

# Node服务端部署

---

## 1. 服务器购入流程

1. 选择合适的厂商，如国内的阿里云、腾讯云、国外的AWS、Vultr等都可以，根据自己的需要选择（下文以阿里云为例）
  - a. 尽量跟自己所在区域近一些，response会快点；
  - b. 如果有自己搭 的诉求，可以考虑海外站点或者国内厂商的海外服务器；
2. <https://www.aliyun.com/>
  - a. 选择ECS，建议买台便宜的自己学习；
  - b. 配置安全组：防火墙按序打开，如果防火墙开启，端口号也要按需配置；

```

1  查看防火墙状态
2  firewall-cmd --state
3
4  启动firewall:
5  systemctl start firewalld
6
7  临时关闭firewall (设置开机自启后会打开)
8  systemctl stop firewalld
9
10 永久关闭firewall
11 systemctl disable firewalld
12
13 设置开机自启:
14 systemctl enable firewalld
15
16 重启防火墙 (每次操作后重启防火墙) :
17 systemctl restart firewalld
18
19 开端口命令: firewall-cmd --zone=public --add-port=80/tcp --permanent
20
21 所有IPV4
22 端口范围: -1/-1 源:0.0.0.0/0
23
24 SSH端口号
25 端口范围: 22/22 源:0.0.0.0/0
26
27 远程连接端口
28 端口范围: 3389/3389 源:0.0.0.0/0

```

- c. 域名: 根据自己的需要, 看是否需要购买域名, 若购买域名, 需要将域名解析到自身的服务器上;
- d. 证书: 购买ECS后, 可以免费生成一个证书, 有效期一年;

## 2. 本地环境设置

主要配置VScode, 常规的开发环境不予补充

### 2.1. VScode插件

此处常规开发中的插件等不介绍了

### 1. remote-SSH

使用VScode远程连接服务器

地址: <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh>

支持用户名&密码配置 和 秘钥配置, 建议使用后者

配置内容

```
▼ Bash | 复制代码
1  本地配置入口
2  ~/.ssh/config
3
4  // example
5  // 本地配置入口
6  Host xianzao
7      HostName 47.243.224.215
8      User root
9      IdentityFile ~/.ssh/id_rsa
10
11 // 服务器配置入口
12 ~/.ssh/authorized_keys
```

### 2. 其他常用的

- a. [Markdown All in One](#): 查看编写md比较好的插件
- b. [MySQL](#): 查看表结构内容&写SQL的插件
- c. [Image preview](#): 图片预览
- d. [Import Cost](#): 查看引入包大小
- e. [koroFileHeader](#): 定义文件&函数头部说明
- f. [open in browser](#): 浏览器中打开
- g. [Time Master](#): 时间管理大师
- h. [TODO Highlight](#): TODO好帮手
- i. etc.....

## 2.2. Mac

### 1. iterm2 & oh-my-zsh & powerlevel10k

- a. [iterm2](#): Mac上比较好用的终端, 可支持扩展的能力比较多, 可玩性强
- b. [oh-my-zsh](#): 基于zsh的, 可扩展性极其强大的终端命令行框架
- c. [powerlevel10k](#): 最牛逼的Zsh插件, 个性化定制首选
  - o 插件可以从<https://github.com/unixorn/awesome-zsh-plugins> 查找, 或者自行Google, 也可以根据业务自定义alias bash等

Bash | 复制代码

```
1  plugins=(git zsh-autosuggestions zsh-syntax-highlighting vi-mode aliases  
    git-open history autojump)
```

小Tips: 搜相关内容, 可以在github上通过awesome-XXX 尝试搜索, 说不定有惊喜

- 2. [Alfred](#): 开发提效神器, 如快速搜索, 快速启动应用等, 也可以根据自身业务诉求, 自定义一些workflow的插件
  - a. <https://github.com/alfred-workflows/awesome-alfred-workflows>

## 2.3. windows

- [cmdr](#): windows上比较好用的终端
- [utools](#): windows上的Desktop插件集合

## 3. 服务器基础设置

建议学习一下linux常见指令, 对服务器操作也有一些理解, 掌握grep等常见的操作, 推荐《[鳥哥的Linux 私房菜](#)》, 早日自己玩转linux

- 1. 防火墙&端口配置: 要求不严格可以关闭防火墙, 如需开启防火墙请配置打开对应端口号
- 2. 环境配置: [iterm2](#) & [oh-my-zsh](#) & [powerlevel10k](#)
- 3. 根据所需配置告警: 内存使用超阈值; 带宽不够; 结合阿里云监控进行配置

## 4. node配置

- [node&npm](#): 下载长期维护版本 (注意添加软链设置全局引用) 网上资源很多, 不一一介绍
- [nvm](#) / [n](#): node版本维护管理, 区别在于nvm是一个完整的shell版本, n是一个npm包, 所以n更轻便, 但安装路径默认在 `/usr/local/lib/node_modules` 下, 还是会共用全局的 node/npm 目

录，因此不能很好的满足『按不同 node 版本使用不同全局 node 模块』的需求。

- [nrm](#)：设置npm源，如果不用可以将npm默认注册为cnpm
- [pm2](#)：node进程管理，可以实现性能监控、自动重启、负载均衡等功能
- [docker](#)：实现node.js应用容器化（保证应用间环境隔离，且宿主机能够资源利用最大化）

## 5. nginx

如果项目中只有node，可以使用node控制path，如果有其他服务器，还是建议使用nginx

是一款轻量级的HTTP服务器，采用事件驱动的异步非阻塞处理方式框架，这让其具有极好的IO性能，常用于服务端的反向代理和负载均衡，建议自学

主要能力包括：

- 静态资源服务，通过本地文件系统提供服务；
- 反向代理服务，延伸出包括缓存、负载均衡等；
- API 服务，OpenResty（集成lua的nginx，而lua是c解释执行的脚本语言，保证nginx可以不用c开发，可以有效解决高并发的问题）

小作业：可以尝试使用[goAccess](#)配置可视化的nginx的访问log

## 6. 搭建个人的静态博客

个人建议，可以使用[gatsby](#)，然后部署在自己的服务器上，部署好了可以分享一下，一起看看谁做的博客最吸引人

1. 支持node配置，插件市场丰富
2. GraphQL
3. 支持数据格式转化，支持以md格式配置blog
4. 支持TS、SSR、PWA，SEO等