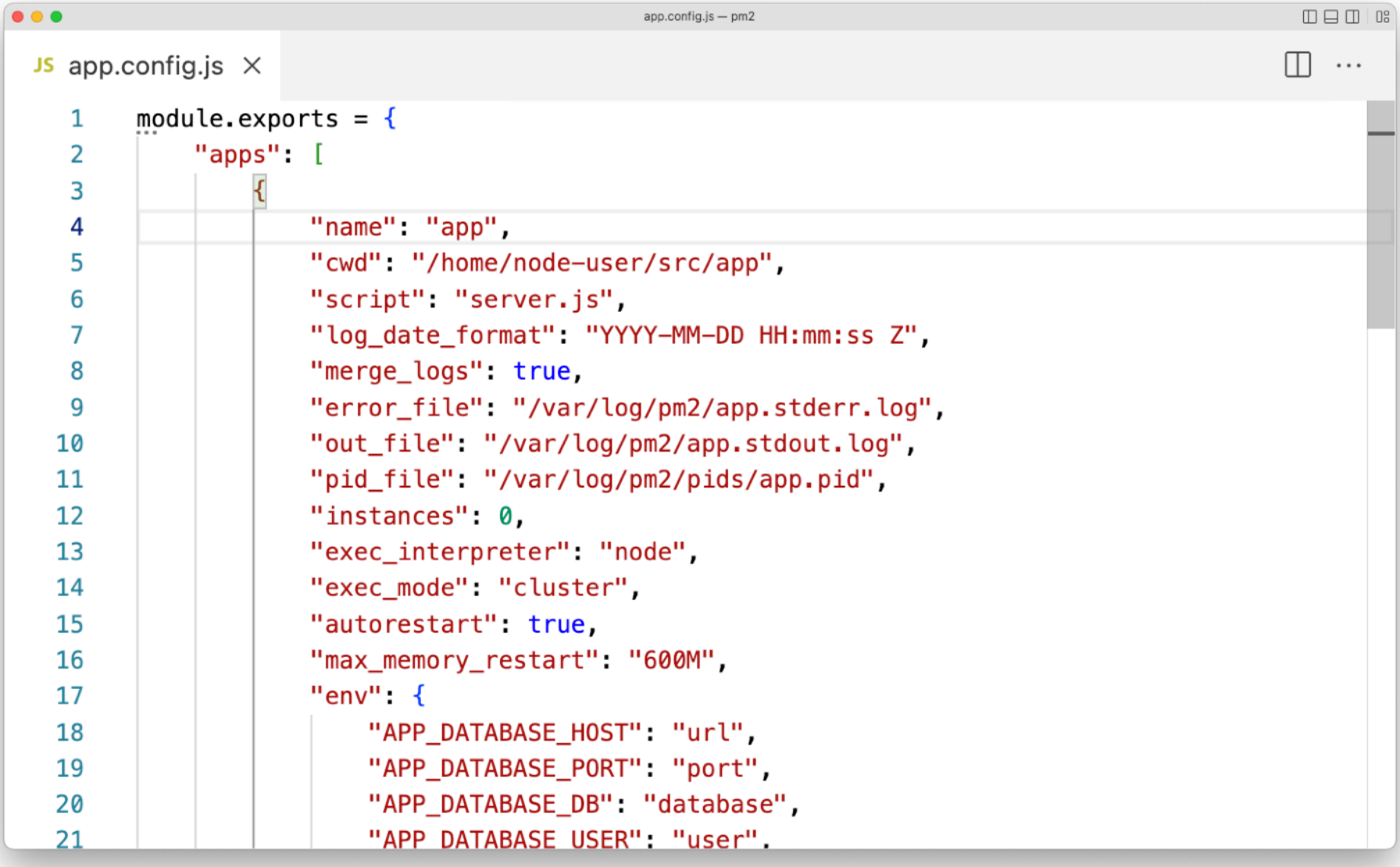

ecosystem.config.js — pm2
ecosystem.config.js
1 module.exports = {
2 apps : [{
3 "name": "main",
4 "script": "main.js",
5 "watch": true,
6 instances: 2,
7 env: {
8 port: 4200
9 }
10 }]
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 start ecosystem.config.js
[PM2][WARN] Applications main not running, starting...
[PM2] App [main] launched (2 instances)
id name mode ↺ status cpu memory
0 main cluster 0 online 0% 39.9mb
1 main cluster 0 online 0% 27.1mb
rodrigobranas pm2 $

pm2 ecosystem

Gera um arquivo de configurações para a aplicação

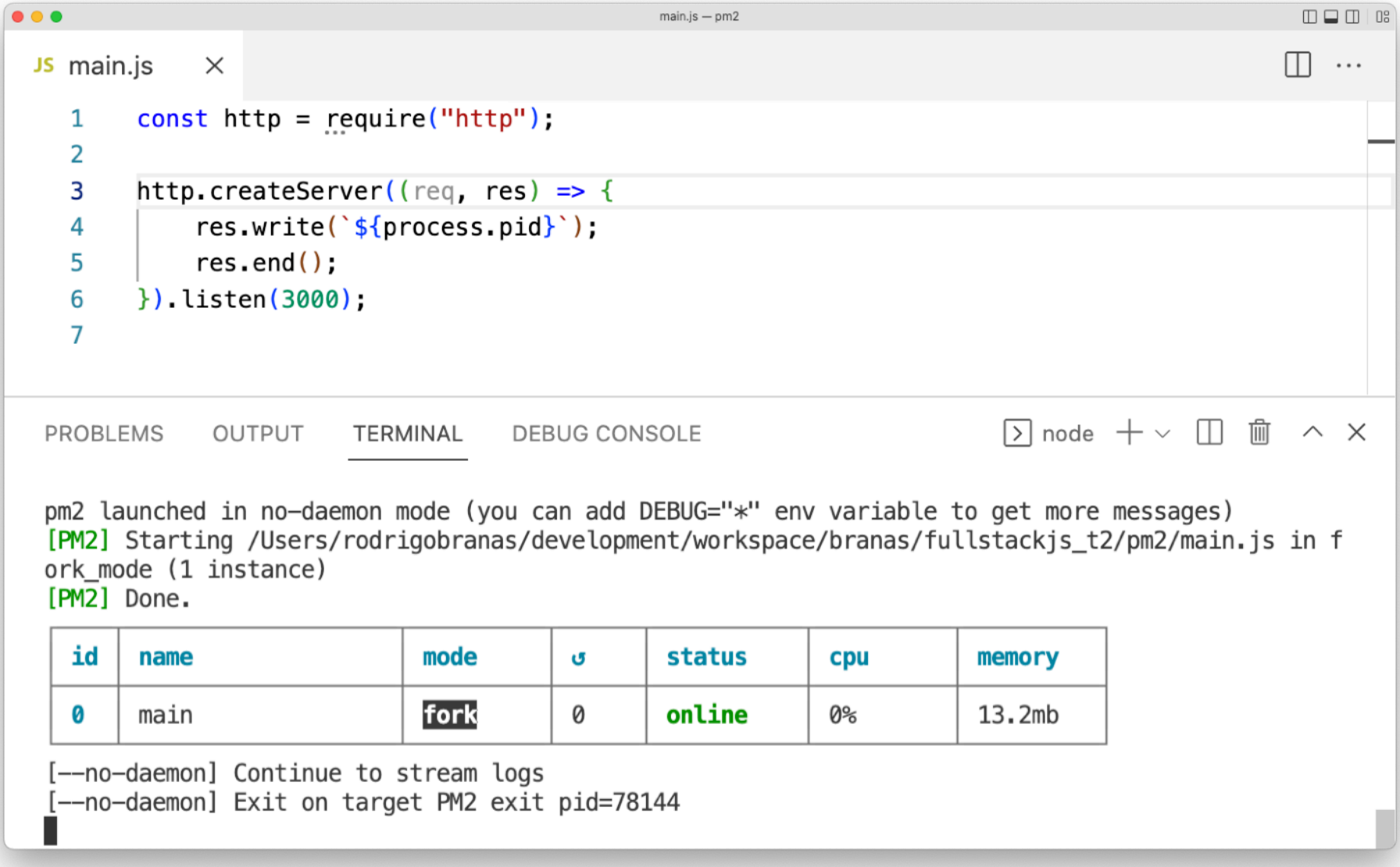

main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 console.log(process.pid);
5 res.write(`${process.pid}`);
6 res.end();
7 }).listen(3000);
8
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 ecosystem
File /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/ecosystem.config.js ge
nerated
rodrigobranas pm2 $


ecosystem.config.js — pm2
ecosystem.config.js
module.exports = {
apps : [{
script: 'index.js',
watch: '.'
}, {
script: './service-worker/',
watch: ['./service-worker']
}],
deploy : {
production : {
user : 'SSH_USERNAME',
host : 'SSH_HOSTMACHINE',
ref : 'origin/master',
repo : 'GIT_REPOSITORY',
path : 'DESTINATION_PATH',
'pre-deploy-local': '',
'post-deploy' : 'npm install && pm2 reload ecosystem.config.js --env production',
'pre-setup': ''
}
}


app.config.js — pm2
app.config.js
module.exports = {
"apps": [
{
"name": "app",
"cwd": "/home/node-user/src/app",
"script": "server.js",
"log_date_format": "YYYY-MM-DD HH:mm:ss Z",
"merge_logs": true,
"error_file": "/var/log/pm2/app.stderr.log",
"out_file": "/var/log/pm2/app.stdout.log",
"pid_file": "/var/log/pm2/pids/app.pid",
"instances": 0,
"exec_interpreter": "node",
"exec_mode": "cluster",
"autorestart": true,
"max_memory_restart": "600M",
"env": {
"APP_DATABASE_HOST": "url",
"APP_DATABASE_PORT": "port",
"APP_DATABASE_DB": "database",
"APP_DATABASE_USER": "user",

