使用docker，通过25,3389,8080三个端口进行网站的部署

* **docker:** 使用宝塔os进行网站的部署与管理，但是宝塔os会对系统文件进行修改冲突，因此在docker容器中安装宝塔os
* **三个端口:** 其中25端口进行后续邮件通讯，3389用于宝塔os的管理入口，8080作为网站的访问端口
* 后续增补3306端口用于外部python访问容器内部mysql！

在部署前需要完全删除之前安装的宝塔os留下的痕迹，可以使用官方的卸载脚本进行卸载，停止docker容器并删除，否则会产生Nginx的各种问题

## 网站的初次部署

1. **确保当前防火墙正在运行且端口已开放**

开放端口号 25,3389,8080，3306

1. **创建docker容器记得宝塔要用到的端口号全部映射从映射过去：**

docker run -i -t -d -p 25:25 -p 3306:3306 -p 3389:3389 -p 8080:8080 --privileged=true -v /root/www:/www centos

**2. 查看容器id**

docker ps

**3. 进入容器**

docker exec -it xxxx /bin/bash

更换docker的DNS防止容器内部连不上网https://blog.csdn.net/weixin\_44861320/article/details/108753530 服务器重启有几率造成/etc/docker/daemon.json丢失？（遇到过几次，但是再次重启确认的时候却不再丢失）

**4.安装必要的软件。**

由于docker中是一个纯净版本，我们首先需要给他升级并且安装必要的软件。

yum check-update -y && yum update -y && yum install initscripts screen wget -y

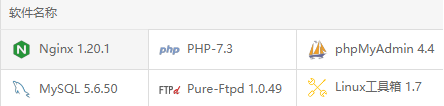
**5.安装宝塔os**

yum install -y wget && wget -O install.sh http://download.bt.cn/install/install\_6.0.sh && sh install.sh

**6.宝塔os修改面板端口为3389**

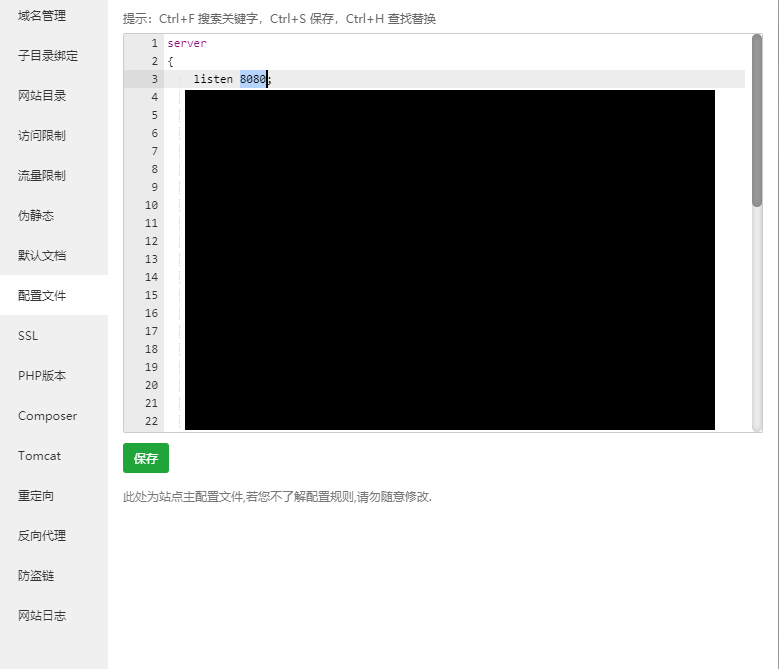
bt 8 (默认端口为888)

**7.宝塔面板安装必要软件（LNMP+PHP）**

纯净的宝塔os在第一次打开时会推荐安装LNMP，如果没有出现，在软件商店里，安装如下软件（首先是linux工具箱）

**8.建立网站并修改端口**

网站会默认使用80端口，在建立完成后，配置文件后，修改配置文件，改80端口为8080



网址后加:8080（www.xxx.com:8080）测试是否成功

**9.修改phpmyadmin端口为8080**

**phpmyadmin**默认使用888端口，修改端口为8080，且修改权限为所有人可通过密码访问。

## 服务器重启后网站的开启

**1.重启服务器**

reboot

**2.查看防火墙策略状态（firewalld在运行，iptables却要手动打开？）**

service iptables status

开启防火墙

service iptables start

**3.重启docker（否则启动容器时端口报错）**

systemctl restart docker

查看全部容器

docker container ls -a

启动相应容器

docker container start xxxxx

查看当前docker容器

docker ps

**4. 进入容器**

docker exec -it xxxxx /bin/bash

**5.重启宝塔os**

bt 1

进入宝塔os管理界面

**6.启动因重启而停止的软件**

查看已安装软件 软件商店---已安装

启动停止运行的软件(nginx,mysql等)

## 动态添加docker映射端口

**1.先停止容器**

docker container stop xxx

**2.** **停止docker**

systemctl stop docker

**3.进度docker目录**

cd /var/lib/docker/containers/xxxxxxx

**4. 修改config.v2.jason文件与hostconfig.json文件**

仿照其他端口格式，增加新端口

**5.启动Docker**

systemctl start docker

**6.启动容器**

docker container start 9e4174f28269