LEARNING AMAZON WEB SERVICES (AWS)

A Hands-On Guide to the Fundamentals of AWS Cloud

MARK WILKINS

Learning Amazon Web Services (AWS)

Learning Amazon Web Services (AWS)

A Hands-On Guide to the Fundamentals of AWS Cloud

Mark Wilkins

Addison-Wesley

ISBN-13: 978-0-13-529834-3

ISBN-10: 0-13-529834-2

Library of Congress Control Number: 2019937606

ScoutAutomatedPrintCode

Acquisition Editor
Paul Carlstroem

Managing Editor
Sandra Schroeder

Development Editor
Kiran Panigrahi

Project Editor
Lori Lyons

Production Manager
Aswini Kumar

Copy Editor
Kitty Wilson

Indexer
Cheryl Lenser

Proofreader
Abigail Manheim

Designer
Chuti Prasertsith

Compositor
codeMantra

Contents at a Glance

Table of Contents

Companion Videos List

In addition to this book, several hours of companion online training videos are available. Throughout the chapters, you'll be invited to watch a video that relates to the topic being covered in that section.

To access the videos, register this book at www.informit.com/register.

Chapter 1: Learning AWS

- Signing up for Amazon Free Tier
- Terra Firma

Chapter 2: Designing with AWS Global Services

- Availability Zones
- Choosing a Region
- Planning Compliance
- Trusted Advisor
- Using the Simple Monthly Calculator

Chapter 3: AWS Networking Services

- Create a Custom VPC
- Creating CIDR Blocks
- Public and Private Subnets
- Exploring Route Tables
- Creating Security Groups
- Network ACLs
- VPC Flow Logs
- Understanding Endpoints
- Adding an Internet Gateway
- Creating VPN Connections

Chapter 4: Compute Services: AWS EC2 Instances

- Creating a Custom AMI
- Creating Reserved Instances
- Spot Instances
- Creating Instances

Chapter 5: Planning for Scale and Resiliency

- Installing the CloudWatch Agent
- CloudWatch in Operation
- Creating a CloudWatch Alarm

Preface

Although the Amazon cloud is well-documented, the Internet includes all types of information. This means you can spend a great deal of time reading AWS technical documentation, only to find that what seems interesting might be five years old or more. There's too much documentation to expect to spend just a couple of evenings researching and getting right up to speed.

My opportunity to create this technical book for understanding AWS began in April 2018 after Mark Taber, an acquisitions editor for Pearson Education, pinged me on LinkedIn. I had written technical books before, and Mark asked if I was interested in writing one on the topic of Amazon Web Services. I asked, "Do people actually buy paper books?" and he replied quickly, "They sure do."

So, I thought about it and realized that most of the customers I had consulted with over the past few years regarding the AWS cloud were smart technical people, but they had been thrown into a bit of a panic because they had to get ready for moving to the cloud—specifically, the Amazon cloud. And they were looking for a starting point to ramp up their technical cloud knowledge and become technically proficient in what was happening in AWS cloud technologies.

I had spent a few years quite involved with AWS cloud services with various clients—including a major Canadian bank, a major American bank, and several small-to-midsize companies working in AWS—because their developers had developed applications they were using quite successfully. The only problem was, they weren't in the AWS cloud.

I thought about all my customers and realized that what was missing was a foundational book on AWS that explained how the core AWS services of compute, storage, networking, scale, security, and automation fit together. I decided to combine a book with a number of videos that would walk through how to set up each service. This approach would allow my customers, and hopefully many others, to visualize how AWS could work for their company or their project.

Writing a technical book is ultimately an abundance of research and rounds of testing, breaking, and fixing until the project comes together. To create a detailed technical overview of Amazon Web Services and how its cloud services fit together, I decided to review all the relevant AWS documentation of the compute, storage, networking, and managed services by following the pattern of reading and testing; then even more reading and testing. I then added some tips and tricks, and finally summarized this last year's work into the technical content found in the chapters of this book. I learned a lot about AWS that I didn't know—that's the great thing about researching and writing a book!

Companion Training Videos

Learning Amazon Web Services (AWS) also has a useful learning companion—several hours of training videos are bundled with the book that will show you how easy it is to set up the core services at AWS and grasp the concepts of what the AWS cloud can offer.

Throughout the chapters, you'll be invited to watch the companion video that relates to the topic that is being covered in a particular section.

Watching the videos will help you get in technical shape to start deploying your company's applications and resources at AWS. The videos take the place of page after page of step-by-step instructions. This reason for no detailed steps is that in the AWS cloud, the steps to perform any task are constantly changing, so up-to date videos as a means of teaching makes more sense. Videos can also be updated easily as changes occur.

The videos can be accessed by registering your copy of this book at www.informit.com/register. The videos can be watched on most any device as they are formatted in a standard MP4 video format. And, don't forget popcorn!

About the Author

Mark Wilkins is an Electronic Engineering Technologist with a wealth of experience in designing, deploying, and supporting software and hardware technology in the corporate and small business world. Since 2013, Mark has focused on supporting and designing cloud service solutions with Amazon Web Services, Microsoft Azure, and the IBM Cloud. He is certified in Amazon Web Services (Architecture and Sys-Ops). Mark is also a Microsoft Certified Trainer (MCT) and holds certifications in MCTS, MCSA, Server Virtualization with Windows Server Hyper-V, and Azure Cloud Services.

Mark worked as a technical evangelist for IBM SoftLayer from 2013 through 2016 and taught both SoftLayer Fundamentals and SoftLayer Design classes to many Fortune 500 companies in Canada, the United States, Europe, and Australia. As course director for Global Knowledge, Mark developed and taught many technical seminars, including Configuring Active Directory Services, Configuring Group Policy, and Cloud and Virtualization Essentials. Mark also developed courseware for the Microsoft Official Curriculum 2008 stream, Managing and Maintaining Windows Server 2008 Network Services, and Active Directory Services.

Mark's published books include *Windows 2003 Registry for Dummies*, *Administering SMS 3.0*, and *Administering Active Directory*.

Acknowledgments

A book is not written by a single person; many help along the way. I'd like to thank Ashley Neace for giving me the opportunity to develop courseware for Global Knowledge way back in 2010 about the AWS cloud, and Rick Morrow, Mark Sluga, and Ryan Dymek for providing their expertise and knowledge over the years working together at Global Knowledge and as valuable technical resources. Thanks also to my editors Paul Carlstroem, Kiran Panigrahi, and Mark Taber for providing support and guidance for this project.

1 Learning AWS

About This Book

This paper book and companion video library are focused on the Amazon Web Services (AWS) cloud—and specifically what is called infrastructure as a service (IaaS)—to help you learn about the cloud services Amazon offers. Services that AWS offers can be broken down into the foundational services of compute, storage, networking, and security—and a big helping of automation. A handy way to think of AWS is as a massive toolbox with a wide variety of specialized tools that can carry out an assortment of infrastructure tasks. If you're a system administrator, developer, or project manager or you've heard about the AWS cloud and want to know more about it, this book is designed for you as a technical baseline of AWS services, what they can do, the major concepts, one of the major components, and how to set up the service to function. I estimate that I reviewed more than 35,000 pages of AWS documentation and summarized all that technical detail into somewhere between 300–400 pages of AWS information. That doesn't mean you won't read AWS documentation because you most definitely will; but hopefully this book and the companion video library will catapult your indoctrination into the AWS jungle.

You may also want to get certified; however, this is not a book that is directly focused on AWS certification. This book is instead focused on the so-called foundational services. All AWS certification tests are focused on problem-solving based on a particular scenario. Your job is to figure out the best one or two answers; therefore, knowing the foundational services is key. If you want to get certified on AWS cloud services, particularly on AWS architecture, you must know the foundational AWS services inside and out. And you'll have to spend a few hours doing hands-on work with AWS services. If you want to develop applications that will be hosted at AWS, you will need to know the foundational services in even more detail. And forget about learning everything about AWS in a single book; it's just not possible, and the reality is that AWS is constantly changing. That's a notion you will learn to embrace.

Each chapter in this book attempts to deal with a specific concept or AWS service and provide a strong detailed technical summary of the AWS service in question. However, there are not pages and pages of step-by-step solutions because the steps change every couple of months. During the writing of this book, AWS changed the design of its icons used in its technical documentation three times. They also added 600 features and made numerous other changes, from cosmetic to substantial.

To get around the issue of immediate obsolescence, there is a companion video library associated with this book that shows you how to set up and install and configure many AWS cloud services. You can access these videos by registering your book at informit.com/register.

Throughout the remainder of the chapters, you'll be invited to watch the companion video that relates to the topic that we are covering. The companion step-by-step videos can be changed and updated or added to as AWS changes. The beauty of a video is that you can pause or rewind it as you learn. Let's begin the journey and see where we end up. This initial chapter includes the following topics:

- Defining the public cloud
- Where AWS fits with IaaS and platform as a service (PaaS)
- Characteristics of cloud computing according to NIST
- Considerations for migrating applications to AWS
- Operational benefits for operating in the cloud
- The cloud service-level agreement (SLA)
- Data, application, and network security at AWS
- Compliance at AWS
- AWS Well-Architected Framework

Trying to Define the Cloud

The roots of public cloud computing are not new; the public cloud providers Amazon Web Services and Microsoft Azure have been established for well over a decade with strong IaaS and PaaS offerings around the world. The Google Cloud Platform (GCP) and the IBM or Oracle Cloud are other viable alternatives. Gartner's Magic Quadrant (www.gartner.com/en/research/methodologies/magic-quadrants-research) in Figure 1-1 shows four types of technology provider a company can align their goals and strategies with. In 2018, IaaS market penetration dominated two of those categories. Under the Leaders quadrant, Amazon Web Services led in that area, followed by Microsoft and then Google. Google also aligned closely to the Visionaries Quadrant. Alibaba Cloud, Oracle, and IBM fell in the Niche Players quadrant.

