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

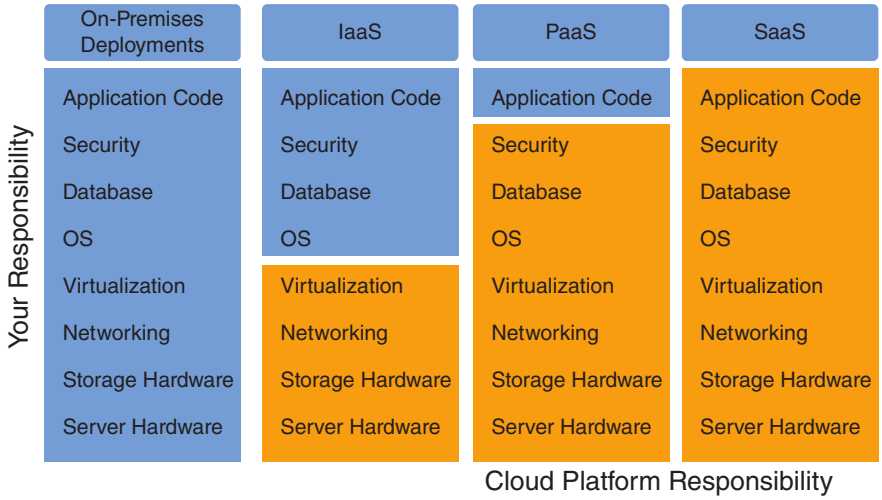
BEN PIPER
DAVID CLINTON

SYBEX
A Wiley Brand

While some may disagree with the designation, AWS SaaS products arguably include Simple Email Service and Amazon WorkSpaces.

Figure 1.2 compares the scope of responsibility you have on IaaS, PaaS, and SaaS platforms with the way it works for on-premises deployments.

FIGURE 1.2 The breakdown of responsibility across multiple infrastructure types



Serverless Workloads

Besides doing an excellent job emulating traditional server behavior, cloud providers can also enable entirely new ways to administrate applications and data. Perhaps the most obvious example is serverless computing.

Now don't be fooled by the name. You can't run a compute function without a computer environment (a "server") somewhere that'll host it. What "serverless" does allow is for individual developers to run their code for seconds or minutes at a time on some else's cloud servers.

The serverless model—as provided by services like AWS Lambda—makes it possible to design code that *reacts* to external events. When, for instance, a video file is uploaded to a repository (like an AWS S3 bucket or even an on-premises FTP site), it can trigger a Lambda function that will convert the file to a new video format. There's no need to maintain and pay for an actual instance running 24/7, just for the moments your code is actually running. And there's no administration overhead to worry about.

according to demand. Should, say, your WordPress site go viral and attract millions of viewers one day, AWS will invisibly ramp up the infrastructure to meet demand. As demand falls, your infrastructure will similarly drop. Keep this in mind, as such variations in demand will determine how much you'll be billed each month.

Deploying Container and Serverless Workloads

Even virtualized servers like EC2 instances tend to be resource-hungry. They do, after all, act like discrete, stand-alone machines running on top of a full-stack operating system. That means that having 5 or 10 of those virtual servers on a single physical host involves some serious duplication because each one will require its own OS kernel and device drivers.

Containers

Container technologies such as Docker avoid a lot of that overhead by allowing individual containers to share the Linux kernel with the physical host. They're also able to share common elements (called *layers*) with other containers running on a single host. This makes Docker containers fast to load and execute and also lets you pack many more container workloads on a single hardware platform.

You're always free to fire up one or more EC2 instances, install Docker, and use them to run as many containers as you'd like. But keeping all the bits and pieces talking to each other can get complicated. Instead, you can use either *Amazon Elastic Container Service* (ECS) or *Amazon Elastic Container Service for Kubernetes* (EKS) to orchestrate swarms of Docker containers on AWS using EC2 resources. Both of those services manage the underlying infrastructure for you, allowing you to ignore the messy details and concentrate on administrating Docker itself.

What's the difference between ECS and EKS? Broadly speaking, they both have the same big-picture goals. But EKS gets there by using the popular open source Kubernetes orchestration tool. They are different paths to the same place.

Serverless Functions

The serverless computing model uses a resource footprint that's even smaller than the one left by containers. Not only do *serverless functions* not require their own OS kernel, but they tend to spring into existence, perform some task, and then just as quickly die within minutes, if not seconds.

On the surface, Amazon's serverless service—AWS Lambda—looks a bit like Elastic Beanstalk. You define your function by setting a runtime environment (like Node.js, .NET, or Python) and uploading the code you want the function to run. But, unlike Beanstalk,

Lambda functions run only when triggered by a preset event. It could be a call from your mobile application, a change to a separate AWS resources (like an S3 bucket), or a log-based alert.

If an hour or a week passes without a trigger, Lambda won't launch a function (and you won't be billed anything). If there are a thousand concurrent executions, Lambda will scale automatically to meet the demand. Lambda functions are short-lived: they'll time out after 15 minutes.

Summary

Configuring EC2 instances is designed to mirror the process of provisioning and launching on-premises servers. Instances are defined by your choice of AMIs, instance type, storage volumes, and pricing model.

AMIs are organized into four categories: Quick Start, custom (My AMIs), AWS Marketplace, and Community. You can create your own AMI from a snapshot based on the EBS volume of an EC2 instance.

EC2 instance types are designed to fit specific application demands, and individual optimizations are generally available in varying sizes.

EBS storage volumes can be encrypted and are more like physical hard drives in the flexibility of their usage. Instance store volumes are located on the same physical server hosting your instance and will, therefore, deliver faster performance.

EC2 on-demand pricing is best for short-term workloads that can't be interrupted. Longer-term workloads—like ecommerce websites—will often be much less expensive when purchased as reserved instances. Spot instances work well for compute-intensive data operations that can survive unexpected shutdowns.

Lightsail, Elastic Beanstalk, Elastic Container Service, Elastic Container Service for Kubernetes, and Lambda are all designed to provide abstracted compute services that simplify, automate, and reduce the cost of compute operations.

Exam Essentials

Understand the elements required to provision an EC2 instance. An instance requires a base OS (AMI) and—optionally—an application stack, an instance type for its hardware profile, and either an EBS or an instance volume for storage.

Understand the sources, pricing, and availability of EC2 AMIs. The Quick Start and Marketplace AMIs are supported by Amazon or a recognized third-party vendor, which may not be true of AMIs selected from the Community collection. In any case, you should confirm whether using a particular AMI will incur extra charges beyond the normal EC2 usage.