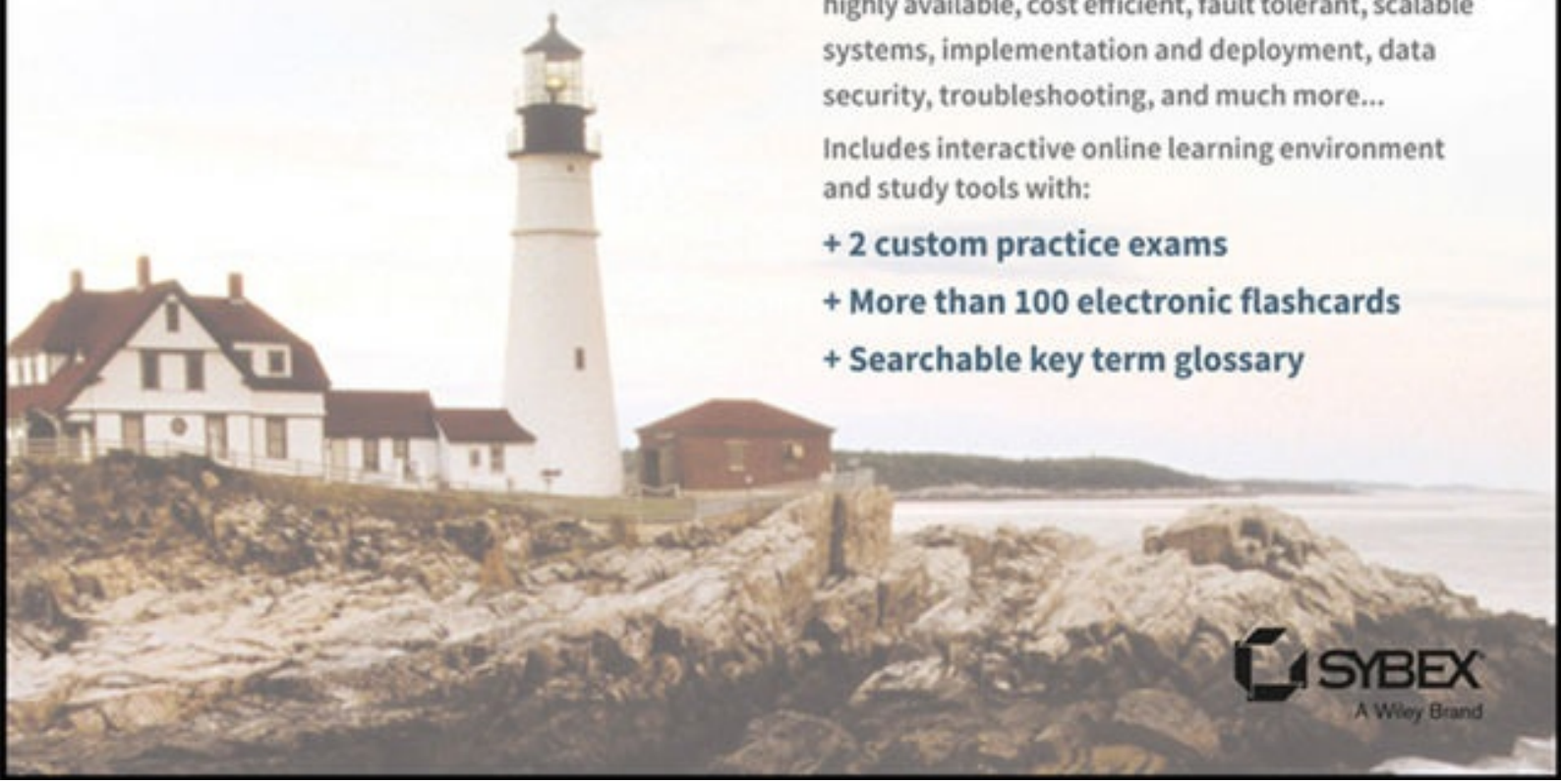
amazon web services

Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut,
Kevin E. Kelly, Sean Senior, John Stamper

AWS Certified Solutions Architect

OFFICIAL STUDY GUIDE

ASSOCIATE EXAM



Covers exam objectives, including designing highly available, cost efficient, fault tolerant, scalable systems, implementation and deployment, data security, troubleshooting, and much more...

