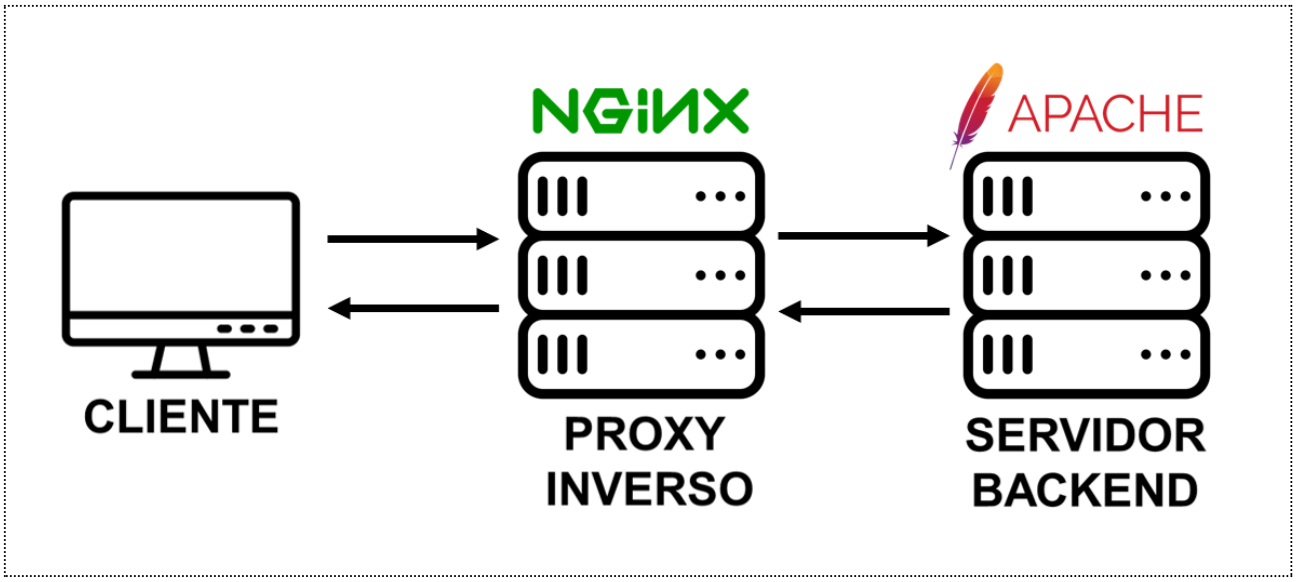
PROXY INVERSO CON NGINX



Práctica: Proxy inverso.
Fecha: 20/02/24.
Autor/a: BUDDY.

IPs:	SO:	Equipos:	Finalidad:	Páginas:
lix: 172.16.1.40 **revproxy:** 172.16.1.41 **ateez:** 172.16.1.30 **zerowave:** 172.16.1.11	Linux (lxc) Linux (Ubuntu)	**lix** (página estática) **revproxy** (proxy inverso) **ateez** (página estática 2) **zerowave** (comprobar)	Implantar un proxy inverso	23

ÍNDICE:

0. Introducción

Antes de entrar a la práctica de cómo implantar un proxy inverso, vamos a ver **qué es un proxy inverso**.

Un **proxy inverso es** un **tipo de servidor que actúa como intermediario entre los usuarios y uno o más servidores web**. Los **usuarios**, **en lugar** de **comunicarse directamente con** los **servidores finales**, todas las **solicitudes** de los usuarios **pasan primero a través** del **proxy inverso**.

Dicho lo que es un proxy inverso, pasamos a ver **cómo** se **implementaría** un **proxy inverso** en un **lxc**.

En esta práctica se utilizarán 3 lxc y un Ubuntu cliente, ***"lix"*** donde estará la primera página estática, ***"revproxy"*** donde se configurará el proxy inverso, ***"ateez"*** donde estará la segunda página estática y ***"zerowave"*** que será el Ubuntu cliente donde comprobaremos que todo ha salido correctamente.

1. Actualizar el sistema

En *"lix"*:

Antes de nada, vamos a ejecutar el comando: ***"sudo apt update && sudo apt upgrade"***, de dicha forma, **actualizamos los paquetes** y **evitamos futuros posibles problemas** por falta de actualizar los paquetes del sistema.

```
root@lix:~# sudo apt update && sudo apt upgrade
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1377 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1049 kB]
Fetched 2655 kB in 3s (800 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@lix:~#
```

2. Instalar Nginx en *"lix"*

A continuación vamos a instalar Nginx completo, para ello vamos a ejecutar el comando: ***"sudo apt-get install nginx-full"***.

```
root@lix:~# sudo apt-get install nginx-full
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libbrotli1 libdeflate0 libfontconfig1 libfreetype6 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-subs-filter
  libnginx-mod-http-upstream-fair libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 libxslt1.1 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libbrotli1 libdeflate0 libfontconfig1 libfreetype6 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-subs-filter
  libnginx-mod-http-upstream-fair libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 libxslt1.1 nginx-common nginx-core nginx-full
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 3638 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Escribimos una **"y"** y **seguimos esperando** a que se **instale Nginx**.

Mientras Nginx se instala, te cuento **por qué hemos puesto "nginx-full"** a la hora de instalarlo.

Resulta que a la hora de instalar Nginx, este nos ofrece 3 versiones, la ***"light"***, la ***"full"*** y la ***"extra"***.

- ***"nginx-light":*** Se instala la **versión básica y ligera de Nginx**, **incluye** únicamente **lo mínimo y** los **módulos necesarios para manejar solicitudes HTTP básicas**. Esta versión está **optimizada para un bajo consumo de recursos**, lo que hace que sea **ideal para servidores con recursos limitados** y **donde se necesite una instalación sencilla**.

- ***"nginx-full":*** Se instala la **versión completa**, **con todos los módulos**. Esta versión **ofrece soporte para características adicionales como SSL**, **proxy inverso**, **módulos de seguridad más avanzados**, etc.

- ***"nginx-extra":*** Esta **versión** es la **extendida de Nginx** e **incluye todos los módulos disponibles**, **incluidos algunos módulos que no están presentes en las otras versiones** comentadas anteriormente. **Algunos módulos adicionales** pueden **proporcionar funcionalidades avanzadas** como: la **integración con diferentes sistemas de almacenamiento en caché**, **autenticación adicional** o **capacidades de red más avanzadas**.

Depende de tus necesidades, debes **considerar que versión vas a instalar.**
En mi caso, instalaré la versión ***"full"***.

Ahora, ¿cómo comprobamos que nginx está instalado correctamente?, podemos mirar la versión instalada con el comando: ***"nginx -v"***.

```
root@lix:~# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
root@lix:~#
```

Otra cosa que podemos hacer, es utilizar ***"systemctl status nginx"*** para ver el estado de este.

