

PM2 - Process Management

Managing applications states

With PM2 you can easily start/restart/reload/stop/list applications in background.

Start

To start an application:

```
➦ > pm2 start http.js
[PM2] Starting /var/examples/http.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	http	default	N/A	fork	48553	0s	0	online	0%	17.7mb	unitech	disabled

You can also start any kind of application like bash commands, script, binaries:

```
$ pm2 start "npm run start"
$ pm2 start "ls -la"
$ pm2 start app.py
```

Start and display log stream

To start an app and check logs stream use the --attach option:

```
$ pm2 start api.js --attach
```

When quitting via Ctrl-C, the app will still run in background.

Passing arguments

All option passed after -- will be passed as argument to the app:

```
$ pm2 start api.js -- arg1 arg2
```

Configuration File

When managing multiple application at the same time or having to specify multiple options, you can use a configuration file. Example with this ecosystem.config.js file:

```
module.exports = {
  apps : [{
    name : "limit worker",
    script : "./worker.js",
    args : "limit"
  },{
    name : "rotate worker",
    script : "./worker.js",
    args : "rotate"
  }]
}
```

Then to start both apps:

```
$ pm2 start ecosystem.config.js
```

Read more about [configuration file](#).

Restart

To restart an application:

To restart all applications:

To restart multiple apps at once:

```
$ pm2 restart app1 app3 app4
```

Updating environment variables and options

To update environment variables or PM2 options, specify the --update-env CLI option:

```
$ NODE_ENV=production pm2 restart web-interface --update-env
```

Stop

To stop a specified application:

```
$ pm2 stop api
$ pm2 stop [process_id]
```

To stop them all:

To stop multiple apps at once:

```
$ pm2 stop app1 app3 app4
```

Note: this will not delete the application from PM2 application list. See next section to delete an application.

Delete

To stop and delete an application:

To delete them all:

Listing Applications

To list all running applications:

```
$ pm2 list
# 0r
$ pm2 [[list|ls|l|status]]
```

pm2 ls

id	name	namespace	version	mode	pid	uptime	⚡	status	cpu	mem	user	watching
2	autoexit	default	N/A	fork	0	0	0	stopped	0%	0b	unitech	disabled
0	http	default	N/A	fork	38914	26m	0	online	0%	42.9mb	unitech	disabled
3	watch ls	default	N/A	fork	41366	7m	0	online	0%	2.9mb	unitech	disabled

To specify which order you want the application to be listed:

```
$ pm2 list --sort name:desc
# 0r
$ pm2 list --sort [name|id|pid|memory|cpu|status|uptime][:asc|desc]
```

Terminal Dashboard

PM2 gives you a simple way to monitor the resource usage of your application. You can monitor memory and CPU easily and straight from your terminal with:

Process list

[0]	http	Mem: 43 MB	CPU: 66 %	online
[1]	http	Mem: 42 MB	CPU: 66 %	online
[2]	http	Mem: 42 MB	CPU: 60 %	online
[3]	http	Mem: 42 MB	CPU: 60 %	online
[4]	output	Mem: 35 MB	CPU: 0 %	online
[5]	output	Mem: 33 MB	CPU: 0 %	online

Global Logs

```
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
output > log message from echo.js
output > { json: true }
output > log message from echo.js
output > { json: true }
output > err msg from echo.js
output > err msg from echo.js
```

Custom metrics (http://bit.ly/code-metrics)

Loop delay 0.51ms

Metadata

App Name	http
Restarts	0
Uptime	12m
Script path	/home/unitech/keymetrics/pm2/examples/http.js
Script args	N/A
Interpreter	node
Interpreter args	N/A
Exec mode	cluster
Node.js version	7.4.0
watch & reload	X
Unstable restarts	0

Showing application metadata

To display metadata about an application:

pm2 show 0

Describing process with id 0 - name http

status

name

namespace

version

restarts

uptime

script path

script args

error log path

out log path

pid path

interpreter

interpreter args

script id

exec cwd

exec mode

node.js version

node env

watch & reload

unstable restarts

created at

online
http
default
N/A
0
1s
/var/examples/http.js
N/A
/home/unitech/.pm2/logs/http-error.log
/home/unitech/.pm2/logs/http-out.log
/home/unitech/.pm2/pids/http-0.pid
node
N/A
0
/var/examples
fork_mode
16.2.0
development
x
0
2021-06-26T10:49:23.764Z

Actions available

km:heapdump

km:cpu:profiling:start

km:cpu:profiling:stop

km:heap:sampling:start

km:heap:sampling:stop

Trigger via: pm2 trigger http <action_name>

Code metrics value

Heap Size

Heap Usage

Used Heap Size

10.14 MiB

64.48 %

6.54 MiB

Reset restart count

To reset the restart counter:

[Contribute to this page](#)