Includes interactive online learning environment and study tools with:

+ 2 custom practice exams
+ More than 100 electronic flashcards
+ Searchable key term glossary

SYBEX
A Wiley Brand

network gateways. In addition, organizations can extend their corporate data center networks to AWS by using hardware or software *virtual private network (VPN)* connections or dedicated circuits by using AWS Direct Connect.

AWS Direct Connect

AWS Direct Connect allows organizations to establish a dedicated network connection from their data center to AWS. Using AWS Direct Connect, organizations can establish private connectivity between AWS and their data center, office, or colocation environment, which in many cases can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based VPN connections.

Amazon Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating human readable names, such as `www.example.com`, into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other. Amazon Route 53 also serves as domain registrar, allowing you to purchase and manage domains directly from AWS.

Storage and Content Delivery

AWS provides a variety of services to meet your storage needs, such as Amazon Simple Storage Service, Amazon CloudFront, and Amazon Elastic Block Store. This section provides an overview of the storage and content delivery services.

Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) provides developers and IT teams with highly durable and scalable object storage that handles virtually unlimited amounts of data and large numbers of concurrent users. Organizations can store any number of objects of any type, such as HTML pages, source code files, image files, and encrypted data, and access them using HTTP-based protocols. Amazon S3 provides cost-effective object storage for a wide variety of use cases, including backup and recovery, nearline archive, big data analytics, disaster recovery, cloud applications, and content distribution.

Amazon Glacier

Amazon Glacier is a secure, durable, and extremely low-cost storage service for data archiving and long-term backup. Organizations can reliably store large or small amounts of data for a very low cost per gigabyte per month. To keep costs low for customers, Amazon Glacier is optimized for infrequently accessed data where a retrieval time of several hours is suitable. Amazon S3 integrates closely with Amazon Glacier to allow organizations to choose the right storage tier for their workloads.

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides persistent block-level storage volumes for use with Amazon EC2 instances. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect organizations from component failure, offering high

Chapter 9
Domain Name System (DNS) and Amazon Route 53

THE AWS CERTIFIED SOLUTIONS ARCHITECT EXAM TOPICS COVERED IN THIS CHAPTER MAY INCLUDE, BUT ARE NOT LIMITED TO, THE FOLLOWING:

Domain 1.0: Designing highly available, cost-efficient, fault-tolerant, scalable systems

✓ **1.1 Identify and recognize cloud architecture considerations, such as fundamental components and effective designs.**

Content may include the following:

- How to design cloud services
- Planning and design
- Monitoring and logging
- Familiarity with:
  - Best practices for AWS architecture
  - Developing to client specifications, including pricing/cost (for example, on-demand vs. reserved vs. spot; RTO and RPO DR design)
  - Architectural trade-off decisions (for example, high availability vs. cost, Amazon Relational Database Service [RDS] vs. installing your own database on Amazon Elastic Compute Cloud—EC2)
  - Elasticity and scalability (for example, auto-scaling, SQS, ELB, CloudFront)

Domain 3.0: Data Security

✓ **3.1 Recognize and implement secure procedures for optimum cloud deployment and maintenance.**

✓ **3.2 Recognize critical disaster-recovery techniques and their implementation.**

- Amazon Route 53

Domain Name System (DNS)

The *Domain Name System (DNS)* is sometimes a difficult concept to understand because it is so ubiquitously used in making the Internet work. Before we get into the details, let's start with a simple analogy. The *Internet Protocol (IP)* address of your website is like your phone number—it could change if you move to a new area (at least your land line could change). DNS is like the phonebook. If someone wants to call you at your new house or location, they might look you up by name in the phonebook. If their phonebook hasn't been updated since you moved, however, they might call your old house. When a visitor wants to access your website, their computer takes the domain name typed in (`www.amazon .com`, for example) and looks up the IP address for that domain using DNS.

