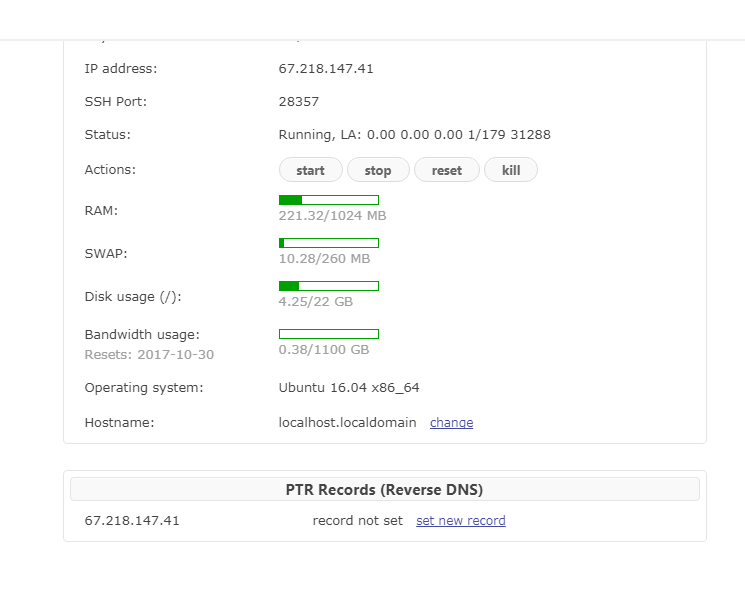
# [【全栈项目上线（vue+node+mongodb）】](https://segmentfault.com/a/1190000011772497)

1. Vps服务器购买操作（服务器购买后最好选择Ubuntu系统）
2. 结账方式：有支付宝极可能选择支付宝，也可以选择paypal，最好不要绑定信用卡，绑定信用卡得用另一张信用卡替换；
3. **选择搬瓦工（支持ss价格实惠）**
4. 选择搬瓦工（需要翻墙）
5. 翻墙下载shadpwsocksR,配置参照下方图片：
6. 
7. 服务器ip:69.171.77.153
8. 服务器端口：1994
9. 密码：wangdapao
10. 搬瓦工main control
11. 
12. **选购阿里云**
13. 注意事项：

A、千万别选择window系统

B、如果选择了window系统，不能更换Linux系统，解决方法：提交工单，去退货，然后重新购买；

C、如果你是学生可以通过学生通道购买9.9一个月；

D、如果你不是学生推荐https://common-buy.aliyun.com/?spm=5176.10003347.843204.1.Ql3ciJ&commodityCode=swas&request=undefined#/buy

E、如果你的域名没有备案，请购买香港主机或其他国外主机

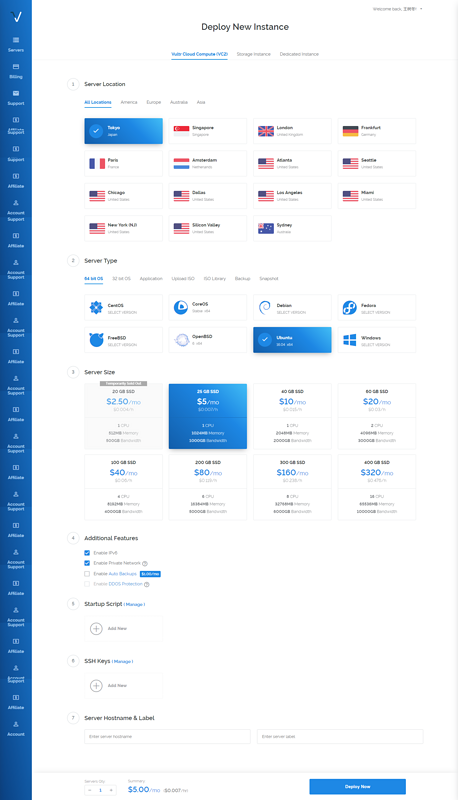
1. 选购vultr

注册账号

登录进去

注意（不要绑银行卡）

支付方面：paypal（所有银联卡都支持）



选择数字海洋 digitalocen

注意：

国外口碑不错，digitalocean 如果你有教育邮箱，就是edu结尾的邮箱，可以去领100美元优惠券，这个查的很严，两个账户用一个支付方式，都使用了优惠券就会被查出来

优惠券有限期：2年；

登录网址：<https://cloud.digitalocean.com/login>

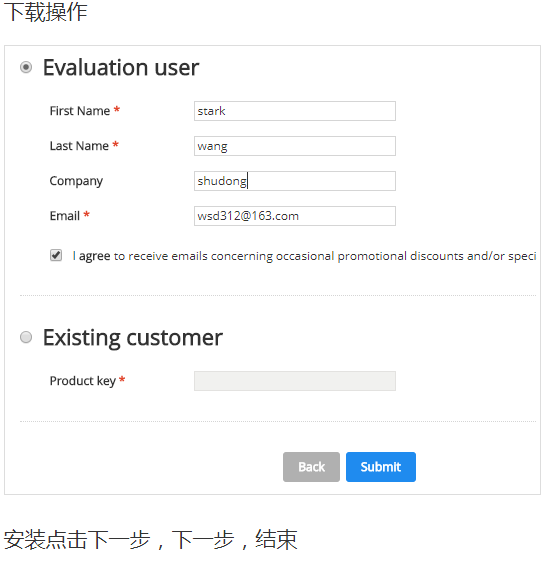
1. **连接LINUX工具**

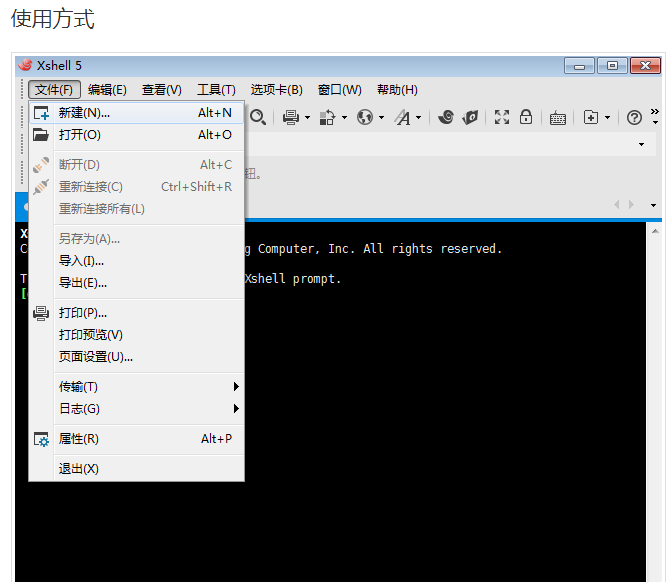
Putty(免费）

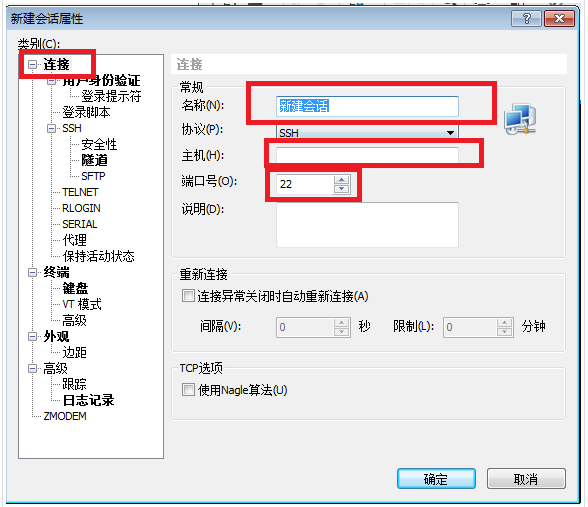
Xshell（收费）（推荐）教程使用xshell操作

**安装xshell**

下载地址（http://www.netsarang.com/download/main.html）





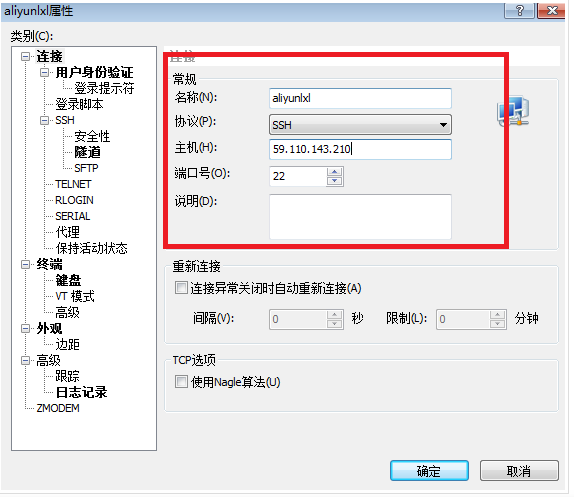


名称：是自己随便填写的，以区分自己的其他主机

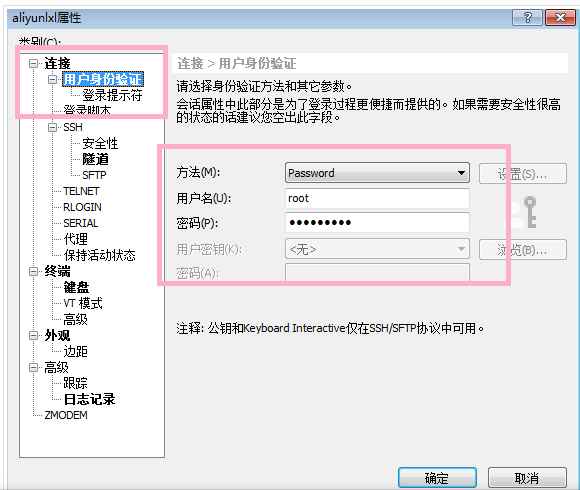
主机：是自己购买的服务器ip

端口号：是自己购买的服务器端口，一般默认是ssh 22，如果你买的国外主机，比如搬瓦工，会像你的邮箱发送几封邮件，其中一封，会标明有主机ip端口号，那个端口号需要在这填写，搬瓦工的端口号，不是22（切记）

