

学知

学知部署文档教程

目录

- [系统要求](#)
- [购买域名和服务器](#)
- [远程登录服务器命令行](#)
- [远程管理服务器文件](#)
- [搭建服务器的nodejs环境](#)
- [通过pm2让node服务常驻](#)
- [配置nginx反向代理nodejs端口](#)
- [安装mongodb](#)
- [向线上mongodb导入数据库](#)
- [数据库备份](#)
- [启动服务](#)
- [项目部署的一般流程总结](#)

系统要求

- Linux 64 位操作系统
- Mongodb 4.0以上
- node.js 4.4以上
- 必须有公网ip
- 建议CPU 双核2GHZ以上
- 建议内存2G以上
- 建议带宽2Mb以上
- 建议剩余硬盘空间20GB以上

购买域名服务器

本项目使用的是腾讯云服务器，操作系统为CentOS。

远程登录服务器命令行

在腾讯云登陆账号，远程连接至服务器命令行。

腾讯云默认关闭25端口，鉴于此项目需要邮箱服务，需要手动开启25端口

远程管理服务器文件

使用winscp、filezilla等软件远程连接服务器，可以进行文件的传输、修改

在/root目录下，上传打包后的后端jar文件

在/home/build目录下，上传打包后的用户前端react项目

在/home/admin目录下，上传打包后的管理员前端react项目

搭建服务器的nodejs环境

更新一下系统

```
sudo apt-get update
```

安装包文件

```
sudo apt-get install vim openssl build-essential libssl-dev wget curl git
```

打开github 找到creationix/nvm

```
复制wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.2/install.sh  
| bash
```

新打开窗口登录服务器

```
nvm install v8.1.3
```

```
nvm use v8.1.3
```

指定系统的默认node版本

```
nvm alias default v8.1.3
```

node -v 查看node版本

安装cnpm

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

输入命令

```
echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf &&  
sudo sysctl -p
```

安装模块

```
cnpm i pm2 webpack gulp grunt-cli -g
```

在服务器更目录下

```
vim app.js
```

编辑文件

```
const http = require("http");

http.createServer(function(req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'})
  res.end('hello world!');
}).listen(8080)

console.log('server running on http://47.94.211.202:8080/');
```

启动服务

```
node app.js
```

如果端口无法访问，需要到阿里云控制台安全组配置里面配置TCP端口，自定义TCP，端口范围8080/8080，地址段访问，授权对象 0.0.0.0/0

通过pm2让node服务常驻

启动node服务

```
pm2 start app.js
```

查看服务列表

```
pm2 list
```

查看应用详细信息

```
pm2 show 应用名字
```

查看日志， control + c退出

```
pm2 logs
```

配置nginx反向代理nodejs端口

原理：nodejs不具备权限通过外网访问80端口，用root权限启动对80端口的监听，把来自80端口的流量分配给node服务的另外一个端口，实现node服务的代理，实际上就是把80端口的请求都转发到nodejs启动的8080端口上

先关闭apache服务

```
sudo service apache stop
```

删除apache服务

```
update-rc.d -f apache2 remove
```

```
sudo apt-get remove apache2
```

更新包列表

```
sudo apt-get update
```

安装nginx

```
sudo apt-get install nginx
```

查看版本

```
nginx -v
```

进入目录`cd /etc/nginx/conf.d` `pwd`查看路径

在`/etc/nginx/vhost`下新建配置文件，文件命名以域名加端口的方式归类，多个项目对应多个服务，考虑负载均衡。

编辑配置文件

```
server {
    listen 8080;
    # server_name XueZhi;

    root /home/build;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ /index.html;
    }
    location ^~ /assets/ {
        gzip_static on;
        expires max;
        add_header Cache-Control public;
    }
    error_page 500 502 503 504 /500.html;
    client_max_body_size 20M;
    keepalive_timeout 10;
}

server {
    listen 8090;
    # server_name admin;

    root /home/admin;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ /index.html;
    }
    location ^~ /assets/ {
        gzip_static on;
        expires max;
        add_header Cache-Control public;
    }
    error_page 500 502 503 504 /500.html;
    client_max_body_size 20M;
    keepalive_timeout 10;
}
```

返回上一层目录 [/etc/nginx](#)

编辑配置文件

```
sudo vi nginx.conf
```

确保`include /etc/nginx/conf.d/*.conf;`注释打开

检测配置文件有没有错误

```
sudo nginx -t
```

重启nginx服务器

```
sudo nginx -s reload
```

重启pm2

```
pm2 restart all
```

在浏览器上直接访问服务器ip地址就可以了

去掉浏览器网络的nginx版本信息，进入/etc/nginx目录

```
sudo vi nginx.conf
```

把`server_tokens off;`注释拿掉

重启服务器`sudo service nginx reload`这样再访问浏览器是，network里的nginx版本信息就不会显示了

安装mongodb

按照文档的步骤操作<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

安装mongodb

```
sudo apt-get update
```

```
sudo apt-get install -y mongodb-org
```