More specifically, DNS is a globally-distributed service that is foundational to the way people use the Internet. DNS uses a hierarchical name structure, and different levels in the hierarchy are each separated with a dot (`.`). Consider the domain names `www.amazon .com` and `aws.amazon.com`. In both these examples, `com` is the Top-Level Domain (TLD) and `amazon` is the Second-Level Domain (SLD). There can be any number of lower levels (for example, `www` and `aws`) below the SLD.

Computers use the DNS hierarchy to translate human readable names (for example, `www.amazon.com`) into the IP addresses (for example, `192.0.2.1`) that computers use to connect to one another. Every time you use a domain name, a DNS service must translate the name into the corresponding IP address. In summary, if you've used the Internet, you've used DNS.

Amazon Route 53 is an *authoritative DNS system*. An authoritative DNS system provides an update mechanism that developers use to manage their public DNS names. It then answers DNS queries, translating domain names into IP addresses so that computers can communicate with each other.

This chapter is intended to provide you with a baseline understanding of DNS and the Amazon Route 53 service that is designed to help users find your website or application over the Internet.

Domain Name System (DNS) Concepts

This section of the chapter defines DNS terms, describes how DNS works, and explains commonly used *record types*.

Top-Level Domains (TLDs)

A *Top-Level Domain (TLD)* is the most general part of the domain. The TLD is the farthest portion to the right (as separated by a dot). Common TLDs are `.com`, `.net`, `.org`, `.gov`, `.edu`, and `.io`.

TLDs are at the top of the hierarchy in terms of domain names. Certain parties are given management control over TLDs by the Internet Corporation for Assigned Names and Numbers (ICANN). These parties can then distribute domain names under the TLD, usually through a domain registrar. These domains are registered with the Network Information Center (InterNIC), a service of ICANN, which enforces the uniqueness of domain names

Amazon Route 53 Overview

Now that you have a foundational understanding of DNS and the different DNS record types, you can explore Amazon Route 53. *Amazon Route 53* is a highly available and scalable cloud DNS web service that is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications.

Amazon Route 53 performs three main functions:

- ***Domain registration***—Amazon Route 53 lets you register domain names, such as `example.com`.
- ***DNS service***—Amazon Route 53 translates friendly domain names like `www.example.com` into IP addresses like `192.0.2.1`. Amazon Route 53 responds to DNS queries using a global network of authoritative DNS servers, which reduces latency. To comply with DNS standards, responses sent over User Datagram Protocol (UDP) are limited to 512 bytes in size. Responses exceeding 512 bytes are truncated, and the resolver must re-issue the request over TCP.
- ***Health checking***—Amazon Route 53 sends automated requests over the Internet to your application to verify that it's reachable, available, and functional.

You can use any combination of these functions. For example, you can use Amazon Route 53 as both your registrar and your DNS service, or you can use Amazon Route 53 as the DNS service for a domain that you registered with another domain registrar.

Domain Registration

If you want to create a website, you first need to register the domain name. If you already registered a domain name with another registrar, you have the option to transfer the domain registration to Amazon Route 53. It isn't required to use Amazon Route 53 as your DNS service or to configure health checking for your resources.

Amazon Route 53 supports domain registration for a wide variety of generic TLDs (for example, `.com` and `.org`) and geographic TLDs (for example, `.be` and `.us`). For a complete list of supported TLDs, refer to the Amazon Route 53 Developer Guide at https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/.

Domain Name System (DNS) Service

As stated previously, Amazon Route 53 is an authoritative DNS service that routes Internet traffic to your website by translating friendly domain names into IP addresses. When someone enters your domain name in a browser or sends you an email, a DNS request is forwarded to the nearest Amazon Route 53 DNS server in a global network of authoritative DNS servers. Amazon Route 53 responds with the IP address that you specified.

