Proyecto: Implementación de un DNS basado en Route 53

1. Introducción a DNS y Route 53

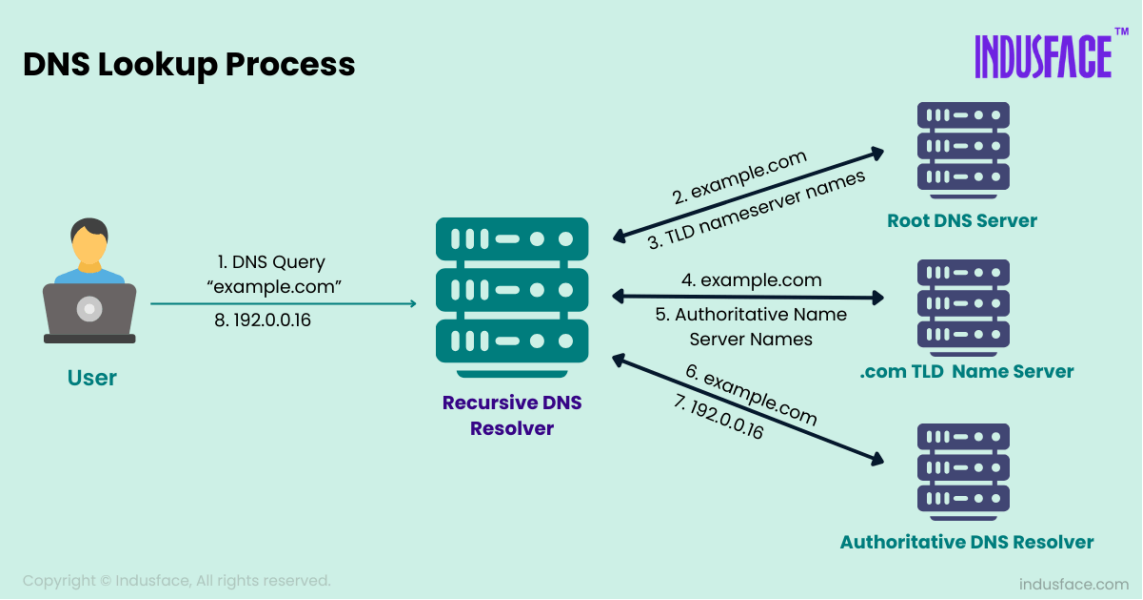
DNS (Domain Name System) es un servicio ampliamente probado con múltiples implementaciones (Microsoft, Google, Easy DNS, etc.).

Se puede crear un DNS propio o utilizar soluciones existentes.

DNS utiliza una estructura jerárquica:

- Raíz: com
- Nivel secundario: google, nacion
- Hojas: hosts

Cada identificador en DNS puede tener hasta 256 caracteres. Los dominios cortos son más costosos porque son más fáciles de recordar.



Referencia imágen: https://www.indusface.com/learning/what-is-dns/

Route 53 (AWS)

Route 53 es el servicio de DNS de Amazon Web Services (AWS).

No es exclusivo de Amazon, sino que es parte de los estándares de Internet.

Route 53 usa hosted zones, que son dominios registrados y administrados en AWS.

Hosted zones tienen múltiples servidores para redundancia y disponibilidad.

Tipos de registros DNS

- A (Address): Mapea un nombre de dominio a una dirección IP.
- Alias: Mapea un nombre a otro nombre.
- Se pueden asignar múltiples IPs a un mismo nombre de dominio, realizando un round-robin para balanceo de carga.

Routing Policy en Route 53

- Simple Routing: Devuelve una única IP.
- Weighted Routing: Asigna pesos a diferentes IPs (ejemplo: IP1 recibe 80% del tráfico, IP2 el 20%).
- Geolocation Routing: Responde con el servidor más cercano según la ubicación del usuario.
- Latency-based Routing: Asigna la respuesta según la menor latencia.
- Failover Routing: Redirige a un servidor alternativo en caso de falla.

2. Implementación del Proyecto

Capas de red involucradas

- En Cloud Providers, se trabaja con capa 7 (Aplicación), capa 4 (Transporte) y capa 3 (Red).
- En capa 4 se usan UDP y TCP:
    - UDP: Usado en aplicaciones de tiempo real como streaming, no garantiza entrega de datos.
    - TCP: Protocolo confiable que asegura la entrega de la información.

