

# 在Linux服务器安装Nginx

- 1. 描述
- 2. 在Linux服务器安装Nginx



## 1. 描述

---

介绍在Linux服务器安装Nginx的操作步骤。



## 2. 在Linux服务器安装Nginx

---

基于Debian（操作系统为Ubuntu）

准备好一个普通的非root用户并且该用户需要有sudo权限，使用这个用户登录Linux服务器，执行以下操作步骤。

操作步骤

1

安装Nginx。

由于Nginx在Ubuntu的默认存储库中可用，因此可以使用apt打包系统从这些存储库中安装Nginx。

执行以下命令，更新本地打包索引，以便可以访问最新的打包列表

```
sudo apt update
```

执行以下命令，安装nginx：

```
sudo apt install nginx
```

执行这两个命令后，apt会将Nginx和所有必需的依赖项安装到您的服务器。

2

调整防火墙。

在测试Nginx之前，需要对防火墙进行调整以允许访问该服务。Nginx在安装时将自己注册为ufw服务，使它可以直接访问Nginx。

执行以下命令，列出ufw可以使用的应用程序配置：

```
sudo ufw app list
```

执行命令后，获得应用程序配置文件的列表，类似如下显示：

OutputAvailable applications:

```
Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH
```

您可以看到，Nginx有三个可用的配置文件：

- Nginx Full：此配置文件同时打开端口80（正常，未加密的网络流量）和端口443（TLS / SSL加密的流量）；
- Nginx HTTP：此配置文件仅打开端口80（正常，未加密的网络流量）；
- Nginx HTTPS：此配置文件仅打开端口443（TLS / SSL加密流量）。

建议您启用限制性最强的配置文件，该配置文件仍将允许您配置的流量。这里我们只需要允许端口80上的流量。

3

检查您的Web服务器。

Nginx安装结束后，Ubuntu将启动Nginx，Web服务器启动并运行。

执行以下命令，使用systemd初始化系统进行检查，以确保服务正在运行。

```
systemctl status nginx
```

输出类似如下内容，表示服务正常运行。

```
Output nginx.service - A high performance web server and a reverse proxy server
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2018-04-20 16:08:19 UTC; 3 days ago
Docs: man:nginx(8)
Main PID: 2369 (nginx)
Tasks: 2 (limit: 1153)
CGroup: /system.slice/nginx.service
        2369 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
        2380 nginx: worker process
```

您还可以使用另一种方式，检查服务是否正常运行，即从Nginx请求一个页面。

您可以通过导航到服务器的IP地址来访问默认的Nginx登录页面，以确认服务是否正常运行。

如果您不知道服务器的IP地址，执行以下命令：

```
ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\./.*$//'
```

获取到返回的信息，您可以逐一在浏览器中尝试来看是否能访问。

或者执行以下命令，获取公共IP地址：

```
curl -4 icanhazip.com
```

获取到服务器的IP地址后，在浏览器的地址中输入：<http://服务器IP>，您会看到以下画面，证明服务在正常运行中。

:

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## 4

管理Nginx进程。使用以下命令，管理Nginx进程。

停止Web服务

```
sudo systemctl stop nginx
```

停止Web服务后启动服务

```
sudo systemctl start nginx
```

重启Web服务

```
sudo systemctl restart nginx
```

更改配置后，在Nginx不断开连接的情况下重新加载

```
sudo systemctl reload nginx
```

默认情况下，Nginx配置为在服务器启动时自动启动。可以通过执行以下命令来禁用此行为：

```
sudo systemctl disable nginx
```

重新启用服务以在启动时启动

```
sudo systemctl enable nginx
```

## 5

设置服务器块（推荐）。

使用Nginx Web服务器时，服务器块可用于封装配置详细信息，并在一台服务器中托管多个域。例如建立一个名为example.com的域名，您可以创建您自己的域名。

Nginx默认情况下启用了服务器块，该服务器块被配置为提供“/var/www/html”目录中的文档。

①执行以下命令，为example.com创建目录，使用-p标志创建所有需要的父目录：

```
sudo mkdir -p /var/www/example.com/html
```

②执行以下命令，使用\$USER环境变量分配目录的所有权：

```
sudo chown -R $USER:$USER /var/www/example.com/html
```

③如果您没有修改umask值，您的网络根目录的权限应该正确。可以执行以下命令，确保权限正确：

```
sudo chmod -R 755 /var/www/example.com
```

④执行以下命令，使用nano创建示例“index.html”页面：

```
nano /var/www/example.com/html/index.html
```

并将以下内容添加至“index.html”中，保存后关闭。

```
<html>
  <head>
    <title>Welcome to Example.com!</title>
  </head>
  <body>
    <h1>Success! The example.com server block is working!</h1>
  </body>
</html>
```

⑤执行以下命令，创建一个新的服务器块。

```
sudo vi /etc/nginx/sites-available/example.com
```

将以下内容粘贴到“example.com”中，更新其中的root的目录和域名。

```
server {
    listen 80;
    listen [::]:80;

    root /var/www/example.com/html;
    index index.html index.htm index.nginx-debian.html;

    server_name example.com www.example.com;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

⑥执行以下命令，通过创建指向站点启用目录的链接来启用文件，该链接在启动过程中从Nginx读取：

```
sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/
```

现在启用了两个服务器块并将其配置为根据它们的listen和server\_name指令来响应请求。

- example.com：将对example.com和www.example.com的请求进行响应。
- 默认值：将对端口80上与其他两个模块不匹配的任何请求进行响应。

⑦为了避免添加其他服务器名称可能引起的哈希存储桶内存问题，需要调整“/etc/nginx/nginx.conf”文件中的值。

执行以下命令，打开文件：

```
sudo vi /etc/nginx/nginx.conf
```

找到“server\_names\_hash\_bucket\_size”指令并删除“#”符号以取消注释该行，保存后关闭。

```
...
http {
    ...
    server_names_hash_bucket_size 64;
    ...
}
...
```

⑧执行以下命令，以确保您的任何Nginx文件中都没有语法错误：

```
sudo nginx -t
```

现在Nginx会为您的域名服务。您可以通过导航到<http://example.com>进行测试，您会看到以下内容：

## Success! The example.com server block is working!

6

熟悉重要的Nginx文件和目录。

内容

- `/var/www/`  
html：实际的Web内容（默认情况下仅由您之前看到的默认Nginx页面组成）从`/var/www/html`目录中提供。可以通过更改Nginx配置文件来更改。

服务器配置

- `/etc/nginx` Nginx配置目录。所有Nginx配置文件都位于此处。
- `/etc/nginx/nginx.conf`：主要的Nginx配置文件。可以对其进行修改以更改Nginx全局配置。
- `/etc/nginx/sites-available/`：可以存储每个站点服务器块的目录。  
Nginx不会使用在此目录中找到的配置文件，除非它们已链接到启用站点的目录。通常，所有服务器块配置都在此目录中完成，然后通过链接到其他目录来启用。
- `/etc/nginx/sites-enabled/`：存储已启用的每个站点服务器块的目录。通常，通过链接到可在`sites-available`目录中找到的配置文件来创建这些文件。
- `/etc/nginx/snippets`：此目录包含一些配置片段，这些片段可以包含在Nginx配置中的其他位置。

服务器日志

- `/var/log/nginx/access.log`：除非将Nginx配置为执行其他操作，否则对Web服务器的每个请求都记录在此日志文件中。
- `/var/log/nginx/error.log`：所有Nginx错误都将记录在此日志中。

---

**结束**

---

基于RPM（操作系统为CentOS、RedHat 7.6、中标麒麟）

操作步骤

1

更新存储库软件包列表。执行以下命令：

```
sudo yum -y update
```

2

安装企业Linux（EPEL）的额外软件包。

Nginx在CentOS软件包随附的标准存储库中不可用，因此您需要在服务器上安装EPEL存储库。  
EPEL是免费使用的，并提供了许多可与Yum一起安装的开源软件包。

要安装EPEL，请使用Yum软件包管理器执行以下命令：

