

گزارش راه اندازی Nginx در VirtualHost

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر میرسامان تاجبخش

چکیده:

در این تمرین قصد داریم یک وب سرویس راه اندازی کنیم که شامل دو هاست مجازی باشد. برای انجام این کار لازم است که ابتدا مقدماتی را درباره ی سرویس های Nginx و نیز مفاهیم Block و Virtual Hosts دانسته و سیستم خود را بر اساس عملکرد این سرویس ها تنظیم کنیم. در گزارش پیش رو مراحل لازم برای طراحی و پیاده سازی چنین وب سرویسی را قدم به قدم شرح خواهیم داد.

زمانی که از وب سرور Nginx استفاده می کنید، بلوک های سرور (server blocks) که معادل میزبان های مجازی (virtual hosts) در وب سرور آپاچی هستند، به منظور جمع بندی جزئیات پیکربندی و میزبانی بیش از یک دامنه روی یک سرور منفرد به کار گرفته می شوند.

در این گزارش کار برای پیاده سازی این وب سرویس از سیستم عامل اوبونتو استفاده خواهیم کرد که به کمک یک ماشین مجازی بالا آورده شده.

پیش نیازها

ما از یک کاربر غیر root با دسترسی های sudo در سراسر این راهنما استفاده خواهیم کرد.

همچنین باید Nginx را روی سرور خود نصب کنید.

اگر قصد استفاده از سایر سرویس های آپاچی را ندارید و صرفاً می خواهید از Nginx استفاده کنید، می توانید آن را با دستورهای زیر نصب کنید:

```
sudo apt-get update
```

```
sudo apt-get install nginx
```

ما صرفاً به منظور نمایش مقصود، دو دامنه را با سرور Nginx راه اندازی می کنیم. نام های دامنه های مورد استفاده در این راهنما به صورت example.com و test.com خواهند بود.

اگر دو نام دامنه متفاوت برای استفاده ندارید، می توانید از نام های ساختگی استفاده کنید. در ادامه روش پیکربندی رایانه محلی برای تست پیکربندی در این شرایط را نیز توضیح خواهیم داد.

مرحله اول – راهاندازی دایرکتوری‌های جدید برای ریشه سند

به طور پیش فرض Nginx روی اوبونتو یک بلوک سرور را در حالت default فعال می‌کند. این بلوک سرور برای عرضه اسنادی که در دایرکتوری زیر قرار دارند، پیکربندی شده است:

```
/usr/share/nginx/html
```

ما از این بلوک سرور پیش فرض استفاده نمی‌کنیم، زیرا کار با دایرکتوری /var/www آسان‌تر است. بسته Nginx اوبونتو از دایرکتوری /var/www به عنوان ریشه سند پیش فرض استفاده نمی‌کند و دلیل آن سیاست توزیع دبیان در خصوص بسته‌هایش است.

از آنجا که ما کاربر هستیم و نه مسئول نگهداری بسته، پس می‌توانیم به Nginx بگوییم که کجا می‌خواهیم ریشه سندمان را قرار دهیم. به طور خاص ما برای هر یک از سایت‌های خود درون دایرکتوری /var/www یک دایرکتوری می‌سازیم و یک دایرکتوری نیز درون آن‌ها به نام html باید داشته باشیم که فایل‌ها را در آن‌ها نگه‌داری کنیم.

ابتدا باید دایرکتوری‌های مورد نیاز را بسازیم. این کار از طریق استفاده از دستورهای زیر ممکن است. فلگ -p به mkdir می‌گوید که دایرکتوری‌های والد ضروری را نیز در این مسیر باید ایجاد کند:

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

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

اینک که دایرکتوری‌ها ایجاد شدند، باید مالکیت را به کاربر معمولی انتقال دهیم. می‌توان از متغیر محیطی \$USER برای جایگزینی حساب کاربری که اکنون وارد آن شده‌ایم استفاده کنیم. بدین ترتیب امکان ایجاد فایل‌هایی در این دایرکتوری را می‌یابیم، به طوری که بازدیدکنندگان دیگر نتوانند محتوایی در آن ایجاد کنند.

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

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

مجوزهای ریشه‌های وب ما در صورتی که مقدار mask را تغییر نداده باشید، باید به طور صحیحی پیکربندی شده باشند؛ اما با وارد کردن دستور زیر می‌توانید در این مورد اطمینان حاصل کنید:

```
sudo chmod -R 755 /var/www
```

ساختار دایرکتوری اینک پیکربندی شده است و می‌توانیم به مراحل بعدی پردازیم.

