AWS® Certified Cloud Practitioner

STUDY GUIDE

FOUNDATIONAL (CLF-C01) EXAM

Includes interactive online learning environment and study tools:

Two custom practice exams

100 electronic flashcards

Searchable key term glossary

BEN PIPER
DAVID CLINTON

SYBEX
A Wiley Brand

Until you actively launch some services, there won't be anything happening that will actually generate billable events.

What's Available Under the Free Tier?

There are actually two kinds of free services:

- Light implementations of most AWS services that are available through the first 12 months of a new AWS account
- Light usage of certain AWS services that are always available—even after your initial 12 months are up

You can get a detailed rundown of everything that's available for both classes at `https://aws.amazon.com/free`. To give you some sense of what's out there, the following are some highlights.

The 12-Month Free Tier

We've already discussed running low-powered EC2 and RDS instances and using 5 GB of S3 storage capacity under the Free Tier. Other available free service features include the following:

- 30 GB of either magnetic or solid-state drive (SSD) volumes from Amazon Elastic Block Storage (EBS). EBS volumes are usually used as the primary data drives for EC2 instances. For context, 8 GB is normally more than enough space to run the Linux OS driving a busy web server.
- 500 MB/month of free storage on the Amazon Elastic Container Registry (ECR). ECR is a service for storing and managing Docker container images. 500 MB is enough space to satisfy the needs of nearly any use case.
- 50 GB of outbound data transfers and 2 million HTTP or HTTPS requests for Amazon CloudFront distributions. (These are annual rather than monthly amounts.) CloudFront is Amazon's content delivery network that you can use to serve cached copies of your content to remote users at low latency rates.
- One million API calls/month on Amazon API Gateway. API Gateway is a tool for managing your organization's application programming interfaces (APIs). APIs allow users and applications to connect with mobile and web-based resources.

Permanently Free Services

Some AWS services can be useful even when consumed at levels so low that they don't need to be billed. In such cases, AWS makes them available at those low levels over any time span. These use cases include the following:

- 10 custom monitoring metrics and 10 alarms on Amazon CloudWatch active at any given time. You use CloudWatch either directly or in combination with other tools to track resource performance and costs.

enough to run parallel resources if they're going to be sitting right next to each other in the same data center. That wouldn't do you a lot of good in the event of a building-wide blackout or a malicious attack, leaving your backup just as dead as the instance it was supposed to replace. So, you'll also need to distribute your resources across remote locations.

In this context, it really makes no difference whether your workload is running in your on-premises data center or in the Amazon cloud. In either case, you'll need resource redundancy that's also geographically parallel. What *is* different is how much *easier*—and sometimes cheaper—it can be to build resilience into your *cloud* infrastructure.

Since the AWS cloud is already available within dozens and dozens of Availability Zones spread across all continents (besides, for now at least, Antarctica), deploying or, at least, preparing quick-launch templates for remote backup instances is easy. And since AWS workloads can be requested and launched on-demand, the job of efficiently provisioning parallel resources is built into the platform's very DNA.

The configuration and automation stage can be a bit tricky, but that's for your developers and administrators to figure out, isn't it? They're the ones who took and passed the AWS Solutions Architect Associate (or Professional) certification, right?

You should at least be aware that AWS slays the application failure dragon using autoscaling and load balancing:

- Autoscaling can be configured to replace or replicate a resource to ensure that a predefined service level is maintained regardless of changes in user demand or the availability of existing resources.
- Load balancing orchestrates the use of multiple parallel resources to direct user requests to the server resource that's best able to provide a successful experience. A common use case for load balancing is to coordinate the use of primary and (remote) backup resources to cover for a failure.

AWS Global Infrastructure: Edge Locations

The final major piece of the AWS infrastructure puzzle is its network of edge locations. An *edge location* is a site where AWS deploys physical server infrastructure to provide low-latency user access to Amazon-based data.

That definition is correct, but it does sound suspiciously like the way you'd define any other AWS data center, doesn't it? The important difference is that your garden-variety data centers are designed to offer the full range of AWS services, including the complete set of EC2 instance types and the networking infrastructure customers would need to shape their compute environments. Edge locations, on the other hand, are much more focused on a smaller set of roles and will therefore stock a much narrower set of hardware.

So, what actually happens at those edge locations? You can think of them as a front-line resource for directing the kind of network traffic that can most benefit from speed.

Edge Locations and CloudFront

Perhaps the best-known tenant of edge locations is CloudFront, Amazon's CDN service. How does that work? Let's say you're hosting large media files in S3 buckets. If users would have to retrieve their files directly from the bucket each time they were requested, delivery—especially to end users living continents away from the bucket location—would be relatively slow. But if you could store cached copies of the most popular files on servers located geographically close to your users, then they wouldn't have to wait for the original file to be retrieved but could be enjoying the cached copy in a fraction of the time.