```
sudo yum install -y epel-release
```

```
[dejanPNAP@localhost ~]$ sudo yum install -y epel-release
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.etf.bg.ac.rs
* extras: mirror.etf.bg.ac.rs
* updates: mirror.etf.bg.ac.rs
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                        Arch      Version      Repository      Size
=====
Installing:
epel-release                  noarch    7-11         extras          15 k

Transaction Summary
=====
Install 1 Package

Total download size: 15 k
Installed size: 24 k
```

3

安装Nginx。

步骤

2

中已将Nginx存储库添加到您的服务器。现在，您可以通过执行以下yum命令来安装Nginx：

```
sudo yum -y install nginx
```

```
[dejanPNAP@localhost ~]$ sudo yum -y install nginx
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
epel/x86_64/metalink | 30 kB 00:00:00
* base: mirror.etf.bg.ac.rs
* epel: mirror.etf.bg.ac.rs
* extras: mirror.etf.bg.ac.rs
* updates: mirror.etf.bg.ac.rs
epel | 4.7 kB 00:00:00
(1/3): epel/x86_64/group_gz | 95 kB 00:00:00
(2/3): epel/x86_64/updateinfo | 1.0 MB 00:00:00
(3/3): epel/x86_64/primary_db | 6.8 MB 00:00:01
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.16.1-1.el7 will be installed
--> Processing Dependency: nginx-all-modules = 1:1.16.1-1.el7 for package: 1:nginx-1.16.1-1.el7.x86_64
--> Processing Dependency: nginx-filesystem = 1:1.16.1-1.el7 for package: 1:nginx-1.16.1-1.el7.x86_64
--> Processing Dependency: nginx-filesystem for package: 1:nginx-1.16.1-1.el7.x86_64
--> Running transaction check
---> Package nginx-all-modules.noarch 1:1.16.1-1.el7 will be installed
--> Processing Dependency: nginx-mod-http-image-filter = 1:1.16.1-1.el7 for package: 1:nginx-all-modules-1.16.1-1.el7.noarch
--> Processing Dependency: nginx-mod-http-perl = 1:1.16.1-1.el7 for package: 1:nginx-all-modules-1.16.1-1.el7.noarch
```

4

启动Nginx服务。安装Nginx后不会自动启动，需要执行以下命令启动Nginx服务：

```
sudo systemctl start nginx
```

执行以下命令，查看状态：

```
sudo systemctl status nginx
```

```
[dejanPNAP@localhost ~]$ sudo systemctl start nginx
[dejanPNAP@localhost ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-05-27 04:37:33 EDT; 10s ago
     Process: 11188 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 11186 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 11180 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 11190 (nginx)
      Tasks: 2
     CGroup: /system.slice/nginx.service
             └─11190 nginx: master process /usr/sbin/nginx
               └─11191 nginx: worker process

May 27 04:37:32 localhost.localdomain systemd[1]: Starting The nginx HTTP and rever....
May 27 04:37:33 localhost.localdomain nginx[11186]: nginx: the configuration file /...k
May 27 04:37:33 localhost.localdomain nginx[11186]: nginx: configuration file /etc/...l
May 27 04:37:33 localhost.localdomain systemd[1]: Failed to parse PID from file /ru...t
May 27 04:37:33 localhost.localdomain systemd[1]: Started The nginx HTTP and revers....
Hint: Some lines were ellipsized, use -l to show in full.
[dejanPNAP@localhost ~]$
```

绿色文字应显示为“active(running)”。如果不显示，则可能是您的Nginx实例未成功启动。

注意：如果您已经在运行Apache服务器，则需要在启动Nginx之前将其禁用。使用sudo service httpd stop命令。禁用Apache将关闭当前托管的任何网站。

禁用了Apache后，在服务器重新引导期间它仍可能会自动启动。执行以下命令来禁用自动启动：

```
sudo systemctl disable httpd
```

5

配置Nginx在服务器启动时自动启动。执行以下命令：

```
sudo systemctl enable nginx
```

6

配置防火墙以允许流量通过。

CentOS 7默认情况下启用防火墙，并阻止访问端口80和443。它将阻止来自Nginx的所有入站HTTPS和HTTP数据包。

执行以下命令，允许HTTP和HTTPS通信：