开启mongodb服务

```
sudo service mongod start
```

检查是否开启，查看日志文件

```
cat /var/log/mongodb/mongod.log
```

打开mongodb命令行环境

```
mongo
```

停止服务

```
sudo service mongod stop
```

重启

```
sudo service mongod restart
```

修改mongodb默认端口号

```
sudo vi /etc/mongod.conf
```

```
net:
  port: 5000
  bindIp: 127.0.0.1
```

重启

```
sudo service mongod restart
```

有防火墙需要修改防火墙端口，没有防火墙配置直接登录**mongo --port 5000**

向线上mongodb导入数据库

使用以下语句

```
mongoexport -d 数据库名 -c 单表名 -q '{"name": {$ne: null}}' -o ./数据库名-表名.json
```

导出打包本地数据库

```
mongoexport -d test -c user -o /home/a.json --type json
mongoexport -d qa -c question -o /Users/xiekang/Desktop/123/b.json --type json
mongoexport -d check -c verification -o /home/c.json --type json
mongoexport -d check -c administrator -o /home/d.json --type json
```

将json文件从本地上传到服务器

运行以下指令

```
mongoimport -d test -c user --file /Users/xiekang/Desktop/123/a.json --type json
mongoimport -d qa -c question --file /Users/xiekang/Desktop/123/b.json --type json
mongoimport -d check -c verification --file /Users/xiekang/Desktop/123/c.json --type json
mongoimport -d check -c administrator --file /Users/xiekang/Desktop/123/d.json --type json
```

导入到服务器数据库

数据库备份

在服务器更目录，创建数据库备份执行脚本

```
mkdir tasks
cd tasks
vi 数据库名.backup.sh
```

编辑脚本文件

```
#!/bin/sh

backUpFolder=/home/管理员名/backup/数据库名
date_now=`date +%Y_%m_%d_%H%M`
backFileName=数据库名_${date_now}

cd $backUpFolder
mkdir -p $backFileName
```



```
mongodump -h 127.0.0.1:29999 -d 数据库名 -u 数据库名_wheel -p 密码 -o $backFileName

tar zcvf $backFileName.tar.gz $backFileName

rm -rf $backFileName
```

在更目录下，创建文件夹

```
mkdir backup
cd backup
mkdir 数据库名
```

在服务器根目录下执行脚本文件

```
sudo sh ./tasks/数据库名.backup.sh
```

启动系统的定时任务

```
crontab -e
```

选择第2. `/bin/nano`

编辑

```
19 00 * * * sh /home/管理员名称/tasks/项目名称.backup.sh
```

//分钟 小时 `control + x shift + y` 回车

启动服务

查看所需端口号是否被占用

```
lsof -i:port
```

如果被占用，使用 `kill %pid` 命令杀掉进程

在后端进入到/root目录下，使用 `nohup &` 命令使jar项目在后台运行

```
nohup java -jar zuul.jar&
nohup java -jar mail.jar&
nohup java -jar user.jar&
nohup java -jar qa.jar&
nohup java -jar admin.jar&
```

运行命令jobs，查看服务都在运行

进入http://49.234.73.158:8085，可以看到服务都已注册，启动完成

项目部署的一般流程总结

- 1.配置服务器、域名
- 2.在本地项目后端服务入口文件写好没有用过的端口号
- 3.本地项目，修改mongodb本地和线上的连接方式，线上写好用户名和密码
- 4.本地，写好发布版本项目，修改名字，仓库地址，服务器ip，端口号
- 5.服务器上，配置代理
- 6.服务器下，新增nginx配置文件，在`/etc/nginx/conf.d`，可直接复制其他配置文件`sudo cp 文件夹名 文件夹名`，编辑nginx 配置文件，`sudo vi ...` 修改域名，端口号，名称
- 7.在本地，先提交代码到服务器
- 8.部署代码，第一次，初始化服务器项目目录结构，`pm2 deploy ecosystem.json production setup`，第二次部署发布`pm2 deploy ecosystem.json production`
- 9.在服务器上，检查`pm2 list`，如有问题，不断重启，先停pm2服务，检查日志，`pm2 logs`
- 10.修改防火墙，`sudo vi /etc/iptables.rules`，增加新的端口，
重启防火墙，`iptables-restore < /etc/iptables.rules`
重启nginx `sudo nginx -s reload`
- 11.启动后台服务，运行jar，使用eureka监测服务运行状况