When I started my career as a computer technician back in the 90s, most corporations that I supported used several computer-based services that were not located on premise. Accounting services were accessed through a fast (at the time) 1200 baud modem that was connected using one of those green-screened digital terminals. The serial cable threaded through the drop ceiling to connect the terminal was strong enough to pull a car.

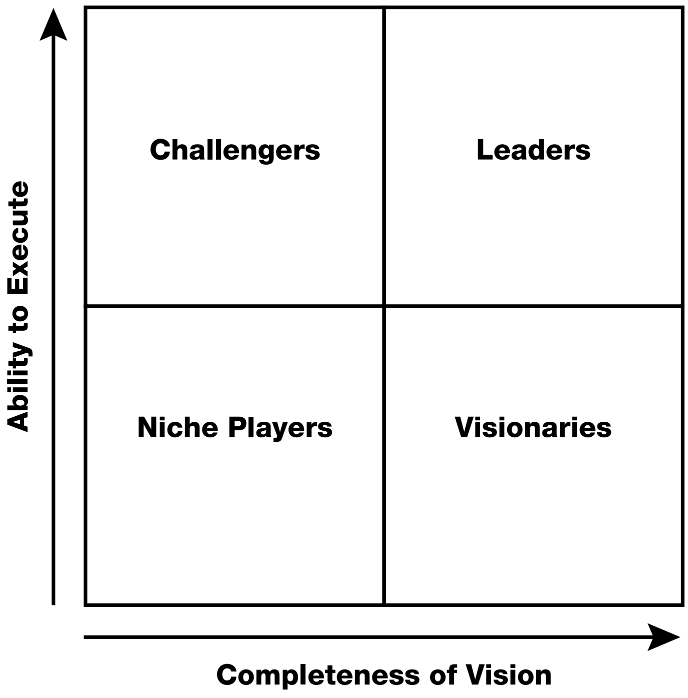


Figure 1-1 Top public cloud providers. Gartner, Magic Quadrant for Cloud Infrastructure as a Service, Worldwide, Dennis Smith et al., 23 May 2018. (Gartner Methodologies, Magic Quadrant, www.gartner.com/en/research/methodologies/magic-quadrants-research)[1]

A customer of mine at the time was utilizing a mainframe computer for accounting hosted locally in town. However, he couldn't access his accounting services any time he liked; he had his allotted slice of processing time every Tuesday, and that was that. Payroll services were provided by another remote service called Automatic Data Processing, or ADP for short. Both service companies and their services are still around today. IBM is continuing to release versions of its z series mainframe, and ADP payroll services was one of the first software as a service (SaaS) companies but remains popular today.

In 2015, IBM bought a cloud provider based in Texas called SoftLayer and merged it into its public cloud offering, today called the IBM Cloud. The z mainframe has ended up being hosted in the IBM cloud providing hosted mainframe services; in April 2018, IBM announced it was launching what it called a "skinny mainframe" for cloud computing built around the IBM z 14 mainframe.

[1]Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

If you work for a bank or financial institution, IBM mainframes probably provide 50% of all your computing services. This could be great news for companies that don't want to have a local mainframe environment to maintain.

Fifty years since the launch of the IBM mainframe, many companies' mainframes are continuing to be relevant and are now part of the public cloud landscape.

The reality is that more than 90 of the world's largest 100 banks, the top 10 insurance companies, a majority of the 25 largest retailers, and most of the world's larger airlines still rely on mainframe computers from IBM.

If you didn't use mainframes, you probably lived through the deployment cycle of Novell NetWare and Windows and Active Directory, and virtualization using VMware or Hyper-V. You likely have a private cloud in your own data centers. You may be wondering why your company is moving to the public cloud.

The reality these days is that it is expensive to build and maintain data centers. Certainly, building a data center is going to cost millions or billions of dollars. Maintaining an existing data center over the long term is expensive as well. Because of virtualization and the rise of the Internet as a useful communication medium, cloud services have replaced many local data centers and will continue to do so. Figuring out the capital costs of hosting your applications in the public cloud instead of running them in your own data center is sometimes categorized as renting instead of buying, as defined in Figure 1-2.

Operational expenses (OpEX) are all you pay for using cloud services. The capital expenditure (CapEX) of building a data center does not have to be borne by a single business. Now let's be clear: operational expenses are still expensive. You might say to your boss, "I don't need $800 million for data center construction, but I will need $2 million a year forever."

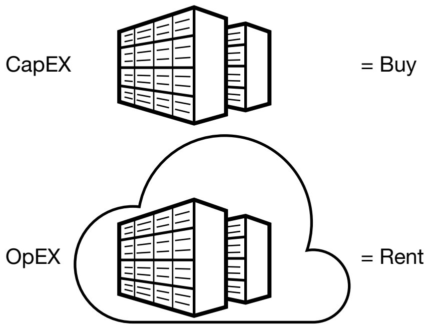


Figure 1-2 No long-term capital expenses

The reality is that the cost of running and hosting your applications in the cloud is cheaper once you add in every expense; however, operating in the cloud is only cheaper if your services being hosted in the cloud are properly designed. Services and applications don't run 24/7; they are turned off or reduced in size when they're not needed. A concept that you may not yet be familiar with is automation. Public cloud providers use automated procedures to build, manage, monitor, and scale every cloud service. By the end of this book, you will understand how automation is the secret sauce for successful cloud deployments. Automated procedures will save you money and allow you to sleep at night.

Let's start by defining the public cloud. The cloud is just a collection of data centers. There is no ownership from the customer's point of view; the cloud provider owns the services, and you rent each service as required. You may be thinking that the cloud is all virtual resources, yet the AWS cloud *can* provide you bare-metal servers. If you want, Amazon will happily host your applications and databases on bare-metal servers hosted in its data centers. Of course, more commonly, AWS will offer you many virtual servers in well over 150 different sizes and designs. Amazon is also quite happy to allow you to continue to operate your on-premise data centers and coexist with cloud resources and services operating at AWS. Microsoft Azure will offer to sell you a copy of its complete Azure cloud operating system to install on your servers in your data centers. As you can see, it's hard to define the public cloud these days other than as a massive collection of compute and storage resources hosted on a network stored in the collection of data centers accessible across the Internet, or by using private connections.

Anything that you host in the public cloud is using compute and storage resources to execute your software application. And anything that used to be a hardware device, such as a router, switch, or storage array, can be replaced by a third-party software appliance or an AWS-managed software service composed of virtual computers, storage, and networking components. This doesn't mean that many companies aren't still using hardware devices. Hardware devices such as routers and switches have incredible speed and can operate much faster in most cases than a software router and switch. But what happens if you can run hundreds or thousands of virtual machines in parallel performing the function of a hardware switch or hardware router device? Perhaps we don't need any hardware devices at all. Most of the AWS-managed cloud services are hosted on virtual machines (defined as EC2 instances, or Elastic Cloud Compute instances), with massive CPU and RAM resources running in massive server farms with custom-designed applications, providing the storage arrays, networking services, load-balancing, and auto-scaling services that we depend on at AWS.

Moving to AWS

Once the decision has been made to move to the AWS cloud, countless moving parts begin to churn. People need to be trained, infrastructure changes must take place, developers potentially need to code in a different way, and IT professionals must get up to speed on the cloud provider that has been chosen; there's no time to waste. Larger companies will usually attempt to convey the message of what moving to the cloud means for them. It's quite common for executives within the company to have strong opinions about what moving to the cloud will do. Sadly, these opinions are not usually based on technical knowledge or real hands-on experience with the cloud provider that has been chosen. Generally, companies utilizing cloud services fall into several mind-sets:

- **The corporate mentality**—You currently have data centers, infrastructure, and virtualized applications. Ever-increasing infrastructure and maintenance costs are driving you to look at what options are available in the public cloud.

- **Born-in-the-cloud mentality**—You're a developer with a great idea, but you don't want to maintain a local data center. In fact, you don't have a local data center, and you want to get going as soon as possible.
- **The startup mentality**—You've just lost your job due to a merger or buyout and are determined to strike out on your own. Your brand-new company has no data center but plenty of ideas combined with a distinct lack of cash.
- **The government client**—You've been told that, to save costs, your government department is moving to the AWS cloud within a defined timeframe.

Each of these starting mind-sets will have differing points of view as to how it should start to migrate or design its cloud infrastructure and hosted applications. Coming from a corporate environment or government department, you will probably expect the cloud provider to have a detailed service-level agreement (SLA) that you can change to match your needs. You will also probably have expectations about how much detail you expect to be provided about the cloud provider's infrastructure and services. In short, you expect to be in control.

If you have started with a public cloud services provider as an individual developer, or you're working with a startup, you will probably have no comparison with current on-premise costs; therefore, the overall costs that you pay for using a cloud provider will be accepted for the short term but, over time, as your experience grows, your overall cloud costs will be analyzed and managed to be as optimized and as cheap as possible.

> **Note**
>
> AWS has options for developers who want to craft and deploy applications hosted at AWS. The site https://aws.amazon.com/startups/ is where you can get further information about how you might be able to qualify for what is called AWS Promotional Credit. There's a possibility of getting up to $15,000 in credits over 2 years, including AWS support and training.

The reality is that moving to the cloud means you will be giving up an element of control. After all, it's not your data center. At AWS, you're not getting deeper into the infrastructure stack than the subnets that host your applications. Remember, the cloud is a data center; it's just not *your* data center. Let's start by looking at the available public cloud computing models of IaaS and PaaS and where AWS fits within these definitions.

Infrastructure as a Service

Most of the services AWS offers fall into the infrastructure as a service (IaaS) definition, as shown in Figure 1-3. This is certainly the most mature cloud model offering; virtualized servers and virtualized storage arrays are hosted on a software defined network with each customer's infrastructure completely isolated as a private resource. Creating resources at AWS typically starts with the creation of what is called a virtual private cloud (VPC). Virtual servers, virtual hard drive volumes, and indeed complete managed services and products can be hosted on your isolated private network. You have the flexibility to create whatever architectural stack you desire at AWS using a vast number of services and utilities contained in the IaaS toolbox. Companies moving to the AWS public cloud will typically first start with IaaS because the compute and storage services closely mirror their current on-premise virtual environment.

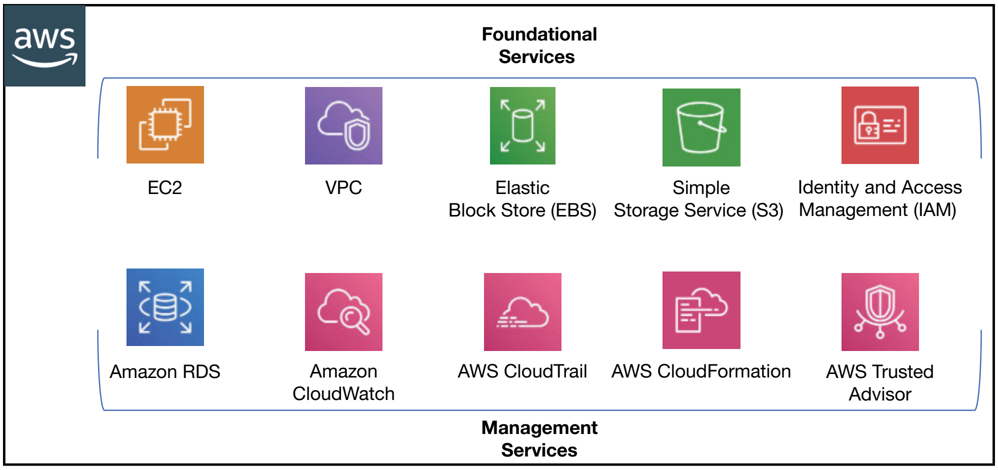


Figure 1-3 Infrastructure as a service at AWS

IaaS cloud services at AWS are bundled with managed services. A managed service is built on the trio of compute, storage, and networking services and customized software providing something you want Amazon to manage and maintain rather than your having to do all the work. For example, AWS offers a managed service called relational database service (RDS). It will build, host, maintain, back up, fail over, synchronize, and monitor a pair of master/standby database servers for you, leaving you the single task of managing your data records. Many other managed services are available at AWS; in fact, many managed services have no additional charges to begin using. For example, an automation service called CloudFormation allows you to automate the procedure of building infrastructure stacks complete with the required compute, storage, networks, and load balancers required for your application stack. In fact, practically anything to do with building, updating, or deleting your infrastructure stacks at AWS can be automated with CloudFormation. Another handy service called CloudTrail is provided free of charge. It tracks and records all application programming interface (API) calls that are carried out in each of your AWS accounts for 90 days. And yes, you can configure CloudTrail to store your API calls forever in S3 storage.

Your internal applications that are running in your on-premise data centers are probably a vast soup of proprietary operating systems (HP, AIX, Linux) and of course Windows. Talk to most departments in a small to midsize corporate environment, and the end users typically express unhappiness with some of the current applications that they use daily. They have learned to live with the ongoing issues of each application. Talk to the IT administrators and developers in the corporate data centers; there very well could be a great deal of unhappiness with the inflexibility of the existing infrastructure that they have to use and manage.

On top of these issues, perhaps each department has its own IT infrastructure. My company once provided compute services for a midsized hospital with 25 separate networks. Typically, in a larger corporation, compute services can be heavily siloed between departments, or each line of business gets to make its own decisions.

Most companies with more than 100 employees have some semblance of virtual infrastructure for their servers typically using VMware. Virtualization was supposed to be the answer to controlling a company's infrastructure costs. However, the cost for virtualization services has become extremely expensive to host, run, and maintain. Companies now know that capital and licensing costs are some of the biggest expenses they incur when running an ever-expanding on-premise private cloud. Replacing VMware with AWS-hosted virtualized servers and services removes a company's need for hypervisor administration expertise. And the landscape of applications used by corporations is now widely available in the public cloud as hosted applications defined as software as a service (SaaS) applications. As a result, there is ever-growing interest at the department level or overall company level in using the public cloud to host applications. And the reality is, you may not have a choice. If you're a Microsoft shop, the odds are quite strong that some of your everyday software applications such as Exchange and Microsoft Office are hosted by Microsoft Azure and Office 365, allowing you to completely replace some of your in-house software deployments. For more details on the compute platform at AWS, check out Chapter 4, "Compute Services: AWS EC2 Instances."

If your company has no experience working with external cloud providers and you are a medium- to large-sized corporation, it's a certainty your company will fit the private cloud model. Most of your company's infrastructure will be hosted within several private data centers. For example, your primary data center may be in Philadelphia, and your second data center could be in Nashville. (If you're a large enough company, your data centers may be spread across multiple continents.) The applications used will number in the hundreds or thousands. You may be lucky enough to have centralized IT standards, but these standards have become an issue due to the applications that multiple departments have installed or created over the years. Maybe if you're unlucky, one of the central applications used by your company was developed by a summer student and plunked into production without a second thought.

At AWS, infrastructure resources are spread across the world in 20 different regions. If you are in a large population center, the odds are that Amazon is close by. If Amazon is not close by, you still may be able to connect into it through one of the edge locations. More details on regions, availability zones, and edge locations can be found in Chapter 2, "Designing with AWS Global Services."

Platform as a Service

Platform as a service (PaaS) cloud providers enable your developers to create custom applications on a variety of popular development platforms such as Java, PHP, and Python. The developers don't have to manually build the infrastructure components required for each application per se; the required infrastructure resources are defined at the beginning of the development cycle and are created and managed by the PaaS cloud provider. After applications have been developed and tested and are ready for prime time, the application is made available to end users using public URLs. The PaaS cloud provider will host and scale the hosted application based on demand. As more users use the application, the infrastructure resources will scale out or in as required. PaaS environments are installed on the IaaS resources of the PaaS cloud provider, as shown in Figure 1-4. In fact, IaaS is always behind all "as a service" monikers. Examples of PaaS providers include Cloud Foundry and Heroku.

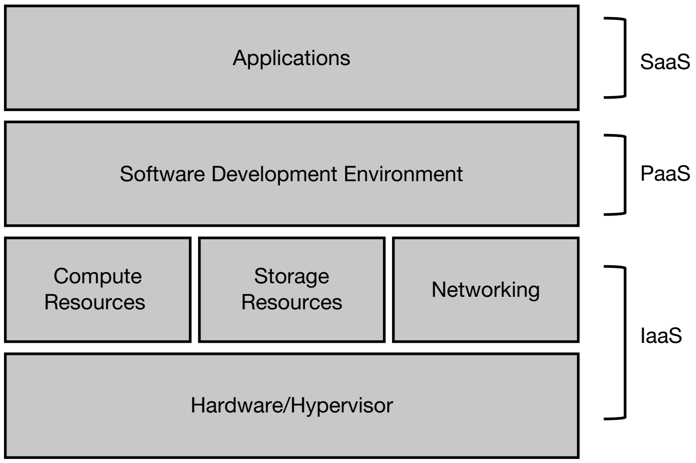


Figure 1-4 IaaS hosts the PaaS layer

Expanding upon Cloud Foundry, this PaaS solution is the foundation of development at IBM Cloud, where the underlying infrastructure is hosted on the IBM public cloud and running a customized version of the Cloud Foundry platform components. Developers can sign up and focus on writing applications. All requests will be handled by the PaaS layer interfacing with the IaaS layer, where the compute, storage, load-balancing, and scaling services operate.

Another popular solution for developing applications in the cloud is Heroku, mentioned in passing earlier. Heroku allows you to create and run hosted applications using a variety of development platforms. Just like the IBM cloud, once the application has been written, Heroku hosts, balances, and auto scales the application as required and sends you a bill for hosting at the end of the month.

If you're dealing with a PaaS provider, remember that programming languages change from time to time; therefore, APIs change as well, and usually without warning. If your developers don't keep up to date, there can be issues when using a PaaS cloud development platform.

Digging into the details on the Heroku website, under "Security," the site states that, "Heroku's physical infrastructure is hosted and managed within Amazon's secure data centers and utilize the Amazon Web services technology." Heroku is owned by another cloud heavyweight, Salesforce. Salesforce indicated in 2018 that future expansion was going to be by utilizing Amazon data center resources. Oh, what a tangled web we weave.

An additional reality is that one cloud provider's PaaS system is not necessarily compatible with another cloud provider's service. Both AWS and Microsoft Azure offer similar cloud services, but internally each cloud provider operates in a completely different fashion with a completely different set of APIs. There is no single standard for defining just what PaaS must be. Compatibility issues begin to reveal themselves at the lower levels of each vendor's proposed solution. RESTful interfaces, manifest file formats, framework configurations, external APIs, and component integration are not necessarily compatible across cloud vendors. AWS deals with platform services using Lambda, the API Gateway, and several code deployment tools.

The applications that your company may have been developing and using internally will be a variety of two- and three-tier architectures with many local dependencies such as network storage, local storage, local users, and databases. The overall architecture design may have been adequate at the beginning but now is straining to function due to the age of the hardware, the sizing of the hardware, and the lack of any flexibility to change.

The distinct difference with on-premise design when compared to hosting applications at AWS is that provisioning hardware and waiting for it to be set up and configured is a thing of the past. In fact, there are many possibilities to consider when designing applications at AWS.

Your choice of language and development framework will determine the PaaS vendor you select. Do you do a lot of development in Python? Are you a Java developer? Amazon has a PaaS solution called Elastic Beanstalk that automates the deployment of applications developed in Java, Python, Ruby, and other development platforms on the required infrastructure components for each application including E2 instances or Docker containers, with load-balancing, auto scaling, and monitoring services.

Amazon has several development solutions, shown in Figure 1-5, including CodeBuild, CodeCommit, Elastic Beanstalk, CodeDeploy. These can be key components in your application deployment at AWS. Chapter 8, "Automating AWS Infrastructure," covers these interesting managed services and additional details on automating your infrastructure.

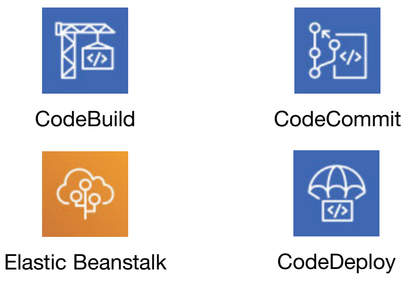


Figure 1-5 Platform options at AWS

Essential Characteristics of AWS Cloud Computing

If you haven't heard of National Institute of Standards and Technology (NIST), a branch of the U.S. government, you're not alone. Around 2010, NIST began documenting the public cloud. After talking to all the major vendors, it released an initial report in June 2011 defining many cloud components that were common across all the public cloud vendors. The report's genius was in defining what the emerging public cloud actually was (the command components). Over the years, NIST's cloud definitions have moved from definitions to becoming standards for how many companies view working in the public cloud. According to NIST, five key definitions of the public cloud have really morphed into a definitive standard methodology of operating in the public cloud:

On-demand self-service—We not only *expect* cloud service to be delivered quickly; we *demand* it. All cloud providers offer a self-serve portal as AWS does, as shown in Figure 1-6. You request a cloud service, and in seconds it's available in your AWS account ready to configure. Gone are the days of requesting a virtual server via email and waiting several days until it's built. At AWS, a virtual server can be started and operational in seconds. Procuring a software-defined network at AWS (called a virtual private cloud) is available and operational in seconds. AWS has an expansive self-serve management console that allows you to order and configure many cloud-hosted services in seconds in any AWS region. Any cloud service that you order from AWS is automatically delivered to you through heavily automated procedures. There are no public cloud providers that survive without a self-service portal driven by heavy-duty automation in the background. This NIST definition is now a standard.

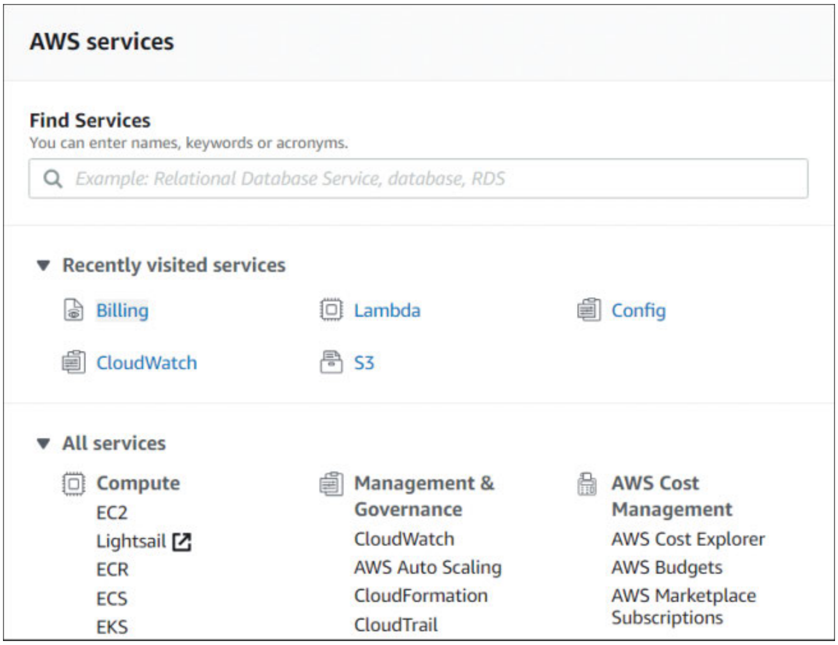


Figure 1-6 The AWS management portal

Broad network access—Cloud services can be accessed from almost anywhere across the globe using the Internet. If you host applications at AWS, perhaps they are public-facing SaaS apps. AWS also provides HTTPS endpoints to access every cloud service hosted at AWS. However, you may not want broad network access, which is defined as public network access to your cloud services. In fact, many companies that are moving to the AWS cloud have no interest in a publicly accessible software solution. They want their hosted cloud services to remain private, accessible only by their employees using private connections. Each cloud customer ultimately defines the real meaning of broad network access. At AWS, applications can be publicly available, or, you can stay completely private. VPN connections from your place of work to AWS are commonplace; in fact, you can order Direct Connect and establish a private fiber connection to AWS running at speeds up to 10 Gbps. Depending on the type of applications you're using in the cloud, high-speed network access is essential. We can even use, access, and administer AWS service from our phone using AWS apps. Certainly, accessing AWS from any device is possible. For more details on networking, check out Chapter 3, "AWS Networking Services."

Resource Pooling—Infrastructure resources for public cloud providers are pooled together in many data centers across the different regions of the world and are dynamically assigned on demand. A company running an on-premise private cloud would pool its virtual machines, memory, processing, and networking capabilities into one or two data centers, and from its own pool offer limited compute resources. All public cloud providers have a massive pool of resources to serve our various needs. AWS has clusters of data centers (known as AZs or availability zones), and each AZ could have over 80,000 bare-metal servers available and online allowing customers to host their application services with a high level of resiliency and failover. Having many available online resources also enables AWS to keep the price down. Without a massive pool of resources, AWS would not be able to offer its cloud services on demand that are able to scale up and down based on customer demand. Having a massive resource pool is a necessary standard for all public cloud providers; customers do not expect to run out of resources. Take, for example, AWS S3 storage, which is unlimited with no defined maximum limit. For more details on regions and AZs, check out Chapter 2.

Rapid Elasticity—Elasticity in the public cloud, or scaling, is *the* key feature required by all hosted cloud applications. Elasticity at AWS is utilized for both compute and storage. Because most services and applications are built on compute and storage, applications in the AWS cloud have the capability to automatically scale, as shown in Figure 1-7. And elasticity, or scaling, is only useful if it's automated based on demand. Turning off a virtual server, adding RAM, and turning it back on is not the elasticity that we are interested in; we want horizontal scale—that is, more application servers—not just a bigger server. Real-time monitoring of a hosted cloud application at AWS allows us to react almost instantaneously before the application's performance is close to degrading. With EC2 Auto Scaling in the background, additional computer resources are automatically ordered and delivered to the application server's cluster, maintaining the application's performance. Rapid elasticity based on demand is only possible with real-time monitoring driving automated scale. This is why the public cloud is so popular; with a massive pool of available cloud resources and the ability to automatically scale applications out and in based on demand, at AWS anybody can easily scale application stacks up and down. For more details on deploying scale and elasticity with EC2 Auto Scale, check out Chapter 5, "Planning for Scale and Resiliency."

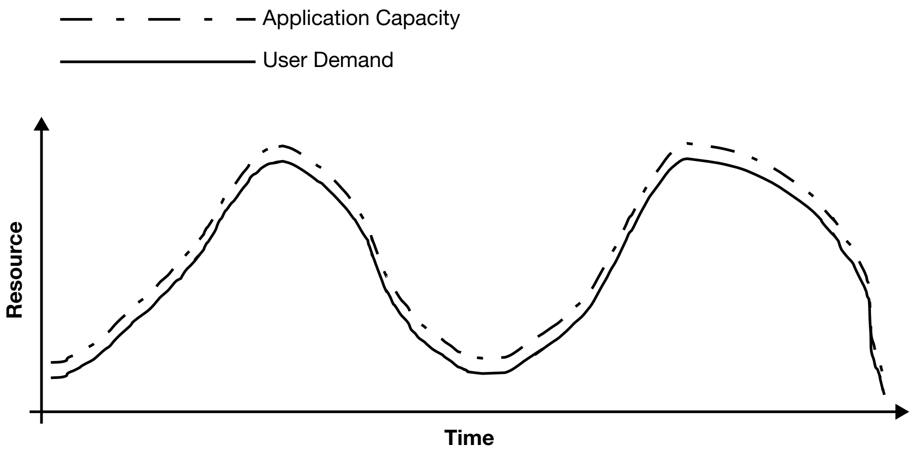


Figure 1-7 Applications can scale based on demand in the public cloud

Measured Service—In the cloud, you are only billed for what you use; that's defined as a measured service. Cloud providers make their money by charging for everything that you use in their data centers, including data transfer costs. Packet flow inbound to the public cloud is usually free; outbound packet flow, or traffic between subnets hosted in different data centers, is usually charged an outbound data transfer fee. Charges are per second, or per minute in the case of computer services like AWS EC2 compute instances, or they are per gigabyte per month in the case of storage services like S3 or virtual hard drives, which at AWS are called elastic block storage (EBS). AWS charges can be broken down into compute, storage, and data transfer charges. If an AWS service is on, the meter is running. Cost management is one of your most important jobs when operating in the cloud. AWS has many useful tools to help you control your costs, including the AWS Simple Pricing Calculator, AWS Budgets, and the Cost Explorer, as shown in Figure 1-8. You can find details on these features in Chapter 2. Being billed for consuming cloud services is a reality that we are all used to. What you also may have to get used to is exactly how you are being billed. Again, you must understand and carefully monitor compute, storage, and data transfer costs. For example, you can order a load balancer at AWS for $30 per month. However, there is an additional charge to be aware of: all the data packets transferred through the load balancer are charged, and that by itself can be a hefty price.