```
root@lix:~# systemctl status nginx
* 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 Sat 2024-02-17 09:15:53 UTC; 1min 50s ago
       Docs: man:nginx(8)
    Process: 11781 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 11782 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 11923 (nginx)
      Tasks: 2 (limit: 19068)
     Memory: 4.1M
        CPU: 41ms
     CGroup: /system.slice/nginx.service
             |-11923 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             `-11926 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "">

Feb 17 09:15:53 lix systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 17 09:15:53 lix systemd[1]: Started A high performance web server and a reverse proxy server.
```

Una vez veamos que está activo, eso es que todo se instaló correctamente.

También podemos ejecutar el comando: ***"nginx -t"*** para ver que todo esté bien.

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

3. Configurar página estática

Llega el momento de **configurar** la **página estática con Nginx**, para ello nos **vamos a cambiar de directorio** y nos vamos a ir **a**: ***"/etc/nginx/sites-available"***.

```
root@lix:/# cd /etc/nginx/sites-available
root@lix:/etc/nginx/sites-available# ls
default
root@lix:/etc/nginx/sites-available#
```

Si listamos el contenido, podemos ver que **hay una página "default"**. Esa página "default", **Nginx la crea por defecto para facilitarle la configuración** al que esté configurando todo, pero **no es necesario usarla**.

En mi caso, **voy a crear una página desde cero** y le vamos a llamar ***"appbuddy"***.

Para crear la página utilizaremos el comando: ***"nano appbuddy"***.

Ahora, ¿qué debemos de escribir en este fichero?, aquí va a ir la configuración de la página estática.

La configuración será la siguiente:

```
  GNU nano 6.2                                                  appbuddy *
server {
    listen 80 default_server;
    server_name appbuddy.com;

    root /var/www/html/buddy;
    error_log /var/log/nginx/app-server-error.log notice;
    index demo-index.html index.html;
    expires -1;


    sub_filter_once off;
    sub_filter 'server_hostname' '$hostname';
    sub_filter 'server_address'  '$server_address:$server_port';
    sub_filter 'server_url'      '$request_uri';
    sub_filter 'remote_addr'     '$remote_addr:$remote_port';
    sub_filter 'server_date'     '$time_local';
    sub_filter 'client_browser'  'http_user_agent';
    sub_filter 'request_id'      '$request_id';
    sub_filter 'nginx_version'   '$nginx_version';
    sub_filter 'document_root'   '$document_root';
    sub_filter 'proxied_for_ip'  '$http_x_forwarded_for';

    location / {
      try_files $uri $uri/ /index.html /demo-index.html;
    }
}
```

Una vez escribamos esta configuración, guardamos y salimos.

Podemos hacer un ***"cat /etc/nginx/sites-available/appbuddy"*** para **ver el contenido** del fichero **sin tener que entrar con *"nano"***.

```
  GNU nano 6.2                                          appbuddy *
server {
    listen 80 default_server;
    server_name appbuddy.com;

    root /var/www/html/buddy;
    error_log /var/log/nginx/app-server-error.log notice;
    index demo-index.html index.html;
    expires -1;


    sub_filter_once off;
    sub_filter 'server_hostname' '$hostname';
    sub_filter 'server_address'  '$server_addr:$server_port';
    sub_filter 'server_url'      '$request_uri';
    sub_filter 'remote_addr'     '$remote_addr:$remote_port';
    sub_filter 'server_date'     '$time_local';
    sub_filter 'client_browser'  'http_user_agent';
    sub_filter 'request_id'      '$request_id';
    sub_filter 'nginx_version'   '$nginx_version';
    sub_filter 'document_root'   '$document_root';
    sub_filter 'proxied_for_ip'  '$http_x_forwarded_for';

    location / {
      try_files $uri $uri/ /index.html /demo-index.html;
    }
}
```

Nos dirigimos a: ***"/etc/nginx/sites-enabled"***.

```
root@lix:/etc/nginx/sites-available# ls
appbuddy  default
root@lix:/etc/nginx/sites-available# cd ..
root@lix:/etc/nginx# cd sites-enabled
root@lix:/etc/nginx/sites-enabled# ls
default
root@lix:/etc/nginx/sites-enabled#
```

En los sitios habilitados, está el ***"default"***. Antes **hemos especificado que *"appbuddy"* sea el sitio por defecto**, por lo que, **debemos eliminar este *"default"*** para que no haya problemas.

```
root@lix:/etc/nginx/sites-enabled# ls
default
root@lix:/etc/nginx/sites-enabled# rm default
root@lix:/etc/nginx/sites-enabled# ls
root@lix:/etc/nginx/sites-enabled#
```

Para borrar el fichero ***"default"***, **utilizamos** el comando: ***"rm default"*** y **cuando volvamos a lista el contenido del directorio, debe estar vacío.**

A continuación **vamos a realizar un enlace simbólico** desde el ***"sites-available"*** al ***"sites-enabled"***, para ello vamos a ejecutar el comando: ***"ln -s /etc/nginx/sites-available/appbuddy /etc/nginx/sites-enabled/"***.

```
root@lix:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/appbuddy /etc/nginx/sites-enabled/
root@lix:/etc/nginx/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 35 Feb 17 09:47 appbuddy -> /etc/nginx/sites-available/appbuddy
root@lix:/etc/nginx/sites-enabled# 
```

Después de realizar el link simbólico, hemos de ejecutar el comando ***"ls -l"***, así **veremos los permisos** y demás información.

Hemos configurado la página estática, ahora **debemos de ver si la sintaxis está bien** y si todo está correctamente, para ello, ejecutaremos uno de los comandos que vimos anteriormente, el ***"nginx -t"***.

```
root@lix:/etc/nginx/sites-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@lix:/etc/nginx/sites-enabled# 
```

Este comando es muy útil, pues **nos dirá lo que está mal en el fichero que creamos previamente para poder ir y modificarlo.**

Como vemos, **todo está correcto**, así que **procedemos a reiniciar el servicio** con el comando: ***"systemctl restart nginx.service"***.

```
root@lix:/etc/nginx/sites-enabled# systemctl restart nginx.service
root@lix:/etc/nginx/sites-enabled# systemctl status nginx
* 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 Sat 2024-02-17 09:52:24 UTC; 4s ago
       Docs: man:nginx(8)
    Process: 11981 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 11982 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 11983 (nginx)
      Tasks: 2 (limit: 19068)
     Memory: 2.7M
        CPU: 42ms
     CGroup: /system.slice/nginx.service
             |-11983 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             `-11984 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">