If you register a new domain name with Amazon Route 53, Amazon Route 53 will be automatically configured as the DNS service for the domain, and a *hosted zone* will be created for your domain. You add resource record sets to the hosted zone, which define how you want Amazon Route 53 to respond to DNS queries for your domain (for example, with the IP address for a web server, the IP address for the nearest Amazon CloudFront edge location, or

the IP address for an Elastic Load Balancing load balancer).

If you registered your domain with another domain registrar, that registrar is probably providing the DNS service for your domain. You can transfer DNS service to Amazon Route 53, with or without transferring registration for the domain.

If you're using Amazon CloudFront, Amazon Simple Storage Service (Amazon S3), or Elastic Load Balancing, you can configure Amazon Route 53 to route Internet traffic to those resources.

Hosted Zones

A *hosted zone* is a collection of resource record sets hosted by Amazon Route 53. Like a traditional DNS zone file, a hosted zone represents resource record sets that are managed together under a single domain name. Each hosted zone has its own metadata and configuration information.

There are two types of hosted zones: private and public. A *private hosted zone* is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more Amazon Virtual Private Clouds (Amazon VPCs). A *public hosted zone* is a container that holds information about how you want to route traffic on the Internet for a domain (for example, `example.com`) and its subdomains (for example, `apex.example.com` and `acme.example.com`).

The resource record sets contained in a hosted zone must share the same suffix. For example, the `example.com` hosted zone can contain resource record sets for the `www.example.com` and `www.aws.example.com` subdomains, but it cannot contain resource record sets for a `www.example.ca` subdomain.


You can use Amazon S3 to host your static website at the hosted zone (for example, `domain.com`) and redirect all requests to a subdomain (for example, `www.domain.com`). Then, in Amazon Route 53, you can create an alias resource record that sends requests for the root domain to the Amazon S3 bucket.


Use an alias record, not a CNAME, for your hosted zone. CNAMEs are not allowed for hosted zones in Amazon Route 53.


Do not use A records for subdomains (for example, `www.domain.com`), as they refer to hardcoded IP addresses. Instead, use Amazon Route 53 alias records or traditional CNAME records to always point to the right resource, wherever your site is hosted, even when the physical server has changed its IP address.

Supported Record Types

Amazon Route 53 supports the following DNS resource record types. When you access Amazon Route 53 using the API, you will see examples of how to format the `Value` element for each record type. Supported record types include:

- A
- AAAA
- CNAME
- MX
- NS
- PTR
- SOA
- SPF
- SRV
- TXT
- Routing Policies

When you create a resource record set, you choose a *routing policy*, which determines how Amazon Route 53 responds to queries. Routing policy options are simple, weighted, latency-based, failover, and geolocation. When specified, Amazon Route 53 evaluates a resource's relative weight, the client's network latency to the resource, or the client's geographical location when deciding which resource to send back in a DNS response.

Routing policies can be associated with health checks, so resource health status is considered before it even becomes a candidate in a conditional decision tree. A description of possible routing policies and more on health checking is covered in this section.

Simple

This is the default routing policy when you create a new resource. Use a simple routing policy when you have a single resource that performs a given function for your domain (for example, one web server that serves content for the `example.com` website). In this case, Amazon Route 53 responds to DNS queries based only on the values in the resource record set (for example, the IP address in an A record).

Weighted

With weighted DNS, you can associate multiple resources (such as Amazon Elastic Compute Cloud [Amazon EC2] instances or Elastic Load Balancing load balancers) with a single DNS name.

Use the weighted routing policy when you have multiple resources that perform the same function (such as web servers that serve the same website), and you want Amazon Route 53 to route traffic to those resources in proportions that you specify. For example, you may use this for load balancing between different AWS regions or to test new versions of your website

(you can send 10 percent of traffic to the test environment and 90 percent of traffic to the older version of your website).

To create a group of weighted resource record sets, you need to create two or more resource record sets that have the same DNS name and type. You then assign each resource record set a unique identifier and a relative weight.

