How to Install Nginx on Debian

alias name=value alias name='command' alias name='command arg1 arg2' alias name='/path/to/script' alias name='/path/to/script.pl arg1'	alias ll=’ls -la’
root@nanopiair:/# alias ll='ls -la' root@nanopiair:/# alias alias ll='ls -la'	
root@nanopiair:/# cat /root/.bashrc # ~/.bashrc: executed by bash(1) for non-login shells. # Note: PS1 and umask are already set in /etc/profile. You should not # need this unless you want different defaults for root. # PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ ' # umask 022 # You may uncomment the following lines if you want `ls' to be colorized: # export LS_OPTIONS='--color=auto' # eval "`dircolors`" # alias ls='ls $LS_OPTIONS'	
alias ll='ls $LS_OPTIONS -l' alias l='ls $LS_OPTIONS -lA'	
# # Some more alias to avoid making mistakes: # alias rm='rm -i' # alias cp='cp -i' # alias mv='mv -i'	

Update system repositories

sudo apt update
sudo apt upgrade

Step 1: Then, install the NGINX package:

sudo apt install nginx

Step 2 – Adjusting the Firewall

UFW (Uncomplicated Firewall) - является самым простым и довольно популярным инструментарием командной строки для настройки и управления брандмауэром в дистрибутивах Ubuntu и Debian.

Брандма́уэр Windows — встроенный в Microsoft Windows межсетевой экран.
Файервол (иногда его еще называют брандмауэр) – это система, которая предотвращает несанкционированный доступ к сети.

Термин брандмауэр *(нем. Brandmauer, от Brand — пожар и Mauer — стена) — глухая противопожарная стена здания или его английский эквивалент* ***файрвол*** *(англ. firewall; fire - огонь, wall – стена) используется также в значении «межсетевой экран».*
Брандмауэр или firewall – это программный комплекс, предназначенный для защиты компьютера от сетевых атак. Следует отметить, что благодаря брандмауэрам увеличивается безопасность работы в сети, а также отражается большинство атак на компьютер путем фильтрации некоторых информационных пакетов.

# **apt install ufw** root@nanopiair:~# **ufw status verbose** Status: inactive root@nanopiair:~# **ufw enable** Firewall is active and enabled on system startup root@nanopiair:~# **ufw status verbose** Status: active Logging: on (low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip	
проверяем	**ufw enable**
	ufw status verbose

`sudo ufw allow 1883`
`sudo ufw allow 9001`
`sudo ufw enable`	

root@nanopiair:~# ufw app list	**Available applications:** **AIM** **...** **Kerberos Admin** **Kerberos Full** **Kerberos KDC** **Kerberos Password** **LDAP** **LDAPS** **LPD** **MSN** **MSN SSL** **Mail submission** **NFS** **Nginx Full** **Nginx HTTP** **Nginx HTTPS** **OpenSSH** **POP3** **POP3S** **PeopleNearby** **SMTP** **SSH** **Socks** **Telnet** **Transmission** **Transparent Proxy** **VNC** **WWW** **WWW Cache** **WWW Full** **WWW Secure**

	...
Nginx Full: This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic) **Nginx HTTP**: This profile opens only port 80 (normal, unencrypted web traffic) **Nginx HTTPS**: This profile opens only port 443 (TLS/SSL encrypted traffic)	

It is recommended that you enable the most restrictive profile that will still allow the traffic you've configured. Since you haven't configured TLS/SSL for your server yet in this guide, you will only need to allow traffic for HTTP on port 80	
`sudo ufw allow 'Nginx HTTP'`	root@nanopiair:~# ufw allow 'Nginx HTTP' Rule added Rule added (v6)
`sudo ufw status`	root@nanopiair:~# ufw status Status: active To Action From -- ------ ---- Nginx HTTP ALLOW Anywhere Nginx HTTP (v6) ALLOW Anywhere (v6)

