

Proxy Inverso

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 libjpeg8
  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-subst-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 libjpeg8
  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-subst-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.

[illegible]

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"***.

```
GNU nano 6.2 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@elix:/etc/nginx/sites-enabled# ln -s /etc/nginx/sites-available/appbuddy /etc/nginx/sites-enabled/
root@elix:/etc/nginx/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 35 Feb 17 09:47 appbuddy -> /etc/nginx/sites-available/appbuddy
root@elix:/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”***.

[illegible]

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;
```

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”*.

```
root@lix:/var/www/html# mkdir buddy
root@lix:/var/www/html# ls
buddy index.nginx-debian.html
root@lix:/var/www/html# cd buddy
root@lix:/var/www/html/buddy# touch demo-index.html
root@lix:/var/www/html/buddy# ls
demo-index.html
root@lix:/var/www/html/buddy#
```

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


```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World - App X <!-- Replace 'X' with '1' or '2' as appropriate --></title>
    <link
href="data:image/png;base64,iVBORw0KGgoAAAANSUUEgAAAEAAAABACAYAAACQaXHe
AAAGPEIEQVR42u1bDUyUdRj/iwpolMlcbZqtXFfnNsuSCez/OIMg1V7SFONuaU8P1MWy1lcPUy
hK1uVbKcXfvy6GikTGKCmpEyoeJjipouUBcgsinhWUKKKJ8PD3vnzsxuLv35Q644+Ue9mwH3P
3f5/d7n6/3/3+OEJ/4xCc+8YQYtQuJwB0kIp+JrzUTB7iJuweBf4baTIJ5oCqw11C/JHp+tnqBb1
ngT4z8WgReTUGbWCBGq0qvKRfChf4eT/ZFBKoLvMBGIbhiYkaQIjcAfLAK+D8z9YhjxMgsVU
Gc84+gyx9AYD0khXcmFLCmUBL68HMZ+PnHxyFw3Uwi8B8hgJYh7j4c7c8PV5CEbUTUzBoHc
U78iIL/FYFXWmPaNeC3q4mz5YcqJPI1JGKql2Z3hkcd5EUznmcu6qiNT+Y2CPEoH3Wm4A/Q
ERWQFe9QQ0caeCDISZJrht1HxGOD3sOuCEICA1aj4ZY3Iplz8LiVtn8hxi5zRgWM8YYPBODF/9
zxOLcVRVs+YGtwFzxCs1Bo9y+avBiOTQeUzwI3F5+kOwxSxkkmWNHHrjUokqtqtSyysW5gUH
V4mtmZEHSdRkl+aELvcFIRN397gPPXD4ZgbxJW1S5OJdA60MgUAyHu1KfAz+pfCUtwr+HuQ
c8ORQ1jK4ZgGsTvcY5uQP5oYkY2HfcK5GLpS6l1xZQwNn7Xkedp3OgMrWC1DX0Qwnms/A1r
K9cF9atNV018DP/3o5ff99BG07LFDRWgmJJQaYQv/PyOcHySP0TITrBIhYb+WSHLrINGEx5N
eXgj2paW8C5rs46h3Dc3kt3G2Ogr9aqoes+f5RvbL1aJ5iXnKnxfIEoB3N/zHeHAMF9ovwryvY
vC9TySnICkeonPX212vvOU8+As6eS+QCDaw0aNLAbq6LO8DkJMSSzmMEfScFFGwCJYXBd
V7lq17RYIQU+QTYpjRUBM3gZQIt+cOwyTpWRpyBQRsKrgU4ceNS4JkCSxLI1+ZsIS0NvXB6s
LE/tL5EQkQJKOm52YON9y7glqJkCSOqrD6Uvc1wZ1EBA07V/IafmN4ckHG+ugJkSEHuVQQ
0ENfy9BLP3R0NR4ymHJGRFWBNz6fPVwMBF9EDgrD2z0USqtoaHJKw49SB0Z2dWggIxmn
cEsvspYLLi4PKNDrvv68OfuKLt/68MqiJAan4Q0IpDm6G7r8fue692X4fI7PiByqA6AqygNh0XH
IaCIDOkpz9aGVRJAB08CTP+3sqfHZJQeqkSgvHZn+xaqEICKAlhECSGO60MWdVF4IcesDL/E
xUSYN3okCrD31fqHZLwcWkq5owPVUoA3UcIgdBv10BrV7vdz3b39kBhw0kVE2BNirG/bqRghy
PqIcBKQkKJcVge1LQ1wr3S5ooqCDBKISeuzGdyFBNwvq1RTQTOb4BOF5+BgoayCUqAtTLMS
XsRzl6uHX8EONoUtXS2KCfAusOsYVwFLV1tznNAuzflAGxb+R/esGuodDcD0bUVbYLelhRf/m
WD08ogdyTtJNwYbIsrORhBIwJMPOTWHh1i6Lriz107FUKviivcZvfp8WZvN8TmbVS2rtsHI8m
Mtn9gSe50KAz79yWw8490OGYpp8lsTUGictd3EA6PHVWB20+mYUNURo/aMs4dhqjsdcoOWG
xH5yYu0g0P0EzfBd7DxZoVHY7aHmWtB6VunwhLB6P0gFULk6zhJnvNBw5HW9D9N5GkpQEj
MBcQOg+JMBNxjMZgHISawvGZHikw+0mybv5ozP0txgvk07AQvWxAoh98sXsur3RmwMStXI
ud9fiIzMAIXTV6yNqxHaH7gg1GA7bgxVvHfEjq1hAl10ZM/A46g00x0bOPoiHPSEDvsMZhXVV
bVRL4TLz2E140EK1dgsnnd9mBaHcmwuigJHeCLkXvHNANHOBP4J/HYmoGbGwsJU1ka0n
AvM2ht40758ZNmvvRRJ24l3roMa7MxVq4jpRdyMRc8bh9wR0TyIRWdR9hzNXaJs3Ftif6KDwU
BcBH0hErky2bNraV5E9jCbjiapE1ExHkO8iEY1OvjLTJakugezh7ySqFUPoXHTtZAR7ncY4rRrYYg
tcCtGHPUgmjEhPmiKXjXc/l4g6HfGJT3ziEw/If86JzB/YMku9AAAAAEIFTkSuQmCC" 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") {

