

Practicas DNS

Route 53

Amazon Route 53 es un servicio web de sistema de nombres de dominio (DNS) escalable y de alta disponibilidad

Se puede usar para tres funciones:

Registro de dominio

Direccionamiento de DNS

Comprobación de estado

Registro de dominio:

Route 53 permite registrar nuevos nombres de dominio o transferir dominios existentes. Esto incluye la gestión de información de contacto, renovaciones y configuración básica del dominio

Direccionamiento de DNS:

Route 53 dirige el tráfico de internet a los recursos adecuados (como servidores web, balanceadores de carga o buckets de S3) mediante la resolución de nombres de dominio a direcciones IP. Ofrece diferentes políticas de enrutamiento, como simple, ponderado, geolocalizado, de latencia y de failover.

Comprobación de estado (Health Checks):

Route 53 monitorea la salud y disponibilidad de los recursos (servidores, aplicaciones, etc.). Si un recurso no está disponible, puede redirigir el tráfico a alternativas saludables, mejorando la confiabilidad de las aplicaciones.

Route53 se puede configurar para enviar solicitudes periódicas a recursos, como servidores web o de correo.

En caso de error, se puede configurar:

Notificación a CloudWatch

CloudWatch puede utilizar Amazon SNS para enviar notificación push a diferentes destinatarios

Route53 permite crear registros DNS:

Nombre: corresponde al dominio o subdominio que queremos que se resuelva.

Tipo: A o AAAA, CNAME, MX, NS, SOA...

Valor: dependerá del tipo de registro, puede ser una IP, el nombre de los NS, etc

TTL: segundos que los solucionadores recursivos guardarán en caché esta información.

Practica 1 – Route 53

En esta actividad vamos a crear un dominio en una zona privada (gratis) y vamos a asignar nombres de dominio a dos instancias EC2 que podrían alojar un servidor de correo y un servidor web, respectivamente (aunque no implementaremos los servidores por simplicidad).

Como no vamos a comprar un dominio, utilizaremos una zona privada alojada, que es un contenedor de registros para un dominio que se aloja en una o varias VPC de Amazon y nos permite usar DNS y nombres de dominio propios dentro de ese espacio privado.

Creación del escenario

Crea en la VPC predeterminada dos instancias EC2, con claves vockey y de tipo t2.micro. Se llamarán WebServer y eMailServer. **Ten la precaución de que ambas máquinas pertenezcan al mismo grupo de seguridad.**

▼ Network settings [Info](#)

VPC - *required* | [Info](#)

vpc-07655e27dee699da0
172.31.0.0/16

(default) ▼



Subnet | [Info](#)

No preference ▼



[Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable ▼

Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Security group name - *required*

Practica Ruta 53

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters are `._-:/()#,@[]+=&{}!$*`

Description - *required* | [Info](#)

Practica ruta 53

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | [Info](#)

ssh ▼

Protocol | [Info](#)

TCP

Port range | [Info](#)

22

Source type | [Info](#)

Anywhere ▼

Source | [Info](#)

Add CIDR, prefix list or security group

0.0.0.0/0

Description - *optional* | [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type | [Info](#)

HTTP ▼

Protocol | [Info](#)

TCP

Port range | [Info](#)

80

Source type | [Info](#)

Anywhere ▼

Source | [Info](#)

Add CIDR, prefix list or security group

0.0.0.0/0

Description - *optional* | [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 3 (ICMP, All, 0.0.0.0/0)

Type | [Info](#)

All ICMP - IPv4 ▼

Protocol | [Info](#)

ICMP

Port range | [Info](#)

All

Source type | [Info](#)

Anywhere ▼

Source | [Info](#)

Add CIDR, prefix list or security group

Description - *optional* | [Info](#)

e.g. SSH for admin desktop

Observarás que AWS proporciona a la instancia de una VPC nombres de host DNS públicos y privados que corresponden a las direcciones IPv4 públicas e IPv4 privadas de la instancia. Cuando se lanza una instancia en una VPC, ésta automáticamente recibe:

- Un nombre de host DNS privado
- Un nombre de host DNS público si tiene una dirección IPv4 pública y si los nombres de host DNS y los atributos de compatibilidad de DNS para su VPC están establecidos en true.

<input type="checkbox"/>	eMailServer	i-0f98243dbc5095487	Running	t2.micro	Initializing	View alarms +	us-east-1b
<input checked="" type="checkbox"/>	Web Server	i-0e1245402288ebcb1	Running	t2.micro	Initializing	View alarms +	us-east-1b

i-0e1245402288ebcb1 (Web Server)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

▼ Instance summary Info

Instance ID

[i-0e1245402288ebcb1](#)

IPv6 address

-

Hostname type

IP name: ip-172-31-83-49.ec2.internal

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

[34.203.203.201](#) [Public IP]

Public IPv4 address

[34.203.203.201](#) | [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

[ip-172-31-83-49.ec2.internal](#)

Instance type

t2.micro

VPC ID

[vpc-07655e27dee699da0](#)

Private IPv4 addresses

[172.31.83.49](#)

Public IPv4 DNS

[ec2-34-203-203-201.comp](#)
[open address](#)

Elastic IP addresses

-

AWS Compute Optimizer findi

[Opt-in to AWS Compute Op](#)
[Learn more](#)

Una VPC tiene atributos que determinan si las instancias lanzadas en la VPC reciben nombres de host DNS públicos que corresponden a sus direcciones IP públicas y si la resolución de DNS a través del servidor DNS de Amazon es compatible con la VPC:

Nombres de host DNS: indica si las instancias con direcciones IP públicas obtienen los nombres de host DNS públicos correspondientes.

Resolución de DNS: indica si la VPC admite la resolución DNS.

Detalles			
ID de la VPC vpc-089f5879e409943c0	Estado Available	Nombres de host DNS Habilitado	Resolución de DNS Habilitado
Tenencia Default	Conjunto de opciones de DHCP dopt-05b99e05f9b76af3b	Tabla de enrutamiento principal rtb-08084ad0da205450f	ACL de red principal acl-00b36b433f0eccfa7

Creación de zonas alojadas privadas

Abrimos ruta 53 y creamos una zona alojada.



Elige un nombre para tu dominio como minombre.net , por ejemplo caso xavi.net, y rellena la descripción.

Elige Zona alojada privada

Elige la Región e ID de la VPC que queremos asociar a la zona alojada.

Create hosted zone [Info](#)

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name [Info](#)

This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } . ~

Description - optional [Info](#)

This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 57/256

Type [Info](#)

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☐ Public hosted zone

A public hosted zone determines how traffic is routed on the internet.

☒ Private hosted zone

A private hosted zone determines how traffic is routed within an Amazon VPC.

VPCs to associate with the hosted zone [Info](#)

To use this hosted zone to resolve DNS queries for one or more VPCs, choose the VPCs. To associate a VPC with a hosted zone when the VPC was created using a different AWS account or programmatically, such as the AWS CLI.

ⓘ For each VPC that you associate with a private hosted zone, you must set the Amazon VPC settings [enableDnsHostnames](#) and [enableDnsSupport](#) to true.

Region [Info](#)

US East (N. Virginia)

VPC ID [Info](#)

Q vpc-07655e27dee699da0

Remove VPC

Add VPC

Crea la zona y, a continuación, podrás ver 2 registros uno de tipo NS y otro de tipo SOA.

El tipo de directiva de enrutamiento es "Simple", esta política de enrutamiento simple nos permite configurar registros DNS estándar, sin enrutamiento especial de Route 53, como ponderado o latencia.

Y en la columna Valor/Enrutar tráfico a, en el primer registro, podemos comprobar los cuatro NS asignados:

ns-1536.awsdns-00.co.uk.

ns-0.awsdns-00.com.

ns-1024.awsdns-00.org.

ns-512.awsdns-00.net.

Records (2) [Info](#) [Refresh](#) [Delete record](#) [Import zone file](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

[Type](#) [Routing p...](#) [Alias](#)

<input type="checkbox"/>	Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...
<input type="checkbox"/>	oscar.net	NS	Simple	-	No	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.	172800
<input type="checkbox"/>	oscar.net	SOA	Simple	-	No	ns-1536.awsdns-00.co.uk. a...	900

El campo TTL es la cantidad de tiempo, en segundos, que queremos que los solucionadores recursivos de DNS almacenen en caché información sobre este registro. Si se especifica un valor más largo, se reduce el número de llamadas que los solucionadores recursivos de DNS deben realizar a Route 53 para obtener la información más reciente de este registro.

Ahora, vamos a crear registros:

Crearemos los registros "A" para nuestras instancias EC2.

Haz clic en el botón Crear registro.

Elige la opción de Cambiar al asistente.

Elige la política de enrutamiento: "Enrutamiento simple" y haz clic en Siguiente

Quick create record

▼ Record 1

Record name [Info](#) oscar.net

Keep blank to create a record for the root domain.

☒ Alias


Record type [Info](#) [Switch to wizard](#) [Delete](#)

A - Routes traffic to an IPv4 address and some AWS resources

Choose routing policy [Info](#)

The routing policy determines how Amazon Route 53 responds

Routing policy

☒ **Simple routing**
Use if you want all of your clients to receive the same response(s).


☐ Latency

Haz clic en **Definir registro simple** y escribe el **nombre de sus instancias EC2** y la **dirección IP privada**. Debes agregar dos registros, el de WebServer y el otro para eMailServer.

