nodejs 应用服务器部署和使用

Name： 曲中岭

Email:zlingqu@126.com

Q Q :441869115

第一章 部署准备

1.1 目的

搭建 nodejs 服务器，直接运行 js 代码。

1.2 规划

OS ：Ubuntu14 x64

IP ：172.16.7.46，nginx 服务器

172.16.7.46,pm2 服务器

第二章 pm2 环境搭建

2.1 添加源

使用如下命令添加 8 版本的源:

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash –
```

```
^Croot@jenkins:~# curl -sL http://deb.nodesource.com/setup_8.x | sudo -E bash -

## Installing the NodeSource Node.js 8.x LTS Carbon repo...


## Populating apt-get cache...

+ apt-get update
Ign http://hk.archive.ubuntu.com trusty InRelease
Hit http://hk.archive.ubuntu.com trusty-updates InRelease
Hit http://hk.archive.ubuntu.com trusty-backports InRelease
Hit http://hk.archive.ubuntu.com trusty Release.gpg
```

其中 setup_8.x 的 8 表示大版本，当前 10 是最新的，当然也可以换成 9、10 之类的，不过最新版 10 不支持 ubuntu14 了。

安装完成提示如下:

```
Hit http://security.ubuntu.com trusty-security/restricted Translation-en
Hit http://security.ubuntu.com trusty-security/universe Translation-en
Fetched 7,415 B in 7s (979 B/s)
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 8.x LTS Carbon and npm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
     echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
     sudo apt-get update && sudo apt-get install yarn
```

2.2 安装 nodejs 和 npm

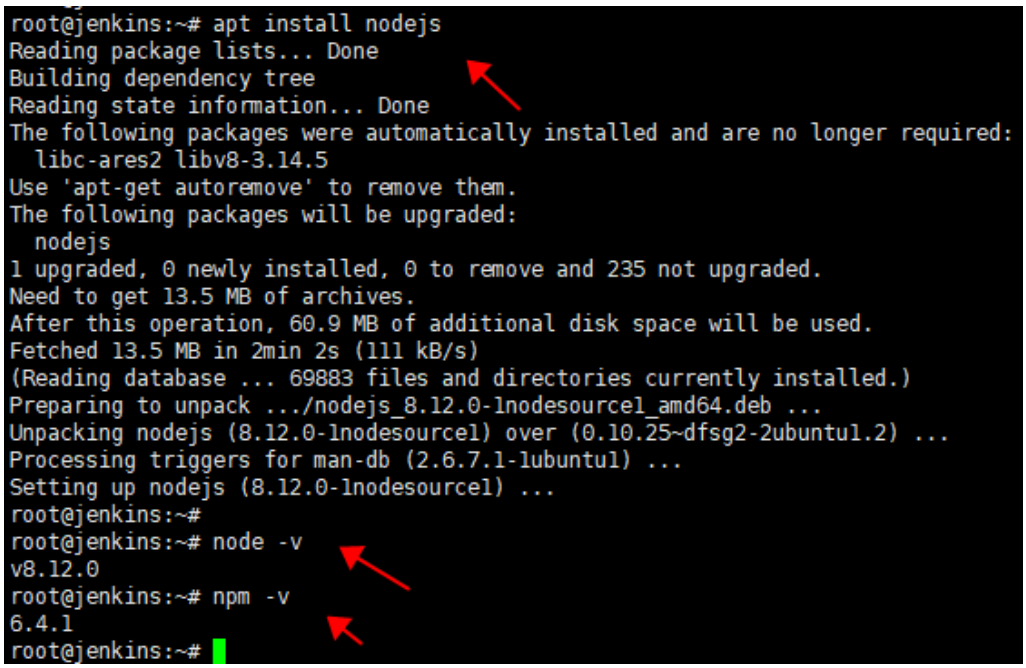
使用 apt 直接安装 nodejs

