# 虚拟机搭建 及环境配置

Centos7 64位 镜像

Supervisor nginx gunicorn gevent

1.下载VMware，32位系统只能支持11版本一下的VMware

2.下载centos镜像

安装VMware ，使用管理员运行，可以在非C盘安装，稍后安装系统，

一直默认就可以，选择配置时，使用centos镜像。

3.安装wget 、vim、lrzsz、等

使用yum安装：类似npm的东西

首先安装wget ：yum –y install wget

## 更新python到2.7.13

进入root： 执行 su root 输入密码

1.下载python包

[root@amio ~]# cd /home/ # 存放下载包的路径

[root@amio home]# wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tgz # wget 后接python官网对应的链接

2.解压 配置 编译 安装

[root@amio home]# tar -zxvf Python-2.7.13.tgz # 解压命令

[root@amio home]# ll # 解压后在当前目录生成Python-2.7.13的目录

安装gcc（编译时会依赖）

[root@amio home]*# yum install gcc\* openssl openssl-devel ncurses-devel.x86\_64 bzip2-devel sqlite-devel python-devel zlib*

配置、编译、安装

[root@amio home]*# cd Python-2.7.13*

[root@amio Python-2.7.13]*# (sudo) ./configure --prefix=/usr/local # [配置]指定可执行文件、库文件、配置文件、资源文件的安装路径。若没有权限加sudo*

[root@amio Python-2.7.13]*# (sudo) make # 编译*

[root@amio Python-2.7.13]*# make altinstall # 不要使用make install，否则会覆盖系统自带python*

## 安装后检查

发现版本还是python2.7.5

[root@amio ~]# ll -l /usr/bin/python\* # 系统自带的

lrwxrwxrwx 1 root root 7 3月 29 22:44 /usr/bin/python -> python2

lrwxrwxrwx 1 root root 9 3月 29 22:44 /usr/bin/python2 -> python2.7

-rwxr-xr-x 1 root root 7136 11月 6 00:29 /usr/bin/python2.7

-rwxr-xr-x 1 root root 1835 11月 6 00:29 /usr/bin/python2.7-config

lrwxrwxrwx 1 root root 16 4月 2 03:27 /usr/bin/python2-config -> python2.7-config lrwxrwxrwx 1 root root 14 4月 2 03:27 /usr/bin/python-config -> python2-config

[root@amio ~]# ll -l /usr/local/bin/python\* # 手工安装的

-rwxr-xr-x 1 root root 8257136 4月 2 04:48 /usr/local/bin/python2.7

-rwxr-xr-x 1 root root 1687 4月 2 04:49 /usr/local/bin/python2.7-config

## 备份旧版本，连接新版本

[root@amio ~]*# mv /usr/bin/python /usr/bin/python2.7.5*

[root@amio ~]*# ll -l /usr/bin/python\**

lrwxrwxrwx 1 root root 9 3月 29 22:44 /usr/bin/python2 -> python2.7

-rwxr-xr-x 1 root root 7136 11月 6 00:29 /usr/bin/python2.7

lrwxrwxrwx 1 root root 7 3月 29 22:44 /usr/bin/python2.7.5 -> python2 # 改为2.7.5 -rwxr-xr-x 1 root root 1835 11月 6 00:29 /usr/bin/python2.7-config

lrwxrwxrwx 1 root root 16 4月 2 03:27 /usr/bin/python2-config -> python2.7-config lrwxrwxrwx 1 root root 14 4月 2 03:27 /usr/bin/python-config -> python2-config

[root@amio ~]# ln -s /usr/local/bin/python2.7 /usr/bin/python # 增加连接

[root@amio ~]# ll -l /usr/bin/python\* lrwxrwxrwx 1 root root 24 4月 2 05:08 /usr/bin/python -> /usr/local/bin/python2.7 # 新增的，并指向新安装的python

lrwxrwxrwx 1 root root 9 3月 29 22:44 /usr/bin/python2 -> python2.7

-rwxr-xr-x 1 root root 7136 11月 6 00:29 /usr/bin/python2.7

lrwxrwxrwx 1 root root 7 3月 29 22:44 /usr/bin/python2.7.5 -> python2

-rwxr-xr-x 1 root root 1835 11月 6 00:29 /usr/bin/python2.7-config

lrwxrwxrwx 1 root root 16 4月 2 03:27 /usr/bin/python2-config -> python2.7-config lrwxrwxrwx 1 root root 14 4月 2 03:27 /usr/bin/python-config -> python2-config