Feb 17 09:52:24 lix systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 17 09:52:24 lix systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-16/16 (END)
^C
root@lix:/etc/nginx/sites-enabled# 
```

Y como manía mía (aunque no es necesario), **volvemos a ver el estado de nginx** para comprobar que todo sigue en orden, para ello, ejecutamos el comando: ***"systemctl status nginx"***.

4. Crear contenido de /var/www/html/buddy

Anteriormente en el fichero de configuración de la web estática, especificamos el directorio root:

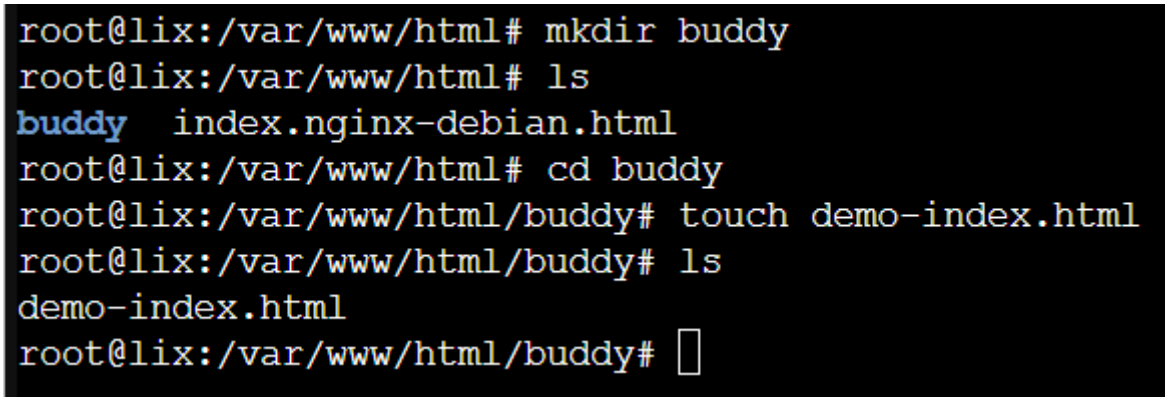
```
root@lix:/etc/nginx/sites-enabled# cat /etc/nginx/sites-enabled/appbuddy
server {
    listen 80 default_server;
    server_name appbuddy.com;

    root /var/www/html/buddy;
    error_log /var/log/nginx/app-server-error.log notice;
    index demo-index.html index.html;
    expires -1;


    sub_filter_once off;
    sub_filter 'server_hostname' '$hostname';
    sub_filter 'server_address'  '$server_addr:$server_port';
    sub_filter 'server_url'      '$request_uri';
    sub_filter 'remote_addr'     '$remote_addr:$remote_port';
    sub_filter 'server_date'     '$time_local';
    sub_filter 'client_browser'  'http_user_agent';
    sub_filter 'request_id'      '$request_id';
    sub_filter 'nginx_version'   '$nginx_version';
    sub_filter 'document_root'   '$document_root';
    sub_filter 'proxied_for_ip'  '$http_x_forwarded_for';

    location / {
      try_files $uri $uri/ /index.html /demo-index.html;
    }
}
root@lix:/etc/nginx/sites-enabled#
```

Dicho directorio no está creado, por lo que vamos a ir a crearlo.

Cambiamos de directorio a: ***"/var/www/html"***.

```
root@lix:/etc/nginx/sites-enabled# cd /var/www/html
root@lix:/var/www/html# ls
index.nginx-debian.html
root@lix:/var/www/html#
```

Si listamos el contenido del directorio, **podemos ver que hay un fichero**, pero **no lo vamos a utilizar**, vamos a **crear una carpeta en ese directorio con el nombre que especificamos en el fichero *"appbuddy"*** (osea, "buddy") y **dentro de dicha carpeta**, **creamos el fichero *"demo-index.html"***.



Después de haber creado esta estructura, **abrimos el fichero *"demo-index.html"*** con nuestro editor de texto favorito (en mi caso, ***"nano"***).

Dentro del fichero, creamos nuestra página html, en mi caso, voy a copiar la página de inicio de nginx.

Código:


```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World - App X <!-- Replace 'X' with '1' or '2' as appropriate --></title>
    <link
href="
HeAAAGPElEQVR42u1bDUyUdRj/iwpolMlcbZqtXFnNsuSCez/OIMg1V7SFONuaU8P1MWy1l
cPUyhK1uVbKcXfvy6GikTGKCmpEyoejJipouUBcgsinhwUKKKJ8PD3vnzsxuLv35Q644+Ue9
mwH3P3f5/d7n6/3/3+OEJ/4xCc+8YQYtQuJwB0kIp+JrzUTB7iJuweBf4baTlJ5oCqw11C/JH
p+tnqBb1ngT4z8WgReTUGbWCBGq0qvKRFcHf4eT/ZFBKoLvMBGIbhiYkaQIjcAfLAK+D8z
9YhjxMgsVUGc84+gyx9AYD0khXcMfLCmUBL68HMZ+PnHxyFw3Uwi8B8hgJYh7j4c7c8PV5
CEbUTUzBoHcU78iIl/FYFXWmPaNeC3q4mz5YcqJPI1JGKql2Z3hkcjD5EUznmcu6qiNT+Y2
CPEoH3Wm4A/QERWQFe9QQ0caeCDlSZJrht1HxG0D3sOuCEiCA1aj4ZY3Ipzl8LiVtn8hxi5z
RgWM8YYPBODF/9zxOLcVRVs+YGtwFzxCs1Bo9y+avBiOTQeUzwI3F5+kOwxsXkkmWNHH
rjUokqtqtSyysW5gUHV4mtmZEHSdRkl+aELvcFIRN397gPPXD4ZgbxJW1S5OJdA60MgUA
yHu1KfAz+pfCUtwr+HuQc8ORQ1jK4ZgGsTvcY5uQP5oYkY2HfcK5sGLpS6l1xZQwNn7Xkedp
3OgMrWC1DX0Qwnms/A1rK9cF9atNVo18DP/3o5fF99BGo7LFDRWgMJJQaYQv/PyOcHyS
P0TITrBIhYb+WSHLrlNGEx5NeXgj2paW8C5rs46h3Dc3kt3G2Ogr9aqoes+f5RvbL1aJ5iXnK
nxkfIEoB3N/zHeHAmF9ovwryvYvC9TysnICkEonPX212vvOU8+As6eS+QCDAw0aNLABq6L
O8DkJMSSznMMEfScFFGwCJYXbDV7lq17RYIQu+QTYpjRUBM3gZQIt+cOwyTpWRpYBQRs
KrgU4ceNS4JkCSxLI1+ZsIS0NvXB6sLE/tL5EQkQJKOm52YON9y7glqJkCSOqzrD6Uvc1wZ
1EBA07V/IafmN4ckHG+ugJkSEHuVQQ0ENFy9BLP3R0NR4ymHJGRWFWBnZ6fPVwMBF9
EDgrD2z0USqtoaHJKw49SBoZ2dWggIxmcEsvspYLLi4PKNDrvv68OfuKLt/68MqiJAan4Q0
```

IpDm6G7r8fue692X4fI7PiByqA6AqygNh0XHIaClDOkpz9aGVRJABo8CTP+3sqfHZJQeqkSg
vHZn+xaqEICKAlhECSGO60MWdVF4IcesDL/ExUSYN3okCrD31fqHZLwcWkq5owPVUoA3
UcIgdBv10BrV7vdz3b39kBhw0kVE2BNirG/bqRghyPqIcBKQkKJcVgE1LQ1wR3S5ooqCDBK
lSEUzGdyFBNwvq1RTQT0b4BOF5+BgoayCUqAtTLMSXsRzl6uHX8EONoUtXS2KCfAusOsyV
wFLV1tznNAuzflAGxb+R/esGuodDcD0bUVbYLelhRf/mWD08ogdYtTjNwYbIsrORhBIwJMP
OTWHh1i6Lriz107FUKviivcZvfp8WZvN8TmbVS2rtsHI8mMtn9gSe50KAz79yWw8490OGYp
p8lsTUGictd3EA6PHVwB20+mYUNURo/aMs4dhqjsdcoOWGxH5yYu0g0P0EzFBd7DxZoVH
Y7aHmWtB6VunwhLB6P0gFULk6zhJnvnBw5HW9D9N5GkpQEjMBcQOg+JMBNxjMZgHISa
wvGZHiKw+0mybv5ozP0txgvk07AQvWxAoh98sXsur3RmwMStxIud9fiIzMAIXTV6yNqxHa
H7gg1GA7bgxVvHfEjq1hAl10ZM/A46gO0x0bOPoiHpSEDvsMZhXVVbVRL4TLz2E140EK1dg
snnd9mBaHcmwuigJHeCGLkXvHNaNHOBP4J/HYmoGbGwsJU1ka0nAvM2ht40758ZNmv
vRRJ24l3roMa7MxVq4jpRdyMRc8bh9wR0TyIRWdR9hzNXaJs3Ftif6KDWuBcBH0hErky2bN
raV5E9jcBjiapE1ExHkO8iEY1OvjLTjAkugezh7ySqFUPoXHTtZAR7ncY4rRrYYgtcCtGHPUgmj
EhPmiKXjXc/l4g6HfGJT3ziEw/If86JzB/YMku9AAAAAElFTkSuQmCC" rel="icon"
type="image/png" />