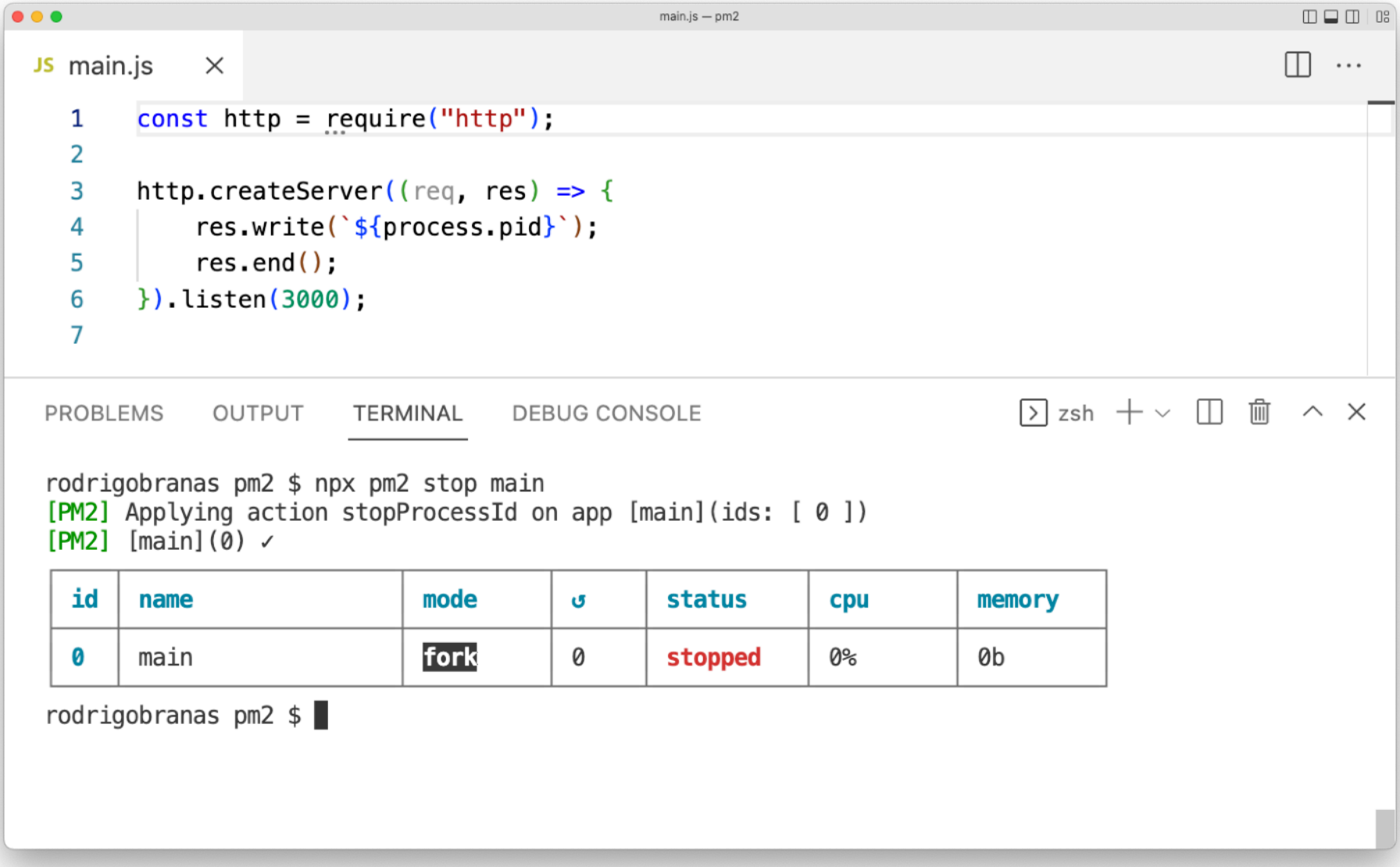
pm2 start main.js --no-daemon

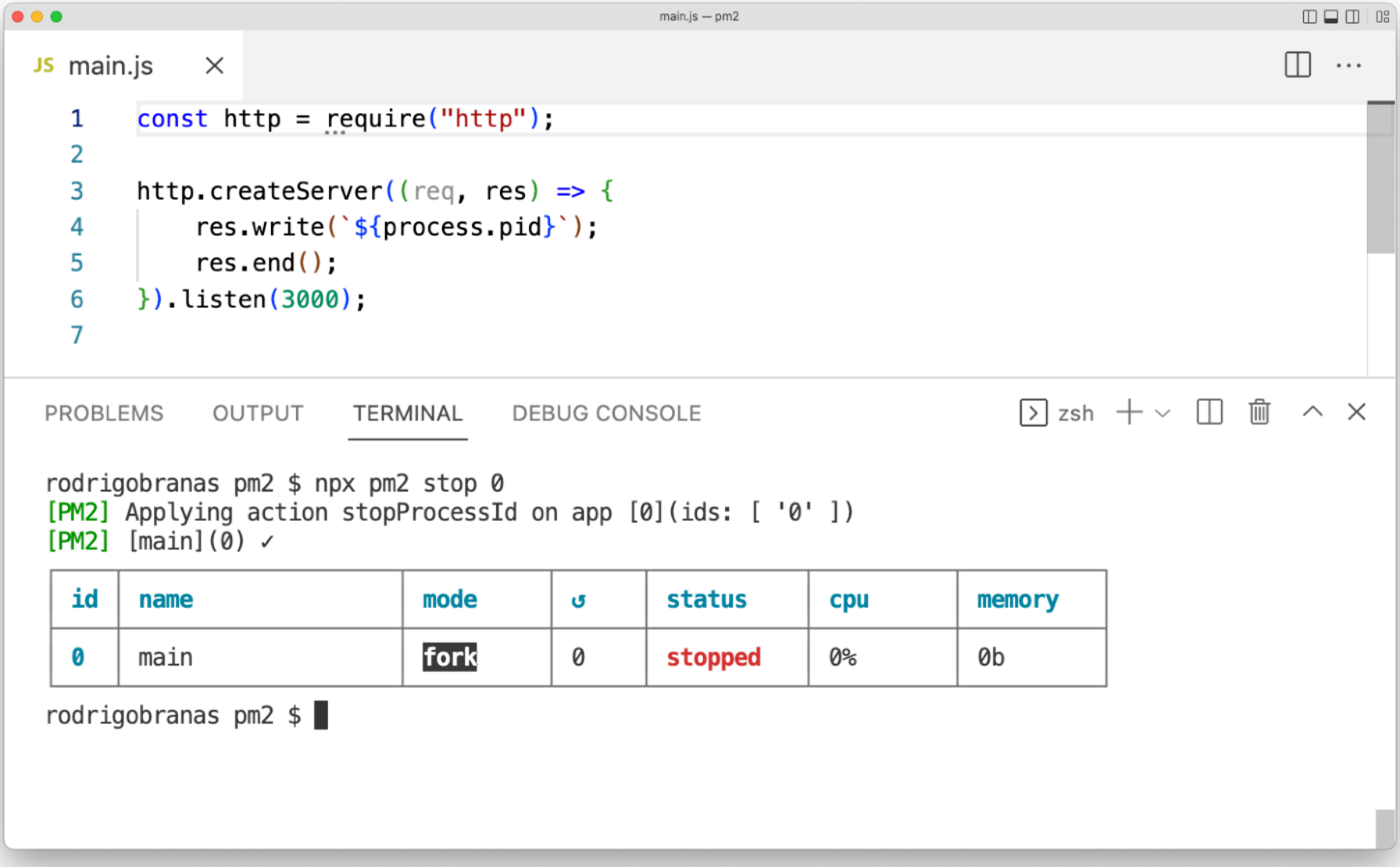
Inicia a aplicação diretamente

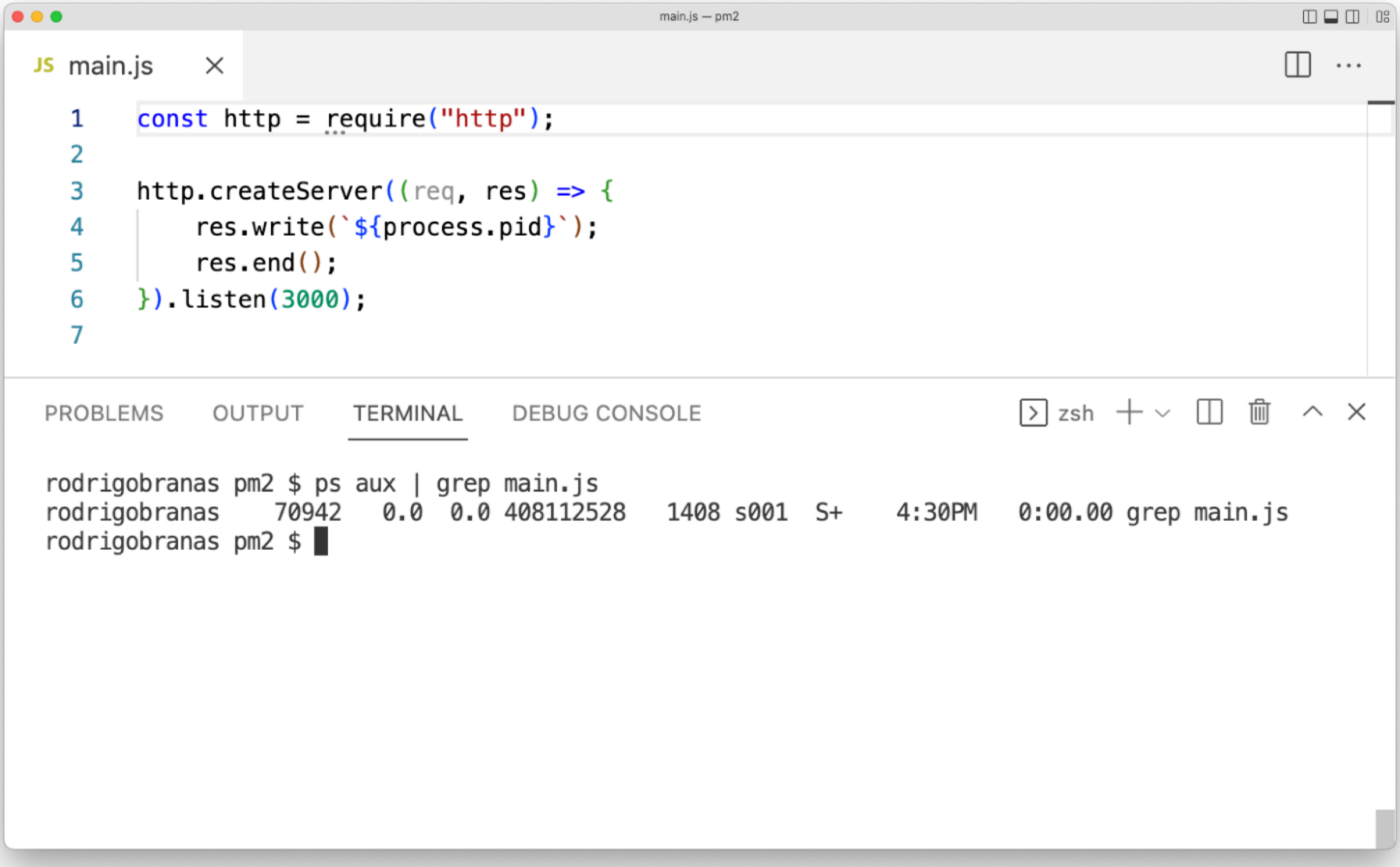

main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
node
pm2 launched in no-daemon mode (you can add DEBUG="*" env variable to get more messages)
[PM2] Starting /Users/rodrigobranas/development/workspace/branas/fullstackjs_t2/pm2/main.js in f
ork_mode (1 instance)
[PM2] Done.
id
name
mode
↺
status
cpu
memory
0
main
fork
0
online
0%
13.2mb
[--no-daemon] Continue to stream logs
[--no-daemon] Exit on target PM2 exit pid=78144

pm2 stop project_name ou id

Para um projeto Node.js

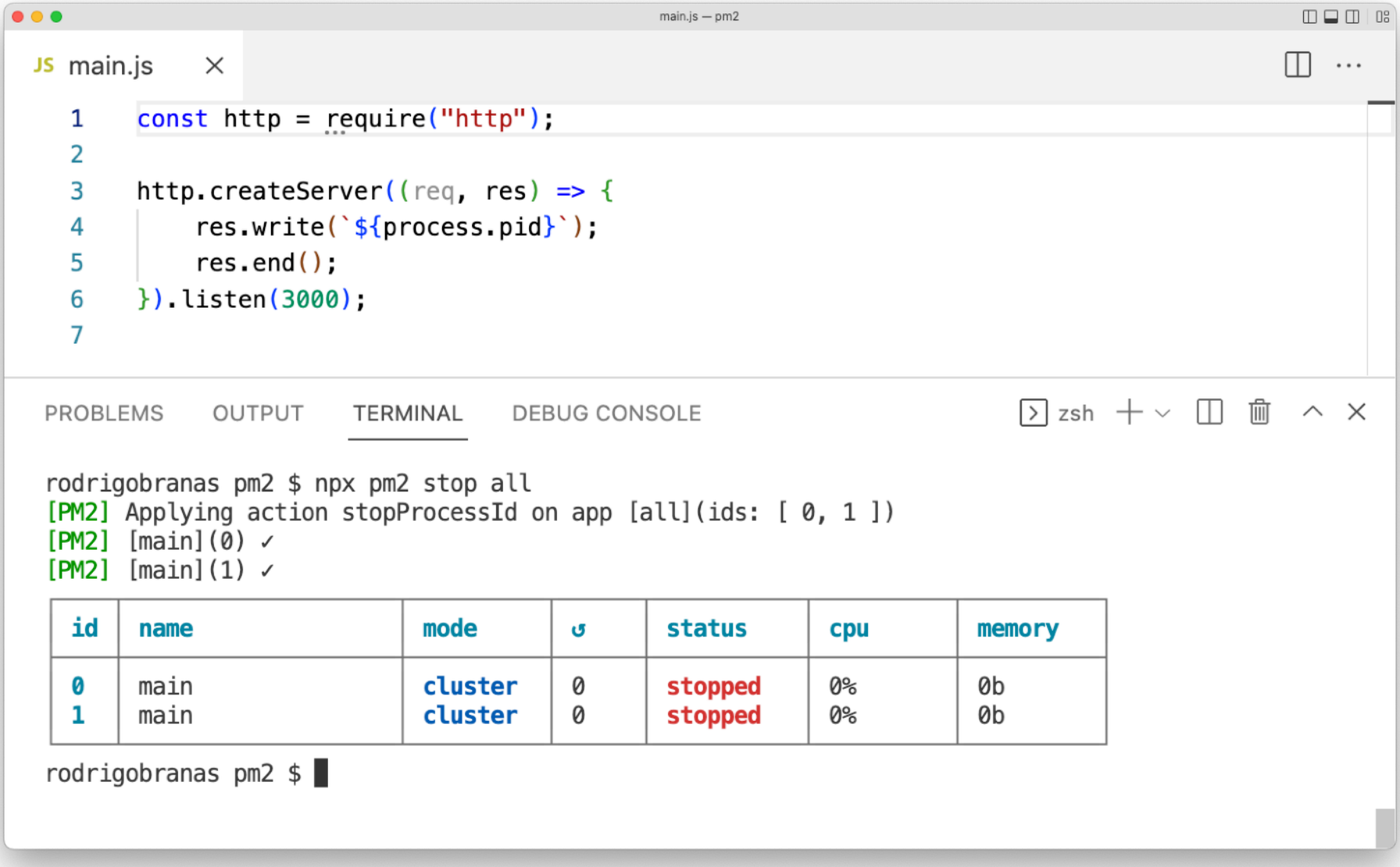

main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 stop main
[PM2] Applying action stopProcessId on app [main](ids: [0])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
0
main
fork
0
stopped
0%
0b
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 stop 0
[PM2] Applying action stopProcessId on app [0](ids: ['0'])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
0
main
fork
0
stopped
0%
0b
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ ps aux | grep main.js
rodrigobranas 70942 0.0 0.0 408112528 1408 s001 S+ 4:30PM 0:00.00 grep main.js
rodrigobranas pm2 $