مرحله دوم – ایجاد صفحه‌های ساده برای هر سایت

اکنون که ساختار دایرکتوری ما آماده شده است، می‌توانیم صفحه‌های پیش فرضی برای هر یک از سایت‌ها ایجاد کنیم که در آن چیزی برای نمایش وجود داشته باشد. یک فایل index.html در دامنه اول خود ایجاد می‌کنیم:

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

درون این فایل، یک محتوای بسیار ابتدایی ایجاد می‌کنیم که نشان‌دهنده این باشد که در حال حاضر به چه سایتی دسترسی یافته‌ایم. محتوای فایل به صورت زیر است:

```
<html>
```

```
<head>
```

```
<title>Welcome to Example.com!</title>
```

```
</head>
```

```
<body>
```

```
<h1>Success! The example.com server block is working!</h1>
```

```
</body>
```

```
</html>
```

زمانی که محتوا را وارد کردید، فایل را ذخیره کرده و خارج شوید.

از آنجا که فایل سایت دوم ما نیز به همین شکل خواهد بود، می‌توانید آن را در ریشه سند سایت دوم به صورت زیر کپی کنید:

```
cp /var/www/example.com/html/index.html /var/www/test.com/html/
```

اینک می‌توانیم این فایل جدید را باز کنیم:

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

و آن را طوری ویرایش کنیم که نشان دهد روی دامنه دوم ما قرار دارد:

```
<html>
```

```
<head>
```

```
<title>Welcome to Test.com!</title>
```

```
</head>
```

```
<body>
```

```
<h1>Success! The test.com server block is working!</h1>
```

```
</body>
```

```
</html>
```

در پایان این فایل را نیز ذخیره کرده و خارج شوید. اینک روی هر دو دامنه خود صفحاتی داریم که به بازدیدکنندگان وبسایت نمایش دهیم.

مرحله سوم – ایجاد فایل‌های بلوک سرور برای هر دامنه

اینک که محتوایی که قرار است عرضه شود را در دست داریم، باید بلوک‌های سروری ایجاد کنیم که Nginx بداند چگونه باید این کار را انجام دهد.

به طور پیش‌فرض Nginx شامل یک بلوک سرور به نام default است که می‌توان از آن به عنوان قالبی برای پیکربندی بلوک‌های دیگر استفاده کرد. در ابتدا بلوک سرور دامنه اول خود را طراحی می‌کنیم. سپس آن را به دامنه دوم کپی کرده و تغییرهای لازم را در آن ایجاد می‌کنیم.

ایجاد فایل بلوک سرور برای دامنه اول

همان طور که قبلاً اشاره کردیم، بلوک سرور اول را با کپی کردن فایل default ایجاد می‌کنیم:

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/example.com
```

اکنون فایل جدید را در ویرایشگر متنی خود با دسترسی root باز کنید:

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

با نادیده گرفتن خط‌های کامنت شده، فایل به صورت زیر باید باشد:

```
server {
```

```
    listen 80 default_server;
```

```
    listen [::]:80 default_server ipv6only=on;
```

```
root /usr/share/nginx/html;
```

```
index index.html index.htm;
```

```
server_name localhost;
```

```
location / {
```

```
try_files $uri $uri/ =404;
```

```
}
```

```
}
```

ابتدا باید دایرکتیوهای listen را بررسی کنیم. تنها یکی از بلوک‌های سرور می‌تواند دارای خصوصیات default_server باشد. این خصوصیت نشان می‌دهد که اگر server_name مورد تقاضا با هیچ یک از بلوک‌های سرور مطابقت نداشته باشد، این بلوک می‌تواند به درخواست مربوطه پاسخ دهد.

در نهایت باید پیکربندی بلوک سرور پیش فرض را غیر فعال کنیم تا بتوانیم گزینه default_server را در بلوک سرور این دامنه یا دامنه دیگر تعیین کنیم. در این راهنما گزینه default_server را روی این دامنه فعال نگه می‌داریم؛ اما شما می‌توانید بسته به نیازهای خود روی هر دامنه دیگری نیز آن را تنظیم کنید.

کار بعدی که باید انجام دهیم، تنظیم ریشه سند است که به وسیله دایرکتیو root مشخص می‌شود. در ریشه سند سایتی که ایجاد کردید به این مقدار اشاره کنید:

```
root /var/www/example.com/html;
```

دقت کنید که هر عبارت در Nginx باید با یک نقطه‌ویرگول (;) بسته شود، بدین ترتیب باید همه خطوط را بررسی کنید تا دچار مشکل نشوید.

سپس اقدام به ویرایش server_name می‌کنیم تا با درخواست‌هایی که برای دامنه ما می‌آید، مطابقت داشته باشد. همچنین می‌توانیم هر نام مستعار (alias) که دوست داریم برای آن تعیین کنیم. برای نمایش طرز کار alias ها یک مورد به صورت www.example.com ایجاد می‌کنیم.