如下图：



输入用户名和密码：



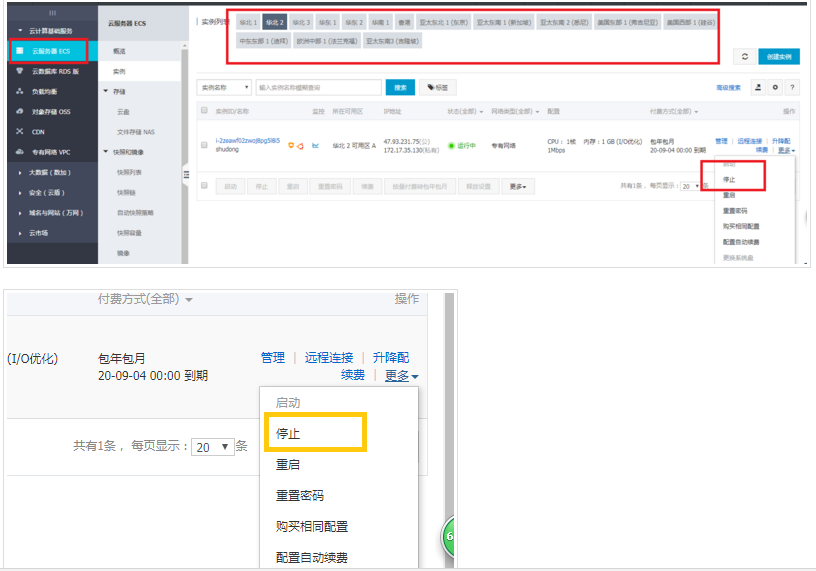
如果你刚买的主机，一般管理员用root来登录

用户名是root

密码是自己设定的，如果是搬瓦工，邮箱会有这个信息

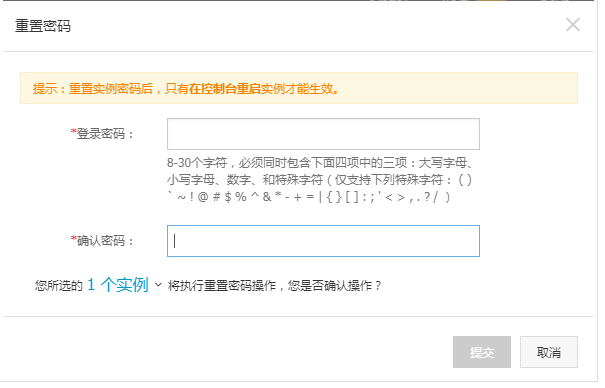
**阿里云怎么找回或设置自己的密码**

如果是阿里云的话需要



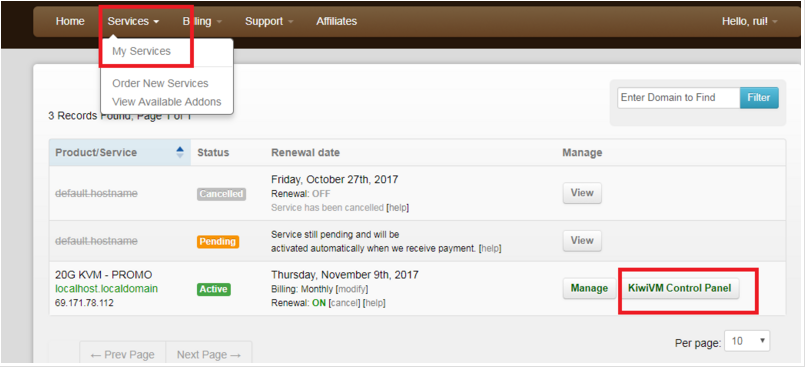
先停止这个服务器，然后





等重置完密码之后，就把自己设定的密码，在这登录

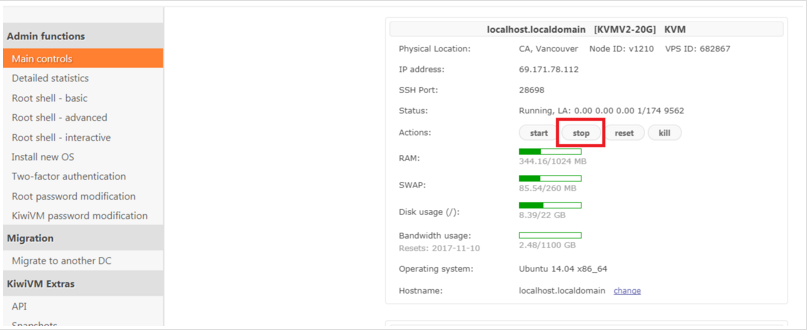
**搬瓦工怎么找回自己的密码**



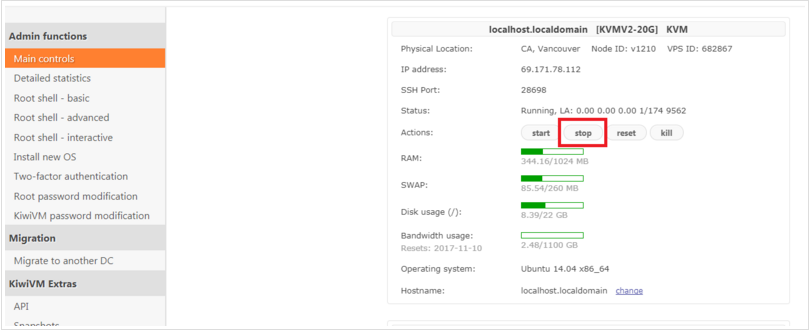
进入控制面板的网址：<https://kiwivm.64clouds.com/main.php>

**修改密码**

改密码之前先stop

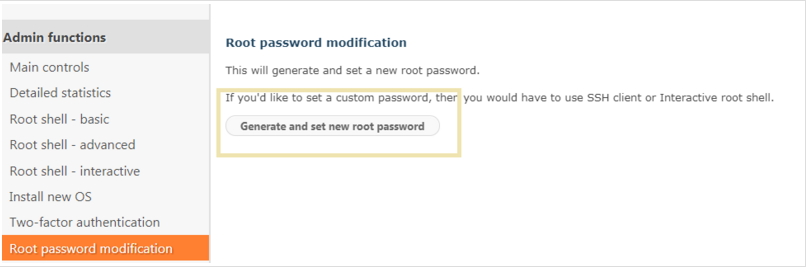


自定义设置密码



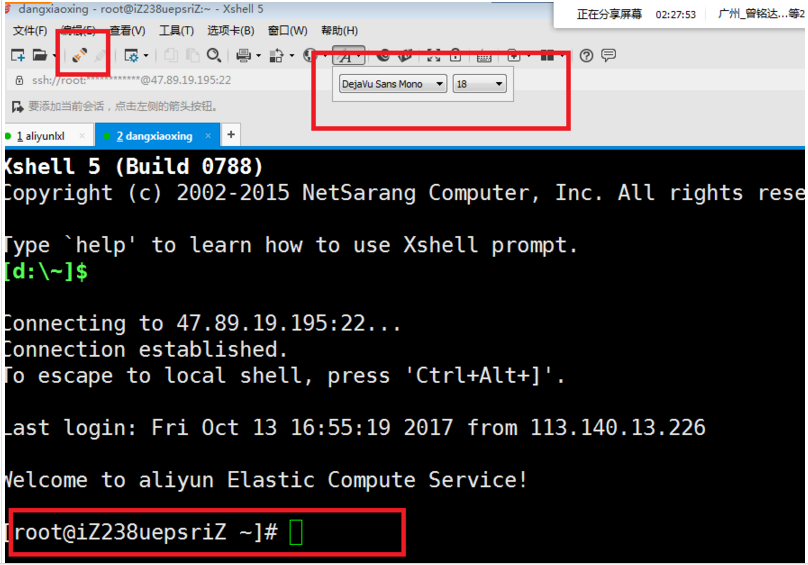
如果这个不能用，请选择自动设置密码：下面图示

自动设置密码：



设置了之后，会出现一个密码，需要记住，只出现一次，一定要记住

当我们吧服务器账户密码度搞定，然后登陆



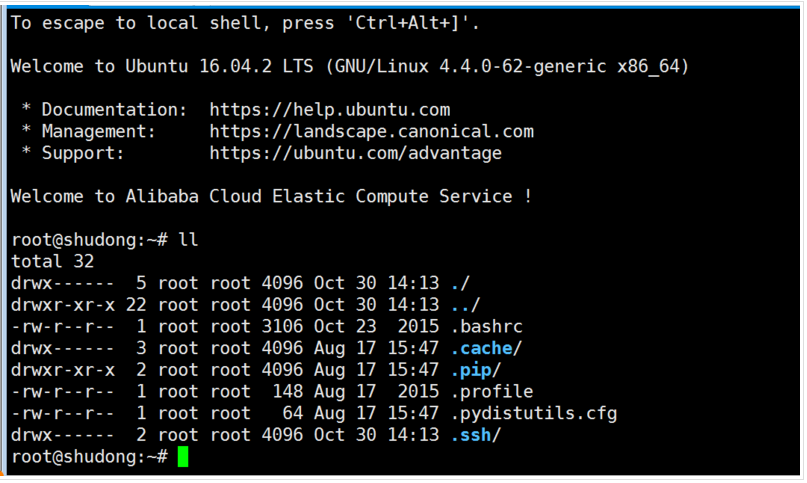
阿里云更换系统盘或初始化磁盘



初始化磁盘，并且设置密码



**链接成功**



出现以下命令

[root@iZ238uepsriZ ~]**#**

root 是用户名  
iZ238uepsriZ 这个是主机名字  
~ 是路径，波浪线代表家目录 家目录就是用户所在的目录  
‘#’ 代码管理员权限操作

**执行一些常用的shell**

**pwd** ：当前所在的路径

**cd** :切换路径

**ls** ：查看当前文件下面都有神马

**ll** ：是**ls** -**l** 的别名

**cp** :复制

mv :重命名，剪切

## 接下来安装Nginx web环境

会安装以下环境：

Nginx

php

mysql

Mongodb

**Node**

**pm2**

git

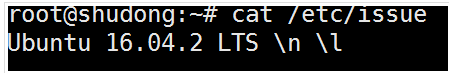
Docker

配置常用环境

安装zshell

**请先检查自己的系统**

cat /etc/issue



如果是centos



下面是Ubuntu环境安装

先更新源码包的仓库：apt-get **update** -y

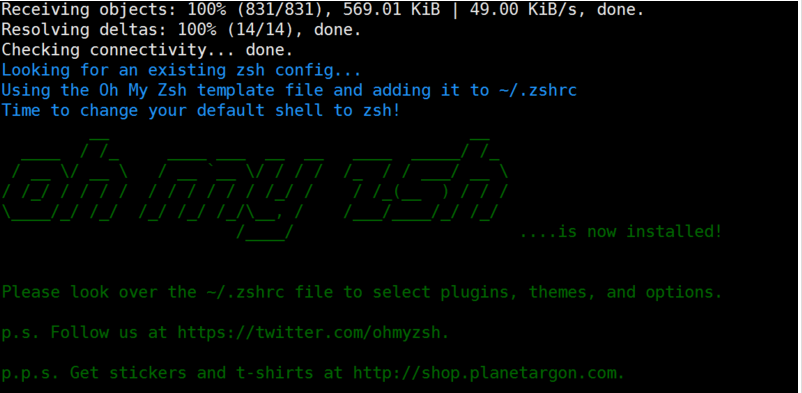
在安装之前我们先安装几个工具：apt-**get** install zsh git curl -y

### 两种安装方式

第一种：sh -c "$(curl -fsSL [https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"](https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh)\")

第二种：sh -c "$(wget https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh -O -)"

### zshell 安装



使用lnmp一键安装包，安装lnmp环境

网址：[https://lnmp.org/install.html](https://lnmp.org/install.html" \t "https://segmentfault.com/a/_blank)  
lnmp：Linux Nginx mysql php

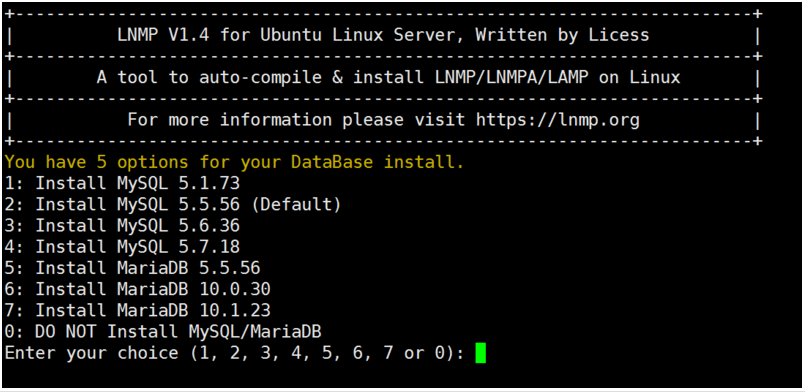
**安装方式**

### 第一步：执行以下命令

wget -c http://soft.vpser.net/lnmp/lnmp1.4.tar.gz && tar zxf lnmp1.4.tar.gz && cd lnmp1.4 && ./install.sh lnmp

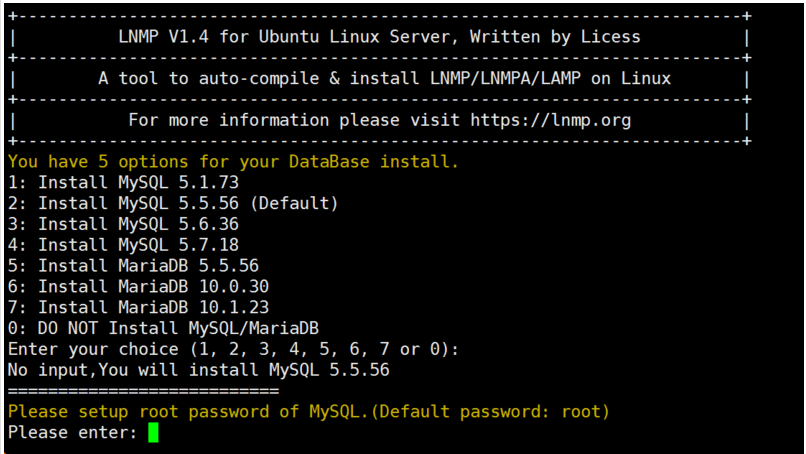
**然后出现以下图片**

第二步：选择mysql版本



直接敲回车选择，第二项就是 选择5.5.56版本

然后出现以下截图

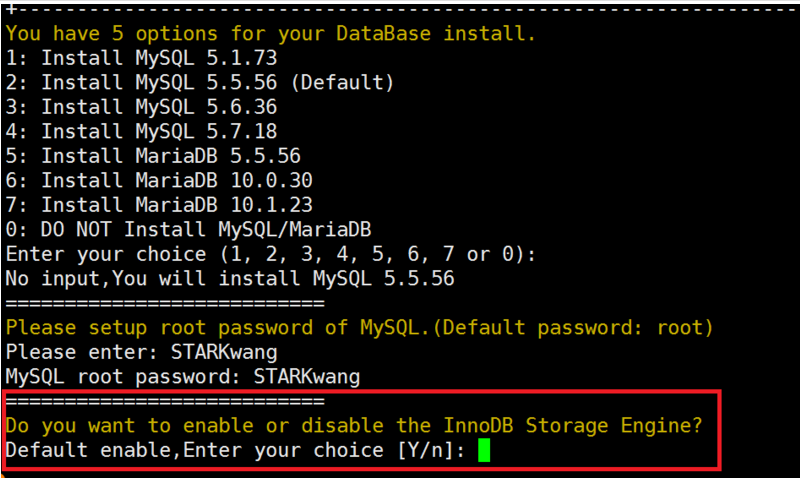


输入mysql密码 输入自己的mysql密码，比较重要，一定要输入一个相对复杂的密码

如果在这里输入错误，请按Ctrl+backspa 键删除

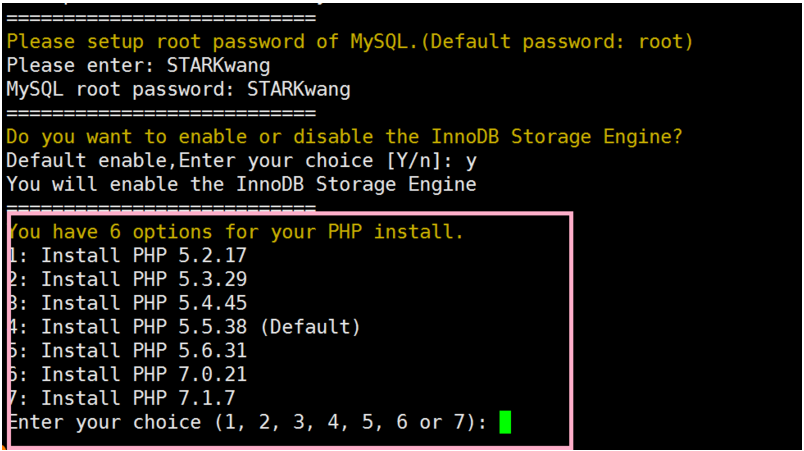
输入完成之后，敲回车

**第三步选择mysql数据库的引擎**



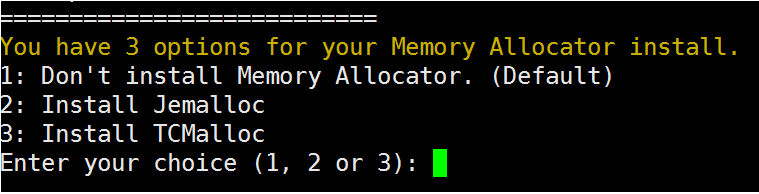
此时输入y 或者敲回车就可以，默认选择 innoDB引擎

### 第四步 选择php版本



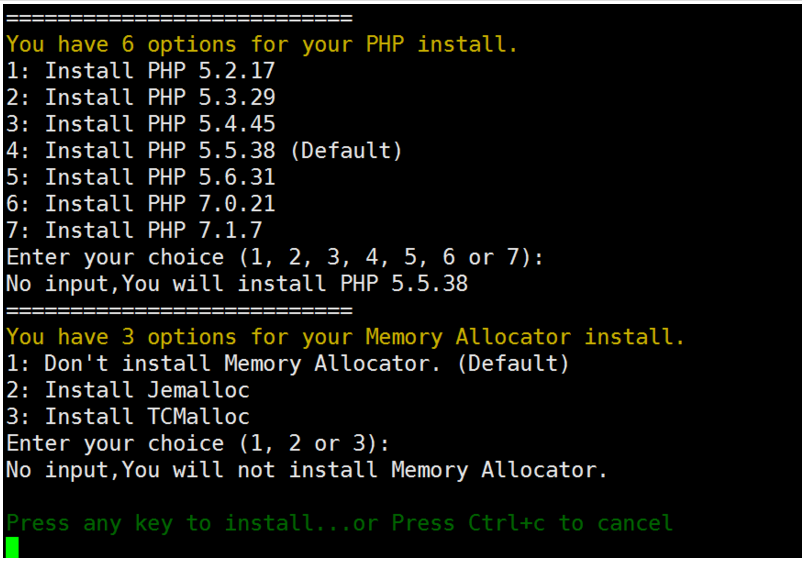
#### 直接敲回车选择默认的php版本 5.5.38

### 第五步： You have 3 options for your Memory Allocator install.



#### 直接敲回车，不安装

### 然后出现



请你按任何键来安装刚才选择的这么多软件环境  
此时正在安装所有软件，大概需要27分钟左右

## 总结

注意点：

选择mysql后，需要自己输入密码  
 其他的一路回车

如果全部回车后，mysql密码是root 账号也是root

## 最终安装的所有信息



### 安装完成lnmp环境

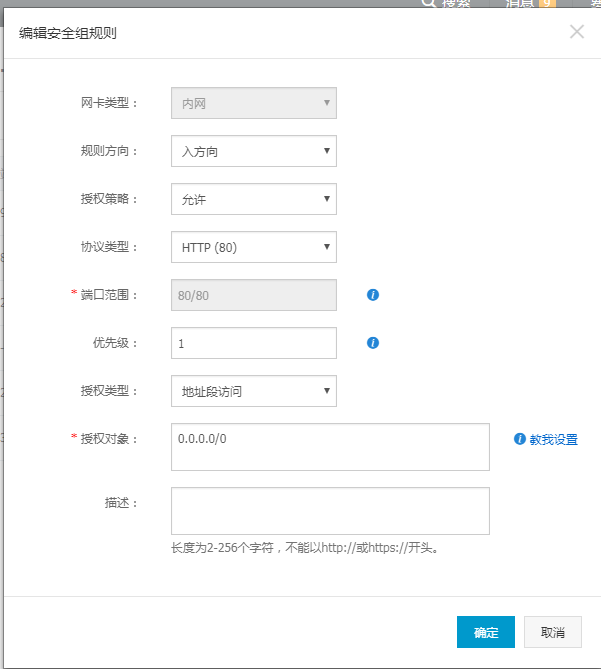


此时就可以使用了

### 使**用方式**

在浏览器里面输入 ip 出现页面证明安装完成

**如果是阿里云的服务器需要配置以下规则**

****

## 安装node环境

### 安装方式

### 采用nvm方式安装管理node版本

参考：<https://github.com/creationix/nvm>

### 安装脚本

### 安装或更新nvm，可以使用以下两种方式

### 第一步安装 nvm

命令：curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.6/install.sh | bash

or Wget:

命令：wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.6/install.sh | bash

### 第二步 配置nvm环境变量

The script clones the nvm repository to ~/.nvm and adds the source line to your profile (~/.bash\_profile, ~/.zshrc, ~/.profile, or ~/.bashrc).