Protocolo DNS

- Utiliza los puertos UDP 53 y TCP 53.
- RFC 2929 define los componentes del protocolo DNS.
- Se usa QR bit para definir el tipo de consulta.
- El protocolo DNS no es visual, utiliza caracteres ASCII y shifts de bits para organizar la información.
- nslookup es un cliente DNS que resuelve nombres de dominio.

Intercepción de consultas DNS

- Se desarrollará un interceptor DNS que:
    - Captura las consultas DNS.
    - Identifica si es un query estándar (bit QR en 0).

    - Busca si el dominio tiene un registro definido en la base de datos.
    - Si existe, responde con la información almacenada.
    - Si no existe, reenvía la consulta a un DNS real.

Manejo de datos en la base de datos

- Se trabajará con Firebase para almacenar registros DNS.
- Claves se guardarán en formato invertido (ejemplo: `www.google.com` → `com/google/www`).

Se almacenarán registros simples con formato:
{
  "type": "A",
  "ip": "8.8.8.8"

}

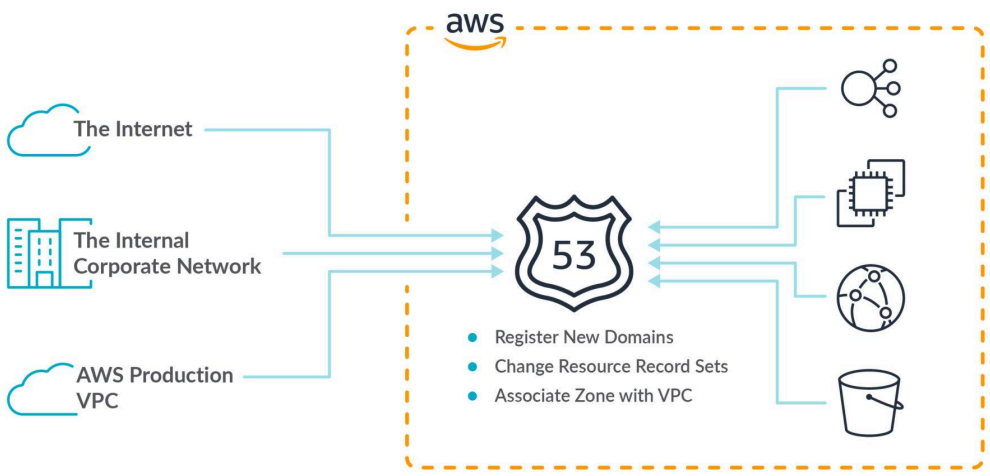
- Se implementará un sistema de round-robin en la base de datos para balanceo de carga.

Optimización de búsquedas

- Se utilizará una base de datos geolocalizada para mejorar la respuesta según la ubicación del usuario.
- En lugar de CSV, se utilizará MMDB (MaxMind Database) para geolocalización.
- Conversión MMDB → CSV mediante mmdb converter.

Comunicación entre módulos

1. Receptor DNS: Recibe consultas en el puerto UDP 53.
2. DNS API: Se comunica con Firebase a través de HTTPS.
3. Interceptor: Decide si responde con información almacenada o reenvía la consulta.
4. Cliente UDP interno: Se conecta con DNS públicos si el registro no existe en Firebase.}



Referencia Imagen: https://sysdig.com/blog/how-to-secure-aws-route-53-with-sysdig/

3. Validación y Pruebas

Análisis de paquetes DNS

- Se analizarán paquetes en crudo con hexdump y hexyl.
- Identificación de bytes en el paquete y decodificación con la tabla ASCII.

Health Checks y Alta Disponibilidad

- Implementación de health checks para verificar la disponibilidad de los servidores DNS.
- Uso de pruebas HTTP y TCP para determinar si un servidor está activo.
- Marcar servidores como healthy o unhealthy según los códigos de error HTTP (400+, 500+).
- Simulación de random delay para pruebas de balanceo de carga.

Herramientas utilizadas

- libcurl: Para realizar pruebas HTTP desde C.
- Troposphere y CDK: Para automatización en la nube.