```
firewall-cmd --zone=public --permanent --add-service=http
firewall-cmd --zone=public --permanent --add-service=https
firewall-cmd --reload
```

在每个命令之后，您应该看到“success”表示该命令已正确执行。

```
[dejanPNAP@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=http
success
[dejanPNAP@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=https
success
[dejanPNAP@localhost ~]$ firewall-cmd --reload
success
[dejanPNAP@localhost ~]$ █
```

7

验证Nginx安装。

检查Nginx是否正常运行的最简单的方法是访问服务器的公共IP地址。打开您的Web浏览器，然后访问“http://服务器IP或域名”

执行以下命令，找到您服务器的公共IP地址。

```
[dejanPNAP@localhost ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ab:71:d5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 83864sec preferred_lft 83864sec
    inet6 fe80::b7c1:dff1:8958:6137/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:ba:f1:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:ba:f1:91 brd ff:ff:ff:ff:ff:ff
[dejanPNAP@localhost ~]$
```

在浏览器中访问，应该显示如下所示的Nginx欢迎界面：

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

8

配置SELinux。

执行以下命令，检查是否启用了SELinux：

```
getenforce
```

如果输出Enforcing，则表示SELinux已打开，需要启用Nginx发送http请求的权限。执行以下命令：

```
sudo setsebool -P httpd_can_network_connect 1
```

如何解决“test failed”错误

如果您收到nginx.conf文件的“test failed”错误消息，则可能是IP地址问题。

默认情况下，Nginx服务侦听IPv4和IPv6。如果您的服务器不支持IPv6，则测试将失败。可以通过修改主配置文件来解决此问题。

打开主配置文件“/etc/nginx/nginx.conf”，查找并注释掉以下行，在前面加“#”即可



```
# listen [::]:80 default_server;
```

保存文件关闭后，执行以下命令重启Nginx服务：

```
sudo systemctl reload nginx
```

重启服务后，在浏览器中访问服务器IP地址，应该显示Nginx页面。

## Nginx配置文件和根目录

您需要知道Nginx配置文件的位置和默认的Nginx服务器根目录。

### 附加服务器块

在Apache上，管理员使用虚拟主机来运行多个网站。使用Nginx，可以使用服务器块在单个服务器上运行多个网站。

通过创建扩展名为.conf的新配置文件来添加其他服务器块。

将这些文件放在“/etc/nginx/conf.d”中，每次Nginx启动时将加载它们。

### 默认的Nginx服务器根目录

默认的Nginx服务器根目录是“/usr/share/nginx”。

这是在默认服务器块配置文件中指定的，该文件位于“/etc/nginx/conf.d/default.conf”。

包含Web文件的默认服务器文档根目录为“usr/share/nginx/html”。

### 全局配置

可以通过修改“/etc/nginx/nginx.conf”中的主要Nginx配置文件来调整全局配置。默认情况下，您可以识别三个内容：

- 事件是全局设置，用于定义Nginx通常如何处理连接。
- HTTP定义服务器如何处理HTTP和HTTPS连接。
- 服务器是在HTTP上下文中定义的。它指定服务器端口，文档根目录等。

### 管理Nginx

#### 停止Nginx

```
sudo systemctl stop nginx
```

#### 重新启动Nginx服务

```
sudo systemctl restart nginx
```

#### 重新加载Nginx

```
sudo systemctl reload nginx
```

#### 禁用服务器启动时自动启动Nginx

```
sudo systemctl disable nginx
```

## 配置新目录

如果您托管多个网站，最好是遵循标准的命名约定。我们使用cPanel的标准命名并创建目录。

```
sudo mkdir -p /var/www/yourdomain.com/public_html
```

然后创建索引页，帮助您测试配置。

```
sudo nano /var/www/yourdomain.com/public_html/index.html
```

为了进行测试，请在index.html中输入一行文本，保存并关闭文件。

执行以下命令，更改Linux文件权限，以便可以在线访问数据。

```
sudo chmod 755 /var/www/yourdomain.com/public_html
```

然后您就可以在线打开index.html页面。

---

**结束**

---