Define simple record ✕

Record name [Info](#)

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to `blog.example.com`, enter `blog`. If you leave this field blank, the default record name is the name of the domain.

.oscar.net

Keep blank to create a record for the root domain.

Record type [Info](#)

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources ▼

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to [Info](#)

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value, depending on the record type ▼

172.31.83.49 #ip privada

Enter multiple values on separate lines.

TTL (seconds) [Info](#)

The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. ("TTL" means "time to live."). This value does not apply to alias records. [Learn more](#)

Records (4)
[Info](#)
⌂
Delete record
Import zone file
Create record

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Type ▼
Routing p... ▼
Alias ▼

< 1 >
⚙

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Alias
<input type="checkbox"/>	oscar.net	NS	Simple	-	No
<input type="checkbox"/>	oscar.net	SOA	Simple	-	No
<input type="checkbox"/>	mail.oscar.net	A	Simple	-	No
<input type="checkbox"/>	www.oscar.net	A	Simple	-	No

Podemos verificar el funcionamiento entrando en una de las maquinas y utilizando **nslookup**. Tambien podemos utilizar ping para verificar si nos devuelve respuesta la maquina.

```

- : sudo ssh — Konsole
ubuntu@ip-172-31-83-49:~$ nslookup
> set type=A
> www.oscar.net
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.oscar.net
Address: 172.31.83.49
>

```

```

> mail.oscar.net
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   mail.oscar.net
Address: 172.31.95.199

```



```
> - : sudo ssh — Konsole
ubuntu@ip-172-31-83-49:~$ ping www.oscar.net
PING www.oscar.net (172.31.83.49) 56(84) bytes of data.
64 bytes from ip-172-31-83-49.ec2.internal (172.31.83.49):
icmp_seq=1 ttl=64 time=0.008 ms
```

EN RESUMEN

Create EC2 Instances:

Launch two EC2 instances in the default VPC:

WebServer and eMailServer.

Use t2.micro instance type and ensure both instances are in the same security group.

Configure the security group to allow SSH (port 22), HTTP (port 80), and ICMP traffic.

Create a Private Hosted Zone in Route 53:

Open Route 53 and create a private hosted zone.

Choose a domain name (e.g., oscar.net).

Associate the hosted zone with the default VPC.

Ensure the VPC settings enableDnsHostnames and enableDnsSupport are set to true.

Create DNS Records:

Create A records for the EC2 instances:

WebServer: Point to the private IP of the WebServer instance.

eMailServer: Point to the private IP of the eMailServer instance.

Use Simple Routing Policy for both records.

Verify DNS Resolution:

SSH into one of the EC2 instances.

Use nslookup or ping to verify that the DNS records resolve correctly:

nslookup www.oscar.net should return the private IP of the WebServer.

nslookup mail.oscar.net should return the private IP of the eMailServer.

Practica 2

En esta actividad vamos a crear un dominio en una red privada en AWS. Para ello vamos a configurar un servidor de nombres de dominio en una instancia EC2 y usaremos otra instancia EC2 como cliente.

Crea en la VPC predeterminada dos instancias EC2, con sistema operativo Ubuntu, claves vockey y de tipo t2.micro. Se llamarán **DNSServer** y **PC1**, respectivamente. **Ten la precaución de que ambas máquinas pertenezcan al mismo grupo de seguridad.**

Description - required [Info](#)

Practica Bind / named

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type [Info](#)

ssh

Protocol [Info](#)

TCP

Port range

22

Source type [Info](#)

Anywhere

Source [Info](#)

🔍 Add CIDR, prefix list or security g

0.0.0.0/0 ✕

Description

e.g. SSH fc

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type [Info](#)

HTTP

Protocol [Info](#)

TCP

Port range

80

Source type [Info](#)

Anywhere

Source [Info](#)

🔍 Add CIDR, prefix list or security g

0.0.0.0/0 ✕

Description

e.g. SSH fc

▼ Security group rule 3 (ICMP, All, 0.0.0.0/0)

Type [Info](#)

All ICMP - IPv4

Protocol [Info](#)

ICMP

Port range

All

Source type [Info](#)

Anywhere

Source [Info](#)

🔍 Add CIDR, prefix list or security g

0.0.0.0/0 ✕

Description

e.g. SSH fc

▼ Security group rule 4 (UDP, 53, 0.0.0.0/0)

Type [Info](#)

DNS (UDP)

Protocol [Info](#)

UDP

Port range

53

Source type [Info](#)

Anywhere

Source [Info](#)

🔍 Add CIDR, prefix list or security g

Description

e.g. SSH fc

<input type="checkbox"/>	DNSServer	i-0fb31ce7c5592bb3a	✔ Running	🔍
<input type="checkbox"/>	PC1	i-0cef3a3f7c042992b	✔ Running	🔍

Bind es el estándar de facto para servidores DNS. Es una herramienta de software libre y se distribuye con la mayoría de plataformas Unix y Linux, donde **también se le conoce con el sobrenombre de named (name daemon)**. **Bind 9 es la versión recomendada** para usarse y es la que emplearemos.

Para instalar el servicio:

```
sudo apt install bind9
```

Para instalar el servicio:

```
sudo systemctl start|stop|restart|status named
```

Nótese que para referirnos al servicio asociado al servidor DNS usamos named y no bind

El archivo principal de configuración del DNS es el archivo /etc/bind/named.conf.

Este archivo sirve simplemente para aglutinar o agrupar a los archivos de configuración que usaremos. Estos 3 includes hacen referencia a los 3 diferentes archivos donde deberemos realizar la verdadera configuración, ubicados en el mismo directorio.

En concreto:

- **named.conf.options:** permite definir opciones genéricas del servidor DNS.
- **named.conf.local:** permite definir las zonas para las que el servidor es autoritativo. Es decir, nuestras zonas de autoridad.
- **named.conf.default-zones:** configuración para zonas por defecto.

En el directorio encontraremos además otros archivos que **no deben ser modificados**, como: **db.0, db.255, db.empty, bind.keys, zones.rfc1918, etc.**

Antes de nada copiaremos los ficheros named.conf, named.conf.options y named.conf.local de configuración y los renombraremos terminando en .old (usando el comando cp). Así podremos volver atrás en caso de que rompamos algo.

Por defecto, al instalar el paquete bind está preconfigurado como servidor caché DNS. Tan solo será necesario editar el archivo /etc/bind/named.conf.options, descomentar la sección forwarders y añadir las IPs de dos servidores DNS donde redirigir las peticiones DNS.

```
(ubuntu) ec2-44-201-123-107.compute-1.amazonaws.com — Konsole
...01-123-107.compute-1.amazonaws.com x ...-205-98-91.compute-1.ama
GNU nano 7.2 named.conf.options *
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers
    // to talk to, you may need to fix the firewall to al
    // ports to talk. See http://www.kb.cert.org/vuls/id
    // If your ISP provided one or more IP addresses for
    // nameservers, you probably want to use them as forw
    // Uncomment the following block, and insert the addi
    // the all-0's placeholder.

    // forwarders {
    //      8.8.8.8, 8.8.4.4, 1.1.1.1;
    // };
    //
```

De esta manera nuestro servidor redirigirá las peticiones DNS que no sepa resolver a los servidores 8.8.8.8 y 8.8.4.4, que son los servidores DNS públicos de Google.

Configuración como servidor DNS maestro

Por razones de accesibilidad y organizativas, deseamos asignar un nombre a todos los equipos de nuestra red, para lo que instalaremos un servidor DNS privado con un dominio ficticio 'minombre.abastos.edu'.

Todos los PCs de nuestra red pertenecerán a dicho dominio ficticio que funcionará solo en nuestra red interna, no en Internet. En tal caso el nombre completo (o también FQDN, Fully Qualified Domain Name) de los PCs terminará con 'minombre.abastos.edu.', por ejemplo: 'pc1.minombre.abastos.edu.'.

Lo ideal en una situación así es disponer de un servidor DNS que sea maestro de nuestro dominio, es decir, maestro del dominio interno 'minombre.abastos.edu'.

Un servidor maestro también puede llevar a cabo la resolución inversa, es decir, que si se recibe una consulta acerca de quién es pc2.minombre.abastos.edu deberá devolver su IP, pongamos por ejemplo 172.31.43.40, pero eso quedaría fuera del ámbito de esta práctica.

Para configurar este esquema deberemos añadir en el archivo /etc/bind/named.conf.local la especificación de maestro para el dominio.

Configuraremos la resolución inversa más adelante.

El fichero `named.conf.local` quedaría así:

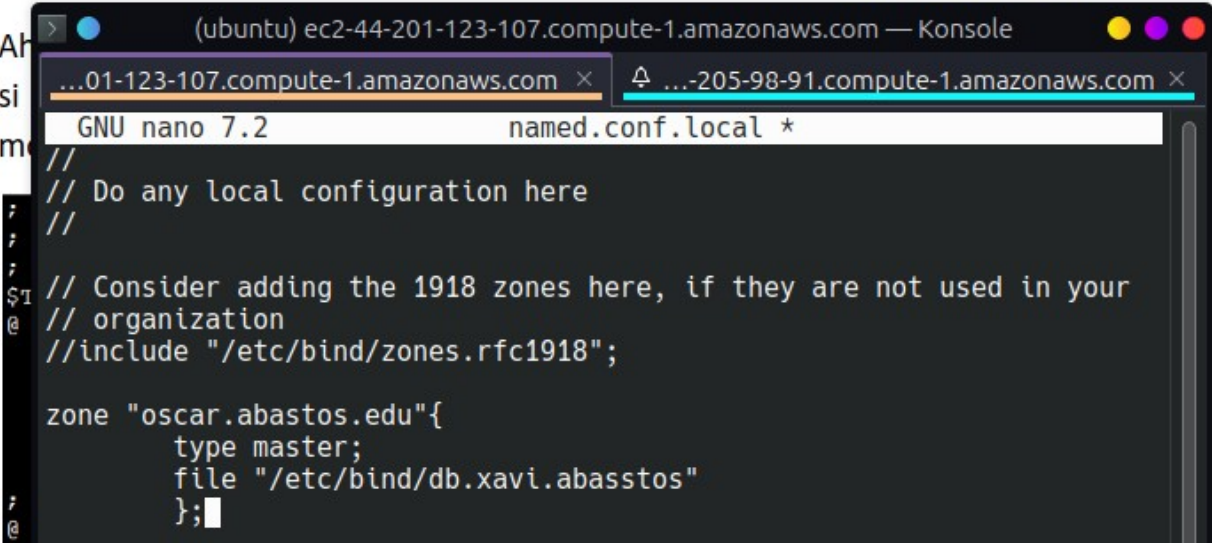
```
//  
// Do any local configuration here  
//  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
zone "xavi.abastos.edu"{  
    type master;  
    file "/etc/bind/db.xavi.abastos";  
};
```

Nombre de la zona

Tipo de servidor para la zona: master

Fichero donde añadiremos los registros de zona

Ahora si me



```
(ubuntu) ec2-44-201-123-107.compute-1.amazonaws.com — Konsole  
...01-123-107.compute-1.amazonaws.com x ...-205-98-91.compute-1.amazonaws.com x  
GNU nano 7.2 named.conf.local *  
//  
// Do any local configuration here  
//  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
  
zone "oscar.abastos.edu"{  
    type master;  
    file "/etc/bind/db.abasstos"  
};
```

Ahora será necesario crear el fichero `/etc/bind/db.minombre.abastos`. Será más fácil si partimos de una copia del fichero `db.local` (cópialo con el comando `cp`). Y lo modificaremos para incluir la información de nuestra zona. Quedaría así:

```
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      xavi.abastos.edu. admin@xavi.abastos.edu. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;

ns1        IN      NS       ns1.xavi.abastos.edu.
ns1        IN      A        172.31.35.131
pc1        IN      A        172.31.43.40
pc2        IN      A        172.31.43.41
```

FQDN de la zona, incluyendo "." final

Registro SOA

Mail del responsable de zona

Identifica al equipo ns.xavi.abastos.edu como servidor de nombres de la zona

Registros A, que asocian una IP a un nombre de equipo

(ubuntu) ec2-44-201-123-107.compute-1.amazonaws.com — Konsole

... ec2-44-201-123-107.compute-1.amazonaws.com × ... ec2-54-205-98-91.compute-1.amazonaws.com ×

GNU nano 7.2 db.oscar.abastos.edu *

```
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      oscar.edu.abastos. admin@oscar.abastos.edu. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;

@         IN      NS       ns1.oscar.abastos.edu.
ns1        IN      A        172.31.89.34
pc1        IN      A        172.31.81.100
```

Ten en cuenta que:

- La IP 172.31.35.131 es la IP privada de la instancia que hemos llamado DNS_Server.
- La IP 172.31.43.40 es la IP privada de la instancia que hemos llamado PC1.
- La IP 172.31.43.41 no corresponde con ningún equipo de nuestra red, pero se pone para futuras ampliaciones.

Una vez modificados y guardados, comprobamos que la configuración es correcta usando dos comandos:

- **named-checkconf**: si la configuración es correcta no devuelve nada
- **named-checkzone** nombre_zona fichero_zona. Por ejemplo:

```
named-checkconf
named-checkzone oscar.abastos.edu /etc/bind/db.oscar.abastos.edu
```



```
(ubuntu) ec2-44-201-123-107.compute-1.amazonaws.com — Konsole
... ec2-44-201-123-107.compute-1.amazonaws.com × ... ec2-54-205-98-91.compute-1.amazonaws.com ×
ubuntu@ip-172-31-89-34:/etc/bind$ named-checkzone oscar.abastos.edu /etc/bind/d
b.oscar.abastos.edu
zone oscar.abastos.edu/IN: loaded serial 2
OK
ubuntu@ip-172-31-89-34:/etc/bind$ named-checkconf
ubuntu@ip-172-31-89-34:/etc/bind$
```

Ya que esta todo correcto, reiniciamos el servicio de named.

Comprobación en local del funcionamiento del servidor DNS

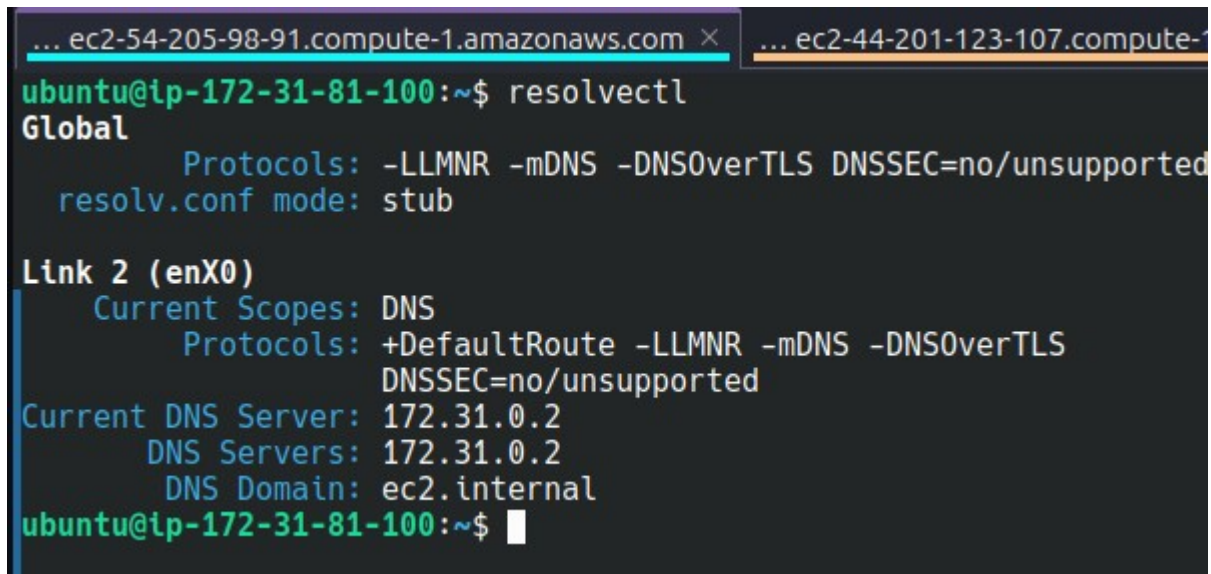
Podemos comprobar que el servidor DNS funciona correctamente desde la propia máquina que actúa como servidor usando nslookup y especificando que use el servidor DNS que está en la IP 127.0.0.1, que es la de localhost.

```
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
> pc1.oscar.abastos.edu
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:   pc1.oscar.abastos.edu
Address: 172.31.81.100
>
```

CONFIGURACIÓN DEL PC1 COMO CLIENTE DNS

En este paso configuraremos la máquina PC1 para que use la máquina que acabamos de configurar como servidor DNS. Primero comprobaremos el servidor DNS actual con **resolvectl**

A terminal window with a dark background and light-colored text. At the top, there are two browser tabs: "... ec2-54-205-98-91.compute-1.amazonaws.com" and "... ec2-44-201-123-107.compute-1.amazonaws.com". The terminal prompt is "ubuntu@ip-172-31-81-100:~\$". The user has entered the command "resolvectl". The output shows the global DNS configuration: "Global", "Protocols: -LLMNR -mDNS -DNSoverTLS DNSSEC=no/unsupported", and "resolv.conf mode: stub". Then, the user enters "Link 2 (enX0)". The output shows the current scopes and protocols for the interface: "Current Scopes: DNS", "Protocols: +DefaultRoute -LLMNR -mDNS -DNSoverTLS", and "DNSSEC=no/unsupported". Finally, the user enters "Current DNS Server: 172.31.0.2". The output shows the current DNS server and domain: "Current DNS Server: 172.31.0.2", "DNS Servers: 172.31.0.2", and "DNS Domain: ec2.internal". The terminal prompt is now "ubuntu@ip-172-31-81-100:~\$".

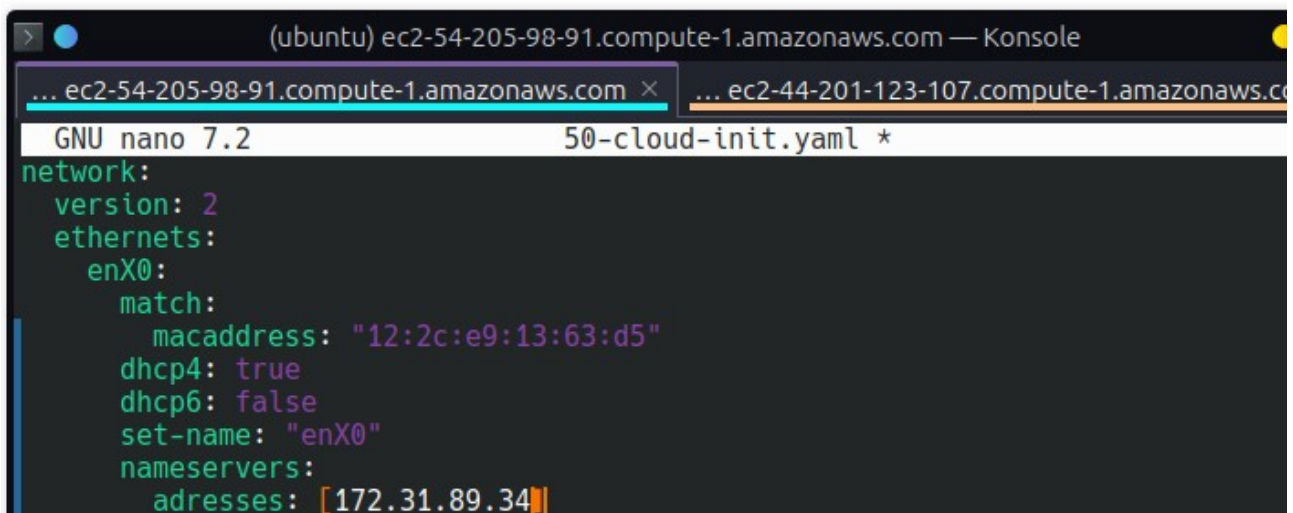
```
... ec2-54-205-98-91.compute-1.amazonaws.com × ... ec2-44-201-123-107.compute-1.amazonaws.com
ubuntu@ip-172-31-81-100:~$ resolvectl
Global
    Protocols: -LLMNR -mDNS -DNSoverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 2 (enX0)
    Current Scopes: DNS
    Protocols: +DefaultRoute -LLMNR -mDNS -DNSoverTLS
               DNSSEC=no/unsupported
Current DNS Server: 172.31.0.2
    DNS Servers: 172.31.0.2
    DNS Domain: ec2.internal
ubuntu@ip-172-31-81-100:~$
```

El actual servidor de nombres está en la IP 172.31.0.2, que es el DNS por defecto de la VPC de AWS. **Para modificarlo debemos cambiar la configuración de red.** Dicha configuración la maneja un programa llamado **Netplan**, que almacena su configuración en el fichero **/etc/netplan/50-cloud-init.yaml**. Lo primero que haremos será crear una copia de dicho archivo por si acaso:

A continuación, editamos el fichero, de manera que incluya el grupo nameservers (ojo, se indenta con espacios):

```
network:
  version: 2
  ethernets:
    enX0:
      match:
        macaddress: "0e:f1:65:c7:3e:03"
      dhcp4: true
      dhcp6: false
      set-name: "enX0"
      nameservers:
        addresses: [172.31.35.131]
```



The screenshot shows a terminal window titled "(ubuntu) ec2-54-205-98-91.compute-1.amazonaws.com — Konsole". The terminal is running the GNU nano 7.2 editor, editing the file 50-cloud-init.yaml. The configuration shown is for a network interface named enX0, with a macaddress of "12:2c:e9:13:63:d5", dhcp4 set to true, dhcp6 set to false, and set-name set to "enX0". The nameservers section is being edited, with the current address being 172.31.89.34.

```
(ubuntu) ec2-54-205-98-91.compute-1.amazonaws.com — Konsole
... ec2-54-205-98-91.compute-1.amazonaws.com x ... ec2-44-201-123-107.compute-1.amazonaws.com
GNU nano 7.2 50-cloud-init.yaml *
network:
  version: 2
  ethernets:
    enX0:
      match:
        macaddress: "12:2c:e9:13:63:d5"
      dhcp4: true
      dhcp6: false
      set-name: "enX0"
      nameservers:
        addresses: [172.31.89.34]
```

Por último, actualizamos la configuración y comprobamos:

```
$ sudo netplan apply
$ resolvectl
```

```
ubuntu@ip-172-31-81-100:/etc/netplan$ sudo netplan apply
ubuntu@ip-172-31-81-100:/etc/netplan$ resolvectl
Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 2 (enX0)
    Current Scopes: DNS
    Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS
               DNSSEC=no/unsupported
Current DNS Server: 172.31.89.34
    DNS Servers: 172.31.89.34 172.31.0.2
    DNS Domain: ec2.internal
ubuntu@ip-172-31-81-100:/etc/netplan$
```

hora debemos modificar el grupo de seguridad para que permita el tráfico de entrada:

- DNS(UDP) desde Anywhere.
- ICMP IPv4 desde Anywhere.

DNS(TCP) no será necesario puesto que no tenemos servidor secundario con lo que no habrá transferencias de zona.