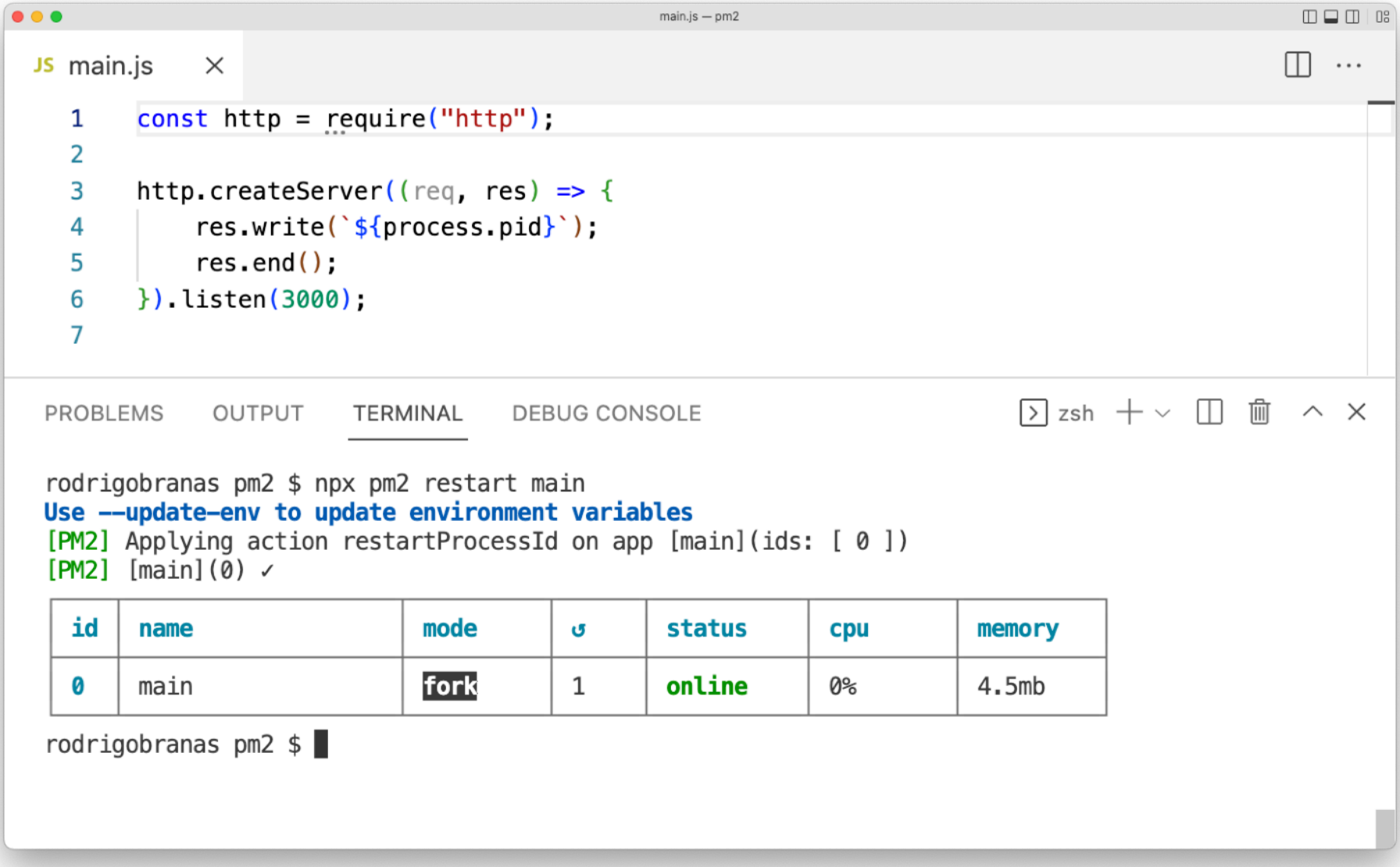
pm2 stop all

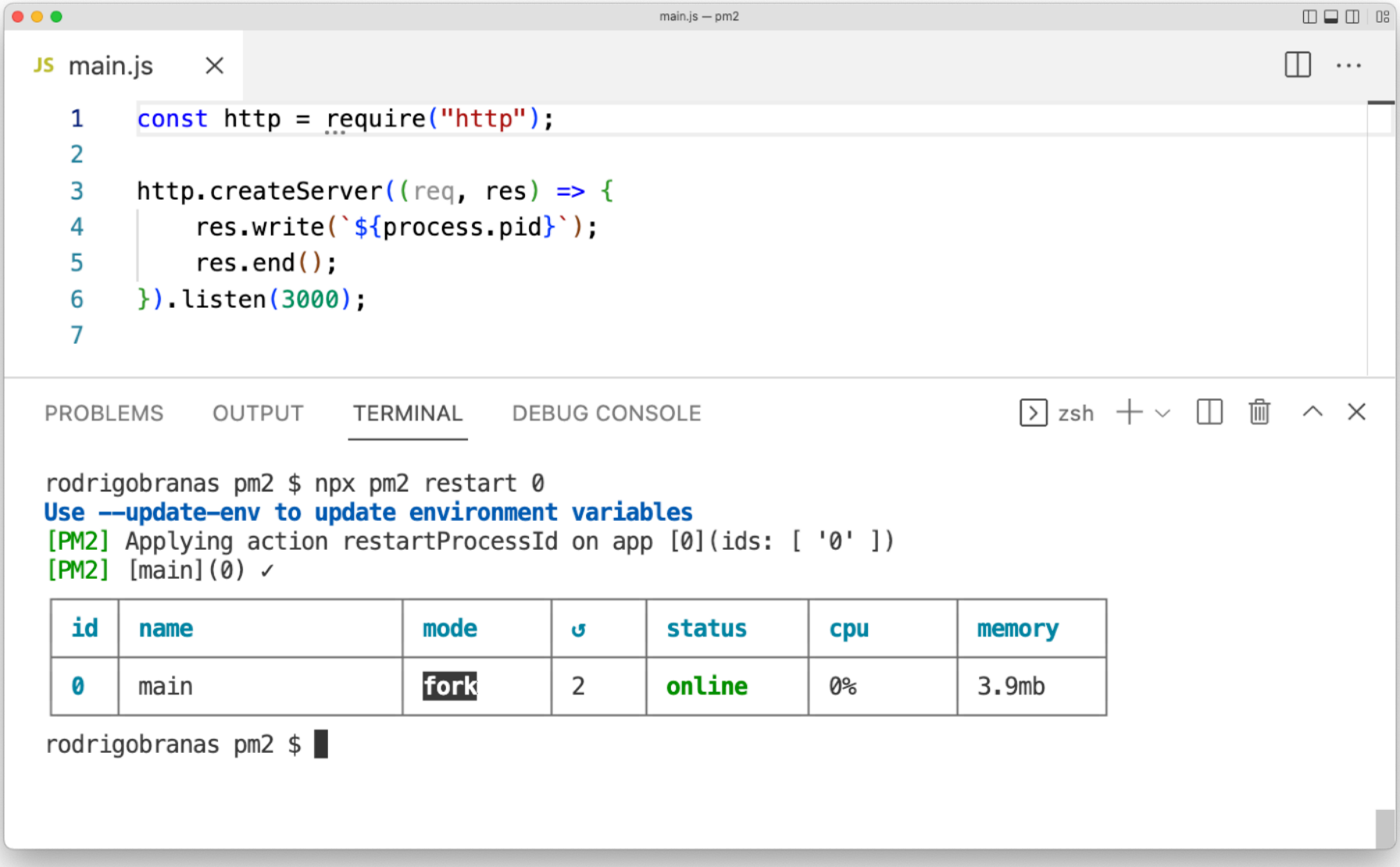
Para todas as aplicações


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 stop all
[PM2] Applying action stopProcessId on app [all](ids: [0, 1])
[PM2] [main](0) ✓
[PM2] [main](1) ✓
id
name
mode
↺
status
cpu
memory
0
main
cluster
0
stopped
0%
0b
1
main
cluster
0
stopped
0%
0b
rodrigobranas pm2 $

pm2 restart project_name ou id

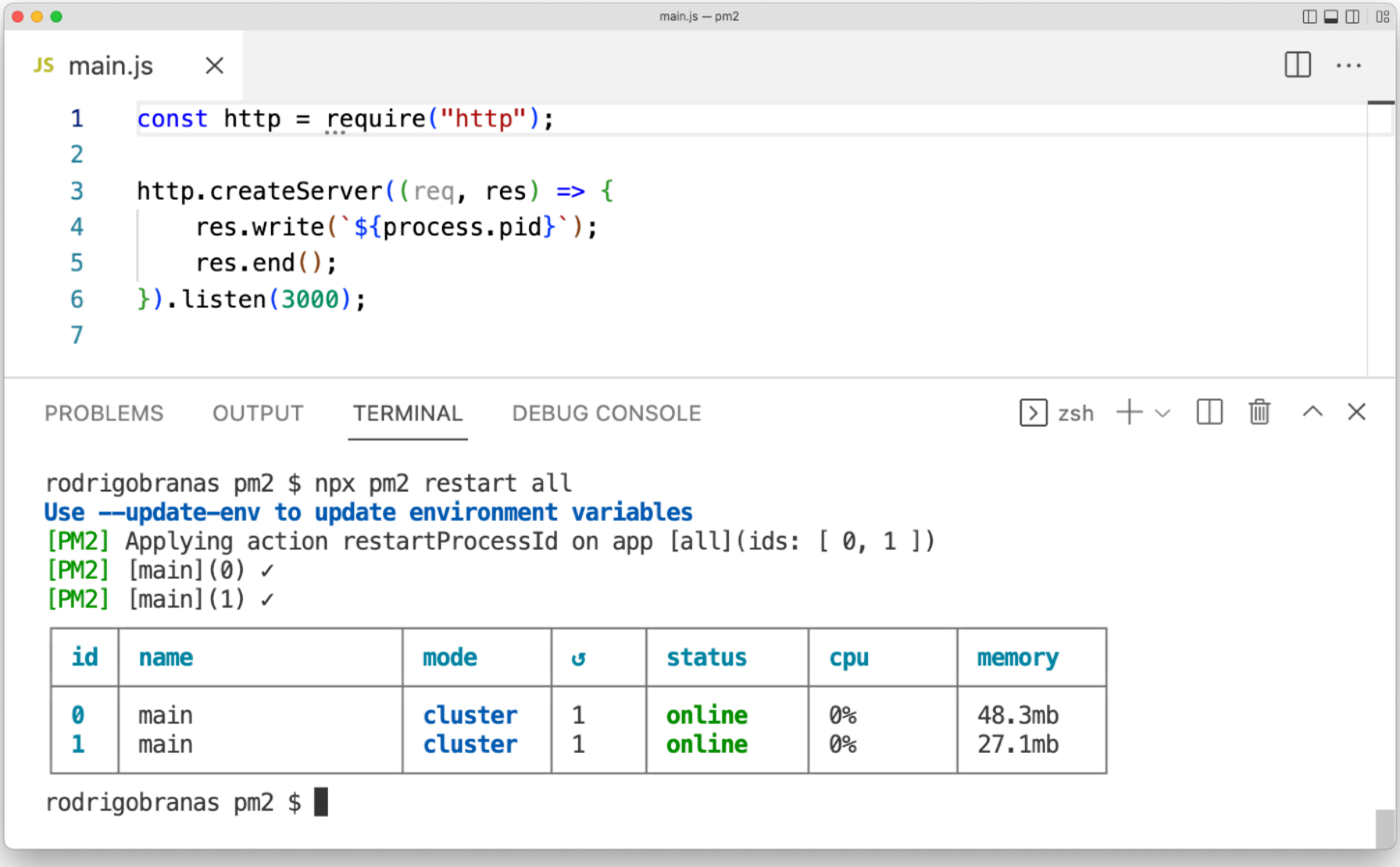
Reinicia um projeto Node.js


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 restart main
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [main](ids: [0])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
0
main
fork
1
online
0%
4.5mb
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 restart 0
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [0](ids: ['0'])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
0
main
fork
2
online
0%
3.9mb
rodrigobranas pm2 $

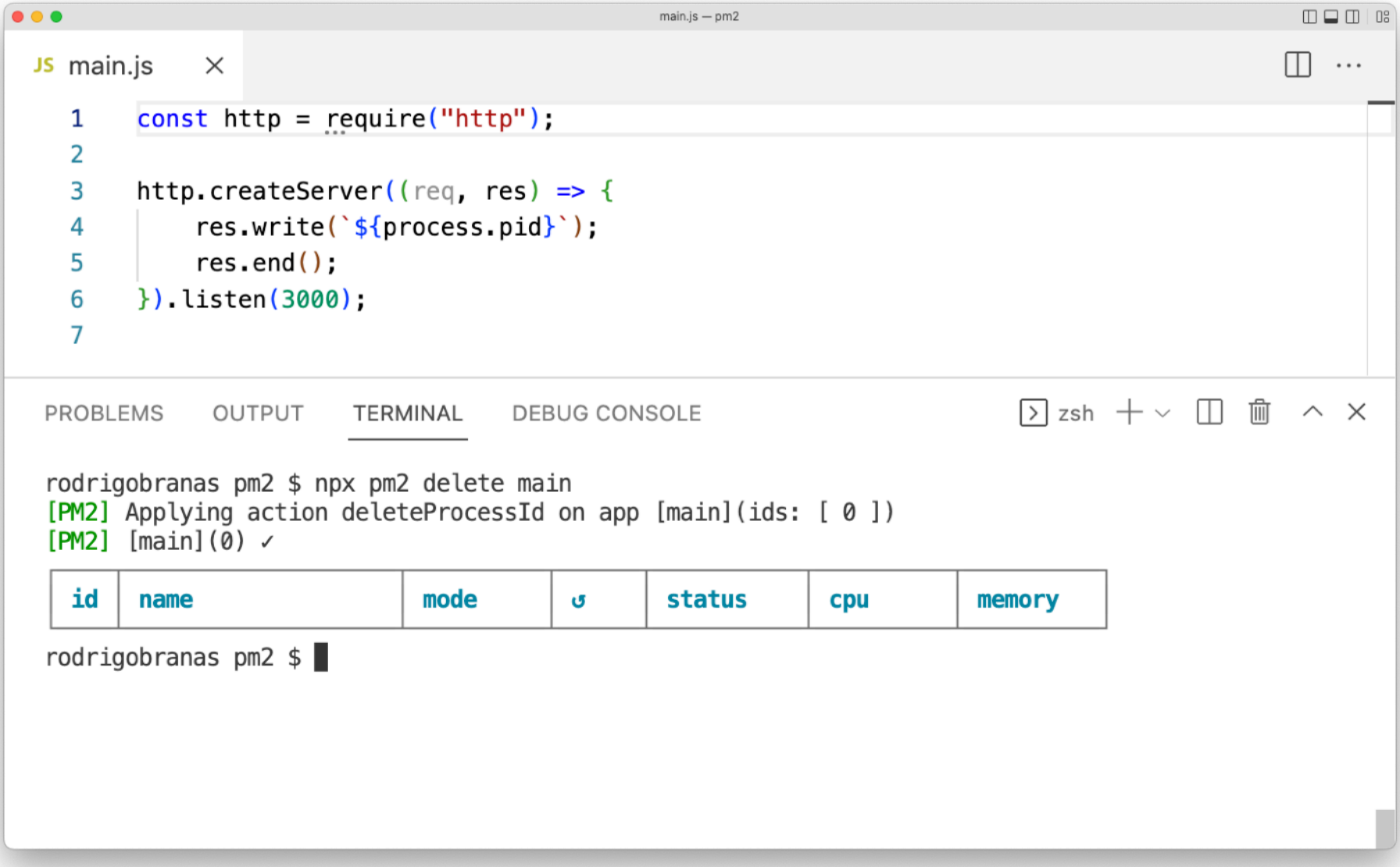
pm2 restart all

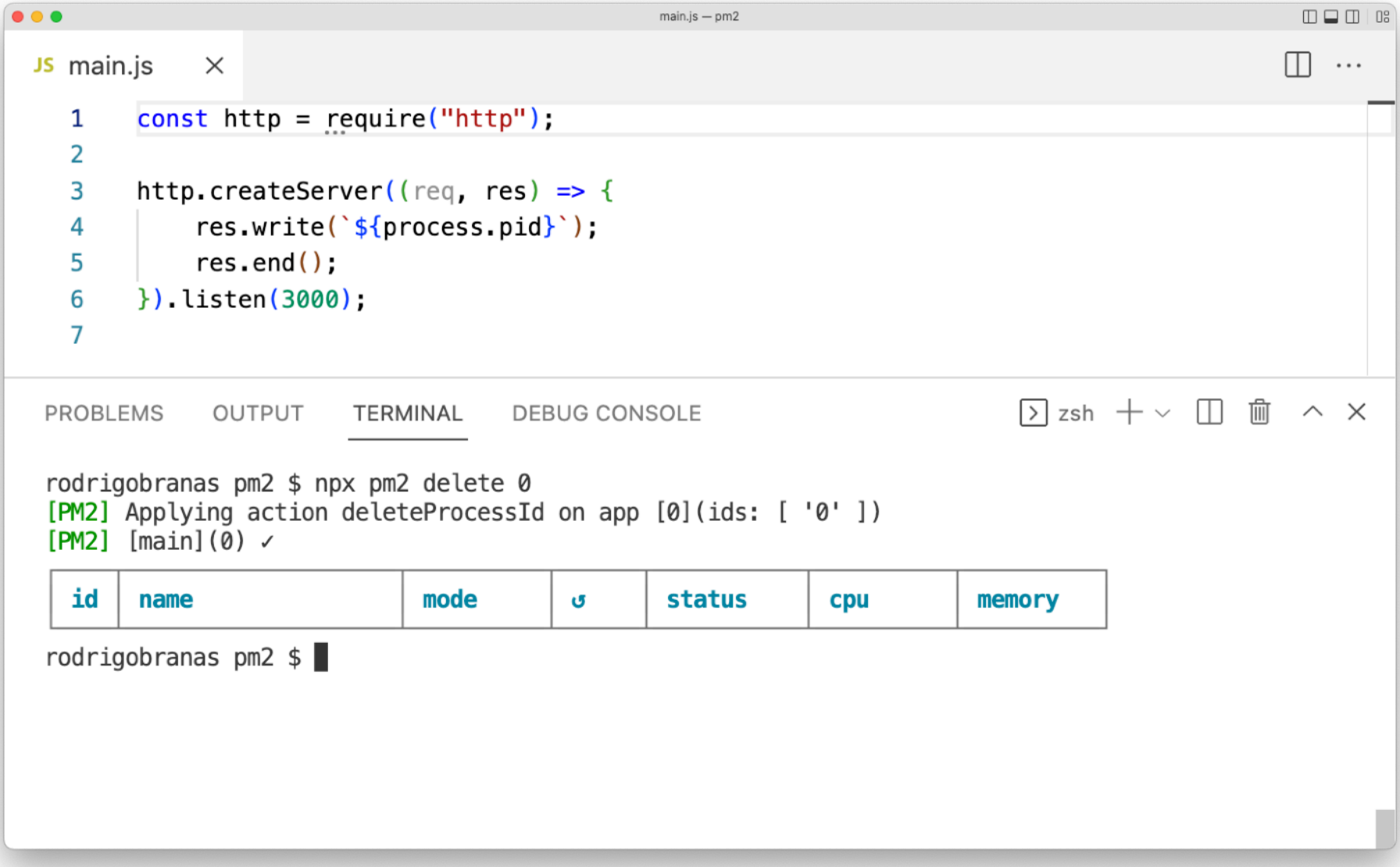
Reinicia todas as aplicações


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 restart all
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [all](ids: [0, 1])
[PM2] [main](0) ✓
[PM2] [main](1) ✓
id name mode ↺ status cpu memory
0 main cluster 1 online 0% 48.3mb
1 main cluster 1 online 0% 27.1mb
rodrigobranas pm2 $

pm2 delete main

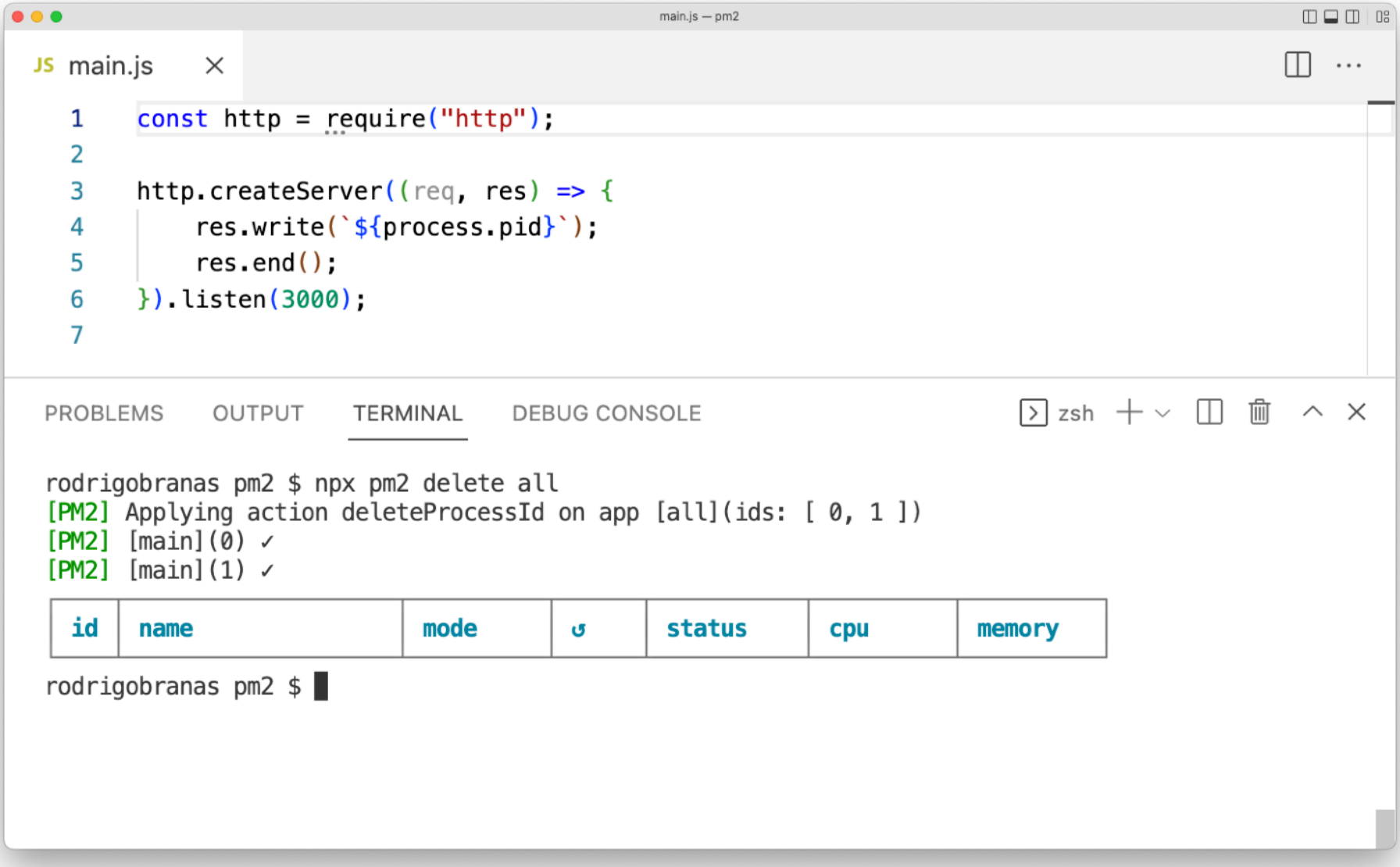
Exclui uma aplicação


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 delete main
[PM2] Applying action deleteProcessId on app [main](ids: [0])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
rodrigobranas pm2 $


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: ['0'])
[PM2] [main](0) ✓
id
name
mode
↺
status
cpu
memory
rodrigobranas pm2 $