When processing a DNS query, Amazon Route 53 searches for a resource record set or a group of resource record sets that have the same name and DNS record type (such as an A record). Amazon Route 53 then selects one record from the group. The probability of any resource record set being selected is governed by the following formula:

$$\frac{\textit{Weight for a given resource record set}}{\textit{Sum of the weights for the resource record sets in the group}}$$

Latency-Based

Latency-based routing allows you to route your traffic based on the lowest network latency for your end user (for example, using the AWS region that will give them the fastest response time).

Use the latency routing policy when you have resources that perform the same function in multiple AWS Availability Zones or regions and you want Amazon Route 53 to respond to DNS queries using the resources that provide the best latency. For example, suppose you have Elastic Load Balancing load balancers in the U.S. West (Oregon) region and in the Asia Pacific (Singapore) region, and you created a latency resource record set in Amazon Route 53 for each load balancer. A user in London enters the name of your domain in a browser, and DNS routes the request to an Amazon Route 53 name server. Amazon Route 53 refers to its data on latency between London and the Singapore region and between London and the Oregon region. If latency is lower between London and the Oregon region, Amazon Route 53 responds to the user's request with the IP address of your load balancer in Oregon. If latency is lower between London and the Singapore region, Amazon Route 53 responds with the IP address of your load balancer in Singapore.

Failover

Use a failover routing policy to configure active-passive failover, in which one resource takes all the traffic when it's available and the other resource takes all the traffic when the first resource isn't available. Note that you can't create failover resource record sets for private hosted zones.

For example, you might want your primary resource record set to be in U.S. West (N. California) and your secondary, Disaster Recovery (DR), resource(s) to be in U.S. East (N. Virginia). Amazon Route 53 will monitor the health of your primary resource endpoints using a health check.

A health check tells Amazon Route 53 how to send requests to the endpoint whose health you want to check: which protocol to use (HTTP, HTTPS, or TCP), which IP address and port to use, and, for HTTP/HTTPS health checks, a domain name and path.

After you have configured a health check, Amazon will monitor the health of your selected DNS endpoint. If your health check fails, then failover routing policies will be applied and your DNS will fail over to your DR site.

Geolocation

Geolocation routing lets you choose where Amazon Route 53 will send your traffic based on the geographic location of your users (the location from which DNS queries originate). For example, you might want all queries from Europe to be routed to a fleet of Amazon EC2 instances that are specifically configured for your European customers, with local languages and pricing in Euros.

You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way so that each user location is consistently routed to the same endpoint.

You can specify geographic locations by continent, by country, or even by state in the United States. You can also create separate resource record sets for overlapping geographic regions, and priority goes to the smallest geographic region. For example, you might have one resource record set for Europe and one for the United Kingdom. This allows you to route some queries for selected countries (in this example, the United Kingdom) to one resource and to route queries for the rest of the continent (in this example, Europe) to a different resource.

Geolocation works by mapping IP addresses to locations. You should be cautious, however, as some IP addresses aren't mapped to geographic locations. Even if you create geolocation resource record sets that cover all seven continents, Amazon Route 53 will receive some DNS queries from locations that it can't identify.

In this case, you can create a default resource record set that handles both queries from IP addresses that aren't mapped to any location and queries that come from locations for which you haven't created geolocation resource record sets. If you don't create a default resource record set, Amazon Route 53 returns a "no answer" response for queries from those locations.

You cannot create two geolocation resource record sets that specify the same geographic location. You also cannot create geolocation resource record sets that have the same values for "Name" and "Type" as the "Name" and "Type" of non-geolocation resource record sets.

More on Health Checking

Amazon Route 53 health checks monitor the health of your resources such as web servers and email servers. You can configure Amazon CloudWatch alarms for your health checks so that you receive notification when a resource becomes unavailable. You can also configure Amazon Route 53 to route Internet traffic away from resources that are unavailable.