```
server_name example.com www.example.com;
```

زمانی که کار ویرایش فایل به اتمام رسید، به صورت زیر درخواست خواهد آمد:

```
server {
```

```
listen 80 default_server;
```

```
listen [::]:80 default_server ipv6only=on;
```

```
root /var/www/example.com/html;
```

```
index index.html index.htm;
```

```
server_name example.com www.example.com;
```

```
location / {
```

```
try_files $uri $uri/ =404;
```

```
}
```

```
}
```

این‌ها همه مواردی بودند که به عنوان پیکربندی مقدماتی لازم داریم. فایل را ذخیره کرده و خارج شوید.

ایجاد فایل بلوک برای سرور دوم

اینک که پیکربندی بلوک سرور اول به پایان رسیده است، می‌توانیم از آن به عنوان پایه‌ای برای فایل دوم استفاده کنیم. آن را کپی کنید تا فایل جدیدی ایجاد شود:

```
sudo cp /etc/nginx/sites-available/example.com /etc/nginx/sites-available/test.com
```

فایل جدید را با دسترسی root در ویرایشگر خود باز کنید:

```
sudo nano /etc/nginx/sites-available/test.com
```

در فایل جدید باید مجدداً دایرکتیو listen را بررسی کنیم. اگر گزینه default_server در فایل قبلی فعال بوده باشد، باید آن را در این فایل حذف کنیم. به علاوه باید گزینه ipv6only=on را نیز حذف کنید چون برای هر ترکیب آدرس/پورت تنها یک بار می‌تواند مورد استفاده قرار گیرد:

```
listen 80;
```

```
listen [::]:80;
```

دایرکتیو document root را طوری تنظیم کنید که به ریشه سند سایت فعلی اشاره داشته باشد:

```
root /var/www/test.com/html;
```

server_name را طوری تنظیم کنید که با alias ها مطابقت داشته باشد:

```
server_name test.com www.test.com;
```

اینک فایل شما با این تغییرات باید به صورت زیر درآمده باشد:

```
server {
```

```
listen 80;
```

```
listen [::]:80;

root /var/www/test.com/html;

index index.html index.htm;

server_name test.com www.test.com;

location / {

    try_files $uri $uri/ =404;

}

}
```

زمانی که کار ویرایش فایل به اتمام رسید، آن را ذخیره کرده و ببندید.

مرحله چهارم – فعال‌سازی بلوک‌های سرور و ری‌استارت Nginx

اینک بلوک‌های سرور ایجاد شده‌اند و باید آن‌ها را فعال کنیم. این کار از طریق ایجاد پیوندهای نمادین (symbolic links) از این فایل‌ها به دایرکتوری sites-enabled ممکن می‌شود. Nginx در طی راه‌اندازی اولیه محتوای این دایرکتوری را می‌خواند. با وارد کردن دستور زیر می‌توان این پیوندها را ایجاد کرد:

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

اکنون این فایل‌ها در دایرکتوری enabled قرار دارند. با این حال، فایل بلوک سرور پیش‌فرض که از آن به عنوان قالب استفاده کردیم هم در حال حاضر فعال است و با فایل ما که پارامتر default_server را برای آن تنظیم کرده‌ایم، تداخل می‌یابد.

می‌توان فایل بلوک سرور پیش‌فرض را با حذف پیوند نمادین آن غیر فعال کرد. البته این فایل همچنان در دایرکتوری sites-available به منظور ارجاع در موارد آتی باقی می‌ماند؛ اما Nginx هنگام راه‌اندازی اولیه آن را نمی‌خواند:

```
sudo rm /etc/nginx/sites-enabled/default
```

همچنین باید یک تنظیم واقعاً کوچک را نیز در فایل پیکربندی Nginx ایجاد کنیم. این فایل را با دستور زیر باز کنید:

```
sudo nano /etc/nginx/nginx.conf
```

در این فایل تنها باید یک خط را از حالت کامنت در بیاوریم. دستور زیر را یافته و کامنت آن حذف کنید:

```
server_names_hash_bucket_size 64;
```

اکنون آماده هستیم که Nginx را ری‌استارت کرده و تغییرات را فعال کنیم. این کار از طریق دستور زیر ممکن است:

```
sudo service nginx restart
```

Nginx اینک می‌تواند هر دو نام دامنه را عرضه کند.

مرحله پنجم – راه‌اندازی فایل میزبان‌های محلی (اختیاری)

اگر از نام‌های دامنه واقعی استفاده نمی‌کنید و به جای آن از مقادیر ساختگی استفاده کرده‌اید؛ می‌توانید پیکربندی رایانه محلی خود را طوری تغییر دهید که امکان تست موقت پیکربندی بلوک سرور Nginx وجود داشته باشد.

البته بدین ترتیب بازدیدکنندگان دیگر امکان دیدن سایت شما را نخواهند داشت؛ اما این امکان را می‌یابید که به طور مستقل هر سایت را ببینید و پیکربندی خود را تست کنید. این وضعیت اساساً از طریق پذیرش درخواست‌هایی که معمولاً به DNS برای resolve می‌روند ممکن می‌شود. به جای این کار می‌توانید آدرس IP که می‌خواهید رایانه محلی در هنگام دریافت نام دامنه مراجعه کند را تنظیم کنید.

ابتدا اطمینان حاصل کنید که در طی این مراحل روی یک سرور VPS کار نمی‌کنید؛ بلکه همه مراحل را رایانه محلی اجرا کرده‌اید. همچنین باید دسترسی root داشته باشید، یا عضو گروه administrative سیستم باشید و یا این که قابلیت ویرایش فایل‌های سیستمی را داشته باشید. اگر روی یک رایانه مک یا لینوکس کار می‌کنید می‌توانید فایل مورد نیاز را با وارد کردن دستور زیر ویرایش کنید:

```
sudo nano /etc/hosts
```

اگر روی ویندوز هستید باید فایل زیر را در یک ویرایشگر متنی مانند Notepad باز کنید:

```
C:\Windows\System32\drivers\etc\hosts
```


در این مرحله باید آدرس IP عمومی و دامنه‌ای که می‌خواهید سرور به آن مسیریابی کند را بدانید. با فرض این که آدرس IP عمومی شما به صورت 111.111.111.111 باشد، خطوط زیر را به فایل فوق اضافه کنید:

```
127.0.0.1 localhost
```

```
127.0.0.1 guest-desktop
```

```
111.111.111.111 example.com
```

```
111.111.111.111 test.com
```

بدین ترتیب همه درخواست‌ها به دامنه example.com و test.com به سرور شما ارسال می‌شود که در صورتی که عملاً این دو دامنه را نداشته باشید، وضعیت مطلوبی به شمار می‌رود. هنگام اتمام کار ویرایش، فایل را ذخیره کرده و خارج شوید.

مرحله ششم – تست نتایج

اینک که همه چیز راه‌اندازی شده است، باید بلوک‌های سرور را تست کنید تا از عملکرد صحیح آن‌ها مطمئن شوید. این کار از طریق بازدید از دامنه‌ها روی مرورگر وب ممکن است:

```
http://example.com
```

در مرورگر خود باید صفحه‌ای ببینید که شبیه تصویر زیر است:

Success! The example.com server block is working!

اگر از دامنه دوم نیز بازدید کنید، باید سایت زیر را با اندکی تفاوت مشاهده کنید:

```
http://test.com
```

Success! The test.com server block is working!

اگر هر دو سایت به طرز صحیحی کار می‌کنند می‌توان نتیجه گرفت که شما موفق شده‌اید دو بلوک سرور مستقل از هم را با Nginx پیکربندی کنید. در این زمان اگر فایل hosts را روی رایانه محلی خود تغییر داده‌اید تا نتیجه کارتان را تست کنید، باید خطوطی را که تغییر داده‌اید به حالت قبل بازگردانید. اگر می‌خواهید در سایت با دسترسی عمومی یک نام دامنه عمومی تنظیم کنید، می‌بایست یک نام دامنه برای هر یک از وبسایت‌های خود تهیه کنید.

منابع:

- i. <https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>
- ii. https://www.nginx.com/resources/wiki/start/topics/examples/server_blocks/
- iii. <https://bitlaunch.io/blog/how-to-set-up-nginx-server-blocks-virtual-hosts-in-ubuntu-20-04/>
- iv. <https://linuxize.com/post/how-to-set-up-nginx-server-blocks-on-ubuntu-20-04/>
- v. <https://linuxhint.com/set-up-nginx-server-blocks-ubuntu-22-04/>

پایان.