python升级教程：<https://www.jianshu.com/p/fad3942fc0ed>

安装pip

yum –y install python-pip

将yum文件的python改为python2.7.5

启动nginx

nginx/sbin/nginx –c nginx/conf/nginx.conf

linux下nginx配置SSL证书

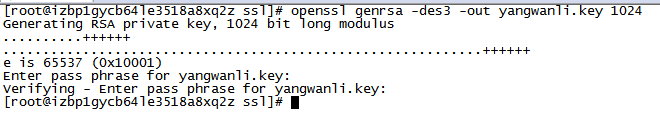
**1.新建证书存放路径(/usr/local/nginx目录下)**

mkdir ssl

**2.生成一个RSA私钥(/usr/local/nginx/ssl目录下)**

openssl genrsa -des3 -out yangwanli.key 1024               --其中 yangwanli可以自定义

输入密码后，再次重复输入确认密码。记住此密码，后面会用到

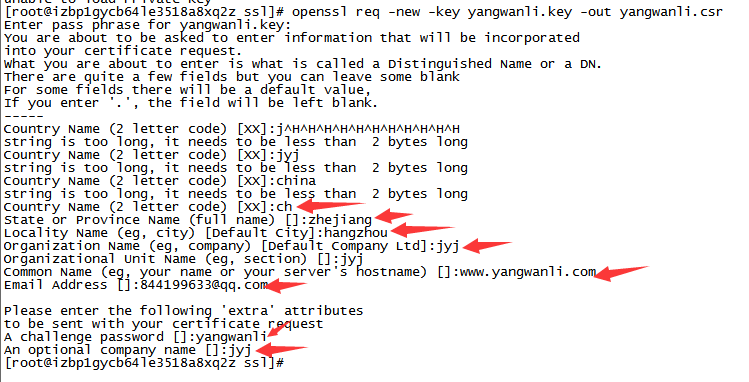


**3.创建csr证书**

openssl req -new -key yangwanli.key -out yangwanli.csr     --其中 yangwanli.csr可以自定义命名

按照命令依次输入内容

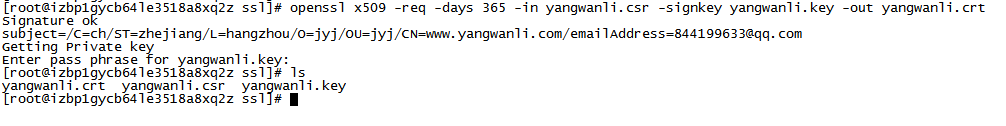
输入私钥密码------->国家(zh)------>省份(浙江)----->城市(杭州)------>公司名称------->组织-------->公司域名-------->邮箱-------->私钥密码



执行完后会在ssl目录下会有两个文件，一个是私钥(key)，一个是证书

yangwanli.csr  yangwanli.key

**4.生成crt证书**

openssl x509 -req -days 365 -in yangwanli.csr -signkey yangwanli.key -out yangwanli.crt  


已生成成功

**5.配置证书**

修改nginx.conf文件

 server {  
        listen       443 ssl;  
        server\_name    www.lee.com;  
      
        ssl\_certificate      /usr/local/nginx/ssl/yangwanli.crt;  
        ssl\_certificate\_key  /usr/local/nginx/ssl/lee.key;  
      
        ssl\_session\_cache    shared:SSL:1m;  
        ssl\_session\_timeout  5m;  
      
        ssl\_ciphers  HIGH:!aNULL:!MD5;  
        ssl\_prefer\_server\_ciphers  on;

**6.设置重定向**

rewrite ^(.\*) https://www.yangwanli.com$1 permanent; #可以把http协议重定向到https上面  
**7. 重启nginx**  
service nginx restart  重启

启动nginx nginx目录 /usr/local/nginx

下 sbin/nginx –c conf/nginx.conf

使用supervisor+gunicorn+gevent

在/etc目录下有个 supervisord.conf文件和supervisor.d文件夹

启动 supervisord –c /etc/supervisord.conf

在更换python版本后要记得改变所有路径

python –m pip install

项目配置文件中 如果被引用了 项目相对路径会发生改变 与引入文件同一级

Xshell连接虚拟机 ip

在vmware中点击左上角 编辑>虚拟网络编辑器 选中目标虚拟机 查看窗口底部IP

在虚拟机中输入 ip addr 查看 ip最后一位

192.168.204.128

# [centos7 cannot find a valid baseurl for repo](https://www.cnblogs.com/phpandmysql/p/7773063.html)

出现这个问题是因为yum在安装包的过程中，虽然已经联网，但是没法解析远程包管理库对应的域名，所以我们只需要在网络配置中添加上DNS对应的ip地址即可。

1. **打开网络配置文件**



vi /etc/sysconfig/network-scripts/ifcfg-eth0

**2.在文件末尾追加DNS**

DNS1=8.8.8.8

DNS2=4.2.2.2

把onboot=no 改为yes

1. **重启网络**

ifup eth0（自己的网卡名）

server {

listen 443 ssl;

server\_name localhost;

ssl\_certificate /usr/local/nginx/ssl/hongzeyu.crt;

ssl\_certificate\_key /usr/local/nginx/ssl/hongzeyu.key;

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html/build;

index index.html index.htm;

}

location /api {

add\_header 'Access-Control-Allow-Origin' \*;

add\_header 'Access-Control-Allow-Credentials' 'true';

add\_header 'Access-Control-Max-Age' 86400;

add\_header 'Access-Control-Allow-Methods' 'GET, POST, DELETE, PUT, OPTION';

add\_header 'Access-Control-Allow-Headers' 'token, host, x-real-ip, x-forwarded-ip, accept, content-type';

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header REMOTE-HOST $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_pass http://localhost:9002;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

Docker 端口映射修改

https://blog.csdn.net/u011241780/article/details/79457876