#### 执行以下命令

export NVM\_DIR="$HOME/.nvm"

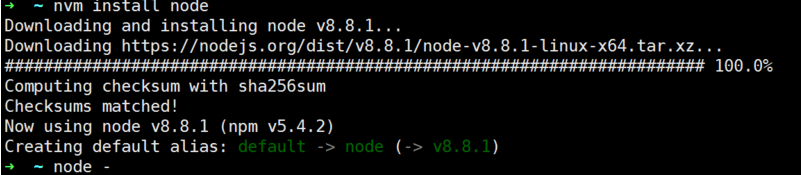
[ -s "$NVM\_DIR/nvm.sh" ] && . "$NVM\_DIR/nvm.sh" # This loads nvm

## 安装nodejs环境

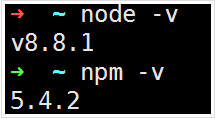
要下载、编译和安装最新版本的节点，请执行以下操作:

nvm install **node**

### 等待安装完成之后



**命令：node** **-v**



### 出现上面的版本号，说明node安装成功

### 下面这些事nvm的一些其他操作，管理node 版本切换版本（了解一下）

或者，您可以在shell中运行任意命令，并使用所需的节点版本

命令：nvm exec 4.2 **node** **--version**

您还可以找到可执行文件所在位置的路径

命令：nvm which 5.0

## 安装mongodb环境

地址：<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

### 第一步：导入公钥

命令：sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6

### 第二步：为mongodb创建一些列表文件

### 根据自己的操作系统执行系统

查看自己的操作系统

命令：cat /etc/issue

#### Ubuntu 12.04

命令：echo "deb [ arch=amd64 ] http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.**list**.d/mongodb-org-3.4.**list**

#### Ubuntu 14.04

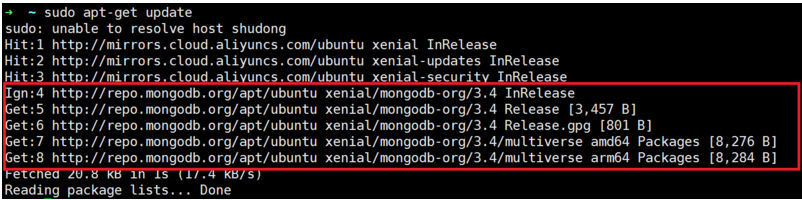
命令：echo "deb [ arch=amd64 ] http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.**list**.d/mongodb-org-3.4.**list**

#### Ubuntu 16.04

命令：echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.**list**.d/mongodb-org-3.4.**list**

### 第三步 更新第二步设置的包

命令：sudo apt-get **update**

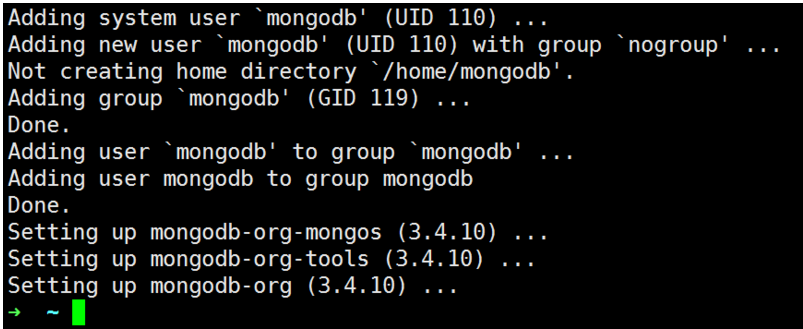


### 第四步 安装mongodb-org

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

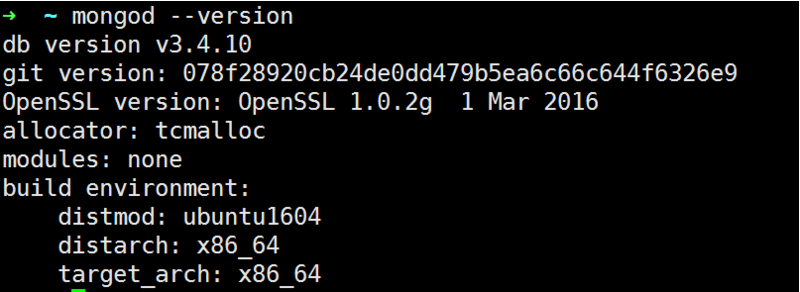
### 安装完成之后

安装成功出现以下图示



### 查看mongodb版本

命令：mongod --version



### 执行 mongo

命令：use shudong

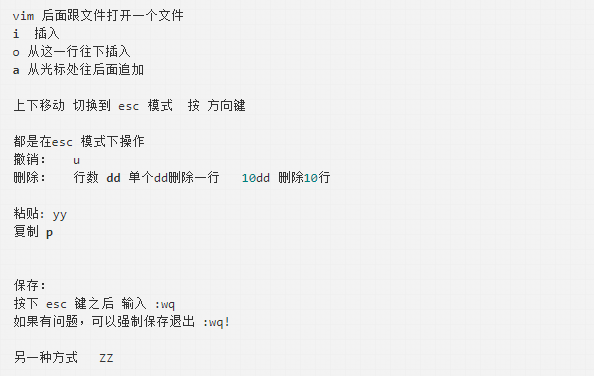
命令：db.stark.insert({'name':'stark','age':18})

命令：db.stark.find({})

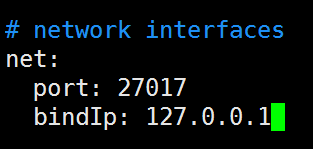
### 配置mongodb远程连接

命令：vim /etc/mongod.conf

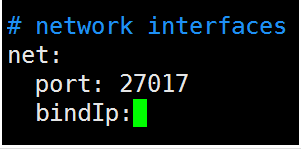
#### vim 操作命令



#### 修改 bandIp 添加你的外网ip



改成



把 ip参数删掉 可以远程访问

#### 然后重启服务

命令：service mongod restart

然后用远程工具 mongodbbooster测试

### 开启mongodb

命令：sudo service mongod start

## 相关问题



需要手动新建/lib/systemd/system/mongod.service文件，并写入下面内容

#### 编辑文件 /lib/systemd/system/mongod.service

命令vim /**lib**/**systemd**/**system**/**mongod**.**service**

#### 把下面文件写入里面

**命令：[Unit]**Description=High-performance, schema-free document-oriented databaseAfter=network.targetDocumentation=https://docs.mongodb.org/manual

**[Service]**User=mongodbGroup=mongodbExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

**[Install]**WantedBy=multi-user.target

**如下图：**



#### 然后执行 （激活mongod service）

命令：sudo systemctl enable mongod

#### 启动、重启和关闭命令

命令：sudo service mongod start

命令：sudo service mongod restart

命令：sudo service mongod stop

## 如果上面没有设置mysql密码

命令：**reset\_mysql\_root\_password**.sh

## 如果以学生的身份买的阿里云主机，或者其他身份，以ssh方式连接不上 阿里云主机

### 解决方案：

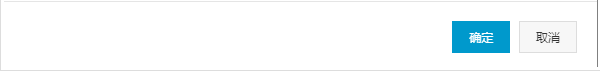


规则或新建规则



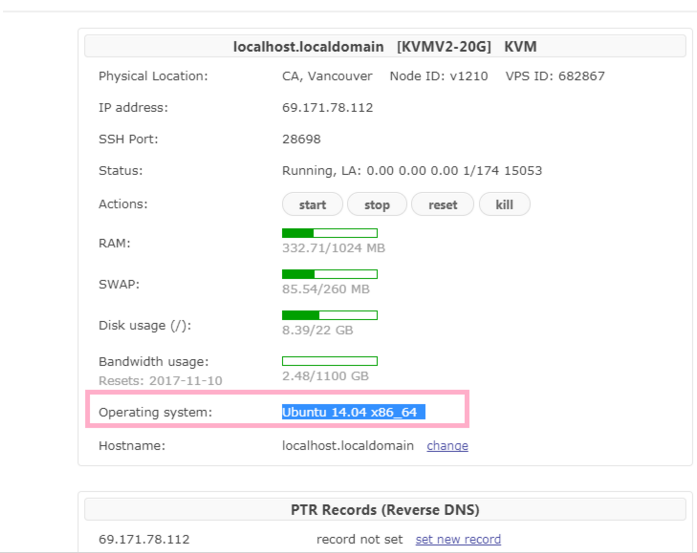






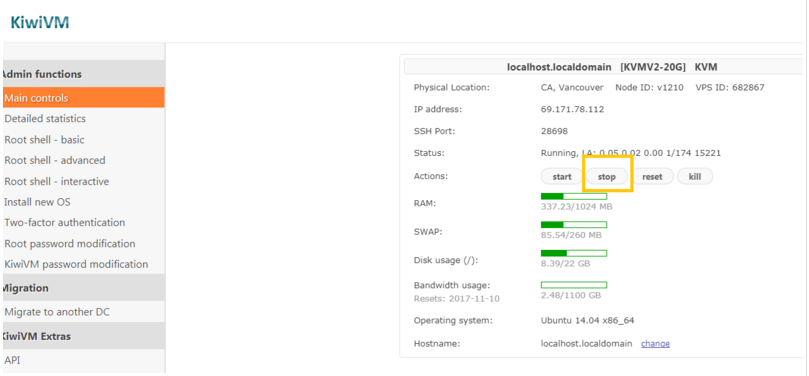
## 更换自己的操作系统

命令：cat /etc/issue

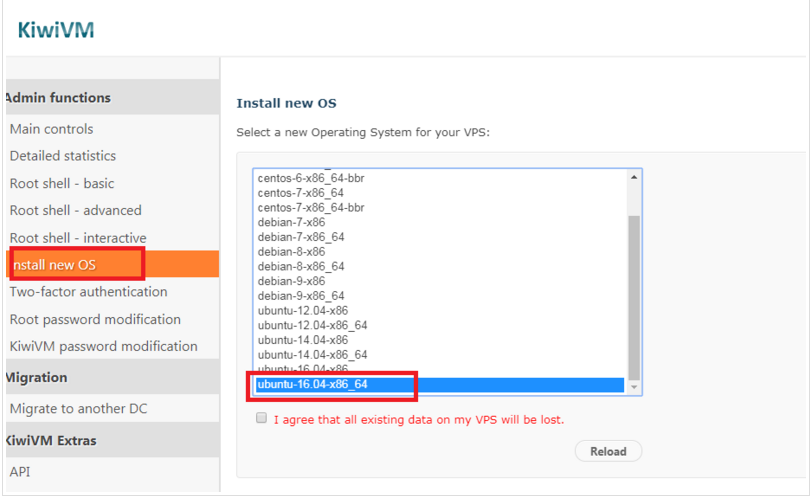


## 更换自己的系统盘

### 第一步先停止



### 第二步选择新的系统



## 阿里云更换系统盘 或初始化磁盘



先停止然后在点击更换系统盘

### 初始化磁盘，并且设置密码



# [04. 怎么在一台主机上面部署多个网站,详细操作指南](https://segmentfault.com/a/1190000011791001)

## 怎么在一台主机上面部署多个网站

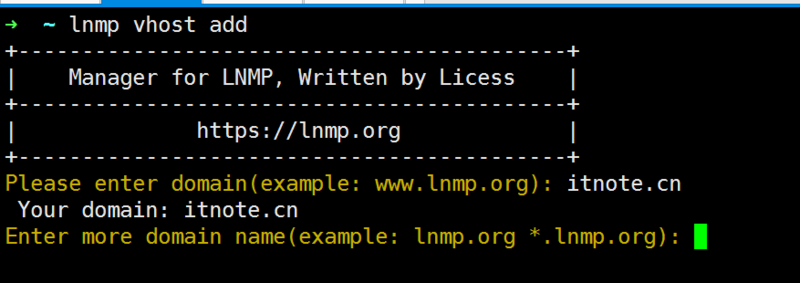
### 使用Nginx的虚拟化配置

## 环境

使用一键安装lnmp环境  
请参考：<https://segmentfault.com/a/1190000011778030#articleHeader31>

### 第一步：执行 lnmp vhost add

命令：lnmp vhost **add**



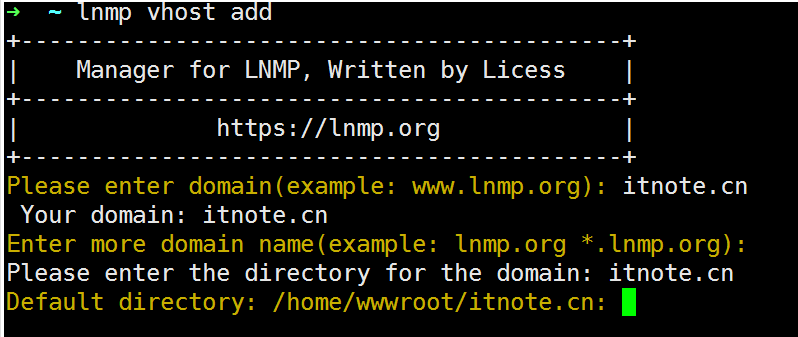
输入自己要绑定的域名，比如我现在需要绑定 itnote.cn 我在这个地方输入，然后敲回车，下面提示让你输入更多需要绑定的域名，如不需要请敲回车，



在这前一定要把自己的域名解析到这台主机，下面图片是绑定自己的主机



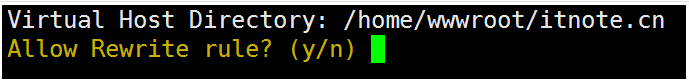
### 第二步：输入自己域名对应的主机文件目录



如果不更改，直接敲回车，会默认给你再 /home/wwwroot/itnote.cn 以你的域名为文件夹名字创建目录  
如果你输入了文件名字，请记住一定要输入全路径

### 第三步：询问你是否要添加静态规则

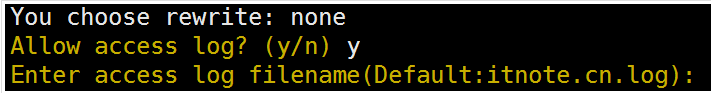
静态规则根据自己的网站程序配置，一般就是url的访问格式



### 第四步：是否保存访问日志

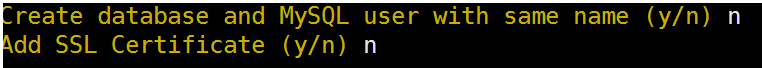


最好保存一下，以便排查错误  
在这里面执行 y 就是需要，然后敲回车，然后会让你输入保存日志的名字，直接敲回车，会自定产生一个名字。

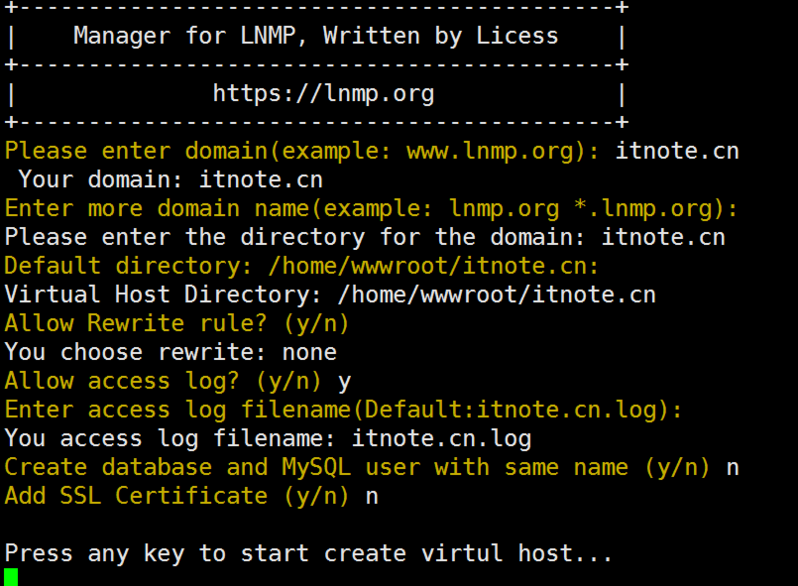


### 第五步：是否创建mysql数据库（不需要）

不需要填写 n 或者敲回车



### end 最后整体截图

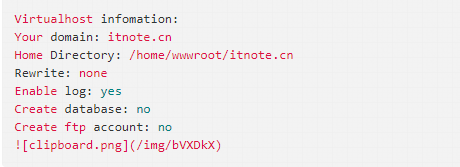


### 总结

域名务必需要自己输入

日志需要选择　ｙ，其它一路回车

**最后创建成功后**



此时就创建了一个虚拟主机，相当于创建一个网站



### 如果想创建多个站点，重复以上过程

记得让你的域名绑定主机哦

### 创建多个站点，列子：stark.itnote.cn

#### 第一步先绑定主机



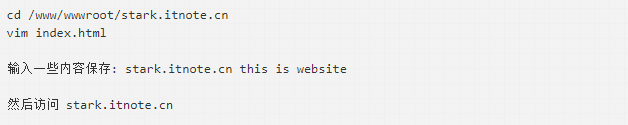
#### 第二步：执行 lnmp vhost add 生成网站需要的Nginx配置文件

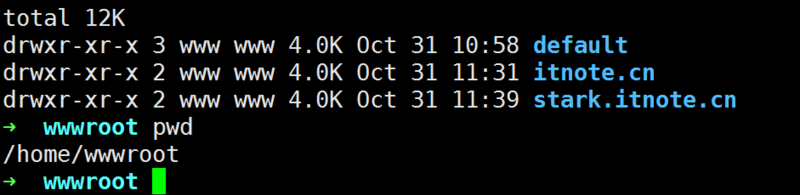


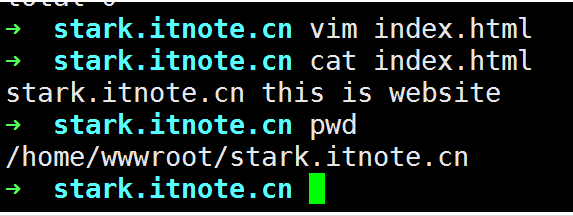
### 最后访问stark.itnote.cn 对应 /www/wwwroot/stark.itnote.cn

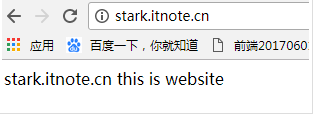
这个里面放置项目文件就是网站 stark.itnote.cn 访问的主机路径地址

### 进入这个文件夹/www/wwwroot/stark.itnote.cn（可省略）









（截止到上一个可省略处可省略）

## 常见问题集锦

修改 /usr/local/nginx/conf/vhost/ 这个路径里面的相应的配置文件  
一般都是以你的域名命名后面加上 .conf  
比如我的配置文件是 itnote.cn.conf

命令：**vim** /usr/local/nginx/**conf**/vhost/itnote.**cn**.**conf**

以上操作都是针对这个配置文件做的修改；

下面是Nginx 配置 （反向代理可以在这添加）

server

{

listen 80;

**#listen [::]:80;**

server\_name itnote.cn ; **# 打算添加多个域名在这后面以空格隔开添加域名**

**index** **index**.html **index**.htm **index**.php **default**.html **default**.htm **default**.php; **# 首访问文件**

root /home/wwwroot/itnote.cn; **# 根路径，就是访问域名后对应的目录**

include none.conf;

**#error\_page 404 /404.html;**

**# Deny access to PHP files in specific directory**

**#location ~ /(wp-content|uploads|wp-includes|images)/.\*\.php$ { deny all; }**

include enable-php.conf;

**location** ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$

{

expires 30d;

}

**location** ~ .\*\.(js|css)?$

{

expires 12h;

}

**location** ~ /.well-known {

allow all;

}

**location** ~ /\.

{

deny all;

}

access\_log /home/wwwlogs/itnote.cn.log;

