Instalación y configuración de servidor web Nginx:

Para instalar y configurar el servidor deberemos actualizar reposistorios. Una vez actualizados instalaremos el servidor Nginx. Utilizaremos el siguiente comando: “sudo apt install nginx”

```
a24gadilu@debian:~$ sudo apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  nginx-common
Paquetes sugeridos:
  fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
  nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Comprobamos que este instalado y funcione correctamente con el comando: “sytemctl status gninx”

```
a24gadilu@debian:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-09-23 10:11:45 CEST; 7min ago
       Docs: man:nginx(8)
    Process: 823 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 824 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 849 (nginx)
      Tasks: 3 (limit: 2306)
     Memory: 2.5M
        CPU: 23ms
     CGroup: /system.slice/nginx.service
             ├─849 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─851 "nginx: worker process"
             └─852 "nginx: worker process"

sep 23 10:11:45 debian systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
sep 23 10:11:45 debian systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
a24gadilu@debian:~$
```

Creamos las carpetas del sitio web de la siguiente manera:

“sudo mkdir -p /var/www/html”

```
a24gadilu@debian:/$ sudo mkdir -p /var/www/a24gadilu/html
a24gadilu@debian:/$ cd /var/www/a24gadilu/html
```

Dentro de esta carpeta tenemos que clonar el siguiente repositorio: “git clone https://github.com/cloudacademy/static-website-example”

```
a24gadilu@debian:/var/www/a24gadilu/html$ sudo git clone https://github.com/cloudacademy/static-website-example
Clonando en 'static-website-example'...
remote: Enumerating objects: 69, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 69 (delta 8), reused 8 (delta 8), pack-reused 45 (from 1)
Recibiendo objetos: 100% (69/69), 668.08 KiB | 1.45 MiB/s, listo.
Resolviendo deltas: 100% (9/9), listo.
a24gadilu@debian:/var/www/a24gadilu/html$
```

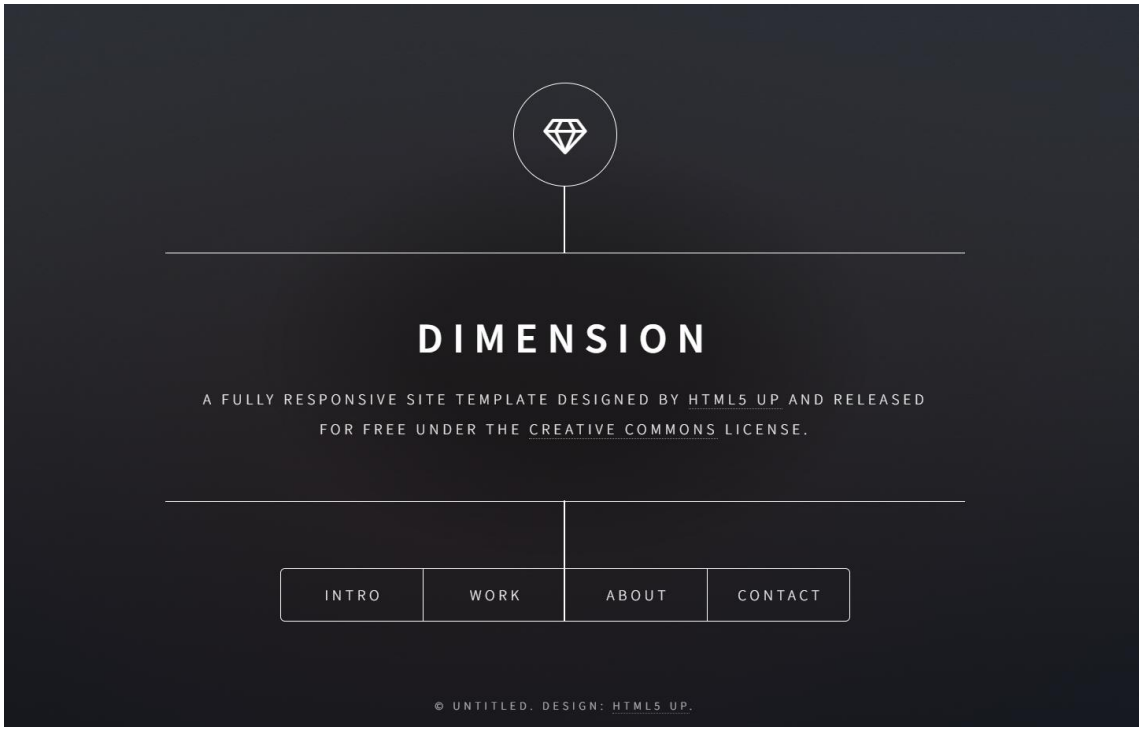
Ahora pondremos el propietario de la carpeta y todo lo que haya dentro, será el usuario a24gadilu que será el usuario del servidor web. Con el siguiente comando: “sudo chown -R www-data:www-data /var/www/a24gadilu/html”

```
a24gadilu@debian:/$ sudo chown -R a24gadilu:a24gadilu /var/www/a24gadilu/html
[sudo] contraseña para a24gadilu:
```

Y le daremos permisos para que no nos de error de acceso no autorizado al entrar al sitio web. Con el siguiente comando: “sudo chmod -R 755 /var/www/nombre_web”

```
a24gadilu@debian:/$ sudo chmod -R 755 /var/www/a24gadilu
a24gadilu@debian:/$
```

Para ver que esta correcta la instalación y configuración deberemos acceder desde nuestro navegador a través de la dirección ip en este caso 192.168.116.3 que es la red) de nuestra máquina.



Ahora configuraremos el servidor para ello deberemos crear un dominio web con nuestro nombre, con el siguiente comando “sudo nano /etc/nginx/sites-available/a24gadilu”.

Deberemos configurarlo en el puerto 80 y ponerle nuestro nombre al servidor.

```
 GNU nano 7.2                                  a24gadilu
erver {
        listen 80;
        server_name a24gadilu.local;

        root /var/www/a24gadilu/html/static-website-example;
        index index.html index.htm;

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


                                [ 11 líneas leídas ]
G Ayuda      ^O Guardar     ^W Buscar      ^K Cortar      ^T Ejecutar    ^C Ubicación   M-U Deshacer   M-A Po
X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar  ^/ Ir a línea  M-E Rehacer    M-6 Co
```

Y crearemos un archivo simbólico entre este archivo y el de sitios que están habilitados, para que se dé de alta automáticamente con este comando:

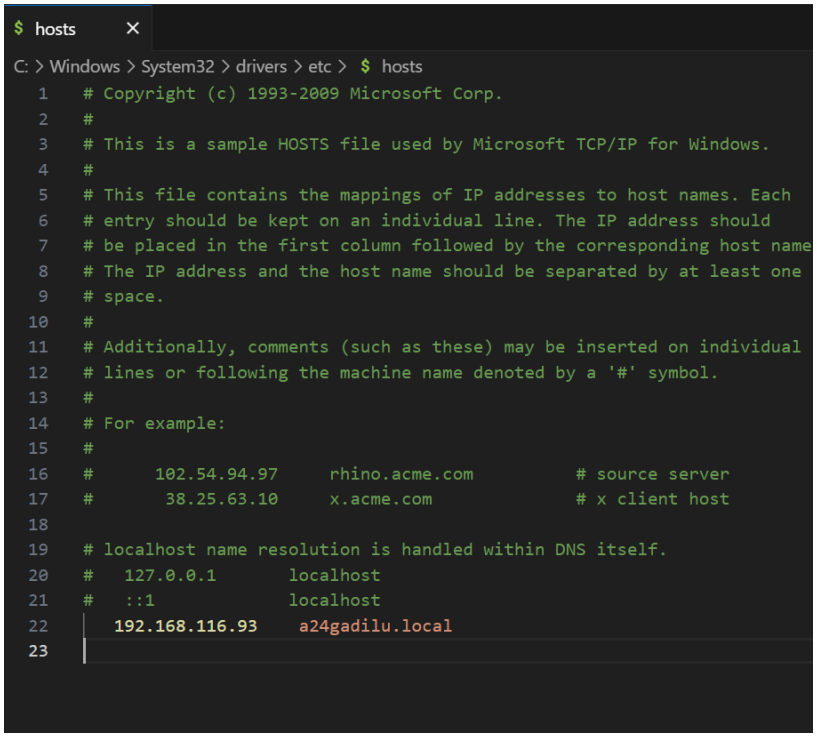
“sudo ln -s /etc/nginx/sites-available/a24gadilu /etc/nginx/sites-enabled/”

```
a24gadilu@debian:~$ sudo ln -s /etc/nginx/sites-available/a24gadilu /etc/nginx/sites-enabled/
```

Una vez creado reiniciamos el servicio con este comando: “sudo service nginx restart”

```
a24gadilu@debian:~$ sudo service nginx restart
Restarting nginx: nginx.
a24gadilu@debian:~$
```

Ahora debemos configurar nuestra maquina anfitriona porque no tenemos DNS y pa ra ello deberemos asignar la ip en el archivo C:\Windows\System32\drivers\etc\hosts



Ahora podremos aceder a la pagina con nuestro nombre de dominio: