

35 拥有自己的全战线上真实产品

更新时间：2019-10-14 09:29:48



“ 更多一手资源请+V：AndyqcI
只有在那崎岖的小路上不畏艰险奋勇攀登的人，才有希望达到光辉的顶点。
aa：3118617541 ————马克思 ”

下面就要在我们上一节申请的云服务器上安装项目部署所需要的相关程序了，如果你在上一节选择的是windows操作系统，可以参考我们之前《项目环境搭建》章节。

安装配置Nginx

Nginx是一个高性能的HTTP和反向代理服务，许多的大型网站都会采用Nginx来进行HTTP服务器托管，当然我们这里也要采用Nginx。

安装编译环境gcc g++:

进入到root目录： `cd /root/`

```
yum -y install make zlib zlib-devel gcc-c++ libtool openssl openssl-devel
```

安装PCRE:

PCRE功能是让nginx有rewrite功能，下载PCRE:

```
wget http://downloads.sourceforge.net/project/pcre/pcre/8.35/pcre-8.35.tar.gz
```

解压安装包:

```
tar zxvf pcre-8.35.tar.gz
```

进入安装包目录: `cd pcre-8.35`

编译: `./configure`

安装: `make && make install`

安装**Nginx**:

回到root目录: `cd /root/`

```
wget http://nginx.org/download/nginx-1.16.0.tar.gz
```

解压安装包:

```
tar zxvf nginx-1.16.0.tar.gz
```

注意, 这里要选好当前解压地址, 解压完成就是**Nginx**所在的目录, 这里我们在**root**根目录下解压。

进入安装目录:

```
cd nginx-1.16.0
```

编译: `./configure --with-http_ssl_module`

安装: `make && make install`

默认情况下, **nginx**会安装到 `/usr/local/nginx` 下面, 你可以指定到安装到**root**目录下:

编译: `./configure --prefix=/root/nginx --with-http_ssl_module`

安装: `make && make install`

启动和停止**Nginx**:

进入**Nginx**安装目录:

```
cd root/nginx/sbin/
```

停止**Nginx**: `./nginx -s stop`

启动**Nginx**: `./nginx`

配置**Nginx**配置文件:

```
user root;
worker_processes 1;

pid /root/nginx/logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;

    sendfile on;
```

```

keepalive_timeout 65;

gzip on;

server {
    listen 80;
    server_name app.nihaoshijie.com.cn localhost;
    # 这里将非HTTPS的地址转向到HTTPS
    rewrite ^(.*) https://$host$1 permanent;

    location / {
        root html;
        index index.html index.htm;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}

# HTTPS server
# HTTPS 配置

server {
    listen 443 ssl;
    server_name app.nihaoshijie.com.cn;

    # 配置证书路径
    ssl_certificate /root/nginx/cert/c.pem;
    ssl_certificate_key /root/nginx/cert/c.key;

    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 5m;

    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    location / {
        # 启用支持websocket连接
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        # 给静态文件添加缓存
        location ~ .*\. (js|css|png|jpg|jpeg) \. {
            proxy_pass http://localhost:3000;
            expires 3d; # 3天
        }

        proxy_pass http://localhost:3000;
    }
    # 给amdinMogo配一个路径
    location /adminMogo {
        proxy_pass http://localhost:1234/adminMogo;
    }
}

```

更多一手资源请+V : AndyqcI
qq : 3118617541

这个配置文件，我们主要做了下面操作：

1. 配置HTTPS服务，可以看到，我们将之前下载的证书配置上去了。
2. 使用 `proxy_pass` 做了域名转发，分别将项目用的3000端口和adminMogo用的1234端口代理了上去。
3. 对一些静态资源文件，例如 `.js`，`.png` 等做了缓存。

建议大家可以直接将代码全部替换，一般修改过配置文件，需要重新加载: `./nginx -s reload`

配置文件地址：

```
vi root/nginx/conf/nginx.conf
```

注意，在开启nginx的http和https服务后，需要在阿里云的云服务器实例安全组策略中添加80端口和443端口的开启。

找到实例中的安全组配置：



进入之后，点击添加安全组配置：



添加http的80端口：

添加安全组规则 添加安全组规则

网卡类型：内网

规则方向：入方向

授权策略：允许

协议类型：HTTP (80)

