

QUICK START

Welcome!

Welcome to the PM2 Quick Start!

PM2 is a daemon process manager that will help you manage and keep your application online. Getting started with PM2 is straightforward, it is offered as a simple and intuitive CLI, installable via NPM.

Installation

The latest PM2 version is installable with NPM or Yarn:

```
$ npm install pm2@latest -g
# or
$ yarn global add pm2
```

To install Node.js and NPM you can use **NVM**

Start an app

The simplest way to start, daemonize and monitor your application is by using this command line:

```
$ pm2 start app.js
```

Or start any other application easily:

```
$ pm2 start bashscript.sh
$ pm2 start python-app.py --watch
$ pm2 start binary-file -- --port 1520
```

Some options you can pass to the CLI:

```
# Specify an app name
--name <app_name>

# Watch and Restart app when files change
--watch

# Set memory threshold for app reload
--max-memory-restart <200MB>

# Specify log file
--log <log_path>

# Pass extra arguments to the script
-- arg1 arg2 arg3

# Delay between automatic restarts
--restart-delay <delay in ms>

# Prefix logs with time
--time

# Do not auto restart app
--no-autorestart

# Specify cron for forced restart
--cron <cron_pattern>

# Attach to application log
--no-daemon
```

As you can see many options are available to manage your application with PM2. You will discover them depending on your use case.

Managing processes

Managing application state is simple here are the commands:

```
$ pm2 restart app_name
$ pm2 reload app_name
$ pm2 stop app_name
$ pm2 delete app_name
```

Instead of `app_name` you can pass:

- `all` to act on all processes
- `id` to act on a specific process id

Check status, logs, metrics

Now that you have started this application, you can check its status, logs, metrics and even get the online dashboard with `pm2.io`.

List managed applications

List the status of all application managed by PM2:

```
$ pm2 [list|ls|status]
```



Display logs

To display logs in realtime:

```
$ pm2 logs
```

To dig in older logs:

```
$ pm2 logs --lines 200
```

Terminal Based Dashboard

Here is a realtime dashboard that fits directly into your terminal:

```
$ pm2 monit
```



pm2.io: Monitoring & Diagnostic Web Interface

Web based dashboard, cross servers with diagnostic system:

```
$ pm2 plus
```



Cluster mode

For Node.js applications, PM2 includes an automatic load balancer that will share all HTTP[s]/Websocket/TCP/UDP connections between each spawned processes.

To start an application in Cluster mode:

```
$ pm2 start app.js -i max
```

Read more about cluster mode [here](#).

Ecosystem File

You can also create a configuration file, called Ecosystem File, to manage multiple applications. To generate an Ecosystem file:

```
$ pm2 ecosystem
```

This will generate an `ecosystem.config.js` file:

```
module.exports = {
  apps: [{
    name: "app",
    script: "./app.js",
    env: {
      NODE_ENV: "development",
    },
    env_production: {
      NODE_ENV: "production",
    }
  }, {
    name: 'worker',
    script: 'worker.js'
  }]
}
```

And start it easily:

```
$ pm2 start process.yml
```

Read more about application declaration [here](#).

Setup startup script

Restarting PM2 with the processes you manage on server boot/reboot is critical. To solve this, just run this command to generate an active startup script:

```
$ pm2 startup
```

And to freeze a process list for automatic respawn:

```
$ pm2 save
```

Read more about startup script generator [here](#).

Restart application on changes

It's pretty easy with the `--watch` option:

```
$ cd /path/to/my/app
$ pm2 start env.js --watch --ignore-watch="node_modules"
```

This will watch & restart the app on any file change from the current directory + all subfolders and it will ignore any changes in the `node_modules` folder `--ignore-watch="node_modules"`.

You can then use `pm2 logs` to check for restarted app logs.

Updating PM2

We made it simple, there is no breaking change between releases and the procedure is straightforward:

```
npm install pm2@latest -g
```

Then update the in-memory PM2:

```
pm2 update
```

CheatSheet

Here are some commands that are worth knowing. Just try them with a sample application or with your current web application on your development machine:

```
# Fork mode
pm2 start app.js --name my-api # Name process

# Cluster mode
pm2 start app.js -i 0          # Will start maximum processes with LB depending on available
pm2 start app.js -i max        # Same as above, but deprecated.
pm2 scale app +3               # Scales 'app' up by 3 workers
pm2 scale app 2                # Scales 'app' up or down to 2 workers total

# Listing

pm2 list                       # Display all processes status
pm2 jlist                      # Print process list in raw JSON
pm2 prettylist                 # Print process list in beautified JSON

pm2 describe 0                 # Display all informations about a specific process

pm2 monit                      # Monitor all processes

# Logs

pm2 logs [--raw]               # Display all processes logs in streaming
pm2 flush                      # Empty all log files
pm2 reloadLogs                 # Reload all logs

# Actions

pm2 stop all                   # Stop all processes
pm2 restart all                # Restart all processes

pm2 reload all                 # Will 0s downtime reload (for NETWORKED apps)

pm2 stop 0                     # Stop specific process id
pm2 restart 0                  # Restart specific process id

pm2 delete 0                   # Will remove process from pm2 list
pm2 delete all                 # Will remove all processes from pm2 list

# Misc

pm2 reset <process>            # Reset meta data (restarted time...)
pm2 updatePM2                  # Update in memory pm2
pm2 ping                       # Ensure pm2 daemon has been launched
pm2 sendSignal SIGUSR2 my-app # Send system signal to script
pm2 start app.js --no-daemon
pm2 start app.js --no-autorestart
```

What's next?

Learn how to declare all your application's behavior options into a [JSON configuration file](#).

Learn how to do [clean stop and restart](#) to increase reliability.

Learn how to [deploy and update production applications easily](#).

Monitor your production applications with [Keymetrics](#).

How to update PM2

Install the latest pm2 version:

```
npm install pm2@latest -g
```

Then update the in-memory PM2:

```
pm2 update
```



CONTRIBUTE TO THIS PAGE