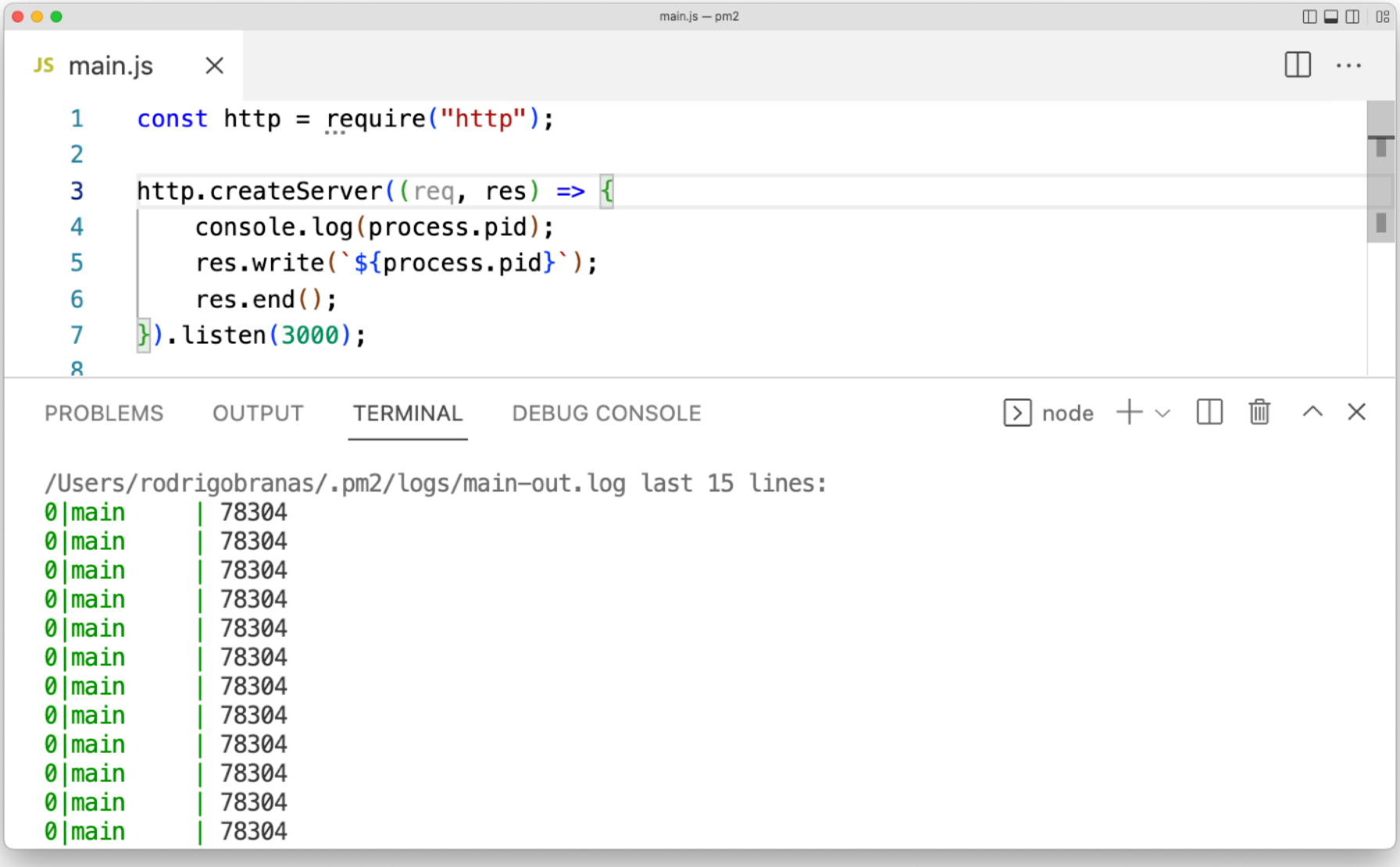
pm2 delete all

Exclui todas as aplicações


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 delete all
[PM2] Applying action deleteProcessId on app [all](ids: [0, 1])
[PM2] [main](0) ✓
[PM2] [main](1) ✓
id
name
mode
↺
status
cpu
memory
rodrigobranas pm2 $

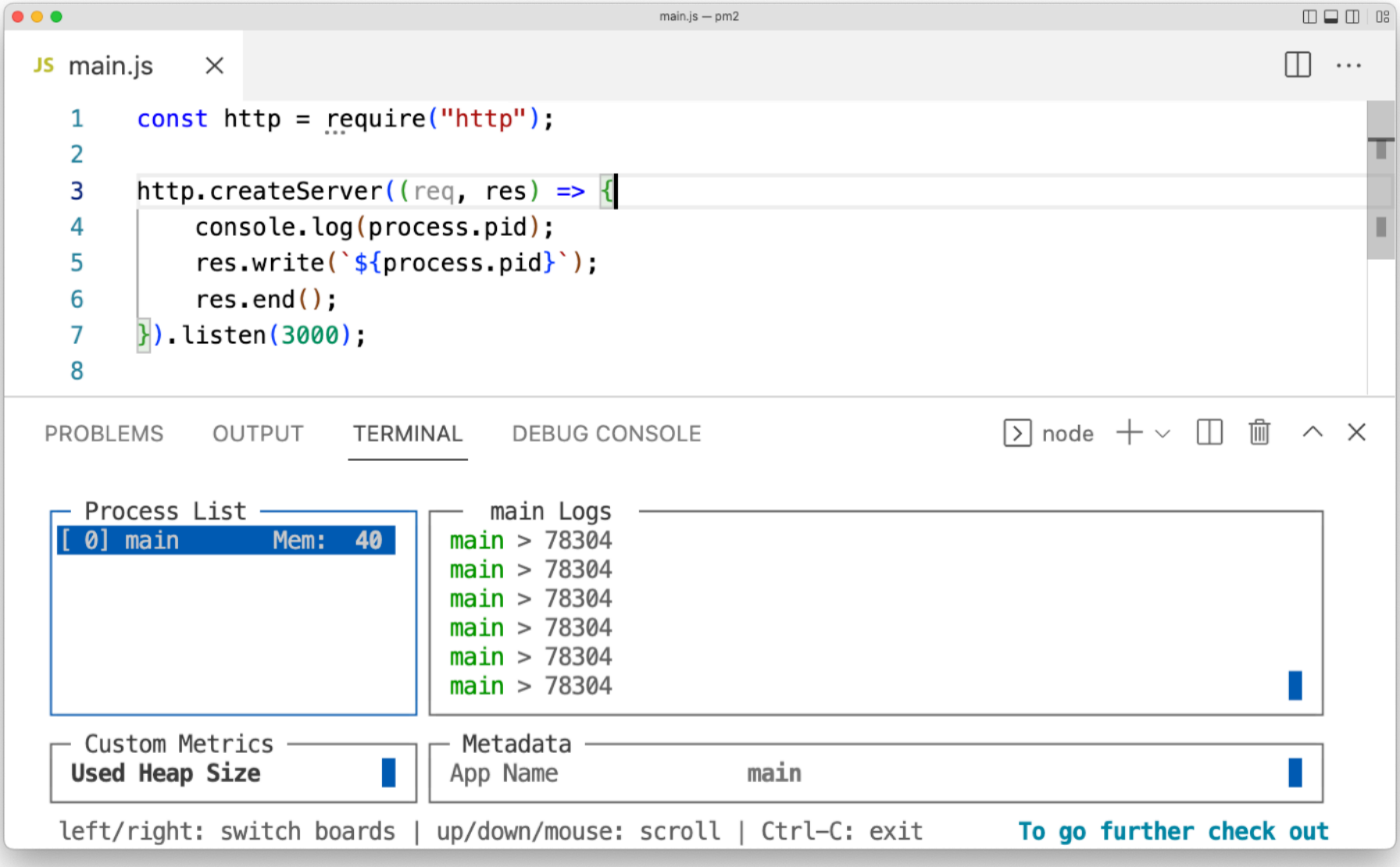
pm2 log

Monitora os logs da aplicação


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 console.log(process.pid);
5 res.write(`${process.pid}`);
6 res.end();
7 }).listen(3000);
8
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
node
/Users/rodrigobranas/.pm2/logs/main-out.log last 15 lines:
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304
0|main | 78304