```
    <style>
        body {
            margin: 0px;
            font: 20px 'RobotoRegular', Arial, sans-serif;
            font-weight: 100;
            height: 100%;
            color: #0f1419;
        }
        div.info {
            display: table;
            background: #e8eaec;
            padding: 20px 20px 20px 20px;
            border: 1px dashed black;
            border-radius: 10px;
            margin: 0px auto auto auto;
        }
        div.info p {
            display: table-row;
            margin: 5px auto auto auto;
        }
        div.info p span {
            display: table-cell;
            padding: 10px;
        }
```

```
    img {
        width: 176px;
        margin: 36px auto 36px auto;
        display:block;
    }
    div.smaller p span {
        color: #3D5266;
    }
    h1, h2 {
        font-weight: 100;
    }
    div.check {
        padding: 0px 0px 0px 0px;
        display: table;
        margin: 36px auto auto auto;
        font: 12px 'RobotoRegular', Arial, sans-serif;
    }
    #footer {
        position: fixed;
        bottom: 36px;
        width: 100%;
    }
    #center {
        width: 400px;
        margin: 0 auto;
        font: 12px Courier;
    }
</style>

<script>
    var ref;
    function checkRefresh() {
        if (document.cookie == "refresh=1") {
            document.getElementById("check").checked = true;
            ref = setTimeout(function(){location.reload();}, 1000);
        } else {
        }
    }
    function changeCookie() {
        if (document.getElementById("check").checked) {
            document.cookie = "refresh=1";
```

```
            ref = setTimeout(function(){location.reload();}, 1000);
          } else {
            document.cookie = "refresh=0";
            clearTimeout(ref);
          }
        }
      </script>
    </head>

    <body onload="checkRefresh();">
      <img alt="NGINX Logo"
src="
+RAAAAVFBMVEUAAAAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAA
mQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAAmQAA
mQAAmQAAmQDBect+AAAAG3RSTlMAB0AY8SD5SM82v1npsJ/YjSl0EVLftqllgMdZgsoQA
AAHd0lEQVR42szZ6XabMBCG4ZGFxSazLzZz//fZc9I4JpbEN8LQ0/dnGwJ5DJGG0HdpM9
kkuzVXiqussmRpLrRdnwqDp9ePyY7zXdFbqptHOz00RTVUxWiyquvJ26Upknp2/heWN0Uy
zt3qYtKMn805ybsW/LdK01YVC6sVELH81XJ9o6j5q6Qkcepe83dJp8ipf161HSgm1TyPK5//c
uN1d5KmE342bsnkLK6hre78LNG0KuWfOrFDwats69w8ln+qFIlrx9Vxf8808e8eJGx9YEXhC
pZ3kX2gfFtbrX4m05IonTE7wsGLnpXY1/Kqr3v/5r+NcAOvy8HXCRt74W+alH568KqCJKmM
37LafVhe3ZTU1/mmA7uV9Ar8vPjZVCPDZI+CDdwFC68yIooZnbhmIAx8XyoZu5mcYO9Hzh
So47gGCqR53ULPlAGPkuyazJVeKWYsjH15Djy/VhPO8LoM/OJE4XNfeJ19LUfRj18KF9gLA
2GZL4/UsLdFHQVccWyTCDjZD9wm7Kt2PgIgjH3ZBlf46iDgnOO7nwusavZmVoCaPU0q1pc
nshyoOwa44PiS66nANw7U0isbK5x7j3gQB0uPAB54T8WZwA/RHrxhLIx9TbsBnLSfA6uRd9
WdBzywCFiNUcJ5wr4eRByu7j8G7nhfpj0LuE0A8OtsSBj7ZooIL+dyYLxFm27+EvfSzgHua/G
YXrK3Qol9a03bwNxEAeMt2ix/bptzgCeGwFhY7ouAufwIOA/PSni3nJ8B3DAElgtjXwxs8k+Al
/BdiVfDWh0PPDAAjhXGvgTnVjkwujzbk1t4TWkOB24TBBwrjH2JQZnaC6xGsPdCT296MHA/
MgKWC2NfL7Blp2ov8AM88/gNbX8osCrc5xMAA2Ho6wIXHTt1+4C1iZwMW8NvzYcCN67vA
ICBMPZ1galip3QXcAXHXzyVlB8AYyiT5wAYCWNfF1gtYGYWAufhNynyTWqiDwPOjeelnQiYS
hMQBr5+YNIWzMwy4CX69afv1NNRwHr07FKEwDT4hTPs6wL7P+tCxQKXm/eifJ963wmMF
7hCYWBXGJdpAsBUopkZAyv3j3+i9PUtTa/U9VcAGC1wmgAwFsa+LnBooLxj4K0t2qjo8AAw
WuAIAO8TznoSANMEZmYErA14p3EyMF7gSgLAQBj4ImBVg5kZAM/8u4VAJwJ7l+2GADAQ
Br4A2D+1Z0oMnKM3Y2cD4wUOAANh5IuB6cJOsxg4Q0eeCwwXuFETBnZLDfSVA1NwZsbA
JXwN/C+B7771BAAjYeyLgX0z8yACVlawx1NaXh+5TcMLHACGwtgXA6OZ2QUObdGsorfabj
Isr4wcNOACB4CBMPLFwOHpcuwx8NWgLXTJURW0H1gtngUOA8cLLz1FAsOZWQ4MfFH5
B8CV7x75b4D/NHduS47CMBCVwYFAiDEmCQT+/z/3ZWumah1otZdL/MxMZc5gybJanU8t
LI9DhF8PESXJ10k64PAxyn1LiPisMhr/N8kNHF+bpwPOis95+juS3IJOrsgQYBlXj2mWFVHRg
HGC+4pj2kKjbG4ufKGRLmdtTTJgc12WKn1BofE7zBTXzAhwtlIqP9h5gmTAbq1xcHqpvBbH
BgRY7suXPTl/ROMB4wR36mUPKjXnNwLcrVxXXimRZTLgDBSiZ15XYj3XAwAWv3zh7gnAXt
IAx6Etnq888cIdX/fZDgDul1tGvf4Vtn0S4M8J7i7ROq1lhCVHzzwGvBpYbJ5AOEgq4EEzZn5
K01MrmqvNOmDTLrft+8FSRzQecFBpO05p26tlnw7oIso14YnJ3i5aL6DF0wMuleqkM4Qn+s
```

```
mcAKRTL1Y65UDQVAO+WK2+7gTplH54usjWAXek+K+LCuxEwGMLul0R4EPFfz8L18zzKmD
xIKSCN95LIuBGr3GujpevErqxGQDuLaPuyUAfBAPGg6Mx4OME2DhQVgUJWAIzQnBFfRAe
MI5N1XEjBBiwjCxg0+qHYG7wt/GA8capDh+CqYkpCoykjPKWesio2gywEwD4qDEuDNjUJG
CptQqUAB5MB3w1APBhg4gYsPQtCbib00Zpi3wrwM1FAOBjR2lrZBXCARY3J623bAS4yAQA
PnIYHAOWkgSc2xS+T7MV4CAA8LF2BhiwBAwYP4+lPBsBdgIAH2XIgQHjTf+SrRw5auEAG5
Dg9ID3t5TBgM3EWR88eMAVCVieYM5aDXgHUyQAmKiZR9nIFckJC/gFnALUgHew9QKAiZ
q5A3+EXspDAw7gP64GvIcxXQvfHl2B7tiozSf+y1JSNQ31gRYDQb6HteKQ4B3s4QucflRrDW
8OKiHBujCO3s0u5qAjwKR0vnkDozL1emgd5W6EWa1ud7l97G0n3jhYzACOEMlHtVpjeBA/m
Lf/7IOoQsa7y+b7GDR3Rbw98fKQLy+5xv7VIXowIhy1ztUfbdzLYrz7cbrvRb/K+nf7wPPQpA
XsEQ/7l2AXW97/AGkCwaNsIif8zU3y5eZaO/mK/jKDV1s872/Fz11K5TLE1zzEiP1km8ndDMcj
3JvmFfqdvubhD8TgHPiN+LViAAAAAElFTkSuQmCC"/>
      <div class="info">
        <p><span>Server name:</span> <span>server_hostname</span></p>
        <p><span>Server address:</span> <span>server_address</span></p>
        <p class="smaller"><span>User Agent:</span> <span>client_browser</span></p>
        <p class="smaller"><span>URI:</span> <span>server_url</span></p>
        <p class="smaller"><span>Doc Root:</span> <span>document_root</span></p>
        <p class="smaller"><span>Date:</span> <span>server_date</span></p>
        <p class="smaller"><span>NGINX Frontend Load Balancer IP:</span>
<span>remote_addr</span></p>
        <p class="smaller"><span>Client IP:</span> <span>proxied_for_ip</span></p>
        <p class="smaller"><span>NGINX Version:</span>
<span>nginx_version</span></p>
      </div>
      <div class="check">
        <input type="checkbox" id="check" onchange="changeCookie()"> Auto
Refresh</input>
      </div>
      <div id="footer">
        <div id="center" align="center">
          Request ID: request_id<br/>
          © NGINX, Inc. 2018
        </div>
      </div>
  </body>
</html>
```

Puedes o no tomar este código, ahí ya decisión de cada uno.

Una vez hayamos escrito el código html y css de nuestra página, guardamos y salimos.

Vamos a realizar una pequeña **comprobación**, para ello **instalamos la herramienta *"curl"*** con el comando ***"sudo apt install curl"***.

```
root@lix:/var/www/html/buddy# apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4 libldap-2.5-0 libldap-common librtmp1 libssh-4
The following NEW packages will be installed:
  curl libcurl4 libldap-2.5-0 libldap-common librtmp1 libssh-4
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 926 kB of archives.
After this operation, 2592 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Escribimos "y".