Si todo es correcto podremos hacer un ping al servidor de nombres ns1.oscar.abastos.edu

```
... ec2-54-205-98-91.compute-1.amazonaws.com x ... ec2-44-201-123-107.compute-1.ar
ubuntu@ip-172-31-81-100:/etc/netplan$ ping ns1.oscar.abastos.edu
PING ns1.oscar.abastos.edu (172.31.89.34) 56(84) bytes of data.
64 bytes from 172.31.89.34: icmp_seq=1 ttl=64 time=0.848 ms
64 bytes from 172.31.89.34: icmp_seq=2 ttl=64 time=0.821 ms
```

MODIFICACIONES EN EL DOMINIO

Ahora queremos añadir un alias para que el mismo equipo que actúa como servidor de nombres sea el que aloje una web corporativa. Para ello queremos que responda al nombre www.xavi.abastos.edu. También queremos añadir un servidor de correo en esa máquina. Debemos modificar el archivo db.oscar.abastos del servidor de nombres para añadir un registro MX (para el servidor de correo) y un registro CNAME (para el alias). De esta manera, el fichero quedará así:

```
... ec2-44-201-123-107.compute-1.amazonaws.com x ... ec2-54-205-98-91.compute-1.ama
GNU nano 7.2 db.oscar.abastos.edu *
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     oscar.abastos.edu. admin@oscar.abastos.edu. (
                                2           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS      ns1.oscar.abastos.edu.
@         IN      MX      10      mail.oscar.abastos.edu.
ns1       IN      A       172.31.89.34
pc1       IN      A       172.31.81.100
pc2       IN      A       172.31.81.99
www       IN      CNAME    ns1
mail.oscar.abastos.edu. IN      A       172.31.81.34
```

Comprobamos la configuración y reiniciamos el servidor.

Podemos comprobar que los cambios se han realizado haciendo ping desde PC1 a la dirección www.oscar.abastos.edu y a la dirección mail.oscar.abastos.edu

... ec2-44-201-123-107.compute-1.amazonaws.com × ... ec2-54-205-98-91.compute-1.ama

```
ubuntu@ip-172-31-81-100:/etc/netplan$ ping www.oscar.abastos.edu
PING ns1.oscar.abastos.edu (172.31.89.34) 56(84) bytes of data.
64 bytes from 172.31.89.34: icmp_seq=1 ttl=64 time=0.445 ms
64 bytes from 172.31.89.34: icmp_seq=2 ttl=64 time=1.75 ms
^C
--- ns1.oscar.abastos.edu ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
```

EN RESUMEN

Steps to Configure a DNS Server with BIND on AWS EC2

1. Create the Scenario:

- Launch two Ubuntu EC2 instances in the default VPC: `DNSServer` and `PC1`.
- Ensure both instances are in the same security group.

2. Install BIND DNS Server:

- On `DNSServer`, install BIND:

```
sudo apt install bind9
```
- Manage the service with:

```
sudo systemctl start|stop|restart|status named
```

3. Configure DNS Server:

- Backup configuration files (`named.conf`, `named.conf.options`, `named.conf.local`) by renaming them to `.old`.
- Edit `/etc/bind/named.conf.options` to set up DNS forwarding to Google's DNS servers (8.8.8.8, 8.8.4.4).

4. Set Up DNS Master Server:

- Edit `/etc/bind/named.conf.local` to define the master zone for your domain (e.g., `oscar.abastos.edu`).
- Create a zone file (`/etc/bind/db.oscar.abastos`) by copying `db.local` and adding DNS records (A, MX, CNAME, etc.).
- Validate the configuration with:

```
named-checkconf  
named-checkzone oscar.abastos.edu /etc/bind/db.oscar.abastos.edu
```
- Restart the DNS service:

```
sudo systemctl restart named
```

5. Test DNS Locally:

- Use `nslookup` on `DNSServer` to verify DNS resolution:

```
nslookup
> server 127.0.0.1
> pc1.oscar.abastos.edu
```

6. Configure PC1 as DNS Client:

- On `PC1`, edit `/etc/netplan/50-cloud-init.yaml` to set `DNSServer`'s private IP as the DNS server.

- Apply the changes:

```
sudo netplan apply
```

- Verify with `resolvectl`.

7. Modify Security Group:

- Allow DNS (UDP) and ICMP traffic from anywhere in the security group.

8. Add DNS Records:

- Add a `CNAME` record for `www.oscar.abastos.edu` and an `MX` record for a mail server in the zone file (`db.oscar.abastos`).

- Validate and restart the DNS service:

```
named-checkzone oscar.abastos.edu /etc/bind/db.oscar.abastos
sudo systemctl restart named
```

9. Test from PC1:

- Use `ping` or `nslookup` on `PC1` to verify the new DNS records (e.g., `www.oscar.abastos.edu`).

This setup configures a private DNS server on AWS, allowing internal domain resolution and testing from a client machine.