}

~

如下图：

图1：

图2：

任何时候修改Nginx 都要重启 Nginx

### 重启Nginx方式

#### 第一种方式：

命令：service nginx restart

#### 第二种方式：

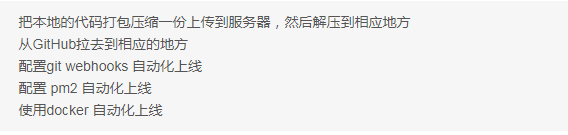
命令：**systemctl** **restart** **nginx**.service

#### 第三种方式：

命令：/etc/init.d/nginx restart

# [05.vue项目上线（webhooks pm2 docker 自动化上线）](https://segmentfault.com/a/1190000011793908)

## 项目上线常用方案：



### 我们以vnshop项目作为案例

<https://github.com/itguide/vnshop>

### 把本地的代码打包压缩一份上传到服务器，然后解压到相应地方



在我们的vue项目里面的client里面执行以下命令

命令：npm **run** build

这个命令用来把vue项目编译成生产环境需要的文件

生成一个dist 文件夹，把这个文件夹，压缩成zip格式的文件

然后通过xshell 上传到服务器

#### 第一步：安装 lrzsz

命令：apt-**get** install lrzsz -y

sz：将选定的文件发送到本地机器  
rz：运行该命令会弹出一个文件选择窗口，从本地选择文件上传到Linux服务器  
rz，sz是Linux/Unix同Windows进行ZModem文件传输的命令行工具，

#### 第二步：把本地的压缩打包的项目文件通过rz命令上传到服务器

在 /home/wwwroot/ 目录里面执行

命令：cd /home/wwwroot/

执行rz命令，然后跳出一个框，让你选择本地需要上传到服务器的文件。

命令：rz

在服务器哪执行的命令，就上传到哪去

#### 第三步：把上传后的项目，解压复制到网站访问的文件夹里面

#### 添加一个网站，虚拟主机

命令：lnmp vhost **add**

添加网详细请看：<https://segmentfault.com/a/1190000011791001>

把上传的压缩包解压：

命令：**unzip** **dist**.zip

解压完之后，生成一个 dist文件夹  
需要把这个文件夹里面的所有文件复制到 vn.itnote.cn文件夹里面去，执行以下命令

命令：**cp** -r dist/\* **vn**.itnote.**cn**

cp 是拷贝 -r 是深度拷贝，如果里面是文件，需要用 -r /\* 代表把dist 文件夹里面的所有文件都拷贝 后面的vn.itnote.cn 文件夹，是要拷贝的目的地

### 从GitHub拉取项目到相应的地方

### 第一步：域名绑定到这个主机，并且创建一个虚拟主机



命令：lnmp vhost **add**

创建详细过程请看 https://segmentfault.com/a/1190000011791001

第二步：把项目拉取到主机

命令：cd /home/wwwroot

命令：git clone https://github.com/itguide/vnshop10.git vnshop

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

命令：cd /home/wwwroot/vnshop/client

命令：cnpm i

命令：npm run build

#### 第三步：修改Nginx 配置

修改配置

别傻乎乎的复制，后面要修改自己需要配置的文件

命令：vim /usr/local/nginx/conf/vhost/vx.itnote.cn.conf



命令：

**server**

{

listen 80;

#listen [::]:80;

server\_name vx.itnote.cn ;

index index.html index.htm index.php default.html default.htm default.php;

root /home/wwwroot/vnshop/client/dist/;

include none.conf;

#error\_page 404 /404.html;

# Deny access to PHP files in specific directory

#location ~ /(wp-content|uploads|wp-includes|images)/.\*\.php$ { deny all; }

include enable-php.conf;

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$

{

expires 30d;

}

location ~ .\*\.(js|css)?$

{

expires 12h;

}

location ~ /.well-known {

allow all;

}

location ~ /\.

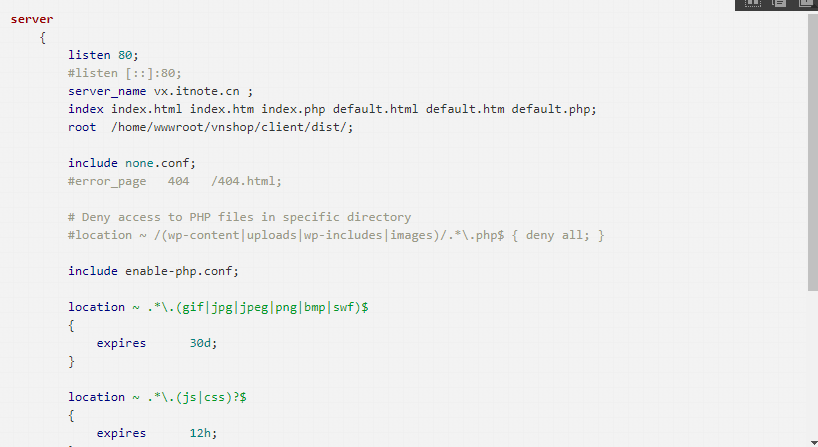
{

deny all;

}

access\_log /home/wwwlogs/vx.itnote.cn.log;

}





#### 记得重启Nginx 配置哦

重启命令

命令：/etc/init.d/nginx restart

# [06.nodejs服务上线（生产环境前后分离的vue项目中怎么解决跨域问题）](https://segmentfault.com/a/1190000011796903)

以下操作使用下面项目为案例（https://github.com/itguide/vnshop）

启动node服务

### 克隆好项目后记得把依赖包安装好

### 使用 node 启动node服务

命令：cd /home/wwwroot/vnshop/server

命令：npm i

命令：npm **run** start

### 使用pm2方式启动node 服务

命令：npm i pm2 -g

命令：cd /home/wwwroot/vnshop/server

命令：pm2 start ./bin/www

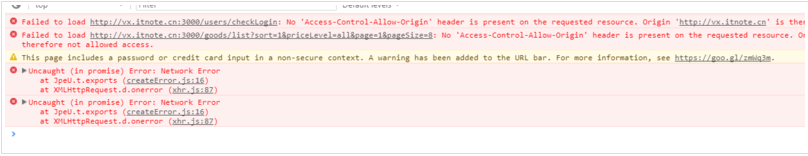
### 测试

http://vx.itnote.**cn**:3000/goods/**list**?**sort**=1&priceLevel=**all**&page=1&pageSize=8

如果返回数据，说明node 启动正常

## 如果单纯访问3000端口会请求出api数据来

## 但是在vue项目里面请求，会产生跨域



### 在本地开发解决跨域问题

后端node服务，启动后是3000端口,在前台vue项目中访问，会产生跨域，在本地开发中我们可以配置代理来解决

下面这个是详细解决方案

<https://segmentfault.com/a/1190000011715088>

## 在线上生产环境前后分离的vue项目中怎么解决跨域问题

### 我们在项目中配置好生产环境和开发环境的访问api地址

在 client/config/api.config.js  
api.config 配置（vim进入文件。I编辑修改，Esc+：+wq=>保存退出）

//判断是开发模式还是本地模式，其实不需要这么麻烦 直接

pr命令：

**const** isPro = Object.is(

ocess.env.NODE\_ENV, 'production')

**module**.exports = {

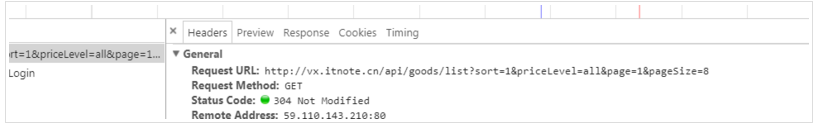
baseUrl: isPro ? 'http://vx.itnote.cn/api/' : 'api/'

}

注意：

[http://vx.itnote.cn/api/](http://vx.itnote.cn/api/" \t "https://segmentfault.com/a/_blank) 这个地址是你自己的服务，能访问的服务  
每次修改这个配置，需要去编译 npm run build

### 如果是线上环境 则会访问 [http://vx.itnote.cn/api/](http://vx.itnote.cn/api/" \t "https://segmentfault.com/a/_blank)



### 在Nginx配置里面修改添加反向代理

每次vue项目请求以 /api/开头的路由自动转换成 3000端口的服务

nginx 配置 修改

**命令：vim** /usr/local/nginx/**conf**/vhost/vx.itnote.**cn**.**conf**

**命令：**

**location** **/api**/ {

proxy\_pass http://127.0.0.1:3000/; # 当访问v1的时候默认转发到 3000端口

}

如下图：



整体nginx 配置

**server**

{

listen 80;

#listen [::]:80;

server\_name vx.itnote.cn ;

index index.html index.htm index.php default.html default.htm default.php;

root /home/wwwroot/vnshop/client/dist/;

include none.conf;

#error\_page 404 /404.html;

# Deny access to PHP files in specific directory

#location ~ /(wp-content|uploads|wp-includes|images)/.\*\.php$ { deny all; }

include enable-php.conf;

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$

{

expires 30d;

}

location /api/ {

proxy\_pass http://127.0.0.1:3000/; # 当访问api的时候默认转发到 3000端口

}

location ~ .\*\.(js|css)?$

{

expires 12h;

}

location ~ /.well-known {

allow all;

}

location ~ /\.