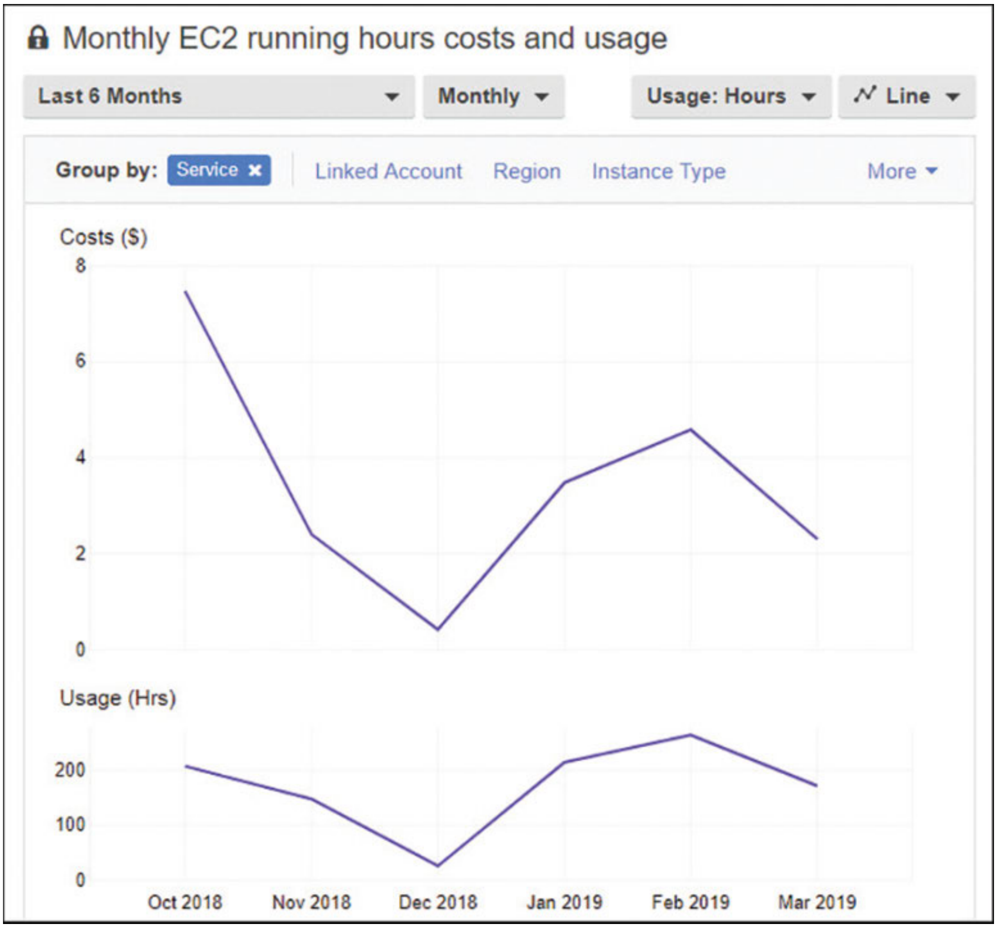


Figure 1-8 AWS Budgets and Cost Explorer track and alert when costs are over budget

Operational Benefits of AWS

Operating in the public cloud has certain benefits. Unlimited access to servers and storage and many management services may make it easier than you expected to operate in the cloud. Table 1-1 summarizes the managed services at AWS that may be able to replace or complement your existing on-premise services and procedures.

Servers—Underutilized servers in your data center are expensive to run and maintain. Moving applications to the public cloud will reduce the size of your on-premise data center. Because you no longer host as many physical servers, your total hosting costs (heating, cooling, and so on) will be lower as well. You also won't have to pay for as many software licenses at the processer level because you're not responsible for running hypervisor services; that's Amazon's job. You may think that moving to the AWS cloud means virtualized resources and only virtualization. However, at AWS, you can get a variety of compute options with virtualization of any size and scale, from a single-core CPU with 512MB of RAM to hundreds of CPU cores and terabytes of RAM. You can also order a bare-metal server and do whatever you want with it. You can find further details on compute options in Chapter 4.

Storage—Using cloud storage has huge benefits due to the unlimited amount of storage promised by cloud providers. Amazon has many options for storage that are similar, but not exactly the same as your on-premise solutions. For storage area network solutions, Amazon has shareable file solutions: the elastic file system (EFS) for Linux workloads, and FSx, a shared file service specifically for Windows File Server workloads. Virtual hard disks are available using EBS. Unlimited storage, and longer-term archive storage, is provided by S3 and S3 Glacier. Details on all the storage options at AWS can be found in Chapter 6, "Cloud Storage."

Managed services—AWS has a variety of managed services, as shown in Table 1-1, that may be able to replace or complement your existing services and utilities currently used on-premise once you move to the AWS cloud.

Table 1-1 **Managed Services at AWS**

IT Operations	On-Premise	AWS Cloud
Monitoring	Nagios, SolarWinds.	CloudWatch monitoring providing metrics for every AWS service. All monitoring and logging data can be stored in S3. All third-party monitoring solutions can access S3 to perform their own custom analysis of log data.
Data backup	Backup tools such as Commvault and NetBackup.	Any third-party vendor that wants to stay in business will be supporting AWS; both Veritas and Commvault have AWS solutions. AWS Storage Gateway can also be installed to cache required content locally, while backing up local disk volumes to an S3 bucket. Backups can be snapshots of local virtual hard disks, or data files from specific volumes can be targeted.

IT Operations	On-Premise	AWS Cloud
Scale	Add additional virtual machines or increase/decrease the size of each virtual machine's RAM and CPU cores.	Scale horizontally by placing multiple virtual machines (instances) behind a load balancer and add automated scaling based on demand to increase and decrease the required amount of compute power using EC2 Auto Scaling.
Testing	Provisioning hardware for testing is expensive.	Provisioning resources for short-term testing at AWS is incredibly inexpensive. Signing up for the AWS free tier allows you to test a variety of AWS services for one year completely free.
Identity management	Active Directory Domain Services for accessing corporate resources.	Extend on-premise Active Directory to the AWS cloud with hosted Directory Services. Utilize AWS single sign-on services (SSO) for managing access to popular business applications that third-party cloud providers are hosting.

Cloud Provider Limitations

Each cloud provider has a published SLA that specifies what services are provided and at what specific operational level. All public cloud providers make promises about how they will handle security, compliance, and overall operations and how their methodology will be contained in the cloud provider's SLA. The challenge is to live up to that agreement. In the SLA, there will be details about acceptable outage time and the responsibility of the cloud provider when outages occur. There also will be statements about not being responsible for events outside the cloud provider's control. Another common term typically used in the SLA is "best effort" or "commercially reasonable effort."

Regardless of the cloud model, the cloud provider is responsible for overall service operation and deployment, service orchestration, the overall management of the cloud, the security of the cloud components, and maintenance of customer privacy. The responsibility of how each customer, the cloud consumer, is to carry out business with the cloud provider will also be described in some detail in the SLA. Each cloud consumer must fully understand what each cloud service offered provides; this is exactly what the cloud service will and will not do.

The reality is that every public cloud provider will not have an SLA that you will like, and the stark reality is that their best effort is the best they can do. This might seem a little harsh, but it's reality; according to AWS, "everything fails all the time." What happens when a key component of your application hosted in the AWS cloud fails? Is it a disaster, or is it manageable? Is it acceptable to expect AWS failures from time to time? It's a reality; AWS is 100% right; everything fails.

Operating in the public cloud means that you must design your hosted application to be able to continue operating even if compute and storage failures occur. That's our responsibility.

All public cloud providers really have the same SLA; here it is, summarized in nine short words: "we are sorry; we will give you a credit." This SLA summary applies to every public cloud provider. Here's another reality check; if you're down, you will have to *prove* that you were actually down by providing network traces and appropriate documentation that leaves no doubt that you were down because of an AWS cloud issue.

Oh, and here's another small detail to be aware of: if you didn't build redundancy into your application design, don't bother calling for a credit. Application designs that have a single instance hosting the application with no failover or high-availability design parameters have no SLA. AWS expects you to be serious about your application design; we need to understand and use the tools in the AWS toolbox to ensure that *your* SLA for availability and performance is achieved.

Not every service at AWS even has a defined SLA; there are more than 100 services and only 8 defined SLAs. Remember: all managed services—in fact, all services—are built from the resources found in Table 1-2.

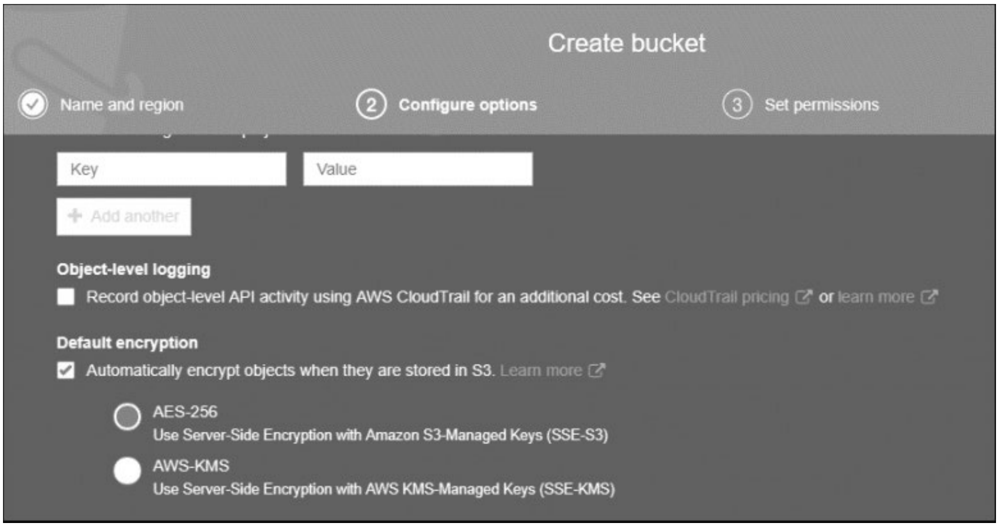
Table 1-2 **SLAs at AWS**

AWS Service	SLA Summary
CloudFront	99.9% during any monthly billing cycle
DynamoDB	Monthly uptime percentage of 99.999% for global tables, or 99.99% for regular tables
EC2 instances (includes elastic container service [ECS] and EBS volumes)	Monthly uptime percentage of at least 99.99%
RDS databases	Monthly uptime percentage of at least 99.95% for multi-AZ instances
Route 53 DNS service	Commercially reasonable efforts to make Route 53 100% available during a monthly billing cycle
S3; S3 Glacier object storage	The number of errors calculated during each 5-minute period subtracted from 100%
Lambda functions	Monthly uptime percentage of 99.95% during any monthly billing cycle
AWS Shield (Advanced)	Any failure of service commitments provided by CloudFront or Route 53 when being protected by AWS Shield Advanced distributed denial of service (DDoS) protection

Data Security at AWS

We can lose many things while operating in the cloud: instances fail, EBS volumes crash, services stop working. But you can't go to your boss and say we've lost some data.

Data security—The reality is that your data is more secure and durable stored in the public cloud. At AWS, except for S3 Glacier archive storage, which is automatically encrypted, all other storage mediums at AWS are unencrypted by default. However, EBS volumes—both boot and data volumes—can be encrypted at rest and at transit using either customer master keys provided by AWS or keys provided by the customer. Shared storage services such as EFS can also be encrypted at rest, as can DynamoDB tables. S3 buckets can be encrypted with keys provided by AWS or supplied by customers, as shown in Figure 1-9. Data durability provides security of a different nature; all data stored in the cloud is stored in multiple locations; EBS volumes are replicated within the data center where they reside. S3 objects are replicated across three separate locations within the selected AWS region, producing a high level of durability. Amazon's level of S3 durability is humorously defined like this: for every 1,000 objects stored in an S3 bucket, you will lose one of those objects every 10 million years. We cannot possibly duplicate this level of durability and security on-premise.



Object-level logging

☐ Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more

Default encryption

✓ Automatically encrypt objects when they are stored in S3. Learn more

AES-256
Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

AWS-KMS
Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Figure 1-9 S3 buckets can be encrypted using AES-256 or AWS-KMS managed keys

Data privacy—AWS does not have data storage isolated for individual customers; all storage arrays at AWS are multitenant in design. This is pretty much the default for all public cloud providers. Amazon's job is to make sure your stored data records are isolated per AWS account.

Data control—Customers are in full control of storing and retrieving their data stored in AWS. All data storage at AWS starts as private, and except for S3 buckets that are changed allowing public access, storage remains private and is not directly accessible from the outside world. Customers can choose to make S3 buckets public; it's the customer's responsibility to define the security and accessibility of all data records stored in AWS.

Security controls—As previously mentioned, all data records can be encrypted at AWS. Resource policies defining the precise level of security and access can be directly attached to resources such as S3 buckets or EFS shared storage and can be defined by the identity and access management (IAM) user and group security policy using the IAM service.

IAM identity and trust policies can be defined at a granular level controlling access by users and roles to *all* resources at AWS, including *any* storage medium. Chapter 7, "Security Services," provides details on IAM.

You can enable multifactor authentication as an additional security control on S3 buckets to control when deletion of data records is performed.

Network Security at AWS

At AWS, networking is managed at the subnet level, and all subnets are created as a private subnet with no access to the outside world. Subnets reside on your private networks, which are called a virtual private cloud (VPC) at AWS. Only by adding a gateway service to a VPC will subnets be able to be accessed from either the Internet or a private VPN connection from an on-premise network. Chapter 3 has the details on networking at AWS.

It's important to note that public and private connectivity choices are decisions that are always carried out by each customer; not AWS.

- Each subnet's ingress and egress traffic can be controlled by a subnet firewall called Network ACLs that define separate stateless rules for both inbound and outbound packet flow.
- Each EC2 instance hosted on a subnet is further protected by an additional firewall called a security group, which defines what traffic is allowed into the instance and where outbound traffic is directed.

VPC flow logs can be enabled to capture network traffic for the entire VPC, a single subnet, or a network interface.

Application Security at AWS

Both Web and application servers hosted at AWS should always be located on private subnets. Private subnets are not directly accessible from the Internet. You may be wondering how to access what was supposed to be a public-facing application with no direct public access. The solution to this question is the absolute best practice to follow at AWS: for Web servers that customers across the Internet access, placing the load balancer on a public subnet, in front of the Web servers, provides the correct design solution. Customers requesting access to the application will be directed by DNS to the DNS name of the load balancer. The load balancer directs incoming traffic from the public subnet to the targeted Web servers hosted in the private subnets.

One load balancer type offered by AWS is the Application Load Balancer, which can perform authentication and SSL offload services. The end-to-end traffic pattern for a three-tier Web application can be designed using many encryption/decryption points, as shown in Figure 1-10 on its path from source to destination:

- **Web application firewall**—A custom traffic filter in front of the Application Load Balancer protecting against malicious traffic.
- **Elastic Load Balancer (ELB)**—Accepts only encrypted HTTPS traffic on port 443; provides secure sockets layer/transport layer security (SSL/TLS) decryption and, optionally, user authentication.
- **EC2 instance hosting Web application**—EBS boot and data drives can be encrypted.
- **EC2 instance hosting application server**—EBS boot and data drives can be encrypted.
- **Database server**—EBS boot and data drives and data community can be encrypted, or Dynamo DB tables can be encrypted.

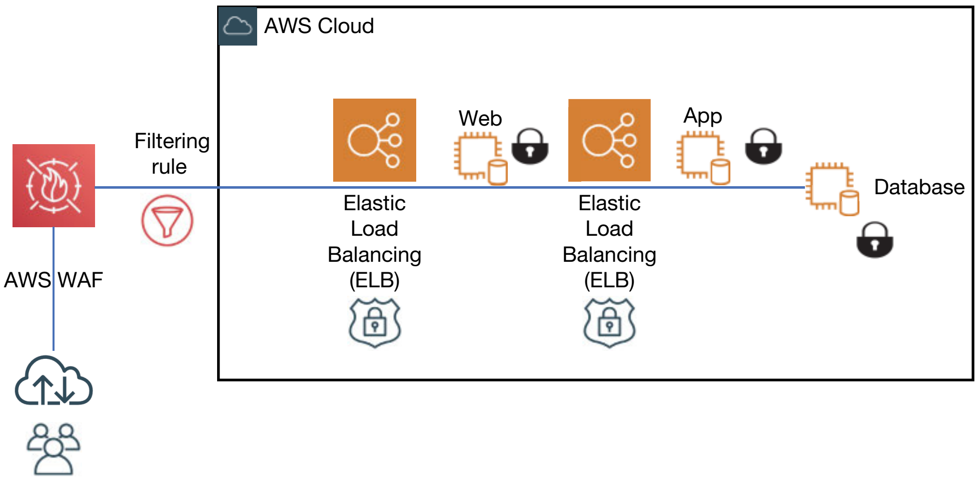


Figure 1-10 Encrypted traffic flow at AWS

Compliance in the AWS Cloud

As a worldwide public cloud provider, AWS operates in many different countries and is subject to a variety of rules and regulations enforced by governments and compliance standards. Depending on the type of business that you operate, there are possibly many different levels of compliance you will have to adhere to when operating in the AWS cloud. Financial, health, and government institutions have strict rules and regulations that must be followed by their clients. In addition, your own company may have specific internal rules and regulations they want to follow.

Many countries in the world are enacting laws, regulations, and mandates in serious attempts to protect the privacy of personal data and the security of corporate information and computer systems. The new data protection laws place the burden of protection and security on the custodian of that data; that is where the data is stored when the data is transferred from source to destination.

The cloud providers have contractual obligations to ensure that when organizations have data records hosted in their cloud, they can adhere to the promises and commitments made in the SLA. Some of the most common compliance regulations that AWS has been successfully audited against include the compliance standards listed in Table 1-3.

Table 1-3 **AWS Supports Many Compliance Standards**

Abbreviation	Scope of Operation	Purpose of Protection	Legal Status
HIPPA	Healthcare	Personal information	Law
GLBA	Financial industry	Personal information	Law
SOX	Publicly traded companies	Shareholder	Law
PCI DSS	Payment card industry	Fraud	Industry regulation
GDPR	EU	Personal information	Law

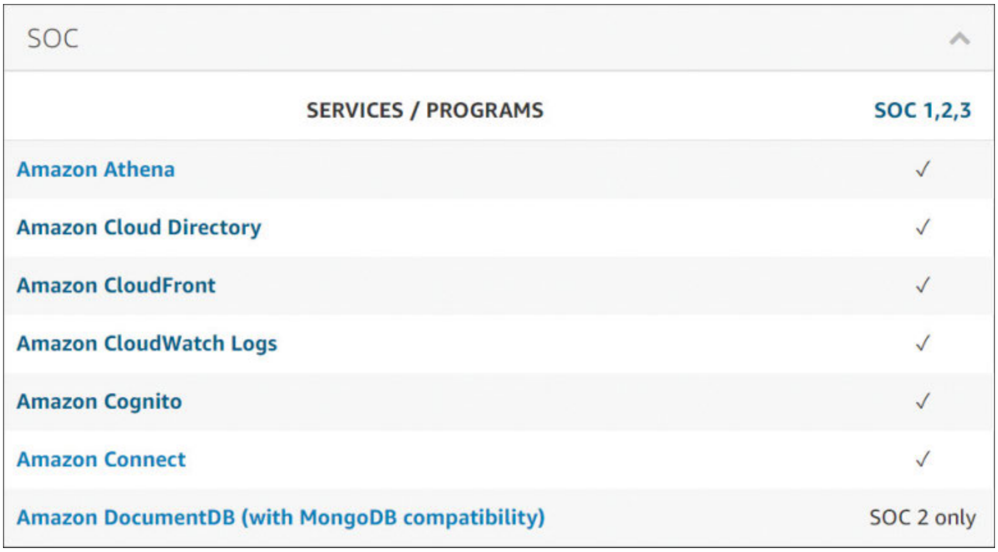
Health Insurance Portability and Accountability Act—Secures the privacy of individual health information records in the United States.

Gramm-Leachy-Billy Act—Mandates protection of customer information by financial industries.

Sarbanes-Oxley—Ensures the integrity of financial operations of publicly traded companies.

PCI DSS—Ensures the processing integrity of credit card data or authentication data.

GDPR—Protects privacy and personal data for all citizens of the EU. Amazon has a decent compliance page at https://aws.amazon.com/compliance/, which has details about all the AWS certifications and attestations that it has achieved or supports. If you are bound by a specific compliance standard, one of your first steps should be to review the AWS services that are available for each compliance standard, as shown in Figure 1-11.



SOC

SERVICES / PROGRAMS	SOC 1,2,3
Amazon Athena	✓
Amazon Cloud Directory	✓
Amazon CloudFront	✓
Amazon CloudWatch Logs	✓
Amazon Cognito	✓
Amazon Connect	✓
Amazon DocumentDB (with MongoDB compatibility)	SOC 2 only

Figure 1-11 Check the AWS compliance page to see what services are supported

Playing in the AWS Sandbox

AWS makes it easy to "try before you buy," frequently doling out promotional credits to developers. Even if you are not a developer, every new AWS customer gets limited access to nearly every AWS service for free (Amazon calls this the "free tier") during the first year. This is a great way to experiment with AWS. The only thing you must provide is a credit card that won't be charged unless you choose to use resources that the free tier doesn't cover. After the first year has passed, you'll start accruing charges for every service you use; any AWS resources that you built during the first year remain in your account but start accruing charges.

In addition, AWS has several free hands-on labs. You can sign up for QwikLabs at https://run.qwiklabs.com/home?locale=en and carry out a variety of AWS tasks in the AWS cloud.

Figure 1-12 illustrates some of the learnig and labs that are available from QwikLabs.

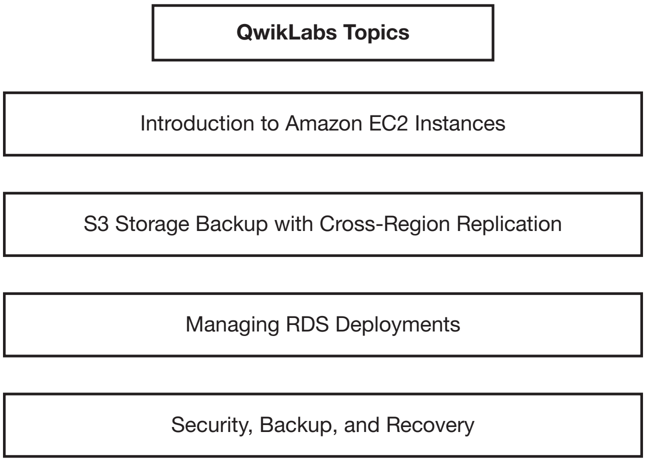

QwikLabs Topics
Introduction to Amazon EC2 Instances
S3 Storage Backup with Cross-Region Replication
Managing RDS Deployments
Security, Backup, and Recovery

Figure 1-12 QwikLabs has more than 20 completely free labs for AWS services

Running experiments, and performing labs raises additional questions that will help further your AWS cloud knowledge and experience.

MAKE SURE TO WATCH THE COMPANION VIDEO "SIGNING UP FOR AWD FREE TIER."

To access the companion videos, register your book at informit.com/register.

What's the Problem That Needs to Be Solved?

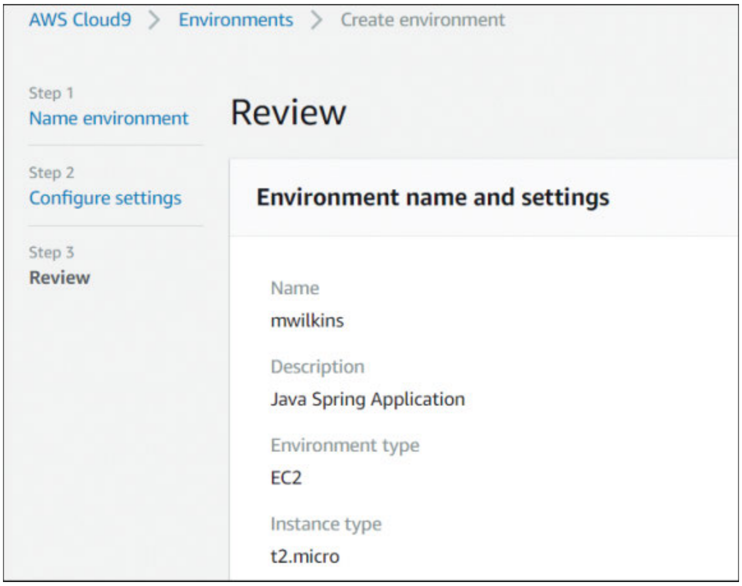
Typical large organizations run hundreds or thousands of applications on thousands of virtual servers. Which applications can be moved to AWS? What should be prioritized?

Start with low value/low risk—It's quite popular to suggest a starting point of high value and low risk when choosing your first application to move to the AWS cloud. Here's a reality check: it's probably going to take you 6 months or longer to move your application to the cloud. Choosing an application with low value provides a valuable timeline to do some additional planning and analysis before finalizing your application in its working form at AWS. I've seen many companies make the pronouncement that applications will be moving to the cloud quickly. It rarely happens successfully because there are so many things to learn and consider. Start with low value. Take your time, and select a working application that has been running successfully for a good time period. Then you can document your lessons learned and what to do differently the next time. The second and third application moved to the cloud generally will be much faster than the first application due to the lessons learned and experience gained.

Create a brand-new application first—The advantage of creating a completely new application at AWS means you are not constrained by anything, such as the type of database that must be used, the type of programming language that must be used, or the type of compute that must be used. Starting anew at AWS allows you to try out some of the new methods to host applications such as serviceless computing, create a mobile application using stateless components, or use DynamoDB instead of SQL. This is where the real learning about what the AWS cloud can do for you will really appear.

Try to solve a single problem—Do you need additional storage? Perhaps that's a great starting point for your adventure in the cloud. Archiving files in S3 Glacier could be as simple as ordering a Snowball device, connecting it up to your network, filling up with files you'd like to archive, and shipping it back to AWS. This is an excellent first project to start working with AWS support, archiving records, and saving your company money.

Define a value proposition—Ideally, the move to AWS is long term and successful. Thousands of companies have been successful moving to AWS; you, too, can be successful. Start off with a defined value proposition that can be validated quickly, in a matter of months rather than years. For developing applications, you could sign up for AWS Cloud9, a cloud-hosted IDE that supports more than 40 programming languages, as shown in Figure 1-13. Armed with a browser, you can try your hand at developing applications at AWS.



Environment name and settings

Name
mwilkins

Description
Java Spring Application

Environment type
EC2

Instance type
t2.micro

Figure 1-13 Cloud9 IDE at AWS

Access to data records—The number-one problem with larger companies when starting to work with cloud providers is working through the internal politics to allow access to data from the cloud. Data record access, and the steps for successful access, should be considered before you move to the cloud:

- How can we access our on-premise data from the cloud?
- What records have to stay on-premise?
- Are we bound by any compliance rules and regulations?
- Is our data in the right format for what we need?

Migrating Applications

For applications that have been chosen as starting candidates to move to the AWS cloud, several decisions need to be made about the application's journey, or path.

Can the application be moved to AWS and hosted on an EC2 instance with no changes?

Applications that fit into this category could be migrated to AWS as an EC2 instance image. Server migration tools, and database migration tools discussed in Chapter 2, can carry out these migration paths quite effectively. However, applications that are lifted and shifted to the cloud will have other dependencies and issues that will have to be considered:

- The application stores its data in a database. Will the database remain on-premise or be moved to the cloud?
- If the database for the application remains on-premise, are there latency issues that need to be considered when communicating with the database?
- Will a high-speed connection need to be established between the AWS cloud and the database remaining on-premise?
- Are there compliance issues regarding the application data? Does the data have to be encrypted at rest? Does communication with the database need to be encrypted?
- Do users authenticate to the application across the corporate network? If so, are federation services required to be deployed at AWS for single sign-on (SSO)?
- Are local dependencies installed on the application server that will interfere with the application server's operation in the AWS cloud?
- Are there licensing considerations for both the operating system and the application when operating in the cloud?

Is there an existing SaaS application hosted by a public cloud provider that should replace the application because it's a better choice?

This can be a very political issue to resolve. With so many hosted cloud applications available in the public cloud, the odds are close to 100% that there will be an existing application that could replace the current on-premise application.

Should the application remain on-premise and eventually be deprecated?

- The application is hosted on legacy hardware that is near end-of-life.
- The application is not virtualized.
- The application does not have support.
- The application is used by a small number of users.

The Well-Architected Framework

Several years ago, AWS introduced documentation called the Well-Architected Framework to help customers plan properly when moving to the AWS cloud. The goal was to give guidance for cloud architects to build secure, resilient, and decent performing infrastructure to host their applications following recognized best practices that have been developed over time by the experience of many AWS customers. Each best practice still must be evaluated as to whether it meets your criteria. A best practice should not be blindly adopted without understanding why it has achieved a best practice designation.

The documentation for the well-architected framework also has many key questions to ponder that can be found in the well-architected framework blueprint. It is useful to discuss these questions out loud with other technical folks in your company; they will help you make key decisions about your infrastructure and applications hosted at AWS. The framework documentation can be found here: https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf. Each application to be deployed at AWS needs to be viewed through the lens of being well architected following these five principles:

Operational excellence—How best to execute, deploy, and monitor applications running at AWS using automated deployment monitoring procedures, continuous improvement, and automated solutions for recovering from failures. Key AWS services to utilize include CloudWatch events and alarms, CloudTrail, EC2 Auto Scaling, AWS Config, and the Trusted Advisor. Check out Chapters 5, 7, and 8. Operational excellence questions to consider include these:

- How are disruptions to applications handled? Manually, or automatically?
- How can you analyze the ongoing health of your applications and infrastructure components hosted at AWS?

Security—How to best design systems that will operate reliably and securely while protecting customer information and data records. Key AWS services to utilize include IAM, AWS Organizations, CloudWatch logs, CloudTrail events, S3 and S3 Glacier, and VPC flow logs. Check out Chapters 3, 6, and 7. Security questions to consider include these:

- How are security credentials and authentication managed at AWS?
- How are automated procedures secured?

Reliability—How can systems and applications hosted at AWS recover from disruption with minimal downtime? How can applications meet your escalating demands? Key AWS services to utilize include ELB, EC2 Auto Scaling, and CloudWatch alarms. Check out Chapter 5. Reliability questions to consider include these:

- How do you monitor resources hosted at AWS?
- How do applications hosted at AWS adapt to changes in demand by end users?

Performance efficiency—How to use compute resources to meet and maintain your application requirements on an ongoing basis. Should your compute solution change from EC2 instances to containers or serviceless? Key services include EC2 Auto Scaling, EBS volumes, and RDS. Check out Chapters 4 and 6. Performance efficiency questions to consider include these:

- Why did you select your database?
- Why did you select your current compute infrastructure?

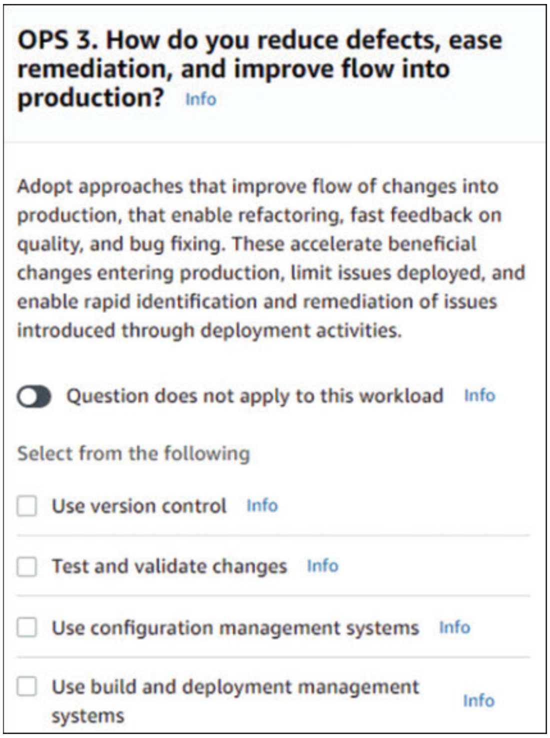
Cost Optimization—How to design systems that meet your needs at the cheapest price point. Key AWS services include Cost Explorer, Budgets, EC2 Auto Scaling, Trusted Advisor, and the Simple Monthly Calculator. Check out Chapters 2, 5, and 7. Cost optimization questions to consider are as follows:

- How do you oversee usage and cost?
- How do you meet cost targets?
- Are you aware of current data transfer charges based on your AWS designs?

The Well-Architected Tool

In the AWS management console under “Management and Governance” is the AWS Well-Architected Tool, as shown in Figure 1-14. It provides a framework for documenting your workloads against AWS best practices as defined in the well-architected framework documentation. In each of the five pillars, there are many questions to consider before deploying your application. As you consider each question, you can enter milestones to mark changes in your architecture as it moves through its deployment and build lifecycle. Working with the well-architected tool, you will receive tips and guidance on how to follow the best practices recommended by AWS while carrying out a full architectural review of an actual workload that you are planning to deploy at AWS. It is well worth the time spent.

Before the review begins, you will select the AWS region where your application will be hosted. The first step is to define the workload and choose the industry type and whether the application is in a production or preproduction environment. During the review process, the well-architected tool will identify potential areas of medium and high risk based on the answers to the questions posed during the workload review. The five pillars of design success will also be included in the plan that is presented showing the recommended improvements to your initial design decisions. The plan as shown in Figure 1-15 will also define both high and medium risks, with recommended improvements to consider implementing.



OPS 3. How do you reduce defects, ease remediation, and improve flow into production? Info

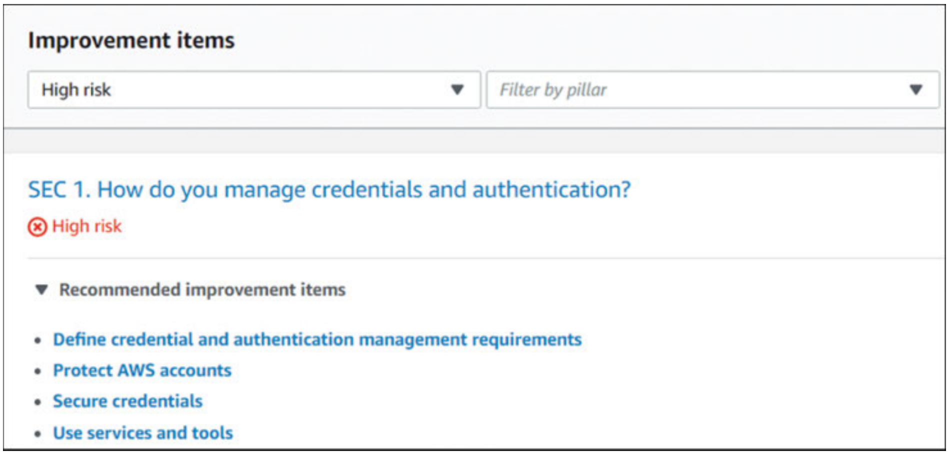
Adopt approaches that improve flow of changes into production, that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

Question does not apply to this workload Info

Select from the following

- [ ] Use version control Info
- [ ] Test and validate changes Info
- [ ] Use configuration management systems Info
- [ ] Use build and deployment management systems Info

Figure 1-14 Using the well-architected framework tool



Improvement items

High risk ▼ | *Filter by pillar* ▼

SEC 1. How do you manage credentials and authentication?

High risk

▼ Recommended improvement items

- Define credential and authentication management requirements
- Protect AWS accounts
- Secure credentials
- Use services and tools

Figure 1-15 Recommended improvements using the well-architected tool review

In Conclusion

In this initial chapter, we looked at just what the public cloud is these days and how AWS fits into the public cloud arena in the areas of infrastructure and development, namely IaaS and PaaS. The cloud is a data center; it's just not yours.

The chapter looked at how NIST has defined the public cloud and how AWS fits into NIST's definition; in most cases the initial NIST definition has morphed into a standard, followed by most corporations that have moved to the AWS cloud. We ended off with a bit of homework, suggesting that you should sign up for an AWS account and look at ways to leverage the free tier to further your learning, and you should review the AWS compliance page to see how your compliance needs match with what AWS can offer. And, of course, you should carefully review the well-architected framework documentation. It's a pretty good guideline and online utility for getting used to how Amazon operates and how you probably want to operate in the cloud. The well-architected framework is also the baseline for the AWS Architecture Associate certification if you're moving toward getting certified in the future.

Don't forget about the companion videos, which are going to be key to working at AWS. In the companion videos, you'll be introduced to Terra Firma, our use case for this book and the videos. Each video will look at a problem or situation that Terra Firma is facing as a company and suggest a solution. Each chapter also starts with several issues and concerns being faced by Terra Firma. It's my hope that you can relate to the company's concerns and the presented solutions. Each chapter ends with some relevant discussion points for consideration.

MAKE SURE TO WATCH THE COMPANION VIDEO ON OUR USE CASE FOR THIS BOOK: "Terra Firma."

Let's start learning about the big picture: regions, availability zones, and edge locations in Chapter 2.

Index