```
apt install nodejs
```

安装后，使用以下命令查看

```
node –v
npm –v
```

如下图所示:



如果使用 CentOS，安装方式如下:

curl --silent --location https://rpm.nodesource.com/setup_10.x | bash -

#10 表示大版本，可换乘 8、9 等

yum install -y nodejs

2.3 替换源（可选）

默认使用官方的源进行下载依赖等，可能比较慢。

查看当前的源:

```
npm get registry
```

切换成淘宝源:

```
npm config set registry http://registry.npm.taobao.org/
```

切换回官方源:

```
npm config set registry https://registry.npmjs.org/
```

2.4 安装 pm2 模块

2.4.1 安装

使用如下命令安装 pm2 模块:

```
npm install pm2 -g
```

确认是否安装成功:

```
npm list –g|grep pm2
```

```
pm2 -v
```

```
[root@iZuf61fwacpba8u3ty82udZ alibaba]# npm list -g|grep pm2
└─┬ pm2@3.2.4
  ├─┬ @pm2/agent@0.5.20
  │ ├── pm2-axon@3.3.0 deduped
  │ └── pm2-axon-rpc@0.5.1 deduped
  ├─┬ @pm2/io@2.4.7
  │ └─┬ @pm2/agent-node@1.1.3
  ├─┬ @pm2/js-api@0.5.41
  ├─┬ pm2-axon@3.3.0
  ├─┬ pm2-axon-rpc@0.5.1
  ├─┬ pm2-deploy@0.3.10
  └─┬ pm2-multimeter@0.1.2
[root@iZuf61fwacpba8u3ty82udZ alibaba]#
[root@iZuf61fwacpba8u3ty82udZ alibaba]#
[root@iZuf61fwacpba8u3ty82udZ alibaba]# pm2 -v
3.2.4
```

2.4.2 安装为系统服务

使用如下命令，即可自动配置为系统服务

```
pm2 startup
```

```
root@demo01:~# pm2 startup
[PM2] Init System found: upstart
Platform upstart
Template
#!/bin/bash
```

配置完成后如下图:

```
 Adding system startup for /etc/init.d/pm2-root ...
   /etc/rc0.d/K20pm2-root -> ../init.d/pm2-root
   /etc/rc1.d/K20pm2-root -> ../init.d/pm2-root
   /etc/rc6.d/K20pm2-root -> ../init.d/pm2-root
   /etc/rc2.d/S20pm2-root -> ../init.d/pm2-root
   /etc/rc3.d/S20pm2-root -> ../init.d/pm2-root
   /etc/rc4.d/S20pm2-root -> ../init.d/pm2-root
   /etc/rc5.d/S20pm2-root -> ../init.d/pm2-root
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup upstart
```

安装为系统服务后，可使用如下命令进行管理，也可加为开机自动启动

```
service pm2-root status
service pm2-root stop
service pm2-root start
service pm2-root restart
```

如果是 Centos 7 等使用 systemctl 进行管理的服务，与此类似，如下图查看状态:

```
[root@iZuf61fwacpba8u3ty82udZ lx2]# service pm2-root status
Redirecting to /bin/systemctl status pm2-root.service
● pm2-root.service - PM2 process manager
   Loaded: loaded (/etc/systemd/system/pm2-root.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://pm2.keymetrics.io/
[root@iZuf61fwacpba8u3ty82udZ lx2]#
[root@iZuf61fwacpba8u3ty82udZ lx2]# service pm2-root start
Redirecting to /bin/systemctl start pm2-root.service
[root@iZuf61fwacpba8u3ty82udZ lx2]#
[root@iZuf61fwacpba8u3ty82udZ lx2]# service pm2-root status
Redirecting to /bin/systemctl status pm2-root.service
● pm2-root.service - PM2 process manager
   Loaded: loaded (/etc/systemd/system/pm2-root.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-12-24 14:37:57 CST; 2s ago
     Docs: https://pm2.keymetrics.io/
  Process: 11773 ExecStart=/usr/lib/node_modules/pm2/bin/pm2 resurrect (code=exited, status=0/SUCCESS)
 Main PID: 11596 (PM2 v3.2.4: God)
   CGroup: /system.slice/pm2-root.service
           ‣ 11596 PM2 v3.2.4: God Daemon (/root/.pm2)
```

2.5 其他

如果 npm 不是最新版，使用如下命令升级

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