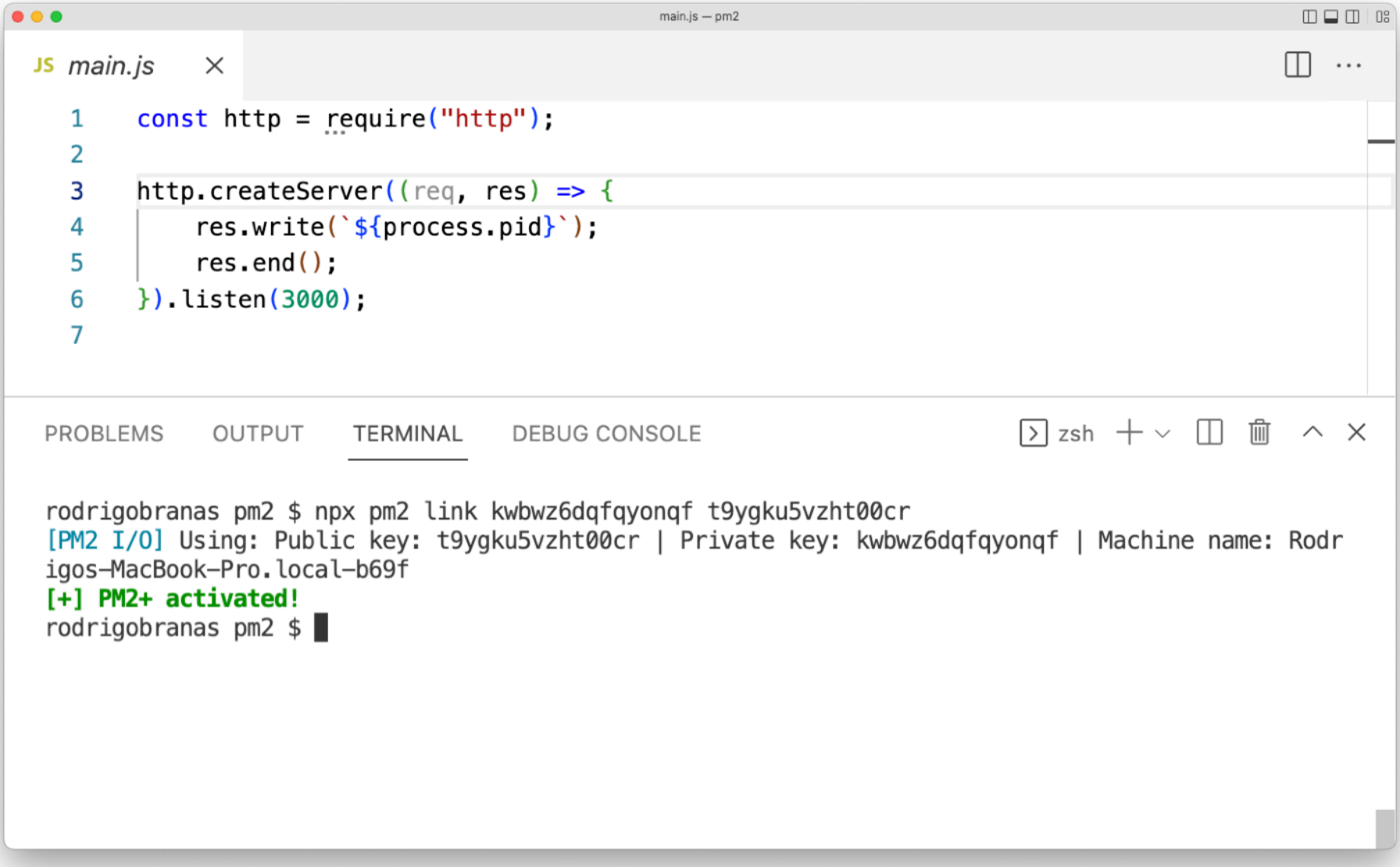
pm2 monit

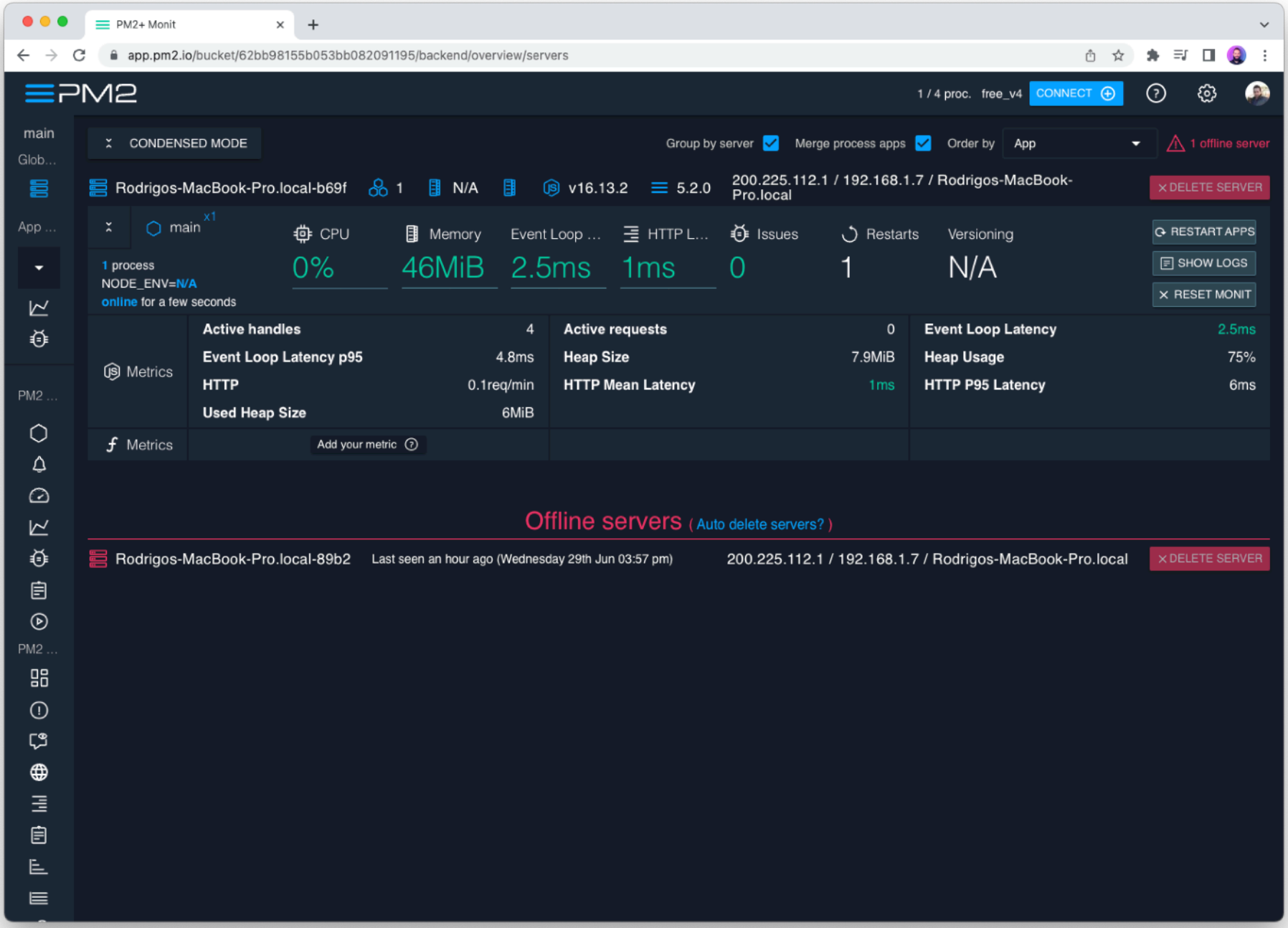
Monitora a aplicação

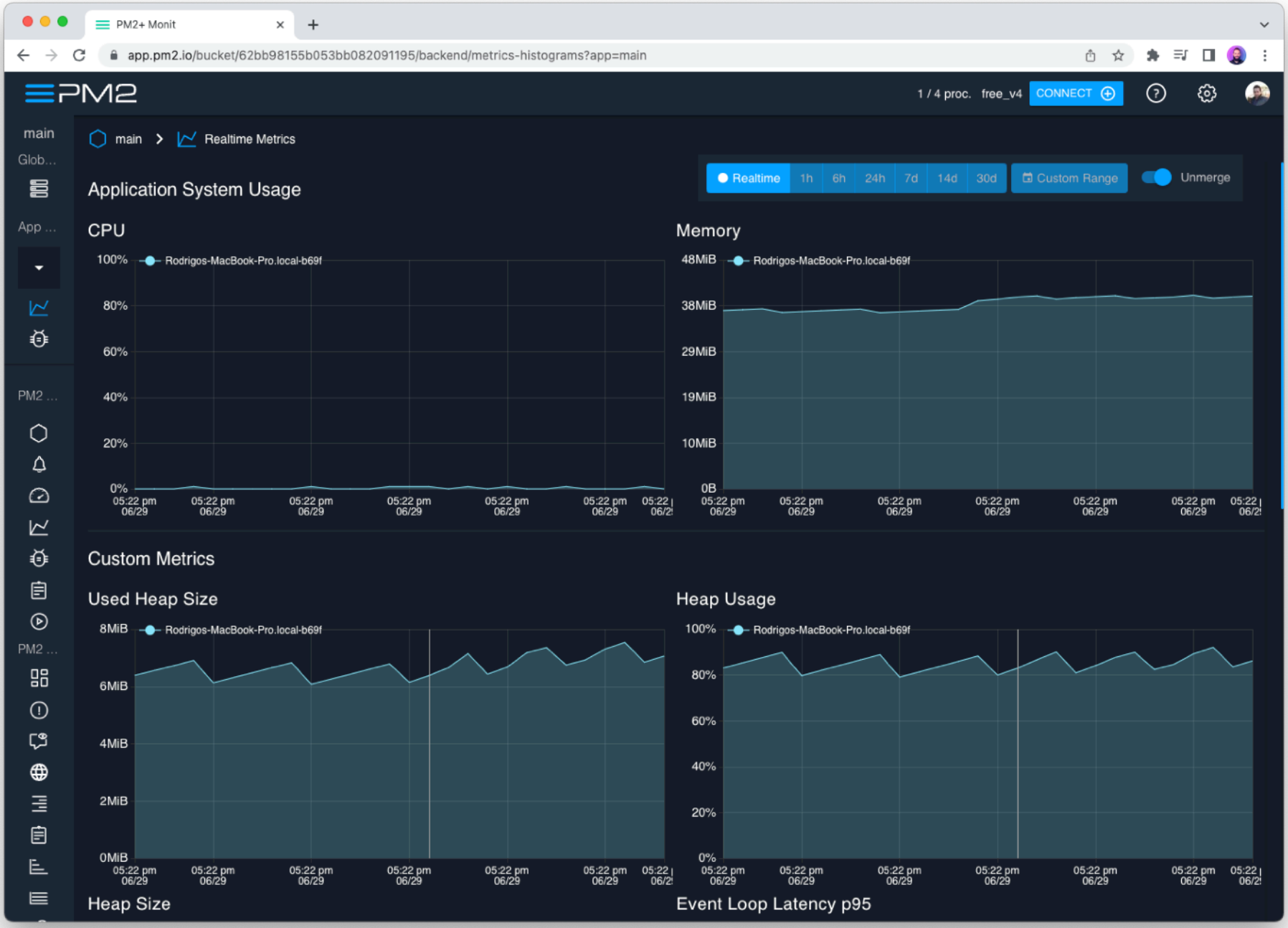

main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 console.log(process.pid);
5 res.write(`${process.pid}`);
6 res.end();
7 }).listen(3000);
8
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
node
Process List
[0] main Mem: 40
main Logs
main > 78304
main > 78304
main > 78304
main > 78304
main > 78304
main > 78304
Custom Metrics
Used Heap Size
Metadata
App Name main
left/right: switch boards | up/down/mouse: scroll | Ctrl-C: exit
To go further check out