```



```
nBFfRAeMI5N1XEjBBiwjCxcg0+qHYG7wt/GA8capDh+CqYkpCoykjPKWesio2gywEwD4qDEuD
NjUJGCptQqUAB5MB3w1APBhg4gYsPQtCbib00Zpi3wrwM1FAOBjR2lrZBXCARY3J623bAS4yA
QAPnIYHAOWkgSc2xS+T7MV4CAA8LF2BhiwBAwYP4+IPBsBdgIAH2XIgQHjTf+SrRw5auEAG
5Dg9ID3t5TBgM3EWR88eMAVCVieYM5aDXgHUyQAmKiZR9nIFckJC/gFnALUgHew9QKAiZq
5A3+EXspDAw7gP64GvIcxXQvfHI2B7tiozSf+y1JSNQ31gRYDQb6HteKQ4B3s4QucfIRrDW8OK
iHBujCO3s0u5qAjwKR0vnkDozL1emgd5W6EWa1ud7l97G0n3jhYzACOEMLHtVpjeBA/mLf/7IO
oQsa7y+b7GDR3Rbw98fKQLy+5xv7VIXowIhy1ztUfbdzLYrz7cbrvRb/K+nf7wPPQpAXsEQ/7l2
AXW97/AGkCwaNsIif8zU3y5eZaO/mK/jKDV1s872/Fz11K5TLE1zzEiP1km8ndDMcj3JvmFfqdv
ubhD8TgHPiN+LVIAAAAAEIFTkSuQmCC"/>
```

```
<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="data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAAAAABACAYAAACQaXHeAAAGPElEQVR42u1bDUyUdRj/iwpo1M1cb
ZqtXFnNsusCez/OIMg1V7SFONuaU8P1Mwy11cPUyhK1uVbKcXfvy6GikTGKcMPEyoejJipouUBcgsinhwUKKKJ8PD3vnzsxuLv35Q644+Ue9mwH3P3f5/d7
n6/3/+OEJ/4xCc+8YQYtQuJwB0kIp+JrzUTB7iJuweBf4baTlJ5oCgw11C/JHp+tnqBb1ngT4z8WgReTUGbWCBGg0qvKRFcHf4eT/ZFBKoLvMBGibhiYka
QIjcAfLAK+D8z9YhjxMgsVUGc84+gyx9AYD0khXcmFLcmUBL68HMZ+PnHxyFw3Uwi8B8hgJYh7j4c7c8PV5CEbUTUZBoHCU78i11/FYFXWmPaNecC3q4mz5Y
cqJPI1JGKq12Z3hkcjD5EUznmCu6qiNT+Y2CPEoH3Wm4A/QERWQFe9Q0QcaeCD1SZJrht1HxG0D3sOuCEiCa1aj4ZY3Ipz18LiVtn8hxi5zRgWM8YYPBODF
/9zxOLCVRVs+YgtWfZxCs1Bo9y+avBiOTQeUzWI3F5+kOwxsXkkmWNHHRjUokqtqtSyysW5gUHV4mtmZEHSdRkl+aELvcFIRN397gPPXD4ZgbxJW1S5OJdA
60MgUAYhulKfAz+pfCUtWr+HuQc8ORQ1jK4ZgGsTvcY5uQP5oYkY2HfcK5sGLpS611xZQwNn7Xkedp3OgMrWC1DX0Qwnms/AlrK9cF9atNVo18DP/3o5ff9
9BGo7LFDRWgMJJQaYQv/PyOchYSP0TITrBIhYb+WSHLr1NGEx5NeXgj2paW8C5rs46h3Dc3kt3G2Ogr9aqoes+f5RvbL1aJ5iXnKnxfIEoB3N/zHeHAmF9
ovwryvYvCtYsnICKeOnP2X12vvOU8+As6es+QCDaw0aNLAbq6LO8DkJMSSznMMEfScFFGwCJYxbDV7lq17RYIQu+QTYpjRUBM3gzQIt+cOwyTPWRpYBQRs
KrgU4ceNS4JkCSxLi1+ZsIS0NvXB6sLE/tL5EQKJQKJom52YON9y7glqJkCSOqzrD6UvclwZ1EBA07V/IafmN4ckHG+ugJkSEHUvQ00ENfy9BLP3RONR4ymH
JGRWFWBnZ6FPVwMBF9EDgrD2z0USqtoahJKw49SBoZ2dWggIxmCEsvspYLLi4PKNDrvv68OfuKlt/68MqiJAan4Q0IpDm6G7r8fue692X4fi7PiByqA6Aqy
qNhoXHiaC1Dokpz9aGVRJABO8CTP+3sqfHZJQeqKsgvHZN+xaqEICKAlhECsGO60MwdVF4IcesDL/ExUSYN3okCrD31fqH2LwcWkq5owPVUoA3UcIgdBv10
BrV7vdz3b39kBhw0kVE2BNirG/bgRghyPqICBKQKJcVgE1LQ1wR3S5ooqCDBK1SEUzGdyFBNwvq1RTQTOb4BOF5+BgoayCuqAtTLMSXsRz16uHX8EONoUt
XS2KCFausOsyVwFLVltznNAuzflAGxb+R/esGuodDcD0bUvYlElhRf/mWD08ogdyTtjNwYbIsrORhBIwJMPOTWHh1i6Lriz107FUKviiVcZvfp8WZvN8Tm
bVS2rtsHI8mMtn9gSe50KAz79yWw8490OGYpp81stUGictd3EA6PHVwB20+mYUNURo/aMs4dhqjsdcoOWGxH5yYu0g0P0EzFBD7DxZoVHY7aHmWtB6Vunwh
LB6P0gFULk6zhJnvnBw5HW9D9N5GkpQEBjMBCQOg+JMBNxmZGHISawvGZHiKw+Omybv5ozP0txgvk07AQvWxAoh98sXsur3RmwMStxIud9fiIzMAIXTV6yN
qxHaH7gg1GA7bgxVvHfEjqlhA110ZM/A46g00x0BoPoiHpsEDvsMZHXVbVRL4TLz2E140EKldgsnnd9mBaHcmwuiGJHecGLKXvHNaNHOB4J/HYmoGbGws
JU1ka0nAvM2ht40758ZnmvRRJ2413roMa7MxVg4jprDyMRC8bh9wR0TyIRWdr9hzNXaJs3Ftif6KDWuBcBH0hErky2bNraV5E9jcbjiapE1ExHk08iEY10
vjlTjAkugezh7ySqFUPoXHTtZAR7ncY4rRrYYgtcCtGHPugmjEhPmiKXjXc/l4g6HfGJT3ziEw/If86JzB/YMku9AAAAAE1FTkSuQmCC" rel="icon" ty
pe="image/png" />
    <style>
      body {
        margin: 0px;
        font: 20px 'RobotoRegular', Arial, sans-serif;
        font-weight: 100;
        height: 100%;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

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:


```

GNU nano 6.2                                     default *
upstream appbuddy{
    server 172.16.1.40;
}
upstream appAteez{
    server 172.16.1.30;
}
[]
server {
    listen 80 default_server;
    server_name appbuddy.com;

    server_name appbuddy.com;

    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://appbuddy/;
    }
}

server {
    listen 80;
    server_name appAteez.com;
    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://appAteez/;
    }
}

```

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/SUCCESS)
    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)
```

9. Acceder a las páginas desde VM

Al estar realizando la práctica en Ixc, 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”***.

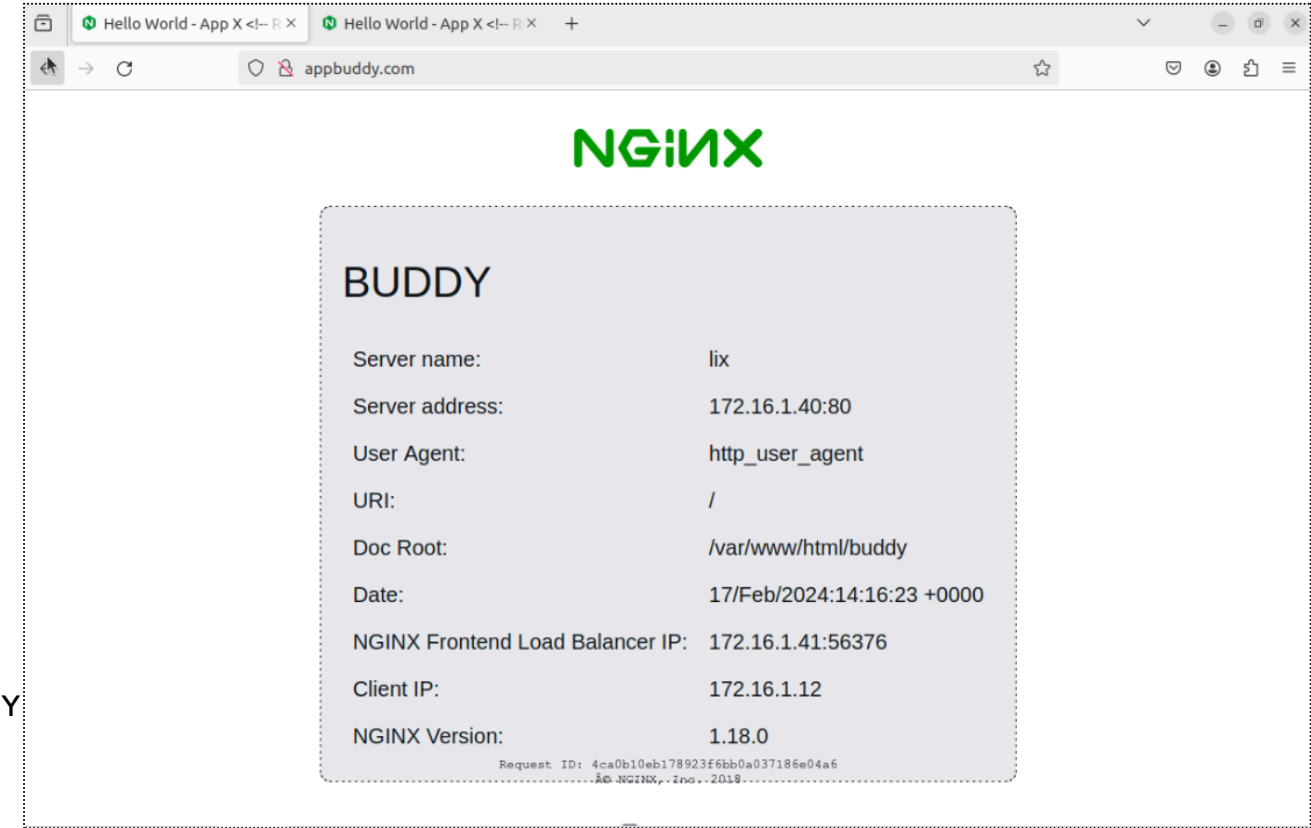
```
GNU nano 6.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    zerowave
172.16.1.41   appbuddy.com appAteez.com

# 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
```