Debemos realizar un ***"curl localhost"*** para comprobar que encuentra la página:

```
root@lix:/var/www/html/buddy# curl localhost
<!DOCTYPE html>
<html>
    <head>
        <title>Hello World - App X <!-- Replace 'X' with '1' or '2' as appropriate --></title>
        <link href="
ZqtXFnNsuSCez/OIMg1V7SFONuaU8P1MWy1lcPUyhK1uVbKcXfvy6GikTGKCmpEyoejJipouUBcgsinhwUKKKJ8PD3vnzsxuLv35Q644+Ue9mwH3P3f5/d7
n6/3/3+OEJ/4xCc+8YQYtQuJwB0kIp+JrzUTB7iJuweBf4baTlJ5oCqw11C/JHp+tnqBb1ngT4z8WgReTUGbWCBGq0qvKRFcHf4eT/ZFBKoLvMBGIbhiYka
QIjcAfLAK+D8z9YhjxMgsVUGc84+gyx9AYD0khXcMfLCmUBL68HMZ+PnHxyFw3Uwi8B8hgJYh7j4c7c8PV5CEbUTUzBoHcU78iIl/FYFXWmPaNeC3q4mz5Y
cqJPI1JGKql2z3hkcjD5EUznmcu6qiNT+Y2CPEoH3Wm4A/QERWQFe9QQ0caeCDlSZJrht1HxG0D3sOuCEiCA1aj4ZY3Ipzl8LiVtn8hxi5zRgWM8YYPBODF
/9zxOLcVRVs+YGtwFzxCs1Bo9y+avBiOTQeUzwI3F5+kOwxsXkkmWNHHrjUokqtqtSyysW5gUHV4mtmZEHSdRkl+aELvcFIRN397gPPXD4ZgbxJW1S5OJdA
60MgUAyHu1KfAz+pfCUtwr+HuQc8ORQ1jK4ZgGsTvcY5uQP5oYkY2HfcK5sGLpS611xZQwNn7Xkedp3OgMrWC1DX0Qwnms/A1rK9cF9atNVo18DP/3o5fF9
9BGo7LFDRWgMJJQaYQv/PyOcHySP0TITrBIhYb+WSHLrlNGEx5NeXgj2paW8C5rs46h3Dc3kt3G2Ogr9aqoes+f5RvbL1aJ5iXnKnxkfIEoB3N/zHeHAmF9
ovwryvYvC9TysnICkEonPX212vvOU8+As6eS+QCDAw0aNLABq6LO8DkJMSSznMMEfScFFGwCJYXbDV7lq17RYIQu+QTYpjRUBM3gZQIt+cOwyTpWRpYBQRs
KrgU4ceNS4JkCSxLI1+ZsIS0NvXB6sLE/tL5EQkQJKOm52YON9y7glqJkCSOqzrD6Uvc1wZ1EBA07V/IafmN4ckHG+ugJkSEHuVQQ0ENFy9BLP3R0NR4ymH
JGRWFWBnZ6fPVwMBF9EDgrD2z0USqtoaHJKw49SBoZ2dWggIxmcEsvspYLLi4PKNDrvv68OfuKLt/68MqiJAan4Q0IpDm6G7r8fue692X4fI7PiByqA6Aqy
gNh0XHIaClDOkpz9aGVRJABo8CTP+3sqfHZJQeqkSgvHZn+xaqEICKAlhECSGO60MWdVF4IcesDL/ExUSYN3okCrD31fqHZLwcWkq5owPVUoA3UcIgdBv10
BrV7vdz3b39kBhw0kVE2BNirG/bqRghyPqIcBKQkKJcVgE1LQ1wR3S5ooqCDBKlSEUzGdyFBNwvq1RTQT0b4BOF5+BgoayCUqAtTLMSXsRzl6uHX8EONoUt
XS2KCfAusOsyVwFLV1tznNAuzflAGxb+R/esGuodDcD0bUVbYLelhRf/mWD08ogdYtTjNwYbIsrORhBIwJMPOTWHh1i6Lriz107FUKviivcZvfp8WZvN8Tm
bVS2rtsHI8mMtn9gSe50KAz79yWw8490OGYpp8lsTUGictd3EA6PHVwB20+mYUNURo/aMs4dhqjsdcoOWGxH5yYu0g0P0EzFBd7DxZoVHY7aHmWtB6Vunwh
LB6P0gFULk6zhJnvnBw5HW9D9N5GkpQEjMBcQOg+JMBNxjMZgHISawvGZHiKw+0mybv5ozP0txgvk07AQvWxAoh98sXsur3RmwMStxIud9fiIzMAIXTV6yN
qxHaH7gg1GA7bgxVvHfEjq1hAl10ZM/A46gO0x0bOPoiHpSEDvsMZhXVVbVRL4TLz2E140EK1dgsnnd9mBaHcmwuigJHeCGLkXvHNaNHOBP4J/HYmoGbGws
JU1ka0nAvM2ht40758ZNmvvRRJ24l3roMa7MxVq4jpRdyMRc8bh9wR0TyIRWdR9hzNXaJs3Ftif6KDWuBcBH0hErky2bNraV5E9jcBjiapE1ExHkO8iEY1O
vjLTjAkugezh7ySqFUPoXHTtZAR7ncY4rRrYYgtcCtGHPUgmjEhPmiKXjXc/l4g6HfGJT3ziEw/If86JzB/YMku9AAAAAElFTkSuQmCC" rel="icon" ty
pe="image/png" />
        <style>
            body {
                margin: 0px;
                font: 20px 'RobotoRegular', Arial, sans-serif;
                font-weight: 100;
                height: 100%;
```

Todo correcto.

Ahora **pasamos al lxc *"revproxy"***.

5. Realizar modificaciones en *"revproxy"*

Cabe comentar que ***"revproxy"*** **es una clonación de** ***"lix"***, así **nos ahorramos tener que instalar Nginx de nuevo**, tener que modificar todo lo que hemos modificado, etc.

Ahora, al ser una clonación, **hay cosas que debemos modificar** en ***"revproxy"***, por lo que vamos a empezar a cambiar cosas.

Lo primero a modificar es **la IP**, ya que **no puede ser la misma que la de** ***"lix"***, en este caso, le pondré la siguiente, osea se, la **172.16.1.41**.

Seguimos con el directorio ***"/var/www/html"***, **si** nos **vamos a ese directorio** y **listamos el contenido**, **podemos ver que está la carpeta con el fichero** ***"demo-index.html"*** que creamos en ***"lix"***.

No necesitamos dicha carpeta, por lo que vamos a ejecutar el comando: ***"rm -r buddy/"*** **para borrar la carpeta y el contenido**.

```
root@revproxy:/var/www/html# rm -r buddy/
root@revproxy:/var/www/html# ls
index.nginx-debian.html
root@revproxy:/var/www/html#
```

Cambiamos de directorio otra vez y nos vamos **a** ***"/etc/nginx/sites-enabled"***, listamos el contenido con ***"ls"*** y vemos que está la página estática.

```
root@revproxy:/var/www/html# cd /etc/nginx/sites-enabled
root@revproxy:/etc/nginx/sites-enabled# ls
appbuddy
root@revproxy:/etc/nginx/sites-enabled#
```

Eliminamos ***"appbuddy"*** con el comando: ***"rm appbuddy"***.

```
root@revproxy:/etc/nginx/sites-enabled# ls
appbuddy
root@revproxy:/etc/nginx/sites-enabled# rm appbuddy
root@revproxy:/etc/nginx/sites-enabled# ls
root@revproxy:/etc/nginx/sites-enabled#
```

6. Realizar backup

Cambiamos de directorio y nos vamos a ir a ***"/etc/nginx/sites-available"***.

```
root@revproxy:/etc/nginx/sites-enabled# cd ..
root@revproxy:/etc/nginx# cd sites-available
root@revproxy:/etc/nginx/sites-available# ls
appbuddy  default
root@revproxy:/etc/nginx/sites-available#
```

¿Qué vamos a hacer aquí?, una copia de seguridad, para realizar el backup de ***"default"***, ejecutamos el comando: ***"cp ../sites-available/default ../sites-available/default.bak"***.

```
root@revproxy:/etc/nginx/sites-available# cp ../sites-available/default ../sites-available/default.bak
root@revproxy:/etc/nginx/sites-available# ls
appbuddy  default  default.bak
root@revproxy:/etc/nginx/sites-available#
```

7. Configurar proxy inverso

Vamos a entrar al fichero ***"default"*** con el comando ***"nano default"***.

```
  GNU nano 6.2                                          default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
```

Lo que vamos a hacer es borrar todos los comentarios, así será más fácil realizar la configuración.

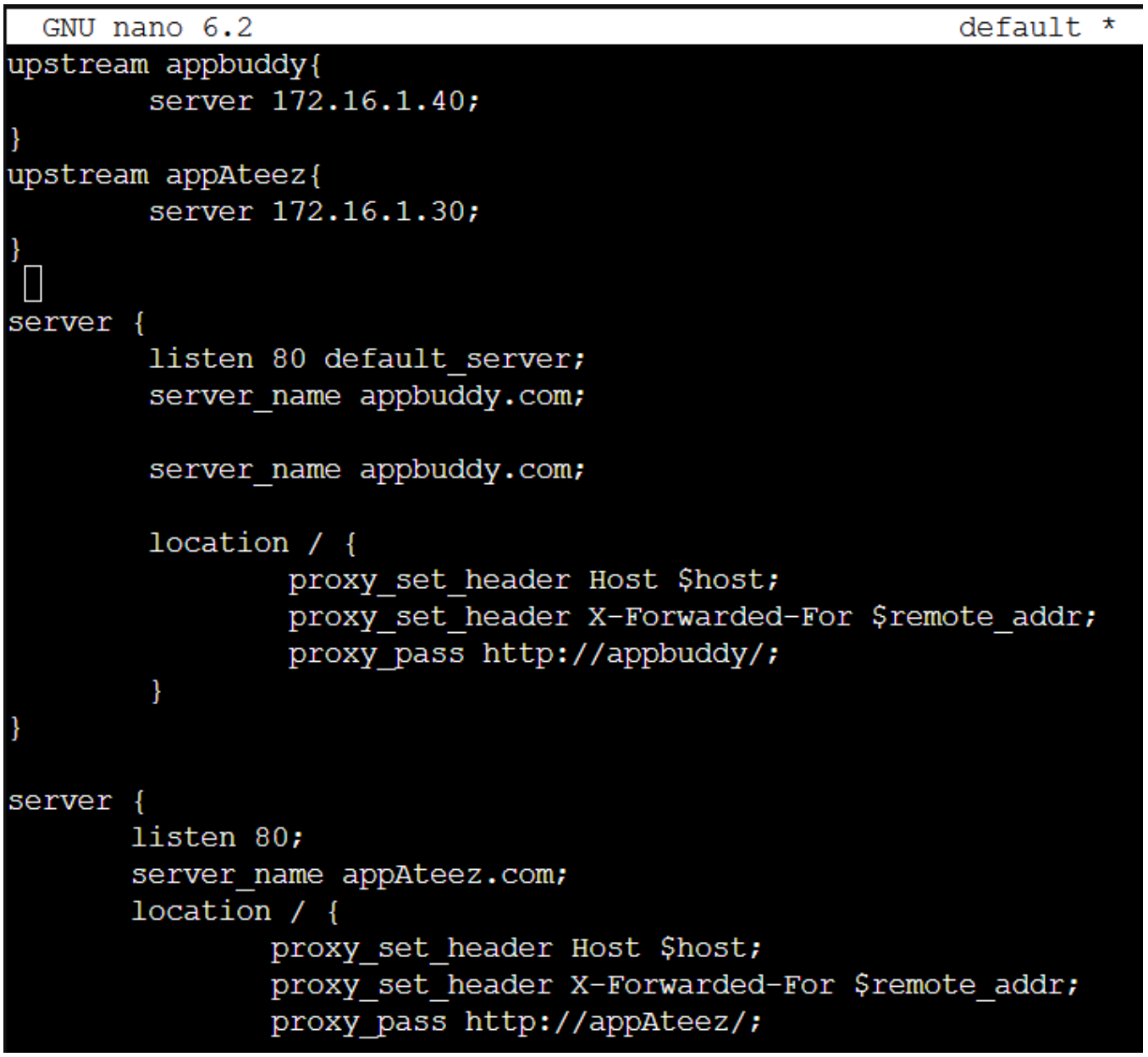
Una vez borrado todos los comentarios, veremos esto:

```
  GNU nano 6.2                                                  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;
        }
}
```

Vamos a añadir algunas líneas arriba de la configuración que vemos, pero aquí entra nuestro tercer lxc ***"ateez"***.

En el lxc llamado ***"ateez"*** tenemos otra página estática diferente.

La configuración del proxy inverso quedaría tal que así:



Guardamos y salimos.

A continuación, vamos a crear un enlace simbólico de ***"sites-available"*** a ***"sites-enabled"***. El comando a ejecutar es: ***"ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/"***.

Después de crear el enlace simbólico, ejecutamos ***"ls -l"*** para ver si se ha creado correctamente.

```
root@revproxy:/etc/nginx/sites-available# ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/
root@revproxy:/etc/nginx/sites-available# ls -l
total 12
-rw-r--r-- 1 root root  833 Feb 17 09:50 appbuddy
-rw-r--r-- 1 root root  554 Feb 17 13:59 default
-rw-r--r-- 1 root root 2412 Feb 17 10:48 default.bak
root@revproxy:/etc/nginx/sites-available#
```

8. Comprobaciones

Y ahora ejecutamos el comando para verificar que todo esté correctamente, osea se: ***"nginx -t"***.

```
root@revproxy:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@revproxy:/etc/nginx/sites-available#
```

Todo ha salido correctamente.

Reiniciamos el servicio y vemos el estado (***"systemctl restart nginx"*** y ***"systemctl status nginx"***).

```
root@revproxy:/etc/nginx/sites-available# systemctl restart nginx
root@revproxy:/etc/nginx/sites-available# systemctl status nginx
* 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 Sat 2024-02-17 14:06:23 UTC; 4s ago
       Docs: man:nginx(8)
    Process: 354 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SU>
    Process: 355 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 356 (nginx)
      Tasks: 2 (limit: 19068)
     Memory: 2.7M
        CPU: 44ms
     CGroup: /system.slice/nginx.service
             |-356 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             `-357 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">

Feb 17 14:06:23 revproxy systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 17 14:06:23 revproxy systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-16/16 (END)
^C
root@revproxy:/etc/nginx/sites-available#
```

9. Acceder a las páginas desde VM

Al estar realizando la práctica en lxc, no tenemos GUI, por lo que vamos a irnos a un equipo con interfaz gráfica para poder ver las páginas desde el navegador.

Aquí es donde entra nuestro Ubuntu cliente llamado ***"zerowave"***, con este equipo, podremos ver si todo funciona correctamente.

