

AZURE Y DNS



ALEJANDRO SAINZ SAINZ
DESPLIEGUE DE APLICACIONES

1COMENZANDO LA PRÁCTICA	3
2COMENZANDO	3
3PREGUNTAS ADICIONALES	13

1 COMENZANDO LA PRÁCTICA

Para esta práctica vamos a preparar dos máquinas virtuales en Azure, una con un servidor web y otra con un servidor dns. El objetivo de la práctica es acceder al servidor web desde una MV local en virtual box siendo redirigida por nuestro servidor dns. Así que, al lío.

2 COMENZANDO

Lo primero es ir creando los dos recursos de Azure. Lo mío lo de siempre, no sé si será por la imagen de Windows que elijo o por las zonas, pero siempre me dice que tengo fallos y que no puedo crear ese recurso. Lo bueno es, en mi caso, que como yo no odio Linux y no me importa enredar, decidí tirar por la vieja confiable. Unas buenas imágenes de Debian 13 y patada a seguir.

de recursos * ⓘ (Nuevo) Decroly
[Crear nuevo](#)

Instancia

Máquina virtual * ⓘ WebServerWindows10 ✓

(Europe) Switzerland North
[Implementación en una zona extendida de Azure](#)

Disponibilidad ⓘ Zona de disponibilidad

Zona ⓘ
☒ Zona autoseleccionada
Elija hasta 3 zonas de disponibilidad, una máquina virtual por zona
☐ Zona seleccionada por Azure (versión preliminar)
Permitir que Azure asigne la mejor zona para sus necesidades

Disponibilidad * ⓘ Zona 1
Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#)

Seguridad ⓘ Máquinas virtuales de inicio seguro
[Configurar características de seguridad](#)

[Siguiendo: Discos >](#) [Revisar y crear](#)

Ilustración 1 Que quede constancia de que lo intenté

Que no se diga. Intenté crear la imagen, pero esta vez Suiza no me acogió como en anteriores ocasiones.

Crear nuevo

Instancia

Imagen virtual * ⓘ WebServerDebian ✓

(Europe) Italy North ✓
Implementación en una zona extendida de Azure

Disponibilidad ⓘ Zona de disponibilidad

Red ⓘ ☒ Zona autoseleccionada
Elija hasta 3 zonas de disponibilidad, una máquina virtual por zona

☐ Zona seleccionada por Azure (versión preliminar)
Permitir que Azure asigne la mejor zona para sus necesidades

Redundancia * ⓘ Zona 1

☒ Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#) ⓘ

Tamaño ⓘ Estándar

Imagen ⓘ Debian 13 "Trixie" - x64 gen. 2
[Ver todas las imágenes](#) | [Configurar la generación de máquinas virtuales](#)

☒ Esta imagen es compatible con características de seguridad adicionales. [Haga clic aquí para cambiar a la versión de inicio seguro.](#)

Arquitectura de VM ⓘ ☐ Arm64 ☒ x64

Pre Spot con descuento ☐

Ilustración 2 Al final Debian.

Si me preguntan que por qué Debian. Por su estabilidad de sobra conocida y porque ya le he usado más veces.

Configuramos, es algo más sencillo que algunas imágenes de Microsoft. Eso sí, no nos da la posibilidad de conectarnos mediante RDP. En este caso SSH. Retro total, por terminal.

tipo de recursos *	<div>Decroly</div> <div>Crear nuevo</div>
de instancia	
le máquina virtual *	<div>ServerDNSUbuntu</div>
	<div>(Europe) Switzerland North</div> <div>Implementación en una zona extendida de Azure</div>
de disponibilidad	<div>Zona de disponibilidad</div>
de zona	<div><input checked="" type="radio"/> Zona autoseleccionada Elija hasta 3 zonas de disponibilidad, una máquina virtual por zona</div> <div><input type="radio"/> Zona seleccionada por Azure (versión preliminar) Permitir que Azure asigne la mejor zona para sus necesidades</div>
disponibilidad *	<div>Zona 1</div> <div><input checked="" type="checkbox"/> Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. Más información</div>
seguridad	<div>Estándar</div>
	<div><div>Debian 13 "Trixie" - x64 gen. 2</div><div>Ver todas las imágenes Configurar la generación de máquinas virtuales</div><div><input checked="" type="checkbox"/> Esta imagen es compatible con características de seguridad adicionales. Haga clic aquí para cambiar a la versión de inicio seguro.</div></div>
ara de VM	<div><input type="radio"/> Arm64</div> <div><input checked="" type="radio"/> x64</div>
de Azure Spot con descuento	<input type="checkbox"/>
	<div>Standard_B2as_v2 - 2 vcpu, 8 GiB de memoria (69,35 US\$)</div> <div>Ver todos los tamaños</div>
hibernación	<input type="checkbox"/>

Ilustración 3 La MV DNS

El segundo recurso. Igual que el otro, solo cambiar el nombre y habilitar los puertos necesarios, SSH, HTTP, etcétera.

Autenticación ⓘ

☒ Clave pública SSH

☐ Contraseña

i Ahora, Azure genera automáticamente un par de claves SSH y le permite almacenarlo para usarlo en el futuro. Es una forma rápida, sencilla y segura de conectarse a la máquina virtual.

Usuario * ⓘ

alejandro ✓

Clave pública SSH

Generar un par de claves nuevo ✓

Formato SSH

☒ Formato RSA SSH

☐ Formato Ed25519 SSH

i Ed25519 proporciona un nivel de seguridad fijo de no más de 128 bits para claves de 256 bits, mientras que RSA podría ofrecer una mejor seguridad con claves de más de 3072 bits.

Nombre de claves *

serverDNSdebiankey ✓

Puerto de entrada

Los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de todo o granular en la pestaña Red.

Entrada públicos * ⓘ

☐ Ninguno

☒ Permitir los puertos seleccionados

Puertos de entrada *

HTTP (80), HTTPS (443), SSH (22) ✓

! Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Ilustración 4 Los puertos

Como ya dije antes, los puertos. Y como se puede ver también en la imagen, aunque no usemos RDP debemos de indicar el formato de las credenciales para poder descargarlas después en nuestro equipo. Un fichero por cada MV, que luego por terminal usaremos para conectarnos a los equipos remotos.

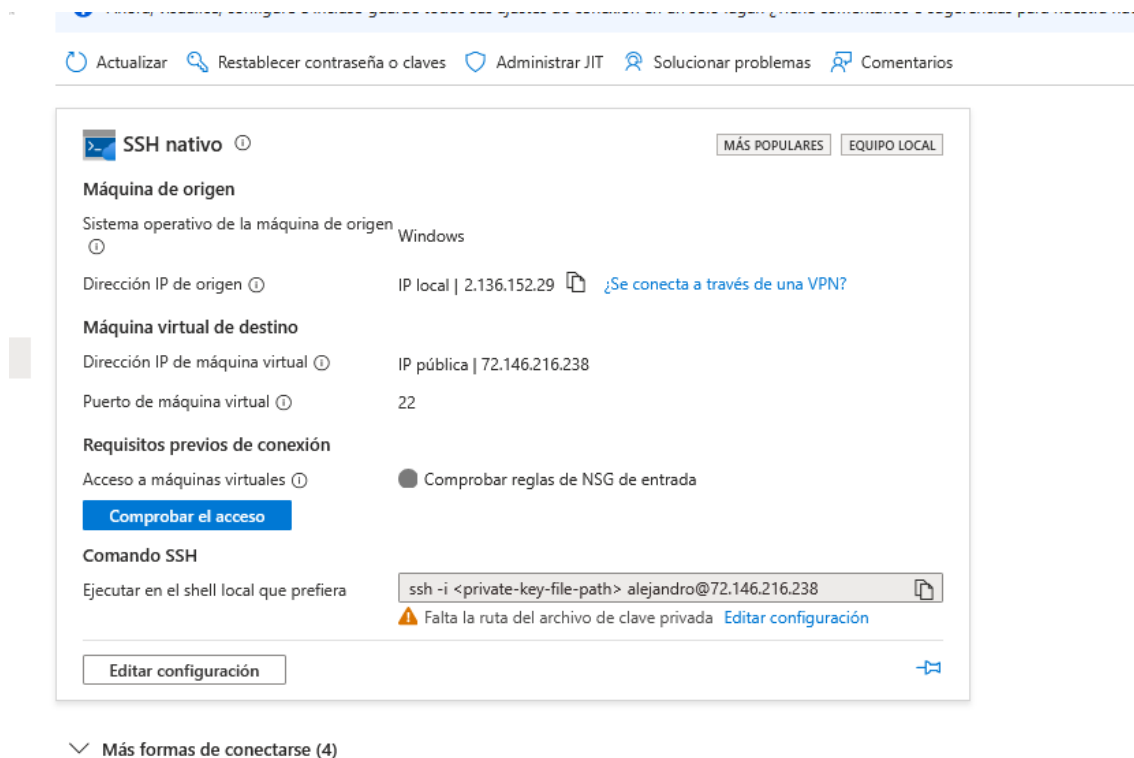


Ilustración 5 Preparando la conexión

Comprobamos los detalles de las direcciones IP y pulsamos en comprobar acceso. Para ello tenemos que indicarle en que ruta tenemos guardado el archivo .pem que almacena nuestras claves. Una vez hecho esto podemos usar la CLI de Azure o nuestra terminal de PowerShell para conectarnos.

Máquina virtual de destino

Dirección IP de máquina virtual ⓘ IP pública | 72.146.216.238

Puerto de máquina virtual ⓘ 22

Requisitos previos de conexión

Acceso a máquinas virtuales ⓘ ☒ Comprobar reglas de NSG de entrada

[Comprobar el acceso](#)

Comando SSH

Ejecute el comando siguiente en el shell local que prefiera para conectarse a través de SSH.

Proporcione una ruta de acceso al archivo de clave privada SSH en la máquina local.

C:\Users\daw2\Documents\DAW225-26\Despliegue de Aplicaciones\DNSAzure\ClavesSSHWebServer\clavedebian.pem

¿No encuentra la clave privada? [Restablezca la clave privada SSH](#)

Conectarse a la VM a través de SSH

ssh -i C:\Users\daw2\Documents\DAW225-26\Despliegue de Aplicaciones\DNSAzure\ClavesSSHWebServer\clavedebian.pem alejandro...

ⓘ Para mayor seguridad, la ruta de acceso del archivo de clave privada SSH solo se guarda localmente en este explorador.

[Guardar cambios](#) [Cerrar sin guardar](#) [Solución de problemas](#) [Enviar comentarios](#)

Ilustración 6 Configurando la ubicación de nuestro par de claves.

Como ya dije antes, configuramos la ubicación donde hemos guardado las claves. Además, aquí podemos ver el comando que debemos de introducir en la terminal para conectarnos de forma remota.

SSH nativo ⓘ

MÁS POPULARES EQUIPO LOCAL

Máquina de origen

Sistema operativo de la máquina de origen ⓘ Windows

Dirección IP de origen ⓘ IP local | 2.136.152.29 ⓘ [¿Se conecta a través de una VPN?](#)

Máquina virtual de destino

Dirección IP de máquina virtual ⓘ IP pública | 72.146.216.238

Puerto de máquina virtual ⓘ 22

Requisitos previos de conexión

Acceso a máquinas virtuales ⓘ ☒ Se puede acceder al puerto 22 desde las direcciones IP de origen [Ver reglas de NSG aplicadas](#)

Comprobar el acceso

Comando SSH

Ejecutar en el shell local que prefiera `ssh -i C:\Users\daw2\Documents\DAW225-26\Despliegue de Aplica ...` ⓘ

¿No encuentra la clave privada? [Restablezca la clave privada SSH](#)

Editar configuración ⓘ

Ilustración 7 Prueba de conexión

Una vez configurado todo medianamente bien, podemos ver como la prueba de acceso es correcta. Podríamos ignorar este paso, me di cuenta después. No es totalmente necesario, pero como valor para la documentación no está mal del todo.

```
ssh: Could not resolve hostname de: Host desconocido.
PS C:\WINDOWS\system32> ssh -i "C:\Users\daw2\Documents\DAW225-26\Despliegue de Aplicaciones\DNSAzure\ClavesSSHWebServer\clavedebian.pem" alejandro@72.146.216.238
The authenticity of host '72.146.216.238 (72.146.216.238)' can't be established.
ED25519 key fingerprint is SHA256:MCbHkRPxz6pxR0zrZB5dsndJ6PXsGU0Agaxgo61rG+E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Ilustración 8 Conectando por PowerShell

Abrimos la terminal e introducimos el comando que nos proporcionaba Azure. Es de indicar que mejor hacerlo fuera de la carpeta system32, más que nada porque intenta crear una carpeta temporal en nuestro directorio de usuario.

```

ED25519 key fingerprint is SHA256:MC0HKKRFX20pxR0Z/2BSuSh0J0PXS00eagxg00i0eF.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Could not create directory 'C:\Users\daw2\.ssh' (Permission denied).
Failed to add the host to the list of known hosts (C:\Users\daw2\.ssh\known_hosts).
Linux WebServerDebian 6.12.63+deb13-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.63-1 (2025-12-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
alejandro@WebServerDebian:~$

```

Ilustración 9 Conectado

Todo listo. Ahora sólo tenemos que buscar cómo preparar un servidor web.

```

alejandro@WebServerDebian: /var/www/html
alejandro@WebServerDebian:/$ pwd
/
alejandro@WebServerDebian:/$ ls
bin boot dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
alejandro@WebServerDebian:/$ cd var
alejandro@WebServerDebian:/var$ cd www
alejandro@WebServerDebian:/var/www$ cd html/
alejandro@WebServerDebian:/var/www/html$

```

Ilustración 10 Apache2

Se me olvidó poner capturas, pero la solución es un viejo conocido. Sudo apt install apache2.

Luego solo tenemos que ir a la dirección que vemos en pantalla, para mover o borrar el archivo index que encontramos allí. Yo en mi caso lo renombré. Así después puedo crear el mio propio.

```

GNU nano 8.4 index.html *
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Servidor Web en Debian Alejandro Sainz</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>

  <header>
    <h1>Servidor Web en Azure</h1>
    <h2>Alejandro Sainz Sainz</h2>
  </header>

  <main>
    <h2>¡Hola mundo!</h2>
    <p>Este es un servidor web sencillo usando HTML y CSS. En Nano.</p>
  </main>

  <footer>
    <p>&copy; 2026 - Mi servidor</p>
  </footer>
</body>
</html>

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-G Copy

```

Ilustración 11 Sabor muy retro

Seguimos con los paseos por el pasado. Editor de texto por terminal, para hacer una página web. Tampoco es tan duro, es pequeña.

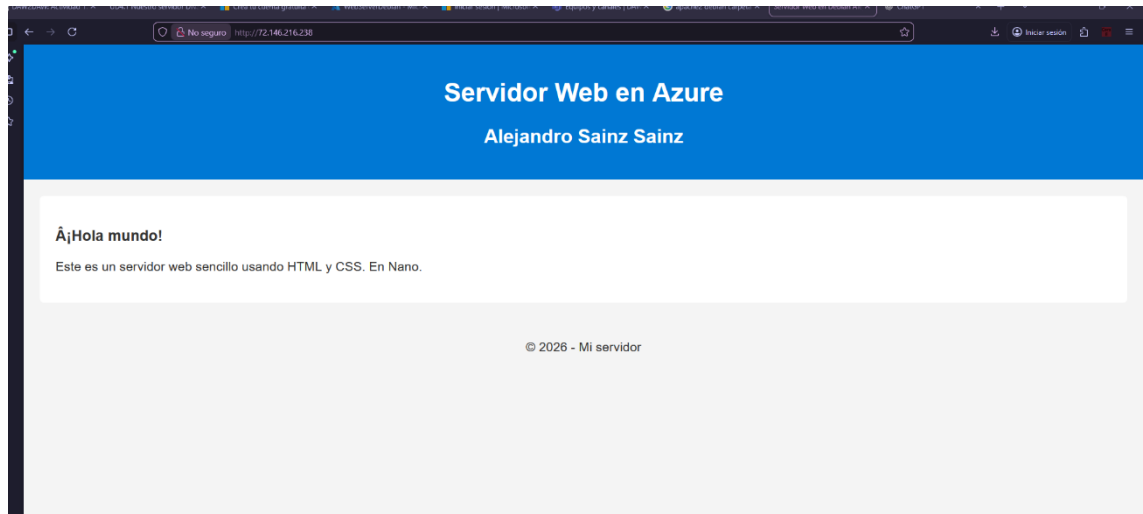


Ilustración 12 Página arriba

Cuando ya dejamos preparado el archivo html y el css navegamos hasta localhost en el navegador. Este es el resultado.

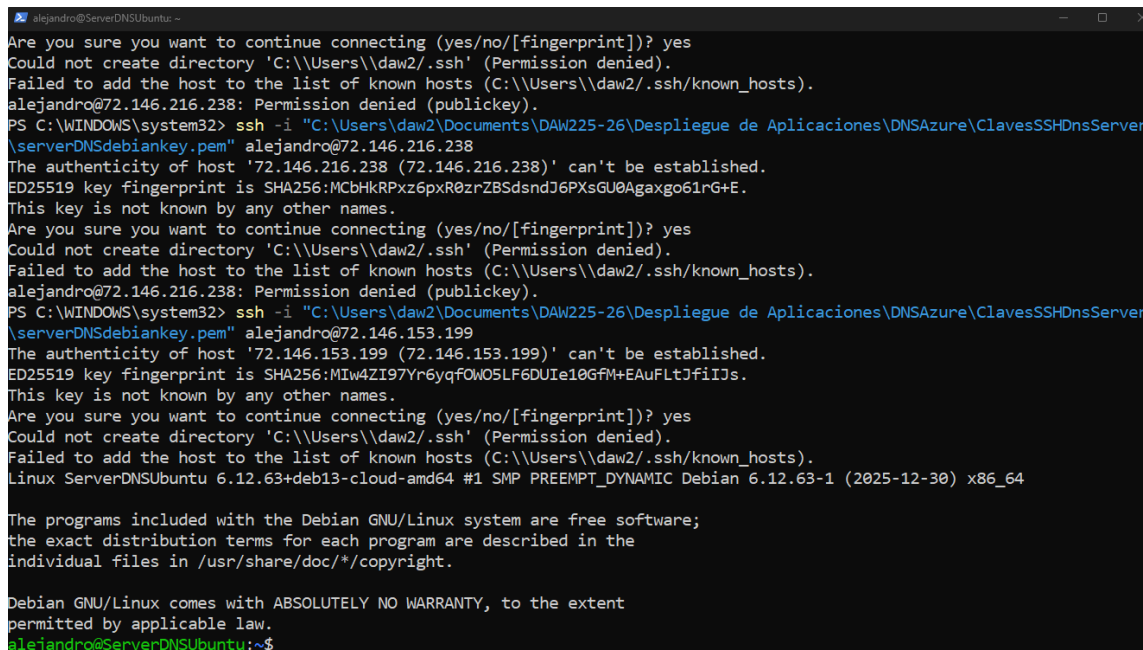


Ilustración 13 Ahora el DNS

Aquí se viene lo más complicado. De estos sólo he configurado uno o dos en toda mi vida antes de este. Pero bueno, allá vamos. Algún que otro error de conexión, pero arreglado al final.

```
root@ServerDNSUbuntu:~# sudo apt install bind9 bind9utils dnsutils -y
Note, selecting 'bind9-utils' instead of 'bind9utils'
Note, selecting 'bind9-dnsutils' instead of 'dnsutils'
Installing:
  bind9  bind9-dnsutils  bind9-utils

Installing dependencies:
  dns-root-data

Suggested packages:
  bind9-doc  ufw

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 0
  Download size: 610 kB
  Space needed: 2209 kB / 28.9 GB available

Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [37 B]
Get:2 http://azure.deb.debian.cloud/debian trixie/main amd64 bind9-utils amd64 1:9.20.15-1~deb13u1 [184 kB]
Get:3 http://azure.deb.debian.cloud/debian trixie/main amd64 dns-root-data all 2025080400~deb13u1 [5948 B]
Get:4 http://azure.deb.debian.cloud/debian trixie/main amd64 bind9 amd64 1:9.20.15-1~deb13u1 [254 kB]
Get:5 http://azure.deb.debian.cloud/debian trixie/main amd64 bind9-dnsutils amd64 1:9.20.15-1~deb13u1 [165 kB]
Fetched 610 kB in 0s (3621 kB/s)
Selecting previously unselected package bind9-utils.
```

Ilustración 14 Instalación del servicio DNS

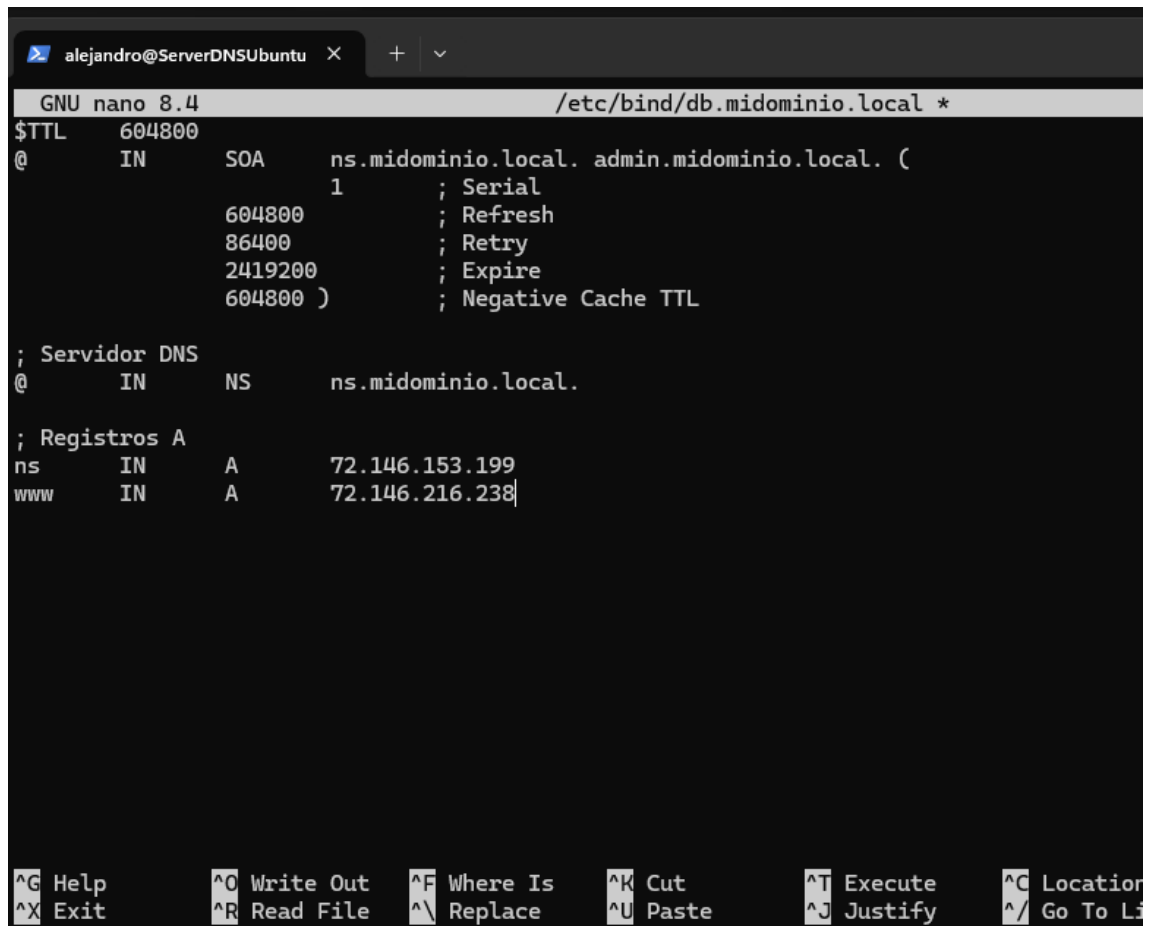
Busqué los paquetes que debía de instalar para un servidor DNS. Evidentemente que si configuración fuese lo más sencilla posible. Según internet, bind9. Aunque siendo Linux, la configuración por terminal.

```
alejandro@ServerDNSUbuntu: ~
GNU nano 8.4 /etc/bind/named.conf.local *
//
// Do any local configuration here
//

zone "midominio.local" {
    type master;
    file "/etc/bind/db.midominio.local";
};
```

Ilustración 15 Configurando zonas

Lo primero que hay que hacer es crear un primer archivo que es el que va a resolver los dominios. En este archivo creamos una zona e indicamos donde se va a ubicar el archivo con las direcciones.



```

GNU nano 8.4 /etc/bind/db.midominio.local *
$TTL      604800
@          IN      SOA      ns.midominio.local. admin.midominio.local. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL

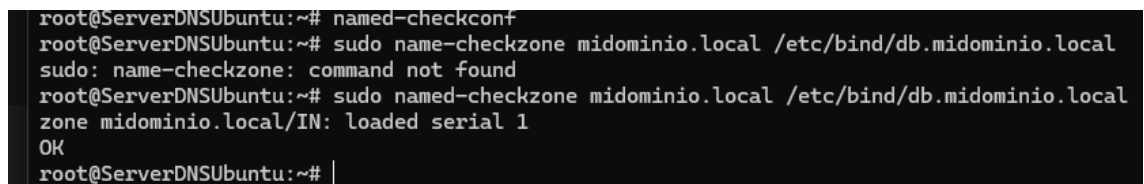
; Servidor DNS
@          IN      NS       ns.midominio.local.

; Registros A
ns         IN      A        72.146.153.199
www        IN      A        72.146.216.238

```

Ilustración 16 Más configuraciones

Esta es un poco más complicada, ya que no entiendo que son todas las cosas. Lo bueno es que si sabía donde iban las IPs de mi servidor DNS y de mi servidor WEB. Muy importante no equivocarse con las IPs.



```

root@ServerDNSUbuntu:~# named-checkconf
root@ServerDNSUbuntu:~# sudo name-checkzone midominio.local /etc/bind/db.midominio.local
sudo: name-checkzone: command not found
root@ServerDNSUbuntu:~# sudo named-checkzone midominio.local /etc/bind/db.midominio.local
zone midominio.local/IN: loaded serial 1
OK
root@ServerDNSUbuntu:~#

```

Ilustración 17 Probando

Hay que hacer un par de pruebas que me permitan confirmar que todo esta correcto. Eso sí, hay que escribir bien los comandos.

```

root@ServerDNSUbuntu:~# dig www.midominio.test

; <<>> DiG 9.20.15-1~deb13u1-Debian <<>> www.midominio.test
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63069
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: e85e784ab6ca961a01000000696fb5c872ea8fa0c58ac5da (good)
;; QUESTION SECTION:
;www.midominio.test.          IN      A

;; ANSWER SECTION:
www.midominio.test.        604800  IN      A      72.146.216.238

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Tue Jan 20 17:05:12 UTC 2026
;; MSG SIZE rcvd: 91

root@ServerDNSUbuntu:~#

```

Ilustración 18 Al final, medianamente bien

Una vez terminado hacemos una prueba para ver que el servidor, por lo menos, se ve a sí mismo. En la imagen, arriba a la derecha, podemos ver que ahora el dominio ha cambiado a .test. De primeras era .local pero este mismo comando reconocía eso como fallo, así que tuve que tocar los ficheros de nuevo para cambiar a .test.

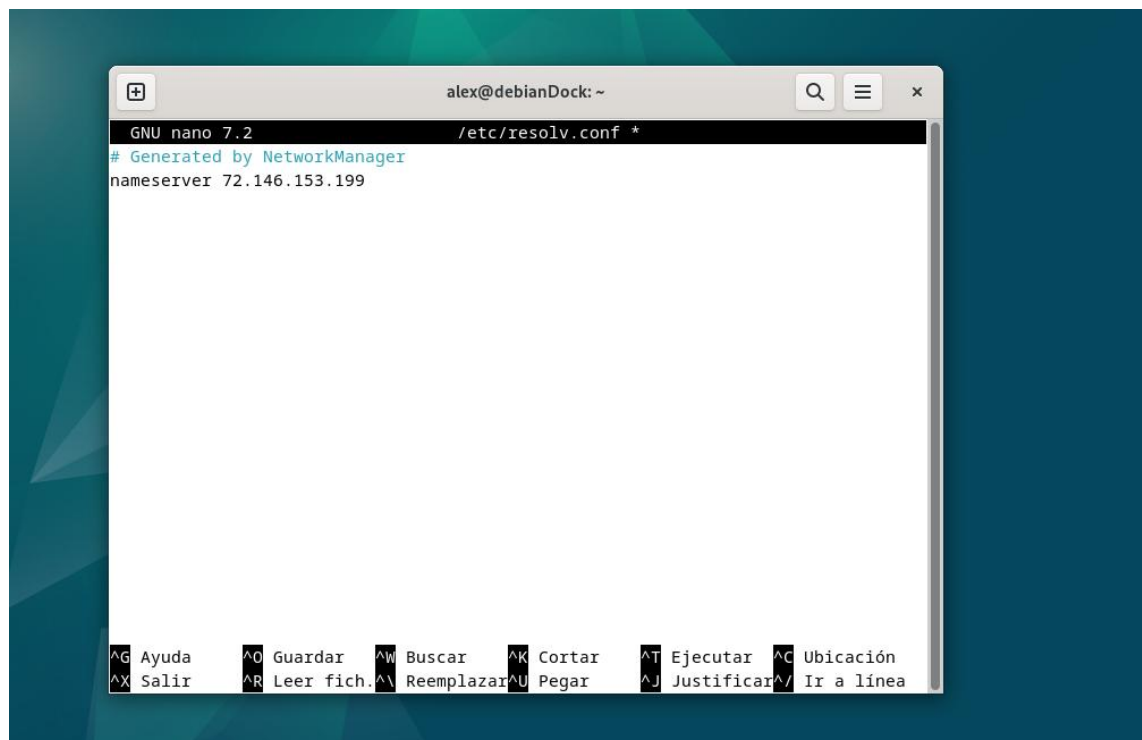


Ilustración 19 Por fin un poco de color

Arranco mi MV local y por fin algo que no es solo blanco y negro. No me importa la terminal, pero a veces se agradece un poco de color. En la imagen anterior vemos el archivo que hay que configurar. Lo que hay que hacer es cambiar la ip que venía por defecto por la de nuestro recurso de Azure.

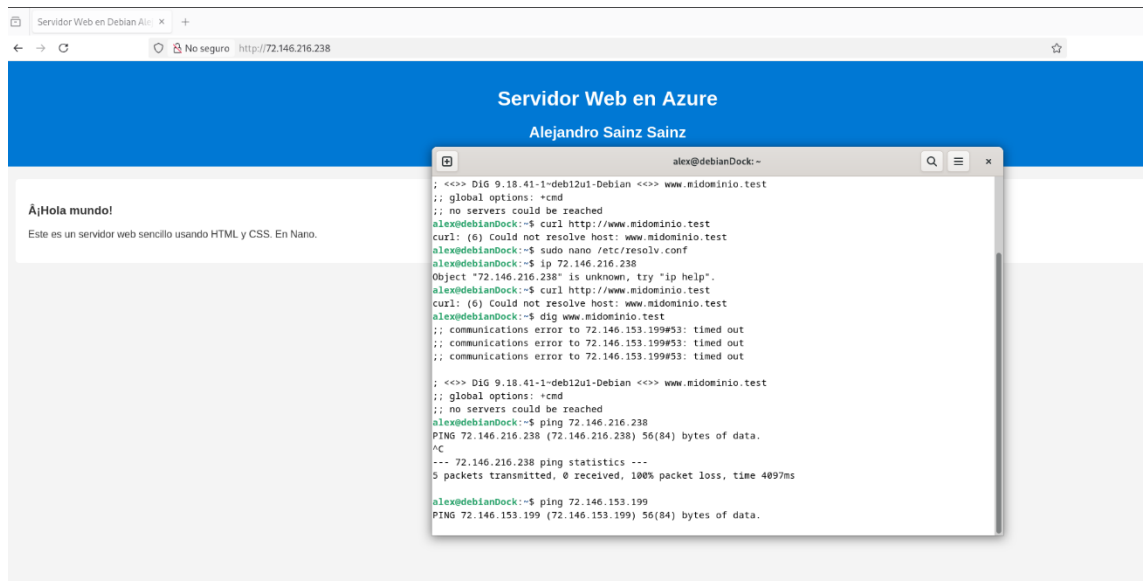


Ilustración 20 Pruebas y más pruebas.

Aquí algún que otro problema. Si ponía la IP directamente en el navegador, accedía a la página web del servidor web. Si ponía el nombre del dominio, no la encontraba. Algo fallaba. Los pings no llegaban al servidor DNS, por lo tanto, era imposible resolver el nombre.

Buscar reglas		Origen == todo	Destino == todo	Protocolo == todo	Acción == todo	Puerto == todo	
Prioridad ↑	Nombre		Puerto	Protocolo	Origen	Destino	Acción
Reglas de puerto de entrada (9)							
300	SSH		22	TCP	Cualquiera	Cualquiera	Allow
320	HTTPS		443	TCP	Cualquiera	Cualquiera	Allow
340	HTTP		80	TCP	Cualquiera	Cualquiera	Allow
350	UDP		53	UDP	Cualquiera	Cualquiera	Allow
360	DNSTCP		53	TCP	Cualquiera	Cualquiera	Allow
370	DNSUDP		53	UDP	Cualquiera	Cualquiera	Allow
65000	AllowVnetInBound		Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound		Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Allow
65500	DenyAllInBound		Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny
Reglas de puerto de salida (3)							

Ilustración 21 Un clásico

La cosa era que no había creado reglas para las entradas y salidas de peticiones para el DNS.

```
--- 72.146.153.199 ping statistics ---
64 packets transmitted, 0 received, 100% packet loss, time 64533ms

alex@debianDock:~$ dig www.midominio.test

; <<>> DiG 9.18.41-1~deb12u1-Debian <<>> www.midominio.test
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49500
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 1232
;; COOKIE: 255eec048f74ede901000000696fbc720b82a9b6df686464 (good)
;; QUESTION SECTION:
;; www.midominio.test.                IN      A

;; ANSWER SECTION:
www.midominio.test.        604800  IN      A          72.146.216.238

;; Query time: 44 msec
;; SERVER: 72.146.153.199#53(72.146.153.199) (UDP)
;; WHEN: Tue Jan 20 18:33:35 CET 2026
;; MSG SIZE rcvd: 91

alex@debianDock:~$
```

Ilustración 22 Repito prueba de conexión

Ahora sí, desde terminal pruebo a buscar el dominio, con dig o curl. Y ahora ya si lo encuentra.

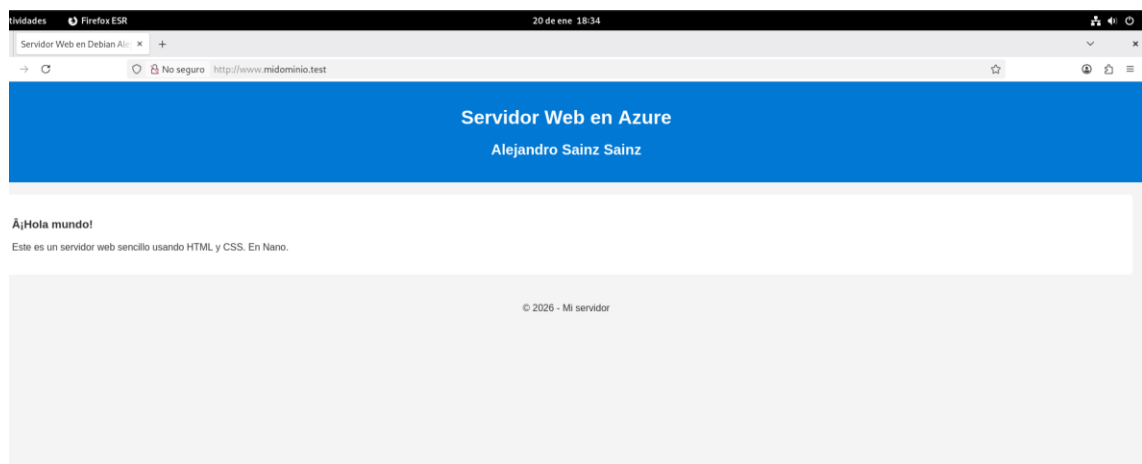


Ilustración 23 Ya está

Ahora ya, si desde mi máquina en debía, busco por el dominio, ya se redirige a la IP que le corresponda.

3 PREGUNTAS ADICIONALES

¿La parte más difícil y cuál fue la solución?

La configuración del servidor DNS. Nunca lo había hecho en Linux y mucho menos todo por terminal. Como lo uso en casa a diario con la terminal me apaño, el saber donde están todos los archivos y lo que hay que tocar, eso ya no. Me llevó un rato, además de poner mal las IPs y tener que modificarlo un par de veces.

Otras Opciones a Windows

Pues como hemos visto aquí, con Debian lo podemos hacer. ¿Otras que podemos usar? PiHole por ejemplo, que además de bloquear anuncios y rastreadores tiene opción para ser tu propio servidor DNS y, si no recuerdo mal, pfSense u opfSense creo que también traían esa opción.