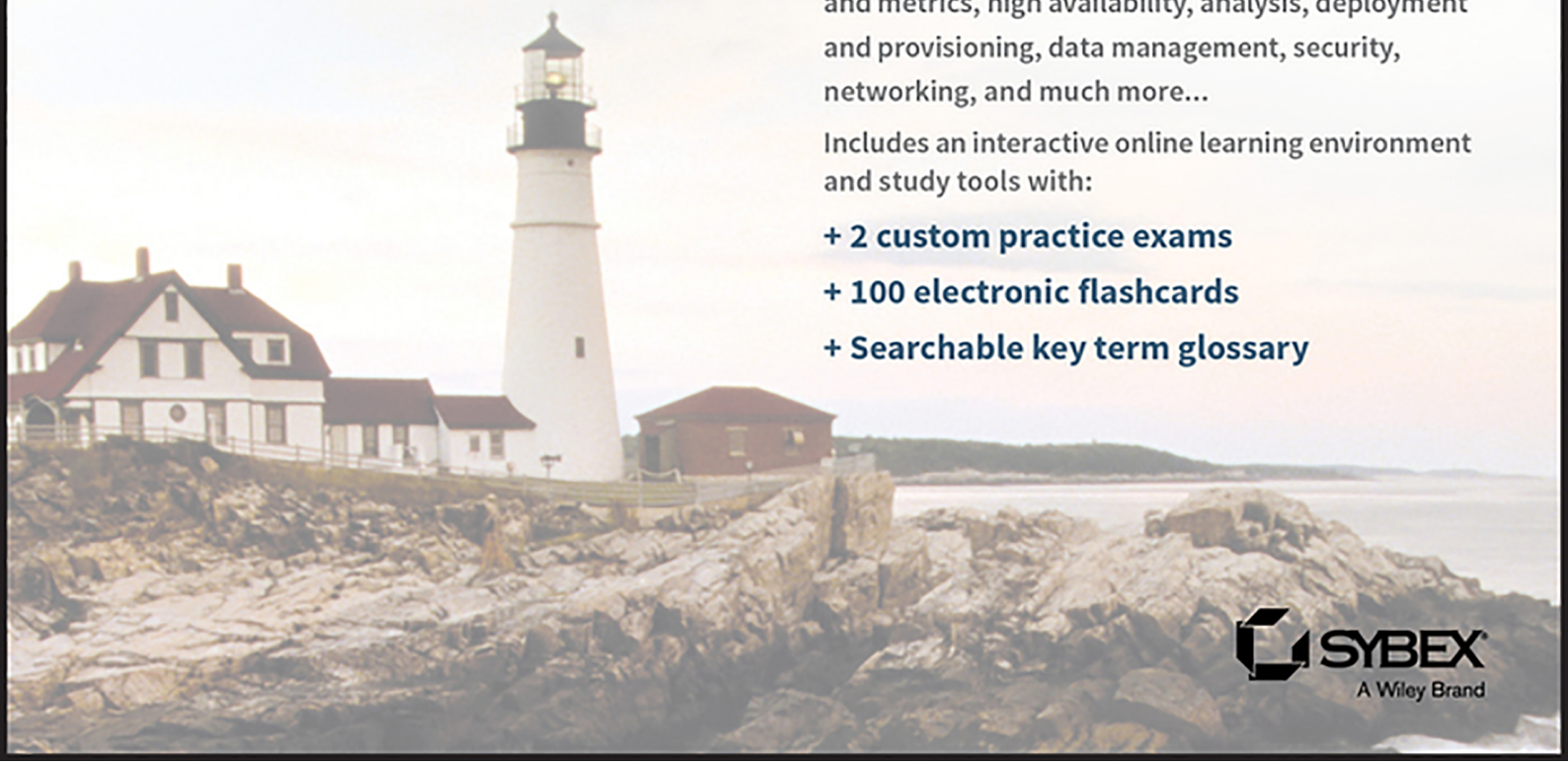
amazon web services

Stephen Cole, Gareth Digby, Chris Fitch,
Steve Friedberg, Shaun Qualheim, Jerry Rhoads,
Michael Roth, Blaine Sundrud

AWS Certified SysOps Administrator

OFFICIAL STUDY GUIDE

ASSOCIATE EXAM



Covers exam objectives, including monitoring and metrics, high availability, analysis, deployment and provisioning, data management, security, networking, and much more...

Includes an interactive online learning environment and study tools with:

+ 2 custom practice exams

+ 100 electronic flashcards

+ Searchable key term glossary

SYBEX
A Wiley Brand

Introduction to Networking on AWS

This chapter introduces you to a number of network services that AWS provides. Some of the services described in this chapter, such as Amazon Virtual Private Cloud (Amazon VPC), are fundamental to the operation of services on AWS. Others, like Amazon Route 53, offer services that, while optional, provide tight integration with AWS products and services.

The primary goal of this book is to prepare you for the AWS Certified SysOps Administrator - Associate exam; however, we want to do more for you. The authors of this book want to provide you with as much information as possible to assist you in your everyday journey as a Systems Operator.

The AWS services covered in this chapter include:

Amazon VPC With Amazon VPC, you provision a logically-isolated section of the AWS Cloud where you launch AWS resources in a virtual network that you have defined. You have complete control over your virtual networking environment.

AWS Direct Connect AWS Direct Connect allows you to establish a dedicated network connection from your premises to AWS.

Elastic Load Balancing Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon Elastic Compute Cloud (Amazon EC2) instances in an AWS Region. You can achieve fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Virtual Private Network (VPN) connections With VPN connections, you can connect Amazon VPC to remote networks. You can take advantage of AWS infrastructure to build a highly available, highly scalable solution, or you can build your own solution.

Amazon Route 53 Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. You can use Amazon Route 53 for domain management and for DNS service to connect to both AWS and non-AWS resources.

Amazon CloudFront Amazon CloudFront is a global Content Delivery Network (CDN) service that accelerates delivery of your websites, Application Programming Interfaces (APIs), video content, or other web assets.

CloudHub

AWS VPN CloudHub also uses a VGW and, using a hub-and-spoke model, connects multiple customer gateways. AWS VPN CloudHub uses BGP with each customer location having a unique ASN.

Software VPN

Creating a software VPN involves spinning up one or more Amazon EC2 instances in one or more Availability Zones within the region and then loading VPN software onto those Amazon EC2 instances. The VPN software can be acquired directly by the customer, or it can be acquired via AWS Marketplace. AWS Marketplace offers a number of options, including OpenVPN, Cisco, Juniper, and Brocade, among others.

VPN Management

VGW is highly available and scalable. Each VGW comes with two publicly accessible IP addresses. This means that a VGW sets up two separate IPsec tunnels. You need to provision two public IP addresses on your side. These can be on a single customer gateway, two customer gateways at the same location, or two customer gateways in two different locations. You can connect multiple customer gateways to the same VGW.

AWS VPN CloudHub is highly available and scalable. With AWS VPN CloudHub, each location advertises their appropriate routes over their VPN connection. AWS VPN CloudHub receives these advertisements and re-advertises them out to the other customer gateways. This allows each site to send and receive data from the other customer sites.

Creating a software VPN gives you both the greatest level of control and the greatest level of responsibility. You spin up the instance or instances and are responsible for their placement, that they are the correct size (and can increase in size or number to meet increased demand), and that they are monitored and replaced if either is not working or working with reduced functionality.

AWS Direct Connect and VPN are the two most common methods used by large enterprises to connect their existing WAN with AWS. Most use both methods. Understanding the advantages and disadvantages of both methods and how you can use the two methods combined is very important.

Amazon Route 53

Amazon Route 53 is a highly available and scalable cloud DNS web service. DNS routes end users to Internet applications by translating names like `www.example.com` into the numeric IP addresses like `192.0.2.1` that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6.

You can use Amazon Route 53 to help you get a website or web application up and running. Amazon Route 53 enables you to perform three main functions:

Register domain names. Your website needs a name, such as `example.com`. Amazon Route 53 lets you register a name for your website or web application, known as a *domain name*.

Route Internet traffic to the resources for your domain. When a user opens a web browser and enters your domain name in the address bar, Amazon Route 53 helps the DNS connect the browser with your website or web application.

Check the health of your resources. Amazon Route 53 sends automated requests over the Internet to a resource, such as a web server, to verify that it is reachable, available, and functional. You also can choose to receive notifications when a resource becomes unavailable and choose to route Internet traffic away from unhealthy resources.

You can use any combination of these functions. For example, you can use Amazon Route 53 both to register your domain name and to route Internet traffic for the domain, or you can use Amazon Route 53 to route Internet traffic for a domain that you registered with another domain registrar. If you choose to use Amazon Route 53 for all three functions, you register your domain name, then configure Amazon Route 53 to route Internet traffic for your domain, and finally configure Amazon Route 53 to check the health of your resources. You can use Amazon Route 53 to manage both public and private hosted zones. So, you can use Amazon Route 53 to distribute traffic between multiple AWS Regions and to distribute traffic within an AWS Region.

Amazon Route 53 Implementation

In addition to registering new domains, you can transfer existing domains. When you register a domain with Amazon Route 53, a hosted zone is automatically created for that domain. This makes it easier to use Amazon Route 53 as the DNS service provider for this domain. You are, however, not obligated to use Amazon Route 53 as the DNS service provider. You may route your DNS queries to another DNS provider.

When you're using Amazon Route 53 as the DNS service provider, you need to configure the DNS service. As mentioned, when you use Amazon Route 53 as the domain registrar, a hosted zone is automatically created for you. If you are not using Amazon Route 53 as the domain registrar, then you will need to create a hosted zone.

A *hosted zone* contains information about how you want to route your traffic both for your domain and for any subdomains that you may have. Amazon Route 53 assigns a unique set of nameservers for each hosted zone that you create. You can use this set of nameservers for multiple hosted zones if you want.

A hosted zone is a collection of resource record sets for a specified domain. You create a hosted zone for a domain (such as `example.com`), and then you create resource record sets to tell the Domain Name System how you want traffic to be routed for that domain.

After you have created your hosted zones, you need to create resource record sets. This involves two parts: the record type and the routing policy. Different routing policies can be applied to different record types.

A *routing policy* determines how Amazon Route 53 responds to queries. There are five routing policies available, and each of them is explained in this chapter.

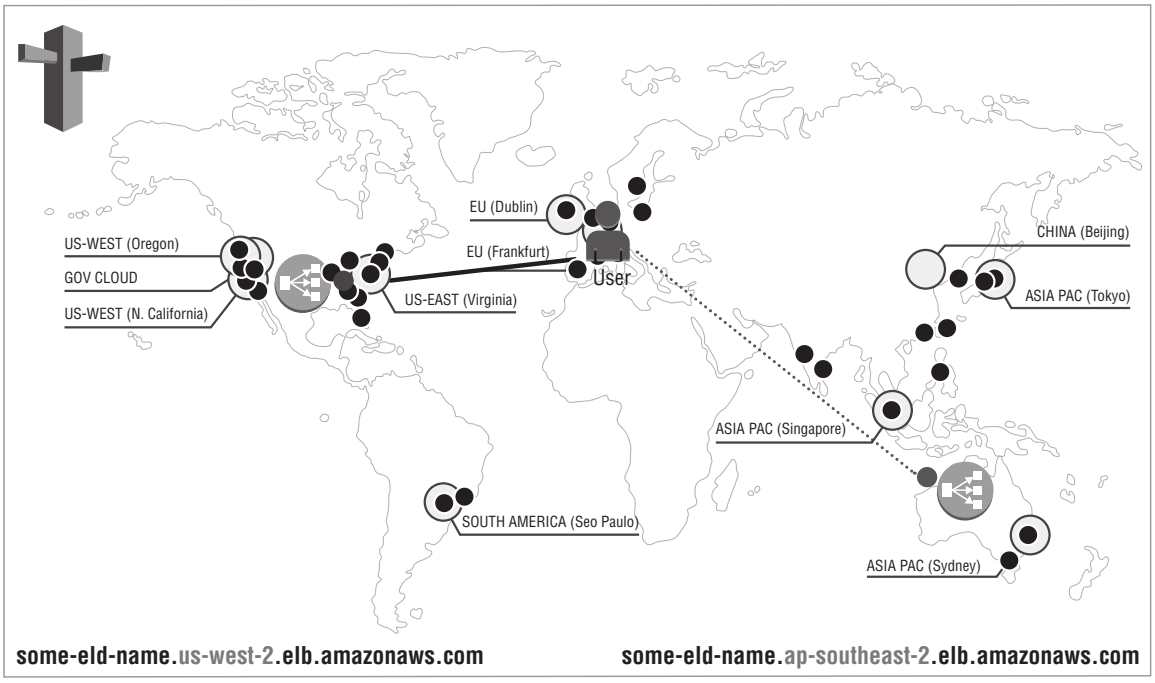
Simple Routing

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 Routing

Use the weighted routing policy when you have multiple resources that perform the same function (for example, 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, one quarter to one server and three quarters to the other). Figure 5.8 demonstrates weighted routing.

FIGURE 5.8 Weighted routing

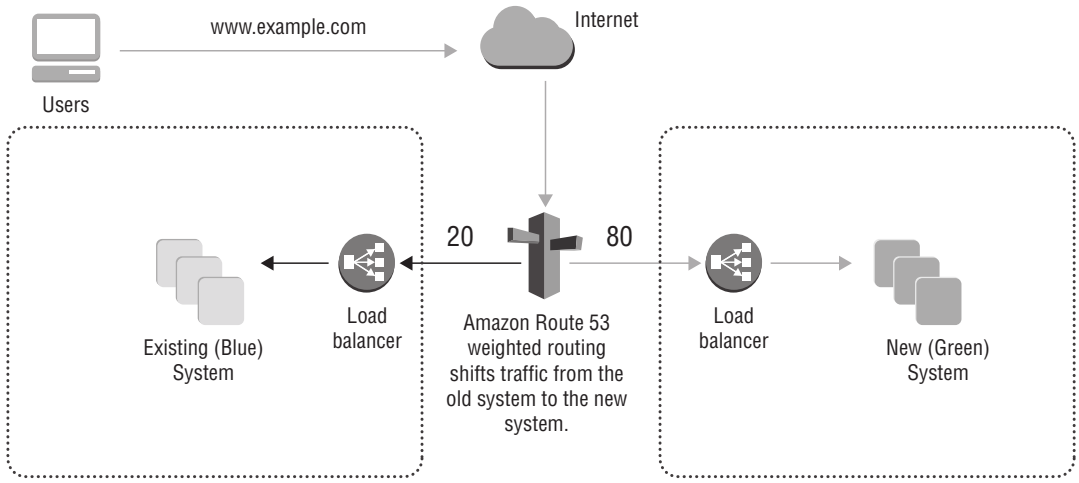


Latency-Based Routing

Use the latency routing policy when you have resources in multiple Amazon EC2 datacenters that perform the same function and you want Amazon Route 53 to respond to DNS queries

with the resources that provide the best latency. For example, you might have web servers for `example.com` in the Amazon EC2 datacenters in Ireland and in Tokyo. When a user browses to `example.com`, Amazon Route 53 chooses to respond to the DNS query based on which datacenter gives your user the lowest latency. Figure 5.9 demonstrates this concept.

FIGURE 5.9 Latency-based routing



Geolocation Routing

Use the geolocation routing policy when you want Amazon Route 53 to respond to DNS queries based on the location of your users. Geolocation routing returns the resource based on the geographic location of the user. You can specify geographic locations by continent, country, or state within the United States.

Some IP addresses aren't mapped to geographic locations, so 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. You can create a default resource record set that handles both queries from IP addresses that aren't mapped to any location. If you don't create a default resource record set, Amazon Route 53 returns a "no answer" response for queries from those locations.

Failover Routing

When using a failover routing policy, you designate a primary resource and a secondary resource. The secondary resource takes over in the event of a failure of the primary resource. To accomplish this, you configure a health check for the primary resource record set. If the health check fails, Amazon Route 53 routes the traffic to the secondary resource. It is recommended, but not obligatory, to configure a health check for the secondary

resource. If both record sets are unhealthy, Amazon Route 53 returns the primary resource record set. Health checks are discussed in greater detail in Chapter 10.

You can combine routing (for example, have geolocation routing backed up with failover routing) to make sure that you provide the highest level of availability possible. As you can imagine, this can get very complex (and confusing) very quickly, so good documentation is important.

DNS Record Types

Explaining the various DNS records types is out of the scope of this book. However, Table 5.7 shows the supported record types for Amazon Route 53.

TABLE 5.7 Amazon Route 53 Supported DNS Record Types

Record Type	Description
A	Address mapping records
AAAA	IPv6 address records
CNAME	Canonical name records
MX	Mail exchanger record
NAPTR	Name authority pointer record
NS	Name server records
PTR	Reverse-lookup Pointer records
SOA	Start of authority records
SPF	Sender policy framework record
SRV	Service record
TXT	Text records

In addition to the standard DNS record types supported, Amazon Route 53 supports a record type called *Alias*. An *Alias record type*, instead of pointing to an IP address or a domain name, points to one of the following:

- An Amazon CloudFront distribution
- An AWS Elastic Beanstalk environment

- An Elastic Load Balancing Classic or Application Load Balancer
- An Amazon S3 bucket that is configured as a static website
- Another Amazon Route 53 resource record set in the same hosted zone

Health Checks

There are three types of health checks that you can configure with Amazon Route 53. They are as follows:

- The health of a specified resource, such as a web server
- The status of an Amazon CloudWatch alarm
- The status of other health checks

In this section, we explore each type. The level of detail covered may not be tested on the exam. However, as an AWS Certified Systems Operator, the material covered here is a must-know.

The health of a specified resource, such as a web server You can configure a health check that monitors an endpoint that you specify either by IP address or by domain name. At regular intervals that you specify, Amazon Route 53 submits automated requests over the Internet to your application, server, or other resource to verify that it's reachable, available, and functional. Optionally, you can configure the health check to make requests similar to those that your users make, such as requesting a web page from a specific URL.

The status of an Amazon CloudWatch alarm You can create CloudWatch alarms that monitor the status of CloudWatch metrics, such as the number of throttled read events for an Amazon DynamoDB database or the number of Elastic Load Balancing hosts that are considered healthy. After you create an alarm, you can create a health check that monitors the same data stream that CloudWatch monitors for the alarm.

To improve resiliency and availability, Amazon Route 53 doesn't wait for the CloudWatch alarm to go into the ALARM state. The status of a health check changes from healthy to unhealthy based on the data stream and on the criteria in the CloudWatch alarm. The status of a health check can change from healthy to unhealthy even before the state of the corresponding alarm has changed to ALARM in CloudWatch.

The status of other health checks You can create a health check that monitors whether Amazon Route 53 considers other health checks healthy or unhealthy. One situation where this might be useful is when you have multiple resources that perform the same function, such as multiple web servers, and your chief concern is whether some minimum number of your resources is healthy. You can create a health check for each resource without configuring notification for those health checks. Then you can create a health check that monitors the status of the other health checks and that notifies you only when the number of available web resources drops below a specified threshold.

Amazon Route 53 Management

You can access Amazon Route 53 in the following ways:

- AWS Management Console
- AWS SDKs
- Amazon Route 53 API
- AWS CLI
- AWS Tools for Windows PowerShell

The best tool for monitoring the status of your domain is the Amazon Route 53 dashboard. This dashboard will give a status of any new domain registrations, domain transfers, and any domains approaching expiration.

The tools used to monitor your DNS service with Amazon Route 53 are health checks, Amazon CloudWatch, and AWS CloudTrail. Health checks are discussed in the management section. Amazon CloudWatch monitors metrics like the number of health checks listed as healthy, the length of time an SSL handshake took, and the time it took for the health check to receive the first byte, among other metrics. AWS CloudTrail can capture all of the API requests made for Amazon Route 53. You can determine the user who invoked a particular API.

Amazon Route 53 Authentication and Access Control

To perform any operation on Amazon Route 53 resources, such as registering a domain or updating a resource record set, IAM requires you to authenticate to prove that you're an approved AWS user. If you're using the Amazon Route 53 console, you authenticate your identity by providing your AWS user name and a password. If you're accessing Amazon Route 53 programmatically, your application authenticates your identity for you by using access keys or by signing requests.

After you authenticate your identity, IAM controls your access to AWS by verifying that you have permissions to perform operations and to access resources. If you are an account administrator, you can use IAM to control the access of other users to the resources that are associated with your account.

Amazon CloudFront

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content—for example, `.html`, `.css`, `.php`, image, and media files—to end users. Amazon CloudFront delivers your content through a worldwide network of edge locations.

When an end user requests content that you're serving with Amazon CloudFront, the user is routed to the edge location that provides the lowest latency, so content is delivered with the

Amazon VPC peering guide:
`http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/Welcome.html`

VPN options:
`http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpn-connections.html`

NAT gateway fundamentals on AWS:
`http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html`

Amazon CloudFront documentation:
`https://aws.amazon.com/documentation/cloudfront/`

Amazon Route 53 documentation: `https://aws.amazon.com/documentation/route53`

Elastic Load Balancing documentation:
`https://aws.amazon.com/documentation/elastic-load-balancing/`

AWS Direct Connect and VPN deep dive:
`https://www.youtube.com/watch?v=Qep11X1r1QA`

Amazon CloudFront best practices: `https://www.youtube.com/watch?v=fgbJJ412qRE`

Exam Essentials

Understand what a VPC is. Know how to set up a VPC, and what are the minimum and maximum size of both a VPC and subnets.

Understand the purpose and use of route tables, network ACLs, and security groups. Know how to use each for controlling access and providing security.

Know what are the default values for route tables, network ACLs, and security groups. Know where those default values come from, how to modify them, and why you would modify them.

Understand the difference between a private and public subnet. Public subnets allow traffic to the Internet; private subnets do not. Know how to use Amazon EC2 instances in private subnets to have access the Internet.

Understand the role and function of the various ways to connect the VPC with outside resources. This includes Internet gateway, VPN gateway, Amazon S3 endpoint, VPC peering, NAT instances, and NAT gateways. Understand how to configure these services.

Understand what is an Elastic IP (EIP). Elastic supports public IPv4 addresses (as of this publication). Understand the difference between EIP and an ENI.

Understand what is an Elastic Network Interface (ENI). Elastic Network Interfaces can be assigned and reassigned to an Amazon EC2 instance. Understand why this is important.

Know what services operate within a VPC and what services operate outside a VPC. Amazon EC2 lives within a VPC. Services such as Amazon S3 live outside the VPC. Know the various ways to access these services.

Know what AWS Direct Connect is. Understand why it is used and the basic steps for setting it up. (Remember the seven steps listed in the AWS Direct Connect section of this chapter.)

Understand the concept of VIFs. Understand what a VIF is and the difference between a public and private VIF. Why would you use one versus the other? Understanding these concepts will be very helpful on the exam.

Understand the options for Elastic Load Balancing (Classic Load Balancer vs. Application Load Balancer). Know how each type of load balancer operates, why you would choose one over the other, and how to configure each.

Understand how health checks work in each type of load balancer. Classic Load Balancers and Application Load Balancers have different health check options. Know what they are!

Understand how listeners work. Understand rules, priorities, and conditions and how they interact.

Know how Amazon CloudWatch, AWS CloudTrail, and access logs work. Know what type of information each one provides.

Understand the role of security groups with load balancers. Be able to configure a security group and know how rules are applied.

Understand the various options for establishing an IPsec VPN tunnel from an Amazon VPC to a customer location. Know the operational and security implications of these options.

Know how Amazon Route 53 works as a DNS provider. Understand how it can be used for both public and private hosted zones.

Know what the different routing options are for Amazon Route 53. Understand how to configure the various routing options and how they work.

Know what an Amazon Route 53 routing policy is. Understand how it is applied in Amazon Route 53.

Understand what record types Amazon Route 53 supports and how they work. Know both standard and non-standard record sets.

Know the tools for managing and monitoring Amazon Route 53. Understand how Amazon CloudWatch and AWS CloudTrail work with Amazon Route 53. Go deep in understanding how all of the services in this chapter are monitored.

Know the purpose of Amazon CloudFront and how it works. Know what a distribution is and what an origin is. Know what types of files Amazon CloudFront can handle.

Know the steps to implement Amazon CloudFront. Remember there are three steps to do this.

Know the various methods for securing content in Amazon CloudFront. Know how to secure your content at the edge and at the origin.

Replacement Upgrade

The replacement upgrade method replaces in-place resources with newly provisioned resources. There are advantages and disadvantages between the in-place upgrade method and replacement upgrade method. You can perform a replacement upgrade in a number of ways. You can use an Auto Scaling policy to define how you want to add (scale out) or remove (scale in) instances. By coupling this with your update strategy, you can control the rollout of an application update as part of the scaling event.

For example, you can create a new Auto Scaling Launch Configuration that specifies a new AMI containing the new version of your application. Then you can configure the Auto Scaling group to use the new launch configuration. The Auto Scaling termination policy by default will first terminate the instance with the oldest launch configuration and that is closest to the next billing hour. This in effect provides the most cost-effective method to phase out all instances that use the previous configuration. If you are using Elastic Load Balancing, you can attach an additional Auto Scaling configuration behind the load balancer and use a similar approach to phase in newer instances while removing older instances.

Similarly, you can configure rolling deployments in conjunction with deployment services such as AWS Elastic Beanstalk and AWS CloudFormation. You can use update policies to describe how instances in an Auto Scaling group are replaced or modified as part of your update strategy. With these deployment services, you can configure the number of instances to get updated concurrently or in batches, apply the updates to certain instances while isolating in-service instances, and specify the time to wait between batched updates. In addition, you can cancel or roll back an update if you discover a bug in your application code. These features can help increase the availability of your application during updates.

Blue/Green Deployments

Blue/green is a method where you have two identical stacks of your application running in their own environments. You use various strategies to migrate the traffic from your current application stack (blue) to a new version of the application (green). This method is used for a replacement upgrade. During a blue/green deployment, the latest application revision is installed on replacement instances and traffic is rerouted to these instances either immediately or as soon as you are done testing the new environment.

This is a popular technique for deploying applications with zero downtime. Deployment services like AWS Elastic Beanstalk, AWS CloudFormation, or AWS OpsWorks are particularly useful for blue/green deployments because they provide a simple way to duplicate your existing application stack.

Blue/green deployments offer a number of advantages over in-place deployments. An application can be installed and tested on the new instances ahead of time and deployed to production simply by switching traffic to the new servers. Switching back to the most recent version of an application is faster and more reliable because traffic can be routed back to the original instances as long as they have not been terminated. With an in-place deployment, versions must be rolled back by redeploying the previous version of the application. Because the instances provisioned for a blue/green deployment are new, they reflect

the most up-to-date server configurations, which helps you avoid the types of problems that sometimes occur on long-running instances.

For a stateless web application, the update process is pretty straightforward. Simply upload the new version of your application and let your deployment service deploy a new version of your stack (green). To cut over to the new version, you simply replace the Elastic Load Balancing URLs in your Domain Name Server (DNS) records. AWS Elastic Beanstalk has a Swap Environment URLs feature to facilitate a simpler cutover process. If you use Amazon Route 53 to manage your DNS records, you need to swap Elastic Load Balancing endpoints for AWS CloudFormation or AWS OpsWorks deployment services.

For applications with session states, the cutover process can be complex. When you perform an update, you don't want your end users to experience downtime or lose data. You should consider storing the sessions outside of your deployment service because creating a new stack will re-create the session database with a certain deployment service. In particular, consider storing the sessions separately from your deployment service if you are using an Amazon RDS database.

If you use Amazon Route 53 to host your DNS records, you can consider using the Weighted Round Robin (WRR) feature for migrating from blue to green deployments. The feature helps to drive the traffic gradually rather than instantly. If your application has a bug, this method helps ensure that the blast radius is minimal, as it only affects a small number of users. This method also simplifies rollbacks if they become necessary by redirecting traffic back to the blue stack. In addition, you only use the required number of instances while you scale up in the green deployment and scale down in the blue deployment. For example, you can set WRR to allow 10 percent of the traffic to go to green deployment while keeping 90 percent of traffic on blue. You gradually increase the percentage of green instances until you achieve a full cutover. Keeping the DNS cache to a shorter Time To Live (TTL) on the client side also ensures that the client will connect to the green deployment with a rapid release cycle, thus minimizing bad DNS caching behavior. For more information on Amazon Route 53, see Chapter 5, "Networking."

Hybrid Deployments

You can also use the deployment services in a hybrid fashion for managing your application fleet. For example, you can combine the simplicity of managing AWS infrastructure provided by AWS Elastic Beanstalk and the automation of custom network segmentation provided by AWS CloudFormation. Leveraging a hybrid deployment model also simplifies your architecture because it decouples your deployment method so that you can choose different strategies for updating your application stack.

Deployment Services

AWS deployment services provide easier integration with other AWS Cloud services. Whether you need to load-balance across multiple Availability Zones by using Elastic Load Balancing or by using Amazon RDS as a back end, the deployment services like AWS Elastic

scale your database horizontally. Amazon RDS MySQL, PostgreSQL, and Maria DB can have up to 5 Read Replicas, and Amazon Aurora can have up to 15 Read Replicas.

You can also place your Read Replica in a different AWS Region closer to your users for better performance. Additionally, you can use Read Replicas to increase the availability of your database by promoting a Read Replica to a master for faster recovery in the event of a disaster. Read Replicas are not a replacement for the high availability and automatic failover capabilities that Multi-AZ architectures provide, however. For a refresher on Amazon RDS high availability and Read Replicas, refer to Chapter 7.

Multi-Region High Availability

In addition to building a highly available application that runs in a single region, your application may require regional fault tolerance. This can be delivered by placing and running infrastructure in another region and then using Amazon Route 53 to load balance the traffic between the regions.

Amazon Simple Storage Service

If you need to keep Amazon Simple Storage Service (Amazon S3) data in multiple regions, you can use cross-region replication. *Cross-region replication* is a bucket-level feature that enables automatic, asynchronous copying of objects across buckets in different AWS Regions. More information on Amazon S3 and cross-region replication is available in Chapter 6, "Storage Systems."

Amazon DynamoDB

Amazon DynamoDB uses DynamoDB Streams to replicate data between regions. An application in one AWS Region modifies the data in an Amazon DynamoDB table. A second application in another AWS Region reads these data modifications and writes the data to another table, creating a replica that stays in sync with the original table.

Amazon Route 53

When you have more than one resource performing the same function (for example, more than one HTTP/S server or mail server), you can configure Amazon Route 53 to check the health of your resources and respond to Domain Name System (DNS) queries using only the healthy resources. For example, suppose your website, `Example.com`, is hosted on 10 servers, 2 each in 5 regions around the world. You can configure Amazon Route 53 to

check the health of those servers and to respond to DNS queries for `Example.com` using only the servers that are currently healthy.

You can set up a variety of failover configurations using Amazon Route 53 alias, weighted, latency, geolocation routing, and failover resource record sets.

Active-active failover Use this failover configuration when you want all of your resources to be available the majority of the time. When a resource becomes unavailable, Amazon Route 53 can detect that it is unhealthy and stop including it when responding to queries.

Active-passive failover Use this failover configuration when you want a primary group of resources to be available the majority of the time and a secondary group of resources to be on standby in case all of the primary resources become unavailable. When responding to queries, Amazon Route 53 includes only the healthy primary resources. If all of the primary resources are unhealthy, Amazon Route 53 begins to include only the healthy secondary resources in response to DNS queries.

Active-active-passive and other mixed configurations You can combine alias and non-alias resource record sets to produce a variety of Amazon Route 53 behaviors. More information on record types can be found in Chapter 5.

In order for these failover configurations to work, health checks will need to be configured. There are three types of health checks: health checks that monitor an endpoint, health checks that monitor Amazon CloudWatch Alarms, and health checks that monitor other health checks.

The following sections discuss simple and complex failover configurations.

Health Checks for Simple Failover

The simplest failover configuration of having two or more resources performing the same function can benefit from health checks. For example, you might have multiple Amazon EC2 servers running HTTP server software responding to requests for your `Example.com` website. In Amazon Route 53, you create a group of resource record sets that have the same name and type, such as weighted resource record sets or latency resource record sets of type A. You create one resource record set for each resource, and you configure Amazon Route 53 to check the health of the corresponding resource. In this configuration, Amazon Route 53 chooses which resource record set will respond to a DNS query for `Example.com` and bases the choice in part on the health of your resources.

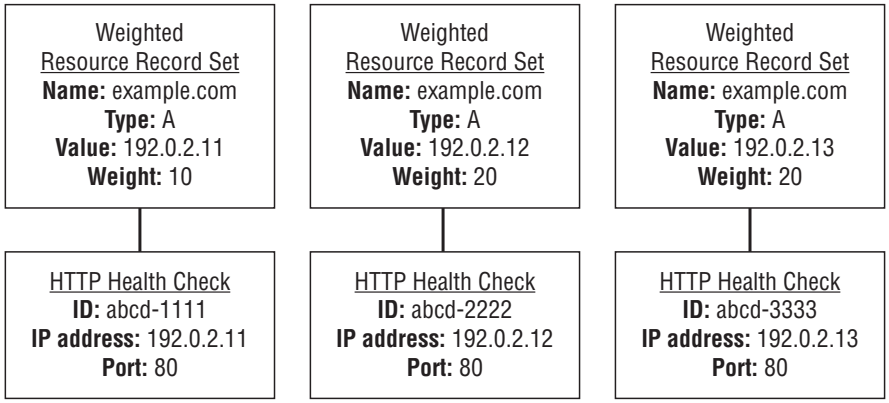
As long as all of the resources are healthy, Amazon Route 53 responds to queries using all of your `Example.com` weighted resource record sets. When a resource becomes unhealthy, Amazon Route 53 responds to queries using only the healthy resource record sets for `Example.com`.

Following are the steps for how you configure Amazon Route 53 to check the health of your resources in this simple configuration and how Amazon Route 53 responds to queries based on the health of your resources.

Configuring Amazon Route 53 to Check the Health of Your Resources

1. You identify the resources whose health you want Amazon Route 53 to monitor. For example, you might want to monitor all of the HTTP servers that respond to requests for `Example.com`.
2. You create health checks for your resources. A health check tells Amazon Route 53 how to send requests to the endpoint whose health you want to check: which protocol (HTTP, HTTPS, or Transmission Control Protocol [TCP]) and which IP address and port to use and also a domain name and path for HTTP/HTTPS health checks.

   A common configuration is to create one health check for each resource and to use the same IP address for the health check endpoint for the resource. If the IP address for your HTTP server is 192.0.2.117, you create a health check for which the IP address is 192.0.2.117.
3. You might need to configure router and firewall rules so that Amazon Route 53 can send regular requests to the endpoints that you specified in your health checks.
4. You create a group of resource record sets for your resources (for example, a group of weighted resource record sets that all have a type of A). You associate the health checks that you created in Step 2 with the corresponding resource record sets. The graphic illustrates how these health checks will operate.



5. Amazon Route 53 periodically sends a request to each endpoint that you specified when you created your health checks; it doesn't perform the health check when it receives a DNS query. Based on the responses, Amazon Route 53 decides whether the endpoints are healthy and uses that information to determine how to respond to queries.
6. When Amazon Route 53 receives a query for `Example.com`:
   a. Amazon Route 53 chooses a resource record set based on the routing policy. In this case, it chooses a resource record set based on weight.

b. It determines the current health of the selected resource record set by checking the status of the health check for that resource record set.

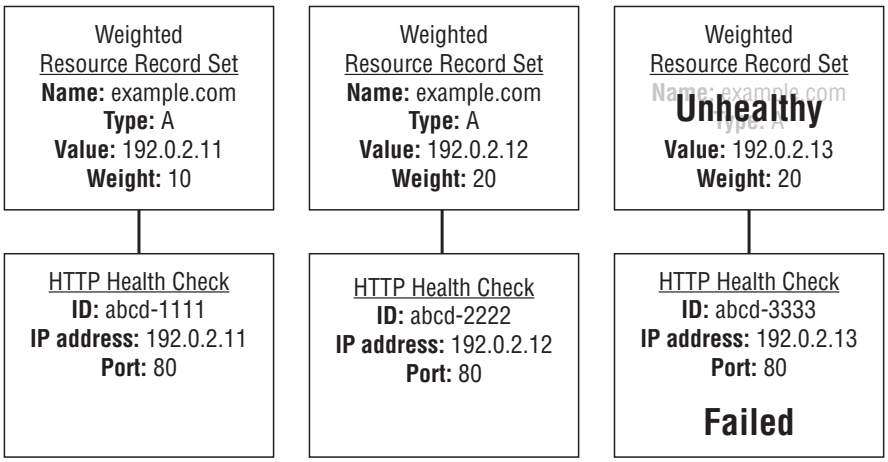
c. If the selected resource record set is unhealthy, it repeats the process of choosing a resource record set based on the routing policy. This time, the unhealthy resource record set isn't considered.

d. It responds to the query with the selected healthy resource record set.

The following example shows a group of weighted resource record sets in which the third resource record set is unhealthy. Initially, Amazon Route 53 selects a resource record set based on the weights of all three resource record sets. If it happens to select the unhealthy resource record set the first time, Amazon Route 53 selects another resource record set, but this time it omits the weight of the third resource record set from the calculation. See Figure 10.7 for an example of an unhealthy health check.

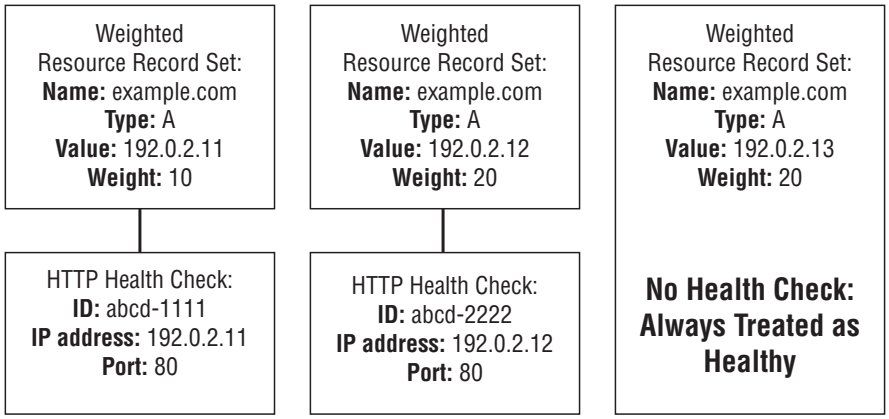
- When Amazon Route 53 initially selects from among all three resource record sets, it responds to requests using the first resource record set about 20 percent of the time: 10/(10 + 20 + 20).
- When Amazon Route 53 determines that the third resource record set is unhealthy, it responds to requests using the first resource record set about 33 percent of the time: 10/(10 + 20).

FIGURE 10.7 Unhealthy health check



If you omit a health check from one or more resource record sets in a group of resource record sets, Amazon Route 53 treats those resource record sets as healthy. Amazon Route 53 has no basis for determining the health of the corresponding resource without an assigned health check and might choose a resource record set for which the resource is unhealthy. See Figure 10.8 for more information.

FIGURE 10.8 No health check enabled

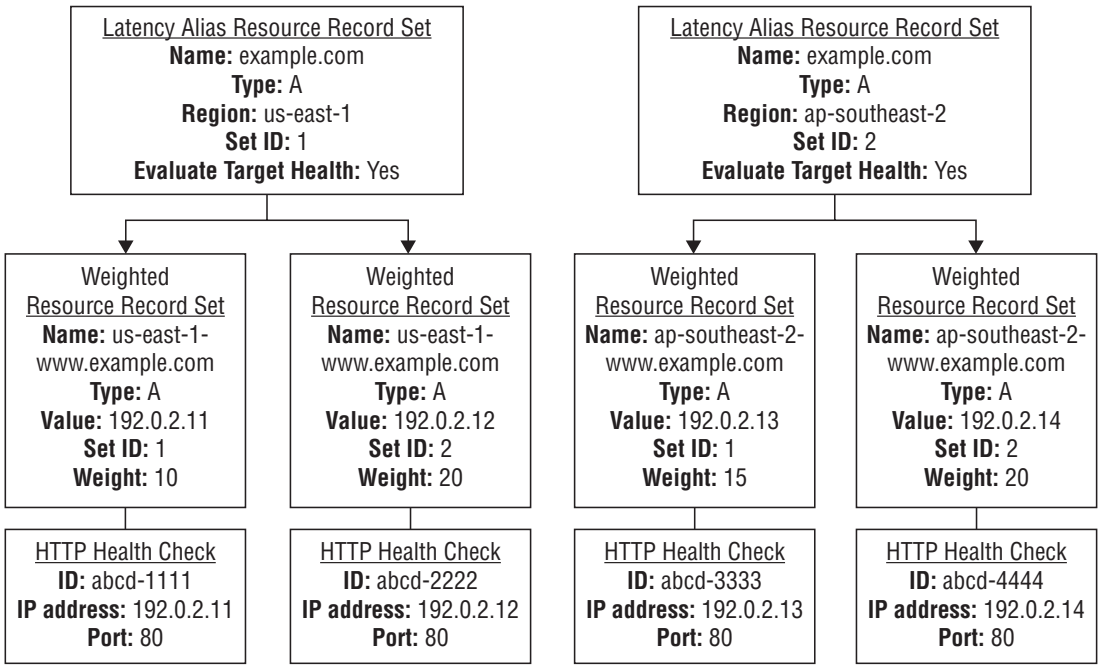


Health Checks for Complex Failover

Checking the health of resources in complex configurations works much the same way as in simple configurations. In complex configurations, however, you use a combination of alias resource record sets (including weighted alias, latency alias, and failover alias) and non-alias resource record sets to build a decision tree that gives you greater control over how Amazon Route 53 responds to requests.

For example, you might use latency alias resource record sets to select a region close to a user and use weighted resource record sets for two or more resources within each region to protect against the failure of a single endpoint or an Availability Zone. Figure 10.9 shows this configuration.

FIGURE 10.9 Health check for a complex failover



An overview of how Amazon EC2 and Amazon Route 53 are configured follows:

- You have Amazon EC2 instances in two regions: us-east-1 and ap-southeast-2. You want Amazon Route 53 to respond to queries by using the resource record sets in the region that provides the lowest latency for your customers, so you create a latency alias resource record set for each region. (You create the latency alias resource record sets after you create resource record sets for the individual Amazon EC2 instances.)
- Within each region, you have two Amazon EC2 instances. You create a weighted resource record set for each instance. The name and the type are the same for both of the weighted resource record sets in each region.

When you have multiple resources in a region, you can create weighted or failover resource record sets for your resources. You can also make even more complex configurations by creating weighted alias or failover alias resource record sets that, in turn, refer to multiple resources.

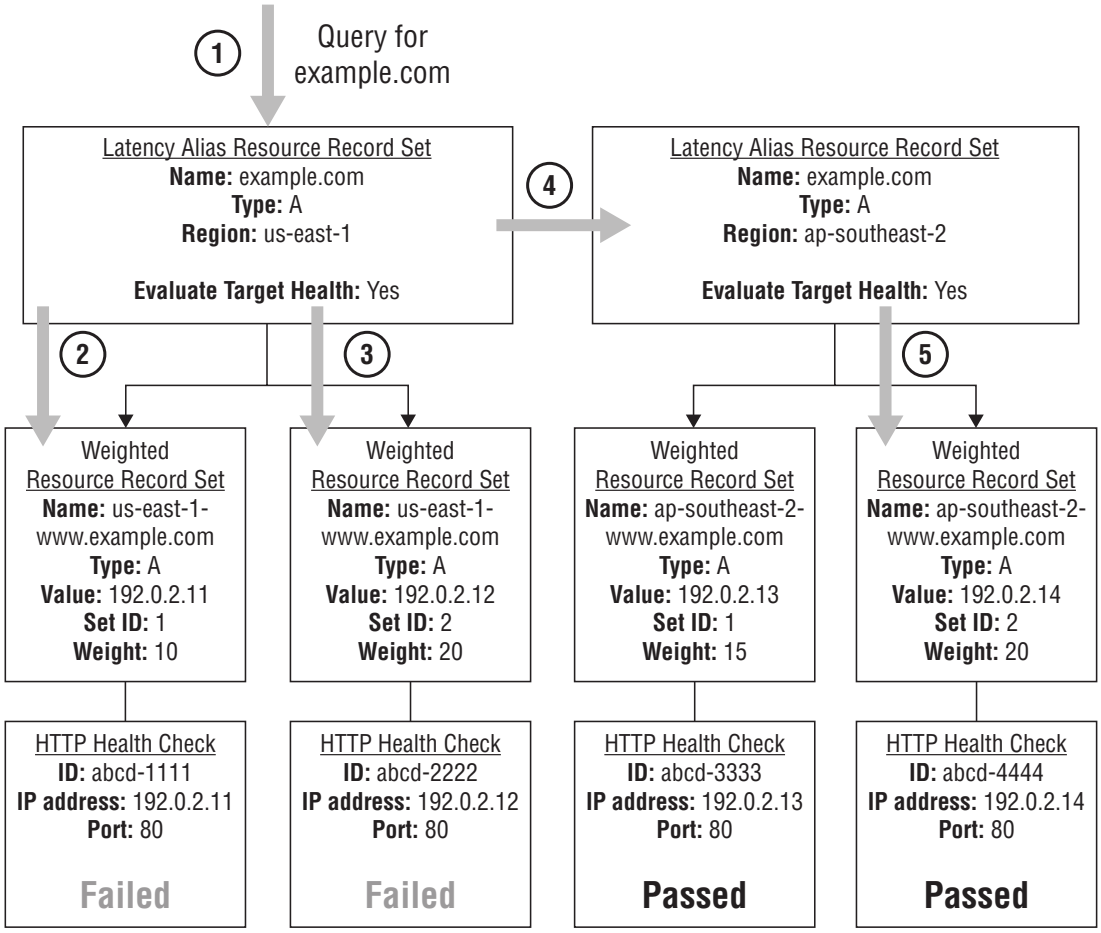
- Each weighted resource record set has an associated health check. The IP address for each health check matches the IP address for the corresponding resource record set. This isn't required, but it is the most common configuration.
- For both latency alias resource record sets, you set the value of Evaluate Target Health to Yes.

You use the Evaluate Target Health setting for each latency alias resource record set to make Amazon Route 53 evaluate the health of the alias targets—the weighted resource record sets—and respond accordingly.

Figure 10.10 demonstrates the sequence of events that follows:

1. Amazon Route 53 receives a query for `Example.com`. Based on the latency for the user making the request, Amazon Route 53 selects the latency alias resource record set for the us-east-1 region.
2. Amazon Route 53 selects a weighted resource record set based on weight. Evaluate Target Health is Yes for the latency alias resource record set, so Amazon Route 53 checks the health of the selected weighted resource record set.
3. The health check failed, so Amazon Route 53 chooses another weighted resource record set based on weight and checks its health. That resource record set also is unhealthy.
4. Amazon Route 53 backs out of that branch of the tree, looks for the latency alias resource record set with the next-best latency, and chooses the resource record set for ap-southeast-2.
5. Amazon Route 53 again selects a resource record set based on weight and then checks the health of the selected resource record set. The health check passed, so Amazon Route 53 returns the applicable value in response to the query.

FIGURE 10.10 Evaluate target health



Highly Available Connectivity Options

In this section of this chapter, we discuss how to make your connections to AWS redundant. In Chapter 5, we discussed the various connectivity options to connect to AWS and, more specifically, your Amazon VPC. We will discuss the Virtual Private Network (VPN) and AWS Direct Connect connectivity options and how to make them highly available.

Redundant Active-Active VPN Connections

To get your connectivity up and running quickly, you can implement VPN connections because they are a quick, easy, and cost-effective way to set up remote connectivity to your Amazon VPC. To enable redundancy, each AWS Virtual Private Gateway (VGW) has two VPN endpoints with capabilities for static and dynamic routing. Although statically routed