Step 3 – Checking your Web Server

systemctl status nginx.service	root@nanopiair:~# systemctl status nginx.service ● nginx.service - A high performance web server and a reverse proxy server Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: en Active: active (running) since Wed 2023-03-29 04:49:39 UTC; 23min ago Docs: man:nginx(8) Main PID: 25071 (nginx) Tasks: 5 (limit: 4915) CGroup: /system.slice/nginx.service ├─25071 nginx: master process /usr/sbin/nginx -g daemon on; master_pr ├─25072 nginx: worker process ├─25073 nginx: worker process ├─25074 nginx: worker process └─25075 nginx: worker process Mar 29 04:49:39 nanopiair systemd[1]: Starting A high performance web server and Mar 29 04:49:39 nanopiair systemd[1]: Started A high performance web server and
systemctl start nginx.service	
`ip addr show eth0 | grep inet | awk '{ print $2; }' \| sed 's/\/.*$//'`	

root@nanopiair:~# ip addr show wlan0 | grep inet | awk '{ print $2; }' \| sed 's/\/.*$//' 1.192.168.137.113 2.fe80::b3c1:1ec5:ef42:89a4

hostname -I	root@nanopiair:/# hostname -I 192.168.137.113
`http://192.168.137.113/`	

Неисправности в сети или не подключается к сети

NetworkManager	

```
root@nanopiair:~# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor p
   Active: active (running) since Wed 2023-03-29 05:38:51 UTC; 9min ago
     Docs: man:NetworkManager(8)
 Main PID: 734 (NetworkManager)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/NetworkManager.service
           └─734 /usr/sbin/NetworkManager --no-daemon

Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3594] device
Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3599] Config:
Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3601] Config:
Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3603] Config:
Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3604] Config:
Mar 29 05:44:06 nanopiair NetworkManager[734]: <info>  [1680068646.3606] Config:
Mar 29 05:44:14 nanopiair NetworkManager[734]: <info>  [1680068654.6955] device
Mar 29 05:44:14 nanopiair NetworkManager[734]: <info>  [1680068654.7012] audit:
Mar 29 05:44:14 nanopiair NetworkManager[734]: <info>  [1680068654.7117] device
Mar 29 05:44:14 nanopiair NetworkManager[734]: <info>  [1680068654.7392] device
```

`systemctl restart NetworkManager`	
`nmcli networking off` `nmcli networking on`	root@nanopiair:~# nmcli networking enabled
`nmcli radio wifi off` `nmcli radio wifi on`	root@nanopiair:~# nmcli radio wifi enabled
`sudo ifconfig wlan0 down` `sudo ifconfig wlan0 up`	
Чтобы проверить, правильно ли обнаружена карта Wi-Fi, сначала выполните команду iwconfig	root@nanopiair:~# iwconfig wlan0 IEEE 802.11 ESSID:off/any Mode:Managed Access Point: Not-Associated Tx-Power=31 dBm Retry short limit:7 RTS thr:off Fragment thr:off Encryption key:off Power Management:on lo no wireless extensions.