* 端口范围：80/80

优先级：1

授权类型：IPv4地址段访问

* 授权对象：0.0.0.0/0

描述：nginx

长度为2-256个字符，不能以http://或https://开头。

确定 取消

添加https的443端口：

更多一手资源请+V：AndyqcI
aa：3118617541

添加安全组规则 添加安全组规则

网卡类型：内网

规则方向：入方向

授权策略：允许

协议类型：HTTPS (443)

* 端口范围：443/443

优先级：1

授权类型：IPv4地址段访问

* 授权对象：0.0.0.0/0

描述：nginx https

长度为2-256个字符，不能以http://或https://开头。

确定 取消

安装配置Node.js

下载Node.js:

wget https://nodejs.org/dist/v10.11.0/node-v10.11.0-linux-x64.tar.xz

解压，同样在root根目录下：

```
tar xvf node-v10.11.0-linux-x64.tar.xz
```

创建软链接，使node和npm命令全局有效：

```
ln -s /root/node-v10.11.0-linux-x64/bin/node /usr/local/bin/node
ln -s /root/node-v10.11.0-linux-x64/bin/npm /usr/local/bin/npm
```

验证Node.js安装成功：

```
node -v
v10.11.0
```

安装配置MongoDB

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.2.9.tgz
```

解压，同样在root根目录下：

```
tar xvf mongodb-linux-x86_64-3.2.9.tgz
```

进入 `mongodb-linux-x86_64-3.2.9/bin` 目录：

新建 `data` 目录： `mkdir data`

进入 `data`： `cd data`

新建 `data/db` 目录： `mkdir db`

新建 `log` 文件： `touch /data/log`

启动MongoDB：

```
cd /root/mongodb-linux-x86_64-3.2.9/bin

./mongod
--dbpath = ../data/db #数据文件存放目录
--logpath = ../data/log #设置日志文件地址
--logappend #以追加的方式记录日志
--fork #以守护程序的方式启用，即在后台运行
```

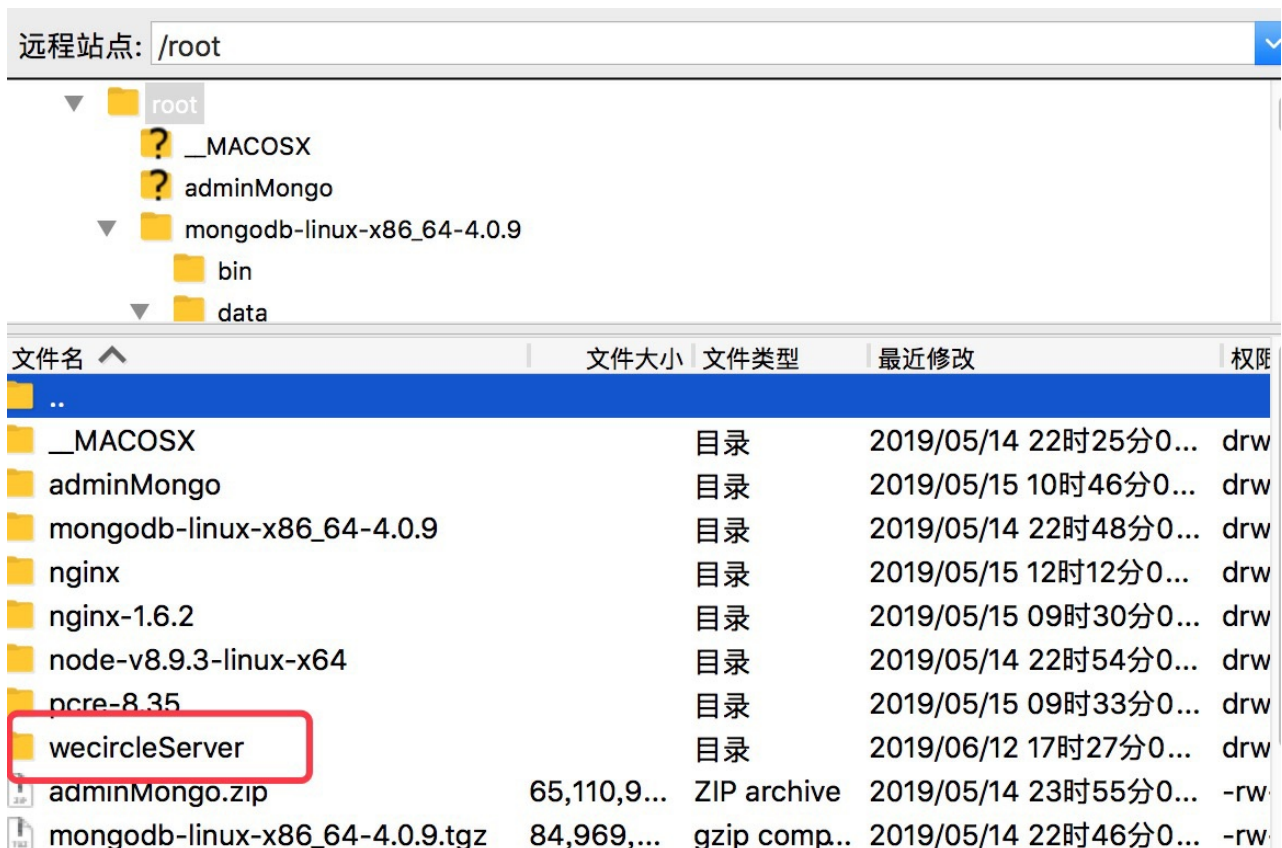
这里和我们在最初讲解的环境搭建章节基本是一致的，`--fork` 代表将后台运行mongodb，启动之后 `/data/db` 目录会有数据内容。

打包项目，发布的服务器

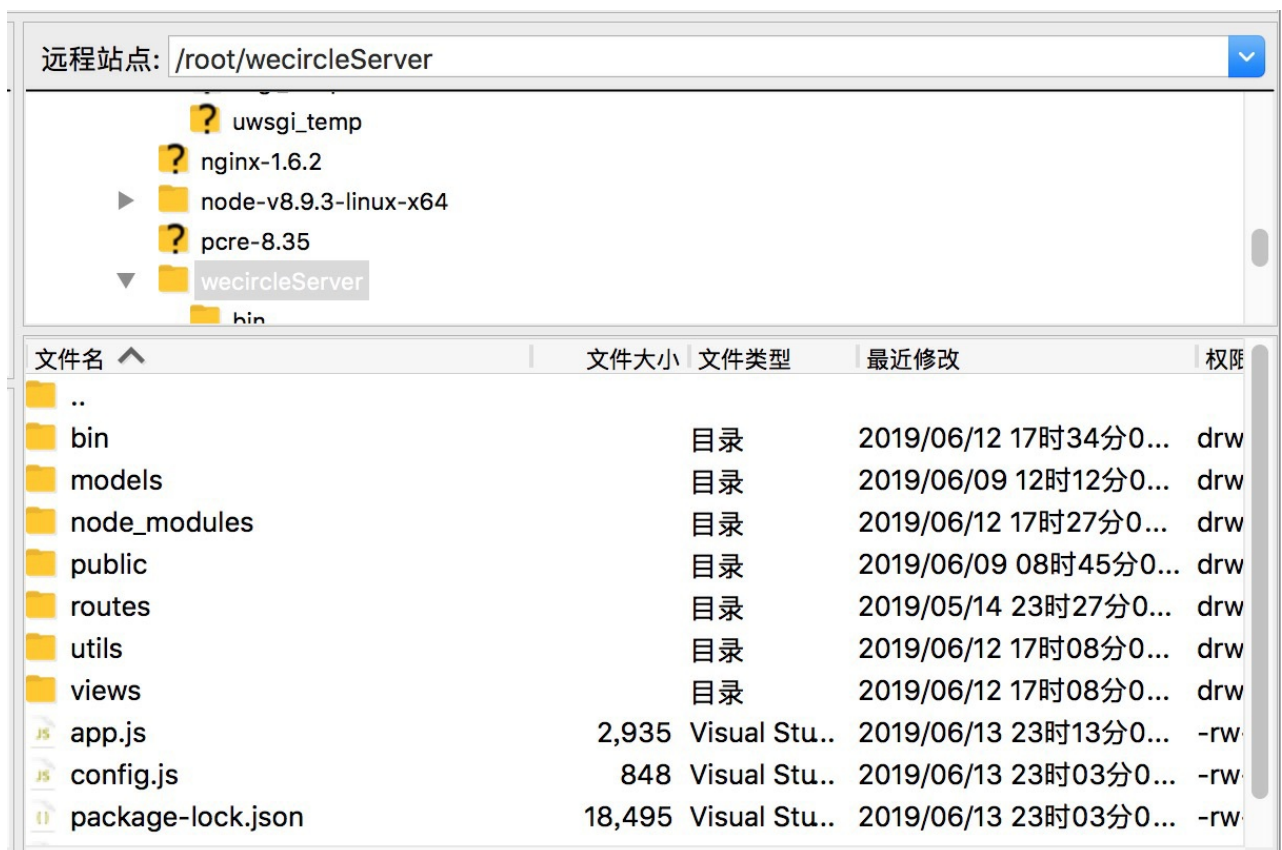
打包前端文件：

