NGINX Server Blocks - Virtual Hosts

NGINX has a feature called server blocks. This is basically the same thing as Apache virtual hosts. This basically allows you to run as many sites as you want on a single server with a single IP. The main limitation is resources. Other than that you can effectively serve unlimited domains for free.
We're going to create two example sites: example.org and test.com.
First, make sure that you've already installed NGINX. See **Our NGINX Install / Setup Doc Here**.
Second, make sure your domains are pionting at your server's IP address. If you just want to test this without owning the domains, we will show you how to do that at the end of this guide in the testing section.
Next, create an HTML directory for each site you create:

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

Next, change ownership the /var/www directory for the user that will be deploying and managing web content. In this case we've called that "user1":

```
sudo chown -R user1:user1 /var/www/
```

Update the permissions on the /var/www directory:

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

Create a new, initial index.html file for the first site:

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

```
<html>
<head>
<title>Site 1 Test</title>
</head>
<body>
<h1>Site 1 Test</h1>
</body>
</html>
```

Also create an index.html file for the second site:

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

```
<html>
<head>
<title>Site 2 Test</title>
</head>
<body>
<h1>Site 2 Test</h1>
</body>
</html>
```

IMPORTANT -You can only have one default server block. The default_server option can only be enabled on one server block.
Copy of the default config file to get you started:

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

Then edit the file to look like this:

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

```
server {
listen 80;
listen [::]:80;
root /var/www/example.org/html;
index index.html index.htm index.nginx-debian.html;
server_name example.org;
location / {
try_files $uri $uri/ =404;
}
}
```

Copy the file for the first site:

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

And edit it where needed to look like this:

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

```
server {
listen 80;
listen [::]:80;
root /var/www/test.com/html;
index index.html index.htm index.nginx-debian.html;
server_name test.com;
location / {
try_files $uri $uri/ =404;
}
}
```

Next, create symlinks under sites-available to enable the two new sites:

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

Uncomment this line to set the server_hash_bucket_size to 64 if it isn’t already.

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

server_names_hash_bucket_size 64;
```

Test the syntax of the NGINX config files before restarting:

```
sudo nginx -t
```

If everything looks OK, go ahead and restart NGINX:

```
sudo systemctl restart nginx
```

Testing

If you actually own the domains you would have them pointing at the IP of your webserver. For this example we don't actually own either of these domains so we will just create entries in /etc/hosts so that when we navigate to them we will reach our server.

```
sudo nano /etc/hosts

203.0.113.5 example.org
203.0.113.5 test.com
```

Load each of these in a browser:

- http://example.com
- http://test.com

NGINX Server Blocks Video

This is our video showing you how to setup NGINX server blocks: