**Nginx**

Nginx性能更高Apache

Apache(并发量3万) Nginx(并发量5万)

Unix和Linux平台

网页文件：Apache、Nginx、Tengine、Lighttpd

java代码网页文件：Tomact、IBM WebSphere、Jboss

index配置默认首页  
F12查看网页的信息

http 明文协议

权限针对用户

什么用户启动的QQ----> 读写哪些文件

当前用户能读的文件,QQ也能读

Root执行命令，以普通用户的身份执行

依赖包一般都叫“包名\_devel” 例：pcre\_devel

Mv /root/123.txt{,.back} //改名

**安装：**

准备项：

采用源码安装，可以自定义模块，不采用yum装

卸载httpd；端口被占用 // yum remove httpd

源码安装：

1. yum -y install gcc pcre-devel openssl-devel
2. Useradd -s /sbin/nologin nginx //普通不可登陆用户

不创建用户，默认降级为nobody用户

1. 进入源码文件夹

./configure --prefix=/usr/local/nginx --user=nginx --group=nginx

--with-http\_ssl\_module //指定安装路径，用户、组，SSL加密功能

//

1. make //编译
2. make install
3. 测试：curl http://192.168.4.5

**升级：**

1.编译新的nginx软件

tar -xf nginx-1.12.2.tar.gz

Cd ./nginx-1.12.2

./configure --prefix=/usr/local/nginx --user=nginx --group=nginx

//产生objs文件夹，将必要的包和我们徐需要的包放到objs下

make //编译链接我们需要的包

2.进行升级处理

mv /usr/local/nginx/sbin/nginx{,.bak} //备份老的版本，改名为nginx.bak

cp ./objs/nginx /usr/local/nginx/sbin/nginx //将新版本程序拷贝过去

Killall nginx //杀死旧的程序

Nginx //重启新版本的nginx

**用户认证：**

虚拟主机：基于域名，基于宽口，基于ip

server表示虚拟主机 多个虚拟主机就有多个server

配置文件：/usr/local/nginx/conf/nginx.conf

http{

全局配置（用户名,日志,进程)

server{

listen 80;

auth\_basic "客官，请验证你的身份：";

auth\_basic\_user\_file "/usr/local/nginx/pass"; //用户验证配置

server\_name [www.example.com;](http://www.example.com;) //虚拟主机名

root html; //虚拟主机的根 默认/usr/local/nginx/html/

}

}

yum -y install httpd-tools //提供htpasswd 命令的包

htpasswd -c /usr/local/nginx/pass xiao //添加验证用户，回车后添加密码

**基于域名的虚拟主机**

85,94s/#// #在vim里将85~94行的第一个#改为空/去掉注释

http{

server {

listen 80;

server\_name www.a.com;

auth\_basic "nihao world:";

auth\_basic\_user\_file "/usr/local/nginx/pass"; //用户认证

location / {

root html; //相对路径，相对与安装路径 ../nginx/html

等同与/usr/local/nginx/web/

index index.html index.htm;

}

}

server {

listen 80;

server\_name www.b.com;

location / {

root web; //同上，在/usr/local/nginx/web/

index index.html index.htm;

}

}

}

基于端口的虚拟主机

将上面基于域名的虚拟主机的listen端口改变，而领域不变

访问时在域名后加冒号即可实现访问 例：www.a.com:800

基于ip的虚拟主机

基于SSL的虚拟主机

源码安装Nginx时必须使用--with-http\_ssl\_module参数，启用加密模块，对于需要进行SSL加密处理的站点添加ssl相关指令（设置网站需要的私钥和证书）。

cd /usr/local/nginx/conf

openssl genrsa > cert.key                            //生成私钥

openssl req -new -x509 -key cert.key > cert.pem     //生成证书

硬盘 内存 CPU

程序-->进程

progr processes

**netstat命令**

可以查看系统中启动的端口信息，该命令常用选项如下：

-a显示所有端口的信息

-n以数字格式显示端口号

-t显示TCP连接的端口

-u显示UDP连接的端口

-l显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p显示监听端口的服务名称是什么（也就是程序名称）

nginx服务默认通过TCP 80端口监听客户端请求：

**总结：**

源码包编译

rpm包

源码包----gcc make---->可以执行的文件--->运行安装

软件的功能可按需选择/定制，安装位置可以选择

1.安装gcc make

2.Tar解包，释放到指定目录

3. ./configure 配置，指定安装目录/功能模块等选项

作用1：检测系统是否安装gcc

作用2：指定安装目录/功能模块等选项

指定安装位置的选项：./configure --prefix=

例子： ./configure --prefix=/mnt/myrpm

4.make编译，生成可执行的二进制程序文件（需要再文件文件夹内执行）

make install 安装，将编译好的文件复制到安装目录

指定安装位置/usr

指定用户nobody

模块--with-模块名称

./configure 检查环境gcc... 指定用户、组、模块、安装位置

make C语言的源码---->二进制

make install 拷贝编译链接好的软件拷贝到安装位置

升级：mv把老的程序重命名

cp objs/nginx 安装目录

kill(pid) killall 名称

全剧配置

http{

//虚拟主机

server{

listen 80;

server\_name www.a.com;

aut\_basic\_ "xxx";

auth\_basic\_user\_file "/user/local/nginx/";

root html;

}

//安全的html，https

server{

listen 443 ssl;

server\_name www.b.com;

sertification "证书名称";

certification\_key "私钥名称";

root html; //相对路径

}

}

:%s 全文替换 ：3,100s/#// 第三行到第100的#去掉

网页登陆验证

yum install httpd-tools

htpasswd -c /usr/local/nginx/conf/pass xiao

>12345 //密码

# nginx -s reload

隐藏服务器的版本号，并并修改为自定义的版本号

在/usr/local/nginx/conf/nginx.conf添加

在http的server之前添加

server\_tokens off;

修改源码包里的三个文件：

1.src/core下的nginx.h文件  
 #define nginx\_version 101010  
 #define NGINX\_VERSION "7.88"

#define NGINX\_VER "SSI/" NGINX\_VERSION

2.src/http下的ngx\_http\_header\_filter\_module.c文件，第49行

static char ngx\_http\_server\_string[] = "Server: SSI" CRLF;

3.src/http下的ngx\_http\_special\_response.c文件   
 static u\_char ngx\_http\_error\_tail[] =  
 "<hr><center>SSI</center>" CRLF

然后

make clean   
./configure --prefix=/usr/local/nginx

make && make install

watch 在linux可以周期性调用命令，并且可以指定周期长度。

-n 指定执行2次命令之间的间隔秒数

-d 高亮两次执行之间输出发生的变化 -> 对于看文本变化很有用！

-t 在输出顶部显示当前系统时间