```
    install
root@testserver01:~# npm install npm@latest -g
/usr/bin/npm -> /usr/lib/node_modules/npm/bin/npm-cli.js
/usr/bin/npx -> /usr/lib/node_modules/npm/bin/npx-cli.js
+ npm@6.4.1
updated 1 package in 22.125s
root@testserver01:~#
root@testserver01:~#
root@testserver01:~#
root@testserver01:~# npm -v
6.4.1
root@testserver01:~#
```

我这里初次安装，默认已经是最新版。

更多 npm 命令参考: https://blog.csdn.net/u014291497/article/details/75193865

第三章 nodejs 服务器运行

3.1 测试-使用 noce 运行 js

vim a.js 输入以下内容:

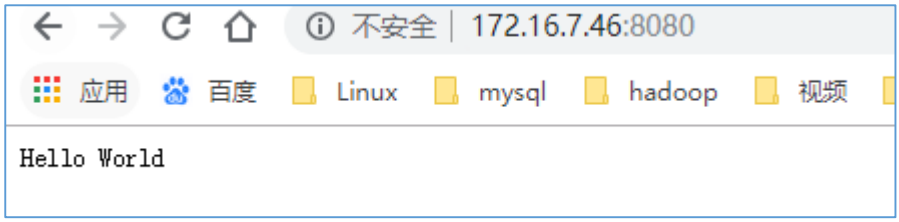
```
var http = require('http');
http.createServer(function (req, res) {
// 发送 HTTP 头部
// HTTP 状态值: 200 : OK
// 内容类型: text/plain

  res.writeHead(200, {'Content-Type': 'text/plain'});
// 发送响应数据 "Hello World"

  res.end('Hello World\n');
}).listen(8080, '172.16.7.46');
// 终端打印如下信息

console.log('Server running at http://172.16.7.46:8080/');
```

输入以下命令运行

```
node a.js
```

```
root@demo01:~# node a.js
Server running at http://172.16.7.46:8080/
```

使用浏览器或则 curl 测试




```
root@jenkins:~# curl 172.16.7.46:8080
Hello World
```

3.2 测试-使用 pm2 运行 js

使用如下命令即可运行上述的 a.js

```
pm2 stat a.js
```

```
root@demo01:~# pm2 start a.js
[PM2] Starting /root/a.js in fork_mode (1 instance)
[PM2] Done.
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
a	0	N/A	fork	13057	online	0	0s	0%	24.5 MB	root	disabled

```
 Use `pm2 show <id|name>` to get more details about an app
root@demo01:~#
```

查看运行状态以及监听的端口

```
root@demo01:~# service pm2-root status
Status for pm2:
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watchi
a	0	N/A	fork	13057	online	0	15s	2.5%	33.2 MB	root	disabl

```
 Use `pm2 show <id|name>` to get more details about an app
root@demo01:~#
root@demo01:~#
root@demo01:~# netstat -tnlp|grep 8080
tcp        0      0 172.16.7.46:8080        0.0.0.0:*               LISTEN      13057/a.js
root@demo01:~#
```

App name 即是 js 的文件名前缀，如图，可使用如下命令管理该 app：

操作所有 app：

```
pm2 stop|reload|restart|delete|start all
```

操作单个 app，使用 name 或者 id 区分：

```
pm2 stop|reload|restart|delete|start name|id
```

如下图，注意 status 一栏

```
root@demo01:~# service pm2-root status
Status for pm2:
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
a	0	N/A	fork	0	stopped	0	0	0%	0 B	root	disabled

