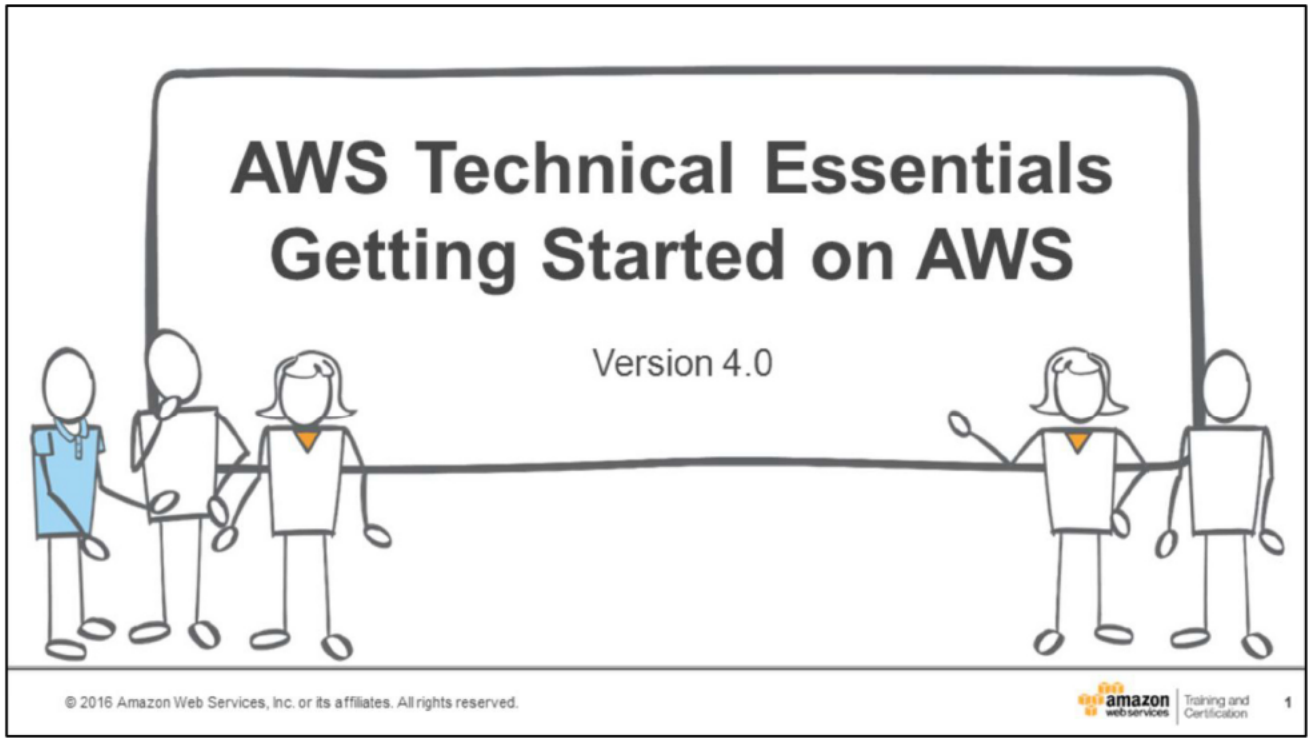
AWS Technical Essentials
Student Guide
Version 4.0

100-ESS-40-EN

Corrections or feedback on the course, please email us at:
aws-course-feedback@amazon.com.
For all other questions, contact us at:
https://aws.amazon.com/contact-us/aws-training/.

Contents



This instructor-led training introduces AWS products and services with exercises and hands-on activities. This training is designed for anyone new to Amazon Web Services so you can gain proficiency in AWS services and make informed decisions about IT solutions based on your business requirements. This training will help you gain the knowledge required to truly get started working on your projects using AWS.

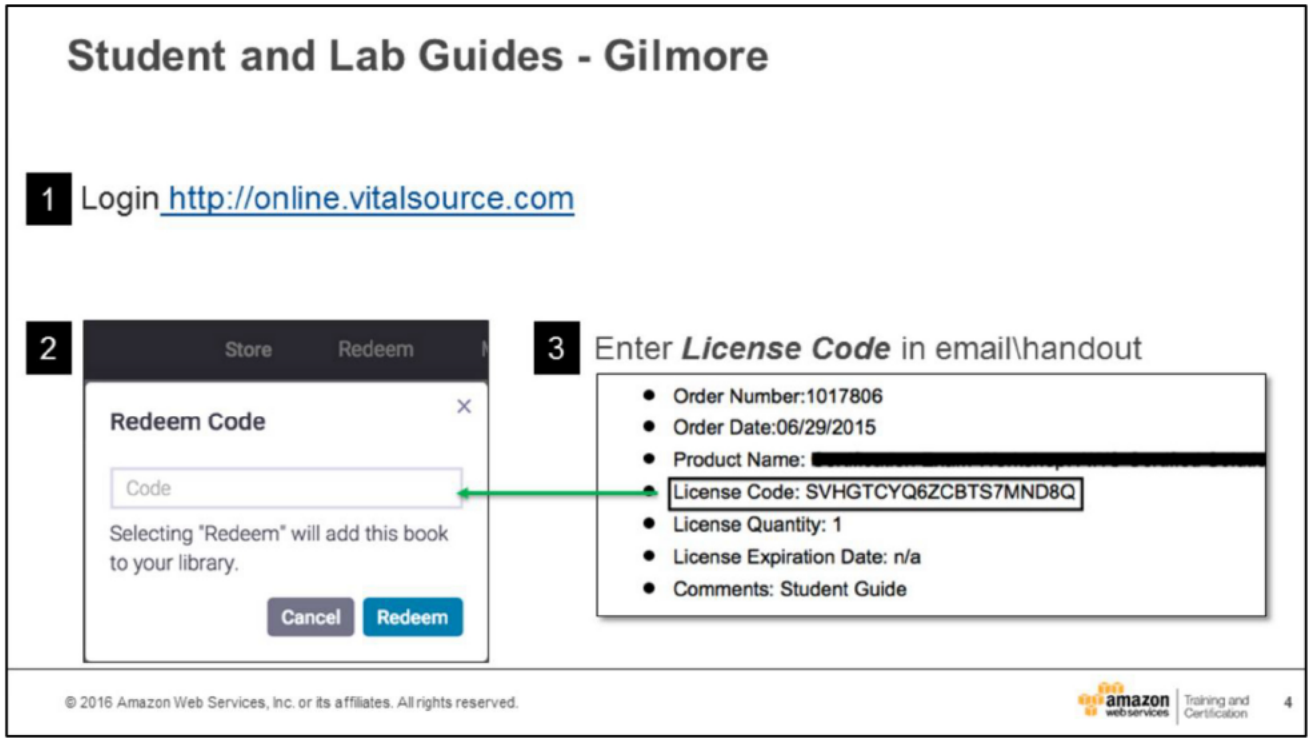
Introductions and Logistics

- Class introductions
- Logistics, bathrooms, breaks
- Participation
- Parking lot
- Cell phones

amazon web services | Training and Certification 2

Module Layout

- **Module 1**: AWS Introduction and History
- **Module 2**: Infrastructure Services: Amazon EC2, Amazon S3, Amazon EBS, and Amazon VPC
  - Lab 1: Build your VPC and launch a web server
- **Module 3**: Security, Identity, and Access Management: IAM
- **Module 4**: Databases: Amazon DynamoDB and Amazon RDS
  - Lab 2: Build your database server and interact with you database using an application
- **Module 5**: AWS Elasticity and Management Tools: Auto Scaling, Elastic Load Balancing, Amazon CloudWatch, and AWS Trusted Advisor
  - Lab 3: Scale and load balance your application and monitor activity
- **Module 6**: Course Wrap-Up



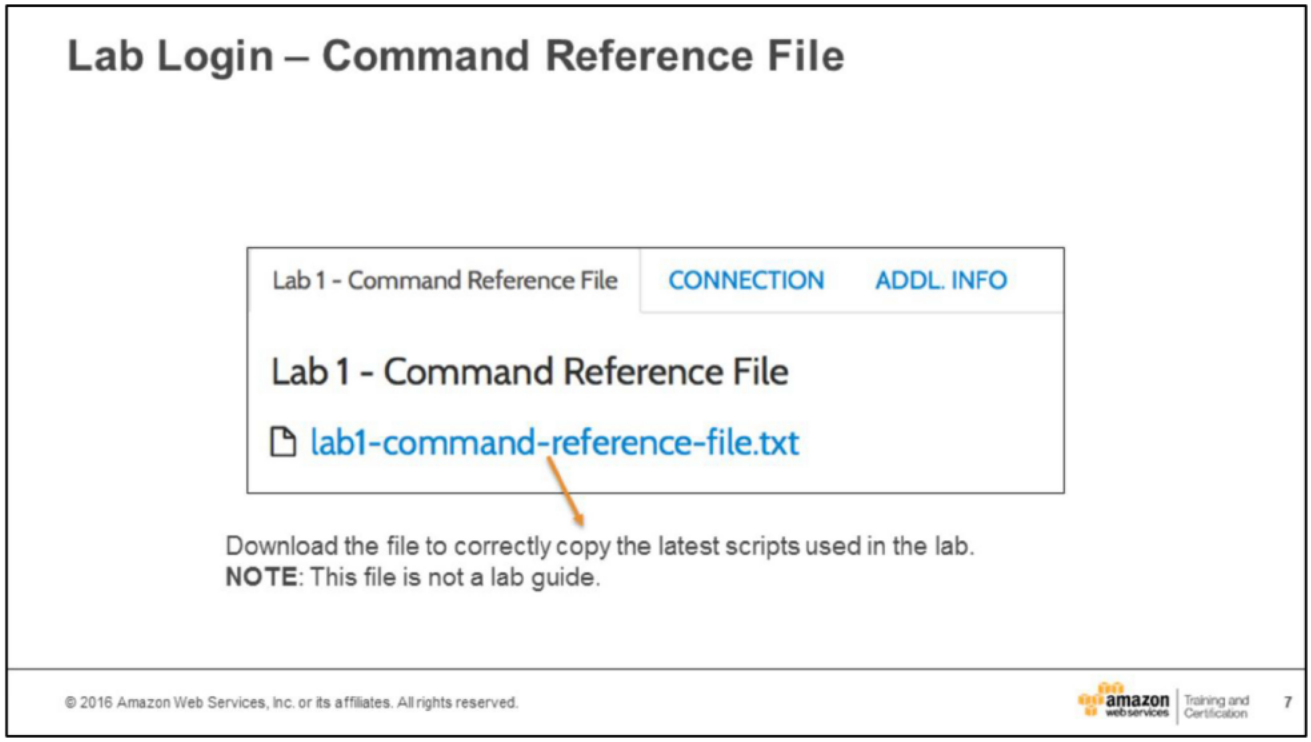
QwikLABS Access

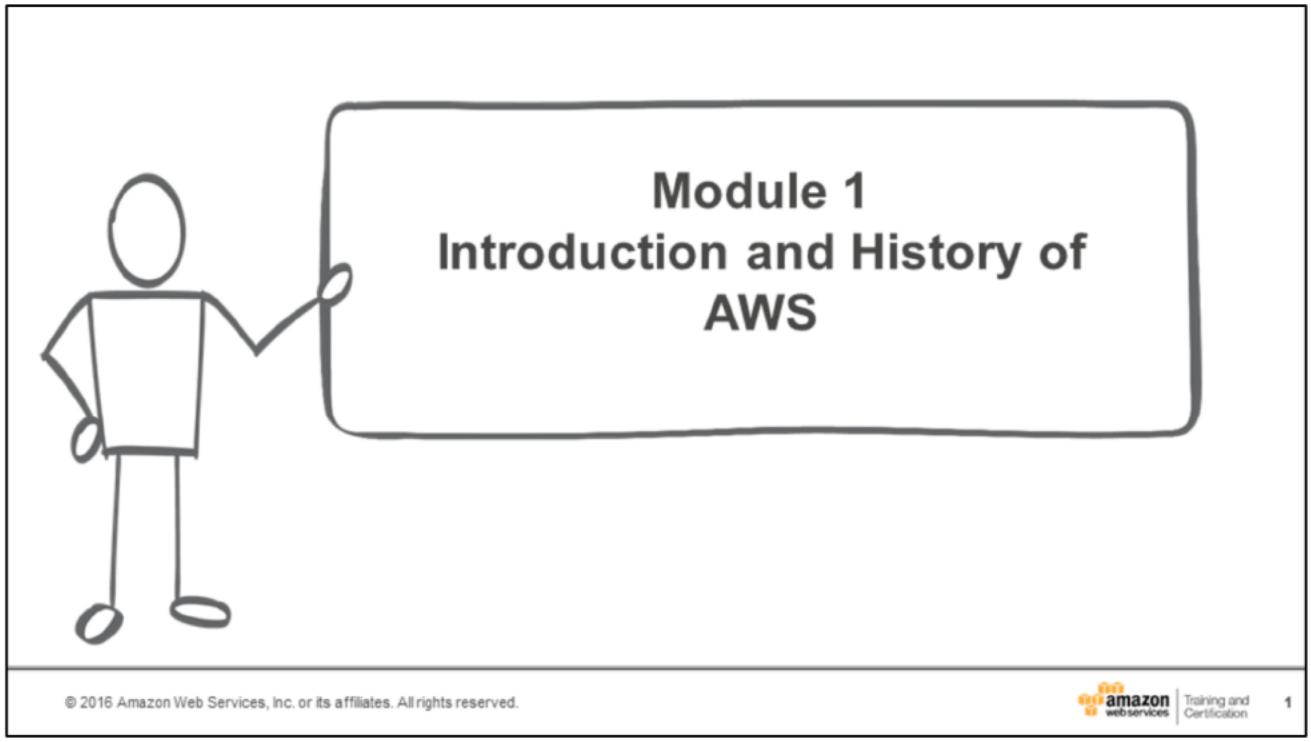
Login (Safari, Chrome, or Firefox preferred)
https://aws.qwiklab.com

amazon web services | Training and Certification 5

Logging into the AWS Management Console from QwikLABS

Gilmore VitalSource Bookshelf contains the lab manual.
NOTE: Do not change the default selected region.

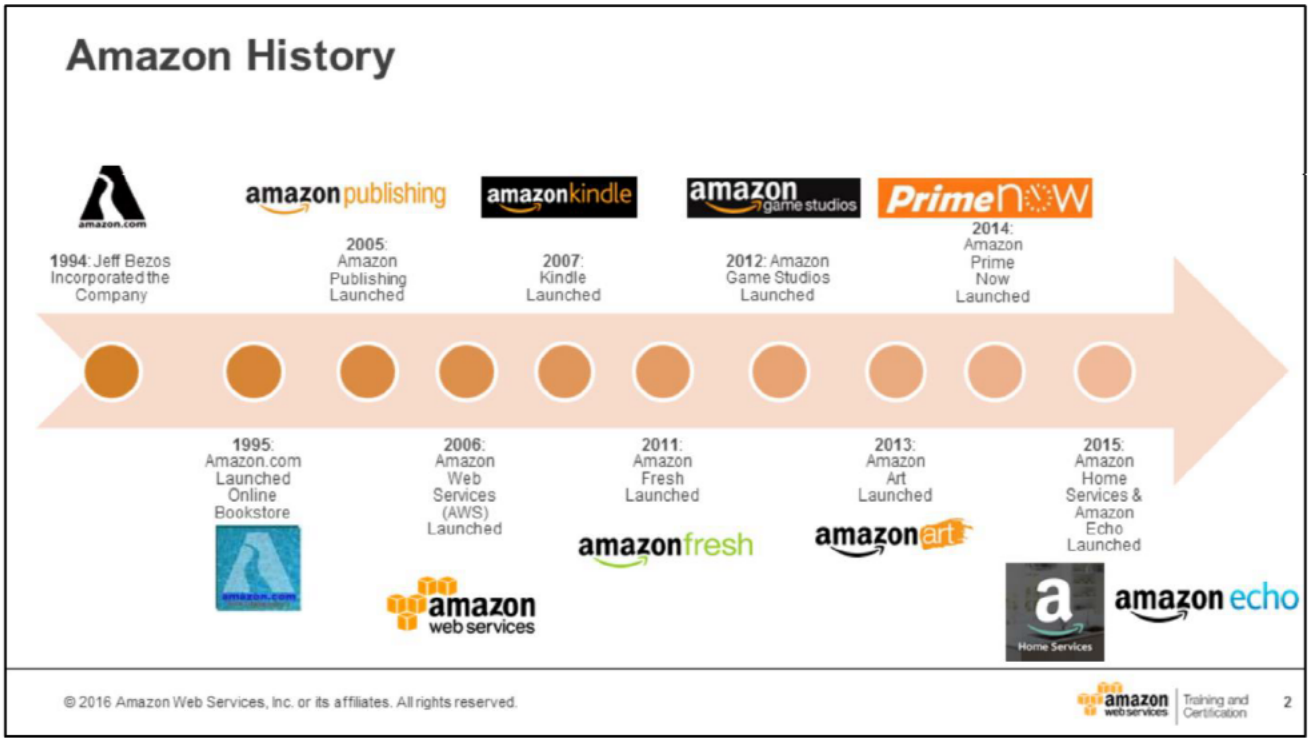




Module 1 describes the history and fundamental elements of Amazon Web Services (AWS), and shows you how to navigate the AWS Management Console. The AWS Management Console is the graphical user interface (GUI) to access AWS. AWS is also accessible through the command line interface (CLI) and software development kits (SDKs). This module discusses the AWS global infrastructure, security measures provided by AWS, and basic principles of deploying on AWS.

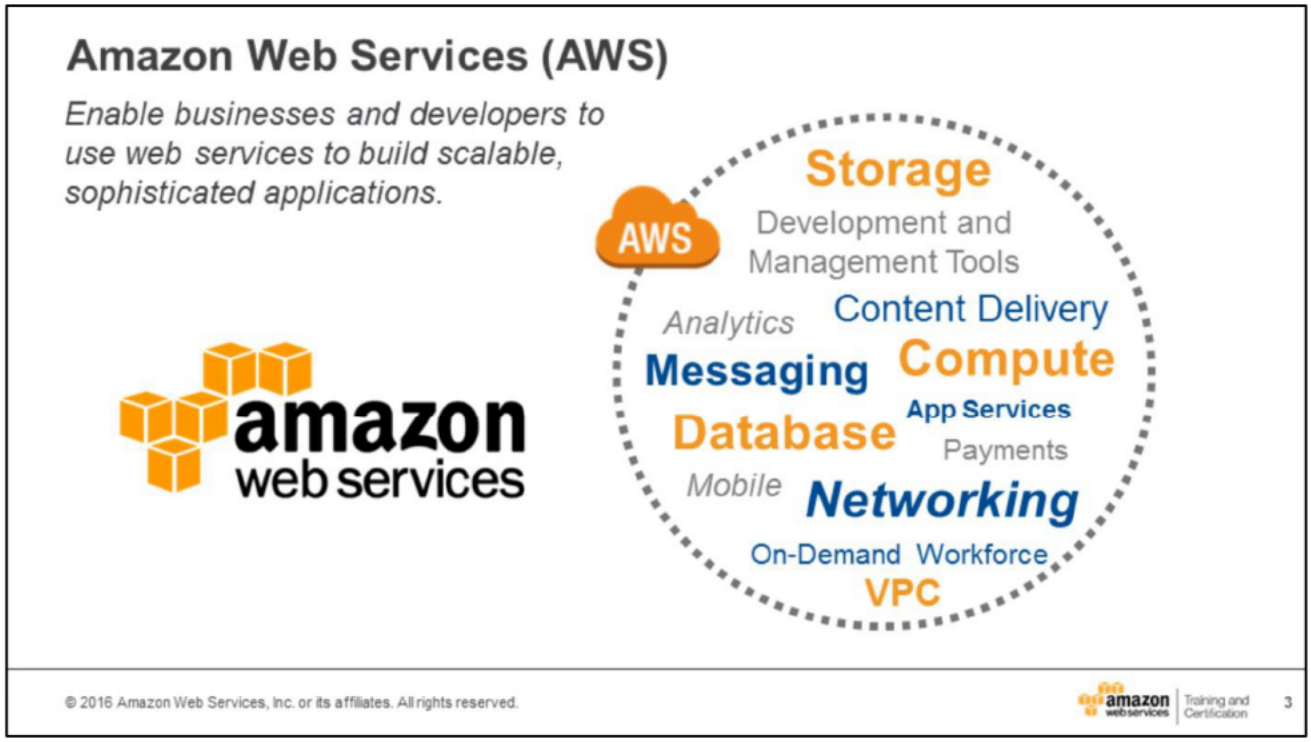
By the end of this module you will be able to:

- Describe the history of AWS.
- Recognize the AWS global infrastructure.
- Navigate the AWS Management Console.



Jeff Bezos incorporated the company in 1994 and Amazon.com was launched in 1995 as an online bookstore. Amazon.com, Inc. is an American multinational electronic commerce company with its headquarters in Seattle, Washington. It is the world's largest online retailer. Amazon has continued to grow and officially launched Amazon Web Services (AWS) in 2006. More came after, including Amazon Publishing, the Kindle, Amazon Game Studios, and Amazon Art.

After over a decade of building and running the highly scalable web application, Amazon.com, the company realized that it had developed a core competency in operating massive scale technology infrastructure and data centers, and embarked on a much broader mission of serving a new customer segment—developers and businesses—with a platform of web services they can use to build sophisticated, scalable applications. Today, AWS is the fastest-growing multi-billion dollar enterprise IT vendor in the world.

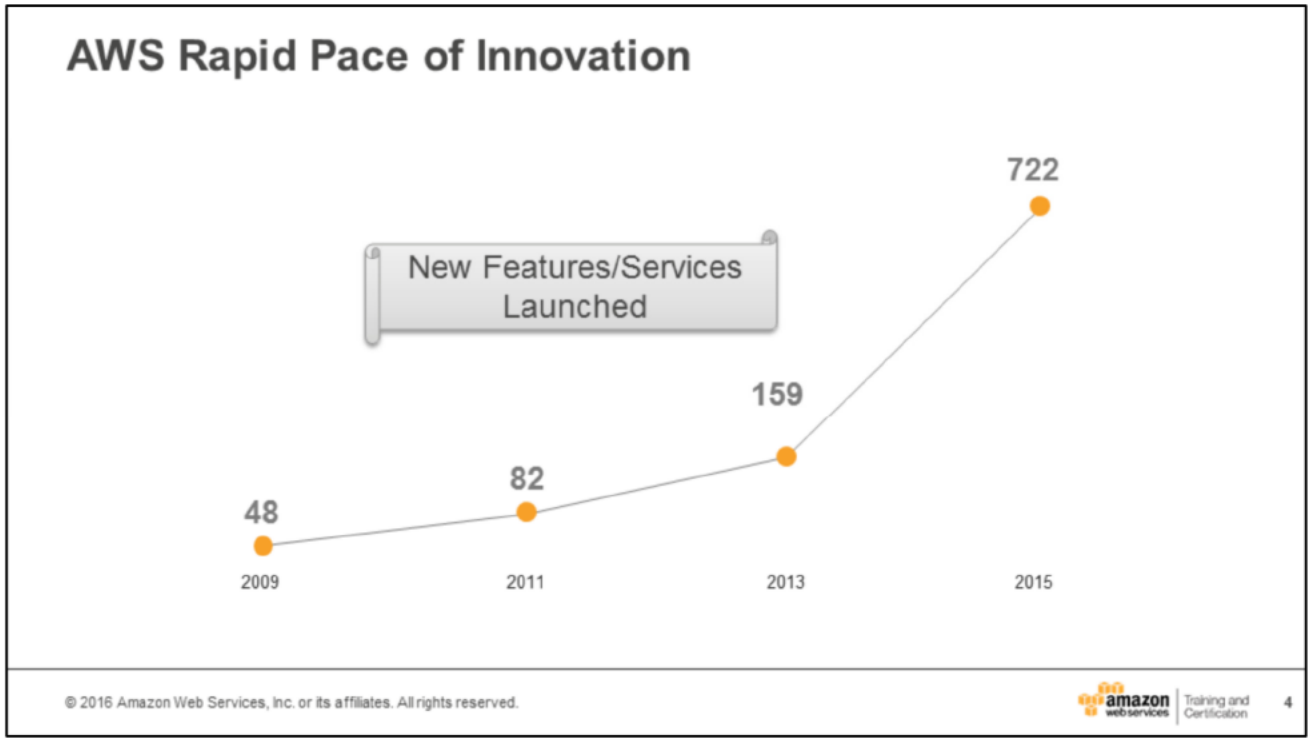


Amazon Web Services is 10+ years in the making. Amazon Web Services, also abbreviated to AWS, is a collection of remote computing services called web services. These web services make up a cloud computing platform offered via the Internet. We deliver web-based cloud services for storage, computing, networking, databases, and more.

The AWS mission is to enable businesses and developers to use web services to build scalable, sophisticated applications. Web services is another name for what people now call “the cloud.”

For more information, see:

Learn more about Amazon Web Services (AWS) - http://aws.amazon.com

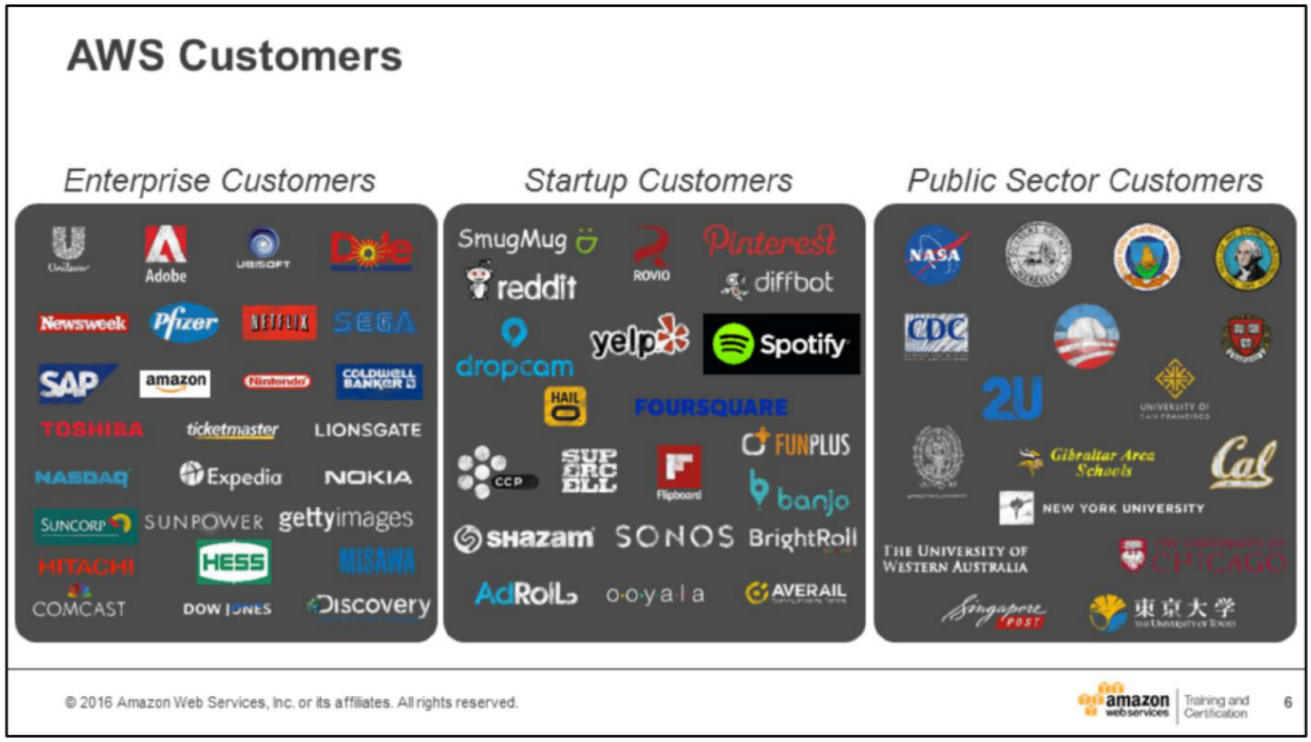


AWS has been continually expanding its services to support virtually any cloud workload. It now has more than 50 services that range from compute, storage, networking, database, analytics, application services, deployment, management and mobile. In 2015, AWS launched 722 new features and/or services for a total of 1,950 new features and/or services since its inception in 2006.

Innovation is in our DNA, and our structure and approach to product development and delivery is fundamentally different than other IT vendors. We have decentralized, autonomous development teams who are working directly with customers. They are empowered to develop and launch based on what they learn from interactions with customers. We iterate products continuously, and the newest/latest is instantly available to customers. No need to upgrade, deploy, or migrate. When a feature or enhancement is ready, we “push” it out, and it is instantly available to any customer that uses that service.

This approach also enables us to very rapidly introduce and iterate on new services.

As of 1 February 2016, AWS has launched 1,950 new services as well as features and updates to existing services.



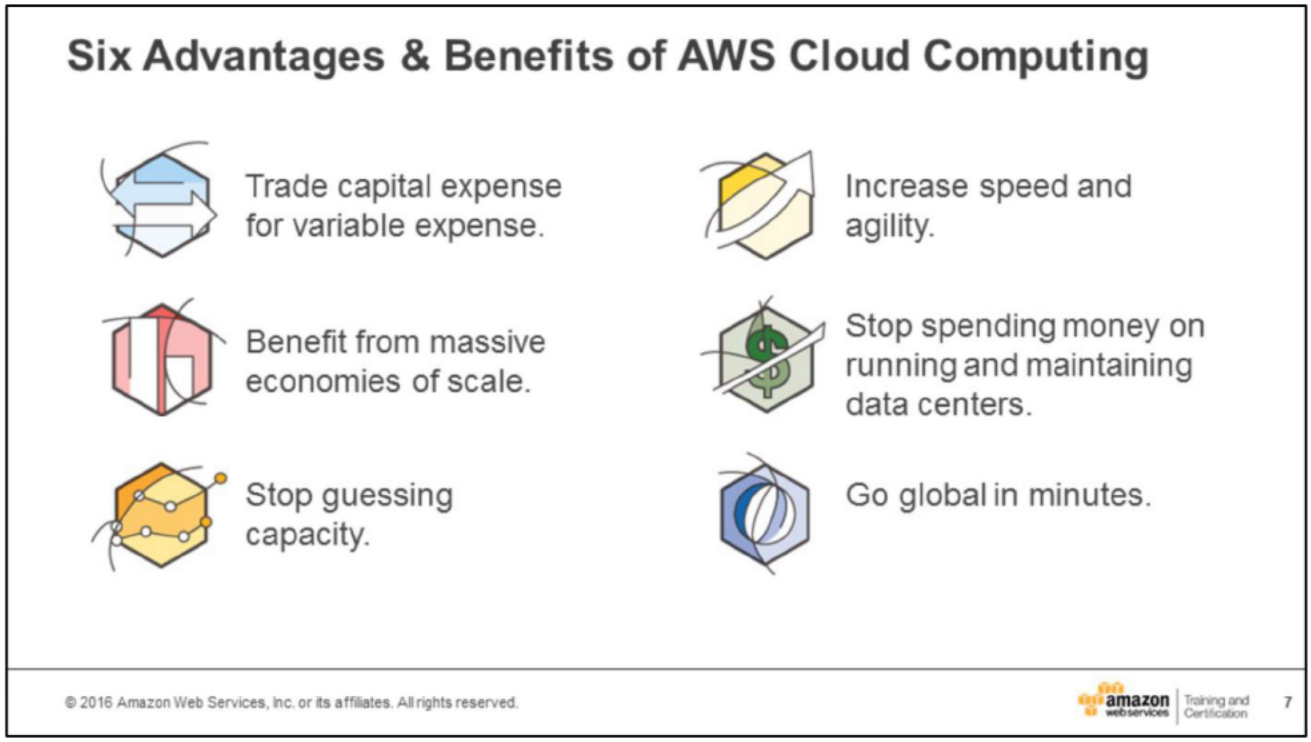
Enterprise Customers: Enterprise cloud computing with AWS can help IT increase innovation, agility, and resiliency; all while reducing cost. With AWS, you can build enterprise cloud solutions quickly and without a big up-front investment. The free tier allows you to prototype virtually any application for free.

Startup Customers: Our innovations free you to scale quickly, go to market faster, control costs, and stay lean. AWS Activate is a free program with resources for startups to get the most out of AWS from day one.

Public Sector Customers: AWS offers scalable, cost-effective cloud services that public sector customers can use to meet mandates, reduce costs, drive efficiencies, and accelerate innovation.

For more information, see:

- http://aws.amazon.com/contract-center/
- Enterprise Cloud Computing - http://aws.amazon.com/enterprise/
- Free Tier - https://aws.amazon.com/free/
- Apply online with a Self-Starter Package - http://aws.amazon.com/activate/self-starters/
- For more information about Startups on Amazon Web Services, see - http://aws.amazon.com/start-ups/



Cloud computing provides a simple way to access servers, storage, databases, and a broad set of application services over the Internet. AWS owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.

Trade capital expense for variable expense: Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consume.

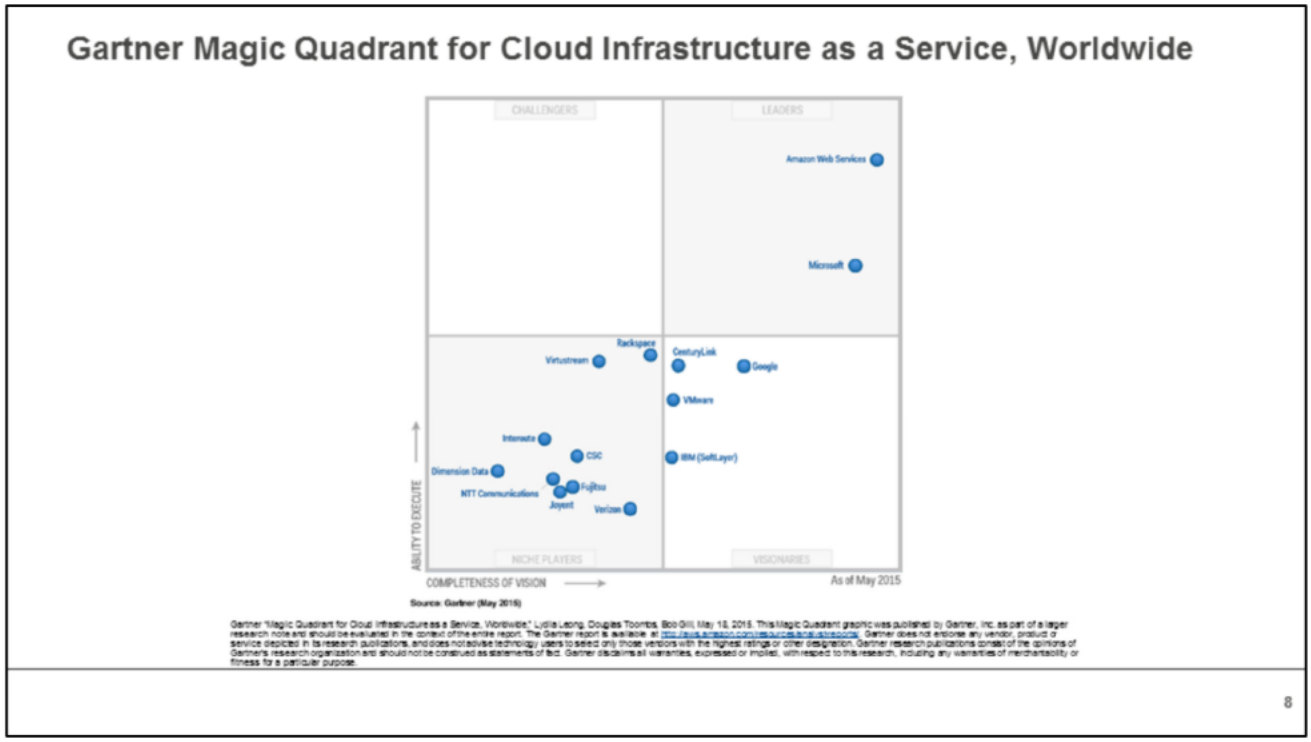
Benefit from massive economies of scale: By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as Amazon Web Services can achieve higher economies of scale, which translates into lower pay as you go prices.

Stop guessing capacity: Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often either end up sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little as you need, and scale up and down as required with only a few minutes' notice.

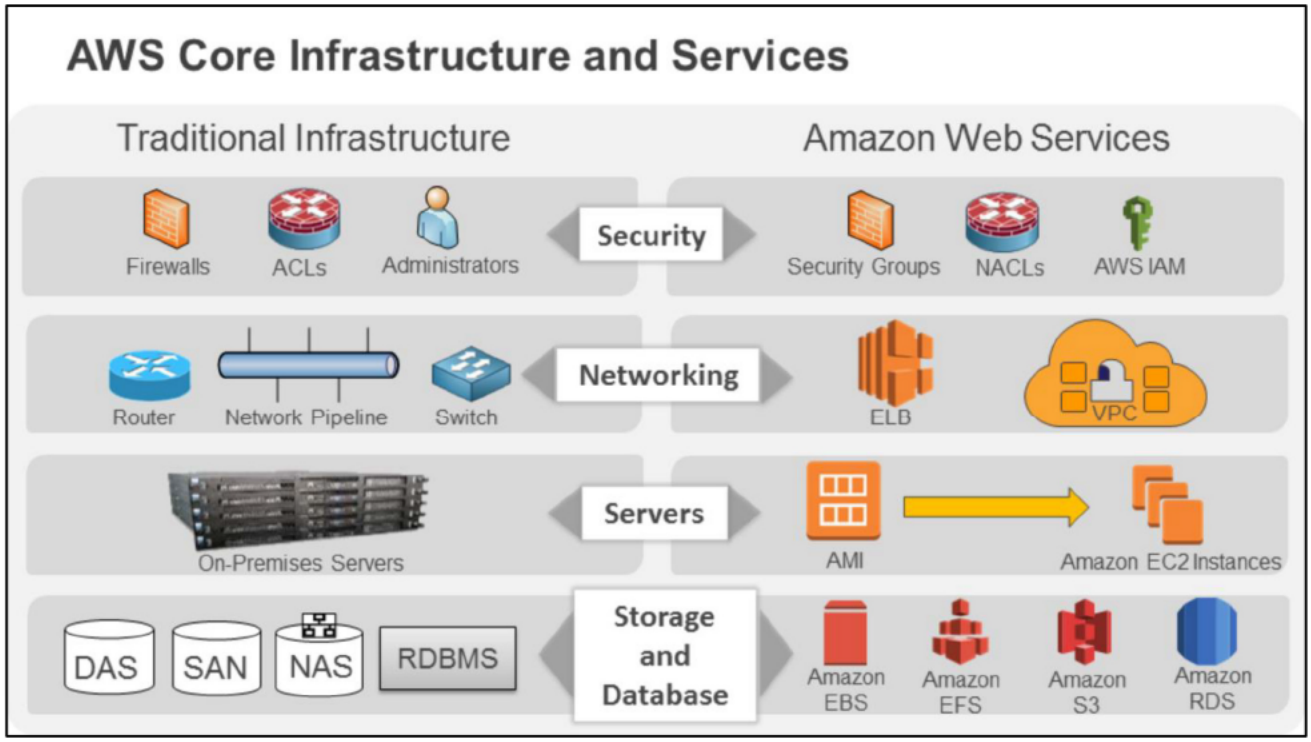
Increase speed and agility: In a cloud computing environment, new IT resources are only ever a click away, which means you reduce the time it takes to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, since the cost and time it takes to experiment and develop is significantly lower.

Stop spending money on running and maintaining data centers: Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking and powering servers.

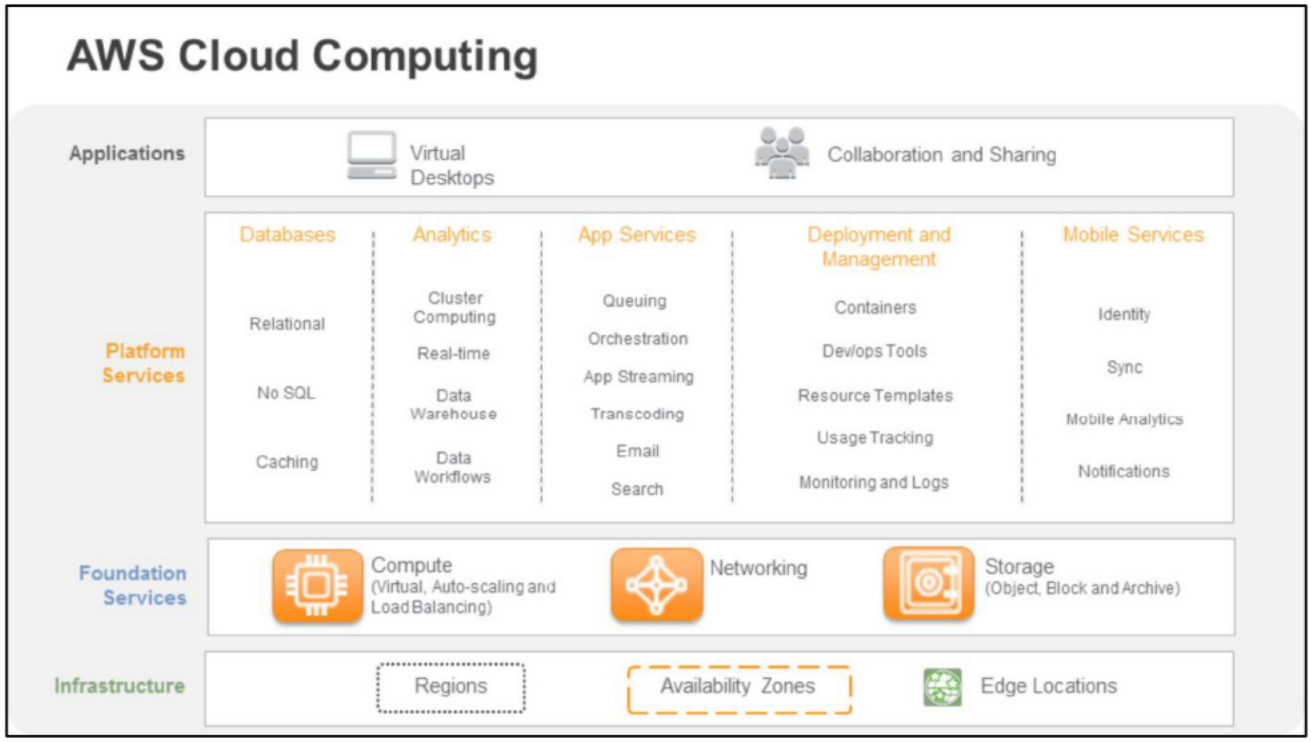
Go global in minutes: Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide a lower latency and better experience for your customers simply and at minimal cost.