Not all content types are good candidates for CloudFront caching. Files that are frequently updated or that are accessed by end users only once in a while would probably not justify the expense of saving to cache.

In addition to CloudFront, there are other AWS services that make use of edge locations. Here are few examples:

Amazon Route 53 Amazon's Domain Name System (DNS) administration tool for managing domain name registration and traffic routing

AWS Shield A managed service for countering the threat of distributed denial-of-service (DDoS) attacks against your AWS-based infrastructure

AWS Web Application Firewall (WAF) A managed service for protecting web applications from web-based threats

Lambda@Edge A tool designed to use the serverless power of Lambda to customize CloudFront behavior

So all this talk will make more sense, we suggest you explore the configuration process for a CloudFront distribution by following Exercise 4.2.

EXERCISE 4.2

Take a Quick Look at the Way CloudFront Distributions Are Configured

1. Choose the CloudFront link from the AWS services menu, choose Create Distribution, and then choose the Get Started button underneath the Web Distribution section of the "Select a delivery method for your content" page.
2. Once you're on the Create Distribution page, choose inside the box next to the Origin Domain Name item. If you have any content (like a publicly available S3 bucket), it should appear as an option. Either way, since you're not planning to actually launch anything right now, it doesn't matter what—if any—value you enter.
3. Scroll down the page and note, for instance, the Distribution Settings section. Choose the Price Class drop-down arrow to see the available options for the scope of your distribution (the number of edge locations where your content will be saved).

4. Note the Distribution State settings at the bottom of the page.
5. When you're done exploring, choose the Cancel link at the bottom to ensure you don't accidentally launch a distribution for no purpose.

Regional Edge Cache Locations

In addition to the fleet of regular edge locations, Amazon has further enhanced CloudFront functionality by adding what it calls a *regional edge cache*. The idea is that CloudFront-served objects are maintained in edge location caches only as long as there's a steady flow of requests. Once the rate of new requests drops off, an object will be deleted from the cache, and future requests will need to travel all the way back to the origin server (like an S3 bucket).

Regional edge cache locations—of which there are currently nine worldwide—can offer a compromise solution. Objects rejected by edge locations can be moved to the regional edge caches. There aren't as many such locations worldwide, so the response times for many user requests won't be as fast, but that'll still probably be better than having to go all the way back to the origin. By design, regional edge cache locations are more capable of handling less-popular content.

The AWS Shared Responsibility Model

The AWS cloud—like any large and complex environment—is built on top of a stack of rules and assumptions. The success of your AWS projects will largely depend on how well you understand those rules and assumptions and on how fully you adopt the practices that they represent. The AWS Shared Responsibility Model is a helpful articulation of those rules and assumptions, and it's worth spending some time thinking about it.

Amazon distinguishes between the security and reliability *of* the cloud, which is its responsibility, and the security and reliability of what's *in* the cloud, which is up to you, the customer.

The cloud itself consists of the physical buildings, servers, and networking hardware used by AWS data centers. AWS is responsible for making sure that its locations are secure, reliably powered, and properly maintained. AWS is also on the hook for patching, encrypting (where relevant), and maintaining the operating systems and virtualization software running its physical servers and for the software running its managed services.

But that's where things can get a bit complicated. What exactly is "managed" and what's "unmanaged"? Figure 4.3 (which you saw previously in Chapter 1, "The Cloud") compares the "of the cloud/in the cloud" mix as it applies across the three key cloud models: Infrastructure as a Service, Platform as a Service, and Software as a Service.

FIGURE 4.3 A general comparison between local and managed deployments

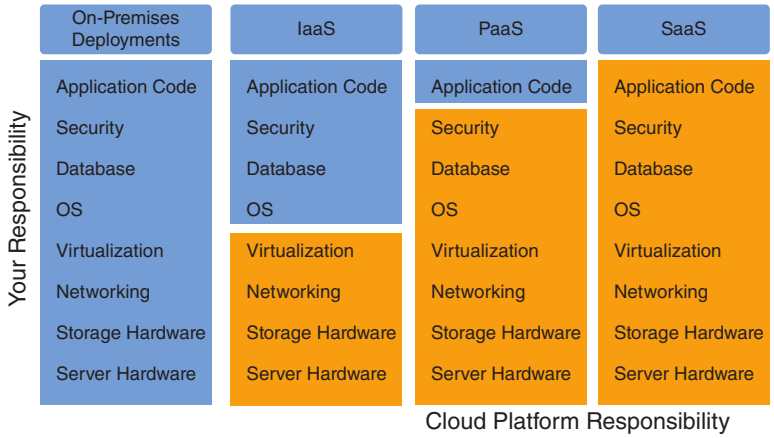
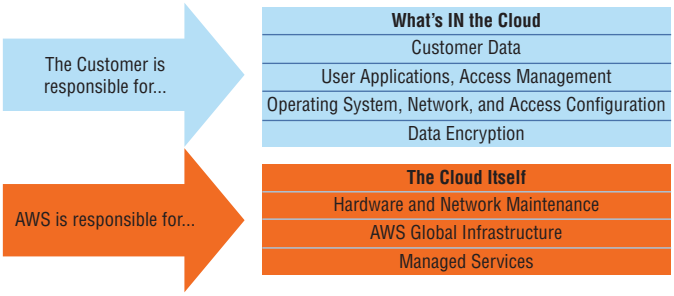


