安装：npm install -g pm2

yarn global add pm2

启动程序：pm2 start <app_name|id|all>

列举进程：pm2 list

pm2 ls

退出程序：pm2 stop <app_name|id|all>

重起应用：pm2 restart

程序信息：pm2 describe id|all

监控：pm2 monit

实时集中log处理: pm2 logs

API:pm2 web (端口：9615)

1. 进入到刚才的Ghost安装目录 执行下面的命令安装PM2：

cd /vat/www/ghost/

sudo npm install -g

2. 2

我们要设置环境变量为“production”生产模式！“index.js”是程序启动的入口。最后给这个PM2的进程命名为"ghost" 于是，执行下面的命令：

NODE_ENV=production pm2 start index.js --name "ghost"

3. 3

让PM2知道在开机后自动运行我们的网站：

1.Ubuntu 系统：

pm2 startup ubuntu

2.CentOS 系统：

pm2 startup centos

3.亚马逊 EC2:

pm2 startup amazon

保存设置（非常重要）

pm2 save

END

如果您希望在另一个用户下执行启动挂钩，请使用以下-u <username>选项--hp <user_home>：

```
pm2 startup ubuntu -u www --hp /home/ubuntu
```

pm2可以通过HTTP提供静态文件（如前端应用程序）：

```
pm2 serve <path> <port>
```

默认值为current folder和8080，您可以使用：

```
pm2 serve
```

在生态系统文件中：

```
module.exports = {
  apps: [{
    name: "static-file",
    script: "serve",
    env: {
      PM2_SERVE_PATH: ".",
      PM2_SERVE_PORT: 8080,
    },
  }]
}
```

PM2的常用命令

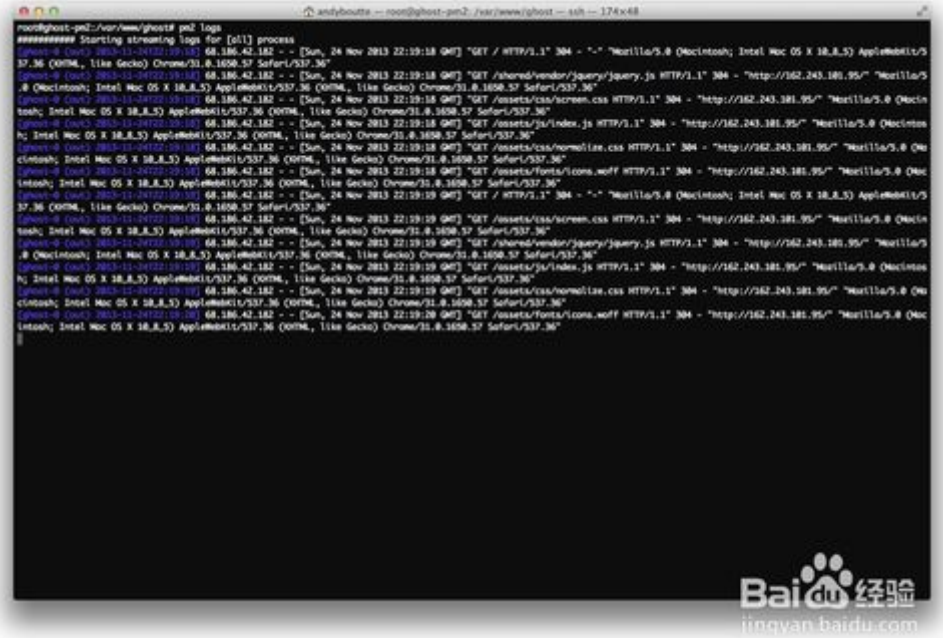
1. 监控运行状态：

pm2 status

Ghost博客指南:用PM2守护Node.js进程

2. 监控Node.js程序的运行日志

pm2 logs



Ghost博客指南:用PM2守护Node.js进程

3. 检查内存占用情况和程序的运行状态：

pm2 monit

4. 4

让pm2能够能够在这3个系统上自动启动。

pm2 startup <centos|ubuntu|amazon>

分别是启动|停止|重启 ghost程序:

pm2 <start|stop|restart> ghost

清除所有正在运行的PM2 Ghost进程:

pm2 kill ghost

主要特性:

内建负载均衡（使用Node cluster 集群模块）
后台运行
0秒停机重载，我理解大概意思是维护升级的时候不需要停机.
具有Ubuntu和CentOS 的启动脚本
停止不稳定的进程（避免无限循环）
控制台检测
提供 HTTP API
远程控制和实时的接口API (Nodejs 模块,允许和PM2进程管理器交互)

测试过Nodejs v0.11 v0.10 v0.8版本，兼容CoffeeScript,基于Linux 和MacOS.

安装

```
npm install -g pm2
```

用法

```
$ npm install pm2 -g # 命令行安装 pm2
$ pm2 start app.js -i 4 #后台运行pm2，启动4个app.js
                                                    # 也可以把'max'
参数传递给 start
                                                    # 正确的进程数目
依赖于Cpu的核心数目
$ pm2 start app.js --name my-api # 命名进程
```

$ pm2 list # 显示所有进程状态
$ pm2 monit # 监视所有进程
$ pm2 logs # 显示所有进程日志
$ pm2 stop all # 停止所有进程
$ pm2 restart all # 重启所有进程
$ pm2 reload all # 0秒停机重载进程 (用于 NETWORKED 进程)
$ pm2 stop 0 # 停止指定的进程
$ pm2 restart 0 # 重启指定的进程
$ pm2 startup # 产生 init 脚本 保持进程活着
$ pm2 web # 运行健壮的 computer API endpoint (http://localhost:9615)
$ pm2 delete 0 # 杀死指定的进程
$ pm2 delete all # 杀死全部进程

运行进程的不同方式：
$ pm2 start app.js -i max # 根据有效CPU数目启动最大进程数目
$ pm2 start app.js -i 3 # 启动3个进程
$ pm2 start app.js -x #用fork模式启动 app.js 而不是使用 cluster
$ pm2 start app.js -x -- -a 23 # 用fork模式启动 app.js 并且传递参数 (-a 23)
$ pm2 start app.js --name serverone # 启动一个进程并把它命名为 serverone
$ pm2 stop serverone # 停止 serverone 进程
$ pm2 start app.json # 启动进程, 在 app.json里设置选项
$ pm2 start app.js -i max -- -a 23 #在--之后给 app.js 传递参数
$ pm2 start app.js -i max -e err.log -o out.log # 启动 并 生成一个配置文件
你也可以执行用其他语言编写的app (fork 模式):
$ pm2 start my-bash-script.sh -x --interpreter bash
$ pm2 start my-python-script.py -x --interpreter python

0秒停机重载:
这项功能允许你重新载入代码而不用失去请求连接。
注意：
仅能用于web应用
运行于Node 0.11.x版本
运行于 cluster 模式（默认模式）
$ pm2 reload all

CoffeeScript:

$ pm2 start my_app.coffee #这就是全部

PM2准备好为产品级服务了吗?

只需在你的服务器上测试

$ git clone https://github.com/Unitech/pm2.git

$ cd pm2

$ npm install # 或者 npm install --dev ，如果devDependencies 没有安装

$ npm test

pm2 list

列出由pm2管理的所有进程信息，还会显示一个进程会被启动多少次，因为没处理的异常。

pm2 monit

监视每个node进程的CPU和内存的使用情况。