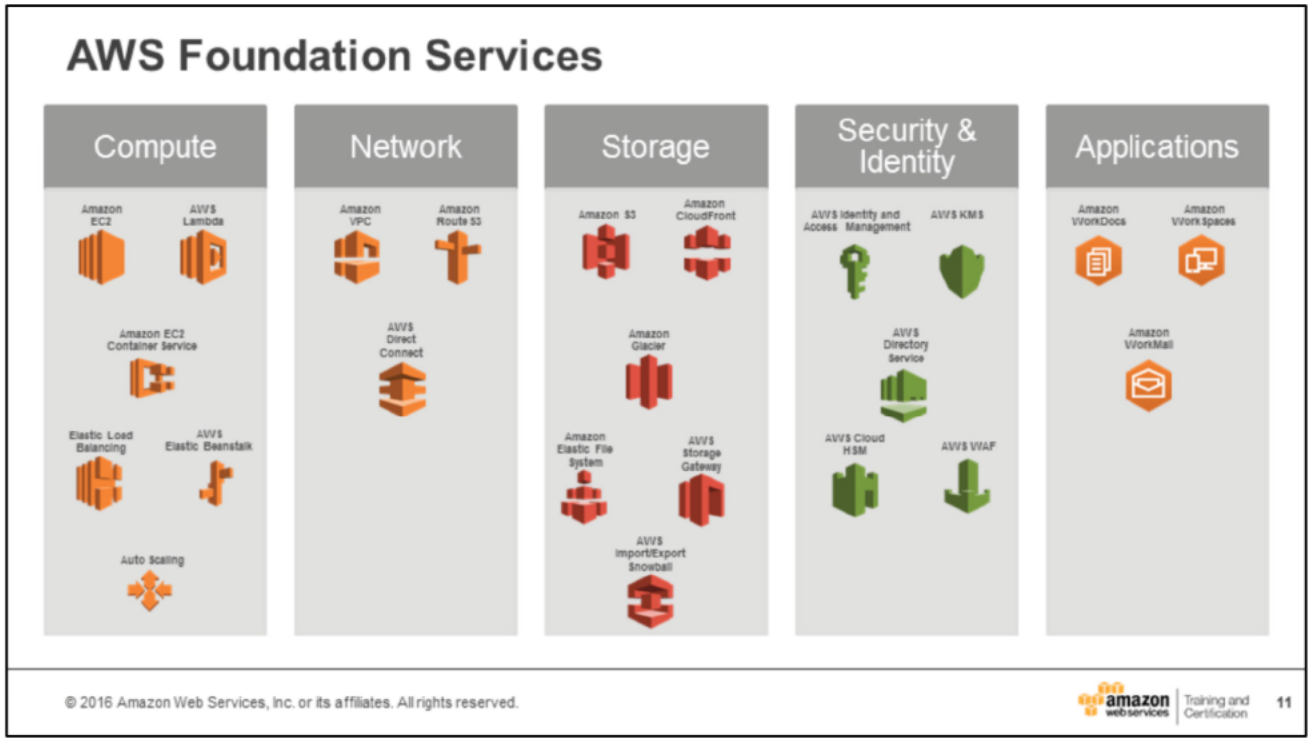
The Gartner report is available at http://aws.amazon.com/resources/analyst-reports/



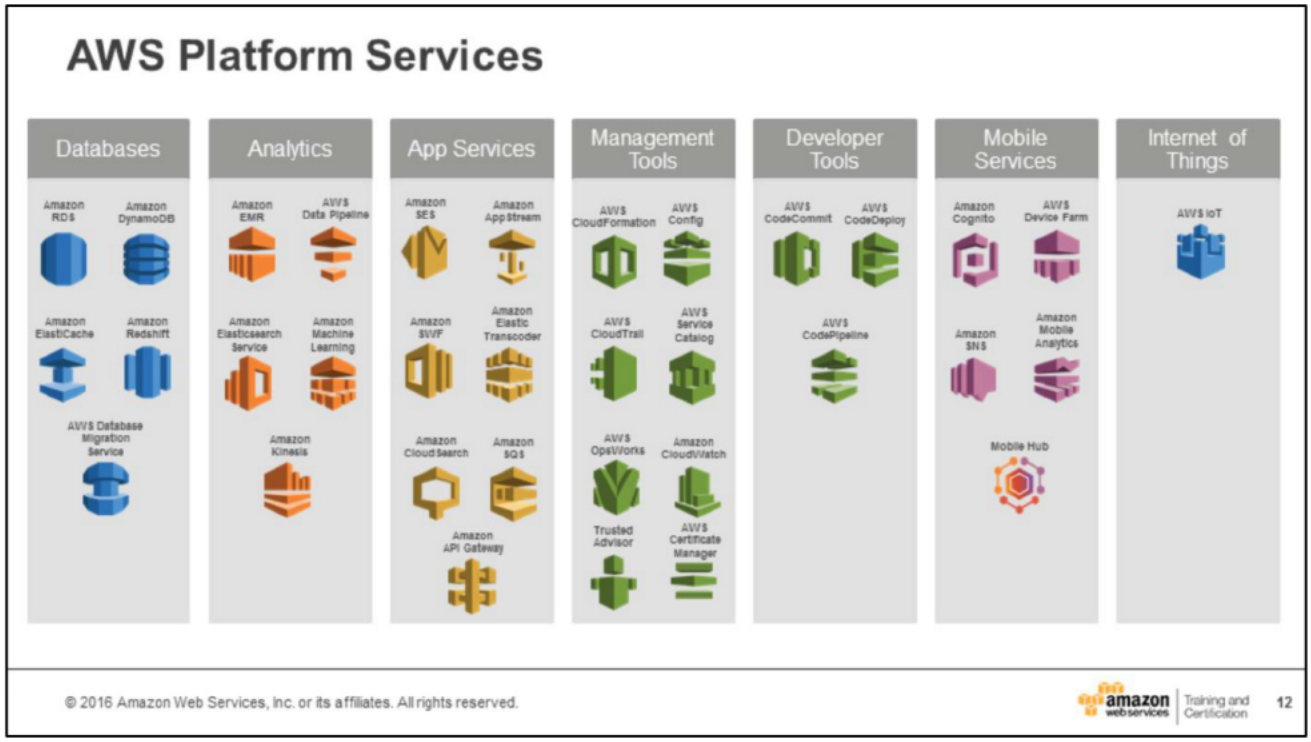
Many of our services have analogs in the traditional IT space and terminology. This side-by-side comparison shows how AWS products and services relate to a traditional infrastructure.



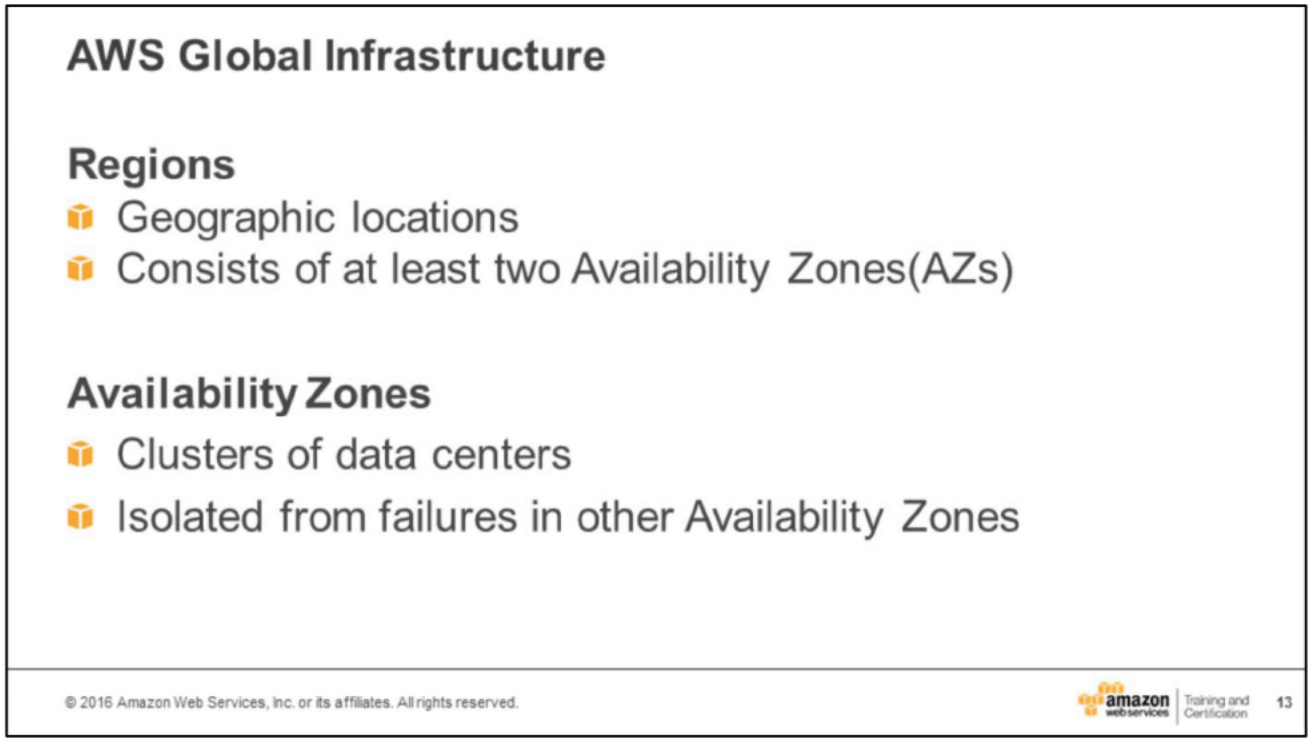
AWS cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the Internet. AWS owns and maintains the network-connected hardware required for these application services, while you provision and use what you need.



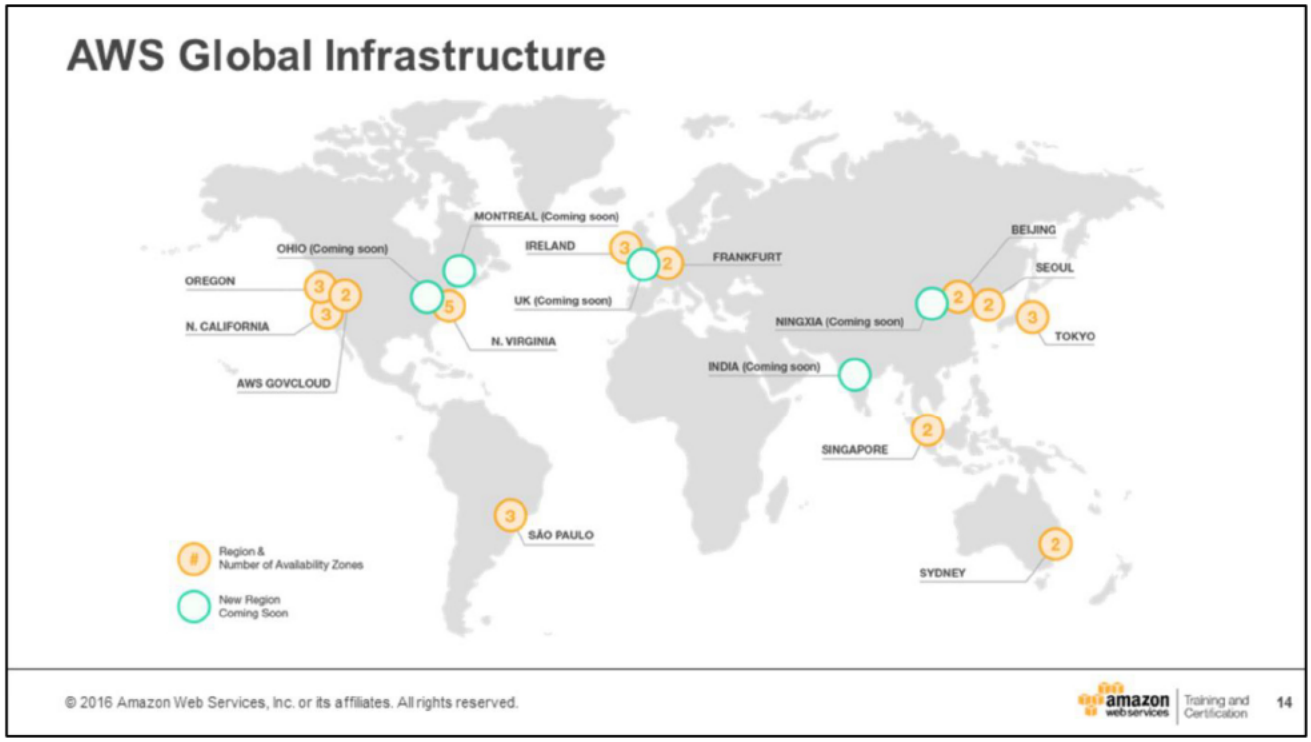
AWS Foundation Services are categorized as shown in the slide.



AWS offers a rich set of platform services that enable you to process a massive number of events from different sources, effectively work with databases, and more.



AWS Regions are geographic locations that contain multiple Availability Zones (AZs). Availability Zones consist of data centers clustered in a region. Each Availability Zone is engineered to be isolated from failures in other Availability Zones.



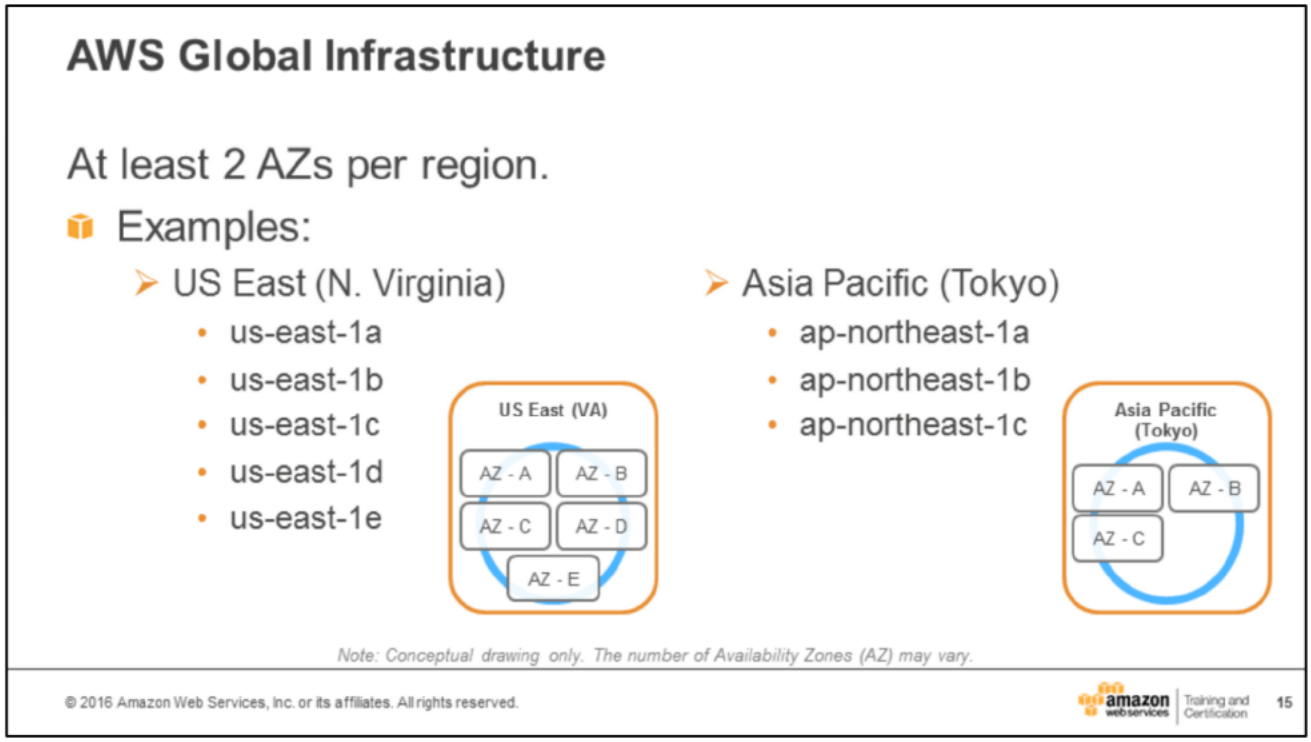
AWS is steadily expanding its global infrastructure to help customers achieve lower latency and higher throughput, and to ensure that your data resides only in the region you specify. As you and all customers grow their businesses, AWS will continue to provide infrastructure that meets your global requirements.

As of January 2016, AWS has 12 geographic Regions with 32 Availability Zones. The AWS GovCloud (US) Region is an isolated region designed to allow US government agencies and customers to move sensitive workloads into the cloud by addressing their specific regulatory and compliance requirements. AWS products and services are available by region so you may not see all regions available for a given service.

You can run applications and workloads from a region to reduce latency to end-users while avoiding the up-front expenses, long-term commitments, and scaling challenges associated with maintaining and operating a global infrastructure. In 2016, the AWS Global Infrastructure will expand with at least 10 new Availability Zones in new geographic Regions including Ohio in North America, Ningxia in China, India, Korea, and the United Kingdom.

For more information, see:

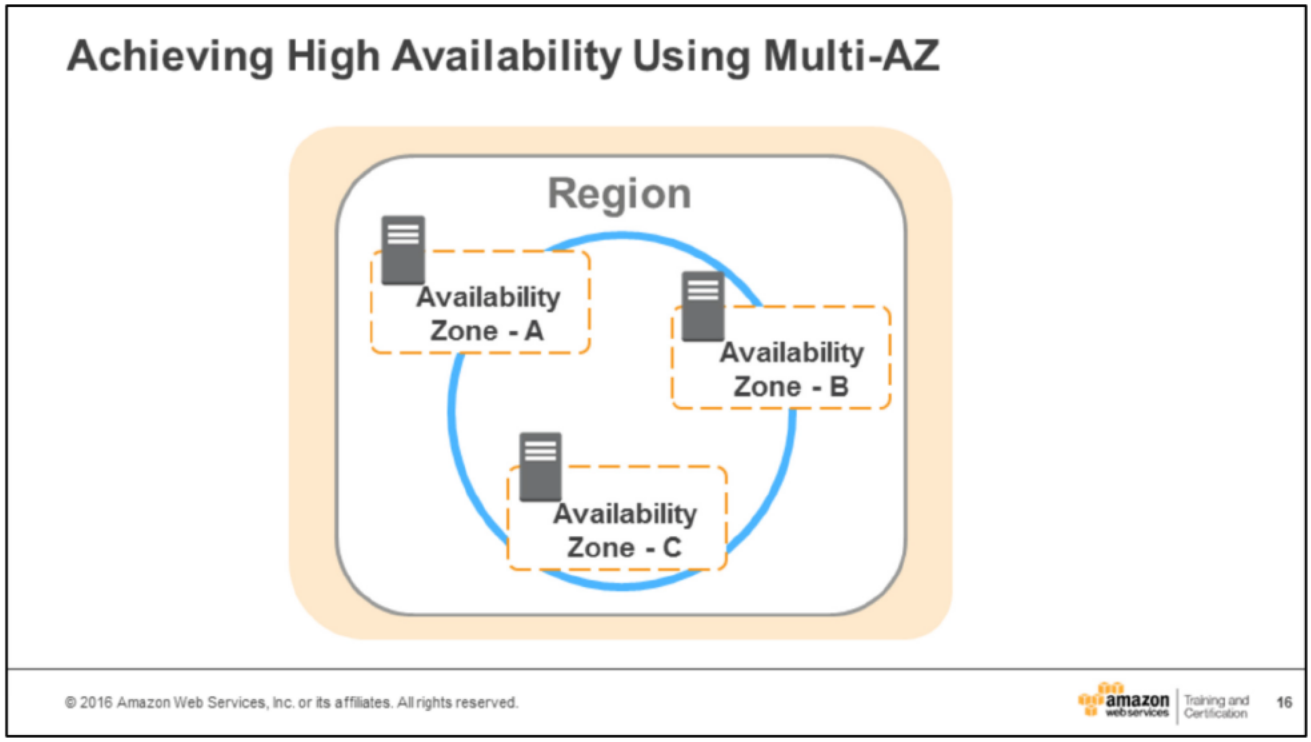
http://aws.amazon.com/about-aws/global-infrastructure/



Each region is a separate geographic area that has multiple locations isolated from each other known as Availability Zones (AZ). Each AZ is isolated, but the AZs in a region are connected through low-latency links. Where natural disasters or fault lines are a consideration, AWS isolates its Availability Zones so that they are not easily affected at the same time. For example, where earthquakes are a problem AWS would not build two AZs on the same fault line. When you launch an instance, you can select an AZ or let AWS choose one for you. If you distribute your instances across multiple AZs and one instance fails, you can design your application so that an instance in another AZ can handle requests.

For more information, see:

- http://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html
- https://www.amazonaws.cn/en/



AWS highly recommends provisioning your compute resources across multiple Availability Zones. If you have multiple instances, you can run them across more than one AZ and get added redundancy. If a single AZ has a problem, all assets in your second AZ will be unaffected.

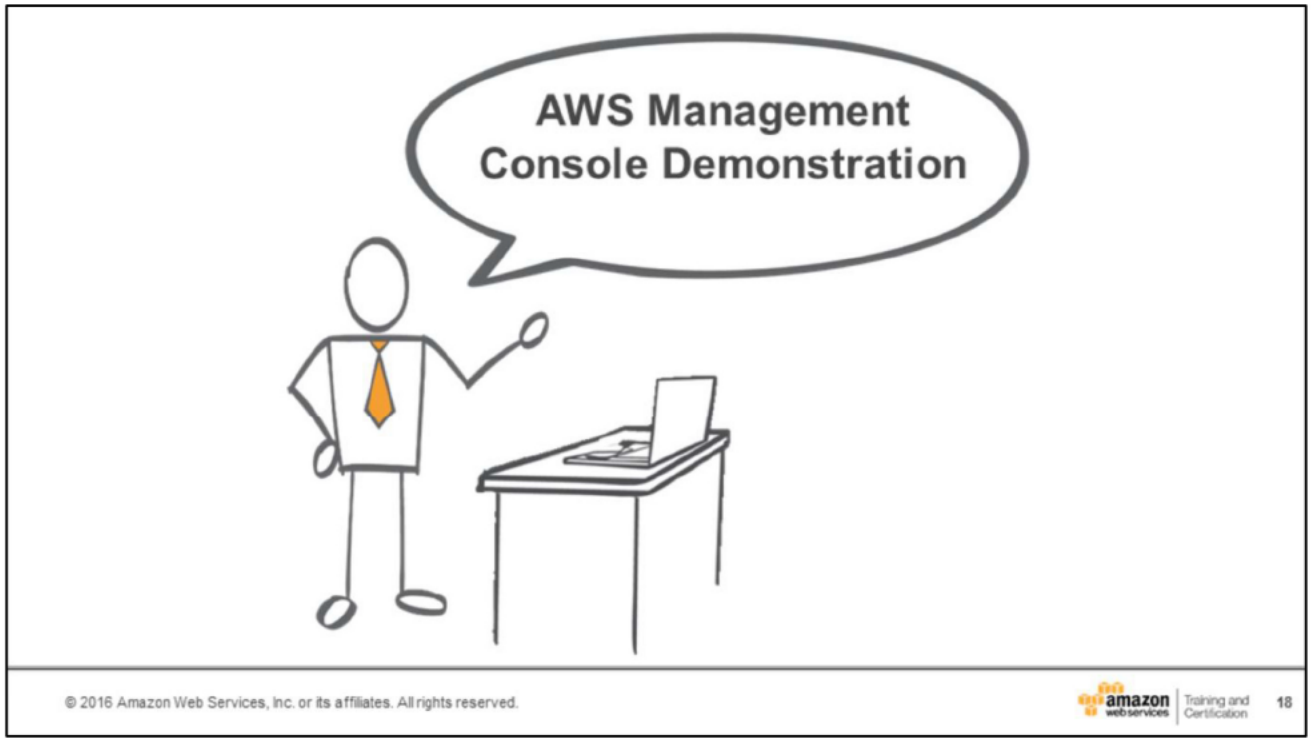
AWS Global Infrastructure

50+ AWS Edge Locations:

- Local points-of-presence commonly supporting AWS services like:
  - Amazon Route 53 
  - Amazon CloudFront 

amazon web services | Training and Certification 17

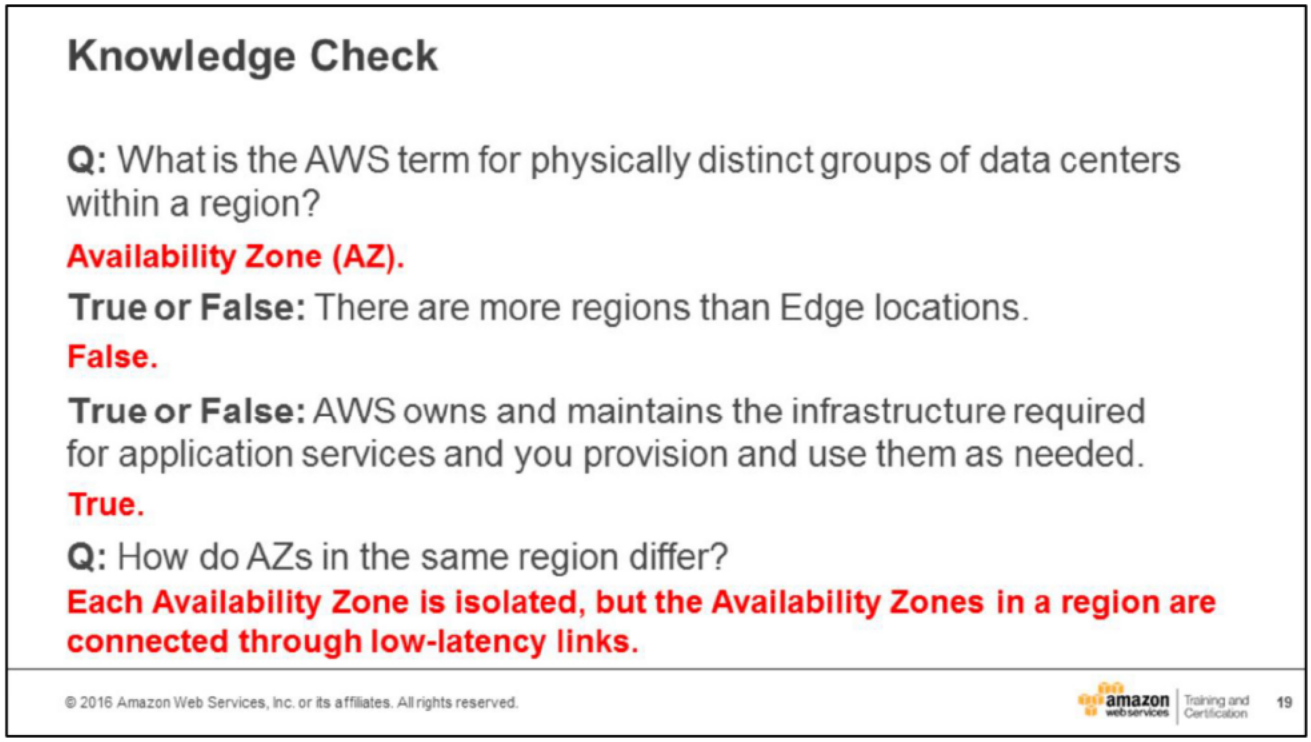
Edge locations help lower latency and improve performance for end users.



Let's start by taking a look at the AWS Management Console so you can get familiar with the navigation:

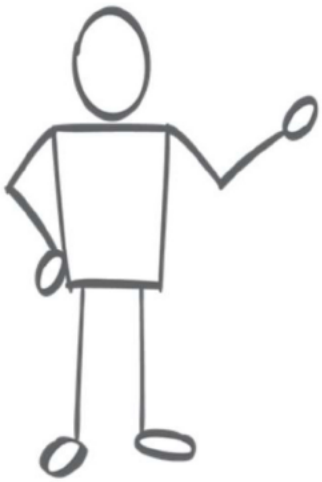
- Selecting a region.
- Dragging and dropping service icons at the top for quick use.
- Verifying what is included in the left navigation menu pane.

AWS Management Console: https://console.aws.amazon.com/console/home

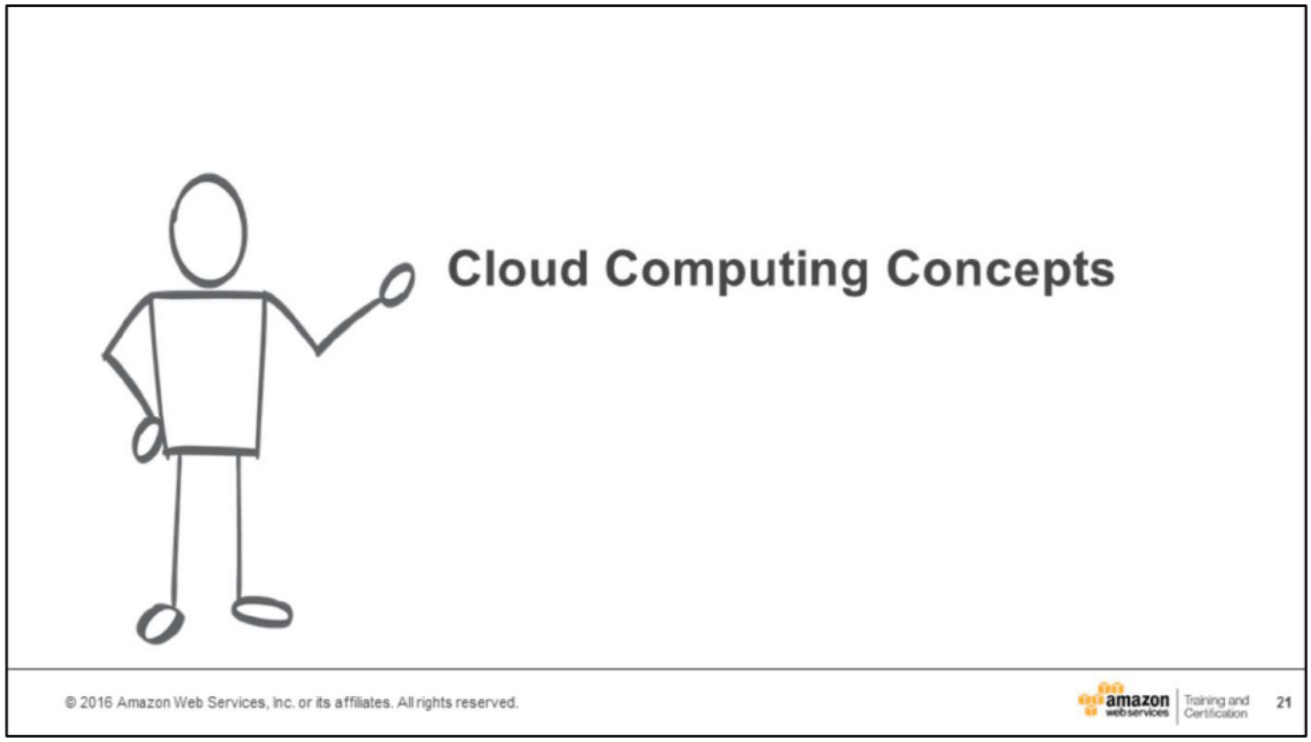


In this module, you learned the history of Amazon and AWS, the global infrastructure of AWS, and how to navigate the AWS Management Console.

Test out some of your new skills!



Appendix



Understand essential characteristics of cloud computing.

What is cloud computing?

Cloud computing is on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing.

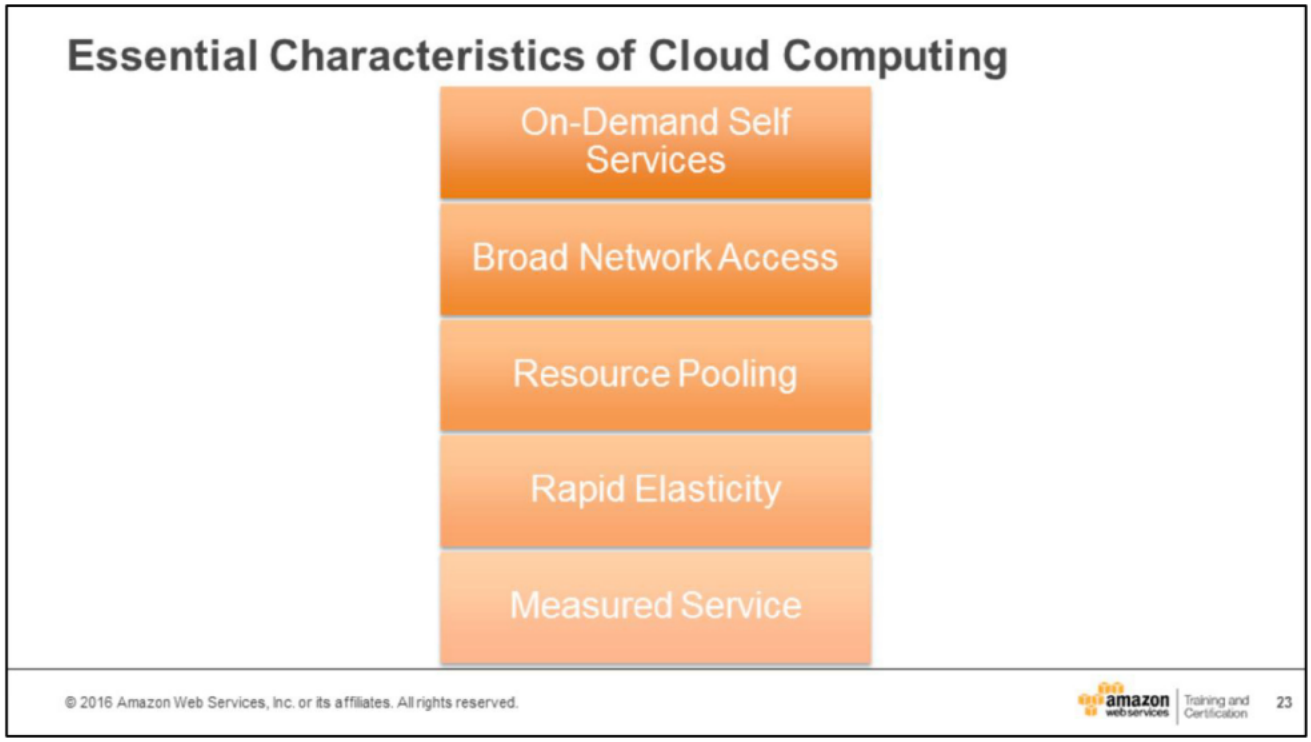
amazon web services | Training and Certification 22

Cloud computing is a common term for a variety of computing concepts that involve large numbers of computers that are connected through a real-time communication network, like the Internet.

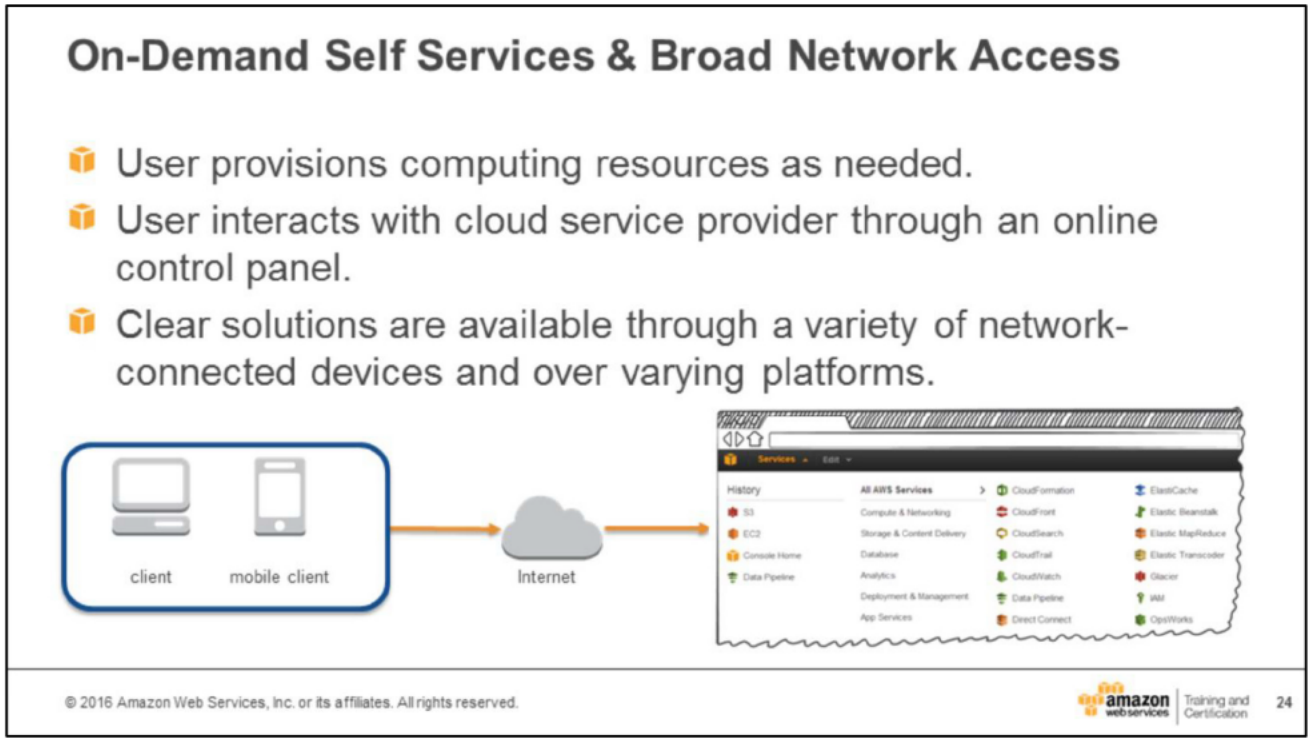
Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction in system diagrams for the complex infrastructure it contains. Cloud computing entrusts remote services with a user's data, software, and computation. Cloud computing allows you to access as many resources as you need, almost instantly, and only pay for what you use.

Instead of buying, owning, and maintaining your own data centers and servers, organizations can acquire technology such as compute power, storage, databases, and other services on an as-needed basis.

Word cloud created at: http://www.tagxedo.com/



Cloud computing is characterized by five characteristics: on-demand self services, broad network access, resource pooling, rapid elasticity, and measured service.



On-demand self services are an essential characteristic of cloud computing. The user provisions computing resources as needed and interacts with the cloud service provider through on online control panel, such as the AWS Management Console.

Broad network access is another characteristic of cloud computing. Clear solutions are available through a variety of network-connected devices and over varying platforms.

Resource Pooling

Securely separate resources to service multiple customers.

A third characteristic of cloud computing is resource pooling. Cloud computing solutions securely separate resources to service multiple customers.

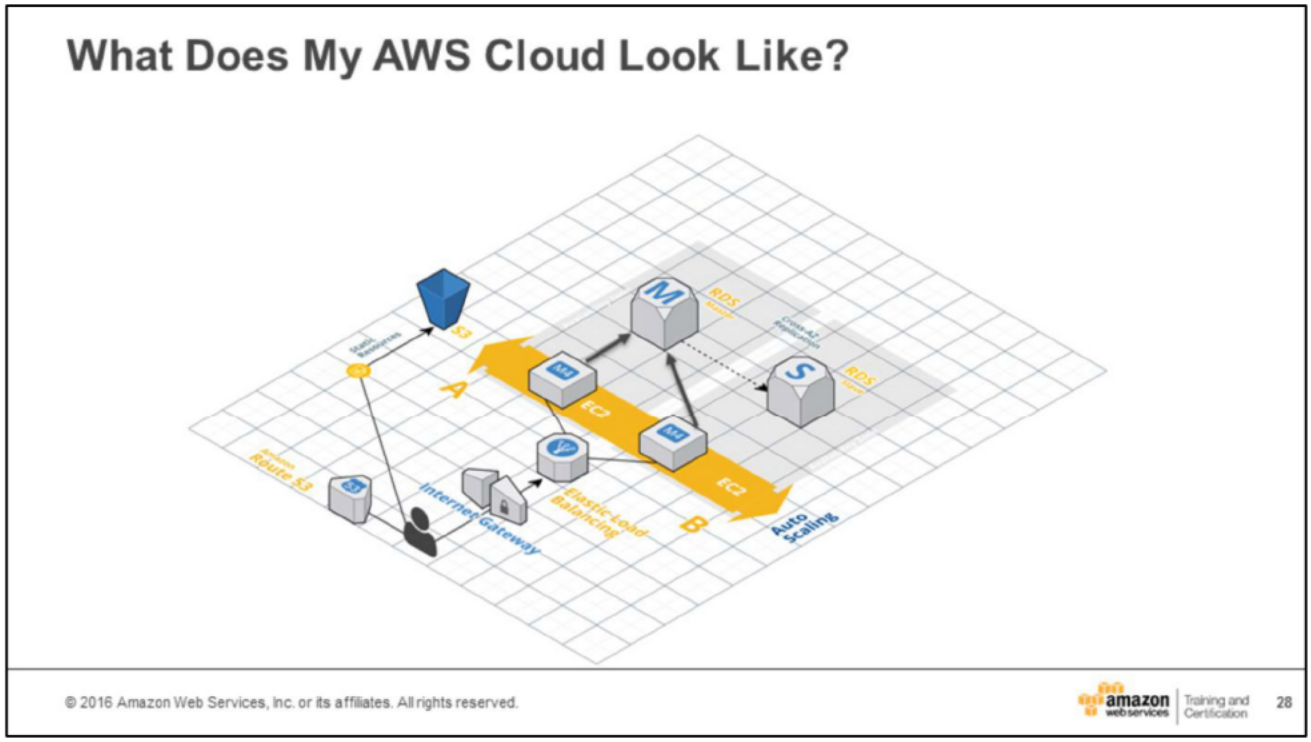
Rapid Elasticity

Resources are quickly scalable and flexible based on business needs.

Customers used to over provision to ensure they had enough capacity to handle their business operations at the peak level of activity. Cloud computing allows customers to provision the number of resources that they actually need, knowing they can instantly scale up or down along with the needs of their business, which also reduces cost and improves the customer's ability to meet their user's demands.

Pay for service you use as you go.

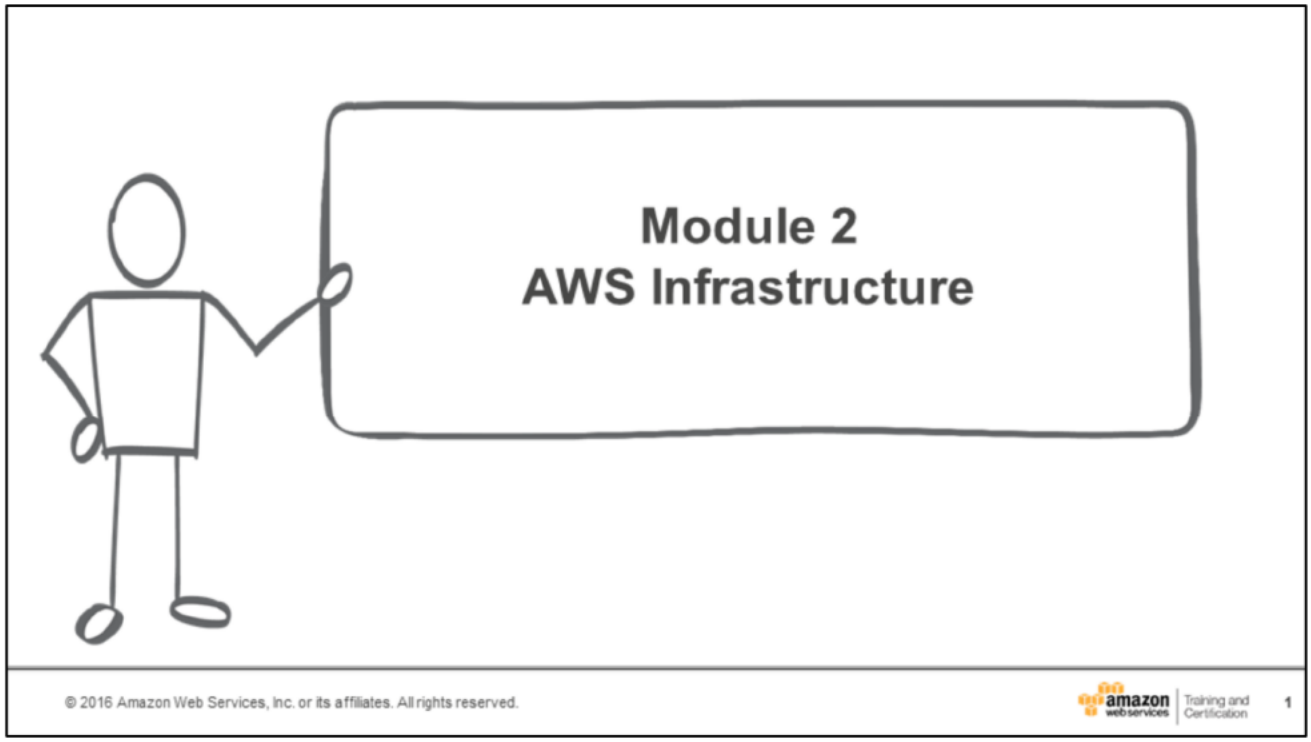
The slide shows an analogy that helps explain cloud computing. Electricity services are a utility that you pay for on-demand: you pay for what you use. You plug electrical appliances into a vast electrical grid that is managed by the power company to get a low cost, reliable power supply. This power is available to you from the power company with much greater efficiency than you could generate on your own.

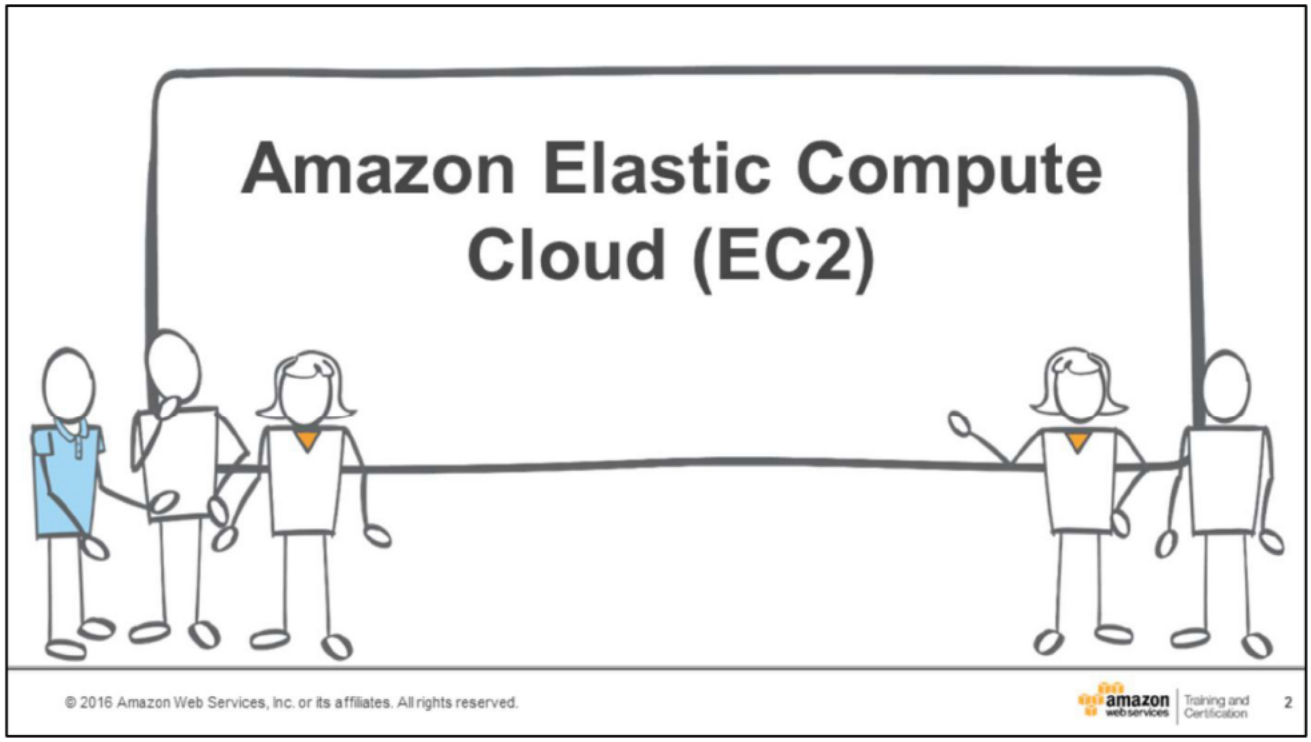


The slide shows the architecture of a highly scalable and reliable web application on AWS. This is an example of one possible configuration among millions.

For more information, see:

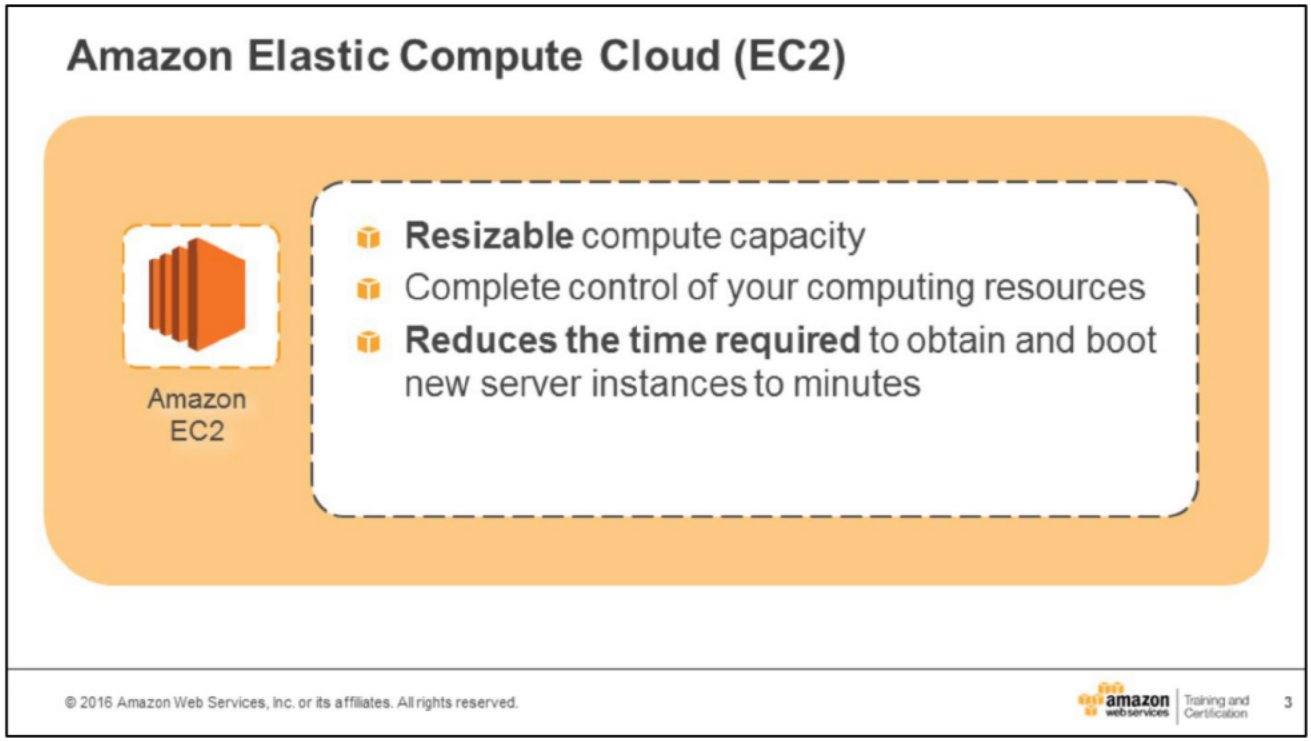
http://docs.aws.amazon.com/gettingstarted/latest/wah-linux/web-app-hosting-intro.html





Understand Amazon Elastic Compute Cloud (EC2) concepts including:

- Instances vs. servers
- Types and families
- Ephemeral vs. persistent storage (root instance volumes)
- Amazon Machine Images (AMIs)
- Bootstrapping/user data



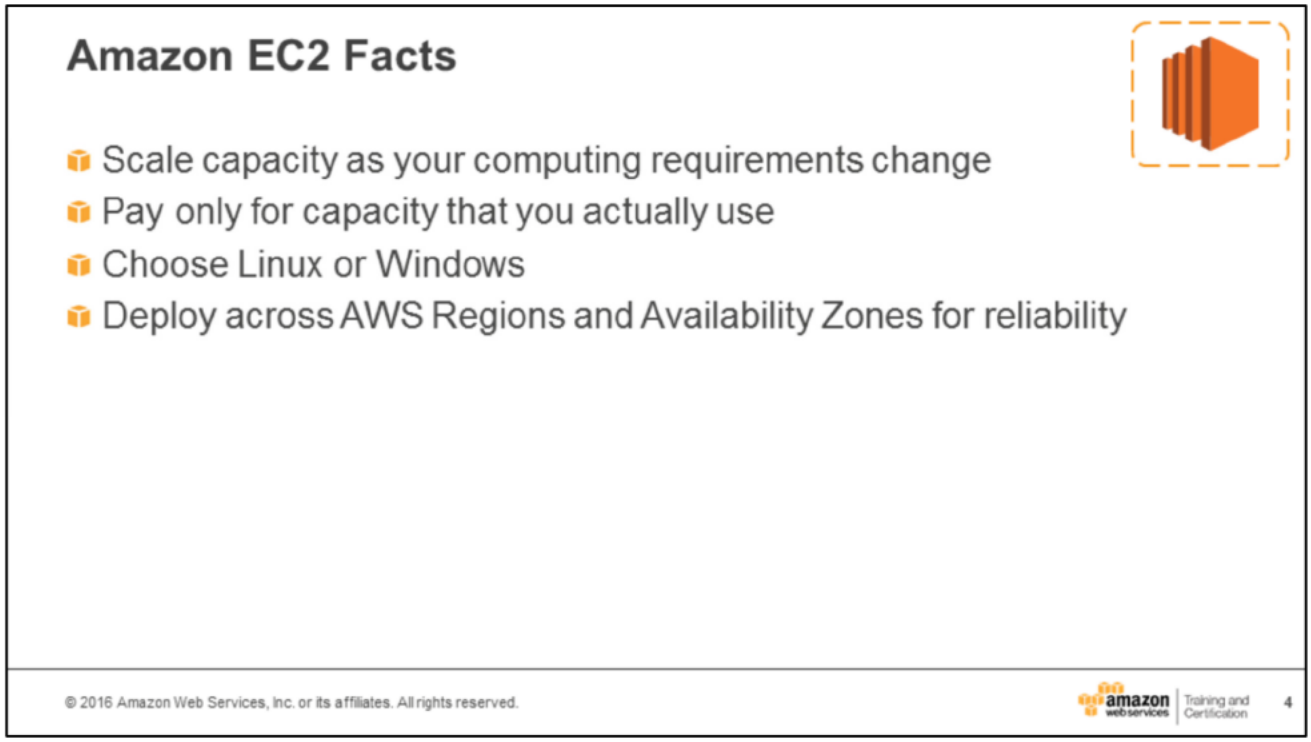
Amazon EC2 instances are virtualized servers in Amazon's data centers.

Amazon EC2 is designed to make web-scale computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and allows you to run on Amazon's proven computing environment.

Amazon EC2 reduces the time required to obtain and boot new server instances, allowing you to quickly scale capacity as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides the tools for you to build failure-resilient applications and isolate the applications from common failure scenarios.

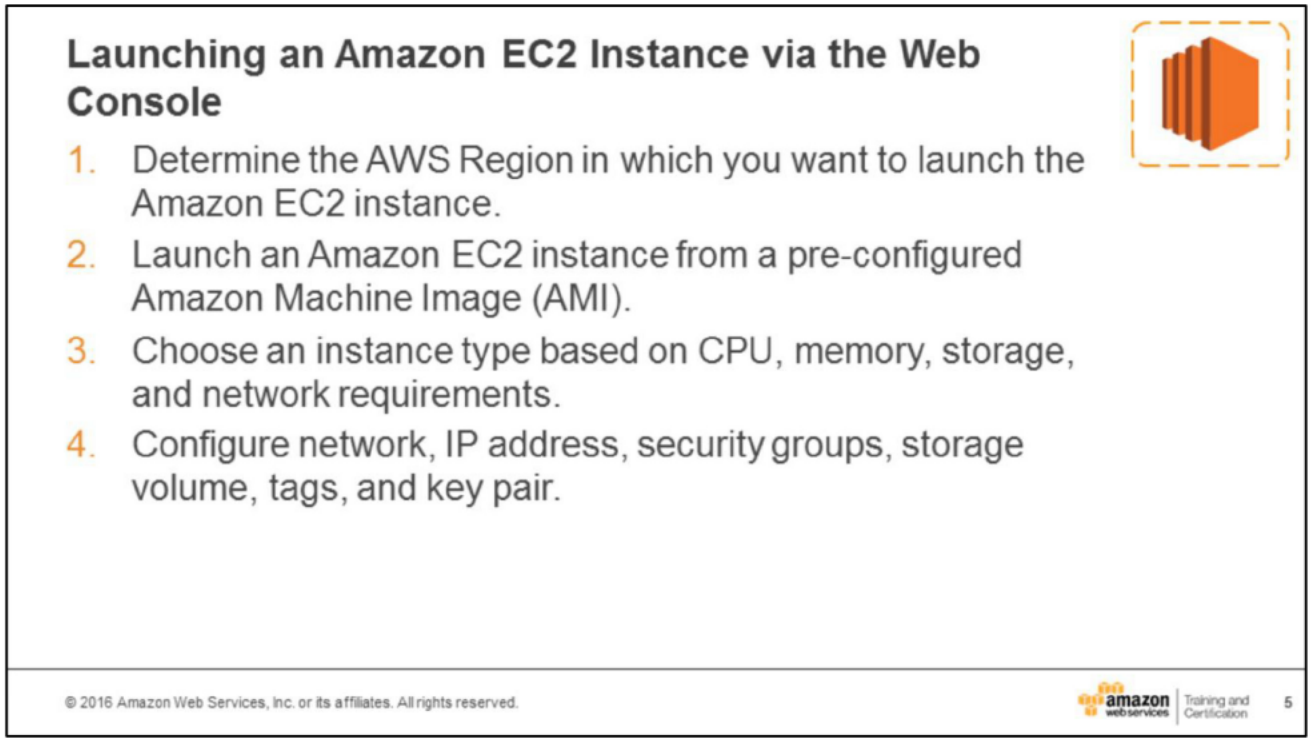
For more information, see:

https://aws.amazon.com/ec2/

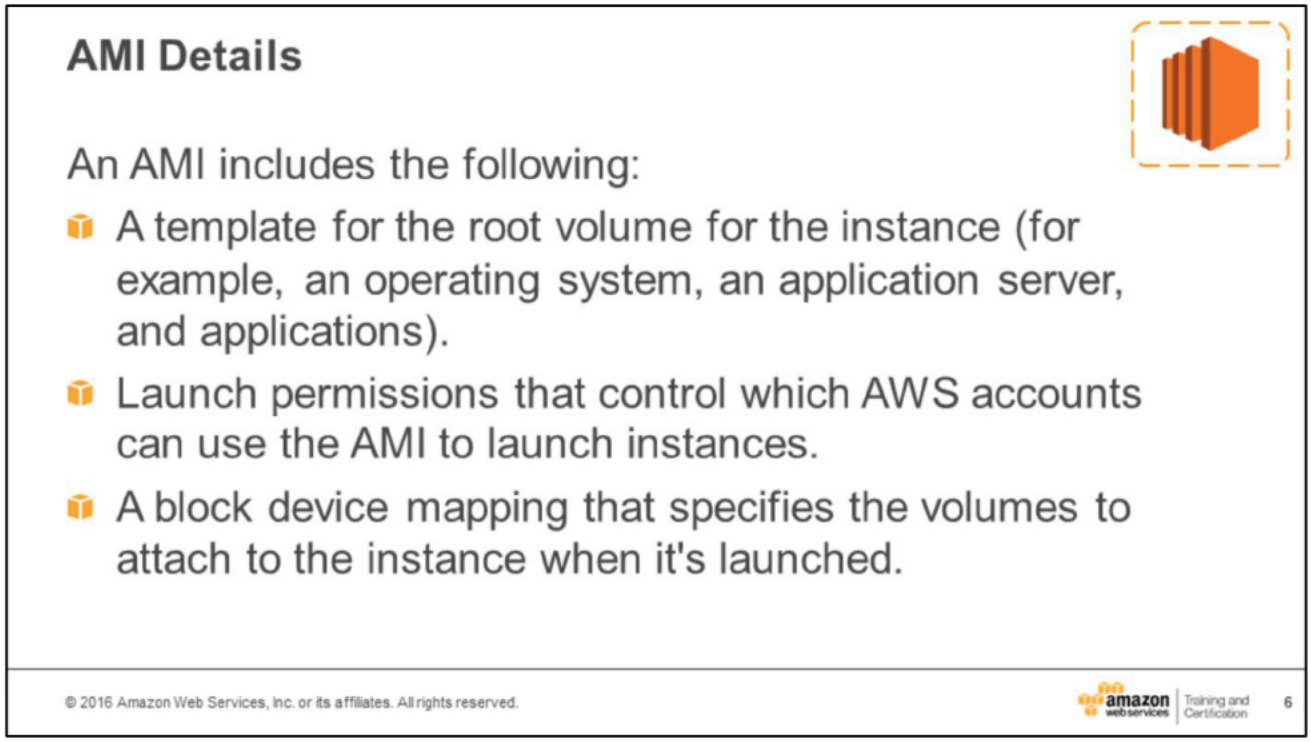


Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you need.

You have the ability to programmatically scale your computing capacity as your requirements change. You pay only for capacity that you actually use and can choose Linux or Windows. You can leverage the AWS global infrastructure to deploy across regions and Availability Zones (AZs) for reliability.



Before you create your first Amazon EC2 instance think about which region you want to have that instance in. The AMI comes pre-installed with many AWS API tools as well as CloudInit. AWS API tools enable scripting of important provisioning tasks from within an Amazon EC2 instance. AMIs are like building blocks of EC2 instances. They are templates of a computer's volumes. AMIs can have public or private access. You can also create gold master images of your Amazon EC2 infrastructure, which allow you to decrease your boot times.



An AMI is a template that contains a software configuration such as an operating system, application server, and applications. You use an AMI to launch an instance, which is the copy of the AMI running as a virtual server on a host computer in Amazon's data center. You can launch as many instances as you want from an AMI. You can also launch instances from as many AMIs as you need.

You can create your own AMI by customizing the instance that you launch from a public AMI and then saving the configuration as a custom AMI for your own use. You can also buy, share, and sell AMIs.

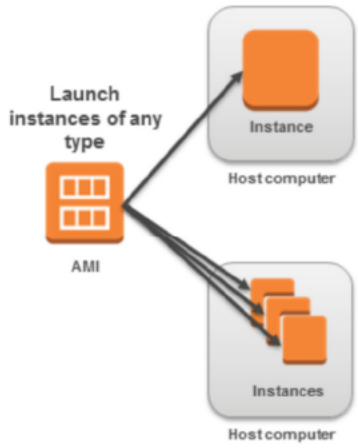
For more information, see:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

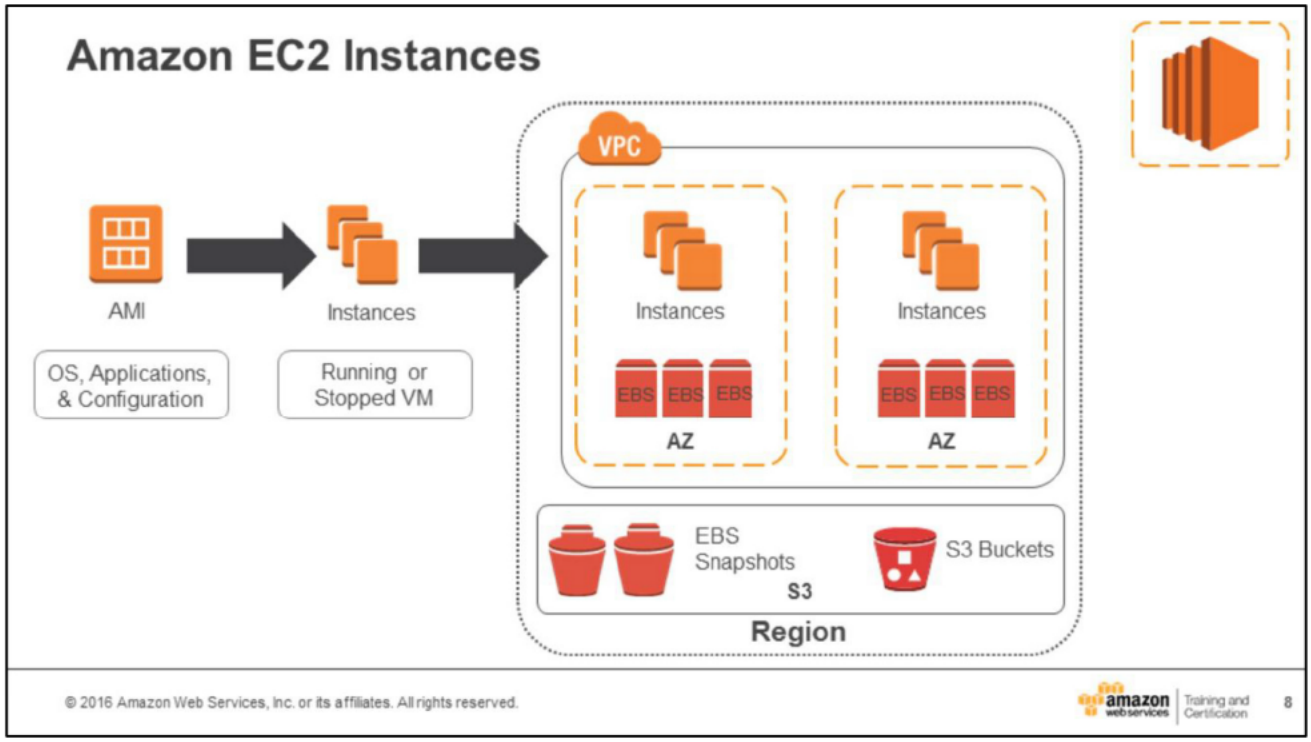
Instances and AMIs

Select an AMI based on:

- Region
- Operating system
- Architecture (32-bit or 64-bit)
- Launch permissions
- Storage for the root device



You select an AMI based on region, operating system, architecture, launch permissions, and storage for the root device. Launch permissions determine availability of an AMI and are either public (the owner grants launch permissions to all AWS accounts), explicit (the owner grants launch permissions to specific AWS accounts), or implicit (the owner has implicit launch permissions for an AMI).

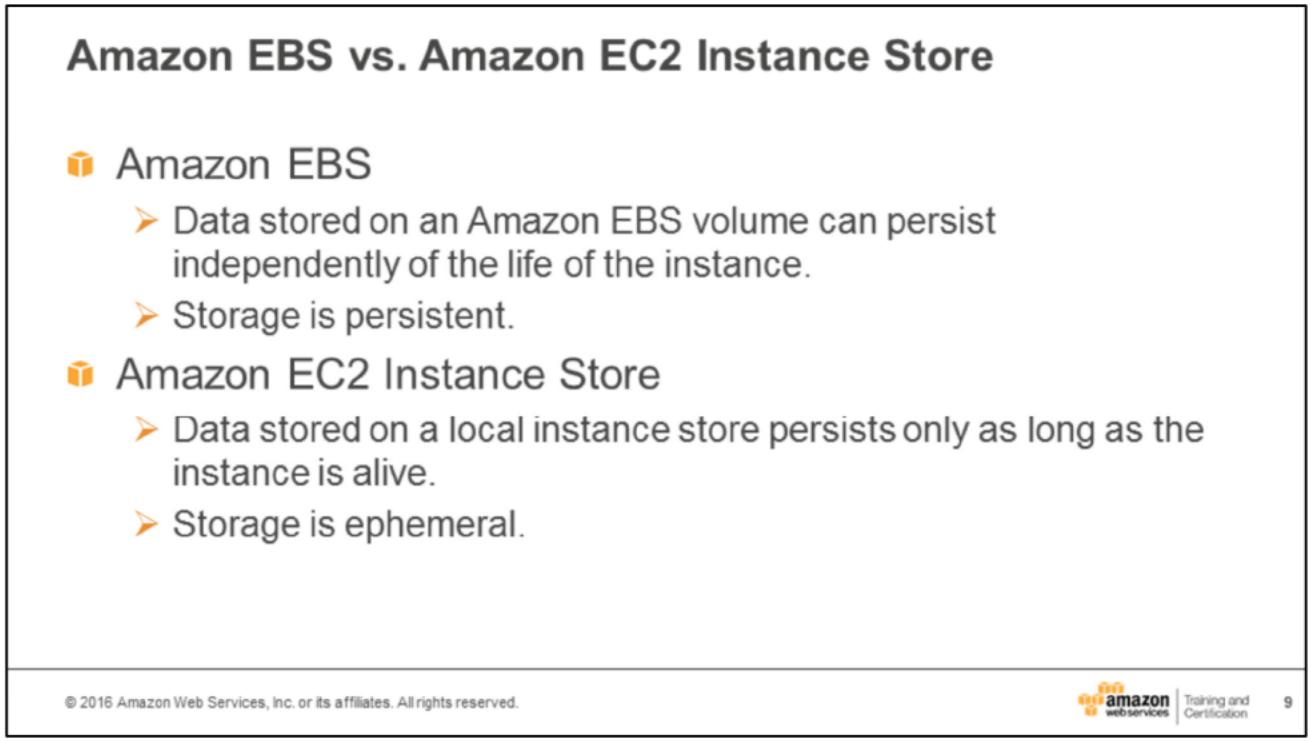


You can launch multiple instances of different types from a single AMI when launching an EC2 instance. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. Your instance keeps running until you stop or terminate it, or until it fails.

Instances are deployed in the Amazon EC2 public cloud or the Amazon Virtual Private Cloud in an Availability Zone (AZ) within a Region. You can configure security and network access on your Amazon EC2 instance.

Customers can deploy to multiple AZs within a Region. You choose which instance types you want, and then start, terminate, and monitor as many instances of your AMI as needed, using the web service APIs or the variety of management tools provided.

Amazon EC2 instances can leverage Amazon Elastic Block Store volumes in each Availability Zone. Determine whether you want to run in multiple locations, utilize static IP endpoints, or attach persistent block storage to your instances. Amazon EBS volumes can be saved using “snapshots.” Additionally, Amazon S3 buckets can be used to store data objects needed by Amazon EC2 instances. Pay only for the resources that you actually consume, like instance-hours or data transfer.



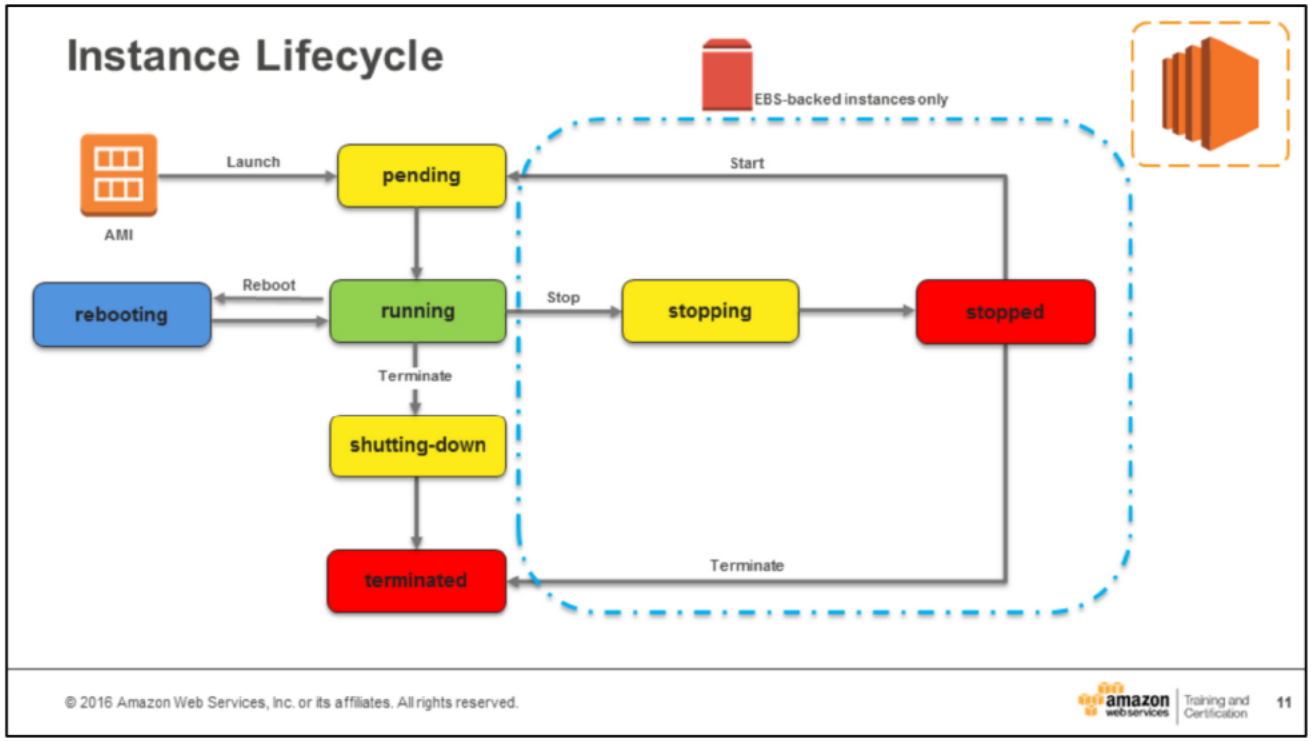
Use the local instance store only for temporary data. For data requiring a higher level of durability, use Amazon EBS volumes or back up the data to Amazon S3, topics that are both discussed later in this module. If you are using an Amazon EBS volume as a root partition, set the Delete on termination flag to "No" if you want your Amazon EBS volume to persist outside the life of the instance.

AMI Types - Storage for the Root Device

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually < 1 minute	Usually < 5 minutes
Size limit	16 TiB	10 GiB
Data persistence	The root volume is deleted when the instance terminates. Data on any other Amazon EBS volumes persists after instance termination.	Data on any instance store volumes persists only during the life of the instance.
Charges	Instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	Instance usage and storing your AMI in Amazon S3.
Stopped state	Can be stopped.	Cannot be stopped.

AMIs are either Amazon Elastic Block Storage (EBS)-backed or backed by instance store. When an AMI is EBS-backed, this means that the root device for an instance is an EBS volume created from an EBS snapshot. When an AMI is instance-store backed, this means that the root device for the instance was created from a template stored in Amazon S3. Key differences between both categories of AMIs are shown in the slide.

Storage will be discussed more extensively later in the modules.

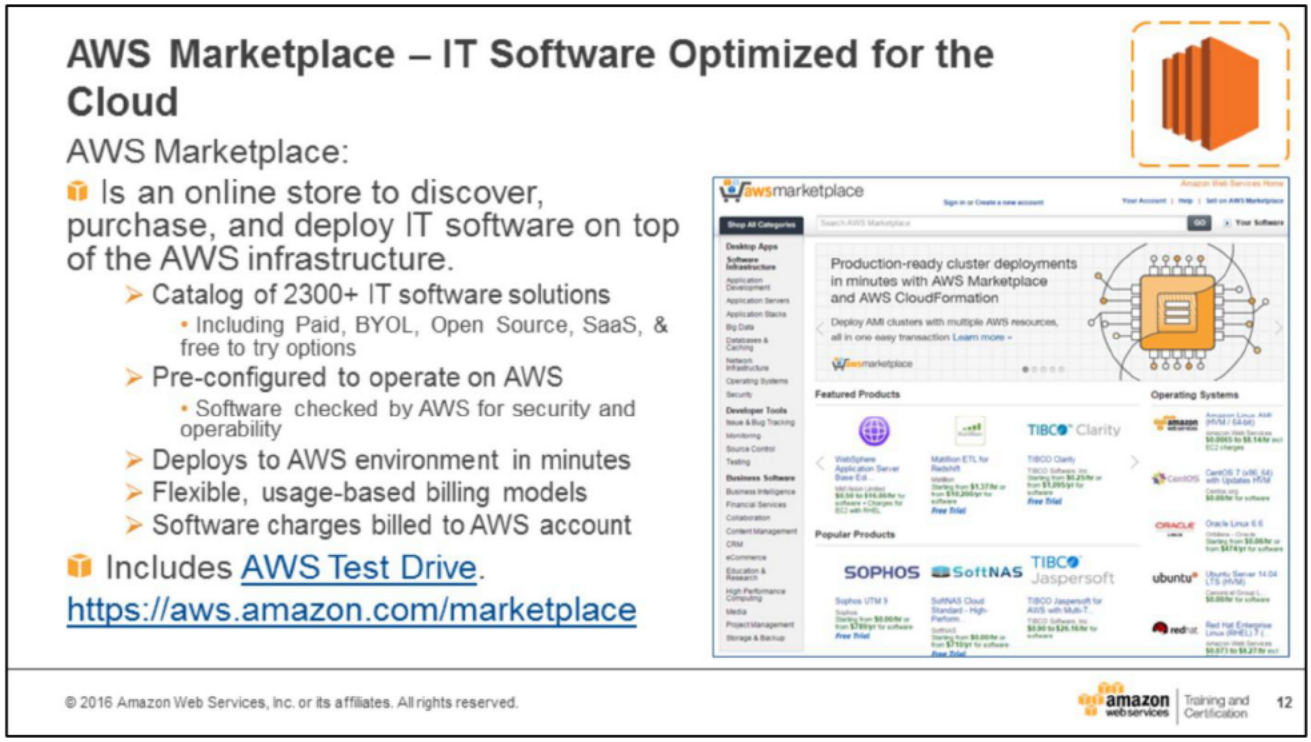


The slide shows the lifecycle of an instance launched from an AMI. Note that you can only stop and start instances that are Amazon Elastic Block Store (EBS)-backed.

An EC2 instance can be in one of the following states:

- Pending: When you launch an instance, it enters the pending state and the instance moves to a new host computer. The instance type specified at launch determines the hardware of the host computer for your instance.
- Running: AWS uses the AMI specified at launch to boot the instance. Once the instance is ready for you, it enters the running state. You can connect to your running instance and use it as you would a computer sitting in front of you. As soon as your instance is in the running state, you're billed for each hour or partial hour that you keep the instance running. You are billed for all running instances, even if they are idle and not being connected to.
- Rebooting: You can reboot your instance through the Amazon EC2 console, Amazon EC2 CLI, and the Amazon EC2 API. It is recommended that you reboot your EC2 instance rather than running the operating system reboot from the instance. When an instance is rebooted, it remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. Rebooting an instance doesn't start a new instance billing hour.

- Shutting Down: When you've decided you no longer need an instance, you can terminate it. The instance will enter the shutting-down state. You will stop incurring charges as soon as the instance enters shutting-down or terminated states.
- Terminated: A terminated instance remains visible in the console for a while before it is deleted. You cannot connect to or recover a terminated instance.
- Stopping: Amazon EBS-backed instances can be stopped. When you stop an instance, it enters the stopping state.
- Stopped: Amazon EBS-backed instances in the stopped state are no longer eligible for hourly usage or data transfer fees. AWS does charge for the storage of EBS volumes on stopped instances. You can modify certain attributes of stopped instances, including the instance type. Starting a stopped instance puts it back into the pending state, which moves the instance to a new host machine. When you stop and start an instance, you lose any data on the instance store volumes on the previous host computer.



AWS and Oracle have worked together to offer customers convenient options for deploying enterprise applications in the cloud. Customers can not only build enterprise-grade solutions hosted by Amazon Web Services using database and middleware software by Oracle, but they can also launch entire enterprise software stacks from Oracle on Amazon EC2.

New and existing SAP customers can deploy their SAP solutions on SAP-certified Amazon EC2 instances in production environments knowing that SAP and AWS have tested the performance of the underlying AWS resources, verified their performance, and certified them against the same standards that apply to servers and virtual platforms.

AWS also provides infrastructure services that allow customers to easily run Microsoft Windows Server applications in the cloud, without the cost and complexity of having to purchase or manage servers or data centers. AMIs are available, allowing customers to start running fully supported Windows Server virtual machine instances in minutes.

Customers may also rely on the global infrastructure of AWS to power everything from custom .NET applications to enterprise deployments of Microsoft Exchange Server, SQL Server, or SharePoint Server.

Software launched from AWS customers automatically deploys onto Amazon Elastic Compute Cloud (EC2), which is the AWS compute service. AWS customers use 143 million hours a month of Amazon EC2 for AWS Marketplace software products.

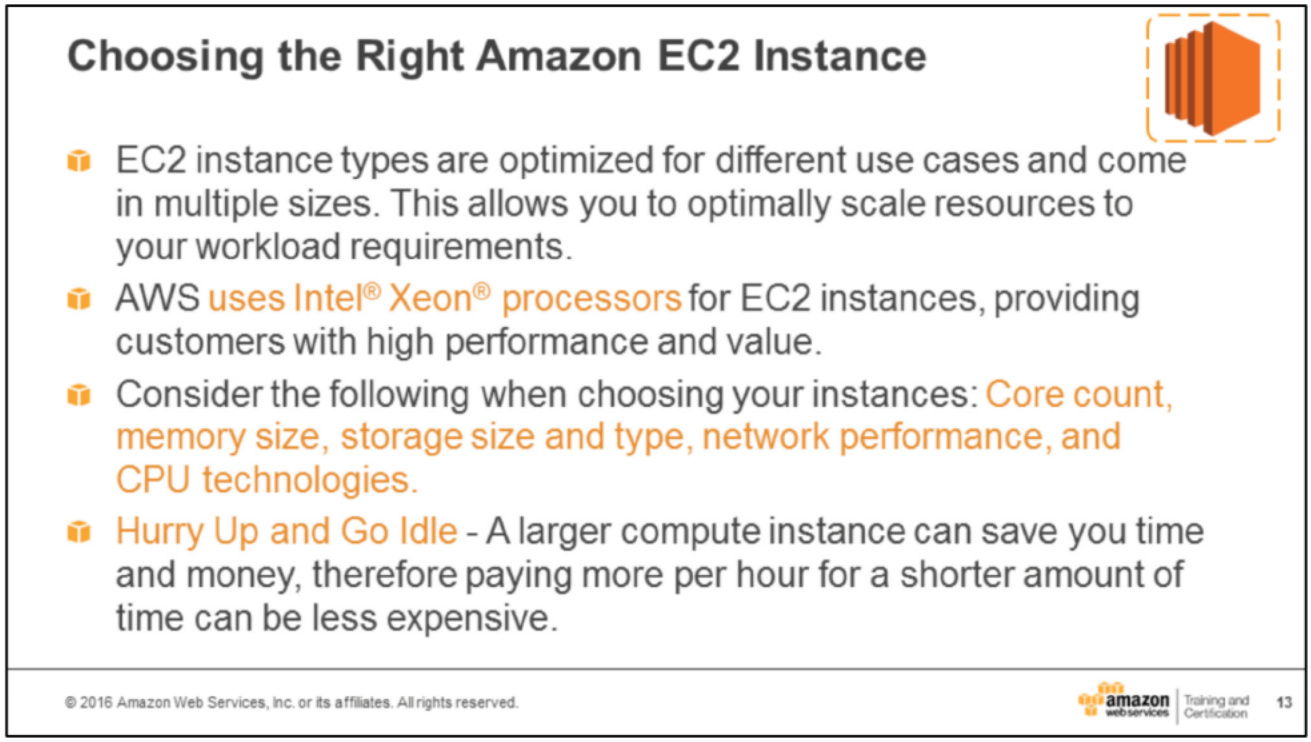
The benefits of AWS Marketplace include:

- Easy product discovery
- Streamlined buying experience
- Simplified billing- software charges on AWS bill
- Expedited deployment cycles
- Optimized software capacity
- Matched spend to actual usage-Opex
- Trust vetted and scanned products

AWS Test Drive provides a private IT sandbox environment containing preconfigured server based solutions. In under an hour, and using a step-by-step lab manual and video, launch, login and learn about these popular 3rd party IT solutions, powered by AWS and AWS CloudFormation.

For more information, see:

https://aws.amazon.com/marketplace



Choosing the right EC2 instance type matters. Selecting an appropriate instance type for your workload can save time and money. AWS has a wide variety of EC2 compute instance types to choose from. Each instance type or family (T2, M3, C4, C3, G2, R3, and so on) is optimized for different workloads or use cases. Within an EC2 family, you can choose from different sizes: for example, micro, small, medium, large, xlarge, and 2xlarge. AWS uses Intel® Xeon® processors for the EC2 instances to provide customers with high performance and value for their computing needs.

When you choose your instance type you should consider the several different attributes of each family, such as number of cores, amount of memory, amount and type of storage, network performance, and processor technologies.

Another important consideration is total cost of ownership (TCO). A lowest-price-per-hour instance is not necessarily a money saver; a larger compute instance can sometimes save both money and time. It is important to evaluate all the options to see what is best for your workload.

General Purpose Instances

T2 instances are a low-cost, burstable performance instance type that provide a baseline level of CPU performance with the ability to burst above the baseline. They offer a balance of compute, memory, and network resources for workloads that occasionally need to burst, such as web servers, build servers, and development environments.

M3 and M4 instances provide a balance of compute, memory, and network resources. These instances are ideal for applications that require high CPU and memory performance, such as encoding applications, high traffic content management systems, and memcached applications.

Compute-Optimized Instances

C3 and C4 instances are optimized for compute-intensive workloads. These instances have proportionally more CPU than memory (RAM). They are well suited to applications such as high performance web servers, batch processing, and high-performance scientific and engineering applications.

Memory-Optimized Instances

R3 instances are optimized for memory-intensive workloads. These instances offer large memory sizes for high throughput applications such as high performance databases, distributed memory caches, in-memory analytics, and large enterprise deployments of software such as SAP.