{

deny all;

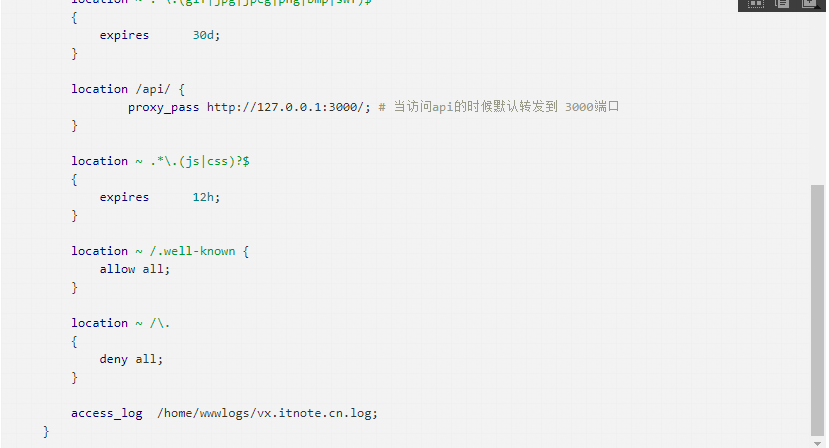
}

access\_log /home/wwwlogs/vx.itnote.cn.log;

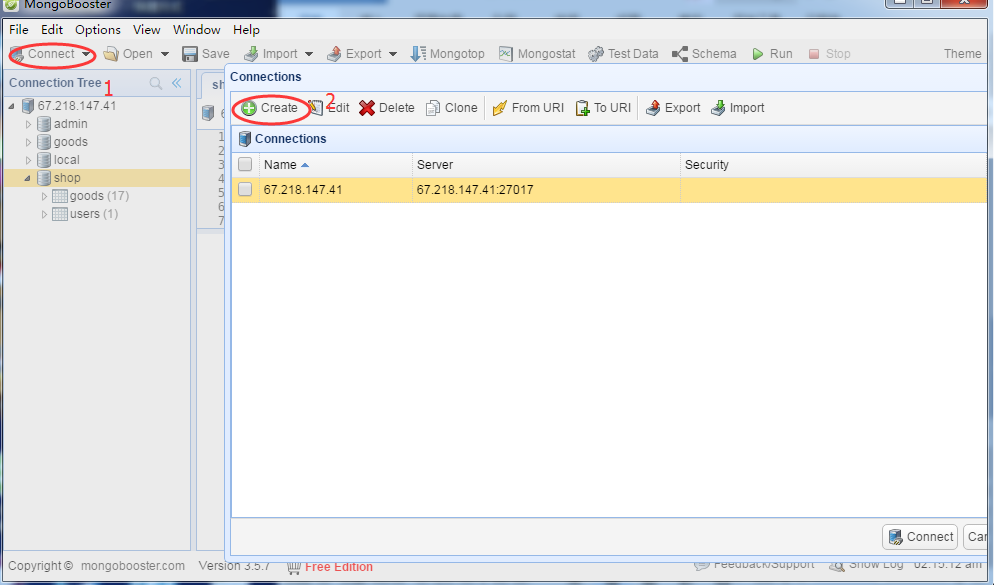
}

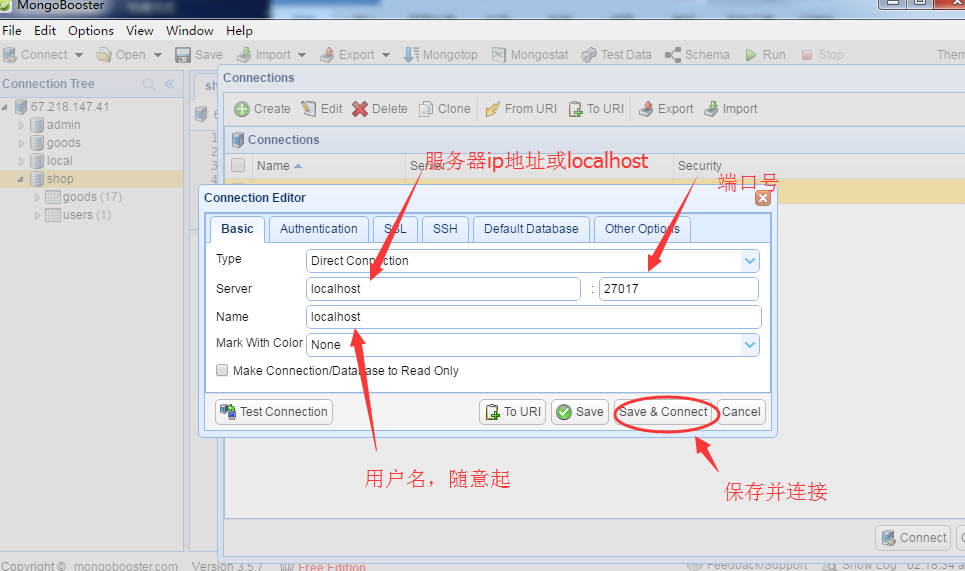
如下图：





Mongobooster操作：





1. 点击服务器67.218.147.41创建数据库ctreate database
2. Import collections 导入数据并修改表名与数据库对应
3. Export collections 导出数据

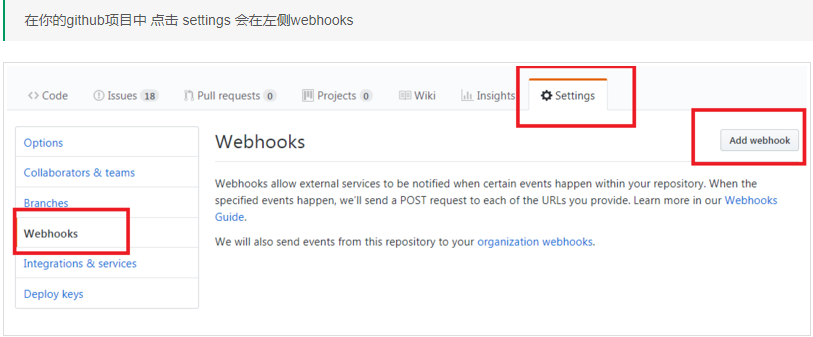
重启下Nginx配置

重启命令：/etc/init.d/nginx restart;

## 使用git webhooks 完美解决这个问题

## webhooks 概念

### 打开方式

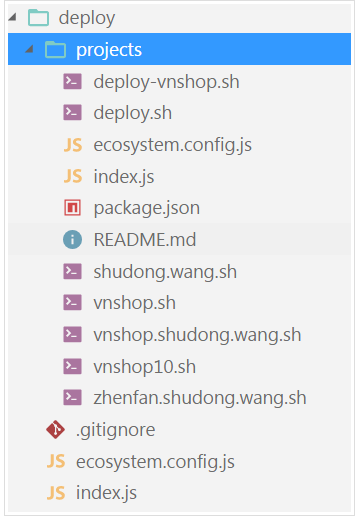


hook 钩子，想要在git某个生命周期触发一个事件，就是想让程序在我们触发一个git事件的时候，做一些事情。  
举个列子：当我们把代码 git push origin master 的时候，push成功以后，希望服务器自动把代码拉取，更新，做一些自己希望做的事情。

下面正式开始：

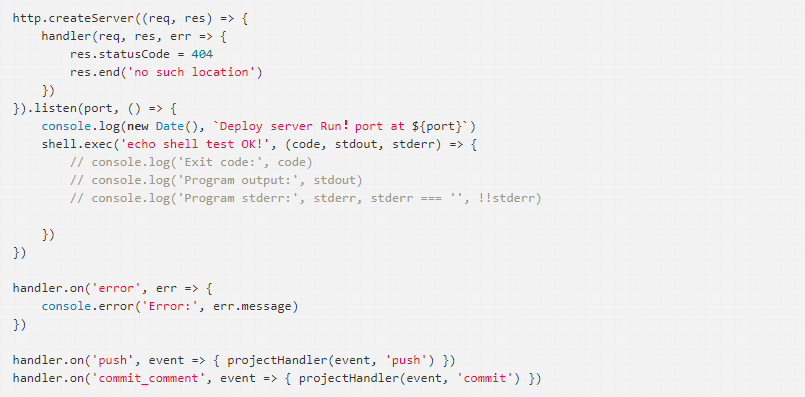
### 把deploy的项目克隆下拉

命令行：git **clone** **https**://github.com/itguide/deploy-vnshop.git deploy

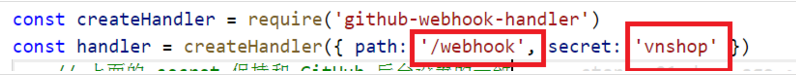


修改projects下的index.js





#### 以上代码重点修改



path：和git webhooks的payload 保持一致 记得前面加上完整url路径 比如： [http://vx.itnote.cn](http://vx.itnote.cn/" \t "https://segmentfault.com/a/_blank):9988/webhook

相当于一个api，每次我们往github 提交事件的时候，github 的webhooks 根据我们设定的事件，然后像这个url提交一个post请求，然后服务器就会根据触发的请求，做一些事情。

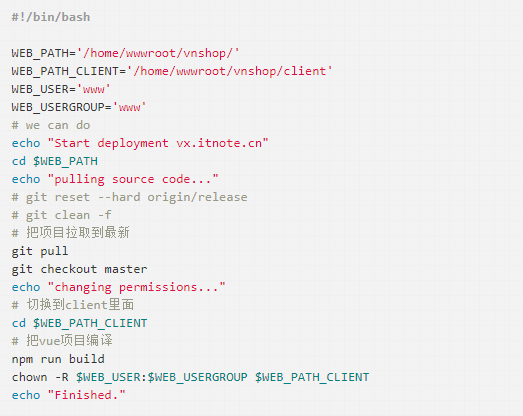
secret：和git webhooks的secret 保持一致

#### 触发的shell 脚本，根据自己的项目业务，编写自己的shell



我的项目是 vnshop10 所以这个shell文件名字是 vnshop10  
需要添加的文件





#### 以上代码修改完后推送到自己 的仓库

命令：

1. git diff //查看版本差异
2. git add . //提交到暂缓区
3. git status //查看状态
4. git remote -v //查看远程仓库地址
5. git remote remove origin //删除远程仓库地址
6. git remote add origin //你的仓库地址
7. git commit -m “提交说明”
8. git push -u origin master //推送到远程仓库

#### 然后去服务器自己把刚才推送的这个项目克隆下拉



命令：

Cd /home/wwwroot //进入文件夹

Git clone 你自己的仓库的地址

Cd /home/wwwroot/你仓库的名字文件

Cnpm i //安装依赖包

**然后启动这个deploy项目**

cd /home/wwwroot/deploy-vnshop

使用pm2 启动，需要提前安装 pm2 npm **i** -g pm2

pm2 start index.js --name deploy --watch -**i** max -e ./logs/deploy/error.log -o ./logs/deploy/out.log