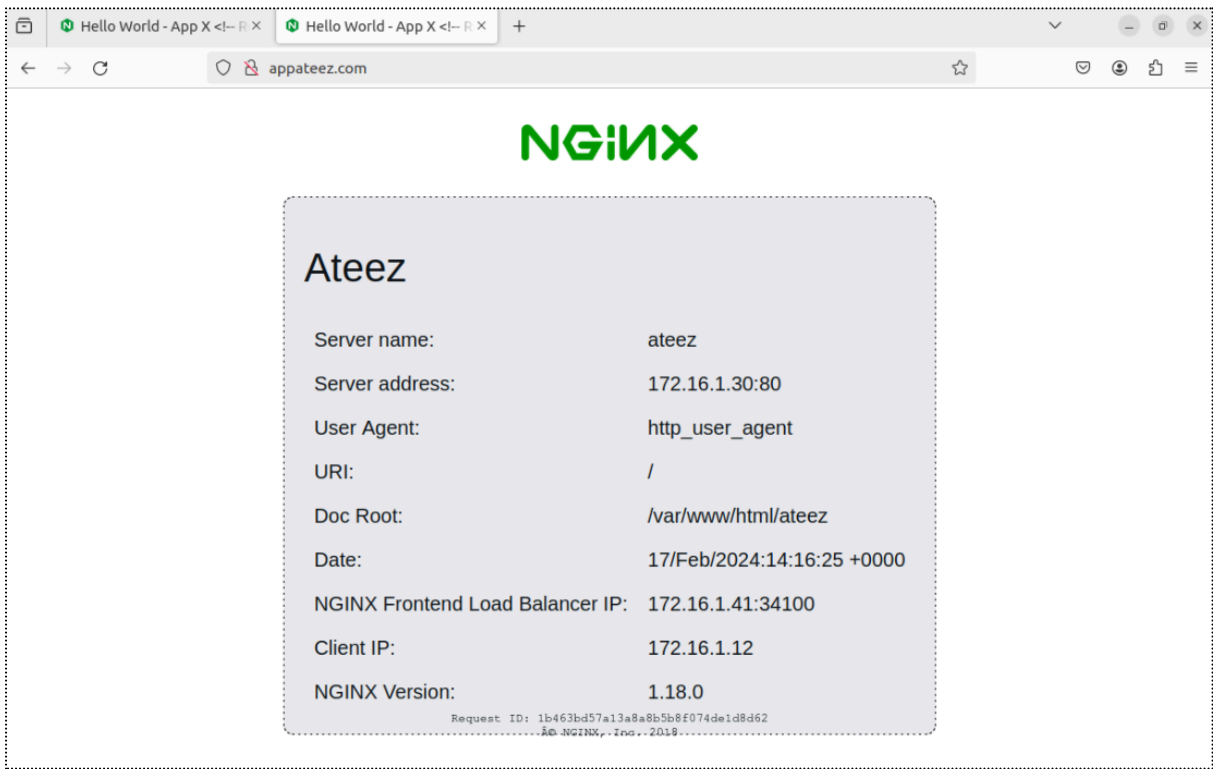
Debemos de poner la IP del proxy inverso y al lado las 2 páginas estáticas que hemos configurado.

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.



“*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>