GPU Instances

G2 instances are optimized for graphics and graphic processing unit (GPU) compute applications, such as machine learning, video encoding, and interactive streaming applications.

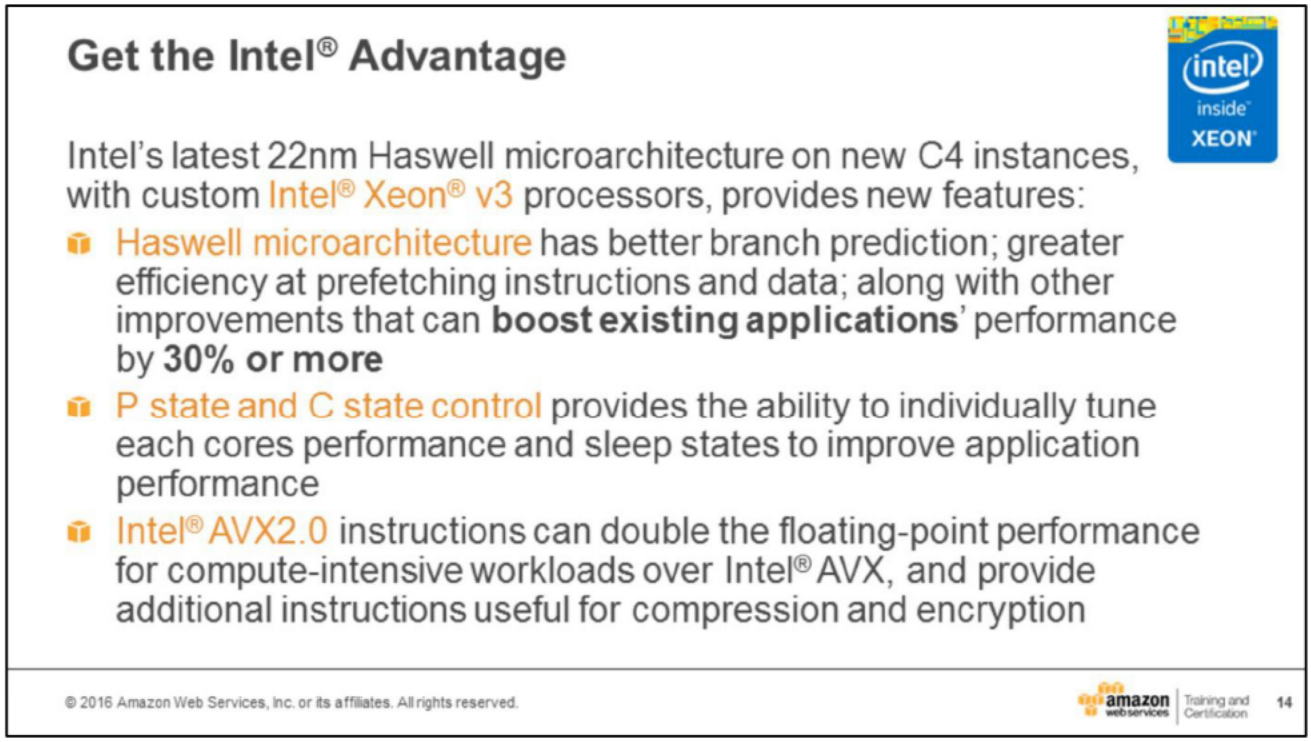
Storage-Optimized Instances

I2 instances are optimized for storage and high random I/O performance, such as NoSQL databases, scale-out transactional databases, data warehousing, Hadoop, and cluster file systems.

D2 instances are optimized for storage and delivering high disk throughput. D2 instances are suitable for Massively Parallel Processing (MPP) data warehousing, MapReduce and Hadoop distributed computing, distributed file systems, and data processing applications.

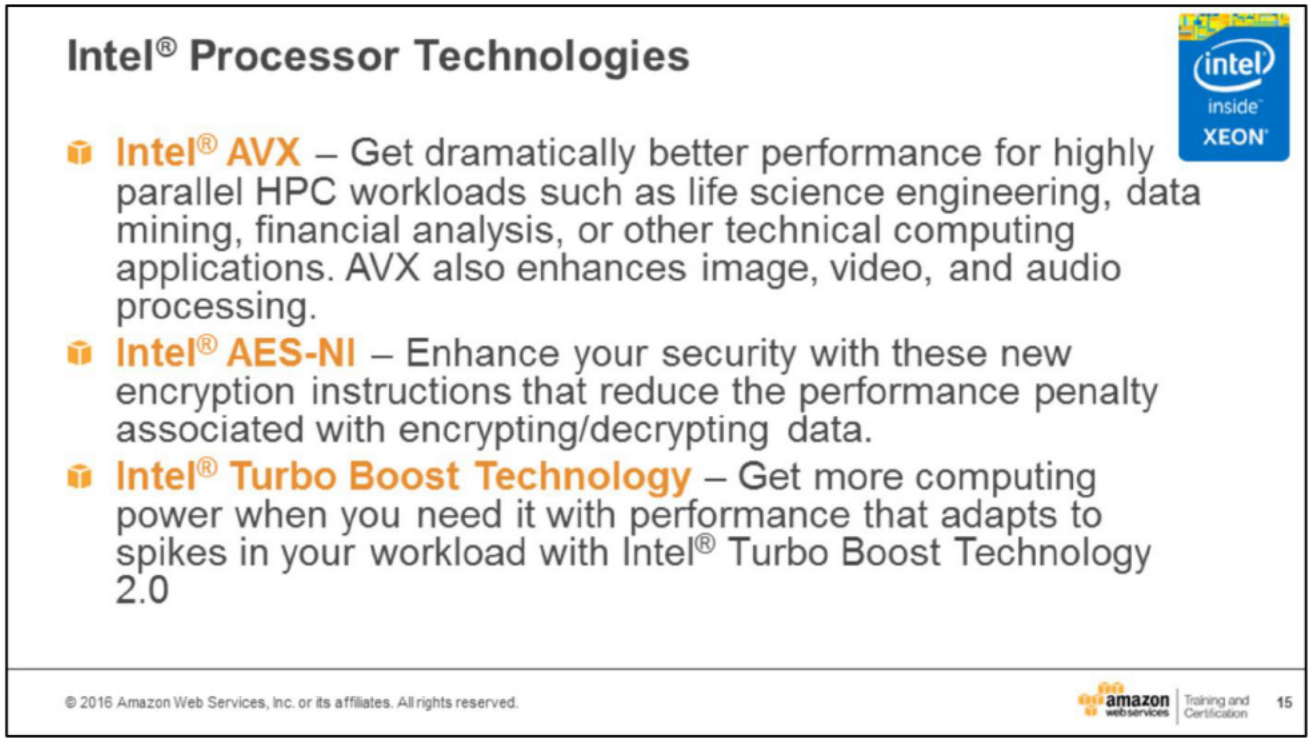
For more information, see:

https://aws.amazon.com/ec2/instance-types/



AWS recently launched C4 compute-optimized instances which utilize Intel's latest 22nm Haswell microarchitecture. C4 instances use custom Intel® Xeon® v3 processors designed and built especially for AWS.

Through its relationship with Intel®, AWS provides its customers with the latest and greatest Intel® Xeon® processors that help in delivering the highest level of processor performance in Amazon EC2.



Intel® Xeon® processors have several other important technology features that can be leveraged by EC2 Instances.

- Intel® AVX is perfect for highly parallel HPC workloads such as life sciences or financial analysis.
- Intel® AES-NI accelerates encryption/decryption of data and therefore reduces the performance penalty that usually comes with encryption.
- Intel® Turbo Boost Technology automatically gives you more computing power when your workloads are not fully utilizing all CPU cores. Think of it as automatic overclocking when you have thermal headroom.

EC2 Instances with Intel® Technologies

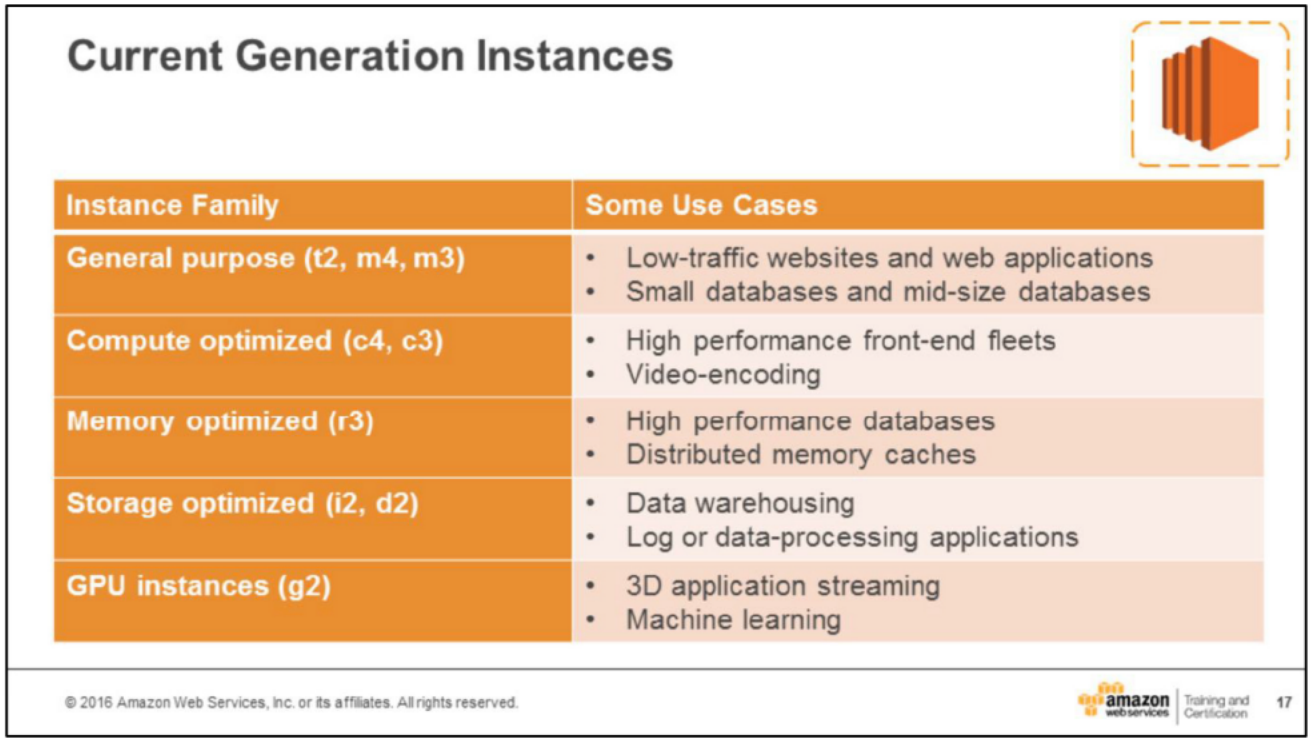
	Burstable	Balanced	Compute	Memory	GPU	I/O	Storage
AWS Instance Type	T2	M4	C4	R3	G2	I2	D2
Intel® processor	Intel® Xeon® family	Intel® Xeon® E5-2676 v3	Intel® Xeon® E5-2666 v3	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2670	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2676 v3
Intel® process technology	●	22nm Haswell	22nm Haswell	22nm Ivy Bridge	32nm Sandy Bridge	22nm Ivy Bridge	22nm Haswell
Intel® AVX	●	●	●	●	●	●	●
Intel® AVX2		●	●				●
Intel® Turbo Boost	●	●	●	●	●	●	●
Storage	EBS only	EBS only	EBS only	SSD	SSD	SSD	HDD

amazon web services | Training and Certification 16

The matrix on the slide highlights the individual Intel® technologies that were discussed previously and the EC2 instance family that can leverage each of these technologies.

For more information, see:

http://aws.amazon.com/ec2/instance-types/

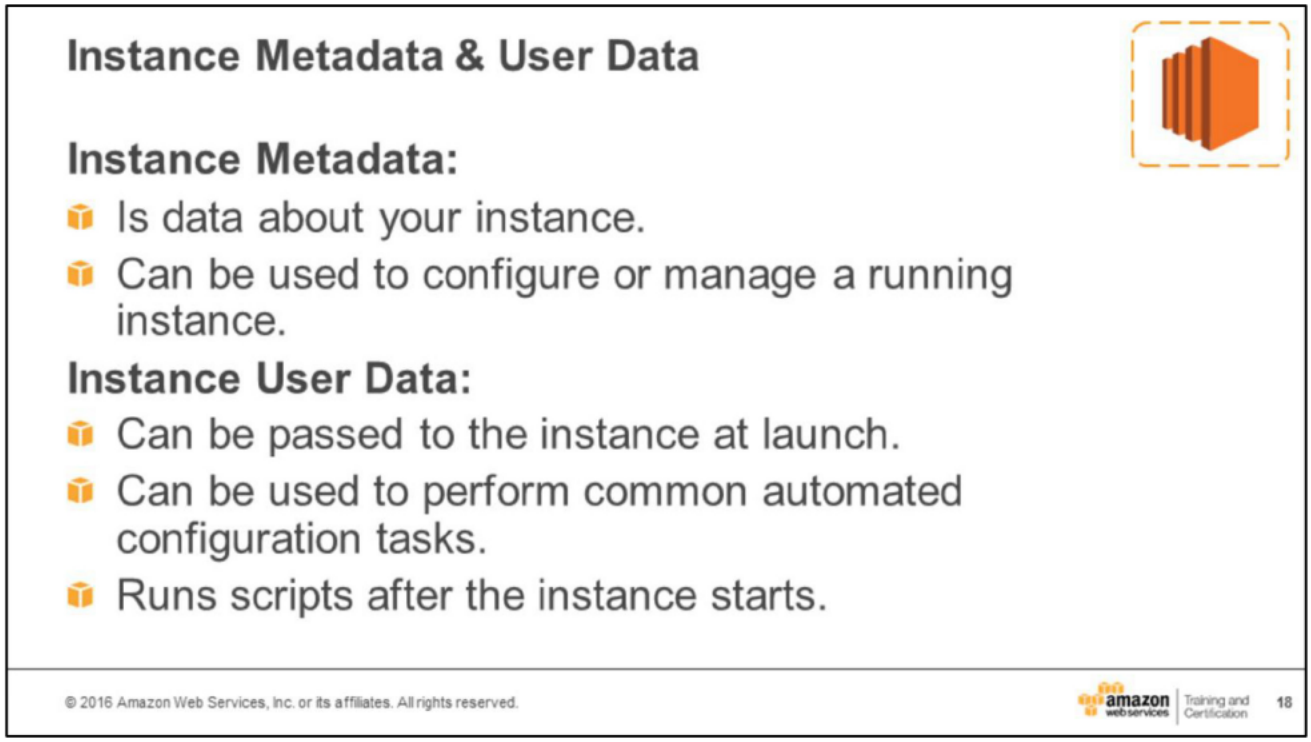


Each vCPU is a hyperthread of an Intel Xeon core for M4, M3, C4, C3, R3, HS1, G2, I2, and D2.

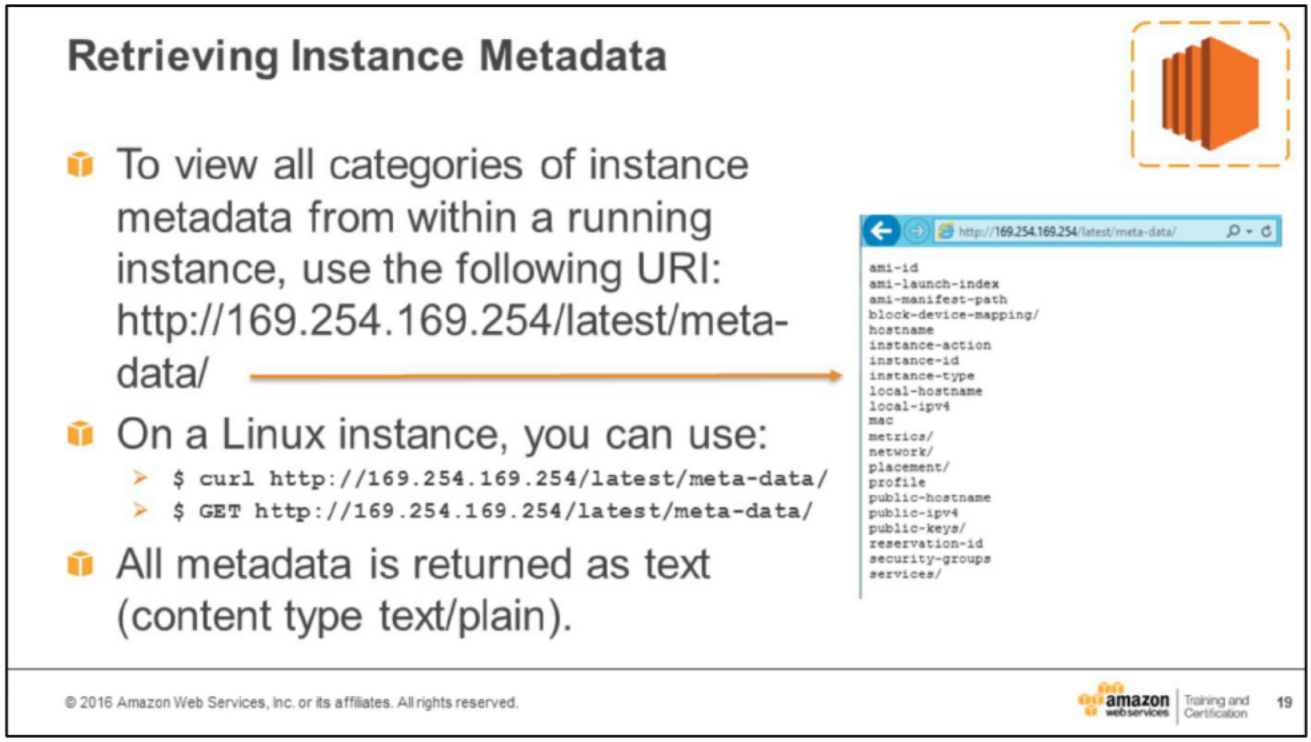
Amazon EC2 lets you choose from a number of different instance types to meet your computing needs. Each instance provides a predictable amount of dedicated compute capacity and is charged per instance-hour consumed. First-generation (M1) general purpose instances provide a balanced set of resources and a low-cost platform that is well suited for a wide variety of applications. Second-generation (M3) general purpose instances provide a balanced set of resources and a higher level of processing performance compared to first-generation general purpose instances. Instances in this family are ideal for applications that require higher absolute CPU and memory performance. Applications that can benefit from the performance of second-generation general purpose instances include encoding applications, high traffic content management systems, and Memcached applications. High-memory instances offer large memory sizes for high throughput applications, including database and memory caching applications. High-CPU instances have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications. There are also various high-storage and cluster-computer instance types available.

For more information, see:

- http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/instance-types.html
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html
- http://aws.amazon.com/ec2/instance-types/



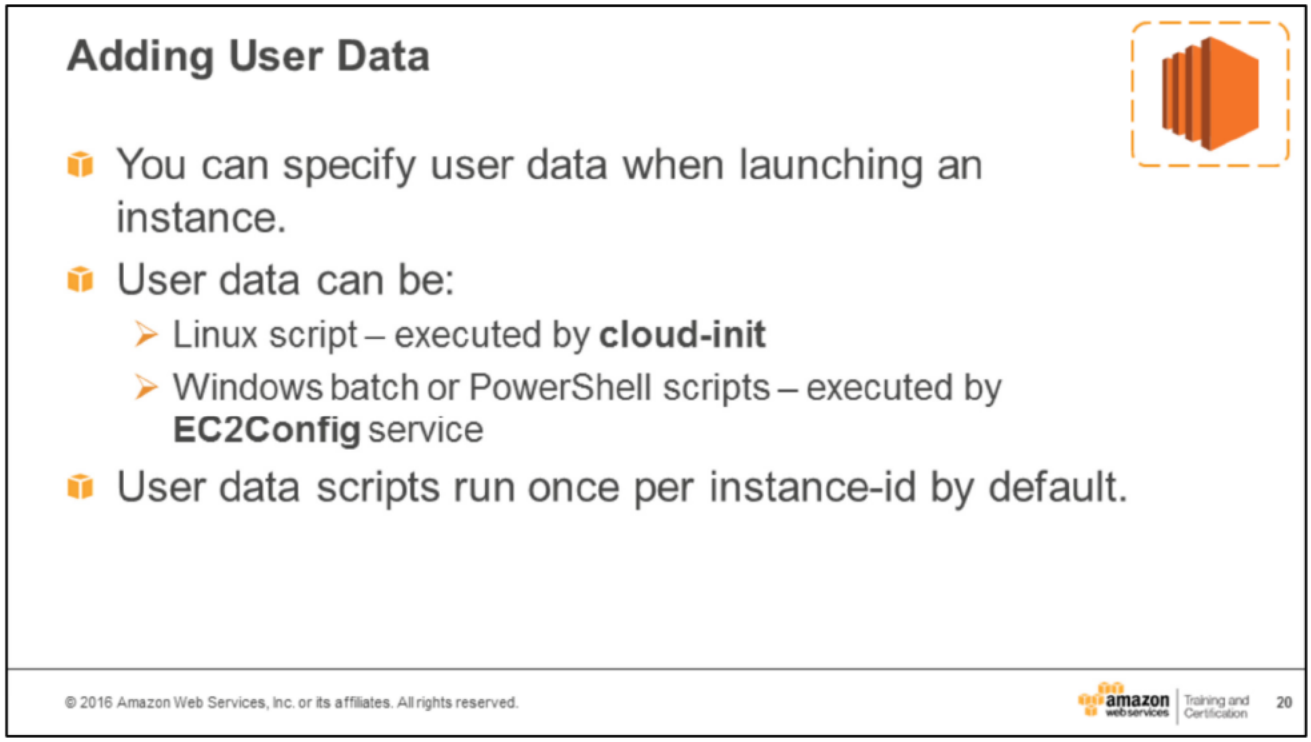
Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.



Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

For more information, see:

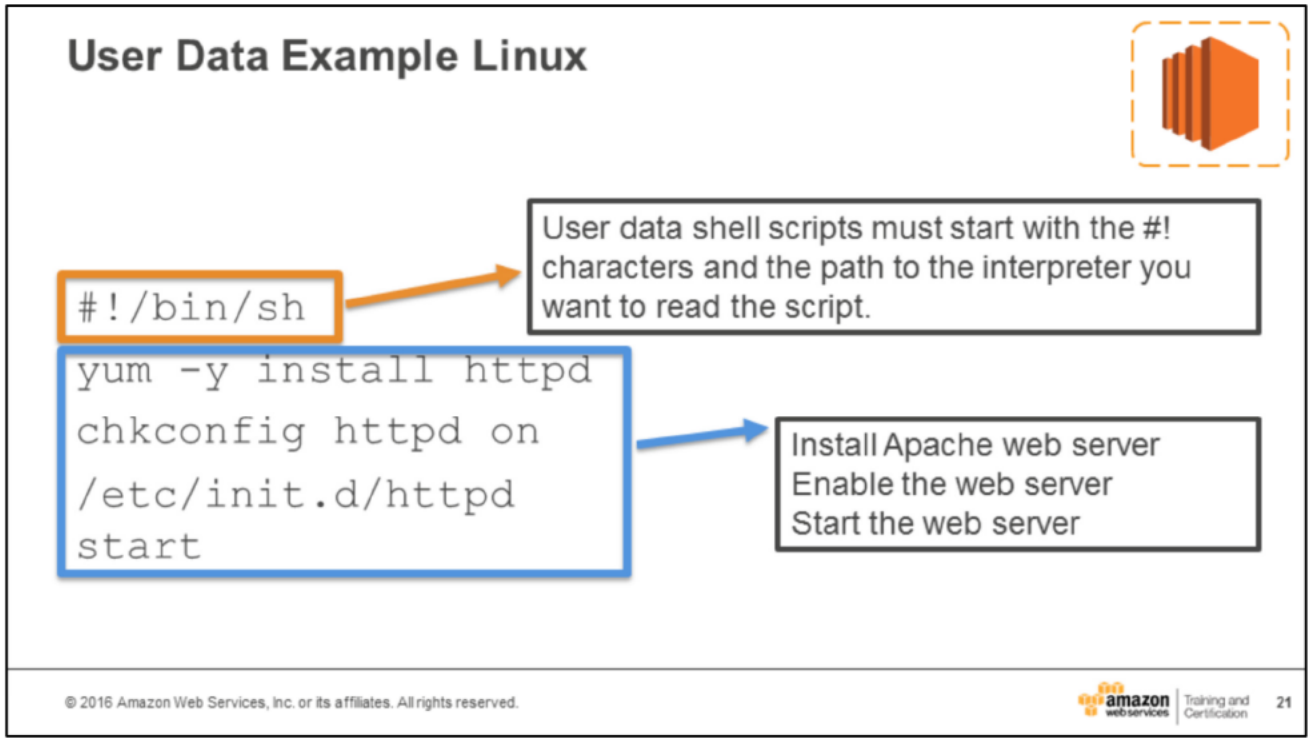
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html



You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the As file option and browse for the file to attach. The cloud-init package is an open source application built by Canonical that is used to bootstrap Linux images in a cloud computing environment, like Amazon EC2.

User data information:

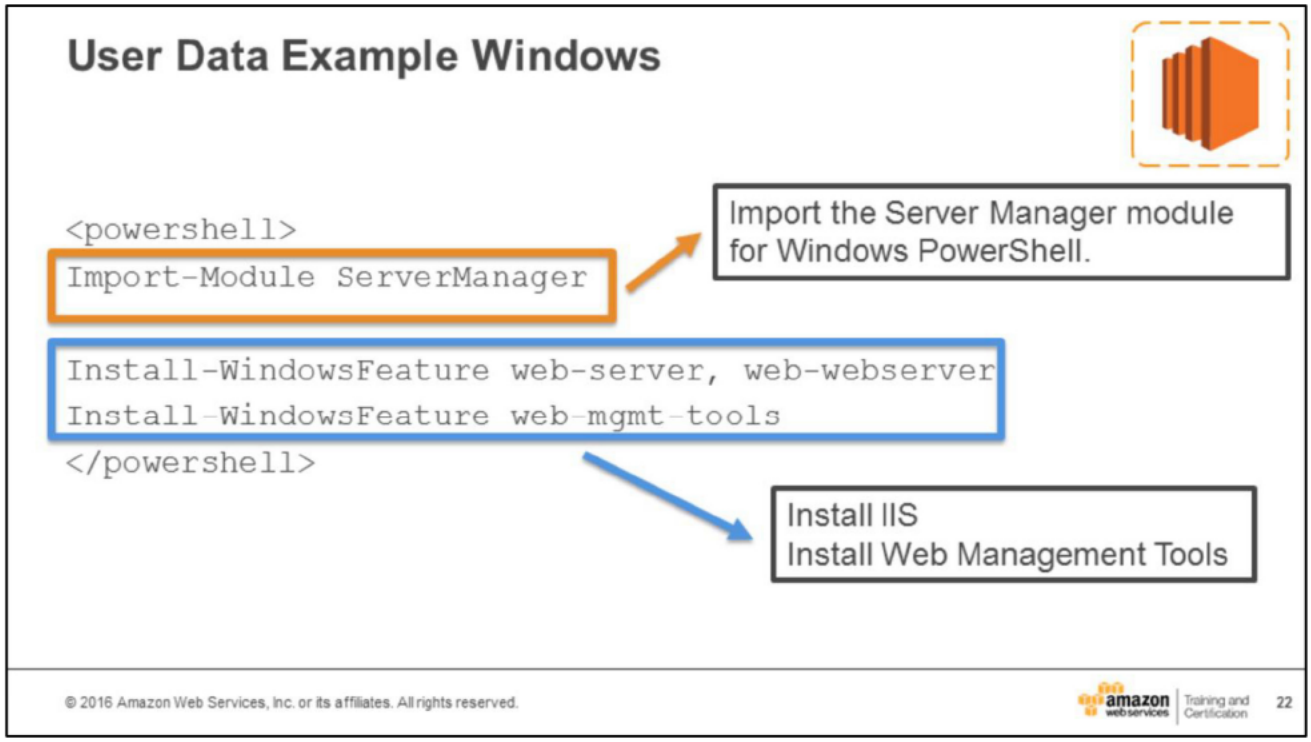
- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded before being submitted to the API. The Amazon EC2 command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance.
- User data is executed only at launch. If you stop an instance, modify the user data, and start the instance, the new user data is not executed automatically by default.



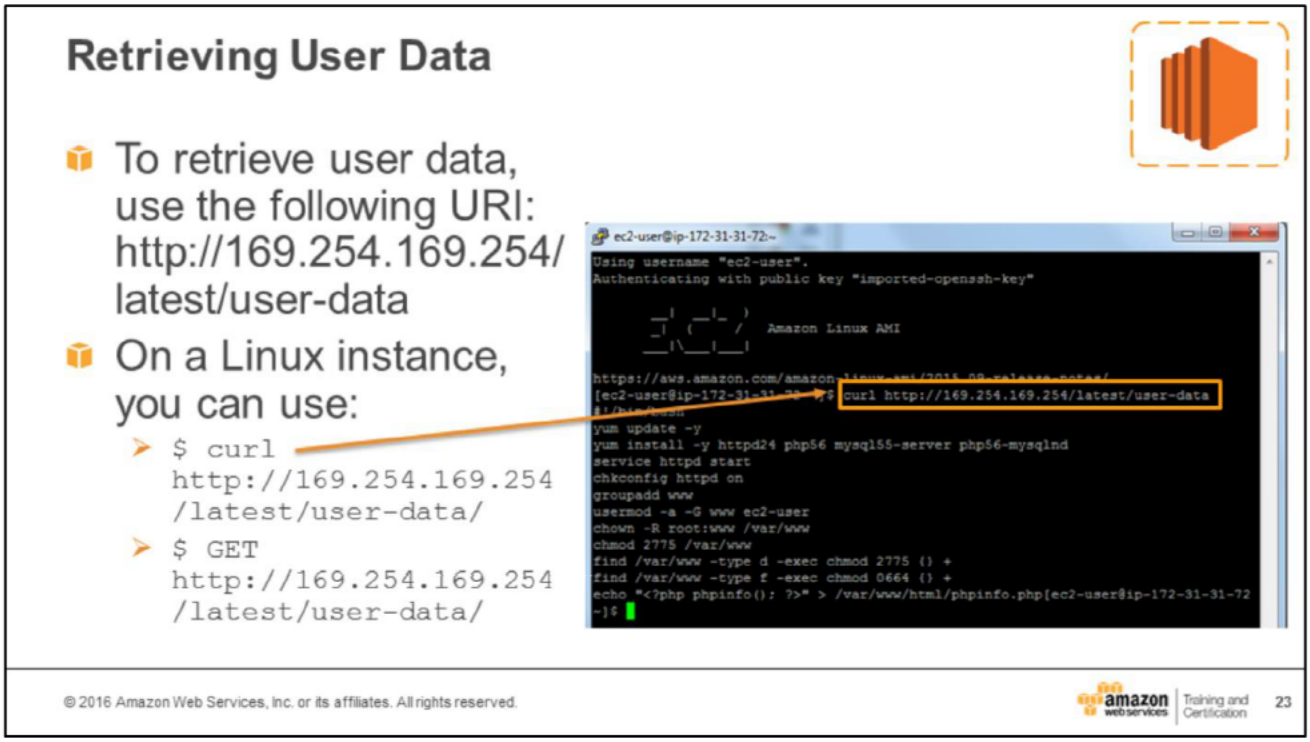
The slide shows an example of user data on Linux. You can also provide user data to an instance on Linux by using the #cloud-config directive – a format defined by cloud-init.

For more information, see:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html



You can send user data to a Windows instance with a PowerShell script (shown in slide) or with a set of Windows batch commands.

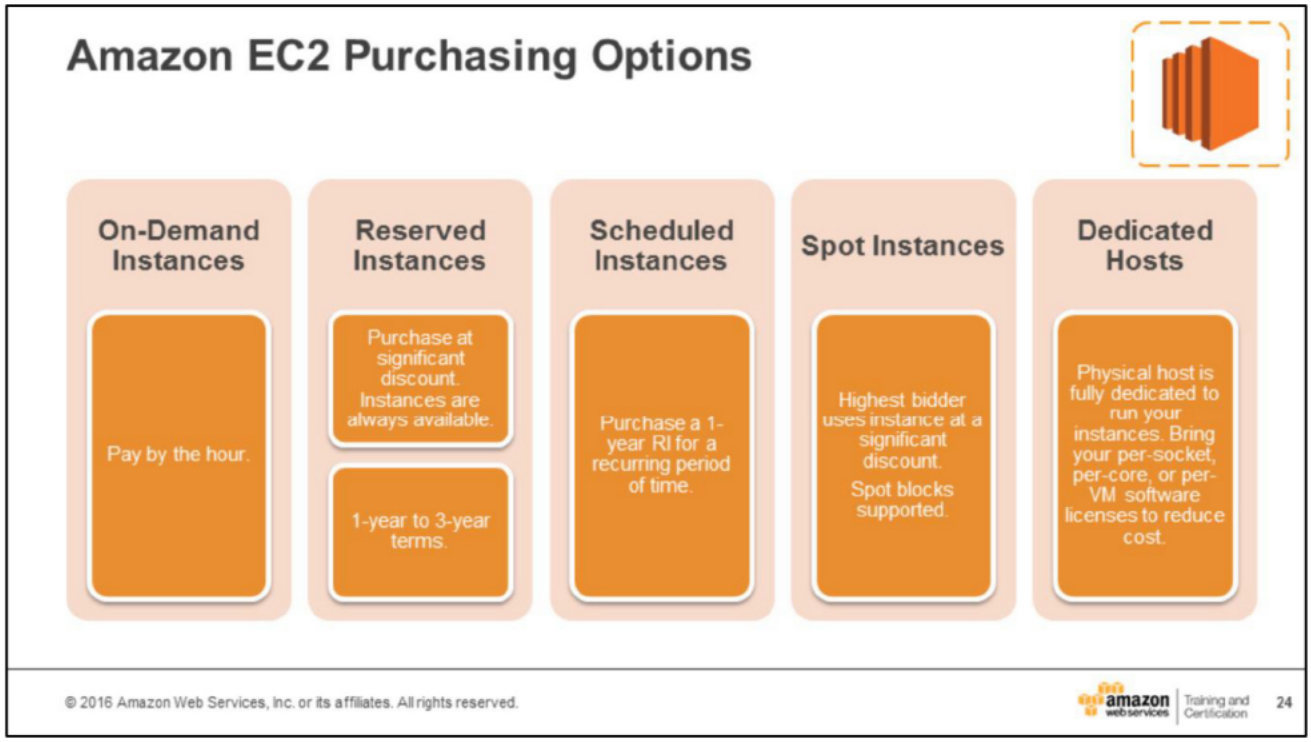


To retrieve instance user data, use the following URI:

http://169.254.169.254/latest/user-data

For more information, see:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html#instancedata-user-data-retrieval



On-Demand instances are free-tier eligible. It has the lowest up-front cost and the most flexibility. You pay for an hour at a time with no up-front commitments or long-term contracts. This is great for applications with short-term, spiky, or unpredictable workloads.

Amazon EC2 Reserved instance pricing allows you to reserve computing capacity for 1-year to 3-year terms at a significantly discounted hourly rate. Reserved instances are a billing discount and capacity reservation that is applied to instances to lower hourly running costs. A Reserved instance is not a physical instance. The discounted usage price is fixed for as long as you own the Reserved instance, allowing you to predict compute costs over the term of the reservation. If you are expecting consistent, heavy, use, Reserved instances can provide substantial savings over owning your own hardware or running only On-Demand instances.

Scheduled Reserved instances enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified duration, for a 1-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time the instances are scheduled, even if you do not use them. Scheduled instances are a good choice for workloads that do not run continuously, but do run on a regular schedule and take a finite time to complete.

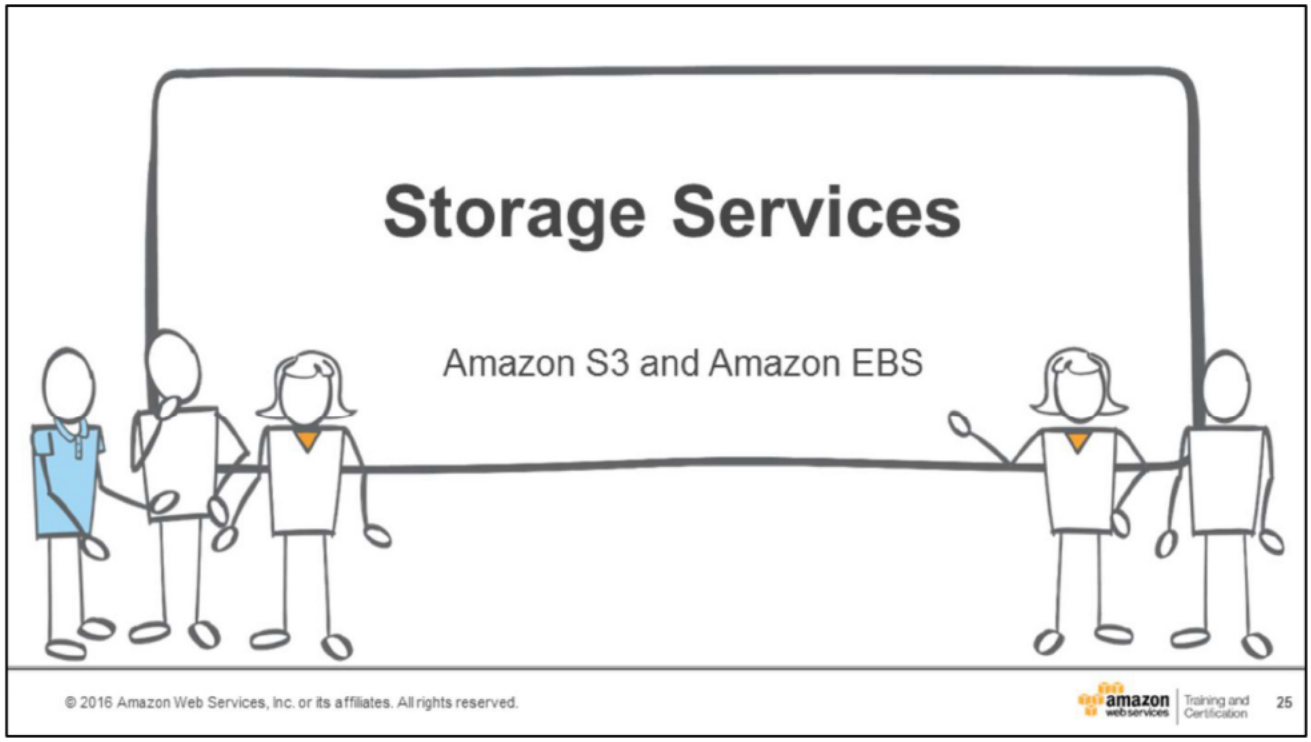
Spot instances enable you to bid on unused EC2 instances, which can lower your Amazon EC2 costs significantly. The hourly price for a Spot instance (of each instance type in each Availability Zone) is set by Amazon EC2, and fluctuates depending on the supply of and demand for Spot instances. Your Spot instance runs whenever your bid exceeds the current market price.

Spot instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. Amazon EC2 does not terminate Spot instances with a specified duration (also known as Spot blocks) when the Spot price changes. This makes them ideal for jobs that take a finite time to complete, such as batch processing, encoding and rendering, modeling and analysis, and continuous integration.

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Microsoft Windows Server, Microsoft SQL Server, SUSE, Linux Enterprise Server, and so on. Dedicated Hosts and Dedicated instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use. There are no performance, security, or physical differences between Dedicated instances and instances on Dedicated Hosts. However, Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server.

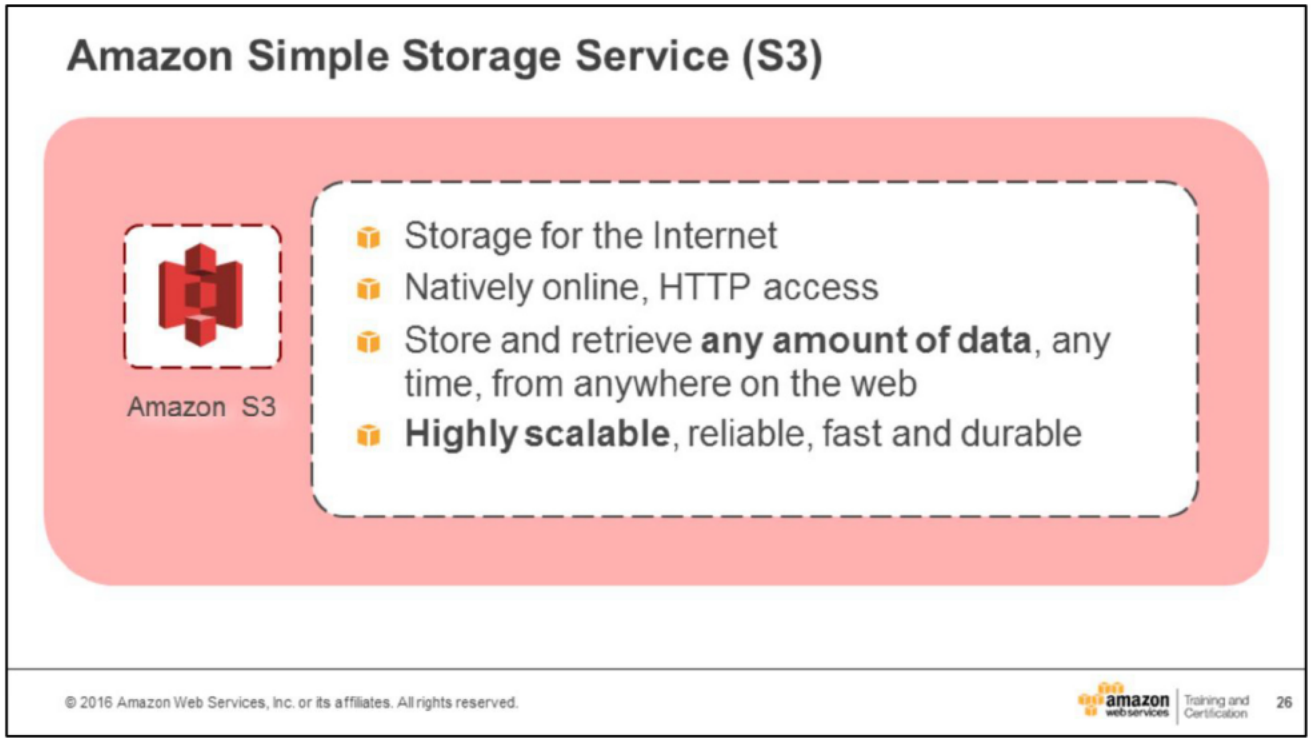
For more information, see:

- http://aws.amazon.com/ec2/pricing/
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html



Understand storage options including:

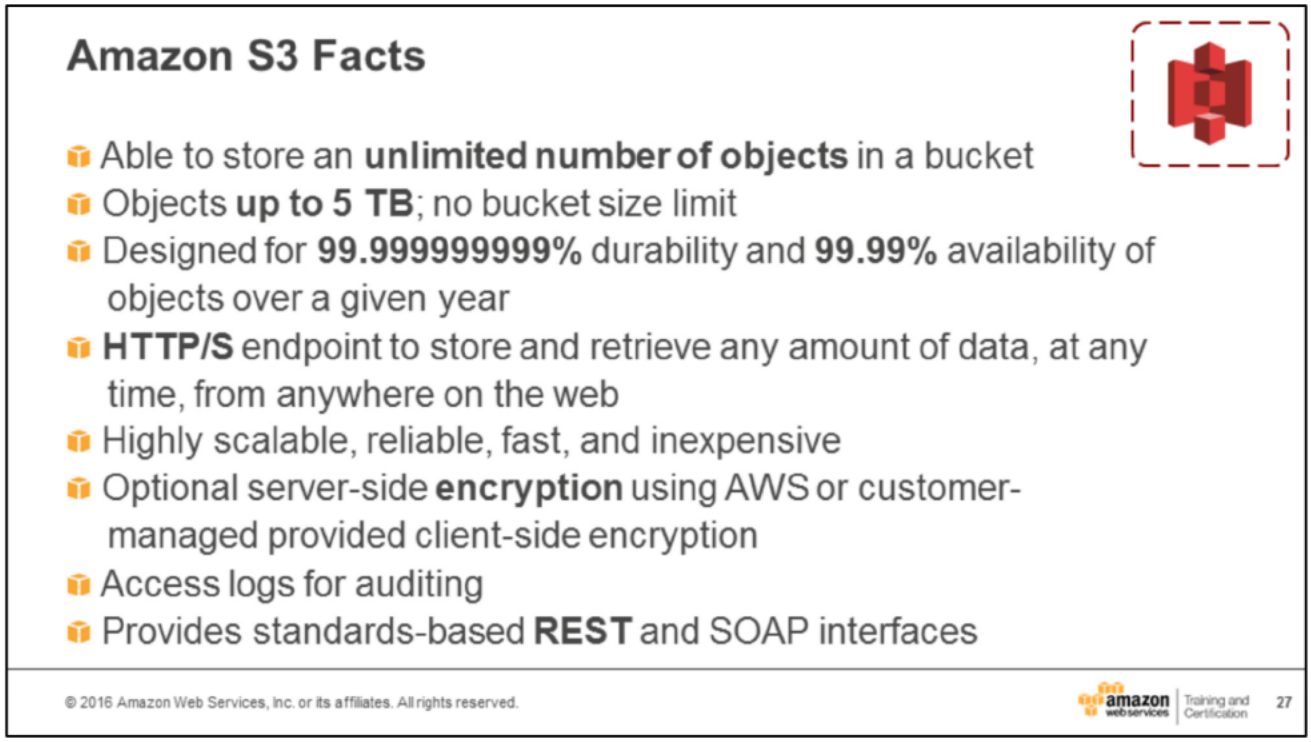
- Amazon S3 (Requests, Buckets, Objects, Access, Protecting Data, Notifications, Replication, Request Routing, Optimization, Lifecycle Management with Glacier)
- Amazon EBS (Volumes, Snapshots, Optimization, Encryption, Performance)



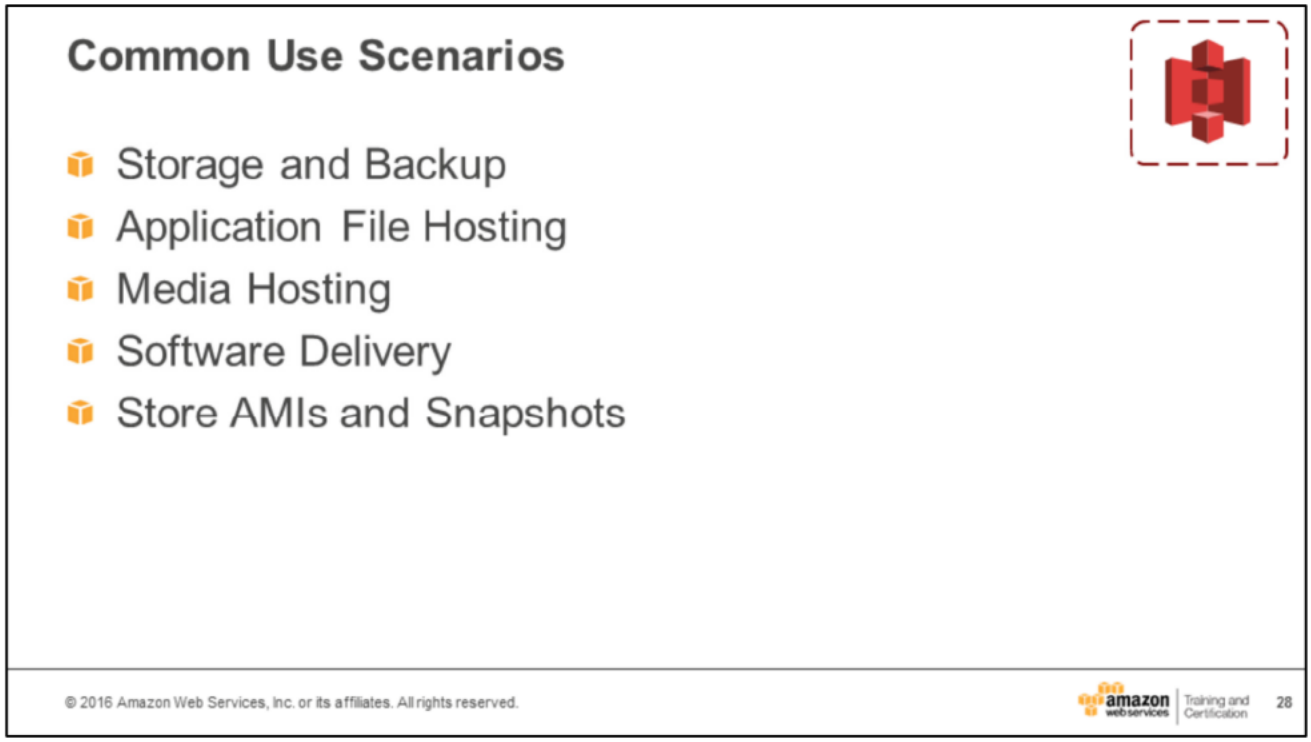
Amazon S3 is designed to make web-scale computing easier for developers. Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of websites.

For more information, see:

http://aws.amazon.com/s3/



Here are some facts about Amazon S3. There is a 100-bucket limit per account. You can store unlimited number of objects in a bucket. The size of an object can be up to 5 TB and there is no limit to the size of a bucket. Amazon S3 is designed for 99.999999999% durability and 99.99% availability of objects over a given year. You can use HTTP or HTTPs endpoints to store and retrieve any amount of data, at any time, from anywhere on the web. Most importantly, Amazon S3 is highly scalable, reliable, fast, and inexpensive.



Advanced use scenarios:

Using Amazon DevPay with Amazon S3: Amazon DevPay enables you to charge customers for using your Amazon S3 product through Amazon's authentication and billing infrastructure. You can charge any amount for your product including usage charges (storage, transactions, and bandwidth), monthly fixed charges, and a one-time charge.

Publishing Content Using Amazon S3 and BitTorrent: You can direct your clients to your BitTorrent accessible objects by giving them the .torrent file directly or by publishing a link to the BitTorrent URL of your object.

Hosting a Static Website on Amazon S3: You can host a static website on Amazon S3 by configuring a bucket for website hosting and then uploading your website content to the bucket.

For more information, see:

- http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingDevPay.html
- http://docs.aws.amazon.com/AmazonS3/latest/dev/S3TorrentPublish.html
- http://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html

Amazon S3 Pricing

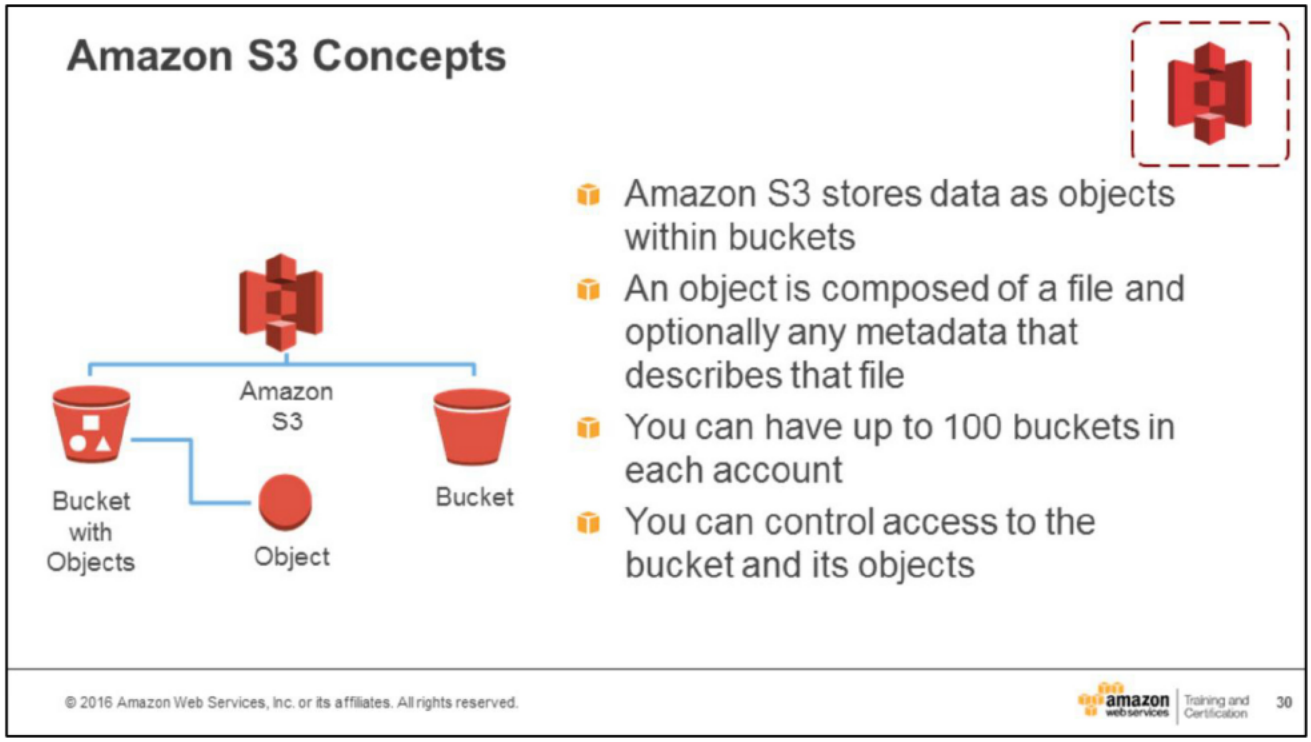
- Pay only for what you use
- No minimum fee
- Prices based on location of your Amazon S3 bucket
- Estimate monthly bill using the **AWS Simple Monthly Calculator**
- Pricing is available as:
  - Storage Pricing
  - Request Pricing
  - Data Transfer Pricing: data transferred out of Amazon S3

Amazon S3 pricing is based on capacity and bandwidth actually used. Since Amazon S3 is an Internet-scale service that runs natively across an entire region; it can handle significant request throughput and bandwidth output. All bandwidth into Amazon S3 is free, but AWS charges a rate on bandwidth out. Most importantly, since Amazon S3 can handle any amount of data, it is important to note that you only pay for the amount of space you use. Prices are based on a prorated GB per month.

There is also a pricing calculator online as a reference. Note that pricing listed is in the US East (N. Virginia) Region at the time this training was developed.

For more information, see:

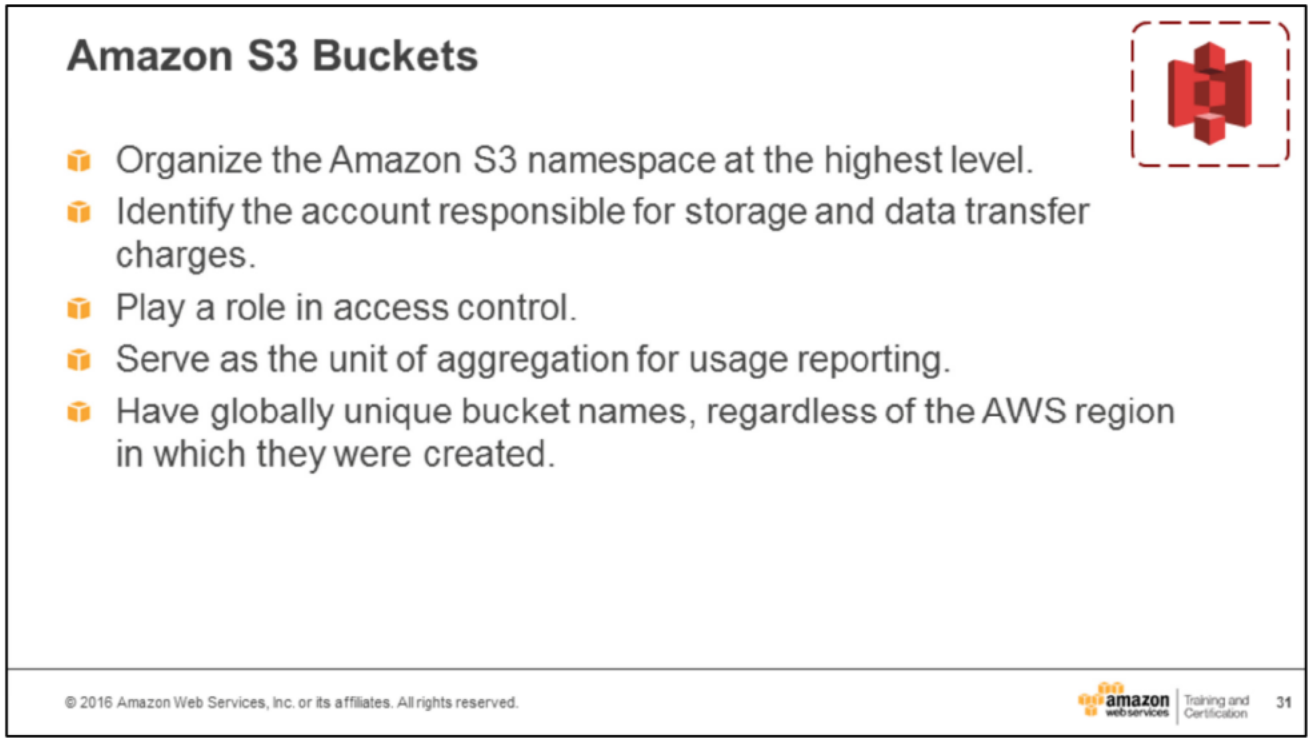
- Online Pricing Calculator - http://calculator.s3.amazonaws.com/calc5.html
- https://aws.amazon.com/s3/pricing/



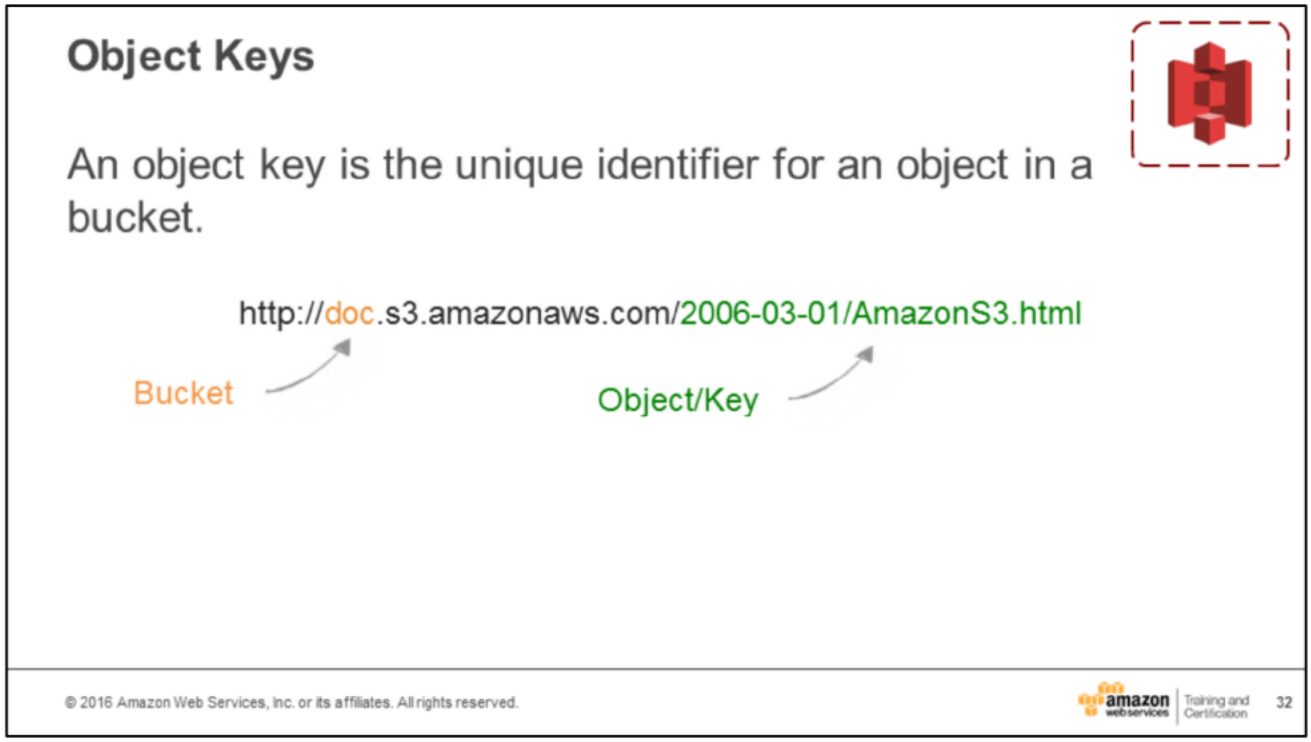
To get the most out of Amazon S3, you need to understand a few simple concepts. First, Amazon S3 stores data as objects within buckets.

An object is composed of a file, and any metadata that describes that file. To store an object in Amazon S3, you upload the file you want to store into a bucket. When you upload a file, you can set permission on the object as well as any metadata.

Buckets are logical containers for objects. You can have one or more buckets in your account. For each bucket, you can control access, in other words, who can create, delete and list objects in the bucket. You can also view access logs for the bucket, and its objects, and choose the geographical region where Amazon S3 will store the bucket and its contents.

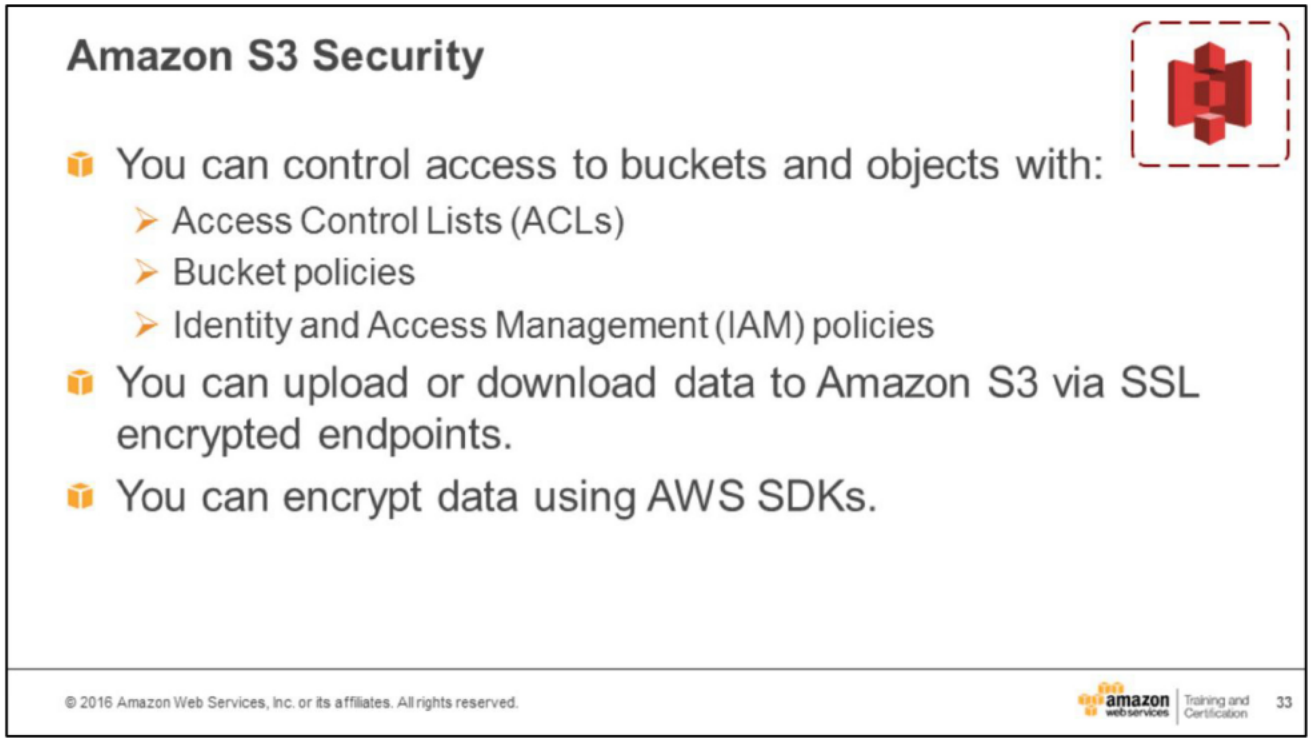


A bucket is a logical container for objects stored in Amazon S3. Every object is contained in a bucket. Buckets serve several purposes: They organize the Amazon S3 namespace at the highest level, they identify the account responsible for storage and data transfer charges, they play a role in access control, and they serve as the unit of aggregation for usage reporting. Amazon S3 bucket names are globally unique, regardless of the AWS Region in which you create the bucket. You specify the name at the time you create the bucket.



Because the combination of a bucket, key, and version ID uniquely identify each object, Amazon S3 can be thought of as a basic data map between "bucket + key + version" and the object itself. Every object in Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, key, and optionally, a version.

For example, in the URL http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.html, "doc" is the name of the bucket and "2006-03-01/AmazonS3.html" is the key.



Data Access:

- IAM policies: With IAM policies, you can only grant users within your own AWS account permission to access your Amazon S3 resources.
- ACLs: With ACLs, you can only grant other AWS accounts (not specific users) access to your Amazon S3 resources.
- Bucket Policies: Bucket policies in Amazon S3 can be used to add or deny permissions across some or all of the objects within a single bucket. Policies can be attached to users, groups, or Amazon S3 buckets, enabling centralized management of permissions. With bucket policies, you can grant users within your AWS account or another AWS account access to your Amazon S3 resources.

Data Transfer: For maximum security, you can securely upload/download data to Amazon S3 via the SSL encrypted endpoints. The encrypted endpoints are accessible both from the Internet and from within Amazon EC2 so that data is transferred securely both within AWS and to and from sources outside of AWS.

Data Storage: Amazon S3 provides multiple options for protecting data at rest. Customers who prefer to manage their own encryption keys can use a client encryption library like the Amazon S3 Encryption Client to encrypt data before uploading to Amazon S3.

Alternatively, you can use Amazon S3 Server Side Encryption (SSE) if you prefer to have Amazon S3 manage encryption keys for you. With Amazon S3 SSE, you can encrypt data on upload simply by adding an additional request header when writing the object. Decryption happens automatically when data is retrieved.

Amazon S3 SSE uses one of the strongest block ciphers available: 256-bit Advanced Encryption Standard (AES-256). With Amazon S3 SSE, every protected object is encrypted with a unique encryption key. This object key itself is then encrypted with a regularly rotated master key. Amazon S3 SSE provides additional security by storing the encrypted data and encryption keys in different hosts. Amazon S3 SSE also makes it possible for you to enforce encryption requirements. For example, you can create and apply bucket policies that require that only encrypted data can be uploaded to your buckets.

Instead of using Amazon S3 SSE, you also have the option of encrypting your data before sending it to Amazon S3. You can build your own library that encrypts your object data on the client side before uploading it to Amazon S3. Optionally, you can use an AWS SDK to automatically encrypt your data before uploading it to Amazon S3.

For more information, see:

http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

Amazon S3 Versioning

- Protects from accidental overwrites and deletes with no performance penalty.
- Generates a new version with every upload.
- Allows easily retrieval of deleted objects or roll back to previous versions.
- Three states of an Amazon S3 bucket
  - Un-versioned (default)
  - Versioning-enabled
  - Versioning-suspended

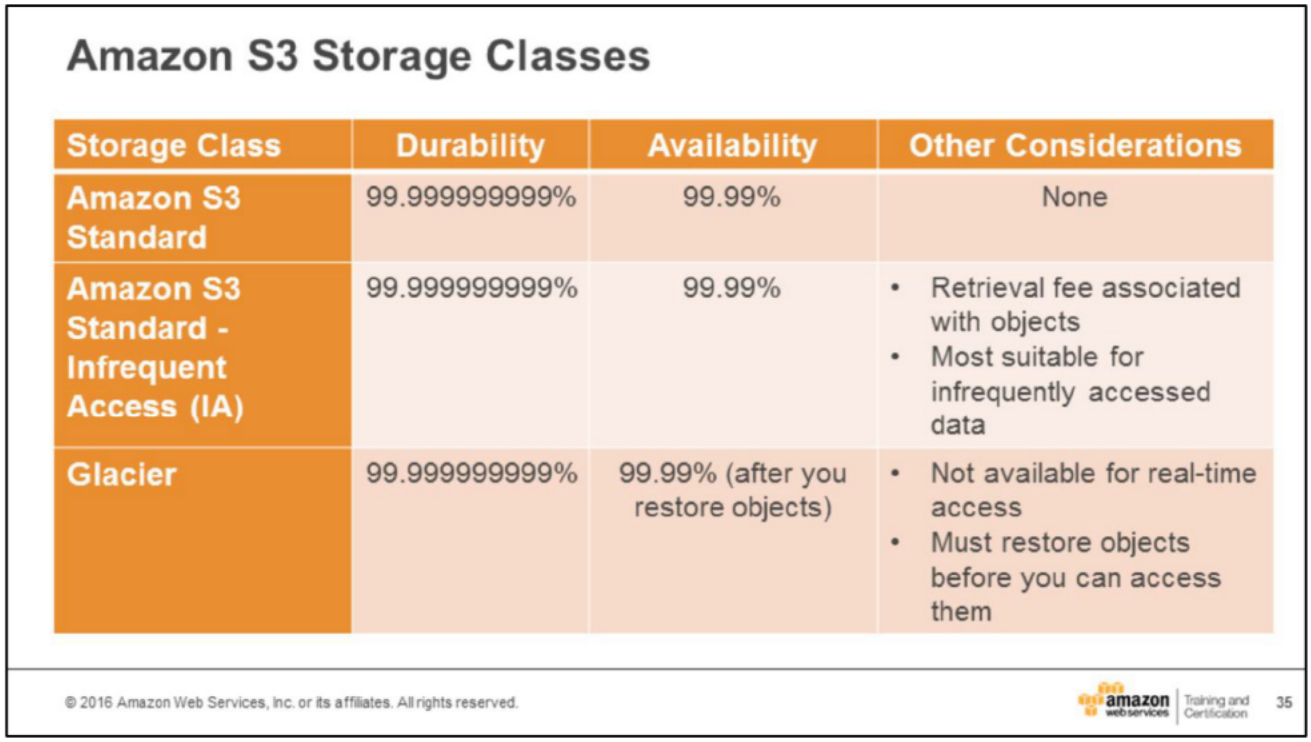
Versioning Enabled

amazon web services | Training and Certification 34

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

In one bucket, for example, you can have two objects with the same key, but different version IDs, such as photo.gif (version 111111) and photo.gif (version 121212).

Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.



Each object in S3 has a storage class associated with it.

S3 Standard is ideal for performance-sensitive use cases and frequently used data. Standard is the default storage class in S3.

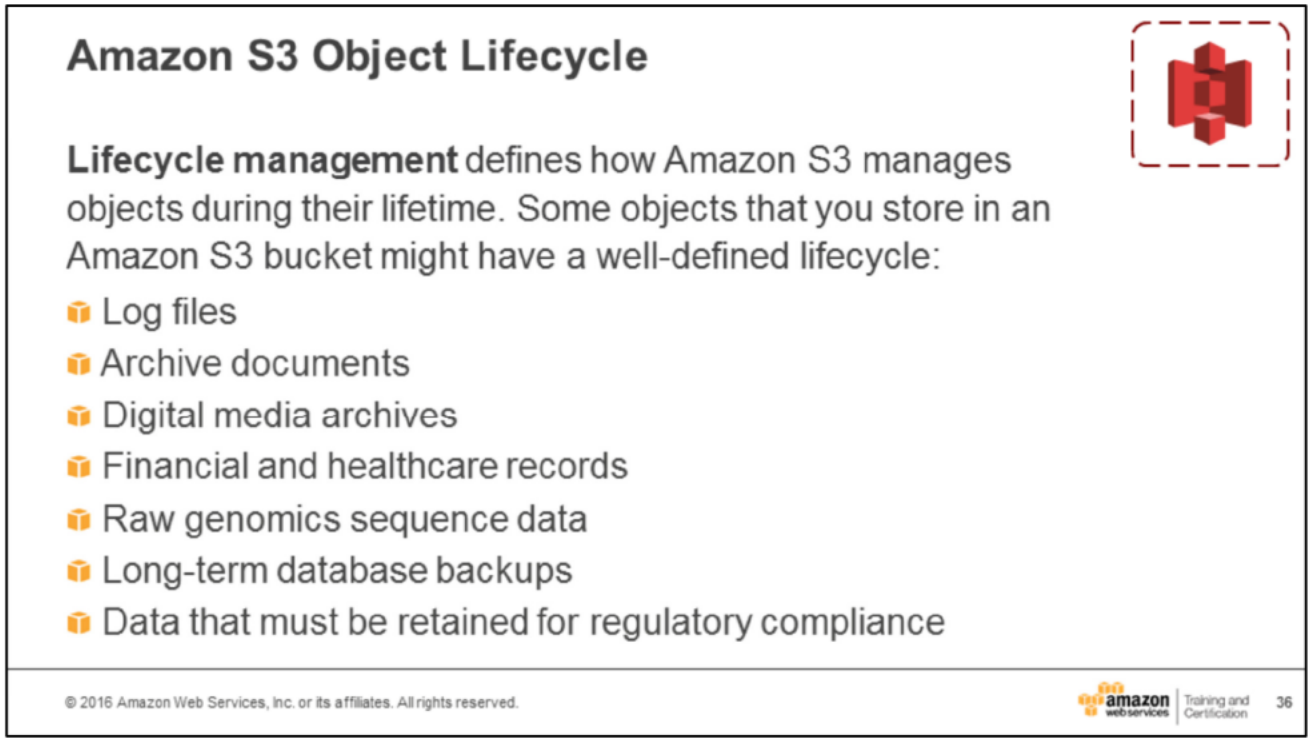
S3 Infrequent Access (IA) is optimized for long-lived and less frequently accessed data such as backups and older data that are accessed less but still require high performance.

Glacier is suitable for archiving data where access is infrequent and a retrieval time of several hours is acceptable. Archived objects are not available for real-time access – they must be restored before they can be accessed. The Glacier storage class is very low-cost.

S3 Reduced Redundancy Storage (RSS) is designed for noncritical, reproducible data stored at lower levels of redundancy standards than the Standard or IA classes, reducing cost.

For more information, see:

- http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html
- https://aws.amazon.com/s3/storage-classes/



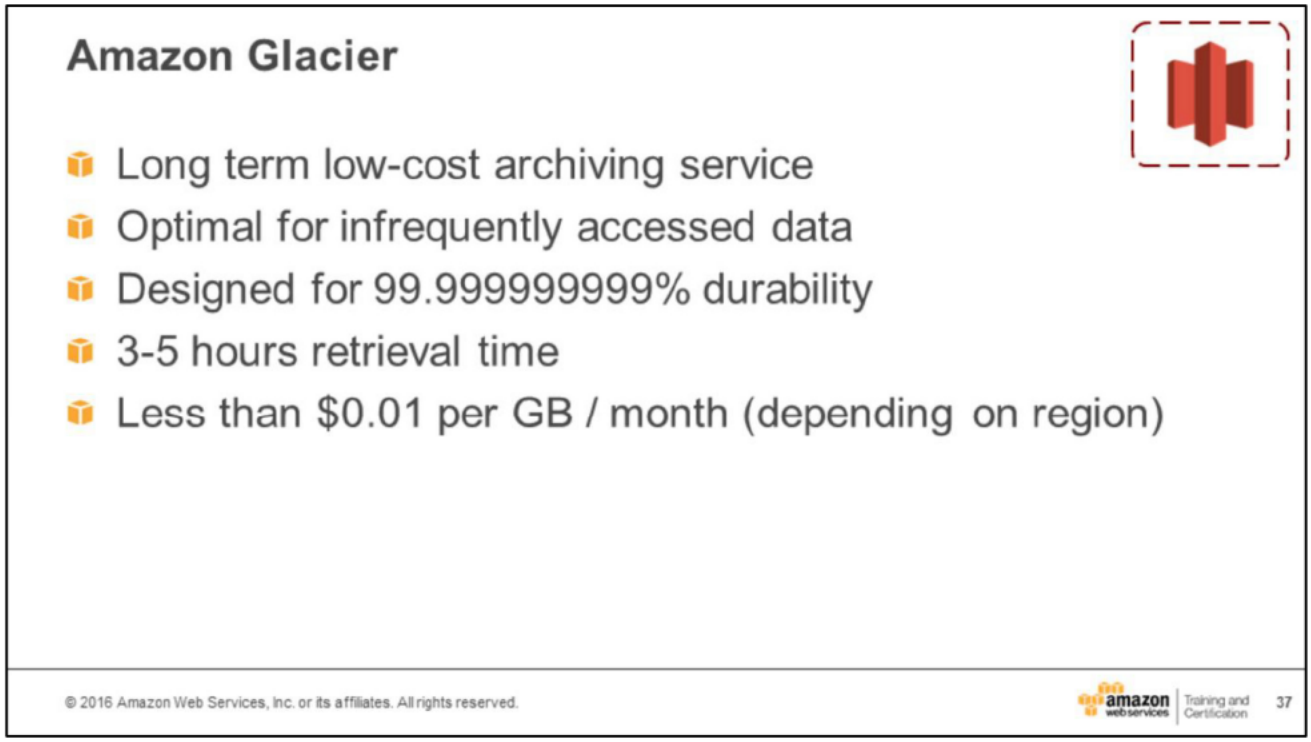
Lifecycle management defines how Amazon S3 manages objects during their lifetime. Some objects that you store in an Amazon S3 bucket might have a well-defined lifecycle: if you are uploading periodic logs to your bucket, your application might need these logs for a week or a month after creation, and after that you might want to delete them. Some documents are frequently accessed for a limited period of time. After that, you might not need real-time access to these objects, but your organization might require you to archive them for a longer period and then optionally delete them.

Digital media archives, financial and healthcare records, raw genomics sequence data, long-term database backups, and data that must be retained for regulatory compliance are some of the kinds of objects that you might upload to Amazon S3 primarily for archival purposes.

When you configure a lifecycle rule, you specify the storage class you want to transition the object to and the number of days after object creation to transition it. You can transition objects to the Standard – Infrequent Access (IA) storage class, archive them to Amazon Glacier, or have them permanently deleted. Standard - IA is useful for data such as backups and other older, infrequently accessed data where high performance continues to be a requirement. It is suitable for objects greater than 128 kilobytes that you want to keep for at least 30 days. There is a retrieval fee associated with Standard - IA objects.

For more information, see:

- Amazon S3 Pricing
- http://aws.amazon.com/solutions/case-studies/yelp/



For more information on Glacier pricing, see:

https://aws.amazon.com/glacier/pricing/

SoundCloud Case Study

SoundCloud:

- Operates worldwide.
- Enables users to upload **12 hours of audio material** to its platform **every minute**.
  - Each audio file must be transcoded and **stored in multiple formats**.
  - Logs and analyzes **billions of events**.

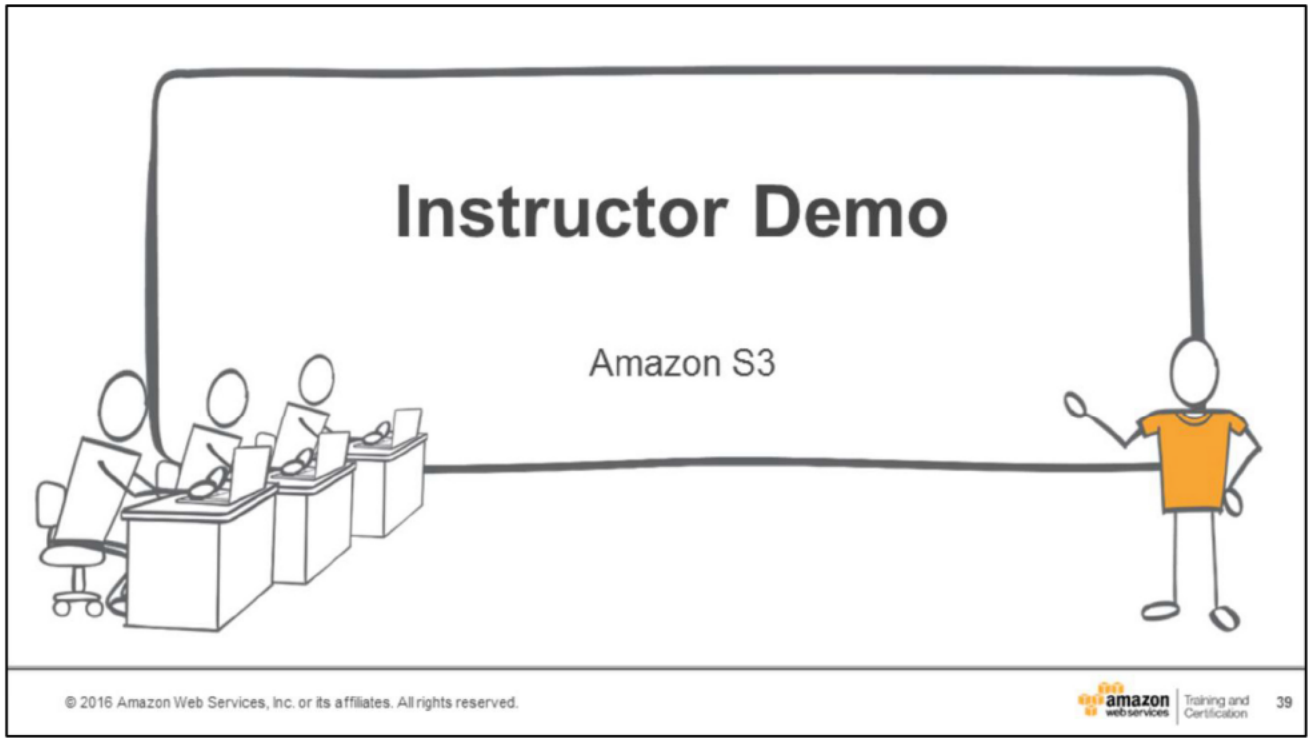
The AWS Solution:

- SoundCloud uses a storage solution comprised of:
  - Amazon S3
  - Amazon Glacier
- The audio files are:
  - Placed in Amazon S3.
  - Distributed from Amazon S3 via the SoundCloud website.
  - Copied to Amazon Glacier.
- The company currently stores **2.5 PB** of data on Amazon Glacier.

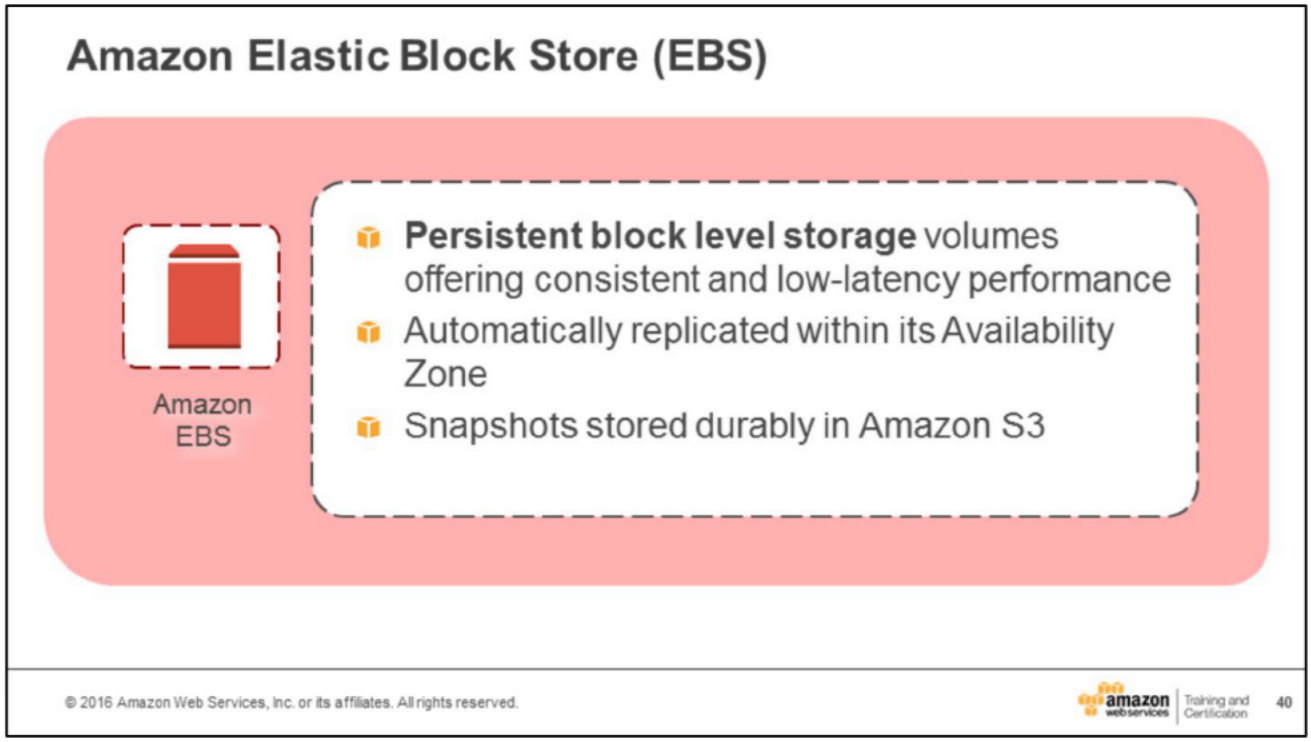
The SoundCloud case study demonstrates one of many ways in which Amazon S3 can be leveraged to securely store data.

For more information, see:

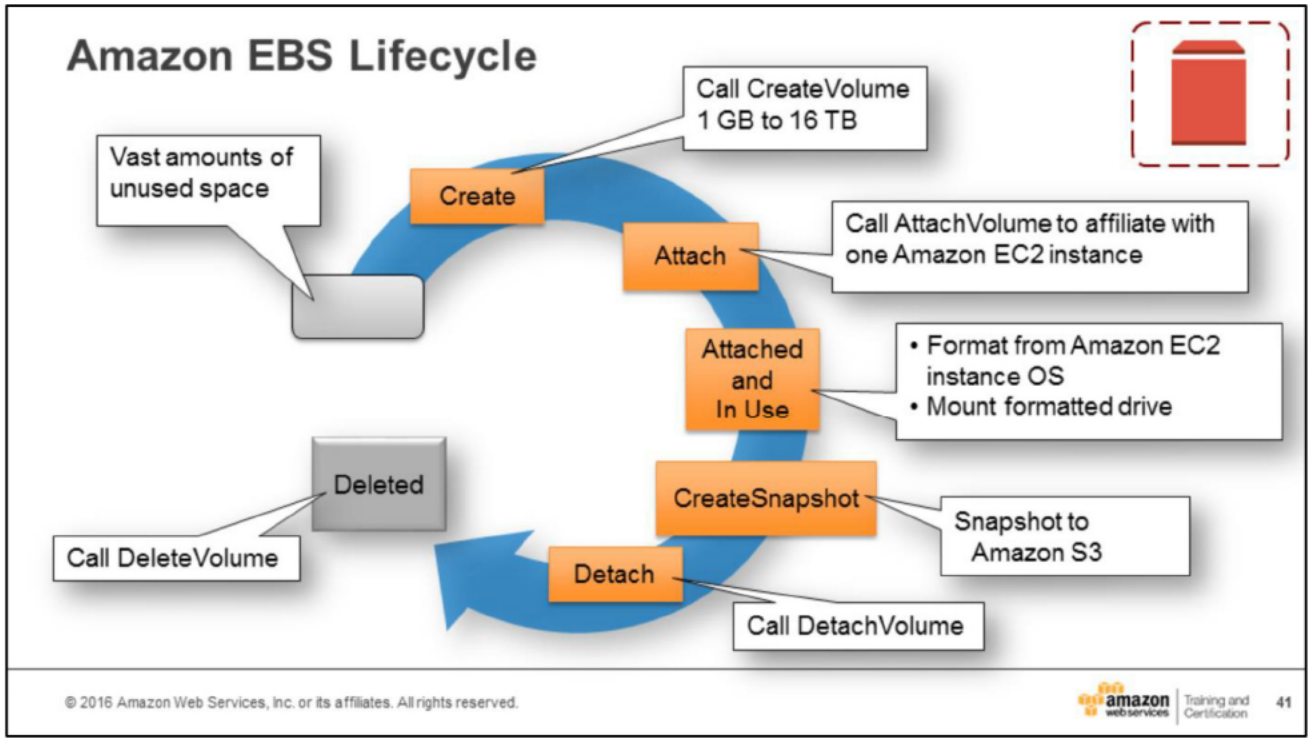
https://aws.amazon.com/solutions/case-studies/soundcloud/



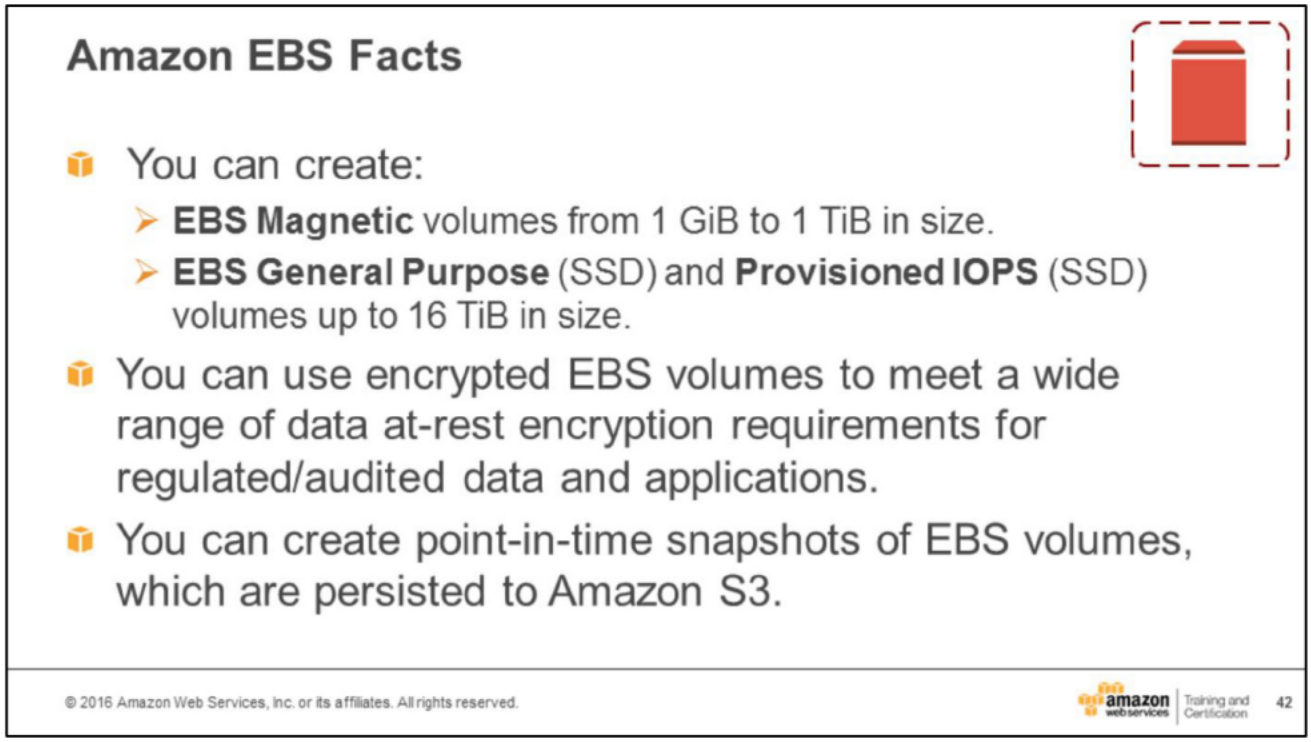
The instructor will demo creating a bucket, setting access control for the bucket with a bucket policy, enabling versioning and server-side encryption, uploading an object, and setting object lifecycle.



Amazon Elastic Block Store, also known as Amazon EBS, provides persistent block-level storage volumes for use with Amazon EC2 instance offering consistent and low-latency performance. Amazon EBS is particularly suited for applications that require a database, file system, or access to raw block-level storage. Amazon EBS snapshots are durable and automatically replicated within their Availability Zone. Snapshots can be stored in Amazon S3.



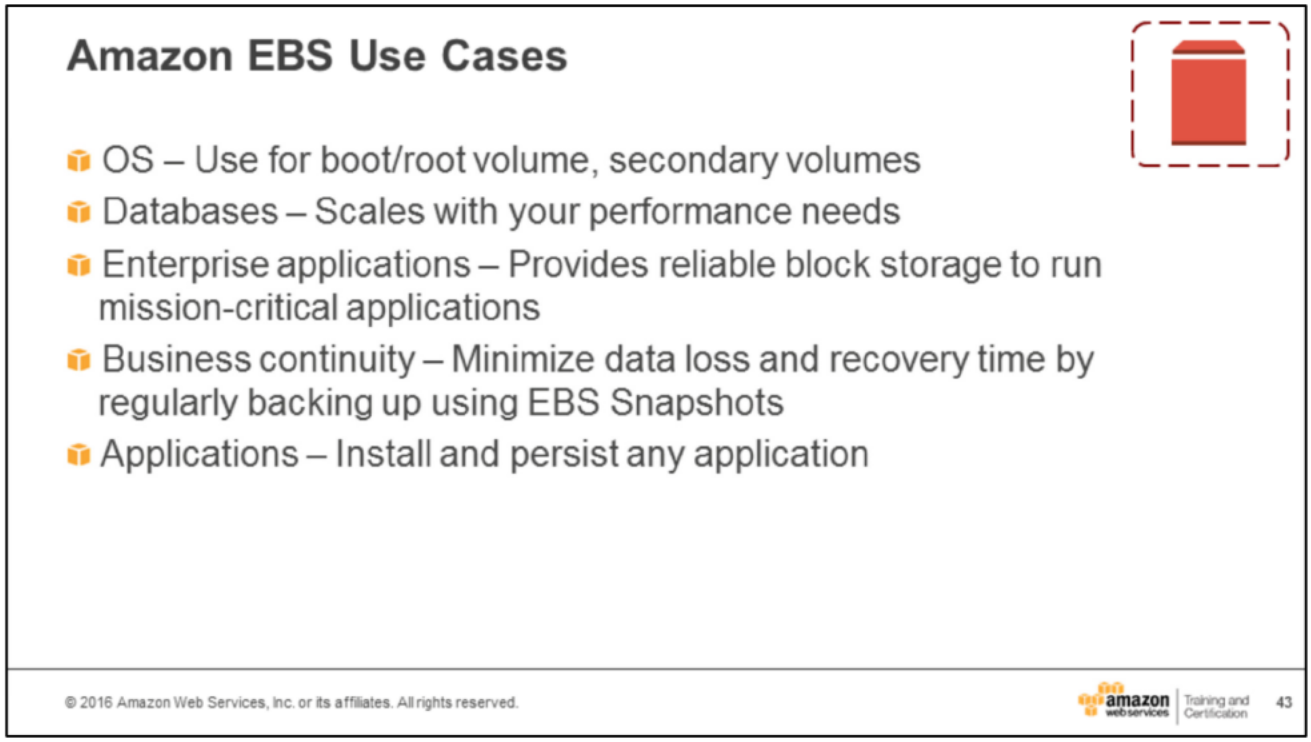
Amazon EBS provides block-level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance in the same Availability Zone. The Amazon EBS volumes attached to an Amazon EC2 instance are exposed as storage volumes that persist independently from the life of the instance. When the volumes are not attached to an EC2 instance, you pay only for the cost of storage.



Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you pay only for what you use.

You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive).

Amazon EBS is recommended when data changes frequently and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is particularly helpful for database-style applications that frequently encounter many random reads and writes across the data set.



The Amazon EBS service is simply a virtual hard drive. So, a great use case for Amazon EBS is when you want the hard drive to persist past the life of the Amazon EC2 instance. Before Amazon EBS existed as a service, AWS only used physical local attached hard drives called ephemeral storage. The problem with that was that you couldn't stop an Amazon EC2 instance without losing all your data, because of the temporary nature of local storage.

That's why we created Amazon EBS to decouple the lifecycle of data persistence from the lifecycle of an EC2 instance. Amazon EBS volumes are ideal for root volumes you need to store and have block-level access to your operating system, database storage, and datasets that are smaller than 1 TB. Given its simple snapshot mechanism Amazon EBS is a great use case for simplifying distributed backups as well.

For more information, see:

Dropcam Case Study using AWS and Amazon EBS - http://aws.amazon.com/solutions/case-studies/dropcam/

Amazon EBS Pricing

Pay for what you provision:

- Pricing based on region
- AWS GovCloud (US) Pricing page
- Review Pricing Calculator online
- Pricing is available as:
  - Storage
  - IOPS

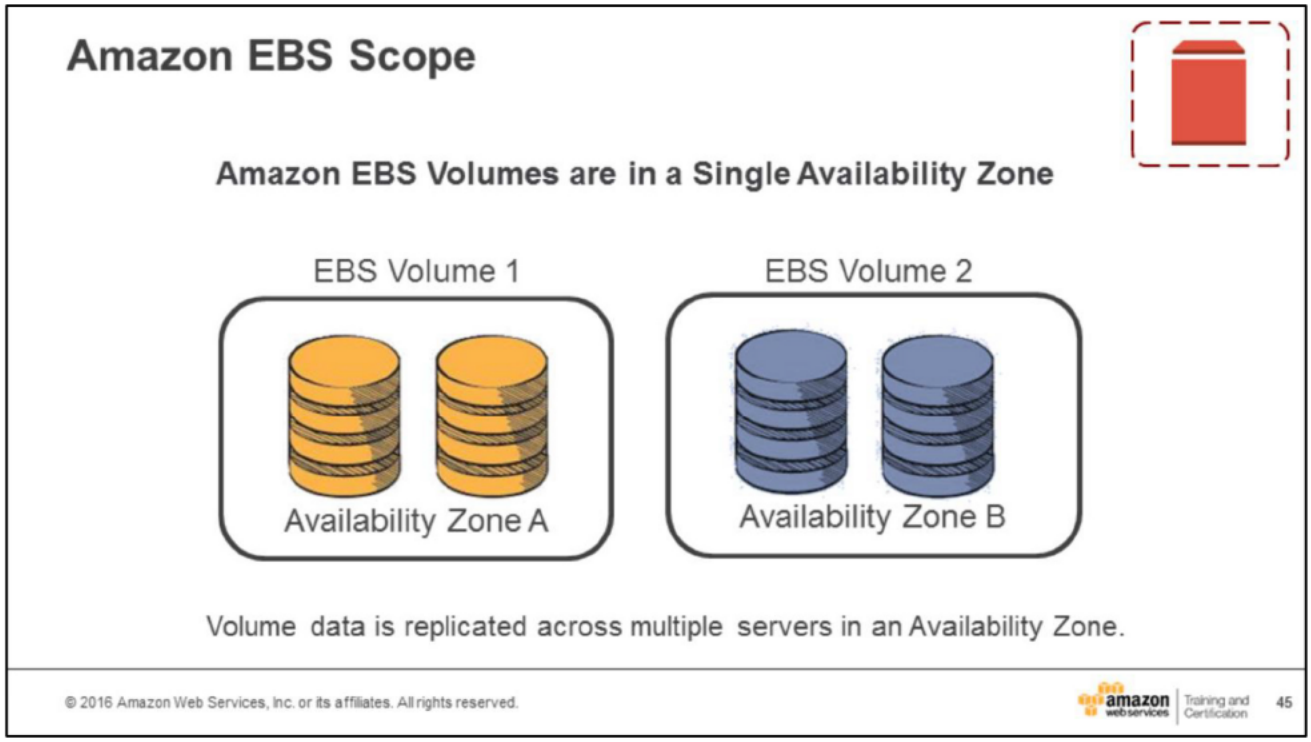
** Check Amazon EBS Pricing page for current pricing for all regions.*

amazon web services | Training and Certification 44

Amazon EBS pricing is based on allocated storage, whether you use it or not. This is unlike Amazon S3, whose pricing is based on space actually in use. Prices may vary based on region or for IOPS.

For more information, see:

- Check online for current pricing for all regions - http://aws.amazon.com/ebs/pricing/
- Gov. Cloud Pricing Page - http://aws.amazon.com/govcloud-us/pricing/
- AWS Simple Monthly Calculator - http://calculator.s3.amazonaws.com/index.html



Amazon EBS volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.

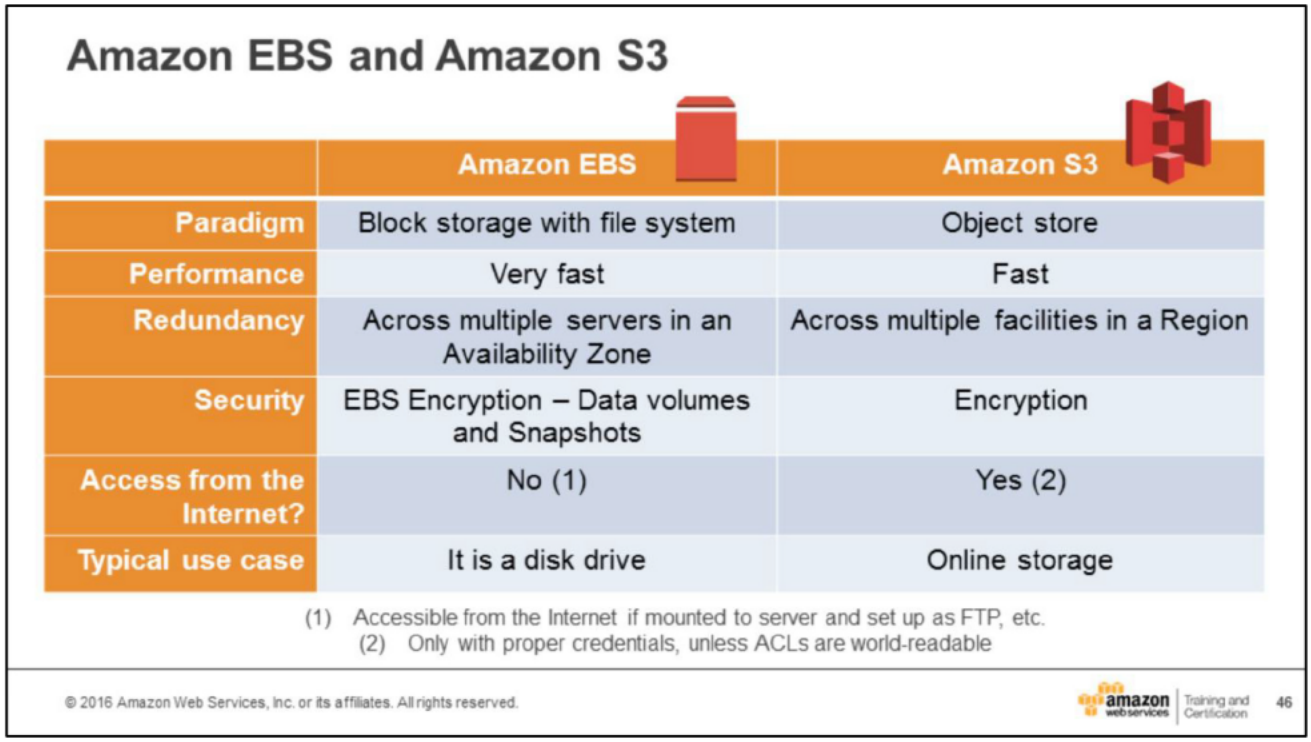
The durability of your volume depends on both the size of your volume and the percentage of the data that has changed since your last snapshot.

Amazon EBS volumes are designed for an annual failure rate (AFR) of between 0.1% - 0.2%, where failure refers to a complete or partial loss of the volume, depending on the size and performance of the volume. This is compared with commodity hard disks that will typically fail with an AFR of around 4 percent, making EBS volumes 10 times more reliable than typical commodity.

Since Amazon EBS servers are replicated within a single Availability Zone, mirroring data across multiple Amazon EBS volumes in the same Availability Zone will not significantly improve volume durability.

For those interested in even more durability, with Amazon EBS you can create point-in-time consistent snapshots of your volumes that are then stored in Amazon S3, and automatically replicated across multiple Availability Zones.

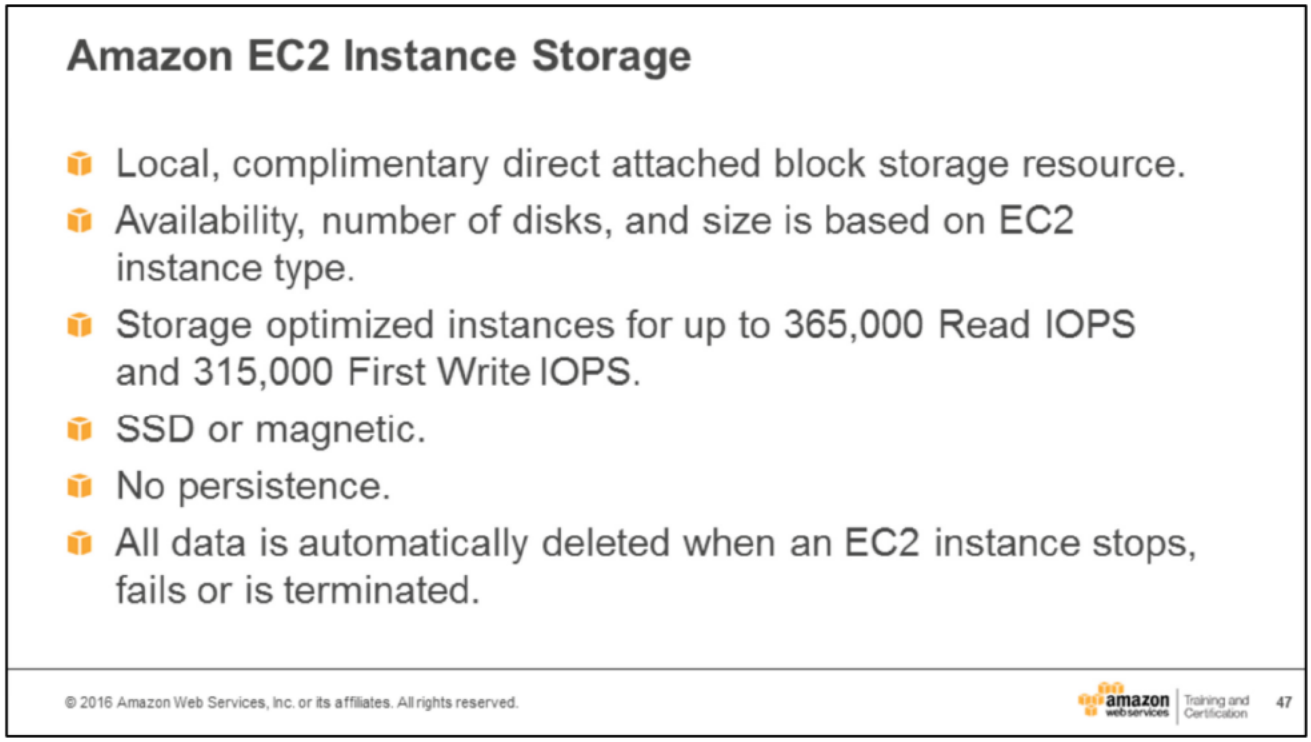
Taking frequent snapshots of your volume is a convenient and cost-effective way to increase the long-term durability of your data. In the unlikely event that your Amazon EBS volume does fail, all snapshots of that volume will remain intact, and will allow you to recreate your volume from the last snapshot point.



This table demonstrates significant differences between Amazon S3 and Amazon EBS. Amazon EBS volumes are network-attached hard drives that can be written to or read from at a block level. Amazon S3 is an object-level storage medium.

This means that you must write whole objects at a time. If you change one small part of a file, you must still rewrite the entire file in order to commit the change to Amazon S3. This can be very time-consuming if you have frequent writes to the same object.

Amazon S3 is optimized for write once/read many use cases. The other major difference is cost. With Amazon S3 you pay for what you use, and with Amazon EBS you pay for what you provision.



An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

For more information, see:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-store-volumes
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/i2-instances.html

Reboot vs. Stop vs. Terminate

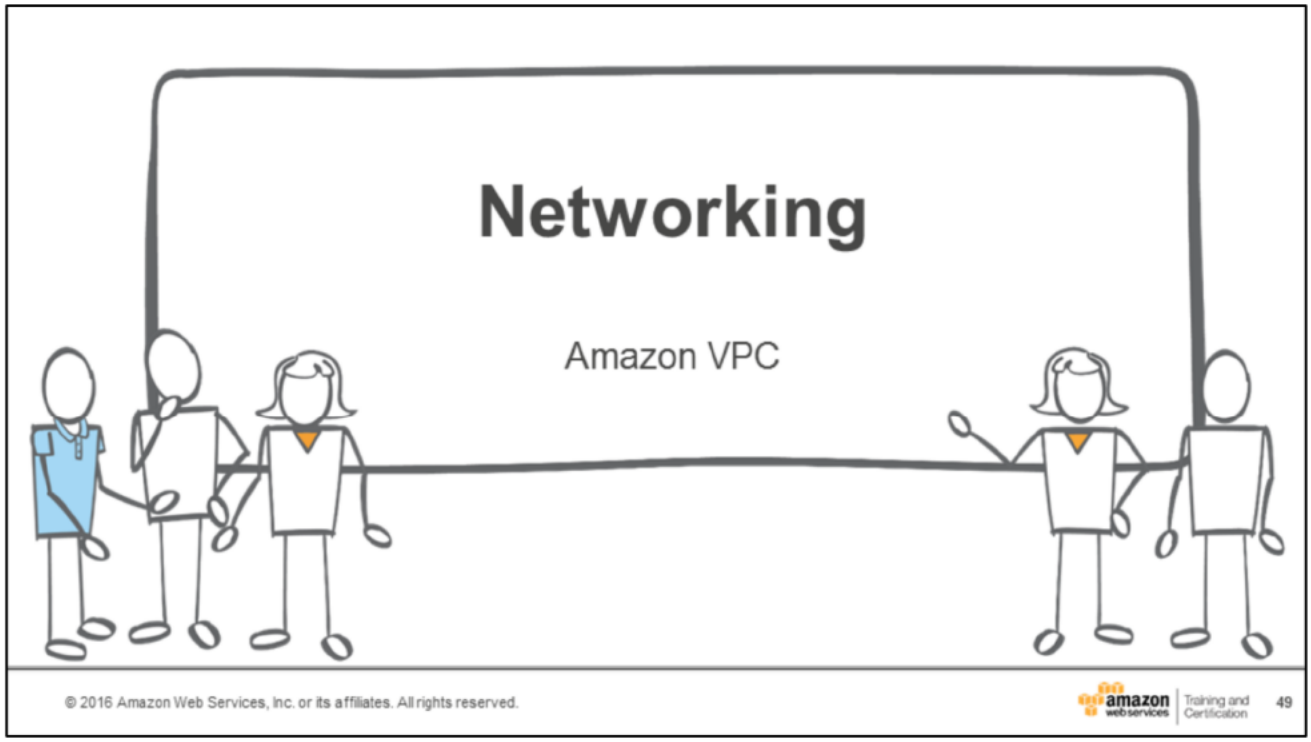
Characteristic	Reboot	Stop/Start (EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer.	The instance runs on a new host computer.	N/A
Private and public IP addresses	Stay the same.	Instance keeps its private IP address and gets a new public IP address.	N/A
Elastic IP addresses (EIP)	EIP remains associated with the instance.	EIP remains associated with the instance.	The EIP is disassociated from the instance.
Instance store volumes	The data is preserved.	The data is erased.	The data is erased.
EBS volume	The volume is preserved.	The volume is preserved.	The volume is deleted by default.
Billing	Instance billing hour doesn't change.	You stop incurring charges as soon as state is changed to stopping.	You stop incurring charges as soon as state is changed to shutting-down.

amazon web services | Training and Certification 48

The table shows the differences between rebooting, stopping, and terminating your instance.

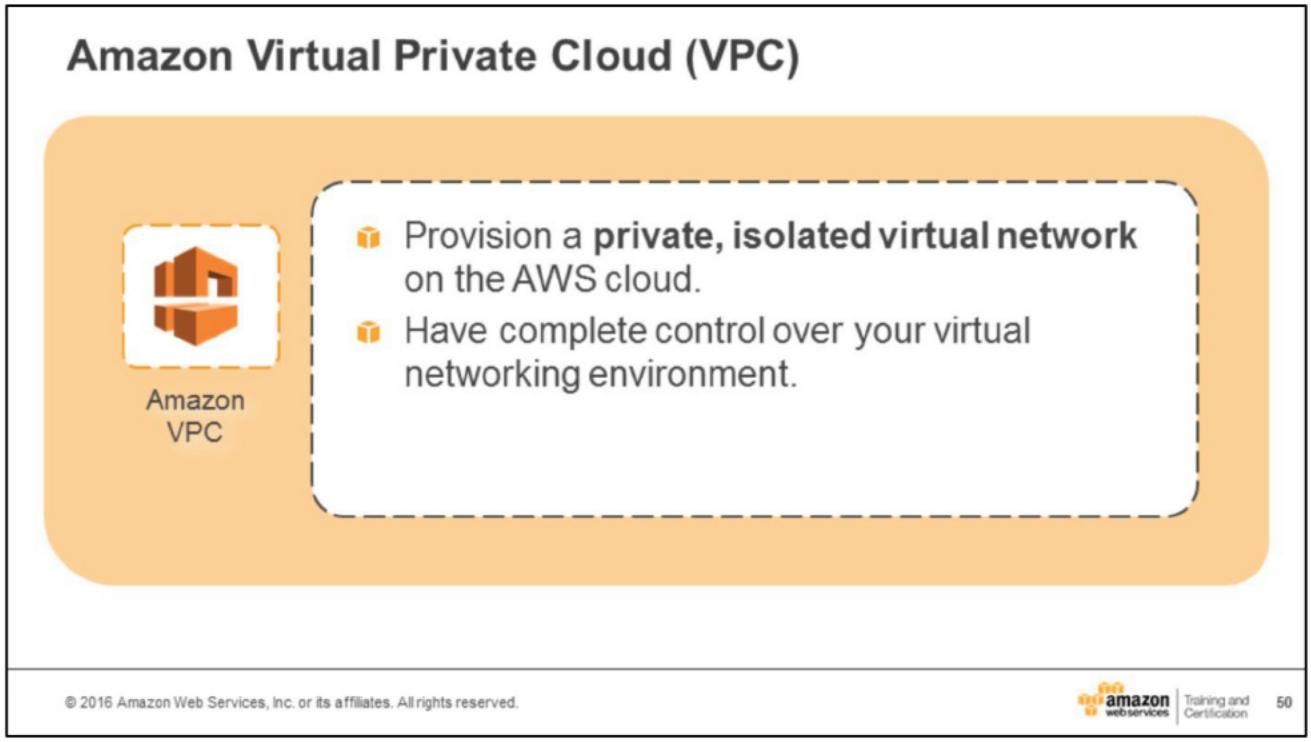
For more information, see:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

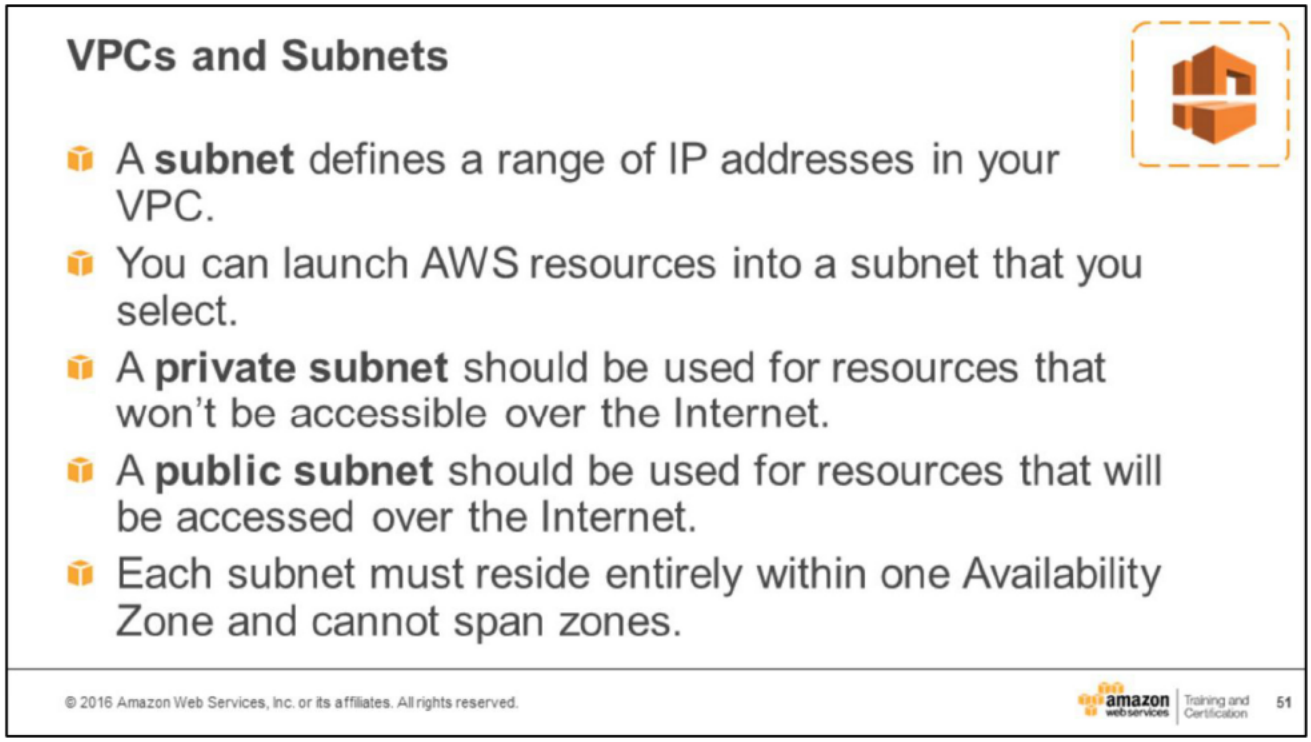


Understand Virtual Private Cloud (VPC) concepts including:

- Subnets
- Security
- Networking
- VPN



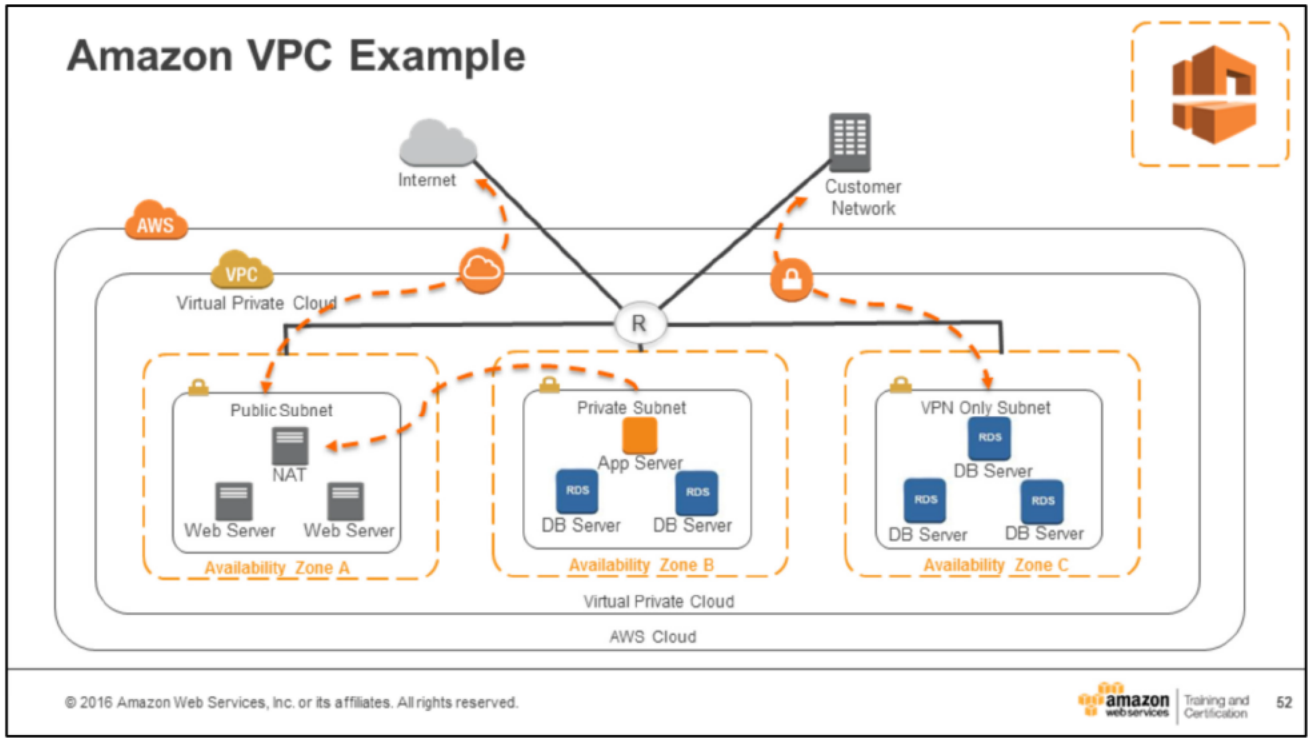
With Amazon Virtual Private Cloud (VPC), you can define a virtual network topology that closely resembles a traditional network that you might operate in your own data center. You have complete control over your virtual networking environment, and you can easily customize the network configuration for your Amazon VPC such as selection of IP address range, creation of subnets, configuration of route tables, and network gateways.



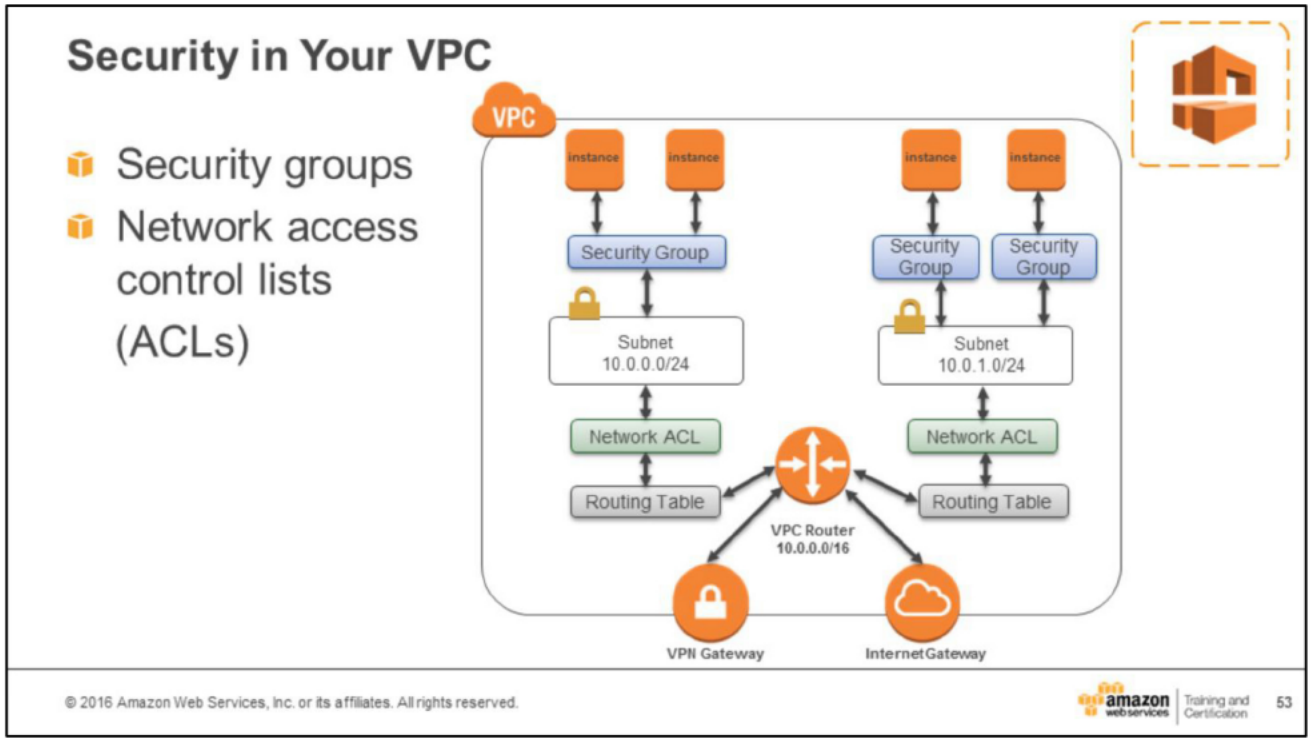
AWS assigns a unique ID to each subnet. Regardless of the type of subnet (public or private), the internal IP address range of the subnet is always private.

A public subnet has a route to an internet gateway (i.e., for a web server accessible from the Internet).

A private subnet has no route to an internet gateway (i.e., for a database server only accessed within the VPC).



Amazon Virtual Private Cloud also known as Amazon VPC, allows you provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, configuration of route tables, network access control lists, and network gateways. You can easily customize the network and configuration for your Amazon VPC instance. For example, you can create a public-facing subnet for your web servers that require access to the Internet, and place your back end systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet. Additionally, you can create a hardware virtual private network (VPN) connection between your corporate data center and your VPC, allowing you to leverage the AWS cloud as an extension of your corporate data center.



Amazon VPC provides three features that you can use to increase and monitor the security for your VPC:

1. Security groups act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.
2. Network access controls lists (ACLs) act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level.

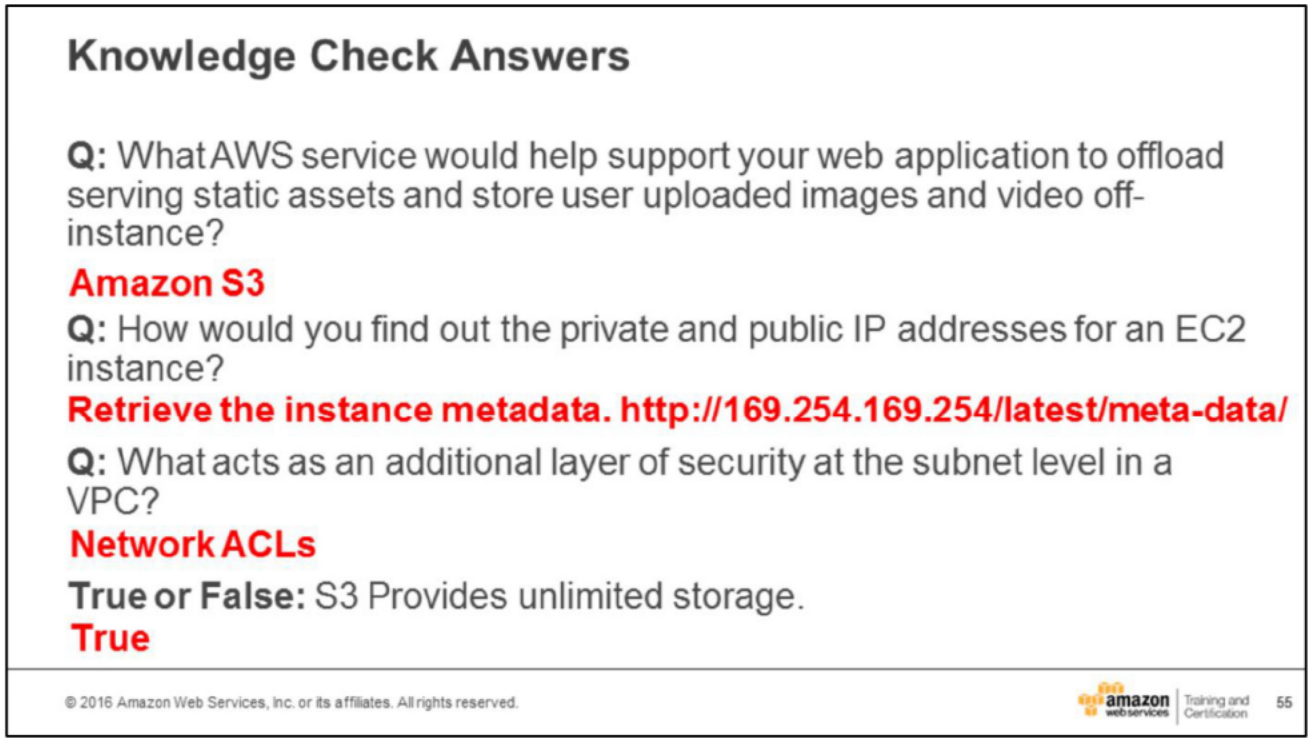
VPN Connections

VPN Connectivity option	Description
AWS Hardware VPN	You can create an IPsec, hardware VPN connection between your VPC and your remote network.
AWS Direct Connect	AWS Direct Connect provides a dedicated private connection from a remote network to your VPC.
AWS VPN CloudHub	You can create multiple AWS hardware VPN connections via your VPC to enable communications between various remote networks.
Software VPN	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a software VPN appliance.

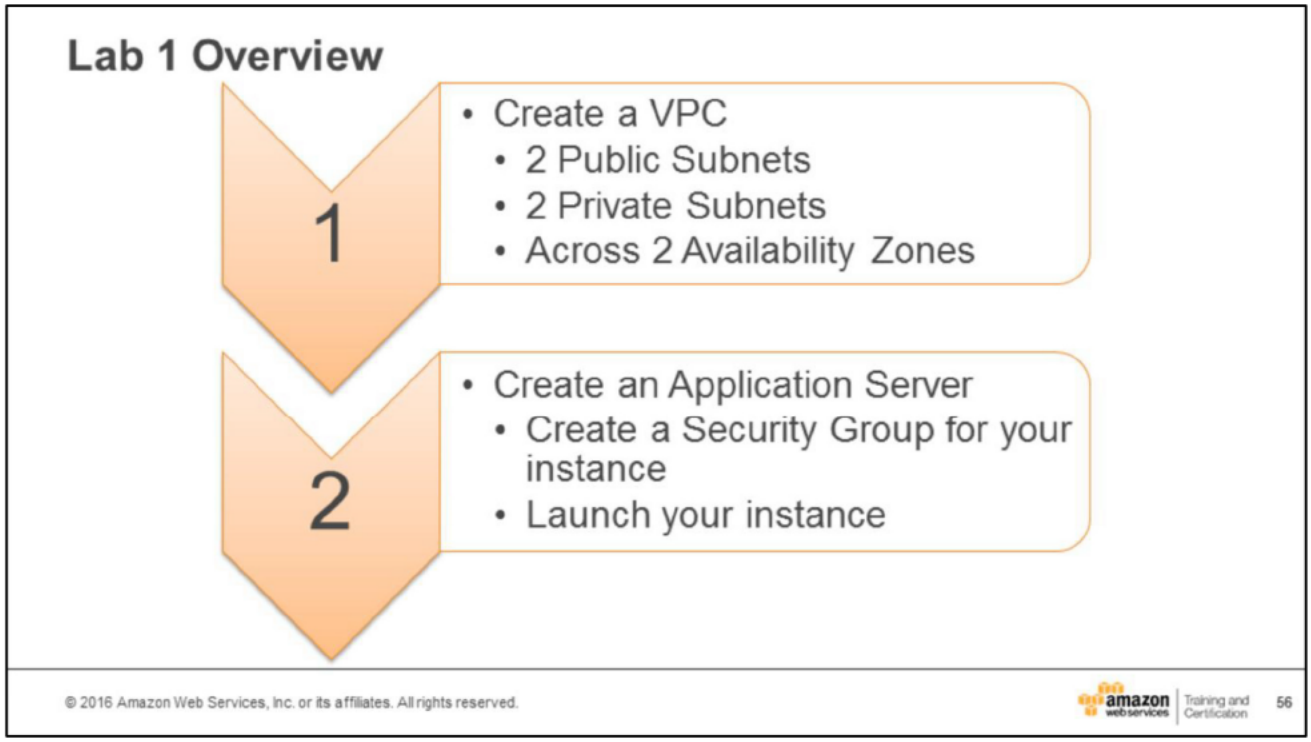
You can connect your VPC to remote networks by using a VPN connection.

For more information, see:

- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpn-connections.html
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPN_CloudHub.html



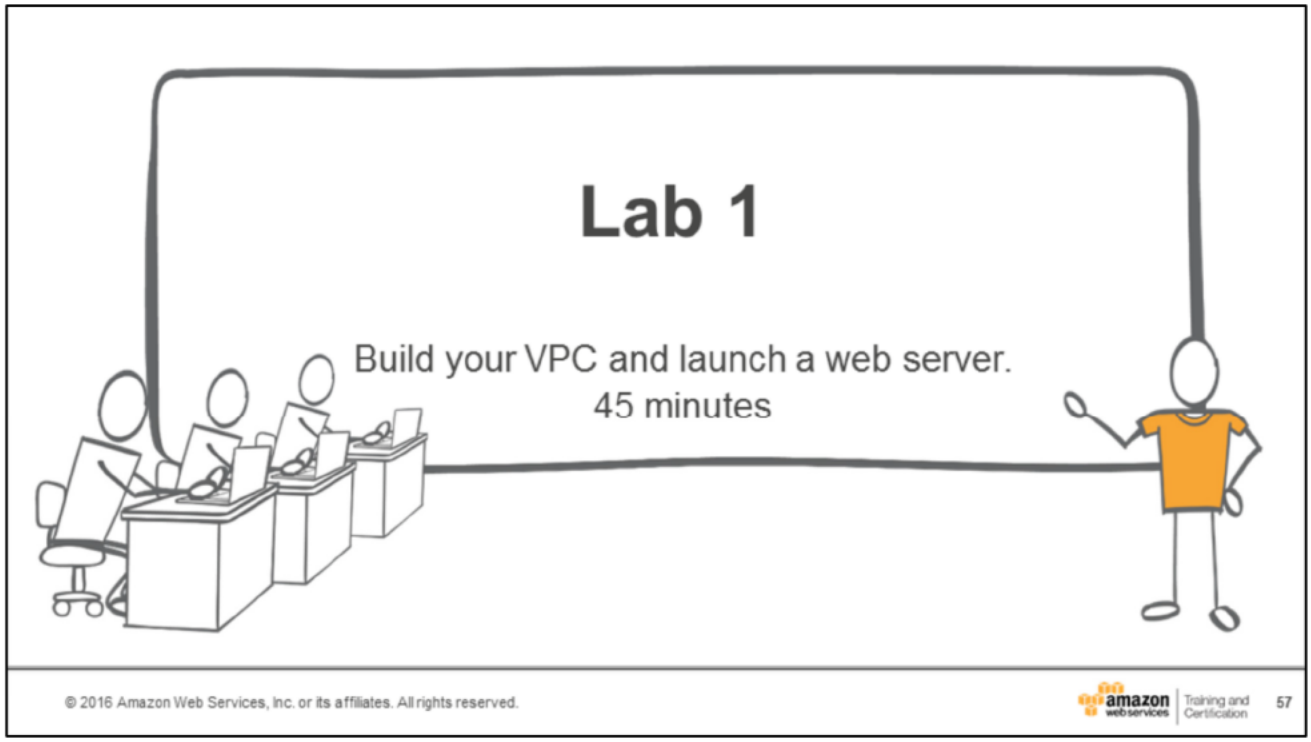
The answers to the knowledge check are shown in the slide.



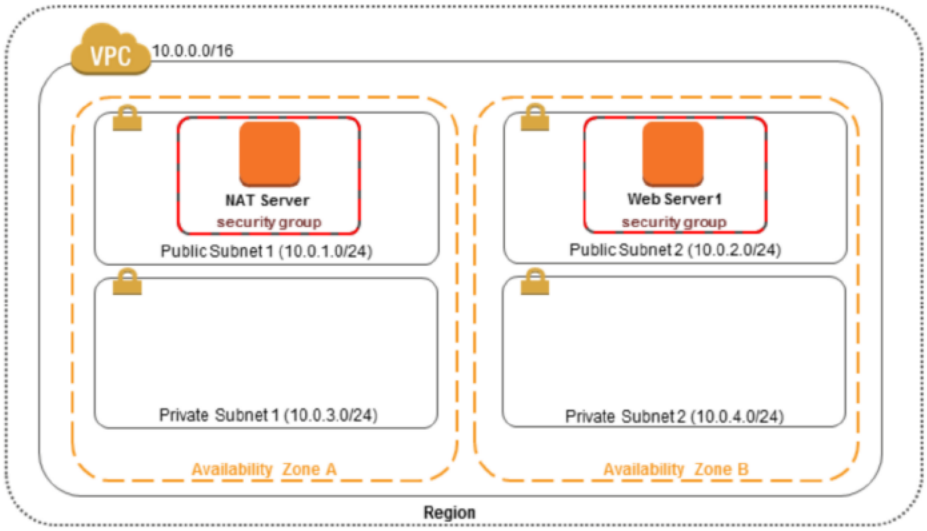
For more information, see:

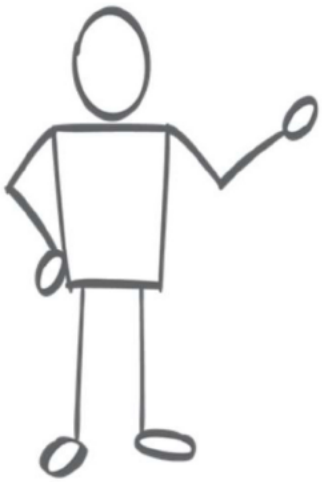
AWS Web Hosting Best Practices:
https://media.amazonwebservices.com/AWS_Web_Hosting_Best_Practices.pdf

Hosting a Web App on Amazon Web Services:
http://docs.aws.amazon.com/gettingstarted/latest/wah-linux/web-app-hosting-intro.html

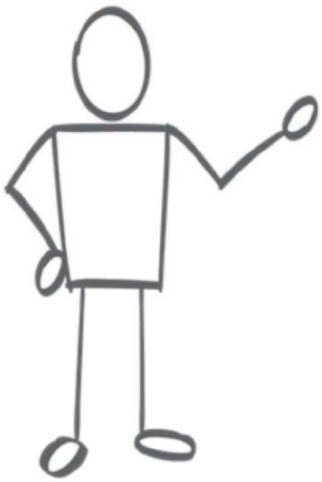


Lab 1 – What You Created

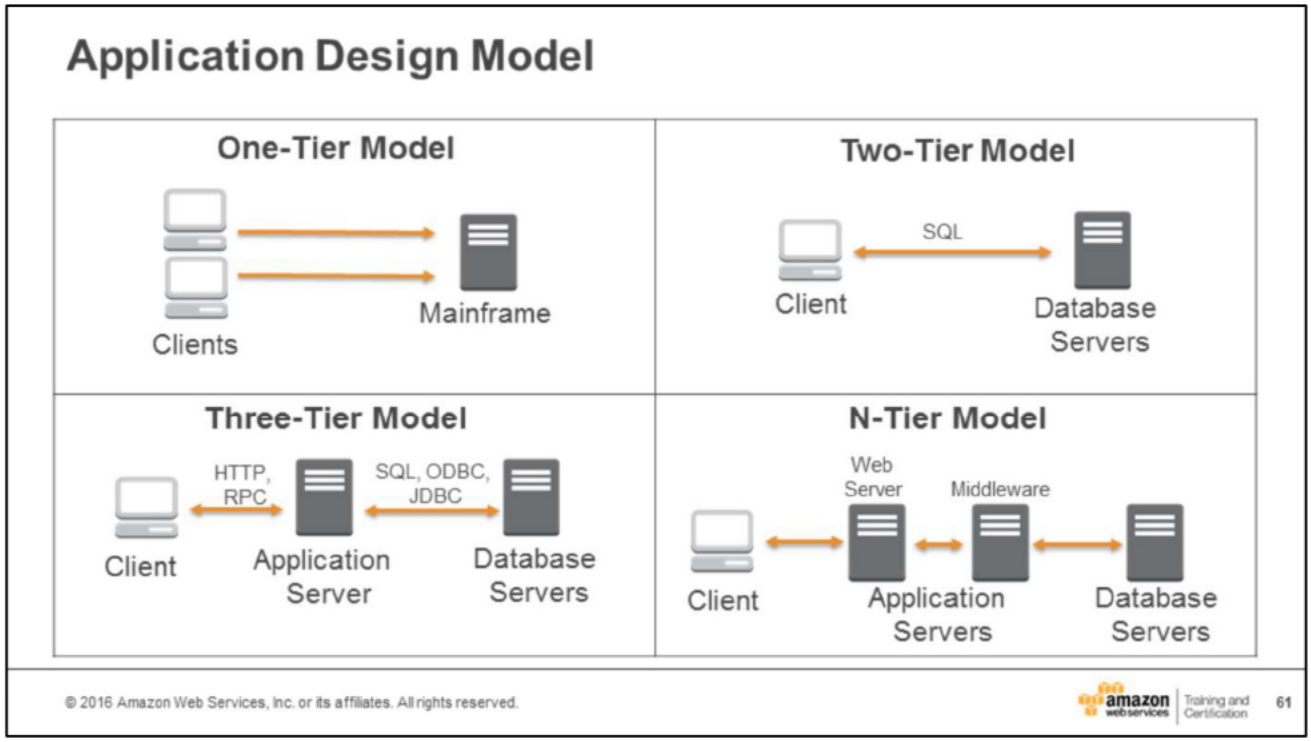




Appendix



Data Center Design Models

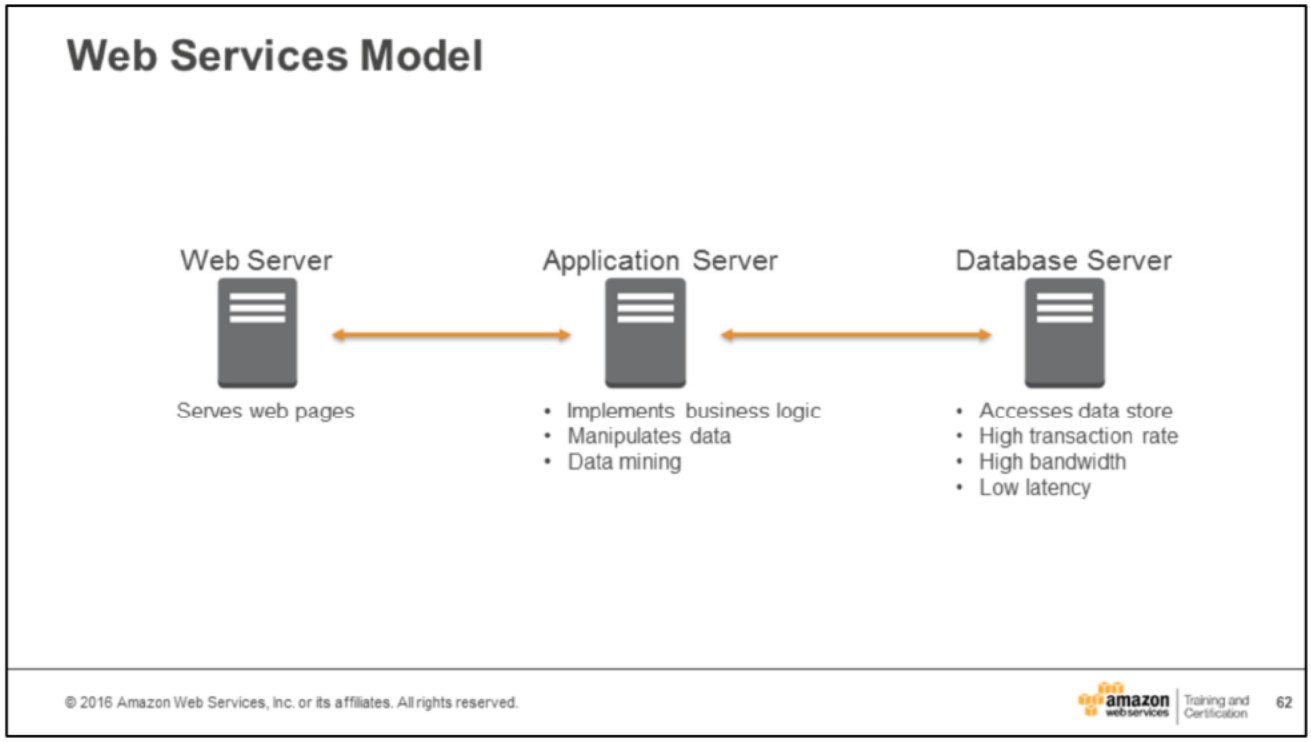


The one-tier model applies to clients (no local processing or storage) connected to a mainframe. This design is usually used for monolithic applications such as kiosks. This model has limited scalability and lacks flexibility.

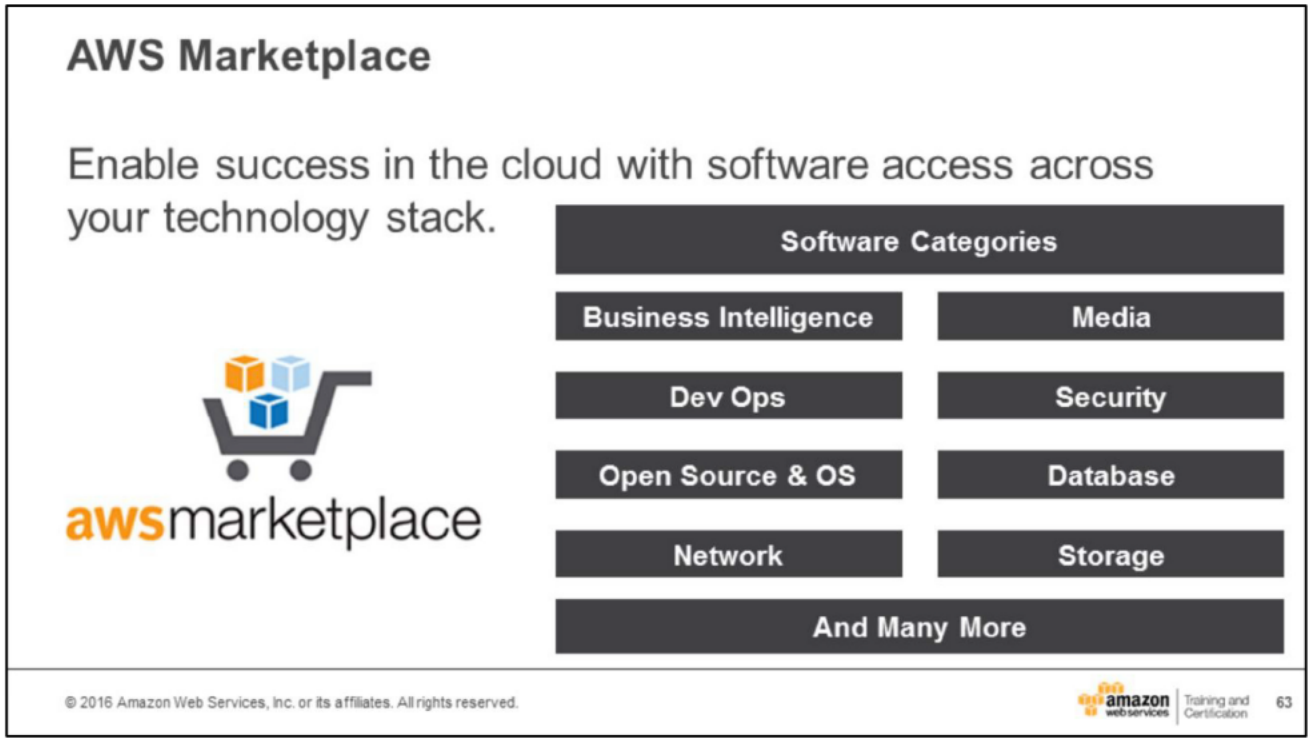
The two-tier model applies to clients connected to database servers. The clients have direct interaction with the database server and do some local application processes. This model has limited scalability and it not recommended for critical applications.

The three-tier model applies to clients connected to application servers. The application servers are then connected to the database servers. This model has more scalability than the first two models.

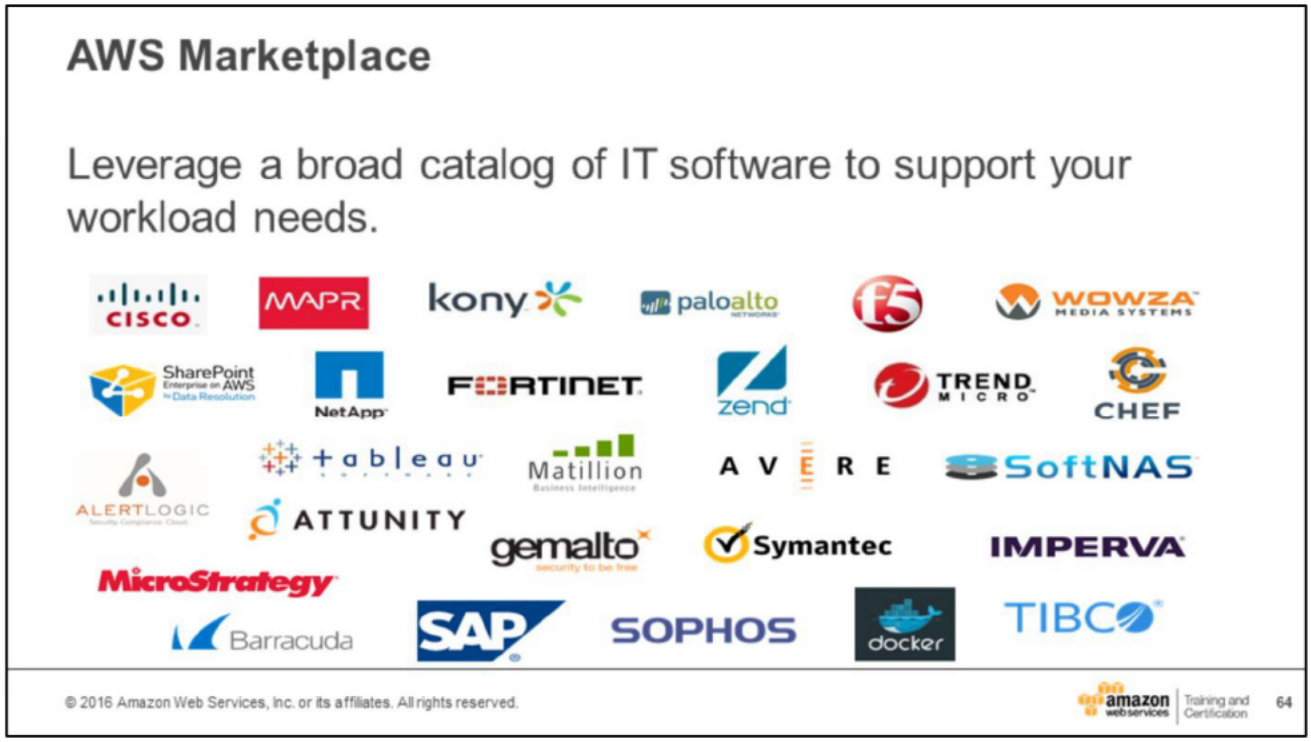
The n-tier model consists of clients connected to n number of application servers, connected to n number of database servers. This model has the most scalability of traditional data center design models and a robust partitioning of application functionality.



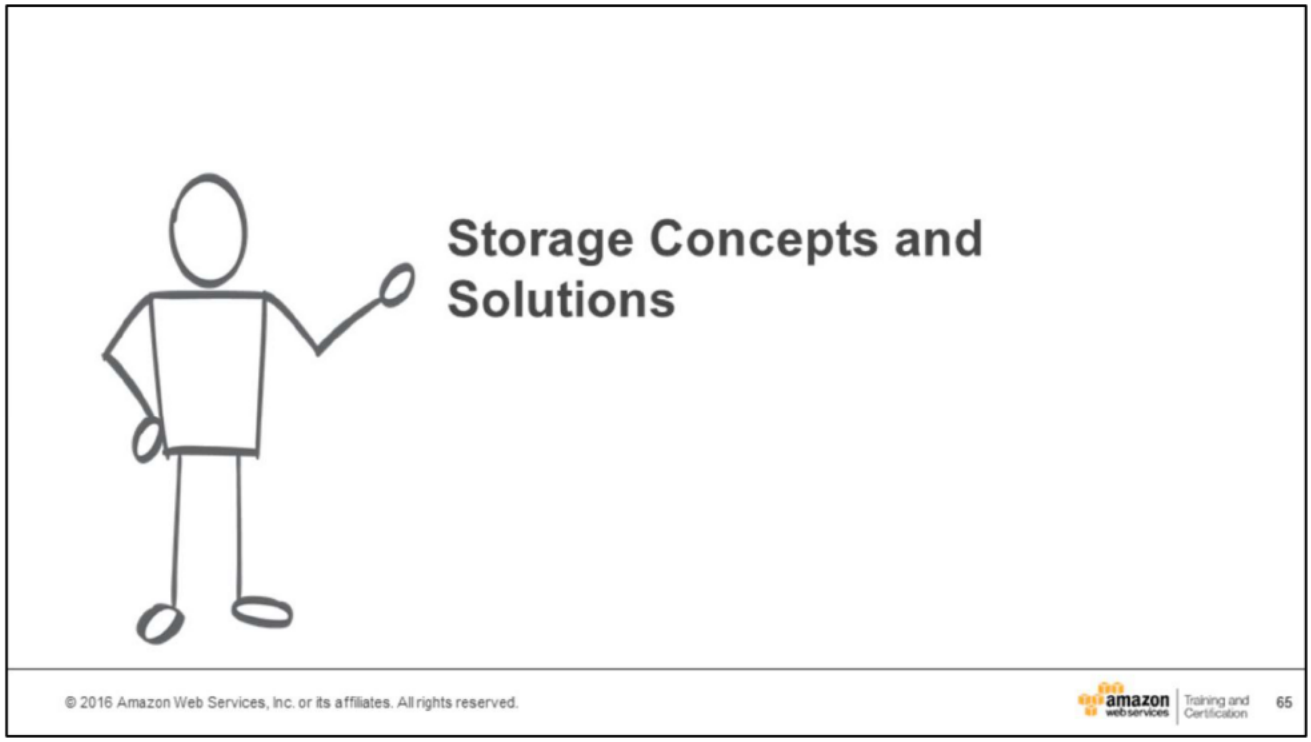
The Web Services Model includes web server, application server, and database server tiers. It includes tasks performed by multiple hosts with specific roles.



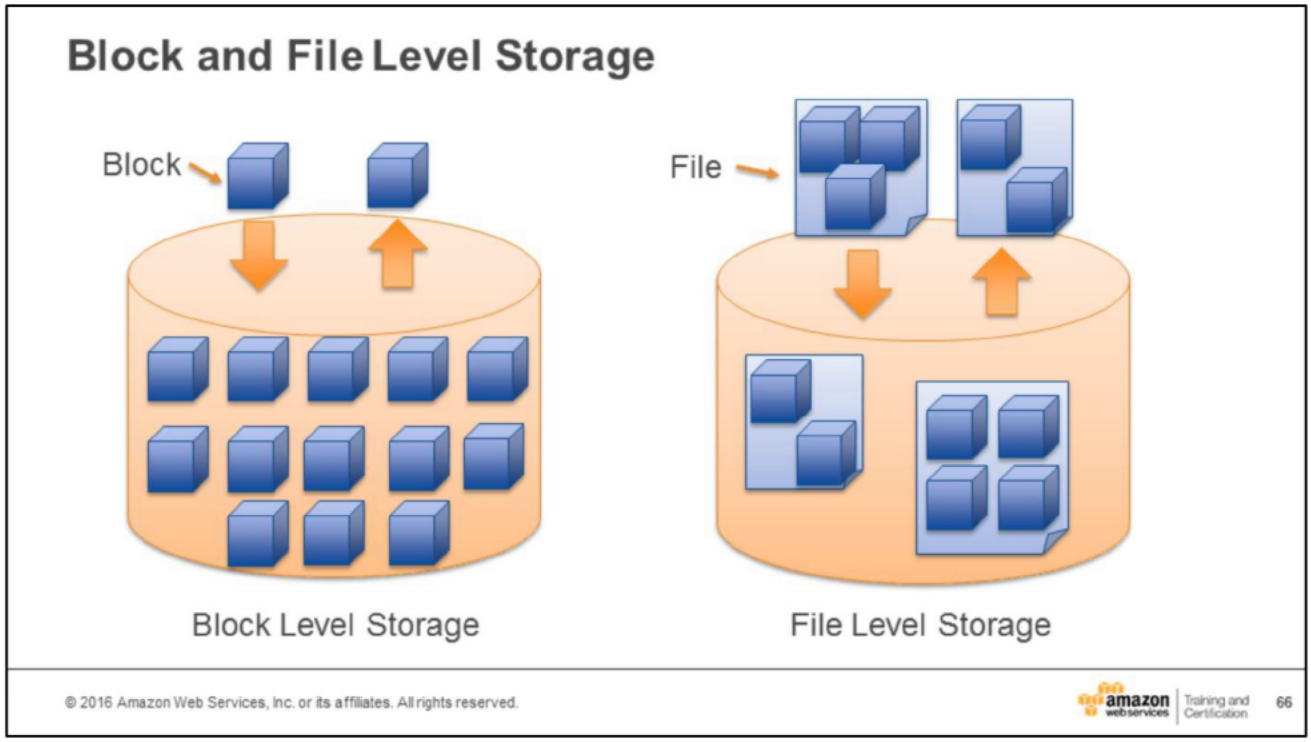
The AWS Marketplace Catalog includes 35 categories of software and more than 2,300 software listings from over 800 ISVs.



Many of the Earth's top IT software vendors are available on the AWS Marketplace. Many products in the catalog have multiple deployment models to fit your needs, including: Bring Your Own License , Hourly/Pay-as-you-go payment models, Software as a Service (SaaS), and Free Trials.

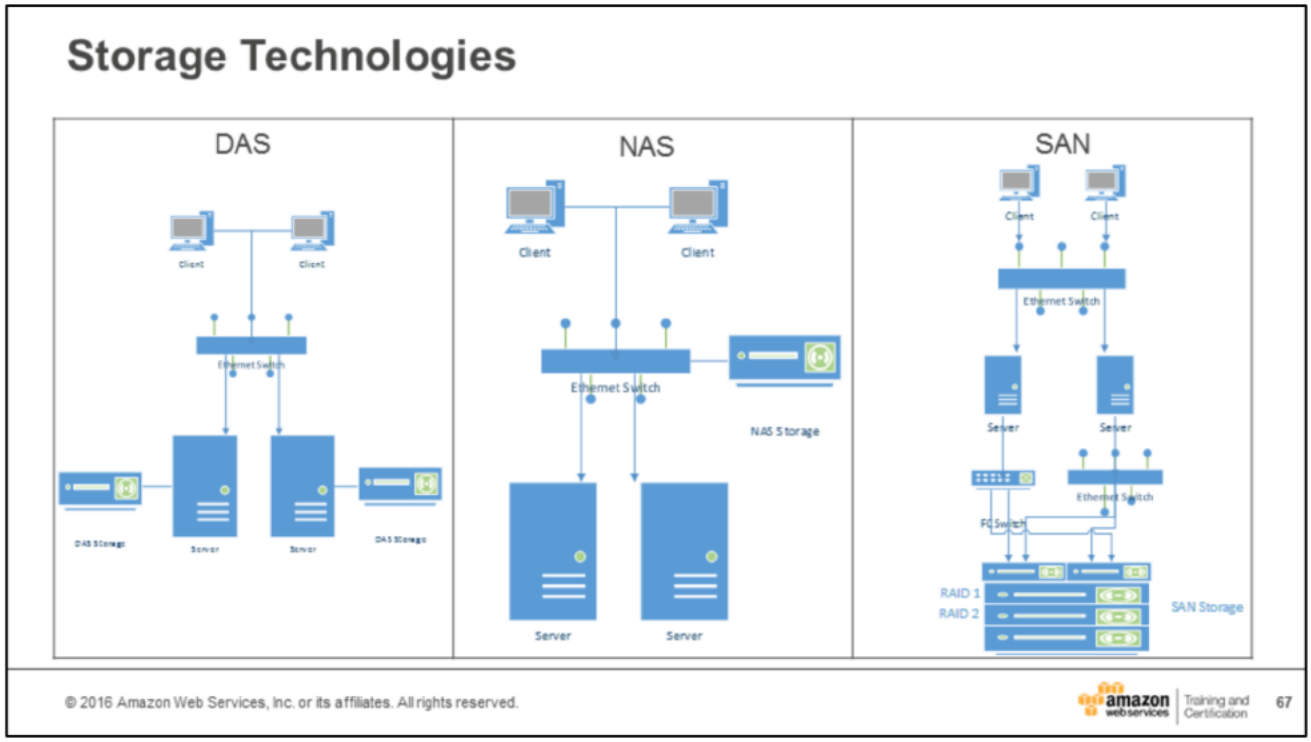


Understand common storage concepts and solutions related to servers and application environments.



Block Level Storage: This is the reading and writing of data to and from a disk using a sequence of bytes in a predetermined length. Files are spread over many blocks. Block level incremental backups are faster as they do not have to backup the entire file each time. Instead they only backup new blocks or blocks that have changed since the last backup. Example of block storage solutions are Direct Attached Storage (DAS) and network based Storage Area Network (SAN). Common DAS protocols are ATA, SATA, SCSI, SAS, and USB. Common SAN protocols are iSCSI and FCoE. With block level storage, a logical unit number (LUN) can be treated as a physical drive. Files are divided into many blocks and if any identical blocks are found, redundant copies are eliminated.

File Level Storage: In file level storage, files and folders can be accessed and managed by the storage system, but the smaller storage blocks that make up the files and folders cannot be directly controlled. Storage drives need to be configured with a storage protocol like Network File System (NFS), Server Message Block (SMB), or Common Internet File System (CIFS). With file level storage, a shared folder can be mounted as a network drive. If one file has to be accessed on different systems, file level storage would be an ideal choice.



The slide shows commonly used storage technologies.

Direct-Attached Storage (DAS): DAS is storage directly attached to the computer or client using it.

Network Attached Storage (NAS) Systems: NAS appliances are servers configured with software designed specifically for storing and providing access to files over a LAN.

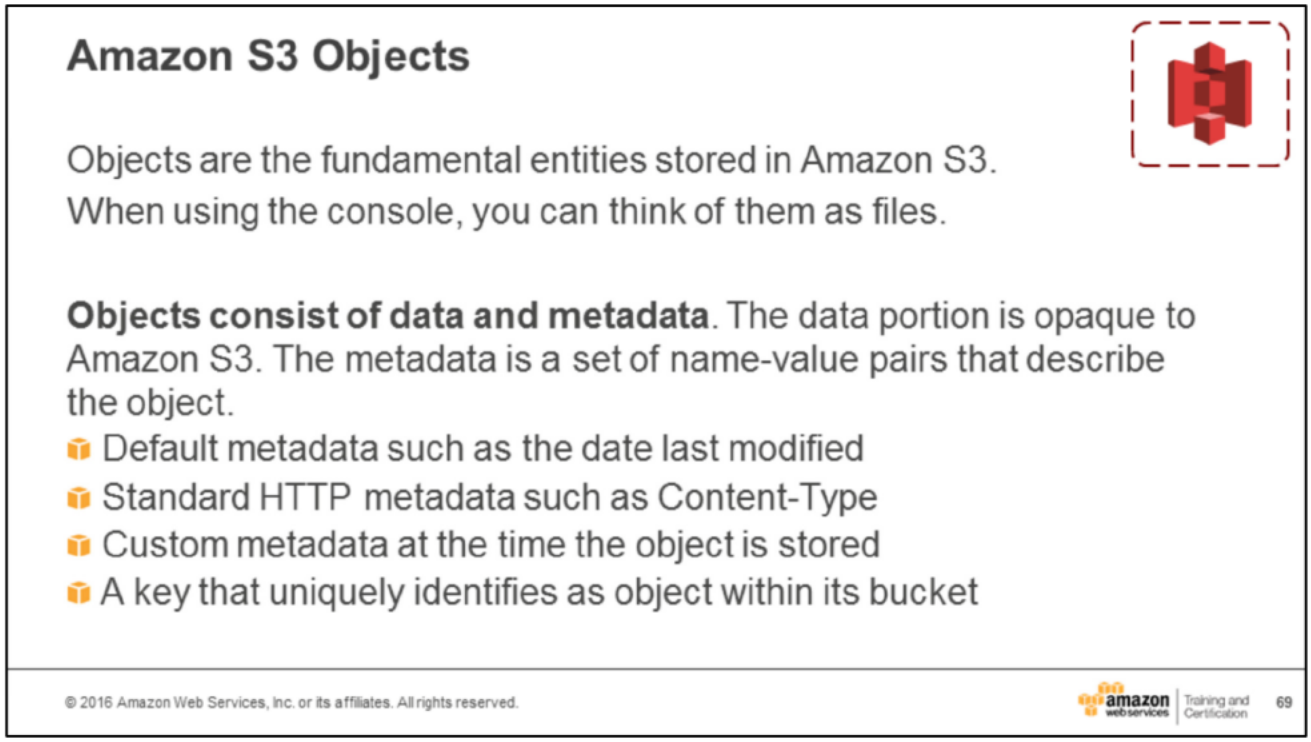
Storage Attached Network (SAN) Architecture: A SAN architecture is composed of servers in a network connected to centralized disk storage.

Amazon S3 Region Considerations

- Amazon S3 creates a bucket in the region you select.
- You can choose a region to:
  - Optimize latency
  - Minimize costs
  - Address regulatory requirements
- Objects stored in a region never leave the region unless you explicitly transfer them to another region.

For more information, see:

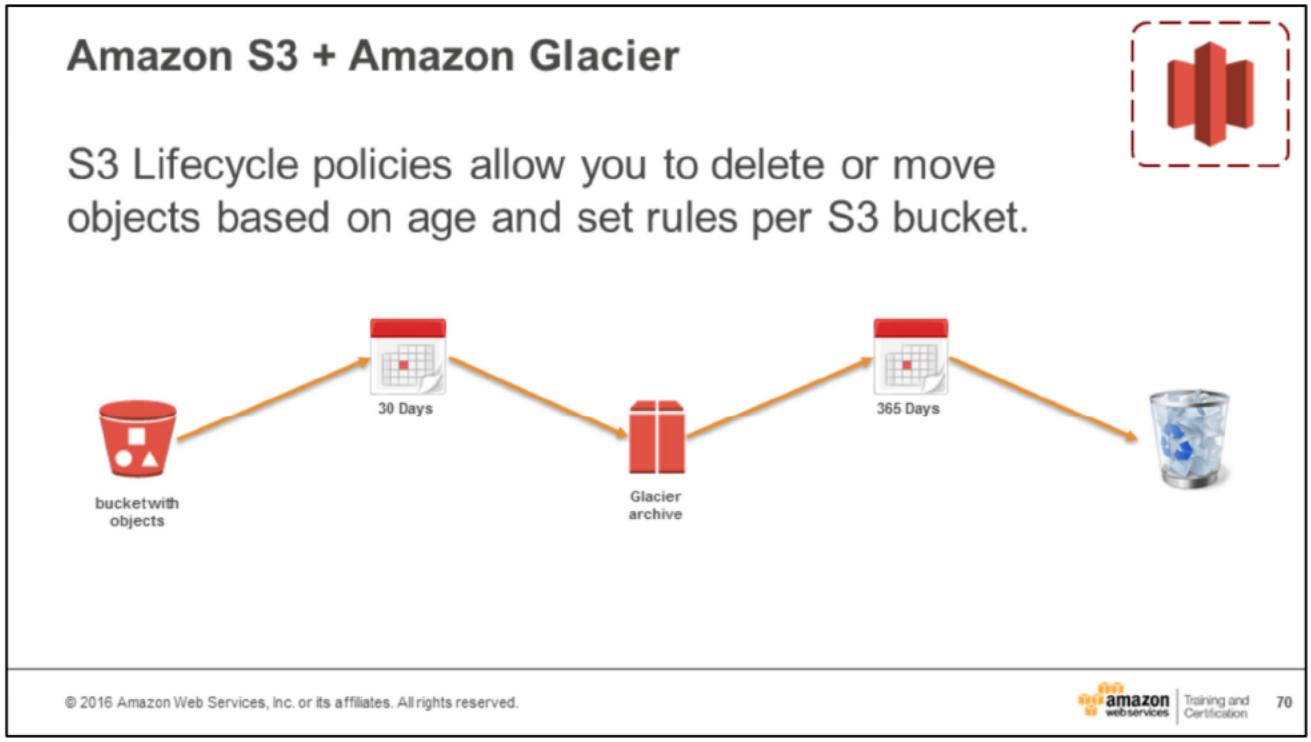
http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region



Objects are the fundamental entities stored in Amazon S3. When using the console, you can think of them as being files. Objects consist of data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. These include some default metadata such as the date last modified, and standard HTTP metadata such as Content-Type. You can also specify custom metadata at the time the object is stored. An object is uniquely identified within a bucket by a key.

