调试

node-inspector: https://github.com/dannycoates/node-inspector。

===============================

下一章将带领大家开发一个让Node.js支持多线程并基于libuv的Node.js包。

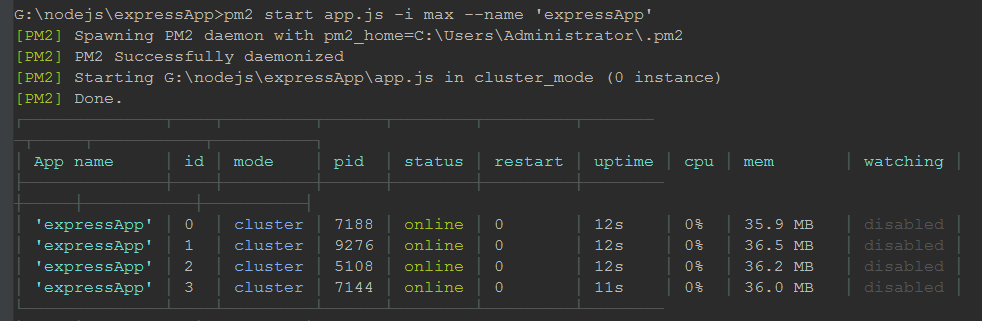
<http://jingyan.baidu.com/article/d8072ac47baf0eec95cefdca.html>

ab 进行压力测试

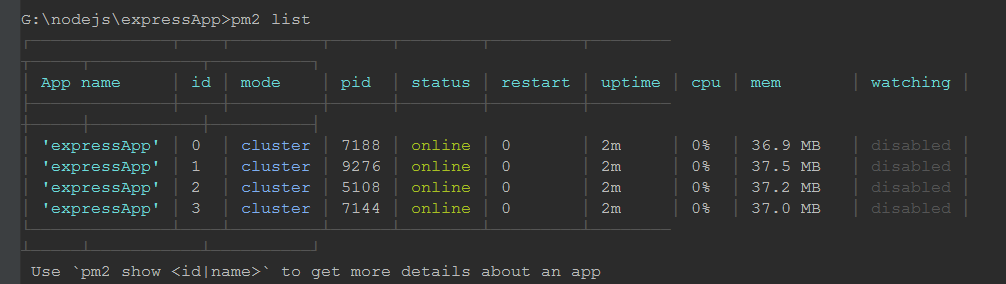
/===================

pm2 管理nodejs

pm2 start app.js -i max --name 'expressApp'



pm2 list



pm2 logs

在另一个终端,运行监控选项:

$ pm2 monit

一个package.json文件通常有以下字段：

- name —— 包的名称，必须是唯一的，由小写英文字母、数字和下划线组成，不能包含空格

- description —— 包的简要说明

- version —— 包的版本

- author —— 包的作者

- contributors —— 贡献者数组，数组每一项为一个包含一个贡献者资料的对象

- dependencies —— 包的依赖，为一个对象，对象的属性为包名称，属性值为版本号

- devDependencies —— 开发环境下的包依赖，为一个对象，对象的属性为包名称，属性值为版本号

- keywords —— 关键字数组，通常用于搜索

- repository —— 仓库托管地址，通常为一个包含type（仓库的类型，如：git）和 url（仓库的地址）的对象

- main —— 包的入口文件，如不指定，则默认为根目录下的index.js或index.node

- bin —— 可执行文件的路径

- bugs —— 提交bug的地址

- maintainers —— 维护者数组，数组每一项为一个包含一个维护者资料的对象

- licenses —— 许可证数组，数组每一项为一个包含type（许可证的名称）和url（链接到许可证文本的地址）的对象

PM2

pm2 start helloworld.js --name 'helloworld'

# 执行部署

pm2 deploy ecosystem.json production

# 更新部署

pm2 deploy production update

出自:[通过 pm2 部署 node.js 项目](http://yijiebuyi.com/blog/77db2532be0afae2959cd918331d9dda.html)

## 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。