	dummy0 no wireless extensions.
Команда iwlist предоставляет дополнительную информацию о беспроводных устройствах	root@nanopiair:~# iwlist Usage: iwlist [interface] scanning [essid NNN] [last] [interface] frequency [interface] channel [interface] bitrate [interface] rate [interface] encryption [interface] keys [interface] power [interface] txpower [interface] retry [interface] ap [interface] accesspoints [interface] peers [interface] event [interface] auth [interface] wpakeys [interface] genie [interface] modulation

```
root@nanopiair:~# iwlist wlan0 scan
wlan0     Scan completed :
          Cell 01 - Address: 90:9A:4A:C5:54:6A
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=16/70  Signal level=-94 dBm
                    Encryption key:on
                    ESSID:"Meeting-room"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 12 Mb/s; 24 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=0000000000000000
                    Extra: Last beacon: 68ms ago
                    IE: Unknown: 000C4D656574696E672D726F6F6D
                    IE: Unknown: 010882848B961224486C
                    IE: Unknown: 030101
                    IE: Unknown: 32040C183060
                    IE: Unknown: 05050001014198
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    IE: Unknown:
DD310050F204104A000110104400010210470010388330923092 18839C77909A4AC554C4103C000101104900060037 2A000120
                    IE: Unknown: 2A0104
                    IE: Unknown: 2D1A6E1017FFFF000001000000000000000000000000000000000000
                    IE: Unknown: 3D160105060000000000000000000000000000000000000000000000
                    IE: Unknown: 4A0E14000A002C01C800140005001900
                    IE: Unknown: DD180050F2020101000003A4000027A4000042435E0062322F00
                    IE: Unknown: 0B05040000127A
                    IE: Unknown: DD07000C4300000000
          Cell 02 - Address: A0:A3:F0:D1:AA:CE
                    Channel:11
                    Frequency:2.462 GHz (Channel 11)
                    Quality=20/70  Signal level=-90 dBm
                    Encryption key:on
                    ESSID:"NPKOSV"
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
                              36 Mb/s; 48 Mb/s; 54 Mb/s
                    Mode:Master
                    Extra:tsf=0000000000000000
                    Extra: Last beacon: 68ms ago
                    IE: Unknown: 00064E504B4F5356
                    IE: Unknown: 01080C1218243048606C
                    IE: Unknown: 03010B
                    IE: Unknown: 050400010000
                    IE: Unknown: 2A0100
                    IE: Unknown: 2D1AEE1902FFFF000000000000000000000000000000000000000000
                    IE: Unknown: 3D160B000200000000000000000000000000000000000000000000
                    IE: Unknown: DD0700E04C02021000
                    IE: IEEE 802.11i/WPA2 Version 1
```

```
                    Group Cipher : CCMP
                    Pairwise Ciphers (1) : CCMP
                    Authentication Suites (1) : PSK
                IE: Unknown: DD180050F2020101800003A4000027A4000042435E0062322F00
                IE: Unknown:
DD2D0050F204104A000110104400010210490006003 72A000120105400080001 0050F2000000101100054E504B3736
                IE: Unknown: DD080050F21102000000
                IE: Unknown: DD0E0050F2122B000600A0A3F0D1AACE
                IE: Unknown: DD17506F9A0902020025BB030600A2A3F0D1AACE0C02000000
                IE: Unknown: 46053200010000
                IE: Unknown: 7F09000000000000000000
```

nmcli d wifi connect LinuxHint password morochita

root@nanopiair:~# nmcli device wifi connect "NPKOSV" password "E?9+7e%q" ifname wlan0

	root@nanopiair:~# nmcli device wifi * SSID MODE CHAN RATE SIGNAL BARS SECURITY NPKOSV Infra 11 54 Mbit/s 30 * WPA2 Meeting-room Infra 1 54 Mbit/s 15 * WPA2
	Error: Connection activation failed: (7) Secrets were required, but not provided.
1	nmcli connection delete *NPKOSV*
	root@nanopiair:~# nmcli connection delete NPKOSV Connection 'NPKOSV' (68eda11c-2390-47c3-a3ae-74fdd2908996) successfully deleted. root@nanopiair:~# nmcli connection delete NPKOSV Error: unknown connection 'NPKOSV'. Error: cannot delete unknown connection(s): 'NPKOSV'.
2	
	root@nanopiair:/# l /etc/NetworkManager/conf.d/ total 8 -rw-r--r-- 1 root root 152 Apr 7 07:27 default-wifi-powersave-on.conf -rw-r--r-- 1 root root 39 Apr 7 07:41 wifi_rand_mac.conf

3	cat /etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
	[connection]
	# Values are 0 (use default), 1 (ignore/don't touch), 2 (disable) or 3 (enable).
	wifi.powersave = 3

	root@nanopiair:/# cat /etc/NetworkManager/conf.d/wifi_rand_mac.conf [device] wifi.scan-rand-mac-address=no
	systemctl restart NetworkManager