Health checks and DNS failover are major tools in the Amazon Route 53 feature set that help make your application highly available and resilient to failures. If you deploy an application in multiple Availability Zones and multiple AWS regions, with Amazon Route 53 health checks attached to every endpoint, Amazon Route 53 can send back a list of healthy endpoints only. Health checks can automatically switch to a healthy endpoint with minimal

disruption to your clients and without any configuration changes. You can use this automatic recovery scenario in active-active or active-passive setups, depending on whether your additional endpoints are always hit by live traffic or only after all primary endpoints have failed. Using health checks and automatic failovers, Amazon Route 53 improves your service uptime, especially when compared to the traditional monitor-alert-restart approach of addressing failures.

Amazon Route 53 health checks are not triggered by DNS queries; they are run periodically by AWS, and results are published to all DNS servers. This way, name servers can be aware of an unhealthy endpoint and route differently within approximately 30 seconds of a problem (after three failed tests in a row), and new DNS results will be known to clients a minute later (assuming your TTL is 60 seconds), bringing complete recovery time to about a minute and a half in total in this scenario.

 The 2014 AWS re:Invent session SDD408, "Amazon Route 53 Deep Dive: Delivering Resiliency, Minimizing Latency," introduced a set of best practices for Amazon Route 53. Explore those best practices to help you get started using Amazon Route 53 as a building block to deliver highly-available and resilient applications on AWS.

Amazon Route 53 Enables Resiliency

When pulling these concepts together to build an application that is highly available and resilient to failures, consider these building blocks:

- In every AWS region, an Elastic Load Balancing load balancer is set up with cross-zone load balancing and connection draining. This distributes the load evenly across all instances in all Availability Zones, and it ensures requests in flight are fully served before an Amazon EC2 instance is disconnected from an Elastic Load Balancing load balancer for any reason.
- Each Elastic Load Balancing load balancer delegates requests to Amazon EC2 instances running in multiple Availability Zones in an auto-scaling group. This protects the application from Availability Zone outages, ensures that a minimal amount of instances is always running, and responds to changes in load by properly scaling each group's Amazon EC2 instances.
- Each Elastic Load Balancing load balancer has health checks defined to ensure that it delegates requests only to healthy instances.
- Each Elastic Load Balancing load balancer also has an Amazon Route 53 health check associated with it to ensure that requests are routed only to load balancers that have healthy Amazon EC2 instances.
- The application's production environment (for example, `prod.domain.com`) has Amazon Route 53 alias records that point to Elastic Load Balancing load balancers. The production environment also uses a latency-based routing policy that is associated with Elastic Load Balancing health checks. This ensures that requests are routed to a healthy load balancer, thereby providing minimal latency to a client.

- The application's failover environment (for example, `fail.domain.com`) has an Amazon Route 53 alias record that points to an Amazon CloudFront distribution of an Amazon S3 bucket hosting a static version of the application.
- The application's subdomain (for example, `www.domain.com`) has an Amazon Route 53 alias record that points to `prod.domain.com` (as primary target) and `fail.domain .com` (as secondary target) using a failover routing policy. This ensures `www.domain.com` routes to the production load balancers if at least one of them is healthy or the "fail whale" if all of them appear to be unhealthy.
- The application's hosted zone (for example, `domain.com`) has an Amazon Route 53 alias record that redirects requests to `www.domain.com` using an Amazon S3 bucket of the same name.
- Application content (both static and dynamic) can be served using Amazon CloudFront. This ensures that the content is delivered to clients from Amazon CloudFront edge locations spread all over the world to provide minimal latency. Serving dynamic content from a Content Delivery Network (CDN), where it is cached for short periods of time (that is, several seconds), takes the load off of the application and further improves its latency and responsiveness.
- The application is deployed in multiple AWS regions, protecting it from a regional outage.

Summary

In this chapter, you learned the fundamentals of DNS, which is the methodology that computers use to convert human-friendly domain names (for example, `amazon.com`) into IP addresses (such as `192.0.2.1`).