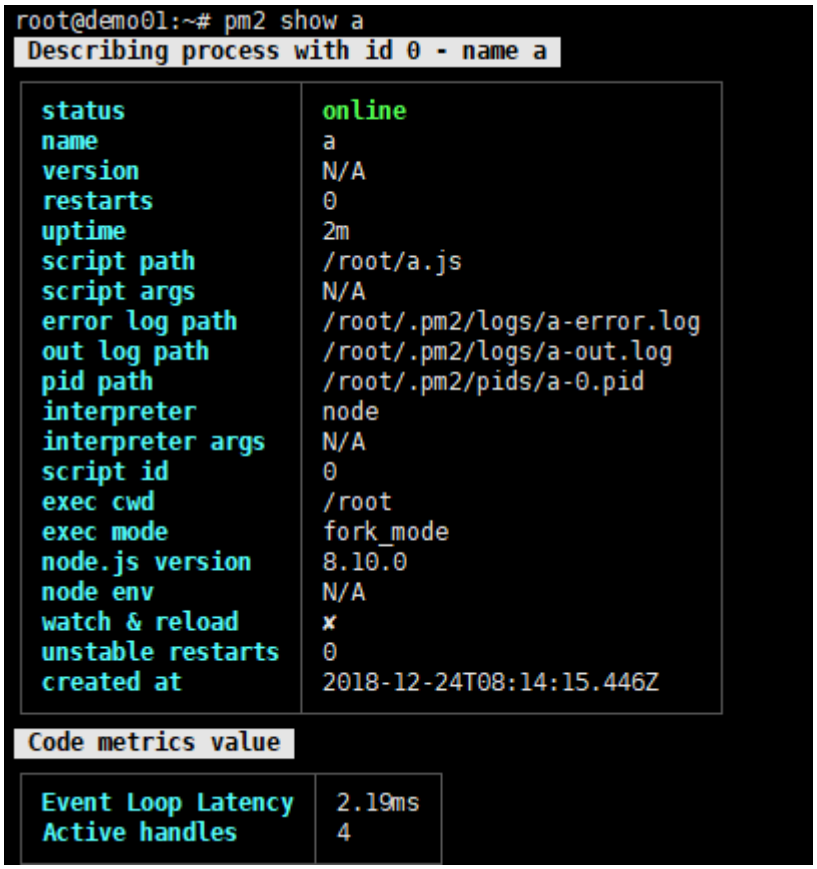
```
 Use `pm2 show <id|name>` to get more details about an app
root@demo01:~#
root@demo01:~#
root@demo01:~#
root@demo01:~#
root@demo01:~#
root@demo01:~# pm2 start 0
[PM2] Applying action restartProcessId on app [0](ids: 0)
[PM2] [a](0) ✓
[PM2] Process successfully started
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
a	0	N/A	fork	13143	online	0	0s	0%	17.6 MB	root	disabled

```
 Use `pm2 show <id|name>` to get more details about an app
root@demo01:~#
```

使用如下命令查看 app 更加详细的信息：

```
pm2 show <id|name>
```


root@demo01:~# pm2 show a
Describing process with id 0 - name a
status online
name a
version N/A
restarts 0
uptime 2m
script path /root/a.js
script args N/A
error log path /root/.pm2/logs/a-error.log
out log path /root/.pm2/logs/a-out.log
pid path /root/.pm2/pids/a-0.pid
interpreter node
interpreter args N/A
script id 0
exec cwd /root
exec mode fork_mode
node.js version 8.10.0
node env N/A
watch & reload ✘
unstable restarts 0
created at 2018-12-24T08:14:15.446Z
Code metrics value
Event Loop Latency 2.19ms
Active handles 4

第四章 nginx 配置

```
server {
    listen 80;
    server_name example.com;

    location / {
        proxy_pass http://172.16.7.46:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```