Figure 4.4 illustrates the way responsibility for the integrity and security of AWS infrastructure is divided between Amazon and its customers.

FIGURE 4.4 A representation of the AWS Shared Responsibility Model



The AWS Shared Responsibility Model

Managed Resources

A managed cloud service will "hide" all or some of the underlying configuration and administration work needed to keep things running, leaving you free to focus on the "business" end of your project. For example, an application running on an EC2 instance might need a database in the backend. You could install and configure a MySQL database engine on the instance itself, but you'd be responsible for patches, updates, and all the regular care and feeding (not to mention letting it out for its morning walks).

Alternatively, you could point your application to a stand-alone database you launch on Amazon's Relational Database Service (RDS). AWS is responsible for patching an RDS database and ensuring its data is secure and reliable. You only need to worry about populating the database and connecting it to your application.

RDS, therefore, is a good example of a partially managed service. How would the next level of managed service work? Look no further than Elastic Beanstalk, which hides just

about all the complexity of its runtime environment, leaving nothing for you to do beyond uploading your application code. Beanstalk handles the instances, storage, databases, and networking headaches—including ongoing patches and administration—invisibly.

Unmanaged Resources

The most obvious example of an unmanaged AWS service is EC2. When you launch an EC2 instance, you're expected to care for the operating system and everything that's running on it exactly the way you would for a physical server in your on-premises data center. Still, even EC2 can't be said to be entirely unmanaged since the integrity of the physical server that hosts it is, of course, the responsibility of AWS.

Think of it as a sliding scale rather than a simple on-off switch. Some cloud operations will demand greater involvement from you and your administration team, and some will demand less. Use this simple rule of thumb: *if you can edit it, you own it*. The key—especially during a project's planning stages—is to be aware of your responsibilities and to always make security a critical priority.

Service Health Status

As part of its end of the bargain, AWS makes regularly updated, region-by-region reports on the status of its services publicly available. Any service outages that could affect the performance of anyone's workload will appear on Amazon's Service Health Dashboard (`https://status.aws.amazon.com`)—often within a minute or two of the outage hitting.

While configuration errors are always a possible cause of a failure in your infrastructure, you should always make the Service Health Dashboard one of your first stops whenever you dive into a troubleshooting session.

AWS Acceptable Use Policy

Because they're so easy to scale up, cloud computing services are powerful tools for accomplishing things no one had even dreamed of just a decade ago. But for that same reason, they're also potential weapons that can be used to commit devastating crimes.

The AWS Acceptable Use Policy (`https://aws.amazon.com/aup`) makes it abundantly clear that it does not permit the use of its infrastructure in any illegal, harmful, or offensive way. Amazon reserves the right to suspend or even terminate your use of its services should you engage in illegal, insecure, or abusive activities (including the sending of spam and related mass mailings). Even running penetration testing operations against your own AWS infrastructure can cause you trouble if you don't get explicit permission from Amazon in advance.

You should take the time to read the document and keep its terms in mind as you deploy on AWS.

Summary

An AWS Region connects at least two Availability Zones located within a single geographic area into a low-latency network. Because of the default isolation of their underlying hardware, building secure, access-controlled regional environments is eminently possible.

An Availability Zone is a group of one or more independent (and fault-protected) data centers located within a single geographic region.

It's important to be aware of the region that's currently selected by your interface (either the AWS Management Console or a command-line terminal), as any operations you execute will launch specifically within the context of that region.

The design structure of Amazon's global system of regions allows you to build your infrastructure in ways that provide the best possible user experience while meeting your security and regulatory needs.

AWS offers some global resources whose use isn't restricted to any one region. Those include IAM, CloudFront, and S3.

You can connect to AWS service instances using their endpoint addresses, which will (generally) incorporate the host region's designation.

EC2 virtual machine instances are launched with an IP address issued from a network subnet that's associated with a single Availability Zone.

The principle of high availability can be used to make your infrastructure more resilient and reliable by launching parallel redundant instances in multiple Availability Zones.

AWS edge locations are globally distributed data servers that can store cached copies of AWS-based data from which—on behalf of the CloudFront service—they can be efficiently served to end users.

The elements of the AWS platform that you're expected to secure and maintain and those whose administration is managed by Amazon are defined by the AWS Shared Responsibility Model.

Exam Essentials

Understand the importance of resource isolation for cloud deployments. Properly placing your cloud resources within the right region and Availability Zone—along with carefully setting appropriate access controls—can improve both application security and performance.

Understand the role of autoscaling in a highly available deployment. The scalability of AWS resources means you can automate the process of increasing or decreasing the scale of a deployment based on need. This can automate application recovery after a crash.

Understand the role of load balancing in a highly available deployment. The ability to automatically redirect incoming requests away from a nonfunctioning instance and to a backup replacement is managed by a load balancer.

Understand the principles of the AWS Shared Responsibility Model. AWS handles security and administration for its underlying physical infrastructure and for the full stack of all its managed services, while customers are responsible for everything else.

Understand the principles of the AWS Acceptable Use Policy. Using AWS resources to commit crimes or launch attacks against any individual or organization will result in account suspension or termination.

CloudFront

Amazon CloudFront is a content delivery network (CDN) that helps deliver static and dynamic web content to users faster than just serving it out of an AWS Region. For example, if you're hosting a website from a single AWS Region, as a general rule, the farther a user is away from that region, the more network latency they'll encounter when accessing it. CloudFront solves this problem by caching your content in a number of data centers called *edge locations*. There are more than 150 edge locations spread out around the world on six continents.

CloudFront works by sending users to the edge location that will give them the best performance. Typically, this is the edge location that's physically closest to them. CloudFront also increases the availability of your content because copies of it are stored in multiple edge locations.

The more edge locations you use, the more redundancy you have and the better performance you can expect. As you might expect, the price of CloudFront increases as you utilize more edge locations. You can't select individual edge locations. Rather, you must choose from the following three options:

- United States, Canada, and Europe
- United States, Canada, Europe, Asia, and Africa
- All edge locations

To make your content available via CloudFront, you must create a distribution. A distribution defines the type of content you want CloudFront to cache, as well as the content's origin—where CloudFront should retrieve the content from. There are two types of distributions: Web and Real-Time Messaging Protocol (RTMP).

Web A Web distribution is the most common type. It's used for static and dynamic content such as web pages, graphic files, and live or on-demand streaming video. Users can access Web distributions via HTTP or HTTPS. When creating a Web distribution, you must specify an origin to act as the authoritative source for your content. An origin can be a web server or a public S3 bucket. You can't use nonpublic S3 buckets.

RTMP The Real-Time Messaging Protocol (RTMP) delivers streaming video or audio content to end users. To set up an RTMP distribution, you must provide both a media player and media files to stream, and these must be stored in S3 buckets.

Summary

Virtual Private Cloud (VPC) provides the virtual network infrastructure for many AWS resources, most notably EC2. VPCs can connect to other networks, including the following:

- The internet via an internet gateway
- External, private networks via Direct Connect or a virtual private network (VPN)
- Other VPCs using VPC peering

The Route 53 service provides two distinct but related Domain Name System (DNS) services. Route 53 functions as a registrar for many top-level internet domain names (TLDs). You can register a new domain with Route 53 or transfer an existing one that you control. Route 53 also provides DNS hosting services. To use Route 53 with a public domain, you must create a public hosted zone. To use Route 53 for name resolution within a VPC, you must create a private hosted zone.

CloudFront is Amazon's content delivery network (CDN). It improves delivery of data to end users by storing content in edge locations around the world. When a user connects to a CloudFront distribution to retrieve content, CloudFront serves the content from the edge location that will give them the best performance.

Exam Essentials

Know the components of a VPC. The key components of a VPC include at least one subnet, security groups, network access control lists (NACLs), and internet gateways.

Understand the different options for connecting to resources in a VPC. You can connect to resources in a VPC over the internet, a Direct Connect link, a VPC peering connection, or a virtual private network (VPN) connection.

Understand the difference between a Route 53 public hosted zone and a private hosted zone. A public hosted zone allows anyone on the internet to resolve records for the associated domain name. A private hosted zone allows resolution only from resources within the associated VPCs.

Be able to select the best Route 53 routing policy for a given scenario. All routing policies except the Simple routing policy can use health checks to route around failures. If you want to direct traffic to any available resource, Failover, Weighted, and Multivalue Answer routing policies will suffice. If performance is a concern, choose a Latency routing policy. If you need to direct users based on their specific location, use a Geolocation routing policy.

Know how CloudFront improves the speed of content delivery. CloudFront caches objects in edge locations around the world and automatically directs users to the edge location that will give them the best performance at any given time.

Be able to identify scenarios where CloudFront would be appropriate. CloudFront is designed to give users the fastest possible access to content regardless of their physical location. By caching content in edge locations that are distributed around the world, CloudFront helps ensure that your content is always close to your users.