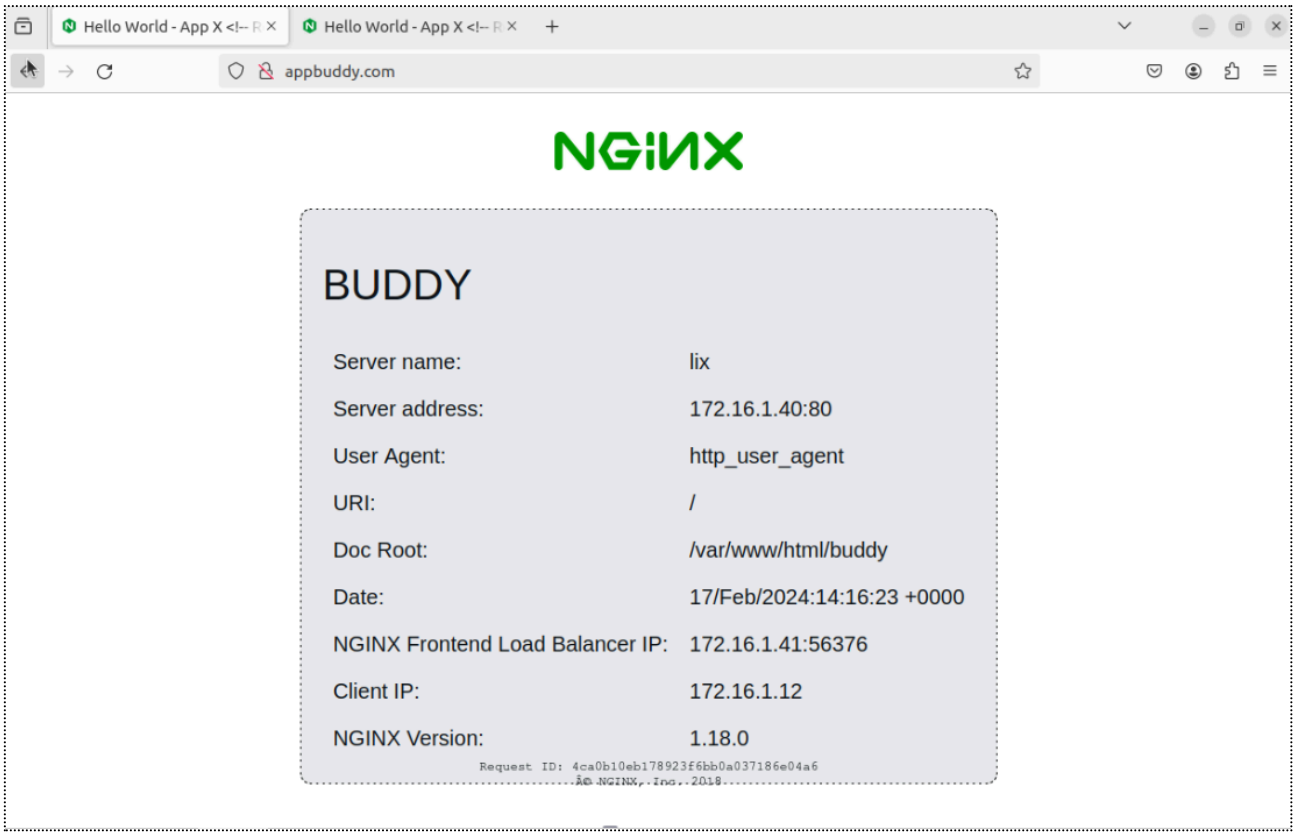
Nos dirigimos a la terminal y escribimos: ***"sudo /etc/hosts"***.

```
  GNU nano 6.2                                                /etc/hosts
127.0.0.1       localhost
127.0.1.1       zerowave
172.16.1.41     appbuddy.com appAteez.com

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

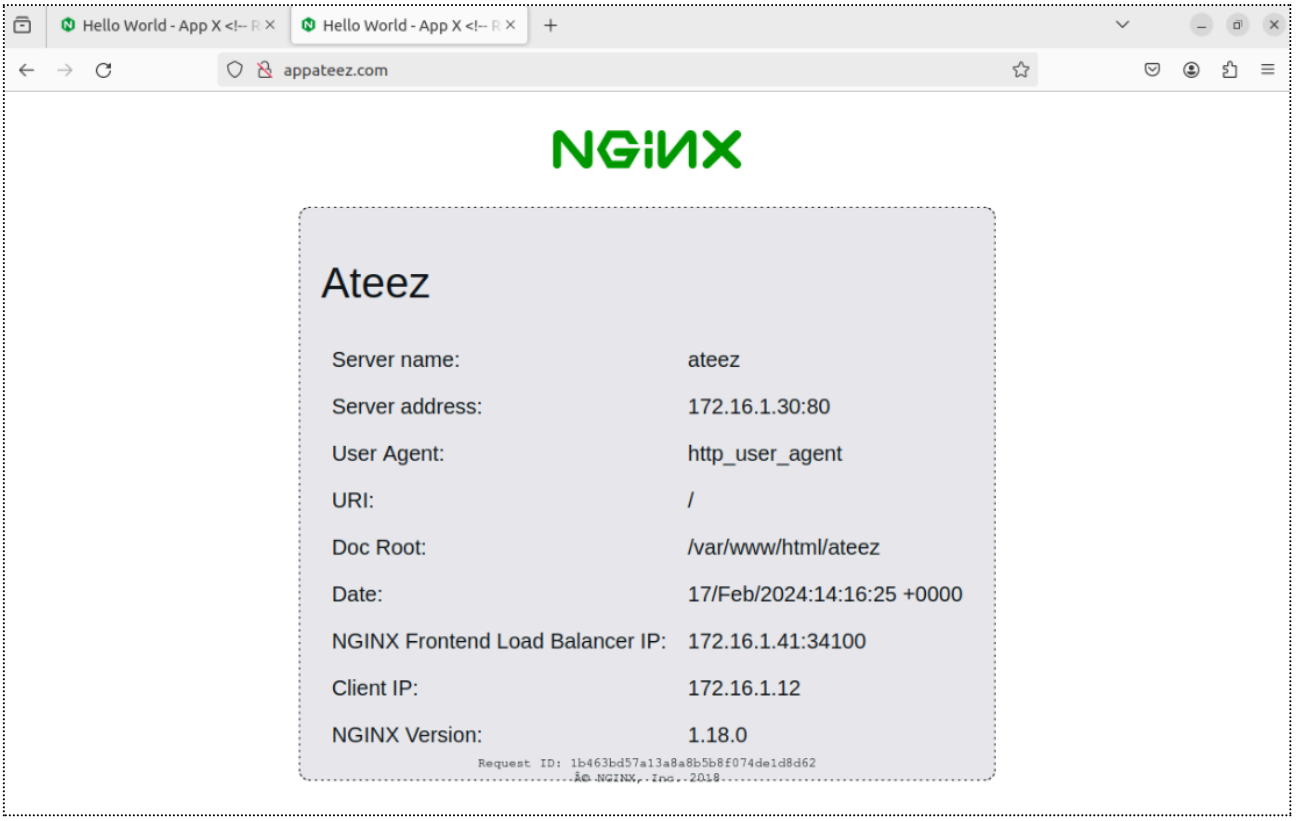
Debemos de poner la IP del proxy inverso y al lado las 2 páginas estáticas que hemos configurado.

Una vez realizados estos cambios, guardamos, vamos a nuestro navegador (en mi caso Mozilla Firefox) y buscamos ***"appbuddy.com"*** y ***"appAteez.com"***.

Vemos que ***"appbuddy.com"*** si contesta.



Y ***"appAteez.com"*** también responde.



Con esto damos por finalizada la práctica.

10. Bibliografía

https://www.maquinasvirtuales.eu/nginx-crear-proxy-inverso-en-centos-7/

https://kinsta.com/es/blog/proxy-inverso/

https://www.youtube.com/watch?v=8wIJNANf4ME

https://serverspace.io/es/support/help/how-to-configure-reverse-proxy-on-ubuntu-server-22-04-using-nginx/

https://learn.microsoft.com/es-es/troubleshoot/developer/webapps/aspnetcore/practice-troubleshoot-linux/2-2-install-nginx-configure-it-reverse-proxy