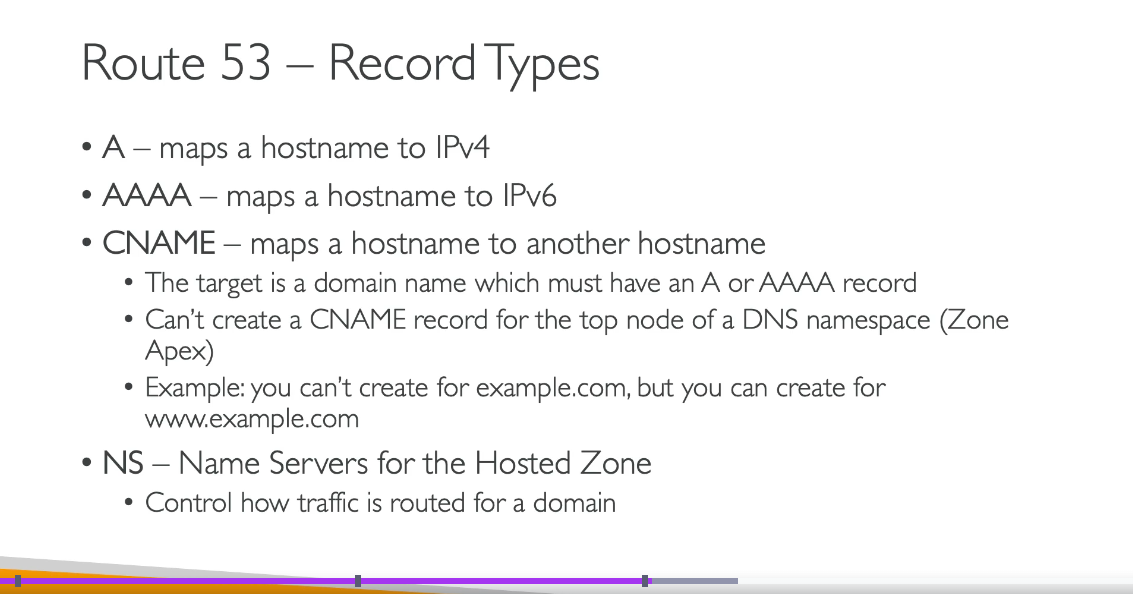
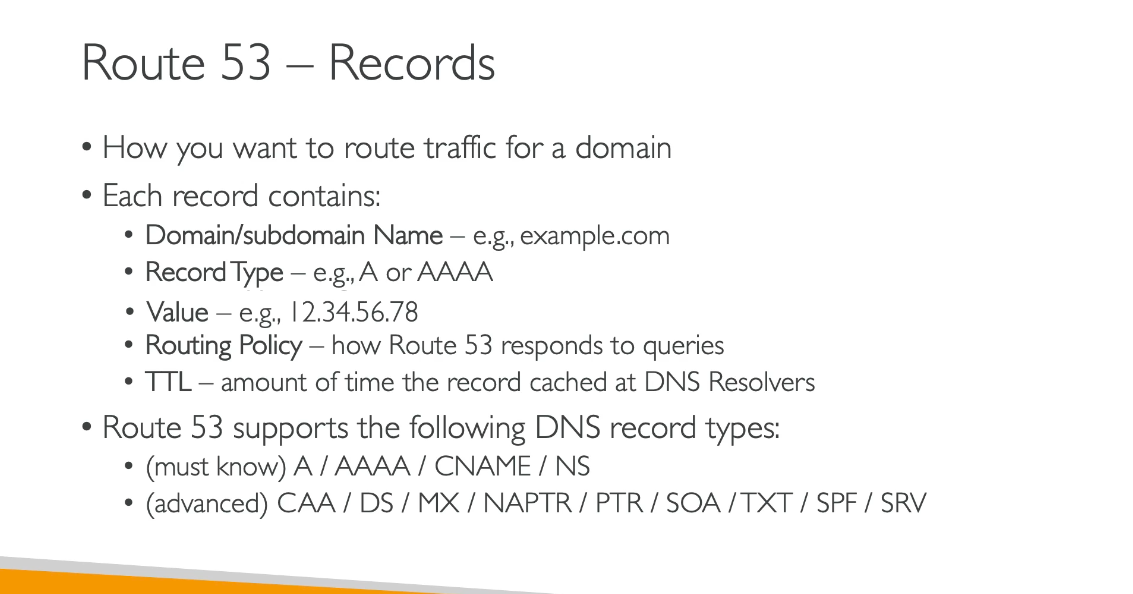
Route 53 Health Checks:

* DNS-level monitoring - Checks endpoint availability before routing traffic
* Global scope - Can monitor endpoints across multiple Regions and external resources
* Flexible targets - Can check endpoints, other health checks, CloudWatch alarms
* DNS failover - Routes traffic away from unhealthy resources at DNS resolution level
* Configurable intervals - 10 or 30 seconds (standard), or fast interval (10 seconds)
* Use case: Multi-Region failover, monitoring external endpoints, DNS-based routing decisions

Load Balancer Health Checks:

* Load balancer-level monitoring - Checks health of registered targets (EC2 instances, containers, IPs)
* Regional scope - Monitors targets within the load balancer's Region
* Target-specific - Only checks resources registered with that load balancer
* Traffic distribution - Routes traffic only to healthy targets within the load balancer
* Configurable parameters - Interval, timeout, healthy/unhealthy thresholds, path
* Use case: Distributing traffic among healthy instances behind a load balancer



|  |  |  |
| --- | --- | --- |
| **Record** | **Points To** | **Typical Use** |
| **A** | IPv4 address | Stable IP targets |
| **AAAA** | IPv6 address | IPv6 / dual-stack |
| **CNAME** | Hostname | Cloud/CDN/services |
| **NS** | Name servers | DNS ownership/delegation |
| **SOA** | Start of Authority | Provides information about the domain and the corresponding Route 53 hosted zone |

**Domain registration and SSL/TLS certificates serve different purposes, which is why both need renewal:**

**Domain Registration (Route 53)**

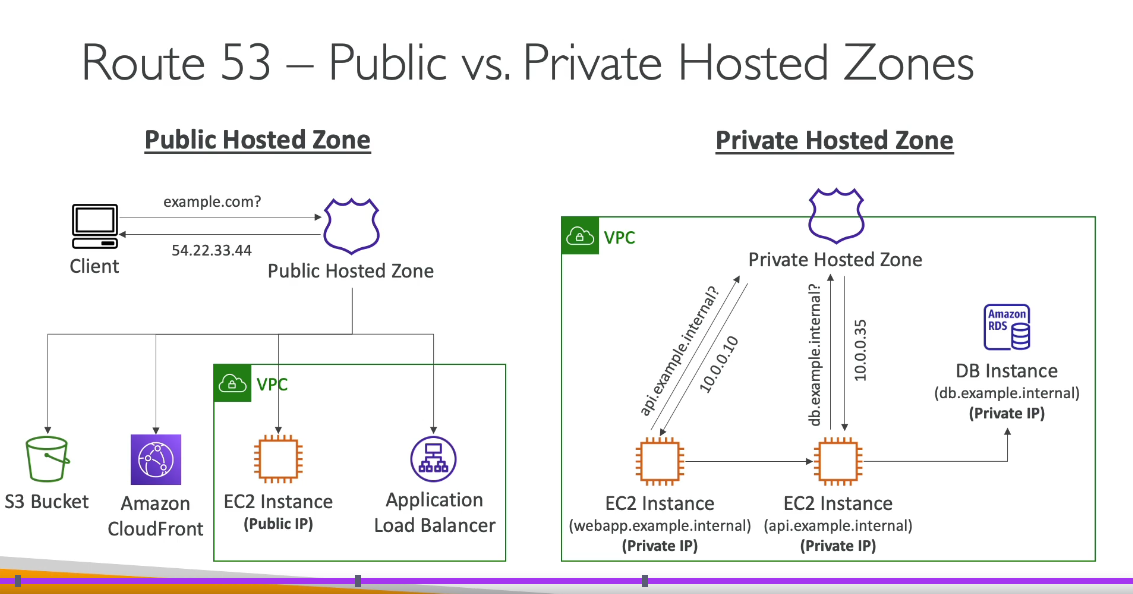
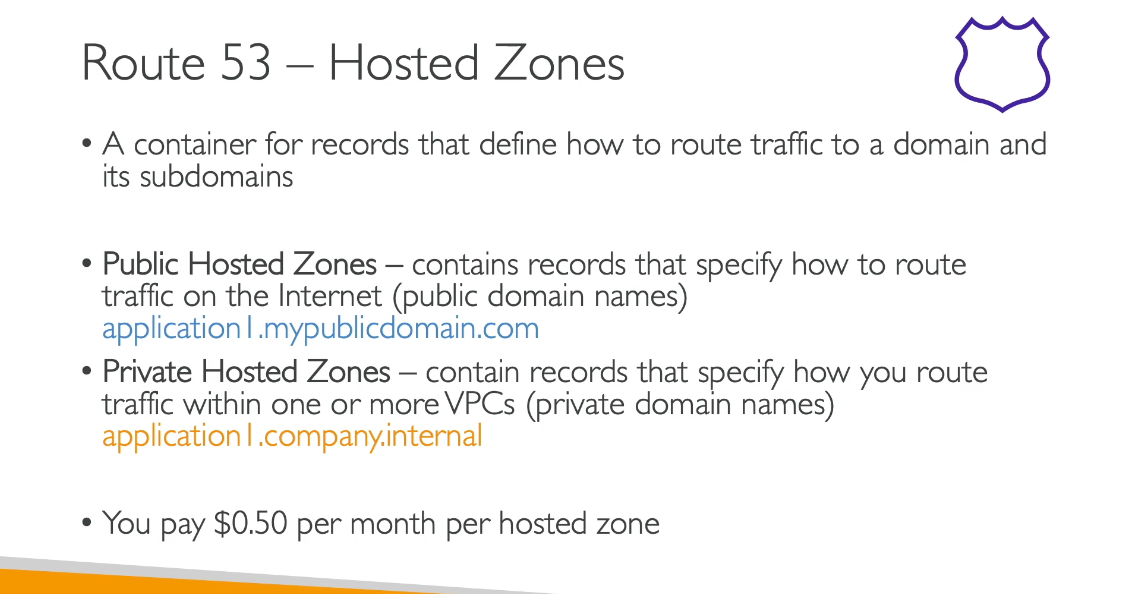
* Maintains your ownership and rights to use the domain name
* Auto-renewal ensures you don't lose the domain name itself

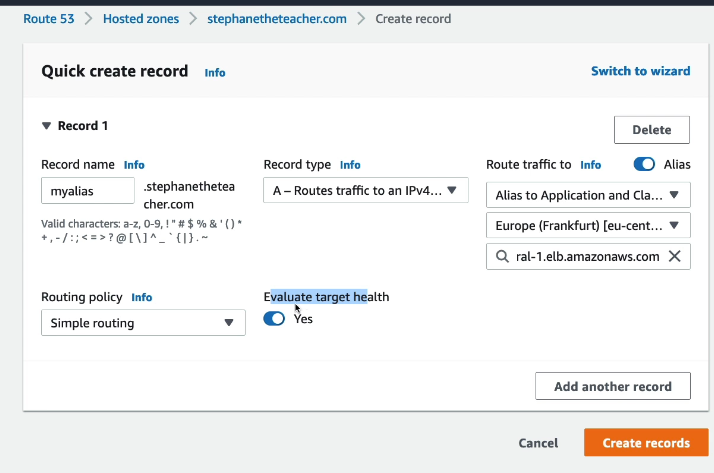
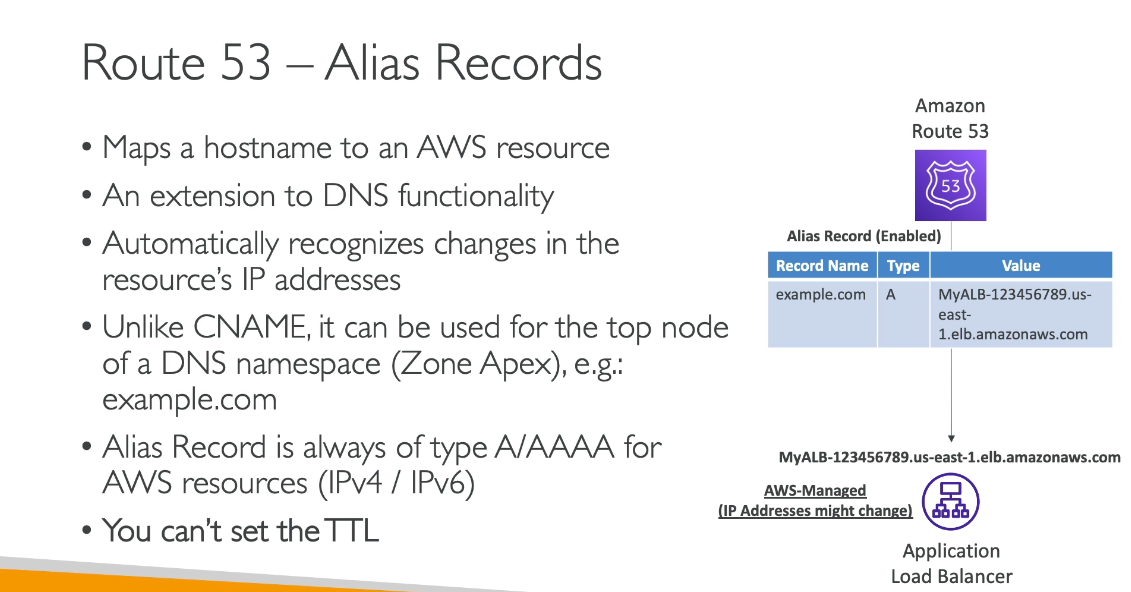
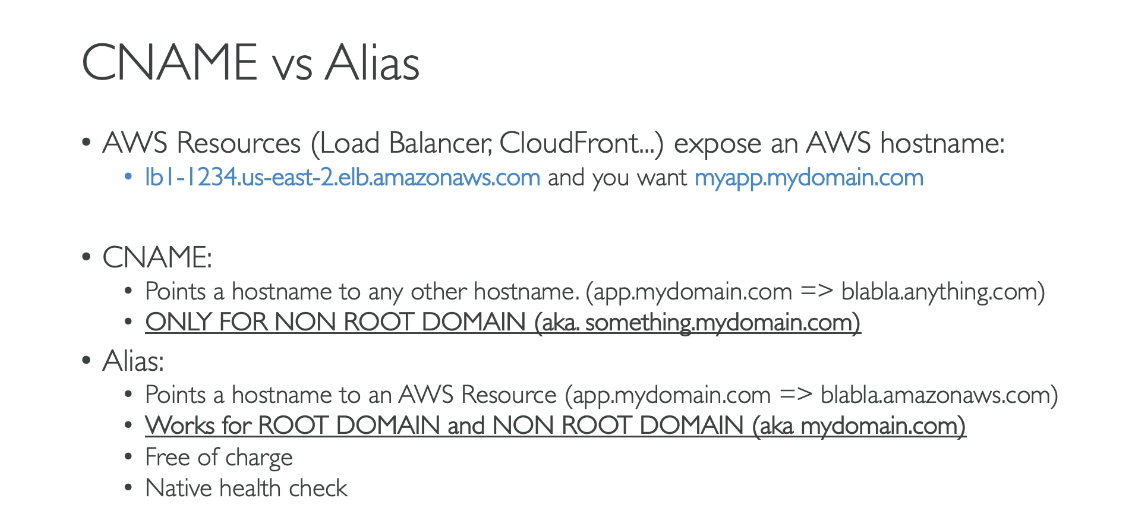
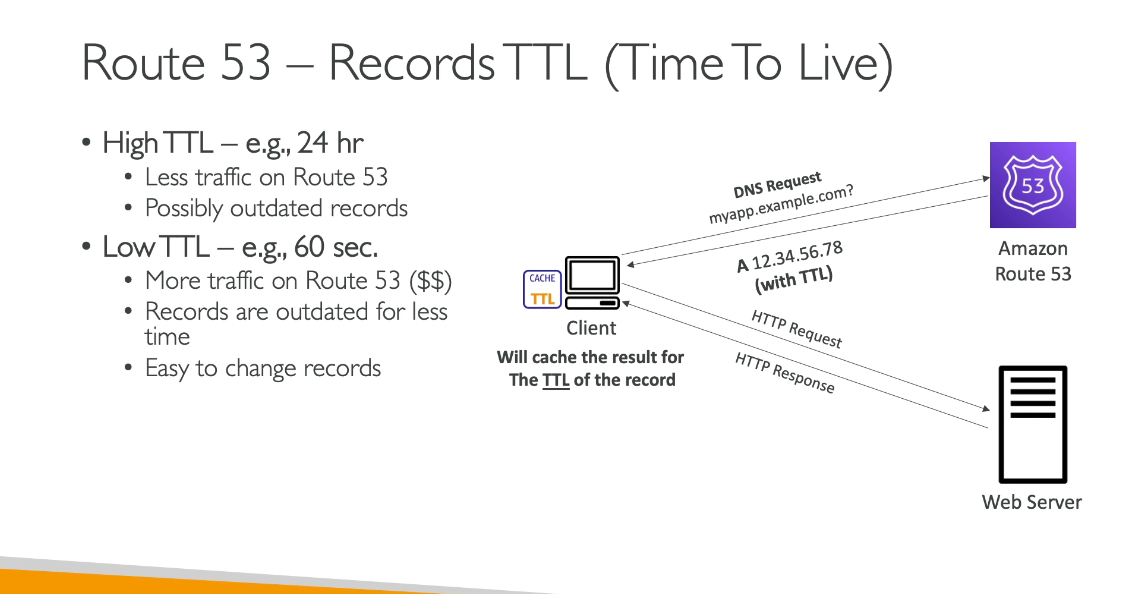
**SSL/TLS Certificates (AWS Certificate Manager)**

