

DNS

- 1. Introducción
- 2. Servicios DNS
 - 2.1. duckdns
 - 2.2. noip
- 3. AWS
 - 3.1. IP elástica
 - 3.2. Ejemplo
 - 3.3. scripts AMI Amazon 2023
- 4. Ejercicio
- 5. Enlaces

1. Introducción

Objetivo: asociar nuestra instancia EC2 a un nombre de dominio

Para conseguir un nombre de dominio para nuestra IP sin pagar hay muchas opciones, entre ellas

- duckdns: Admite wildcards xxx.midominio.duckdns.org y admite registros tipo TXT. <https://duckdns.org>
- noip.com: <https://www.noip.com/es-MX/free>
- FreeDNS: <http://freedns.afraid.org>
- Nominalia: primer año gratis, pero hay que dar tarjeta <https://www.nominalia.com/>
- Freenom (parece que no funciona): <https://www.freenom.com/>

Además en el caso de AWS habría que resolver que la IP va cambiando al detener la máquina. Posibles soluciones:

1. Asociarla manualmente cuando cambie
2. Usar una IP Elástica de AWS (tiene coste)
3. DNS dinámico (DDNS): instalar cliente en EC2 que detecta cambios en IP pública y actualiza el DNS

2. Servicios DNS

2.1. duckdns

- Alta con github o Google
- Admite wildcard **xxxx.midominio.duckdns.org** (puede venir bien para la práctica de traefik)
- Admite registros TXT
- Asociar la IP:
 - Se puede especificar a mano en la web
 - Se puede auto detectar desde su página (detecta el equipo desde donde se accede, no vale para ejemplos de la práctica de clase)
 - Se puede hacer update con peticiones HTTP. [Enlace](#)

2.2. noip

- Sólo un registro para un host: **midominio.ddns.net**

- Se puede asociar la IP a mano en la web o instalar cliente en el host

3. AWS

3.1. IP elástica

En este caso la IP es fija, no cambia, y podemos asociar la IP al dominio directamente en la interfaz web de las plataformas una única vez. También es posible hacerlo con un cliente DDNS o que los scripts que proporcionan.

3.2. Ejemplo

En este ejemplo usaremos la AMI Linux de Amazon de 2023 basada en Fedora.

- Crear EC2 Amazon Linux 2023
- Instalar nginx

```
[ec2-user@ip-172-31-18-67 ~]$ sudo dnf install -y nginx
Last metadata expiration check: 0:00:49 ago on Sat Oct 28 13:13:01 2023.
Dependencies resolved.

...

Complete!
[ec2-user@ip-172-31-18-67 ~]$
```

- Iniciar el servidor

```
[ec2-user@ip-172-31-18-67 ~]$ sudo systemctl start nginx
[ec2-user@ip-172-31-18-67 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
   preset: disabled)
   Active: active (running) since Sat 2023-10-28 13:15:30 UTC; 1s ago
     Process: 25768 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
   (code=exited, status=0/SUCCESS)
     Process: 25769 ExecStartPre=/usr/sbin/nginx -t (code=exited,
   status=0/SUCCESS)
     Process: 25770 ExecStart=/usr/sbin/nginx (code=exited,
   status=0/SUCCESS)
    Main PID: 25771 (nginx)
       Tasks: 2 (limit: 1114)
      Memory: 2.2M
         CPU: 56ms
    CGroup: /system.slice/nginx.service
            └─25771 "nginx: master process /usr/sbin/nginx"
               └─25772 "nginx: worker process"

...
Oct 28 13:15:30 ip-172-31-18-67.ec2.internal systemd[1]: Started
```

```
nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-18-67 ~]$
```

- Probar servidor en local

```
[ec2-user@ip-172-31-18-67 ~]$ curl localhost
<!DOCTYPE html>
<html>
<head>
....
<h1>Welcome to nginx!</h1>
...
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[ec2-user@ip-172-31-18-67 ~]$
```

- Habilitar arranque en reinicio:

```
[ec2-user@ip-172-31-18-67 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-172-31-18-67 ~]$ sudo systemctl status nginx
• nginx.service - The nginx HTTP and reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled;
  preset: disabled)
  Active: active (running) since Sat 2023-10-28 13:15:30 UTC; 7min ago
  Main PID: 25771 (nginx)
    Tasks: 2 (limit: 1114)
  Memory: 2.2M
    CPU: 57ms
  CGroup: /system.slice/nginx.service
          └─25771 "nginx: master process /usr/sbin/nginx"
            └─25772 "nginx: worker process"

...
Oct 28 13:15:30 ip-172-31-18-67.ec2.internal systemd[1]: Started
nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-18-67 ~]$
```

- Anotar IP pública y modificar registro DNS (duckdns o noip)
- Esperar unos minutos a actualizaciones DNS y desde equipo del aula:

```
clases@d12:~$ curl www.midominio.duckdns.org
<!DOCTYPE html>
<html>
<head>
```

```
...
<h1>Welcome to nginx!</h1>
....

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
clases@d12:~$
```

Aviso: al reiniciar cambia la IP.

3.3. scripts AMI Amazon 2023

Para instalar **nginx** directamente al crear la instancia (o desde plantilla de instancia)

```
#!/bin/bash
sudo dnf install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

4. Ejercicio

- Crear EC2 AL2023 con servidor nginx
- Crear dominio y asignarle la IP pública de la instancia. Probar
- Usar IP elástica. Modificar IP del dominio. Probar
- Liberar y eliminar la IP elástica

5. Enlaces

- Duck DNS: <https://www.duckdns.org/>
- noip DNS: <https://my.noip.com/dynamic-dns/duc> Otros:
 - Best Free Dynamic DNS Services: <https://www.gnutomorrow.com/best-free-dynamic-dns-services/>
 - Freedns <https://freedns.afraid.org/> y Dynamic DNS: <https://freedns.afraid.org/dynamic/>
 - Namecheap: <https://www.namecheap.com/> y Nc.me: Namecheap is giving university students a free bundle to kickstart their online presence.: <https://nc.me/>
- ¿Qué son las direcciones IP elásticas de AWS y qué hacen? https://es.linux-console.net/?p=7127#google_vignette
- Configurar DNS dinámico en la instancia de Amazon Linux - Amazon Elastic Compute Cloud https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/dynamic-dns.html