```
npm run build
```

将 `dist`，生成的文件拷贝到 `wecircleServer` 的 `public` 目录下，然后将 `wecircleServer` 打包发到服务器上，可以选择 FTP 工具：



之前我们讲过 `wecircleServer` 是一个基于 `Express` 项目，而同时也是一个 `web` 服务器，这里我们将前端的静态文件直接放在 `Express` 项目的 `public` 文件夹下，采用同域，利用了 `Express` 项目的 `web` 服务器静态资源托管功能。



PM2是Node.js进程管理工具，可以利用它来简化很多node应用管理的繁琐任务，如性能监控、自动重启、负载均衡等，而且使用非常简单。

PM2的主要特性：

- 内建负载均衡（使用node cluster集群模块，可以使用服务器上的所有cpu）
- 后台运行（node app.js 这种命令是直接在前台运行的，不稳定，很容易断）
- 0秒停机重载（应该是上线升级的时候 不需要停机）
- 停止不稳定的进程（避免无限循环）
- 控制台检测

安装PM2(全局安装):

```
npm install -g pm2
```

创建软链接，使pm2命令全局有效：

```
ln -s /root/node-v10.11.0-linux-x64/bin/pm2 /usr/local/bin/pm2
```

启动wecircle服务，进入wecircleServer根目录：

```
pm2 start server.js --name wecircle
```

使用 `pm2 list` 可以查看项目运行情况：

更多一手资源请+V：AndyqcI
qq：3118617541

```
[root@iZ2zea6tb04zq3q2l8s7y7Z ~]# pm2 list
```

Name	id	mode	status	u	cpu	memory
adminMogo	1	fork	stopped	2	0%	0 B
wecircle	2	fork	online	55	0.3%	97.7 MB

Use `pm2 show <id|name>` to get more details about an app

```
[root@iZ2zea6tb04za3a2l8s7v7Z ~]# █
```

重启wecircle服务：

```
pm2 restart wecircle
```

查看项目日志：

```
pm2 log wecircle
```



```
2|wecircle | GET /manifest.json 200 1.796 ms - 458
2|wecircle | GET /static/js/chatlist.d9e6187e.js?__uncache=2019-6-13%2000%3A31%3A56 200 0.435 ms - 3796
2|wecircle | GET /static/js/chunk-vendors.e1323244.js?__uncache=2019-6-13%2000%3A31%3A56 200 0.491 ms - 322544
2|wecircle | GET /index.html?__uncache=2019-6-13%2000%3A31%3A56 200 0.409 ms - 1809
2|wecircle | GET /favicon.ico?__uncache=2019-6-13%2000%3A31%3A56 200 0.397 ms - 4286
2|wecircle | GET /lib/slider/slider.js?__uncache=2019-6-13%2000%3A31%3A56 200 0.524 ms - 8356
2|wecircle | GET /manifest.json?__uncache=2019-6-13%2000%3A31%3A56 200 0.429 ms - 458
2|wecircle | GET /lib/weui/weui.min.css?__uncache=2019-6-13%2000%3A31%3A56 200 0.498 ms - 51037
2|wecircle | GET /lib/weui/weui.min.js?__uncache=2019-6-13%2000%3A31%3A56 200 0.417 ms - 36854
2|wecircle | GET /lib/weui/weui.js?__uncache=2019-6-13%2000%3A31%3A56 200 0.560 ms - 159923
2|wecircle | GET /favicon.ico 304 0.426 ms - -
2|wecircle | GET /img/icons/android-touch-icon-192x192-1-touming.png 200 0.517 ms - 14640
```

PM2会收集项目中使用 `console.log()` 和 `console.error()` 的日志，并记录在文件中。

更多的PM2命令，可以[参考文档](#)。

小结

更多一手资源请+V：Andyqc1
qq：3119617541

本章节主要讲解了云服务器项目环境的搭建，项目部署的步骤，其中所使用到的相关安装包和配置文件，都会在源码里面供大家参考使用。

}