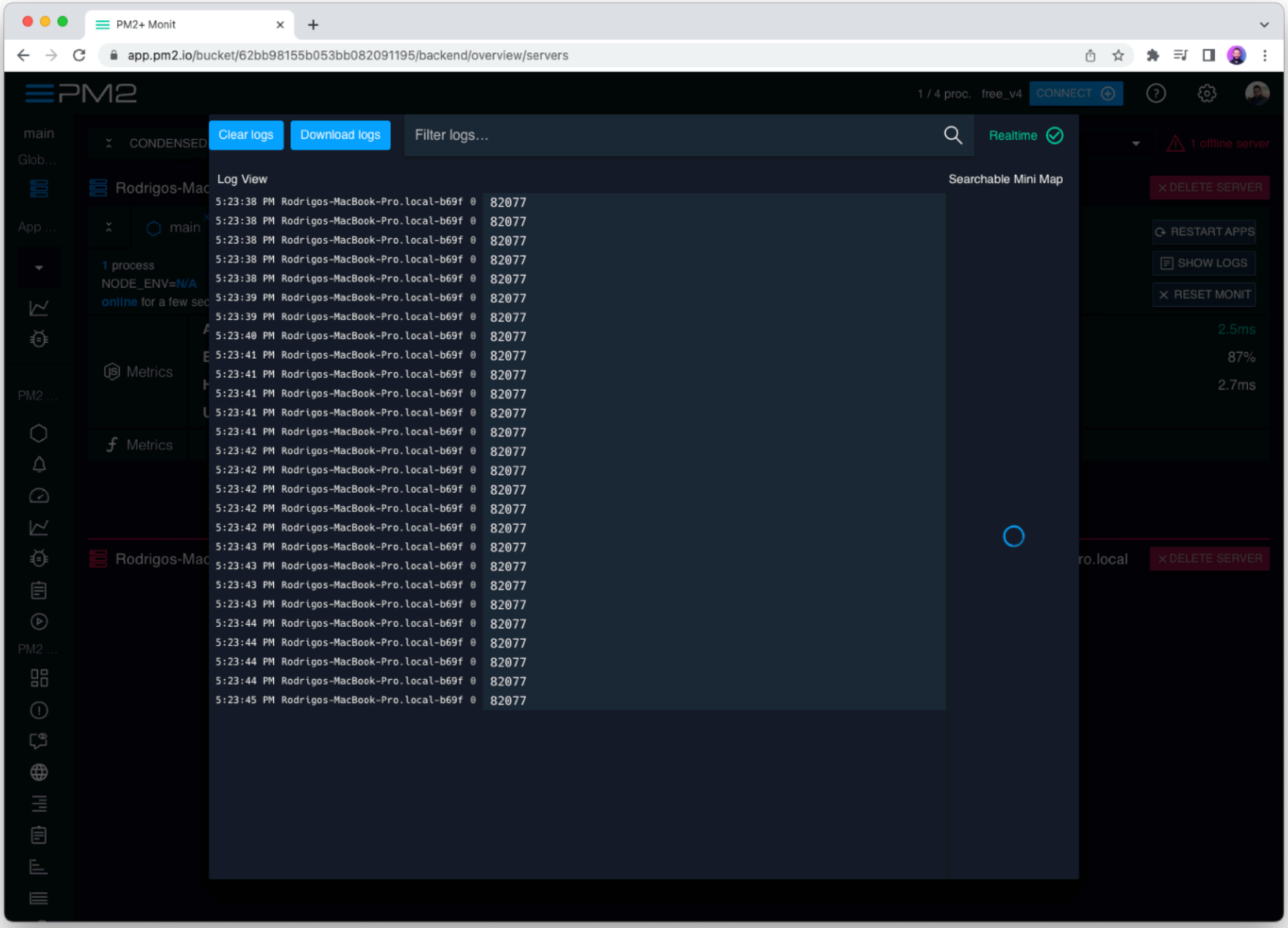
pm2 link

Conecta o monitoramento


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 res.write(`${process.pid}`);
5 res.end();
6 }).listen(3000);
7
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 link kwbwz6dqfqyonqf t9ygku5vzht00cr
[PM2 I/O] Using: Public key: t9ygku5vzht00cr | Private key: kwbwz6dqfqyonqf | Machine name: Rodr
igos-MacBook-Pro.local-b69f
[+] PM2+ activated!
rodrigobranas pm2 $

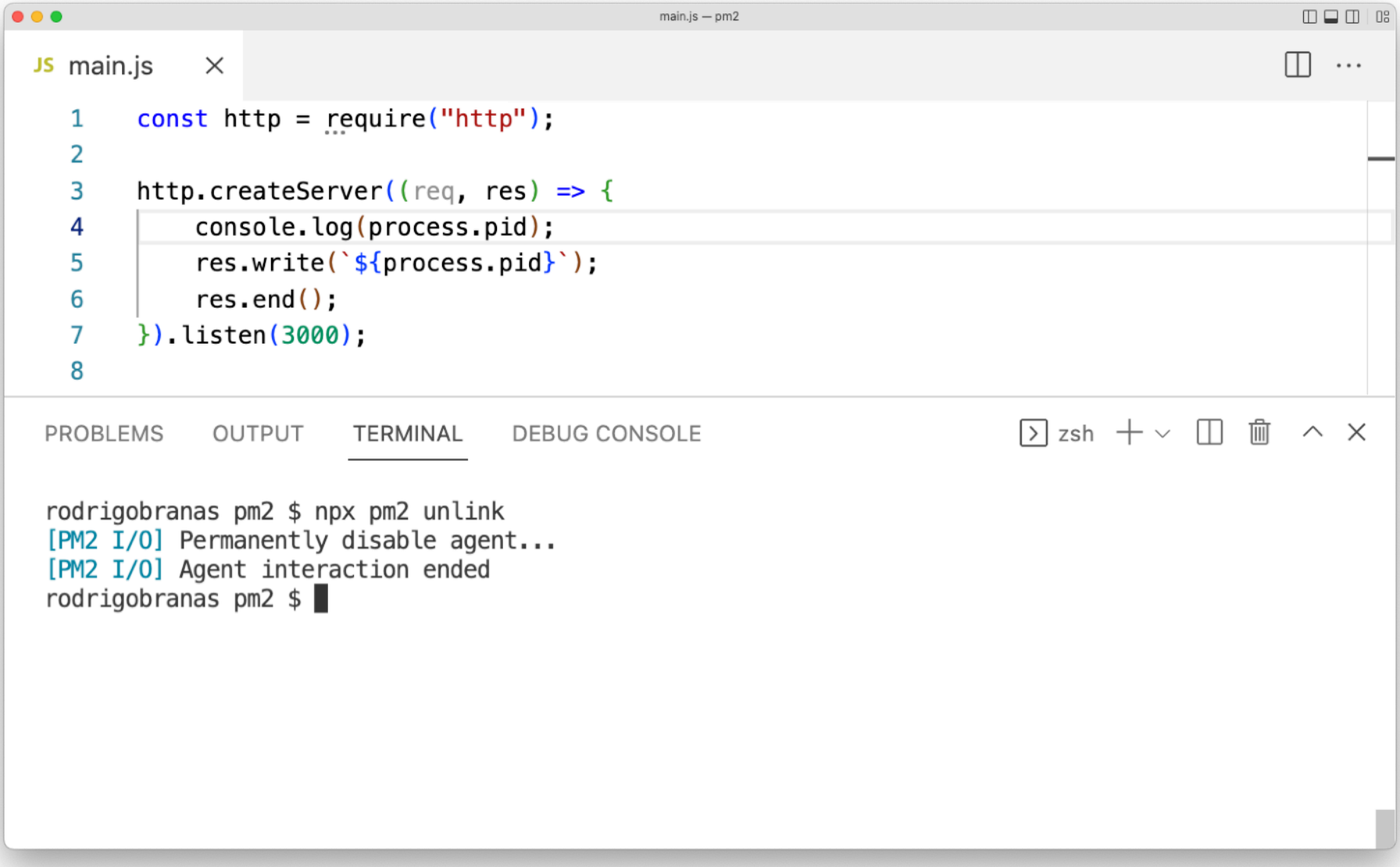

PM2+ Monit
app.pm2.io/bucket/62bb98155b053bb082091195/backend/overview/servers
PM2
1 / 4 proc. free_v4
CONNECT
main
Glob...
App ...
PM2 ...
PM2 ...
CONDENSED MODE
Group by server
Merge process apps
Order by
App
1 offline server
Rodrigos-MacBook-Pro.local-b69f
1
N/A
v16.13.2
5.2.0
200.225.112.1 / 192.168.1.7 / Rodrigos-MacBook-Pro.local
DELETE SERVER
main x1
1 process
NODE_ENV=N/A
online for a few seconds
CPU
Memory
Event Loop ...
HTTP L...
Issues
Restarts
Versioning
0%
46MiB
2.5ms
1ms
0
1
N/A
RESTART APPS
SHOW LOGS
RESET MONIT
Metrics
Active handles 4
Event Loop Latency p95 4.8ms
HTTP 0.1req/min
Used Heap Size 6MiB
Active requests 0
Heap Size 7.9MiB
HTTP Mean Latency 1ms
Event Loop Latency 2.5ms
Heap Usage 75%
HTTP P95 Latency 6ms
Metrics
Add your metric
Offline servers (Auto delete servers?)
Rodrigos-MacBook-Pro.local-89b2
Last seen an hour ago (Wednesday 29th Jun 03:57 pm)
200.225.112.1 / 192.168.1.7 / Rodrigos-MacBook-Pro.local
DELETE SERVER


PM2+ Monit
app.pm2.io/bucket/62bb98155b053bb082091195/backend/metrics-histograms?app=main
PM2
1 / 4 proc. free_v4
CONNECT
main
Realtime Metrics
Application System Usage
Realtime
1h
6h
24h
7d
14d
30d
Custom Range
Unmerge
CPU
Rodrigos-MacBook-Pro.local-b69f
100%
80%
60%
40%
20%
0%
Memory
48MiB
38MiB
29MiB
19MiB
10MiB
0B
05:22 pm
06/29
Custom Metrics
Used Heap Size
8MiB
6MiB
4MiB
2MiB
0MiB
Heap Usage
Heap Size
Event Loop Latency p95


PM2+ Monit
app.pm2.io/bucket/62bb98155b053bb082091195/backend/overview/servers
PM2
1 / 4 proc. free_v4
CONNECT
Clear logs
Download logs
Filter logs...
Realtime
Log View
Searchable Mini Map
5:23:38 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:38 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:38 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:38 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:38 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:39 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:39 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:40 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:41 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:41 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:41 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:41 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:41 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:42 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:42 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:42 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:42 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:42 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:43 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:43 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:43 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:43 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:44 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:44 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:44 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:44 PM Rodrigos-MacBook-Pro.local-b69f 0 82077
5:23:45 PM Rodrigos-MacBook-Pro.local-b69f 0 82077

pm2 unlink

Desconecta o monitoramento


main.js — pm2
main.js
1 const http = require("http");
2
3 http.createServer((req, res) => {
4 console.log(process.pid);
5 res.write(`${process.pid}`);
6 res.end();
7 }).listen(3000);
8
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh
rodrigobranas pm2 $ npx pm2 unlink
[PM2 I/O] Permanently disable agent...
[PM2 I/O] Agent interaction ended
rodrigobranas pm2 $