



PM2 Basics

Starting Applications

<code>pm2 start app.js</code>	Start a Node.js application. Example: <code>pm2 start my_app.js</code>
<code>pm2 start app.js --name my-api</code>	Start an application with a specific name. Example: <code>pm2 start worker.js --name background-tasks</code>
<code>pm2 start app.js -i max</code>	Start an application in cluster mode, utilizing all available CPU cores. Example: <code>pm2 start api.js -i max</code>
<code>pm2 start app.js -i 4</code>	Start an application in cluster mode with a specified number of instances. Example: <code>pm2 start worker.js -i 2</code>
<code>pm2 start npm --start</code>	Start an application using npm. Example: <code>pm2 start npm -- run production</code>
<code>pm2 start app.js --watch</code>	Automatically restart the app when files change. Example: <code>pm2 start index.js --watch</code>
<code>pm2 start app.js --ignore-watch="node_modules", "test"</code>	Ignore specific paths when watching for file changes. Example: <code>pm2 start app.js --watch --ignore-watch="node_modules", "logs", "tmp"</code>
<code>pm2 start ecosystem.config.js</code>	Start applications defined in an ecosystem file. Example: <code>pm2 start ecosystem.config.js</code>

Managing Processes

<code>pm2 stop <app_name id></code>	Stop a specific process by name or ID. Example: <code>pm2 stop my-api</code> <code>pm2 stop 0</code>
<code>pm2 restart <app_name id></code>	Restart a specific process by name or ID. Example: <code>pm2 restart worker</code> <code>pm2 restart 1</code>
<code>pm2 delete <app_name id></code>	Delete a specific process by name or ID. Example: <code>pm2 delete my-api</code> <code>pm2 delete 0</code>
<code>pm2 stop all</code>	Stop all running processes. Example: <code>pm2 stop all</code>
<code>pm2 restart all</code>	Restart all running processes. Example: <code>pm2 restart all</code>
<code>pm2 delete all</code>	Delete all processes from PM2. Example: <code>pm2 delete all</code>
<code>pm2 reload all</code>	Reload all processes, minimizing downtime. Example: <code>pm2 reload all</code>

Monitoring and Logs

<code>pm2 list</code>	List all processes managed by PM2. Example: <code>pm2 list</code>
<code>pm2 show <app_name id></code>	Show detailed information about a specific process. Example: <code>pm2 show my-api</code> <code>pm2 show 0</code>
<code>pm2 monit</code>	Open a real-time process monitoring dashboard. Example: <code>pm2 monit</code>
<code>pm2 logs <app_name id></code>	Display logs for a specific process. Example: <code>pm2 logs my-api</code> <code>pm2 logs 0</code>
<code>pm2 logs</code>	Show aggregated logs for all processes. Example: <code>pm2 logs</code>
<code>pm2 flush</code>	Clear all PM2 log files. Example: <code>pm2 flush</code>
<code>pm2 reloadLogs</code>	Reload all PM2 logs. Example: <code>pm2 reloadLogs</code>

Ecosystem Configuration

Ecosystem File Structure

An `ecosystem.config.js` or `ecosystem.config.json` file defines application configurations for PM2.

Example `ecosystem.config.js` :

```
module.exports = {
  apps : [{
    name : 'my-api',
    script : 'api.js',
    instances : 'max',
    autorestart : true,
    watch : false,
    max_memory_restart : '1G',
    env : {
      NODE_ENV : 'development'
    },
    env_production : {
      NODE_ENV : 'production'
    }
  ]
};
```