* Secure the connection between users and your website/application
* Verify your identity and encrypt data in transit
* Must be validated to prove you control the domain

Why Both Are Needed: Even if your domain is auto-renewed, SSL/TLS certificates expire independently (typically after 13 months) for security reasons. However, AWS Certificate Manager (ACM) makes this easier:

* DNS Validation: When you use DNS validation with ACM, certificates can be automatically renewed before expiration as long as the DNS CNAME record remains in place
* Email Validation: Requires manual approval for each renewal





**Costs for records:**

Alias records mapped to certain AWS resources are free, but other record types do incur charges. Here's the breakdown:

Free DNS Queries:

* Alias A/AAAA records mapped to these AWS services are NOT charged:
  + Elastic Load Balancers
  + Amazon CloudFront distributions
  + AWS Elastic Beanstalk environments
  + Amazon API Gateways
  + Amazon VPC endpoints
  + Amazon S3 website buckets
  + AWS AppRunner
  + Amazon AppSync
  + Amazon OpenSearch
  + Amazon Lightsail
  + Amazon Global Accelerator

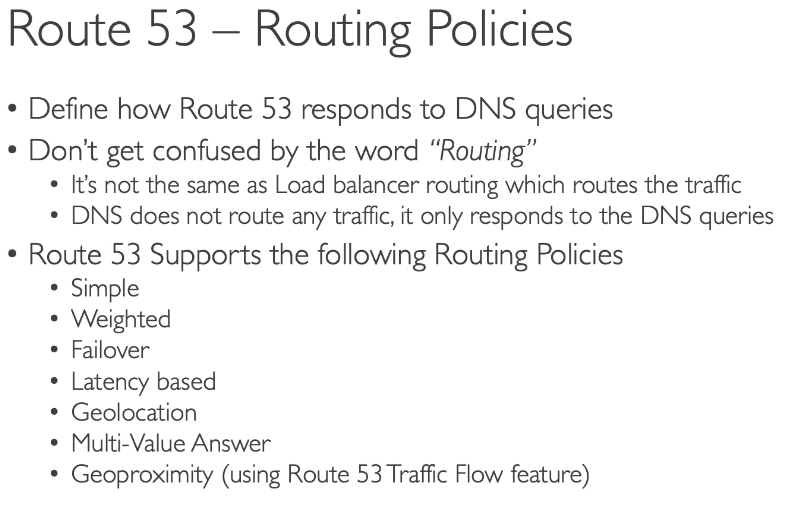
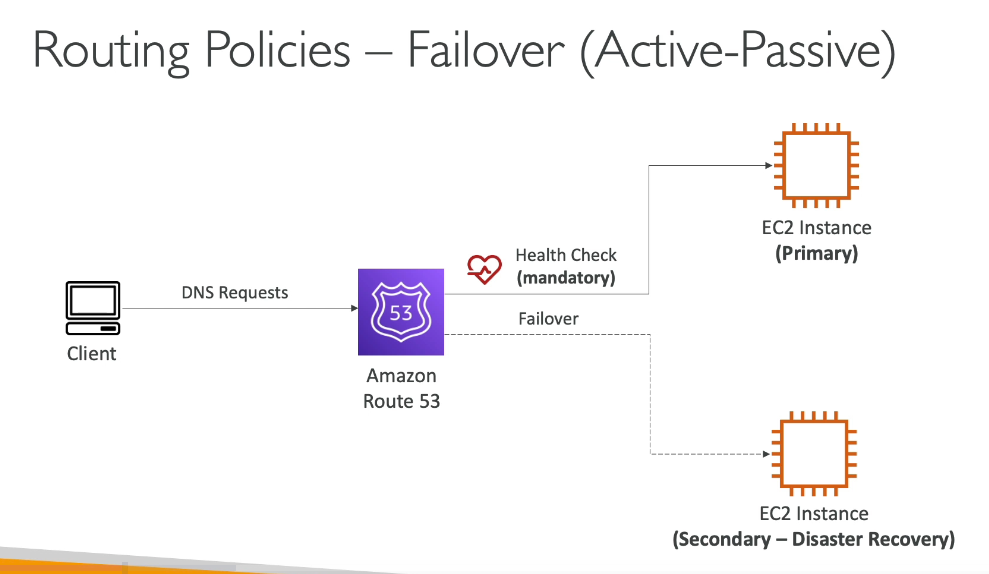
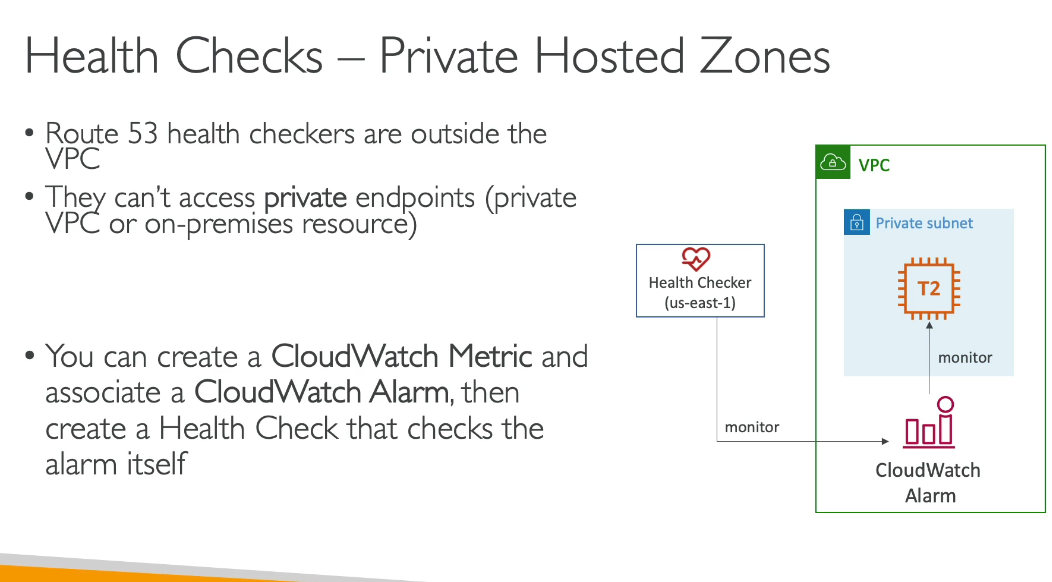
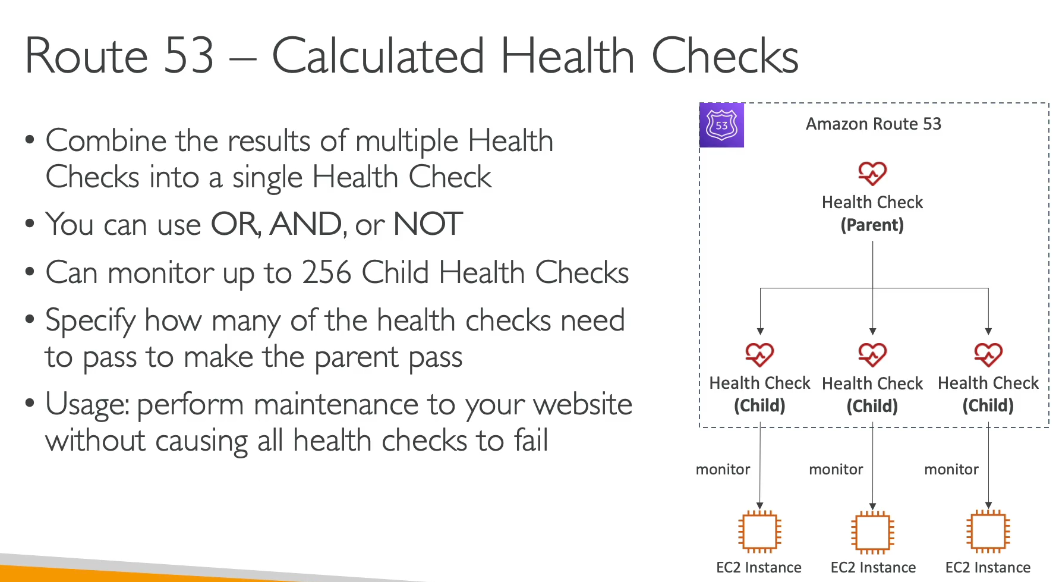
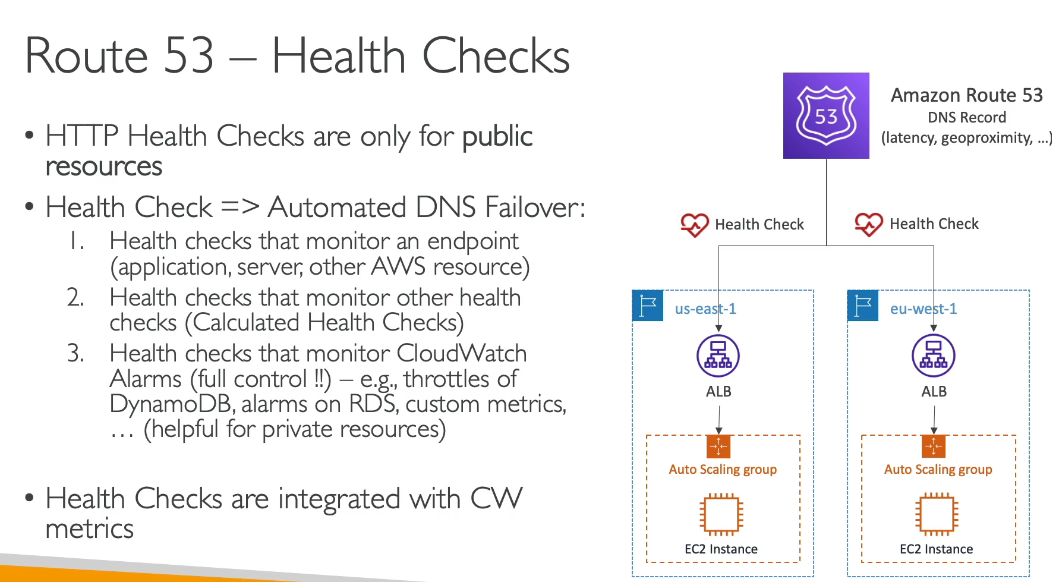
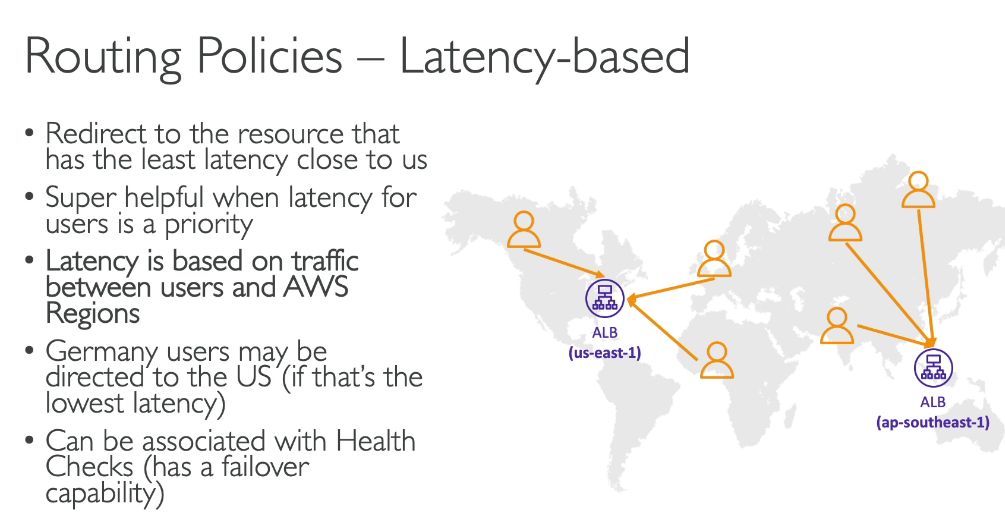
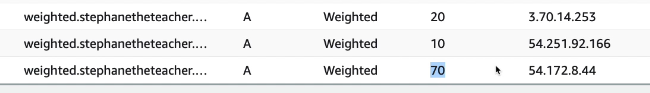
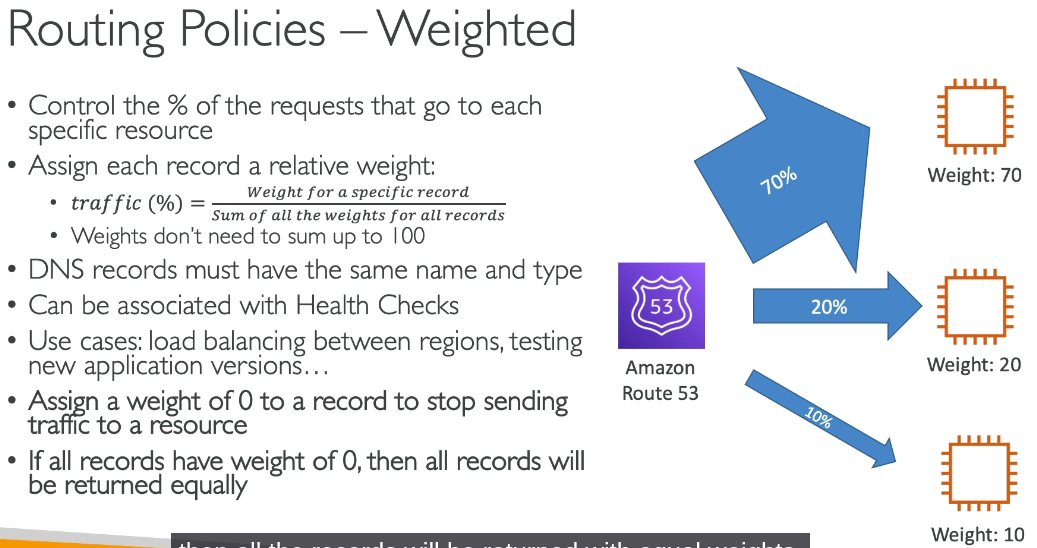
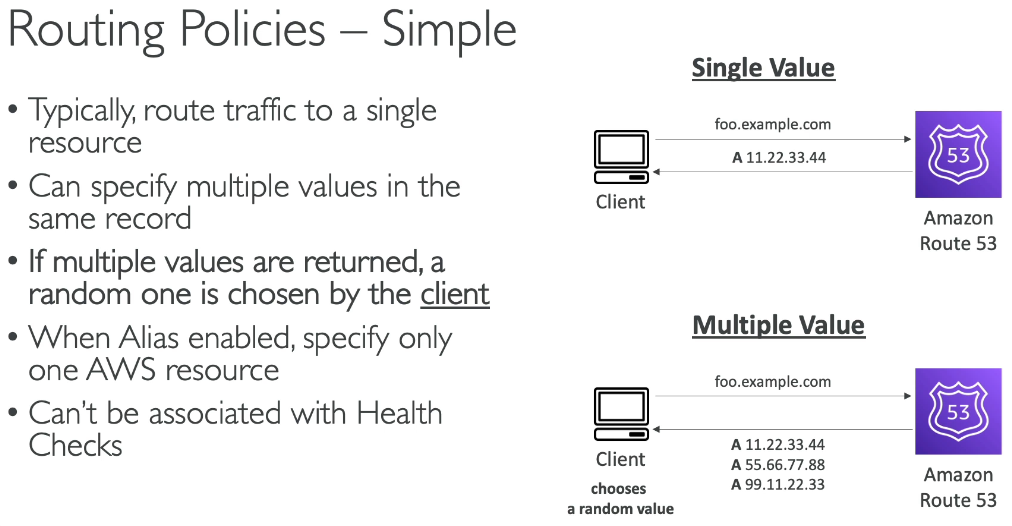
Charged DNS Queries:

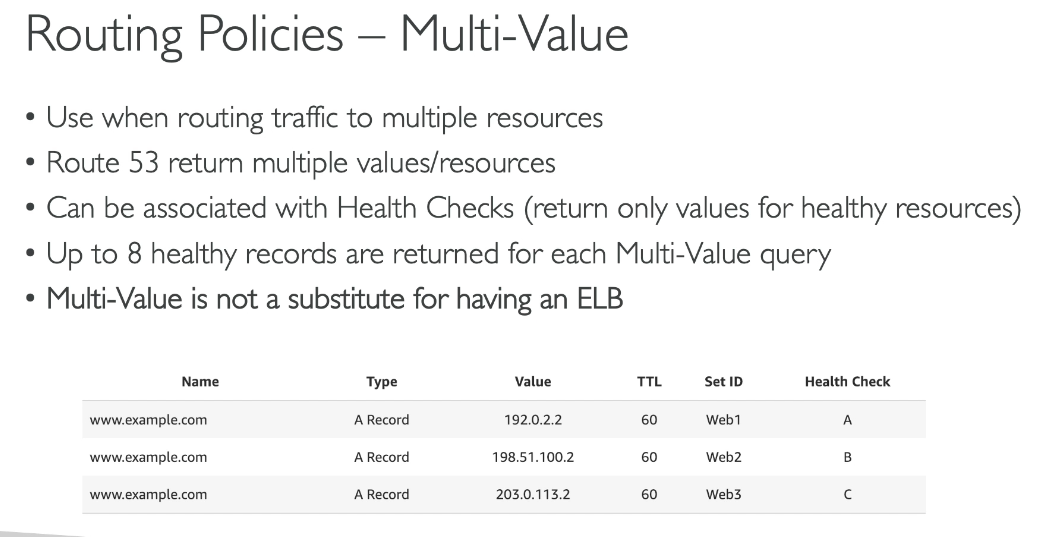
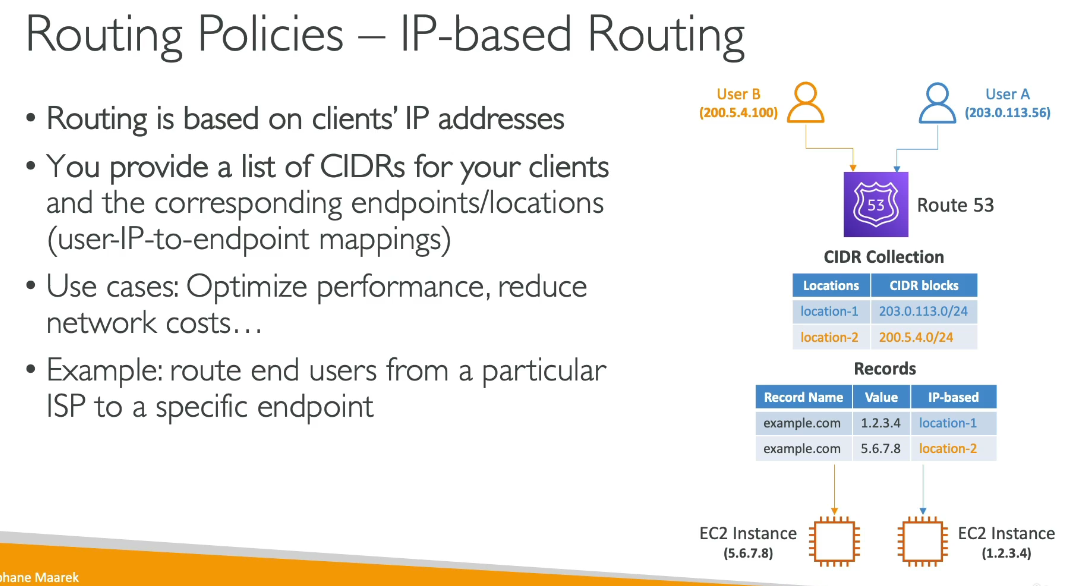
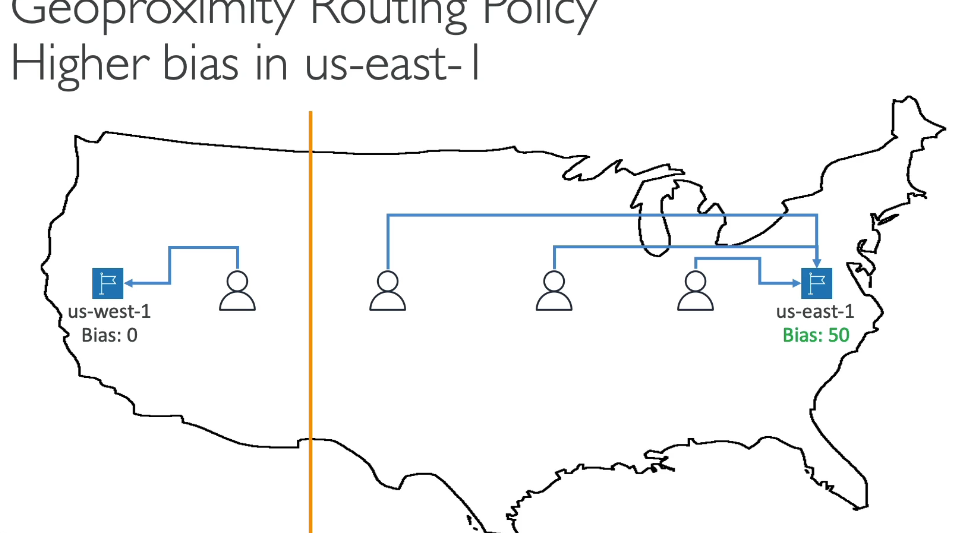
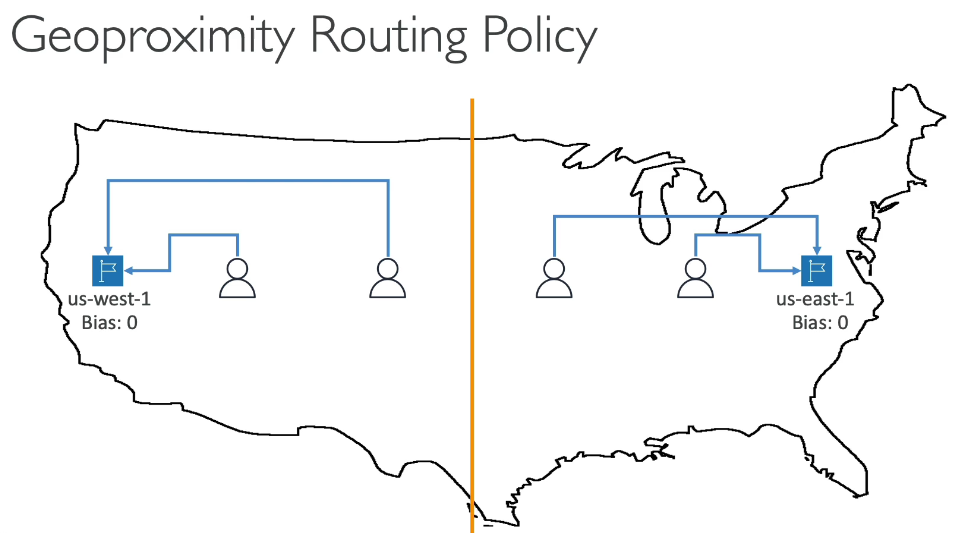
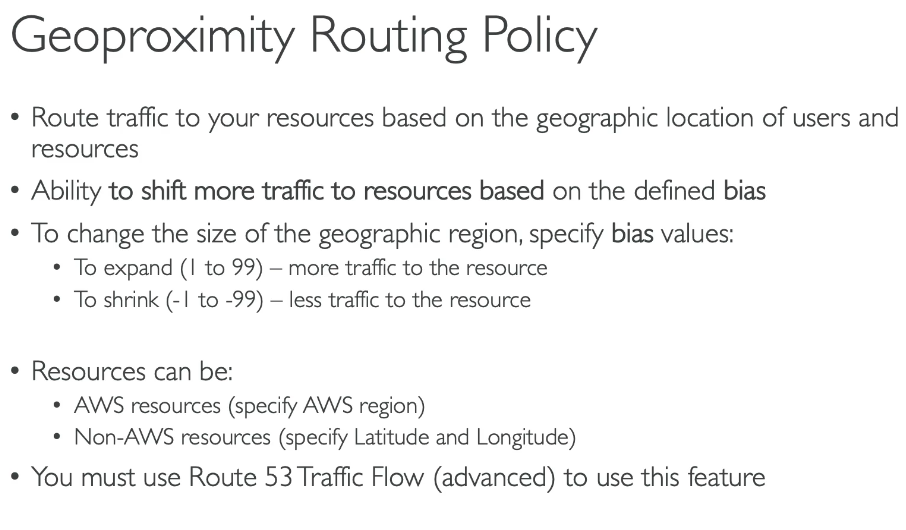
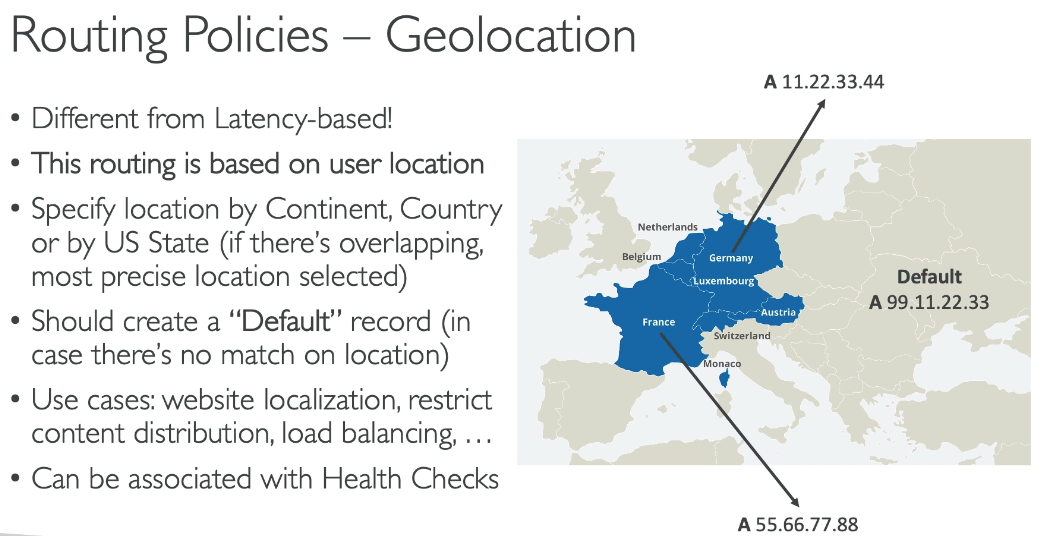
* Standard queries (A, AAAA, and other record types): $0.40 per million queries for the first 1 billion queries/month
* CNAME records: You are charged for CNAME queries. Additionally, if a CNAME redirects to another Route 53 record, you're charged for two queries - one for the CNAME response and another for resolving the target record
* Latency-based routing queries: $0.60 per million queries for the first 1 billion queries/month