如下图：



#### 测试接口启动是否成功



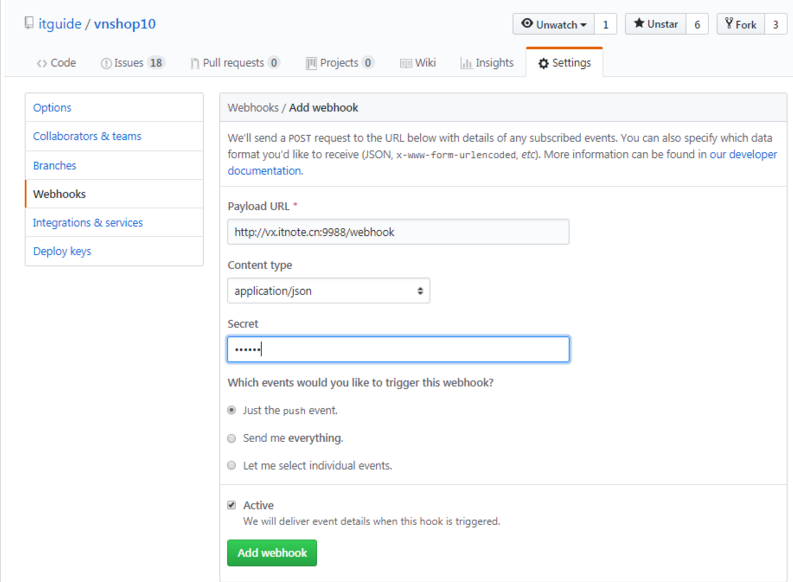
对应命令：netstat -anp | grep 9988

可以使用浏览器打开

http://vx.itnote.cn:9988/webhook

如果浏览器不可以访问，是阿里云主机的话，需要配置安全组规则

### 去填写git webhooks 这些配置



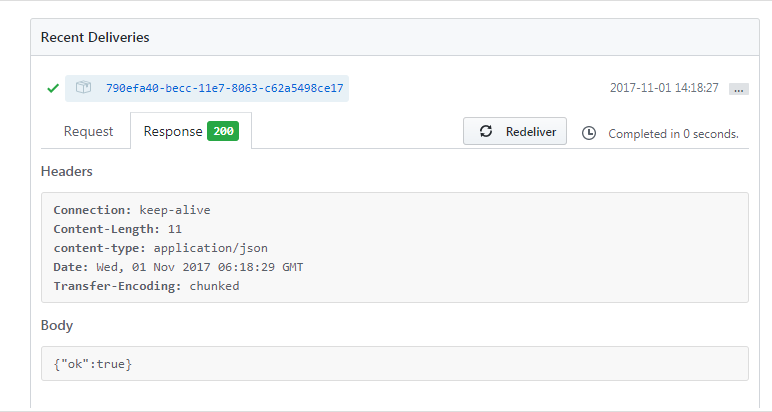
上面的信息，都要和自己填写的保持一致

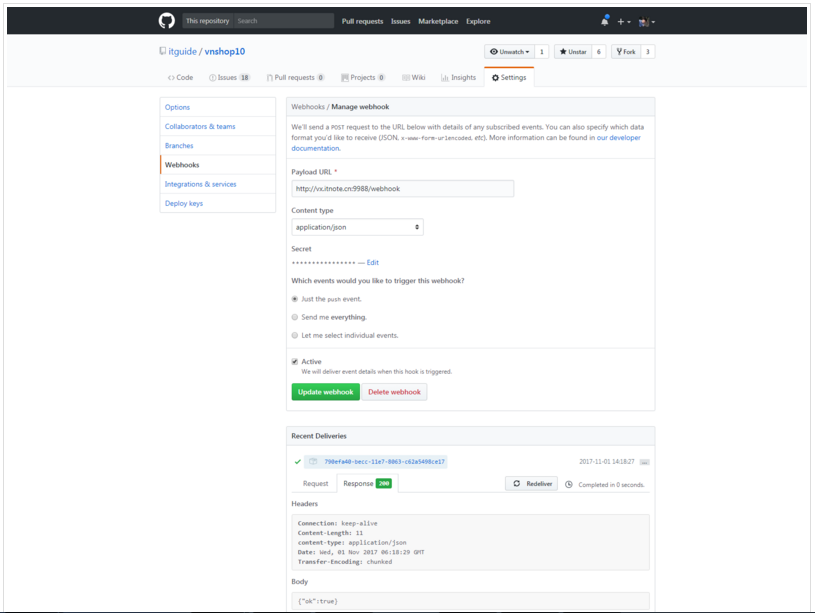
### 填写好点击 add webhook 按钮

### 测试git webhook是否配置成功

生成一个 webhook 点击进去

出现以下情况，表示配置成功





注意：在vue项目仓库配置webhook，而不是deploy项目；

创建关闭服务器，永久连接服务方式：

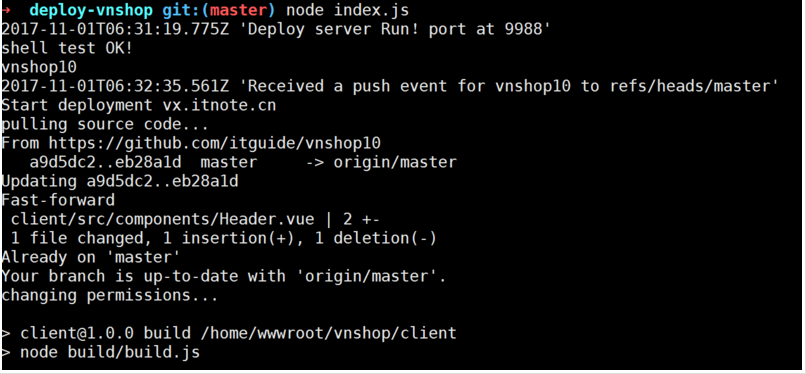
方法1：

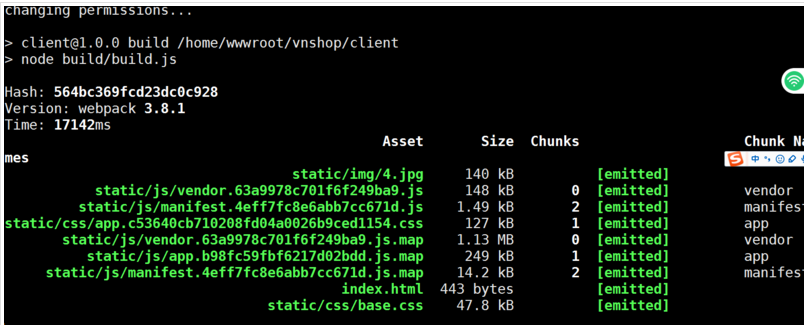
在deploy目录下执行 pm2 start index.js（index.js在deploy目录下）

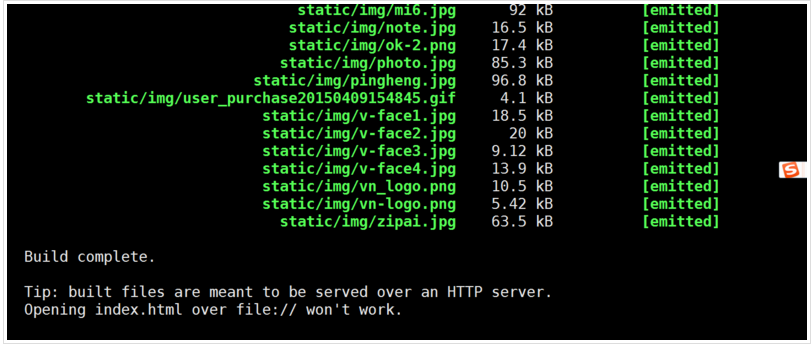
方法2：



### 测试整个流程是否ok，本地编写代码，push到仓库，检查线上代码是否发生改变。







### 大功告成，如果不能帮助到你，请留言

Pm2介绍：

pm2 是一个带有负载均衡功能的Node应用的进程管理器.  
当你要把你的独立代码利用全部的服务器上的所有CPU，并保证进程永远都活着，0秒的重载， PM2是完美的。它非常适合IaaS结构，但不要把它用于PaaS方案（随后将开发Paas的解决方案）

特征：

内建负载均衡（使用Node cluster 集群模块）  
后台运行  
0秒停机重载，我理解大概意思是维护升级的时候不需要停机.  
具有Ubuntu和CentOS 的启动脚本  
停止不稳定的进程（避免无限循环）  
控制台检测  
提供 HTTP API  
远程控制和实时的接口API ( Nodejs 模块,允许和PM2进程管理器交互 )

git webhooks介绍：

1. 在Git服务器上创建一个普通git用户（无需密码），如server-115.159.10.10，然后把这个用户加入到目标项目里的成员里，只需要有pull权限即可，飘易一般设置为Reporter。
2. 在生产服务器上为www用户生成 SSH 公钥，路径一般在 /home/www/.ssh/，把生成的公钥 id\_rsa.pub 复制粘贴到Git服务器上第一步里的普通用户的ssh公钥里



1. 【**可选**】切换用户身份从root变成www，注意要把www的默认shell从**/sbin/nologin**改成**/bin/bash**，不然无法cd和git pull等；注：www用户没有密码，而Linux的pam验证拒绝没有密码的用户登录系统，所以我们不用担心www用户可以直接登录系统。



1. 首次从git服务器上抓取资源用clone，指定目录需要为空





1. 先cd到项目根目录，后续再抓取直接pull



1. 网站根目录下放置 git-webhook-handler.php 自动部署文件，然后Git服务器的对应项目的 webhooks 配置上该文件的网址（Push events），需启用php的shell\_exec函数
2. 以下参考链接：

<http://www.piaoyi.org/linux/Gitlab-GitHub-webhook-push-auto-deployment.html>

Docker介绍：

Docker 是一个[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/246339" \t "https://baike.baidu.com/item/Docker/_blank)的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 [Linux](https://baike.baidu.com/item/Linux" \t "https://baike.baidu.com/item/Docker/_blank) 机器上，也可以实现[虚拟化](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E5%8C%96" \t "https://baike.baidu.com/item/Docker/_blank)。容器是完全使用[沙箱](https://baike.baidu.com/item/%E6%B2%99%E7%AE%B1/393318" \t "https://baike.baidu.com/item/Docker/_blank)机制，相互之间不会有任何接口

组成结构：

dockerClient客户端

Docker Daemon守护进程

Docker Image镜像

DockerContainer容器

Mongobooster介绍

mongobooster是一款运用MongoDB数据库可视化操作的辅助工具，在MongoDB数据库中，开发数据和利用数据库用到可视化的地方很多，所谓的可视化就是可以帮助用户使用程序的时候更加清晰，以图形的形式显示数据的每一个操作步骤，在您在研究数据、数据处理、决策分析等方面得到巨大的效率提升；mongobooster拥有多种设计工具，在可视化方面的运用非常广泛，不仅是针对MongoDB数据库而已，其连接的方式也非常简单，让您可以轻松访问MongoDB服务器，实现数据可视化操作！