DNS starts with TLDs (for example, `.com`, `.edu`). The Internet Assigned Numbers Authority (IANA) controls the TLDs in a root zone database, which is essentially a database of all available TLDs.

DNS names are registered with a domain registrar. A registrar is an authority that can assign domain names directly under one or more TLDs. These domains are registered with InterNIC, a service of ICANN, which enforces the uniqueness of domain names across the Internet. Each domain name becomes registered in a central database, known as the WhoIS database.

DNS consists of a number of different record types, including but not limited to the following:

- A
- AAAA
- CNAME
- MX
- NS
- PTR
- SOA
- SPF
- TXT

Amazon Route 53 is a highly available and highly scalable AWS-provided DNS service. Amazon Route 53 connects user requests to infrastructure running on AWS (for example, Amazon EC2 instances and Elastic Load Balancing load balancers). It can also be used to route users to infrastructure outside of AWS.

With Amazon Route 53, your DNS records are organized into hosted zones that you configure with the Amazon Route 53 API. A hosted zone simply stores records for your domain. These records can consist of A, CNAME, MX, and other supported record types.

Amazon Route 53 allows you to have several different routing policies, including the following:

- ***Simple***—Most commonly used when you have a single resource that performs a given function for your domain
- ***Weighted***—Used when you want to route a percentage of your traffic to one particular resource or resources
- ***Latency-Based***—Used to route your traffic based on the lowest latency so that your

users get the fastest response times

- ***Failover***—Used for DR and to route your traffic from your resources in a primary location to a standby location
- ***Geolocation***—Used to route your traffic based on your end user's location

Remember to pull these concepts together to build an application that is highly available and resilient to failures. Use Elastic Load Balancing load balancers across Availability Zones with connection draining enabled, use health checks defined to ensure that the application delegates requests only to healthy Amazon EC2 instances, and use a latency-based routing policy with Elastic Load Balancing health checks to ensure requests are routed with minimal latency to clients. Use Amazon CloudFront edge locations to spread content all over the world with minimal client latency. Deploy the application in multiple AWS regions, protecting it from a regional outage.

Exam Essentials

Understand what DNS is. DNS is the methodology that computers use to convert human-friendly domain names (for example, `amazon.com`) into IP addresses (such as `192.0.2.1`).

Know how DNS registration works. Domains are registered with domain registrars that in turn register the domain name with InterNIC, a service of ICANN. ICANN enforces uniqueness of domain names across the Internet. Each domain name becomes registered in a central database known as the WhoIS database. Domains are defined by their TLDs. TLDs are controlled by IANA in a root zone database, which is essentially a database of all available TLDs.

Remember the steps involved in DNS resolution. Your browser asks the resolving DNS server what the IP address is for `amazon.com`. The resolving server does not know the address, so it asks a root server the same question. There are 13 root servers around the world, and these are managed by ICANN. The root server replies that it does not know the answer to this, but it can give an address to a TLD server that knows about `.com` domain names. The resolving server then contacts the TLD server. The TLD server does not know the address of the domain name either, but it does know the address of the resolving name server. The resolving server then queries the resolving name server. The resolving name server contains the authoritative records and sends these to the resolving server, which then saves these records locally so it does not have to perform these steps again in the near future. The resolving name server returns this information to the user's web browser, which also caches the information.

Remember the different record types. DNS consists of the following different record types: A (address record), AAAA (IPv6 address record), CNAME (canonical name record or alias), MX (mail exchange record), NS (name server record), PTR (pointer record), SOA (start of authority record), SPF (sender policy framework), SRV (service locator), and TXT (text record). You should know the differences among each record type.

Remember the different routing policies. With Amazon Route 53, you can have different routing policies. The simple routing policy is most commonly used when you have a single resource that performs a given function for your domain. Weighted routing is used when you want to route a percentage of your traffic to a particular resource or resources. Latency-based routing is used to route your traffic based on the lowest latency so that your users get the fastest response times. Failover routing is used for DR and to route your traffic from a primary resource to a standby resource. Geolocation routing is used to route your traffic based on your end user's location.