Note: Route 53 does not charge for queries on private hosted zones.

The key advantage of Alias records is that they provide CNAME-like functionality for AWS resources without the query charges, making them cost-effective for AWS-to-AWS routing.

**Routing Policies:**

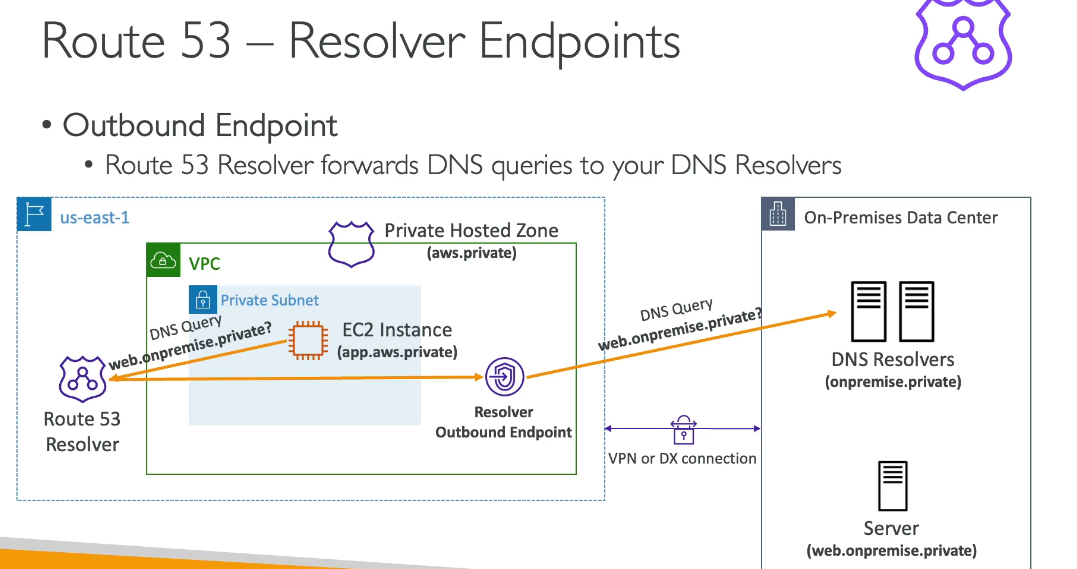
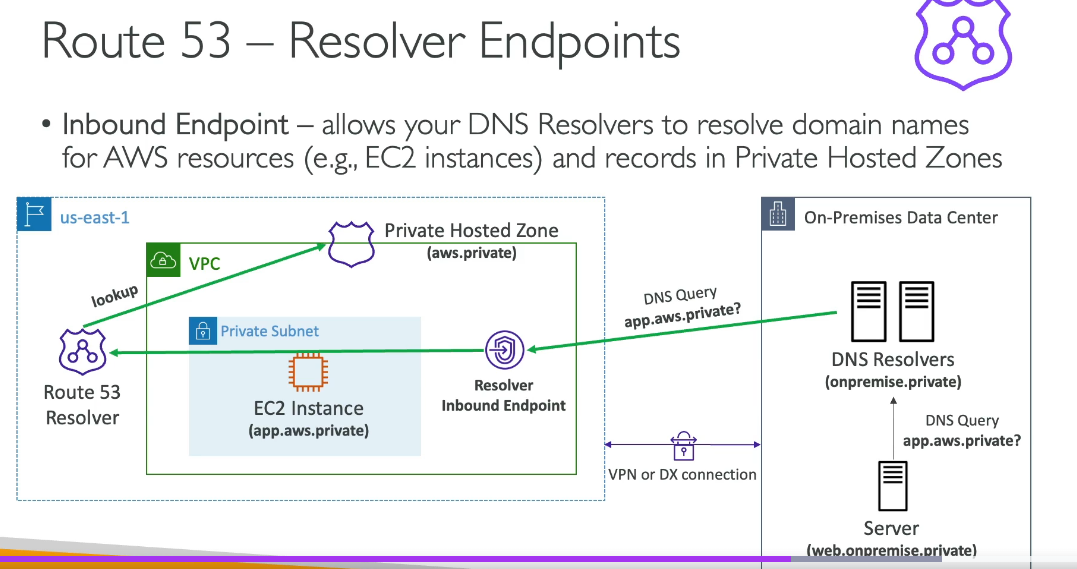
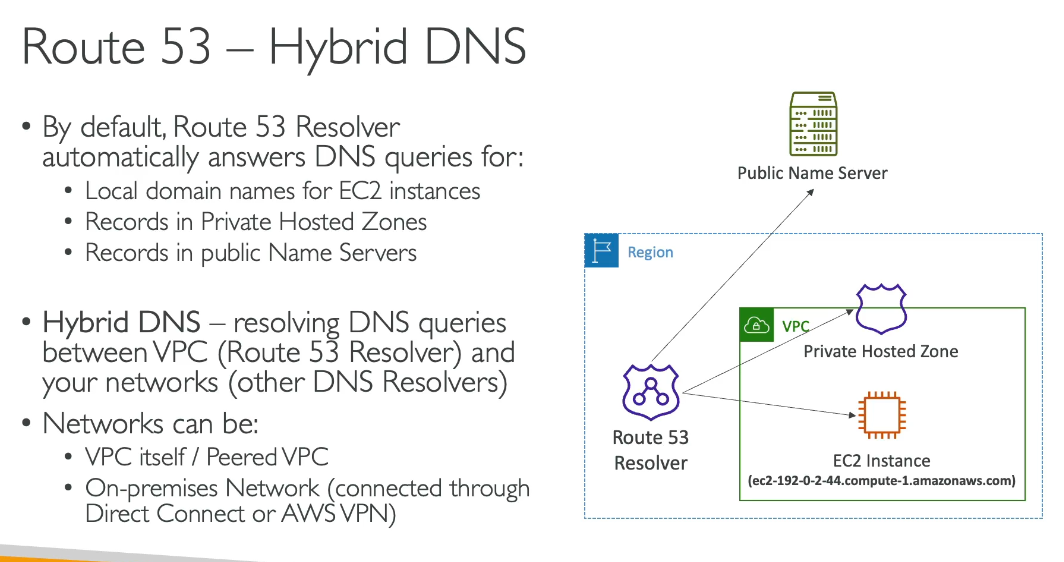
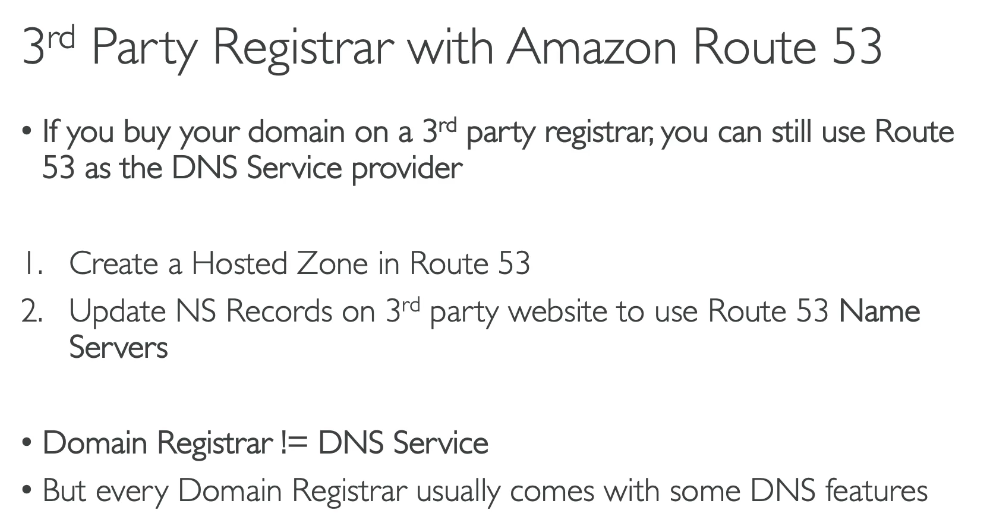
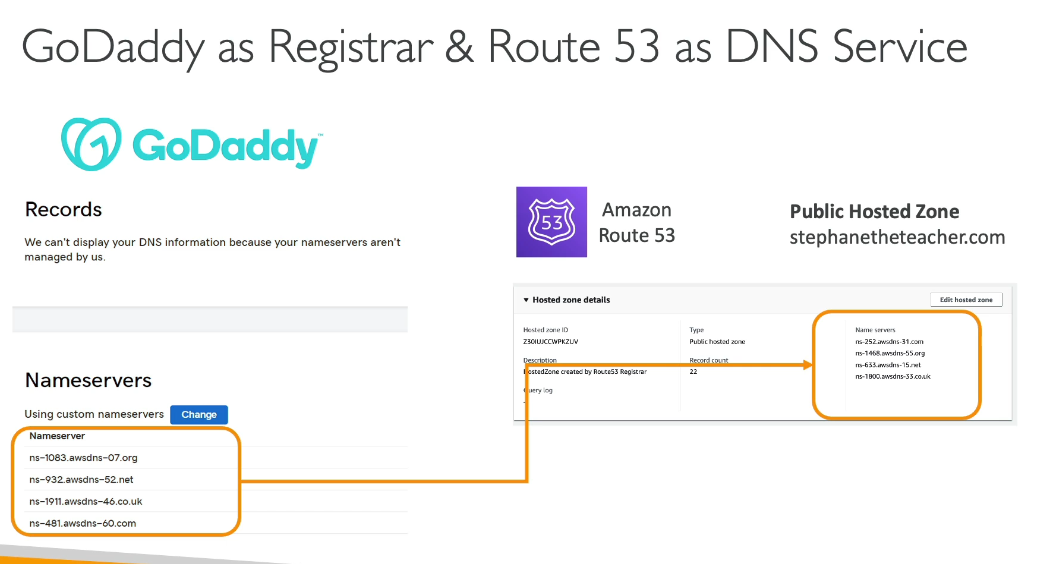
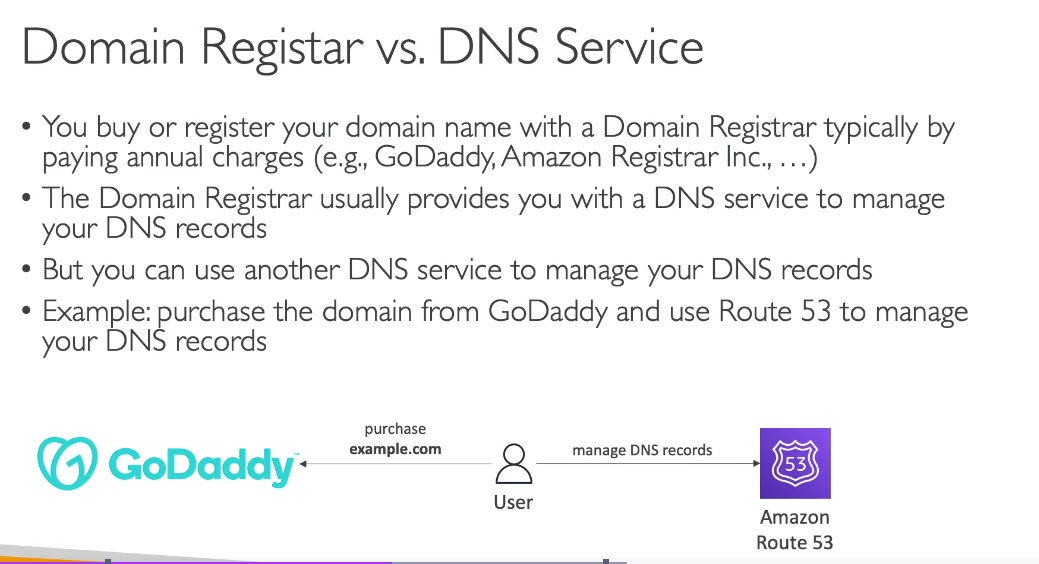
**Include: IP Based Routing Policy too**



**Amazon Route 53 supports the following routing policies:**

* **Simple routing policy** – Routes internet traffic to a single resource that performs a given function for your domain (e.g., a web server)
* **Failover routing policy** – Configures active-passive failover, automatically redirecting traffic from an unhealthy primary resource to a healthy secondary resource
* **Geolocation routing policy** – Routes traffic based on the geographic location of your users
* **Geoproximity routing policy** – Routes traffic based on the geographic location of your resources and users, with optional biasing to shift traffic between locations
* **Latency routing policy** – Routes traffic to the AWS Region that provides the best (lowest) latency for your users
* **Weighted routing policy** – Distributes traffic across multiple resources in proportions you specify
* **Multivalue answer routing policy** – Returns multiple healthy records (up to 8) selected randomly in response to DNS queries
* **IP-based routing policy** – Routes traffic to resources based on the IP addresses of your users

You can use these routing policies in both public and private hosted zones (where supported), and they can be combined to create sophisticated routing strategies tailored to your specific requirements.



**Route53 Left Sidebar Features:**

## [**Route 53**](https://241533148034-3dahyxga.us-east-1.console.aws.amazon.com/route53/v2/home?region=us-east-1#Home)

## **1. Dashboard**

**What it is:**  
 Just a summary page. Health checks, hosted zones, queries, alarms.

**Use case:**  
 Monitoring. Nothing more.

**Example:**  
 You glance here to confirm:

* DNS queries are flowing
* Health checks aren’t failing
* No sudden spikes (possible attack or misconfig)

👉 **No configuration happens here. Ignore it after setup.**

## **2. Hosted Zones (CORE FEATURE)**

This is **the heart of Route 53**.

### **What it is**

A **hosted zone = DNS records for a domain**.

Two types:

* **Public hosted zone** → Internet-facing DNS
* **Private hosted zone** → Internal DNS inside VPCs

### **What you do here**

Create records like:

* A / AAAA → map domain to IP
* CNAME → alias another name
* ALIAS → AWS-native CNAME (ELB, CloudFront, S3)
* MX, TXT, etc.

### **Use case**

Whenever you own a domain and want it to point somewhere.

### **Example**

You own rosu.com.

Public hosted zone:

rosu.com → ALB  
api.rosu.com → ECS service  
cdn.rosu.com → CloudFront

Private hosted zone:

db.internal → RDS endpoint  
redis.internal → ElastiCache

👉 **If you don’t understand hosted zones, you don’t understand Route 53. Period.**

## **3. Health Checks**

**DNS that reacts to failures.**

### **What it is**

Route 53 pings:

* HTTP/HTTPS endpoints
* TCP ports
* Or CloudWatch alarms

### **Use case**

Failover and high availability.

### **Example**

Two web servers:

* us-east-1
* eu-west-1

Health check:

* If US is down → DNS stops returning it
* Traffic shifts to EU

👉 **This is DNS-level failover, not load balancing. Slow but global.**

## **4. Profiles (NEW / Advanced)**

**Policy-based routing across AWS accounts and regions.**

### **What it is**

A modern replacement/extension for complex routing policies.

### **Use case**

Large enterprises with:

* Multi-account
* Multi-region
* Centralized traffic control

### **Example**

One global profile:

* Routes users to nearest region
* Applies health checks
* Managed centrally

👉 **If you’re a solo engineer or small team, you probably don’t need this yet.**

## **5. Global Resolver (Advanced / Enterprise)**

### **Global resolvers**

**Central DNS resolution across regions.**

### **What it is**

Lets multiple VPCs, across regions, share DNS logic.

### **Use case**

Hybrid / enterprise networks.

### **Example**

* VPC in us-east-1
* VPC in ap-south-1  
   Both resolve:

corp.internal

👉 **Overkill unless you run a global internal network.**

## **6. VPC Resolver (IMPORTANT FOR HYBRID)**

This is about **DNS inside VPCs**, not public internet.

### **a) VPCs**

Attach private hosted zones to VPCs.

### **b) Inbound endpoints**

**On-prem → AWS DNS**

Example:

* On-prem app queries db.aws.internal
* Inbound resolver answers

### **c) Outbound endpoints**

**AWS → On-prem DNS**

Example:

* EC2 asks for legacy.corp
* Resolver forwards to on-prem DNS

### **d) Rules**

Conditional forwarding.

Example:

\*.corp → on-prem DNS  
\*.aws → Route 53

### **e) Query logging**

Log DNS queries to CloudWatch/S3.

👉 **If you’ve ever debugged “why can’t my EC2 resolve this hostname”, this is where the truth lives.**

## **7. Outposts**

DNS support for **AWS Outposts** (on-prem AWS hardware).

Use case:

* AWS services running physically in your data center

👉 **Ignore unless your company paid millions for Outposts.**

## **8. Domains**

### **a) Registered domains**

Buy domains from AWS.

### **b) Requests**

Transfer, renew, update contacts.

Example:

* Buy rosu.dev
* Route 53 auto-creates hosted zone

👉 **Convenient, not special. DNS works the same no matter where you buy domains.**

## **9. IP-based routing (CIDR collections)**

**Route users based on their IP range.**

### **What it is**

Match client IP → different DNS response.

### **Use case**

Compliance, geo rules, corporate access.

### **Example**

* Office IPs → internal app
* Public IPs → public site

👉 **Blunt truth:** Geo routing is easier unless you have strict IP lists.

## **10. Traffic Flow**

**Visual DNS routing editor.**

### **What it is**

Drag-and-drop UI to build routing logic.

Supports:

* Weighted
* Latency
* Geo
* Failover
* IP-based

### **Use case**

Complex DNS logic with visualization.

### **Example**

Users →  
 If EU → eu-alb  
 If US → us-alb  
 If unhealthy → backup

👉 **Nice UI, but Terraform/CDK is cleaner if you’re serious.**

## **11. Policy Records**

DNS records created by **Traffic Flow policies**.

Use case:

* Apply Traffic Flow logic to actual domains

👉 **Just implementation detail.**

**Difference between Profiles and Global Resolvers:**

**Profiles** control **DNS decision-making**. They define *how Route 53 chooses an answer* (latency, health checks, failover, weighted traffic) and let you reuse that routing logic across multiple hosted zones, accounts, and regions. They’re about **policy and consistency**, not networking. Use Profiles when you’re tired of duplicating the same DNS routing rules everywhere.

**Global Resolvers** control **DNS resolution infrastructure**. They define *where DNS queries are resolved* for private DNS, letting multiple VPCs and regions share the same internal namespaces. They’re about **connectivity and plumbing**, not traffic optimization. Use Global Resolvers when you need global, cross-region private DNS that just works.

**Quiz:**