Step 4 – Managing the Nginx Process

`sudo systemctl stop nginx`	To stop your web server,
`sudo systemctl start nginx`	To start the web server when it is stopped
`sudo systemctl restart nginx`	To stop and then start the service again
`sudo systemctl reload nginx`	If you are making configuration changes, Nginx can often reload without dropping connections.
`sudo systemctl disable nginx`	By default, Nginx is configured to start automatically when the server boots. If this is not what you want, you can disable this
`sudo systemctl enable nginx`	To re-enable the service to start up at boot
`sudo systemctl status nginx`	
service nginx `stop / start / restart / reload / status`	

SD-Card Operations

root@nanopiair:/# lsblk	root@nanopiair:/var/www# lsblk NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT mmcblk2 179:0 0 7.3G 0 disk └─mmcblk2p1 179:1 0 7.2G 0 part / mmcblk2boot0 179:8 0 4M 1 disk mmcblk2boot1 179:16 0 4M 1 disk zram0 254:0 0 50M 0 disk /var/log zram1 254:1 0 245.9M 0 disk [SWAP] root@nanopiair:/var/www# lsblk NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT mmcblk2 179:0 0 7.3G 0 disk └─mmcblk2p1 179:1 0 7.2G 0 part / mmcblk2boot0 179:8 0 4M 1 disk mmcblk2boot1 179:16 0 4M 1 disk mmcblk0 179:24 0 14.7G 0 disk └─mmcblk0p1 179:25 0 14.7G 0 part zram0 254:0 0 50M 0 disk /var/log zram1 254:1 0 245.9M 0 disk [SWAP]
root@nanopiair:/var/www# blkid /dev/mmcblk2p1: UUID="6dd62d92-d2ba-40a9-8d79-4d74eb96d53d" TYPE="ext4" PARTUUID="304157cb-01" /dev/zram0: LABEL="log2ram" UUID="d4bc885f-2269-409e-9a28-15430833d380" TYPE="ext4" /dev/mmcblk2: PTUUID="304157cb" PTTYPE="dos" /dev/zram1: UUID="0b2ad13f-43a0-465f-85a1-5848d788024d" TYPE="swap" /dev/mmcblk0: PTUUID="47cc951a" PTTYPE="dos" /dev/mmcblk0p1: UUID="4E7B-B9EE" TYPE="vfat" PARTUUID="47cc951a-01"	

mount -t vfat -o rw /dev/mmcblk0p1 /media	
...	
root@nanopiair:/# umount /dev/mmcblk0p1	
# dd if=/dev/mmcblk2 of=/dev/mmcblk0 bs=4096 status=progress # dd if=/dev/zero of=/dev/mmcblk1 bs=1M count=32 status=progress	

download over github

Nanopi	root@nanopiair:/home/stepan/redmondPage# pwd /home/stepan/redmondPage root@nanopiair:/home/stepan/redmondPage# ls -la total 36 drwxr-xr-x 2 root root 4096 Mar 30 05:32 . drwxr-xr-x 4 stepan stepan 4096 Mar 30 05:06 .. -rw-r--r-- 1 root root 11919 Mar 30 05:32 FASTRUN.HTML -rw-r--r-- 1 root root 5240 Mar 30 05:32 M300S.HTML -rw-r--r-- 1 root root 2666 Mar 30 05:32 PROGRAMS.HTML -rw-r--r-- 1 root root 1396 Mar 30 05:32 RECIPE.HTML
github.com	https://github.com/StepanOsotov/RedmondPage
Получить ссылку	https://github.com/StepanOsotov/RedmondPage.git
cd /home/stepan	git clone https://github.com/StepanOsotov/RedmondPage.git

DNS

nano /etc/hosts	root@nanopiair:/# cat /etc/hosts 127.0.0.1 localhost nanopiair ::1 localhost nanopiair ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters
У меня что то не работает	root@nanopiair:/# cat /etc/hosts 127.0.0.1 localhost nanopiair ::1 localhost nanopiair ip6-localhost ip6-loopback 192.168.137.113 osotovsv.ru fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters

Step 5 – Setting Up Server Blocks

Стартовая страница (index.nginx-debian.html) находится в директории	`/var/www/html`
Можно создавать свою иерархию папок	`/var/www/my_folder`

`sudo mkdir -p /var/www/your_domain/html`	`mkdir -p /var/www/osv/html`
-p или –parent Создать все директории, которые указаны внутри пути. Если какая-либо директория существует, то предупреждение об этом не выводится.	