Common Ecosystem Options

<code>name</code>	Name of the application. Example: <code>name: 'my-app'</code>
<code>script</code>	Path to the application script. Example: <code>script: 'index.js'</code>
<code>instances</code>	Number of application instances. Use <code>max</code> for maximum available cores. Example: <code>instances: 'max'</code>
<code>autorestart</code>	Automatically restart the application if it crashes. Example: <code>autorestart: true</code>
<code>watch</code>	Enable file watching for automatic restarts on code changes. Example: <code>watch: true</code>
<code>max_memory_restart</code>	Maximum memory usage before restarting the application. Example: <code>max_memory_restart: '1G'</code>
<code>env</code>	Environment variables for all environments. Example: <code>env: { NODE_ENV: 'development' }</code>
<code>env_<environment></code>	Environment-specific variables (e.g., <code>env_production</code>). Example: <code>env_production: { NODE_ENV: 'production', PORT: 80 }</code>

Deploying with Ecosystem

<code>pm2 deploy ecosystem.config.js production setup</code>	Setup the production environment. Example: <code>pm2 deploy ecosystem.config.js production setup</code>
<code>pm2 deploy ecosystem.config.js production update</code>	Update the production environment with new code. Example: <code>pm2 deploy ecosystem.config.js production update</code>
<code>pm2 deploy ecosystem.config.js production</code>	Deploy the application to the production environment. Example: <code>pm2 deploy ecosystem.config.js production</code>
<code>pm2 deploy ecosystem.config.js production revert <n></code>	Revert to a previous deployment (n = number of revisions back). Example: <code>pm2 deploy ecosystem.config.js production revert 1</code>
<code>pm2 ecosystem</code>	Generate a sample <code>ecosystem.config.js</code> file. Example: <code>pm2 ecosystem</code>

Advanced PM2 Usage

Startup Management

<code>pm2 startup</code>	Generate a startup script to automatically start PM2 and managed processes on system boot. Example: <code>pm2 startup</code>
<code>pm2 save</code>	Save the current process list to be restored on system boot. Example: <code>pm2 save</code>
<code>pm2 unstartup</code>	Disable PM2 from starting on boot (remove the startup script). Example: <code>pm2 unstartup</code>
<code>pm2 resurrect</code>	Restore previously saved processes. Example: <code>pm2 resurrect</code>

Process Interaction

<code>pm2 sendSignal <signal> <app_name id></code>	Send a specific signal to a process. Example: <code>pm2 sendSignal SIGUSR2 my-app</code>
<code>pm2 trigger <event_name> <app_name id></code>	Trigger a custom event in a PM2 managed process, if the application is set up to listen for the event. Example: <code>pm2 trigger reload my-app</code>
<code>pm2 ping</code>	Check if the PM2 daemon is running. Example: <code>pm2 ping</code>
<code>pm2 pull</code>	Pull new updates. Example: <code>pm2 pull</code>

Module Management

<code>pm2 install <module_name></code>	Install a PM2 module. Example: <code>pm2 install pm2-logrotate</code>
<code>pm2 uninstall <module_name></code>	Uninstall a PM2 module. Example: <code>pm2 uninstall pm2-logrotate</code>
<code>pm2 list</code>	Will also display modules list. Example: <code>pm2 list</code>

Troubleshooting and Tips

Common Issues

If an application fails to start, check the PM2 logs for error messages using `pm2 logs <app_name|id>`. Ensure that all dependencies are installed (`npm install`) and that the application code is free of errors.

If the application is not automatically restarting after a crash, check the `autorestart` setting in the ecosystem file or command line arguments. Also, ensure that the `max_memory_restart` setting is appropriate for your application's memory usage.

Useful Tips

Use an ecosystem file to manage application configurations. It provides a centralized way to define application settings, environment variables, and deployment options.

Utilize PM2 modules for extended functionality. For example, `pm2-logrotate` can be used to automatically rotate log files and prevent them from growing too large.

Monitor application performance using `pm2 monit` to identify and address issues such as high CPU usage or memory leaks.

Configure environment-specific variables in the ecosystem file to easily switch between development, staging, and production environments.

Updating PM2

`pm2 update` Update PM2 to the latest version, ensuring the daemon and all managed processes are updated.

Example:

```
pm2 update
```

`pm2 show` Check the current PM2 version.

Example:

```
pm2 show pm2
```