Смена прав			
Права nginx страницы	root@nanopiair:/# l /var/www/html/ total 4 -rw-r--r-- 1 root root 612 Mar 29 04:49 index.nginx-debian.html - / rw- / r-- / r-- пользователь группа остальные		
Права моей страницы	root@nanopiair:/# l /var/www/osv/html/ total 8 -rwxr-xr-x 1 root root 5240 Mar 30 05:32 index.html		
chmod буквеные права	chmod X Y Z name_file		
	X	Y	Z
	u : пользователь g : группа o : остальные a : все	- : запрещение + : разрешение = : присвоение	r : чтение w : запись x : исполнение
chmod цифровые права	chmod A B C name_file		
	A	B	C
	Пользователь	Группа	Остальные
	7 — все разрешено 6 — чтение и запись 5 — чтение и использованиеё 4 — только чтение 0 — все запрещено		
chmod a-x	chmod a-x /var/www/osv/html/index.html		
	root@nanopiair:/# chmod a-x /var/www/osv/html/index.html root@nanopiair:/# l /var/www/osv/html/ total 8 -rw-r--r-- 1 root root 5240 Mar 30 05:32 index.html		

Next, assign ownership of the directory with the $USER environment variable, which should reference your current system user:

`sudo chown -R $USER:$USER /var/www/your_domain/html`	
root@nanopiair:/# chown -R $USER:$USER /var/www/osv/html	
root@nanopiair:/# chown -R -v $USER:$USER /var/www/osv/html ownership of '/var/www/osv/html/index.html' retained as root:root ownership of '/var/www/osv/html' retained as root:root	

The permissions of your web root should be correct if you haven't modified your umask value, but you can make sure by typing:

`sudo chmod -R 755 /var/www/your_domain`
root@nanopiair:/# chmod -R -v 755 /var/www/osv mode of '/var/www/osv' retained as 0755 (rwxr-xr-x) mode of '/var/www/osv/html' retained as 0755 (rwxr-xr-x) mode of '/var/www/osv/html/index.html' changed from 0644 (rw-r--r--) to 0755 (rwxr-xr-x)

создай или скопируй index.html файл

/home/stepan/redmondPage	/var/www/osv/html
	/var/www/osv/html/index.html

```
<html>
  <head>
    <title>Welcome to OSV</title>
  </head>
  <body>
    <h1>Success! Your Nginx server is successfully configured for <em>OSV</em>. </h1>
    <p>This is a sample page.</p>
  </body>
</html>
```

In order for Nginx to serve this content, you must create a server block with the correct directives that point to your custom web root. Instead of modifying the default configuration file directly, make a new one at /etc/nginx/sites-available/your_domain :

`sudo nano /etc/nginx/sites-available/your_domain`	

/etc/nginx/sites-available/your_domain

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

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

        server_name your_domain www.your_domain;

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

```
nano /etc/nginx/sites-available/osv
```

```
root@nanopiair:/# cat /etc/nginx/sites-available/osv
server {

        listen 192.168.137.113:80;
#       listen 80;          #IPv4
#       listen [::]:80;     #IPv6

        server_name osv.ru;

        root /var/www/osv/html;

        index index.html index.htm;

        location / {

                try_files $uri $uri/ =404;
#               try_files $uri $uri.html $uri/ /osv/index.html $uri/ =404;
        }

}
```

```
server
{
  listen 192.168.137.113:80;
  root /var/www/osv/html;
  index m300.html;
  location /
  {
    try_files $uri $uri/ =404;
  }
}
```

```
server
{
  listen 192.168.137.113/fastrun:80;
#  listen 192.168.137.113/programs:80;
#  listen 192.168.137.113/recipe:80;
  root /var/www/osv/html;
  index fastrun.html;
#  index programs.html;
#  index recipe.html;
  location /
  {
    try_files $uri $uri/ =404;
  }
}
```

```
root@nanopiair:/etc/nginx/sites-available# cat default
...
server {
        listen 80 default_server;
        listen [::]:80 default_server;
...
        root /var/www/html;

...
        index index.html index.htm index.nginx-debian.html;

        server_name _;

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

...
}

...
```

<table>
<tr><td>директивы</td><td>server {
listen 192.168.137.113:80;
server_name osv.ru;
root /var/www/osv/html;
index index.html index.htm;
location / {
try_files $uri $uri/ =404;
}
}</td><td>server
{
listen 192.168.137.113:80;
root /var/www/osv/html;
index m300s.html;
location /
{
try_files $uri $uri/ =404;
}
}</td></tr>
<tr><td>server</td><td colspan="2">Синтаксис:server { ... }
Умолчание:—
Задаёт конфигурацию для виртуального сервера. Чёткого разделения виртуальных серверов на IP-based (на основании IP-адреса) и name-based (на основании поля “Host” заголовка запроса) нет. Вместо этого директивами listen описываются все адреса и порты, на которых нужно принимать соединения для этого сервера, а в директиве server_name указываются все имена серверов.

Блок server — это часть конфигурации Nginx, которая определяет виртуальный сервер, используемый для обработки запросов заданного типа. Администраторы часто настраивают несколько блоков server и определяют, какой из них будет отвечать за конкретное соединение на основании запрошенного доменного имени, порта и IP-адреса.</td></tr>
<tr><td>location</td><td colspan="2">Синтаксис:location [= | ~ | ~* | ^~] uri { ... }
location @имя { ... }
Умолчание:—
Кроме того, с помощью модификатора “=” можно задать точное совпадение URI и location. При точном совпадении поиск сразу же прекращается. Например, если запрос “/” случается часто, то указав “location = /”, можно ускорить обработку этих запросов, так как поиск прекратится после первого же сравнения. Очевидно, что такой location не может иметь вложенные location’ы.

Блок location располагается внутри блока server и определяет, как Nginx будет обрабатывать запросы различных ресурсов и URI для родительского сервера. Администратор, использующий эти блоки, может разделить пространство URI любым удобным способом. Это чрезвычайно гибкая модель.</td></tr>
<tr><td>listen</td><td colspan="2">Директива listen обычно определяет IP-адрес и порт, на которые отвечает серверный блок.
Директиву listen можно задать следующим образом:

●Сочетание IP-адреса и порта.
●Отдельный IP-адрес, который будет прослушивать порт 80 по умолчанию.
●Одиночный порт, который прослушивает каждый интерфейс этого порта.
●Путь к сокету Unix.
например:
listen 127.0.0.1:8000;
listen 127.0.0.1;
listen 8000;
listen *:8000;
listen localhost:8000;
IPv6-адреса задаются в квадратных скобках:
listen [::]:8000;
listen [::1];</td></tr>
<tr><td>try_files</td><td colspan="2">Директива try_files – очень полезный инструмент, который проверяет наличие файлов в заданном порядке и использует первый найденный файл для обработки запроса.</td></tr>
<tr><td>root</td><td colspan="2">Синтаксис: root путь;
Умолчание: root html;</td></tr>
</table>

	Задаёт корневой каталог для запросов.
index	Синтаксис: **index** *файл* . . . ; Умолчание: index index.html; Определяет файлы, которые будут использоваться в качестве индекса. В имени файла можно использовать переменные. Наличие файлов проверяется в порядке их перечисления. В конце списка может стоять файл с абсолютным путём.

nano /etc/nginx/sites-available/osv
 	
nginx -t
 	
systemctl restart nginx.service
	server
	{
	listen 192.168.137.113:80;
	root /var/www/osv/html;
	location = / {
	index m300s.html;
	try_files $uri $uri/ =404;
	}
	location /fastrun {
	index fastrun.html;
	try_files $uri $uri/fastrun/ =404;
	}
	location /programs {
	index programs.html;
	try_files $uri $uri/programs/ =404;
	}
	location /recipe {
	index recipe.html;
	try_files $uri $uri/recipe/ =404;
	}
	}

	http://192.168.137.113/ http://192.168.137.113/fastrun.html http://192.168.137.113/programs.html http://192.168.137.113/recipe.html

Next, enable this server block by creating a symbolic link to your custom configuration file inside the sites-enabled directory, which Nginx reads from during startup:

```
root@nanopiair:/# ll /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 Mar 29 04:49 .
drwxr-xr-x 8 root root 4096 Mar 29 04:49 ..
lrwxrwxrwx 1 root root   34 Mar 29 04:49 default -> /etc/nginx/sites-available/default
```

```
sudo ln -s /etc/nginx/sites-available/your_domain /etc/nginx/sites-enabled/
```

```
ln -s /etc/nginx/sites-available/osv /etc/nginx/sites-enabled/
```

Чтобы удалить символическую ссылку, используйте команду rm или unlink, за которой следует имя символической ссылки в качестве аргумента.

проверить целый каталог на наличие ссылок	root@nanopiair:/# ls -l /etc/nginx/sites-enabled/ total 0 lrwxrwxrwx 1 root root 34 Mar 29 04:49 default -> /etc/nginx/sites-available/default lrwxrwxrwx 1 root root 34 Mar 30 06:37 redmond -> /etc/nginx/sites-available/redmond
Удалить с помощью rm	Команда rm позволяет удалять любые файлы в файловой системе Linux, в том числе и ссылки. А значит, она подходит для нашей задачи. Здесь важно упомянуть, что в результате удаления ссылки оригинальный файл или директория затронуты не будут. С выводом данных rm -v /etc/nginx/sites-enabled/redmond

```
root@nanopiair:/# ln -s /etc/nginx/sites-available/osvredmond.com /etc/nginx/sites-enabled/
root@nanopiair:/# ll /etc/nginx/sites-enabled/
```

```
total 8
drwxr-xr-x 2 root root 4096 Mar 30 06:37 .
drwxr-xr-x 8 root root 4096 Mar 29 04:49 ..
lrwxrwxrwx 1 root root   34 Mar 29 04:49 default -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root   34 Mar 30 06:37 osv.com -> /etc/nginx/sites-available/osv.com
```

Your server now has two server blocks enabled and configured to respond to requests based on their listen and server_name

your_domain: Will respond to requests for your_domain and www.your_domain.
default: Will respond to any requests on port 80 that do not match the other two blocks.

To avoid a possible hash bucket memory problem that can arise from adding additional server names to your configuration, it is necessary to adjust a single value in the /etc/nginx/nginx.conf file. Open the file:

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

```
root@nanopiair:/# cat /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
	worker_connections 768;
	# multi_accept on;
}

http {

	##
	# Basic Settings
	##

	sendfile on;
	tcp_nopush on;
	tcp_nodelay on;
	keepalive_timeout 65;
	types_hash_max_size 2048;
	# server_tokens off;

	server_names_hash_bucket_size 64;
	# server_name_in_redirect off;

	include /etc/nginx/mime.types;
	default_type application/octet-stream;

	##
	# SSL Settings
	##

	ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
	ssl_prefer_server_ciphers on;

	##
	# Logging Settings
	##
```

```
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##

        gzip on;
        gzip_disable "msie6";

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss
text/javascript;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}


#mail {
#       # See sample authentication script at:
#       # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen     localhost:110;
#               protocol   pop3;
#               proxy      on;
#       }
#
#       server {
#               listen     localhost:143;
#               protocol   imap;
#               proxy      on;
#       }
#}
```

Find the server_names_hash_bucket_size directive and remove the # symbol to uncomment the line:

/etc/nginx/nginx.conf

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

Next, test to make sure that there are no syntax errors in any of your Nginx files:

`sudo nginx -t`	If there aren't any problems, the following is the output:
	`Output`
	`nginx: the configuration file /etc/nginx/nginx.conf syntax is ok`
	`nginx: configuration file /etc/nginx/nginx.conf test is successful`

	root@nanopiair:/# nginx -t nginx: the configuration file /etc/nginx/nginx.conf syntax is ok nginx: configuration file /etc/nginx/nginx.conf test is successful

Once your configuration test passes, restart Nginx to enable your changes:

`sudo systemctl restart nginx`	
root@nanopiair:/# systemctl restart nginx.service	

ss -tnlp	root@nanopiair:/# ss -tnlp State Recv-Q Send-Q Local Address:Port Peer Address:Port LISTEN 0 128 *:9001 *:* users:(("mosquitto",pid=775,fd=7)) LISTEN 0 128 *:80 *:* users:(("nginx",pid=3317,fd=6), ("nginx",pid=3316,fd=6),("nginx",pid=3315,fd=6),("nginx",pid=3314,fd=6),("nginx",pid=2632,fd=6)) LISTEN 0 128 *:22 *:* users:(("sshd",pid=836,fd=3)) LISTEN 0 100 *:1883 *:* users:(("mosquitto",pid=775,fd=8)) LISTEN 0 128 :::80 :::* users:(("nginx",pid=3317,fd=7), ("nginx",pid=3316,fd=7),("nginx",pid=3315,fd=7),("nginx",pid=3314,fd=7),("nginx",pid=2632,fd=7)) LISTEN 0 128 :::22 :::* users:(("sshd",pid=836,fd=4)) LISTEN 0 100 :::1883 :::* users:(("mosquitto",pid=775,fd=9))
netstat -tnlp	root@nanopiair:/# netstat -tnlp Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name tcp 0 0 0.0.0.0:9001 0.0.0.0:* LISTEN 775/mosquitto tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 2632/nginx: master tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 836/sshd tcp 0 0 0.0.0.0:1883 0.0.0.0:* LISTEN 775/mosquitto tcp6 0 0 :::80 :::* LISTEN 2632/nginx: master tcp6 0 0 :::22 :::* LISTEN 836/sshd tcp6 0 0 :::1883 :::* LISTEN 775/mosquitto

Step 6 – Getting Familiar with Important Nginx Files and Directories

Now that you know how to manage the Nginx service itself, you can take some time to familiarize yourself with a few important directories and files.

Content

- `/var/www/html`: The actual web content, which by default only consists of the default Nginx page you saw earlier, is served out of the `/var/www/html` directory. This can be changed by altering Nginx configuration files.

Server Configuration

- `/etc/nginx`: The Nginx configuration directory. All of the Nginx configuration files reside here.
- `/etc/nginx/nginx.conf`: The main Nginx configuration file. This can be modified to make changes to the Nginx global configuration.
- `/etc/nginx/sites-available/`: The directory where per-site server blocks can be stored. Nginx will not use the configuration files found in this directory unless they are linked to the `sites-enabled` directory. Typically, all server block configuration is done in this directory, and then enabled by linking to the other directory.
- `/etc/nginx/sites-enabled/`: The directory where enabled per-site server blocks are stored. Typically, these are created by linking to configuration files found in the `sites-available` directory.
- `/etc/nginx/snippets`: This directory contains configuration fragments that can be included elsewhere in the Nginx configuration. Potentially repeatable configuration segments are good candidates for refactoring into snippets.

Server Logs

- `/var/log/nginx/access.log`: Every request to your web server is recorded in this log file unless Nginx is configured to do otherwise.
- `/var/log/nginx/error.log`: Any Nginx errors will be recorded in this log.