For more information, see:

http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-dg.pdf

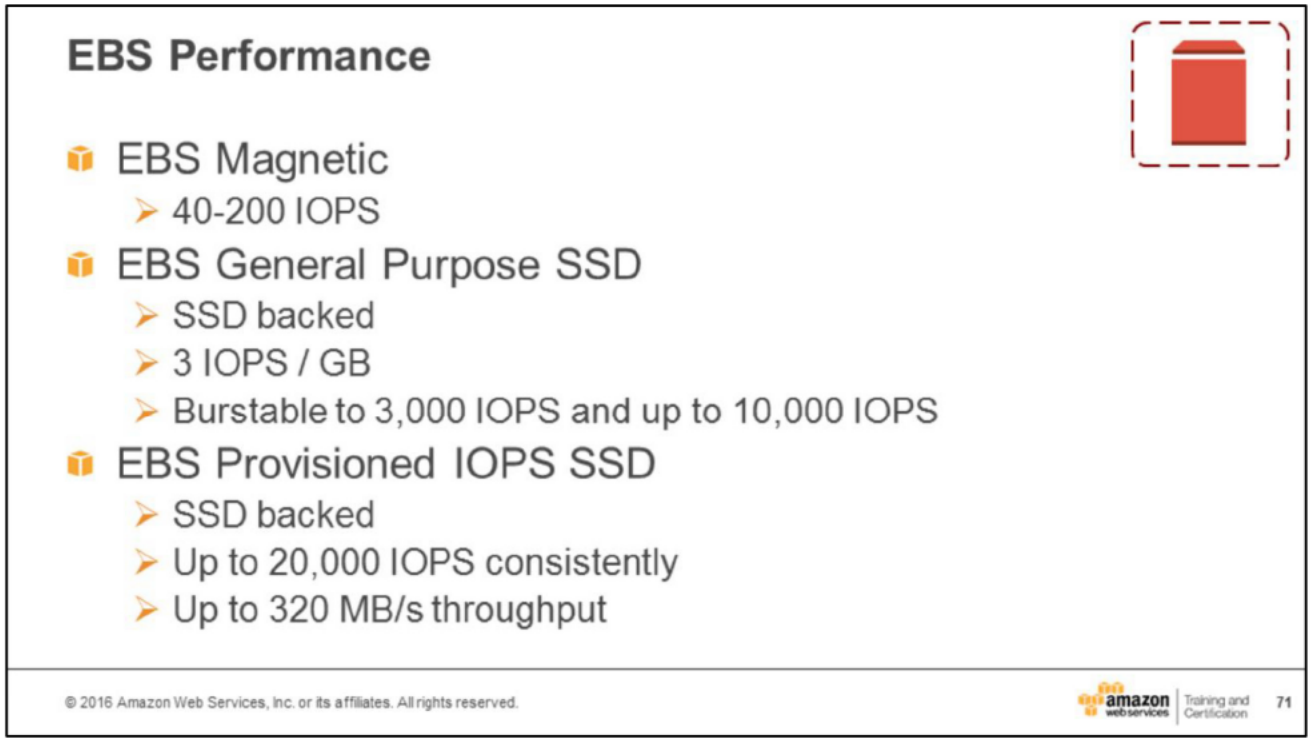


Example:

1. Move object to Amazon Glacier after 30 days
2. Delete object after 365 days

For more information, see:

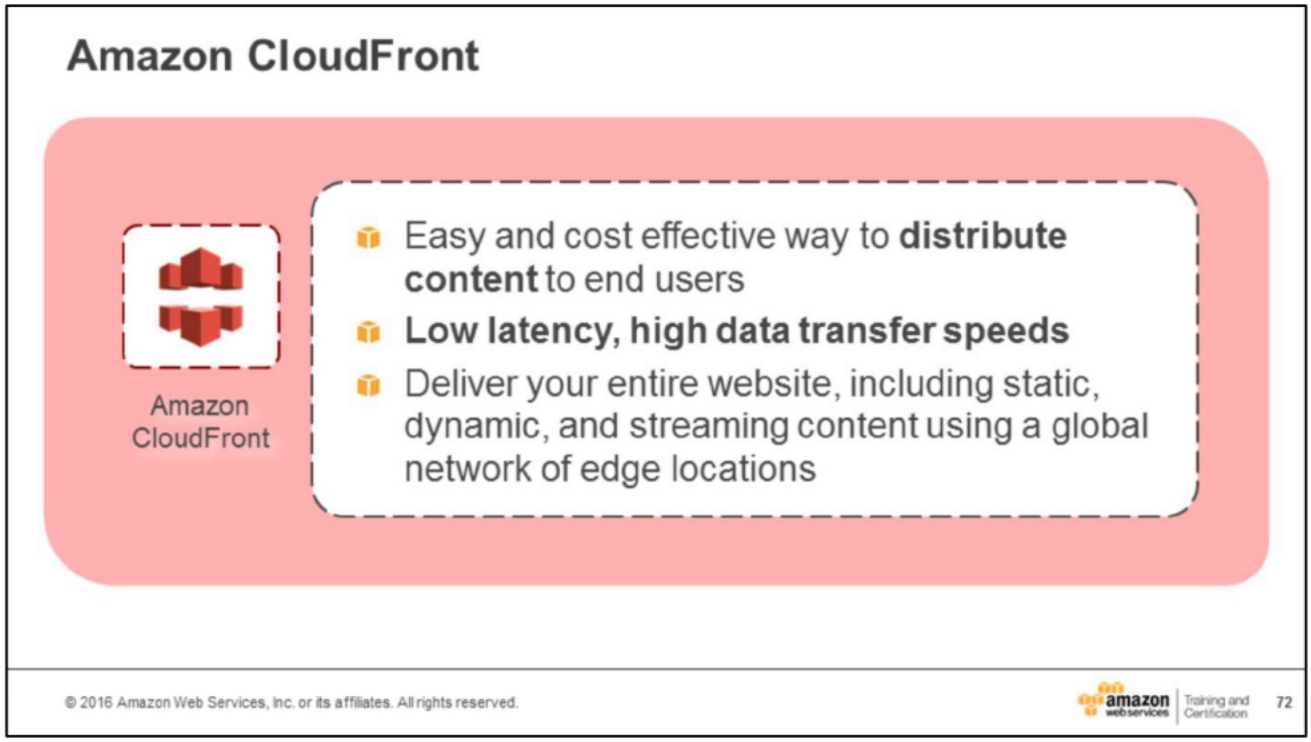
http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html



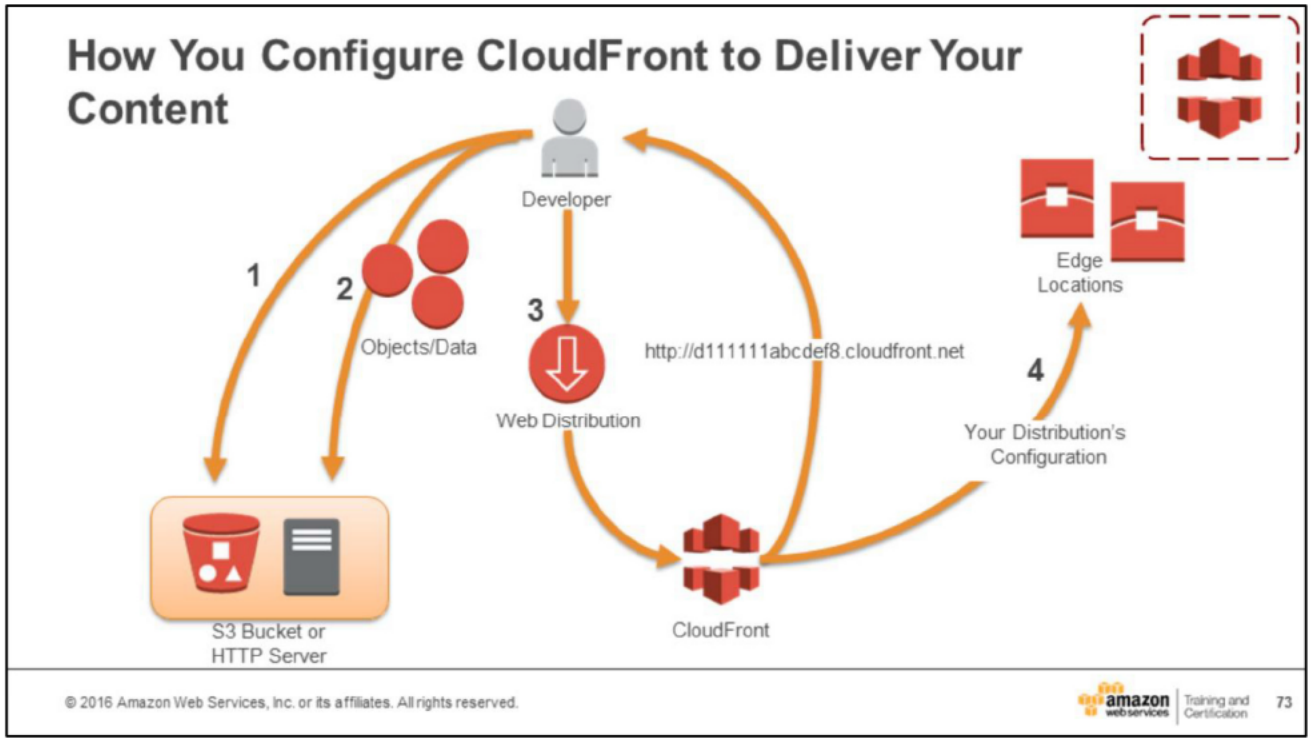
Amazon EBS volume types are shown in the slide.

For more information, see:

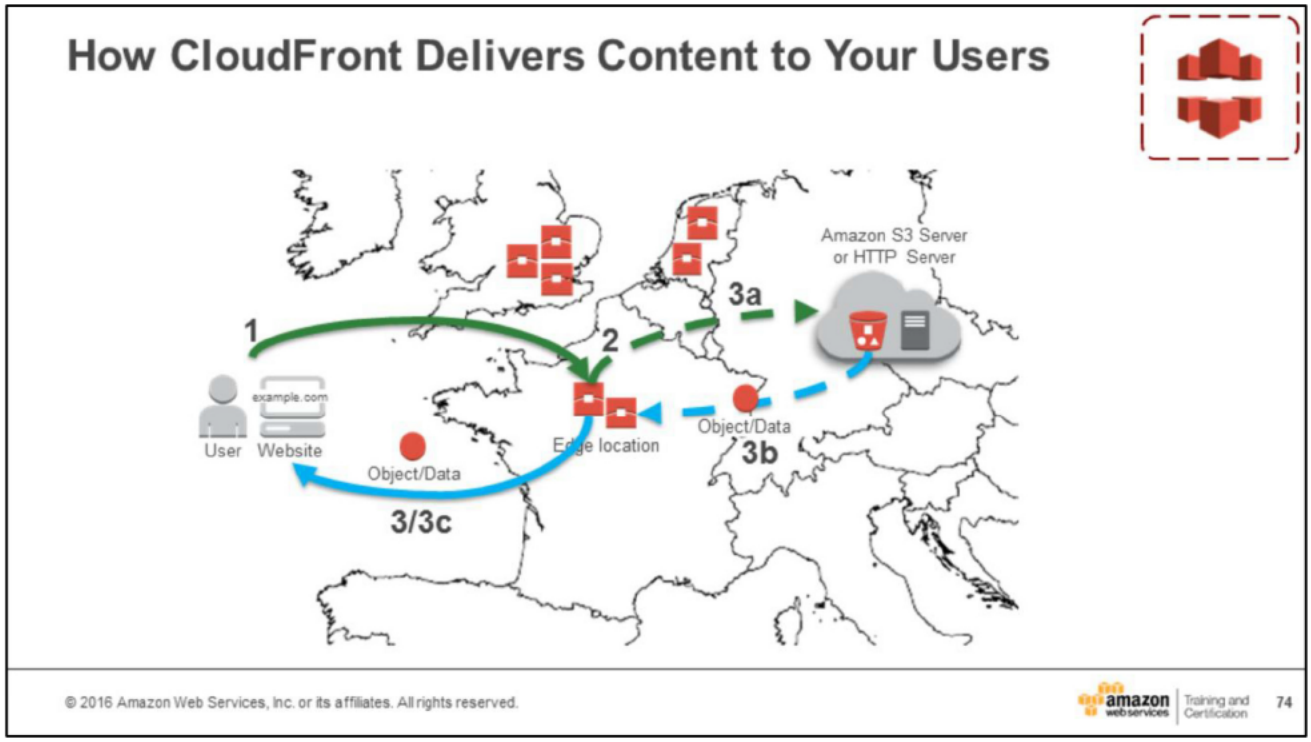
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html



Amazon CloudFront integrates with other Amazon Web Services to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum commitments. Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, for example, .html, .css, .php, and image files, to end users. Amazon CloudFront delivers your content edge locations.



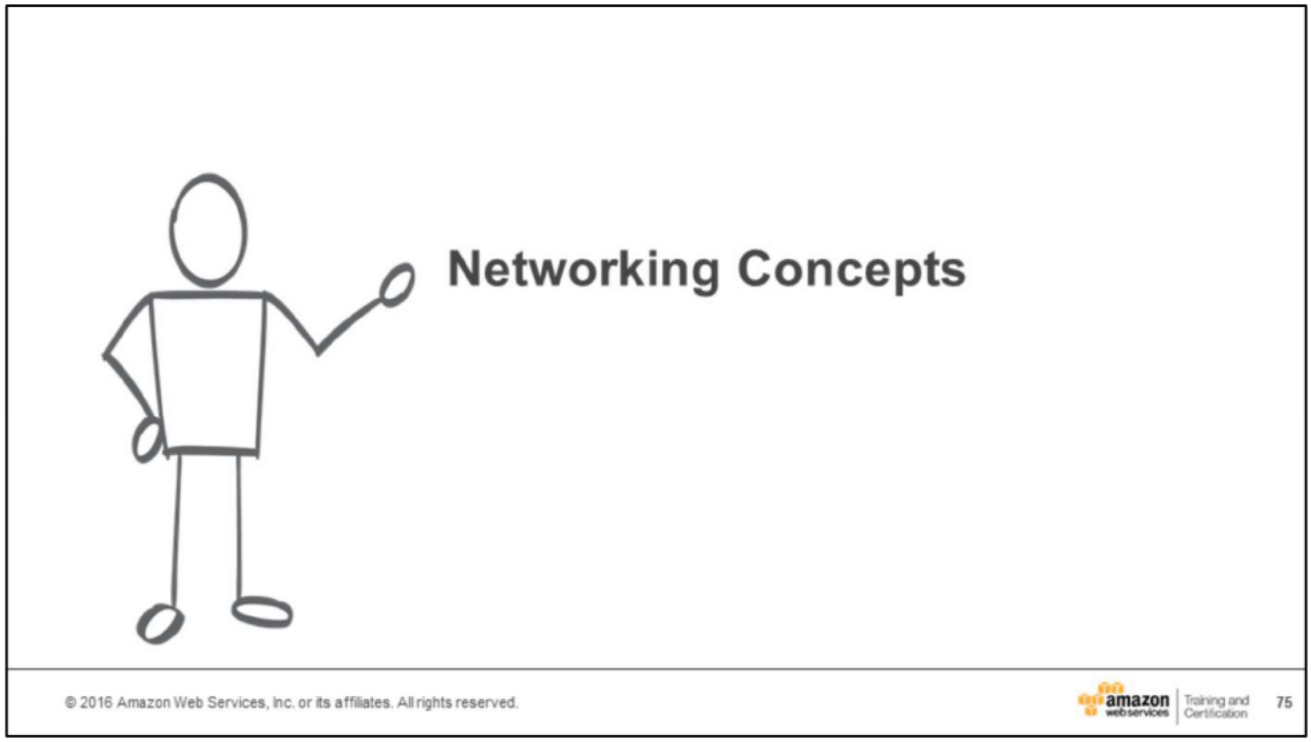
1. Configure your origin servers.
2. Upload files to origin servers.
3. Create an Amazon CloudFront distribution.
4. Amazon CloudFront sends your distribution's configuration to all of its edge locations.
5. As you develop your website or application, you use the domain name that Amazon CloudFront provides for your URLs.
6. Optionally, you can configure your origin server to add expiry headers to the files; the headers indicate how long you want the files to stay in the cache in Amazon CloudFront edge locations.



1. A user accesses your website or application and requests one or more objects, such as an image file.
2. DNS routes the request to the Amazon CloudFront edge location that can best serve the user's request, typically the nearest CloudFront edge location in terms of latency.
3. In the edge location, CloudFront checks its cache for the requested files. If the files are in the cache, CloudFront returns them to the user. If the files are not in the cache, it does the following:
    a. CloudFront compares the request with the specifications in your distribution and forwards the request for the files to the applicable origin server for the corresponding file type—for example, to your Amazon S3 bucket for image files.
    b. The origin servers send the files back to the CloudFront edge location.
    c. As soon as the first byte arrives from the origin, CloudFront begins to forward the files to the user. CloudFront also adds the files to the cache in the edge location for the next time someone requests those files.
4. After an object has been in an edge cache for 24 hours or for the duration specified in your file headers, CloudFront does the following:
    a. CloudFront forwards the next request for the object to your origin to determine whether the edge location has the latest version.

b. If the version in the edge location is the latest, CloudFront delivers it to your user.

c. If the version in the edge location is not the latest, your origin sends the latest version to CloudFront, and CloudFront delivers the object to your user and stores the latest version in the cache at that edge location.



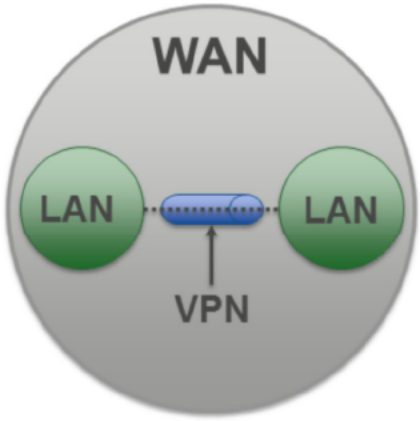
Understand typical networking devices, protocols, and services.

What is a Network?

A network is two or more computers linked to share resources, exchange files, or allow electronic communications.

Network Types:

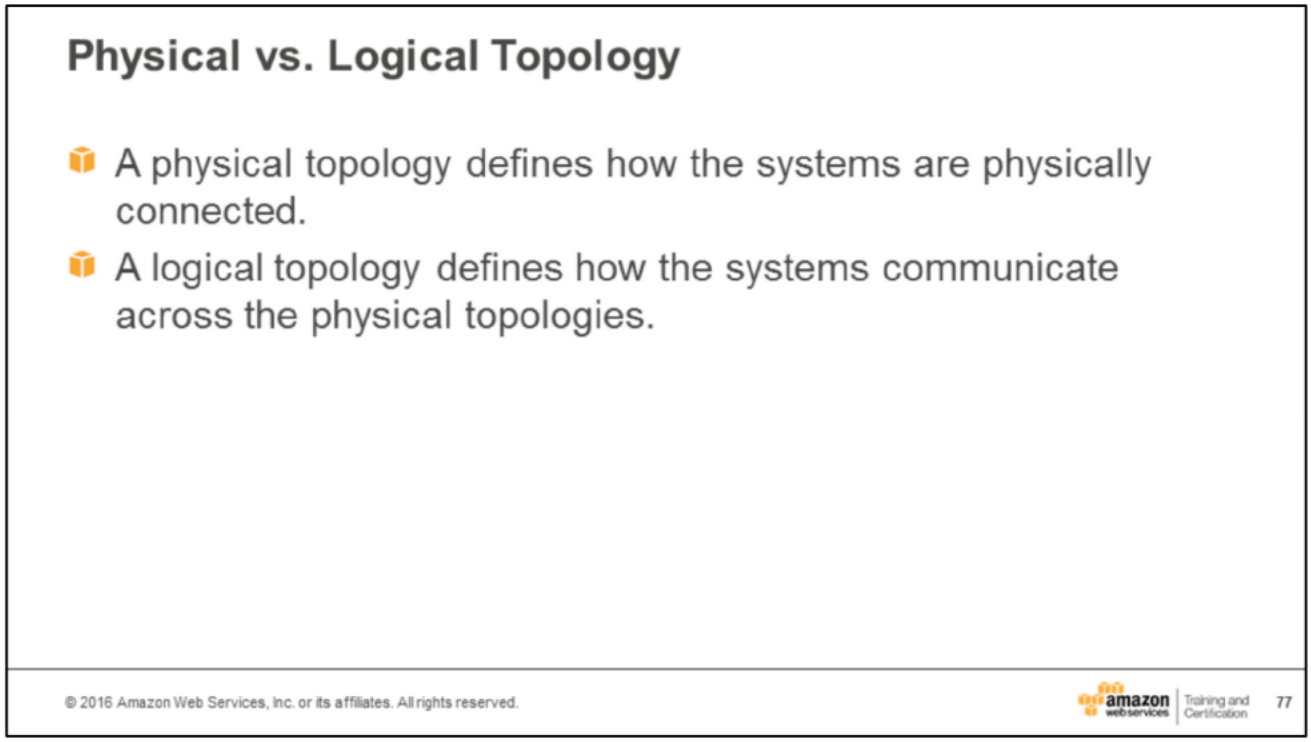
- Local Area Network (LAN)
- Wide Area Network (WAN)
- Virtual Private Network (VPN)



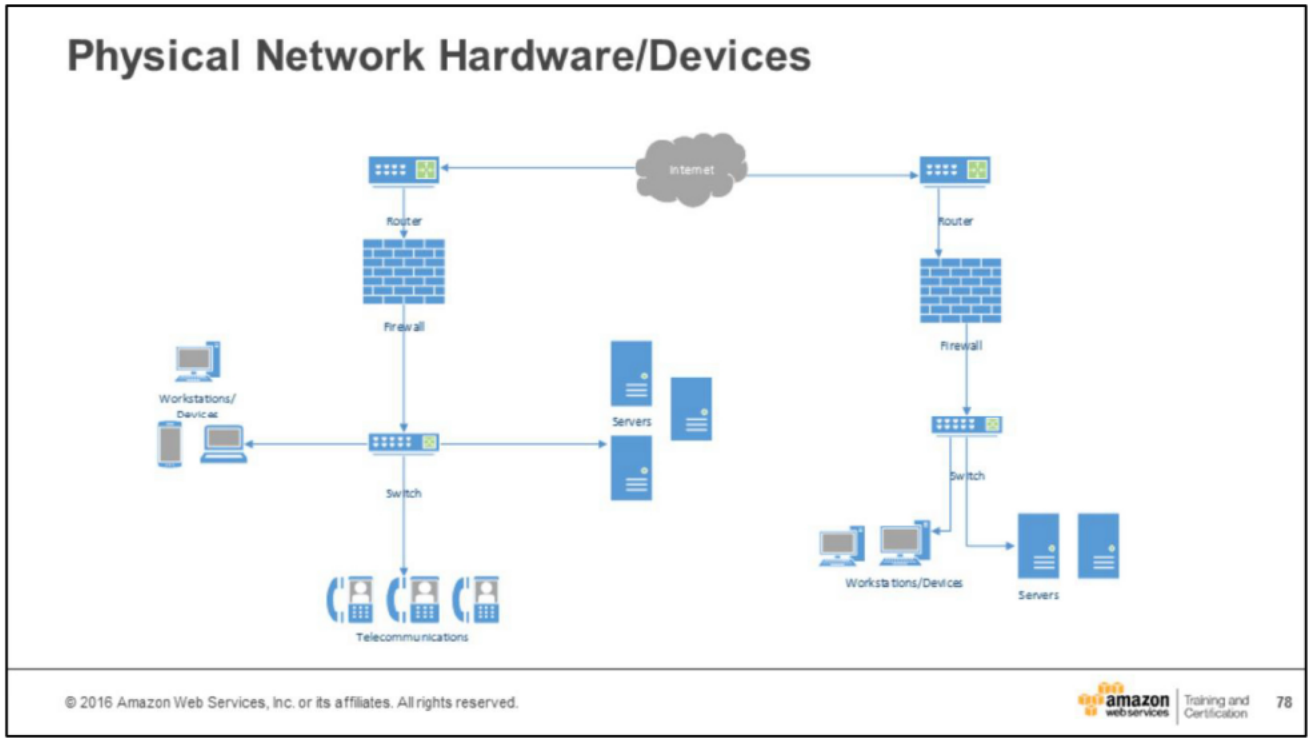
A local area network (LAN) covers a local area such as an office or home.

A wide area network (WAN) covers an area wider than a LAN. A WAN can range from a corporate network to the Internet.

A virtual private network (VPN) allows users to securely connect to a network through the Internet and remotely access network resources.



It is important to understand the difference between a physical and logical topology. You should know how a network is laid out and how the devices communicate on that network to make security decisions to protect your environment.



The slide shows a basic network diagram consisting of various physical network devices.

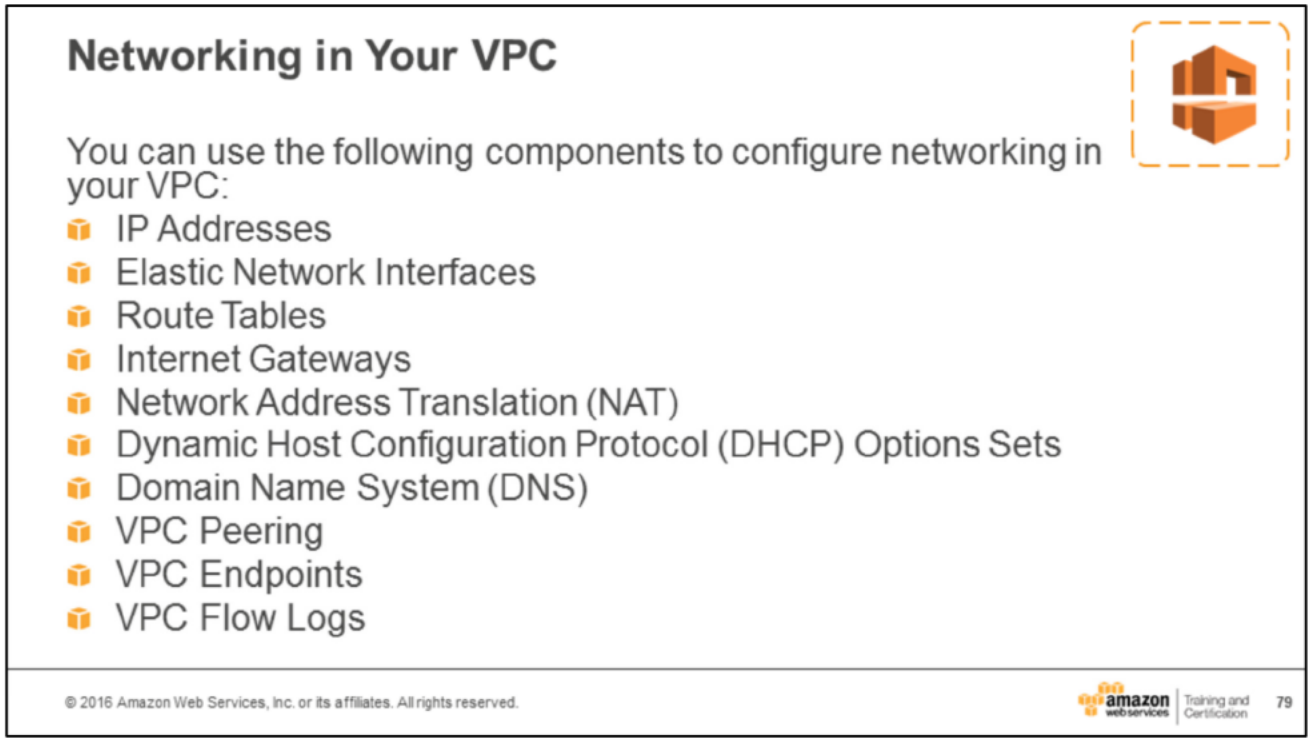
Servers are very fast computers with a large amount of RAM and storage space and one or more fast network interface card(s). Servers are the central repository of data and applications shared by users in a network.

Workstations are computers and devices with a network interface card or wireless adapter to allow quick connections to networks.

Switches are devices that provide a central connection point for cables from workstations, servers, and peripherals.

Routers forward data packets between computer networks.

Firewalls are hardware or software that help secure your network by creating rules to block/allow access.

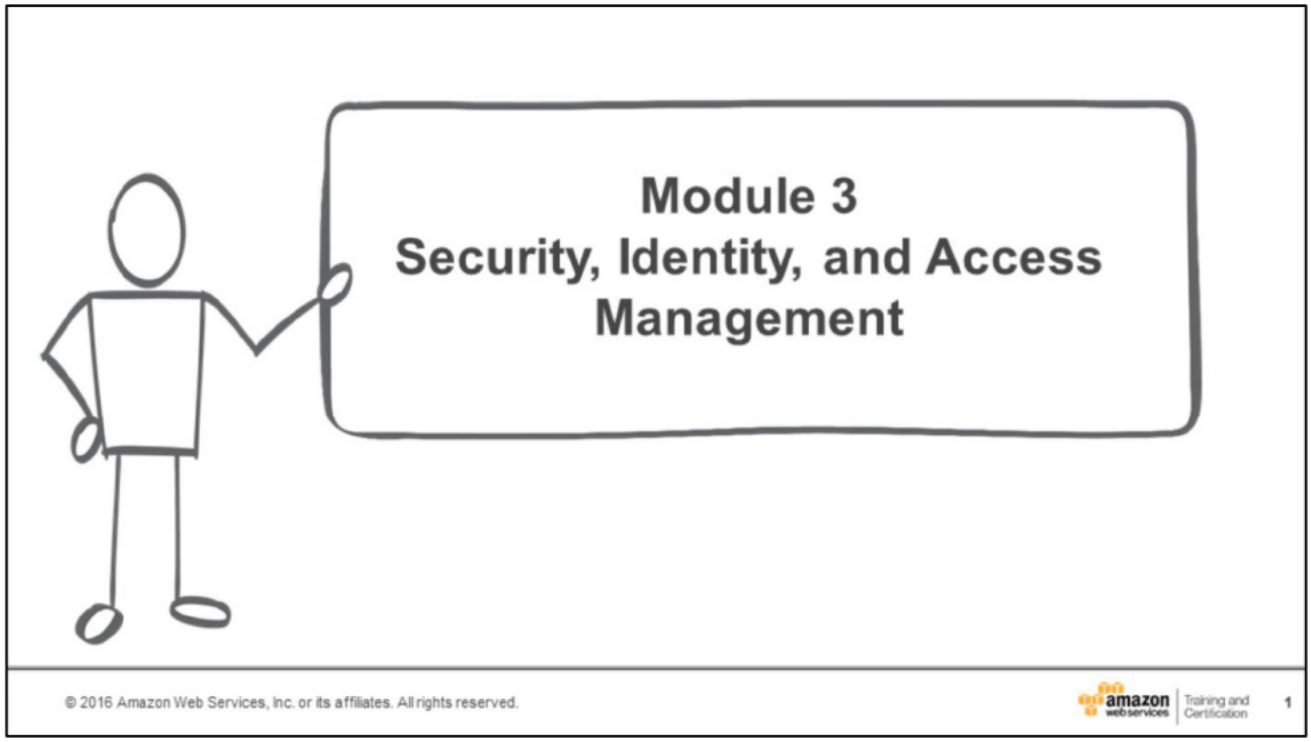


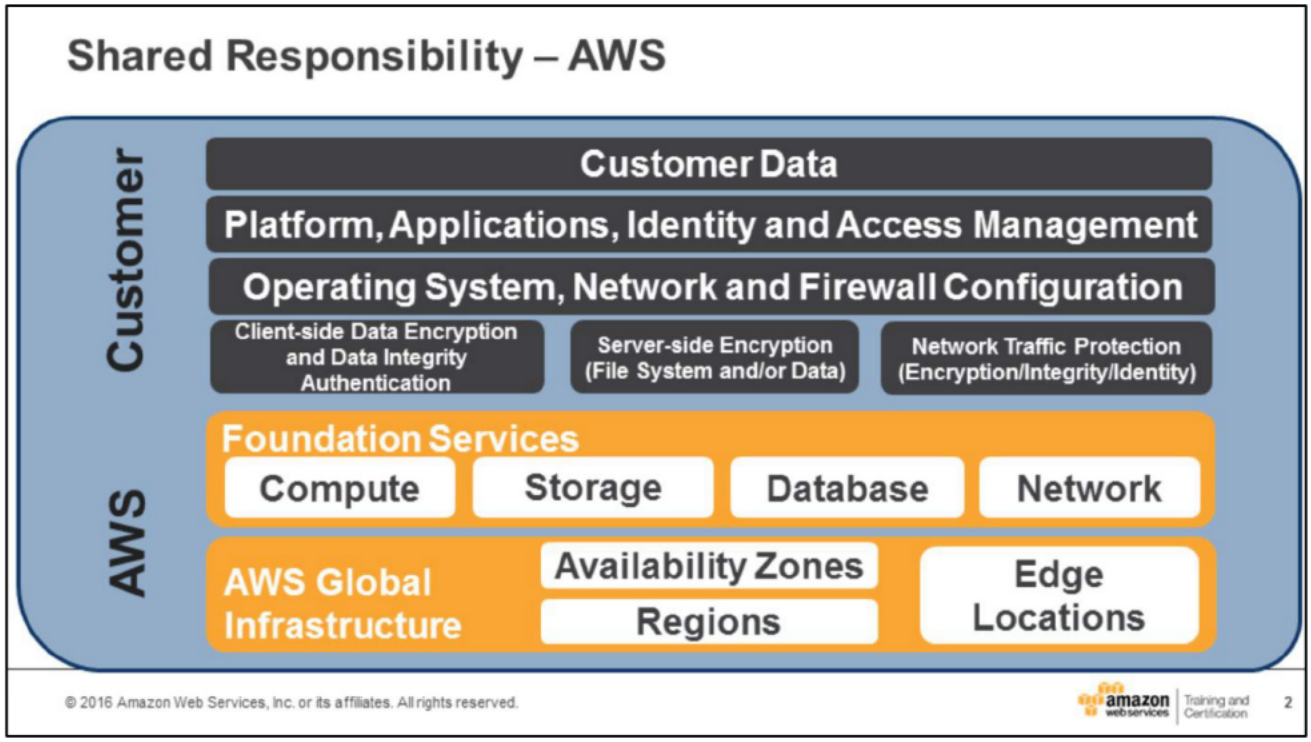
Flow logs capture information about the IP traffic going to and from network interfaces in your VPC.

The diagram shows the layers of security provided by security groups and network ACLs.

For more information, see:

- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Networking.html
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html


Module 3
Security, Identity, and Access Management
© 2016 Amazon Web Services, Inc. or its affiliates. All rights reserved.
amazon web services | Training and Certification
1



When we talk about cloud security, we like to start with a discussion of the Shared Security Responsibility Model. While AWS takes care of provisioning and maintaining the underlying cloud infrastructure, you will still need to perform several security configuration tasks to ensure that you stay safe in the cloud. AWS's responsibility goes from the ground up to the hypervisor. AWS secures the hardware, software, facilities, and networks that run all products and services. Customers are responsible for securely configuring the services they sign up for as well as anything they put on those services.

AWS also performs the following responsibilities:

- Obtaining industry certifications and independent third party attestations
- Publishing information about AWS security and control practices in whitepapers and web site content
- Providing certificates, reports, and other documentation directly to AWS customers under NDA (as required).

The amount of security configuration work you have to do varies, depending on how sensitive your data is and which services you select. For example, AWS services such as Amazon EC2 and Amazon S3 are completely under your control and require you to perform all of the necessary security configuration and management tasks. In the case of Amazon EC2, you are responsible for management of the guest OS (including updates and security patches), any application software or utilities you install on the instances, as well as the configuration of the AWS-provided firewall (called a security group) on each instance.

When you use any of AWS's managed services like Amazon RDS, Amazon RedShift, or Amazon WorkDocs, you don't have to worry about launching and maintaining instances or patching the guest OS or applications—AWS handles that for you. For these managed services, basic security configuration tasks like data backups, database replication, and firewall configuration happen automatically.

However, there are certain security features—such as IAM user accounts and credentials, SSL for data transmissions, and user activity logging—that you should configure no matter which AWS service you use.

AWS Support provides a highly personalized level of service for customers seeking technical help.

For more information, see:

- https://aws.amazon.com/premiumsupport/
- http://d0.awsstatic.com/whitepapers/Security/Intro_to_AWS_Security.pdf
- https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf

Physical Security

- 24/7 trained security staff
- AWS data centers in nondescript and undisclosed facilities
- Two-factor authentication for authorized staff
- Authorization for data center access

amazon web services | Training and Certification 3

One of the main security responsibilities of AWS is the physical security of the data centers that house the AWS cloud infrastructure. Amazon has many years of experience designing, constructing, and operating large-scale data centers.

The physical security measures that protect these data centers are some of the most comprehensive in the industry and include: 24/7 trained security guards; locations in nondescript, undisclosed facilities; two-factor authentication for ingress; authorization for data center access only for an approved, specific need; and continuous monitoring, logging, and auditing of physical access controls.

For more information, see:

Security Center - http://aws.amazon.com/security/

Hardware, Software, and Network

- Automated change-control process
- Bastion servers that record all access attempts
- Firewall and other boundary devices
- AWS monitoring tools

The hardware and software that supports AWS cloud services has been architected to be not only highly available and redundant, but also extremely secure. All changes to AWS hardware and software are managed through a centralized, automated change control process, and all access to hardware or software must be authorized.

Privileged access to software and systems requires SSH logon and is allowed only through bastion servers that record all access attempts. AWS network devices, including firewall and other boundary devices, monitor and control communications at the external boundary of the network and at key internal boundaries.

AWS monitoring tools are designed to detect unusual or unauthorized activities and conditions at ingress and egress communication points. These tools monitor server and network usage, port scanning activities, application usage, and unauthorized intrusion attempts. AWS security monitoring tools help identify several types of denial of service (DoS) attacks, including distributed, flooding, and software/logic attacks.

For more information, see:

Security Center - http://aws.amazon.com/security/

AWS has successfully completed multiple audits, attestations, and certifications. AWS publishes a Service Organization Controls SOC 1 report, published under both the SSAE 16 and the ISAE 3402 professional standards, as SOC 2-Security and SOC 3 Report.

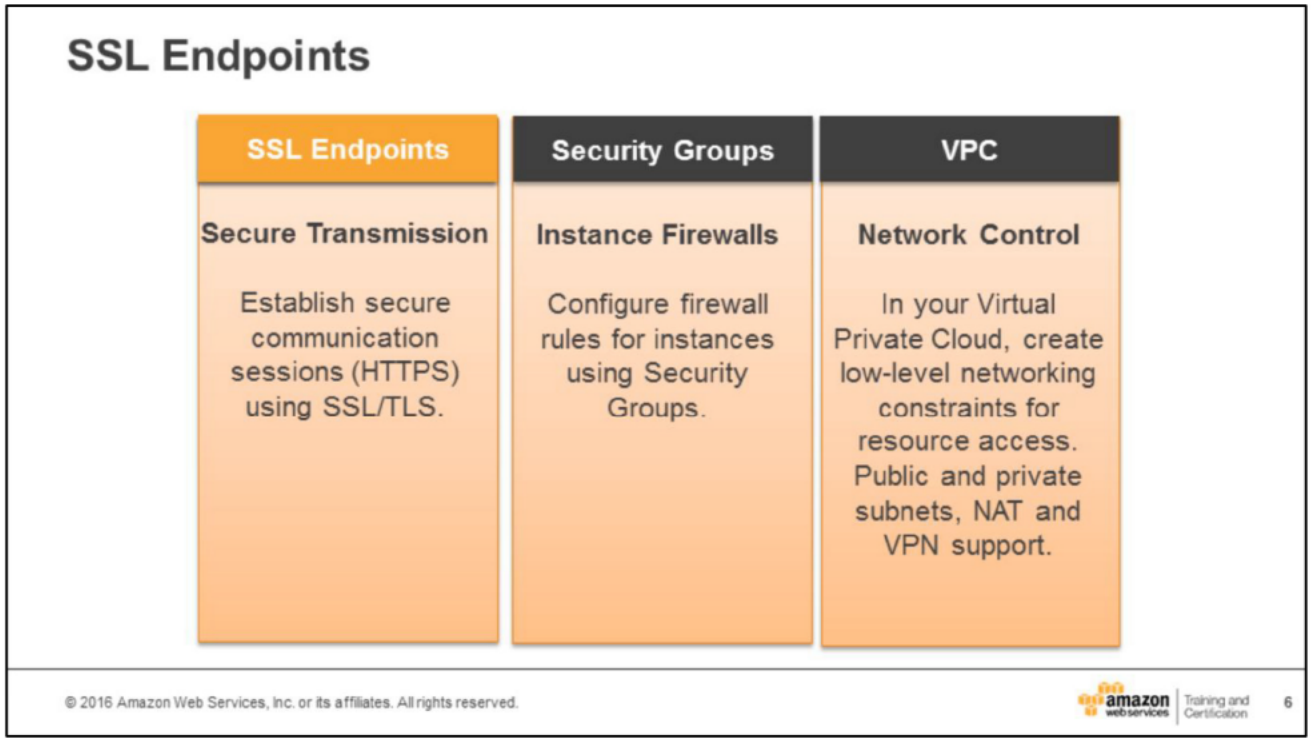
In addition, AWS has achieved ISO 9001, ISO 27001, ISO 27017 and ISO 27018 certifications, has been successfully validated as a Level 1 service provider under the Payment Card Industry (PCI) Data Security Standard (DSS), and currently offers HIPAA Business Associate Agreements to covered entities and their business associates subject to HIPAA.

In the realm of public sector certifications, AWS has achieved FedRAMP compliance, has received authorization from the U.S. General Services Administration to operate at the FISMA Moderate level, and is also the platform for applications with Authorities to Operate (ATOs) under the Defense Information Assurance Certification and Accreditation Program (DIACAP).

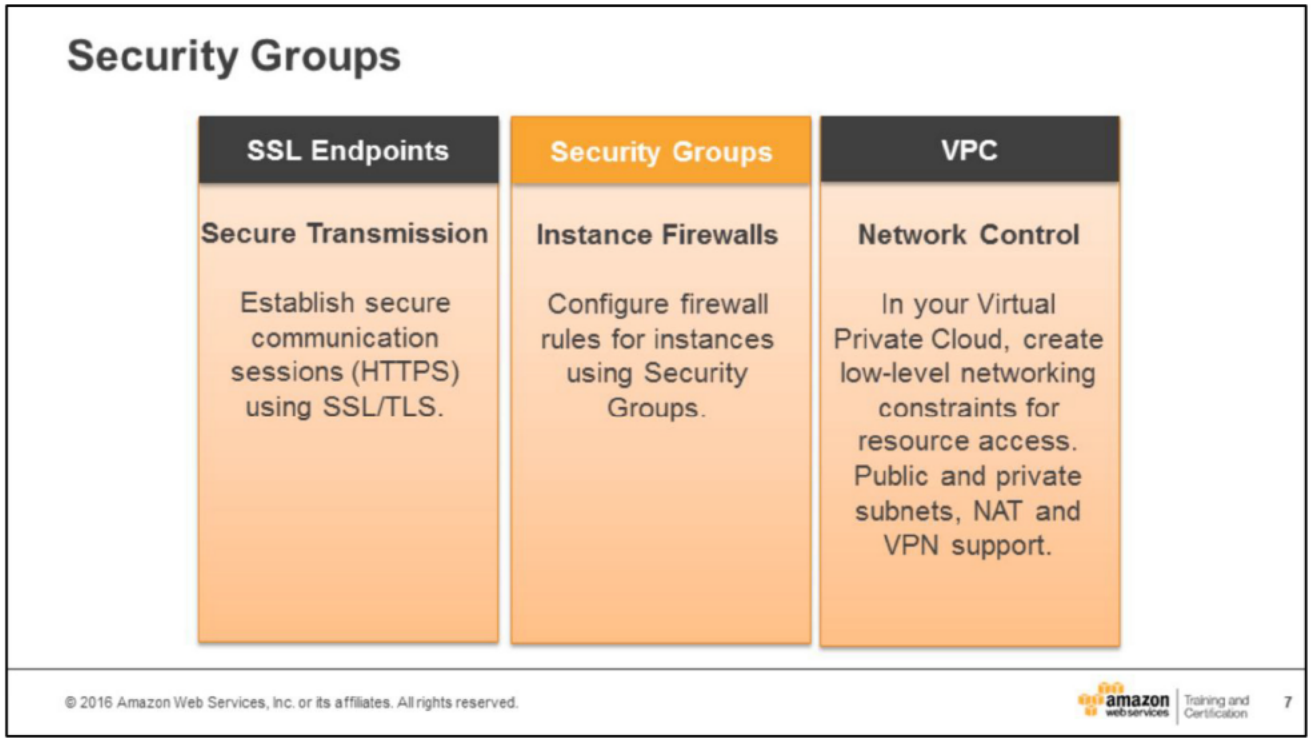
NIST, FIPS 140-2, CJIS, DoD SRG Levels 2 and 4 are some of the other certifications AWS has received.

For more information, see:

AWS Compliance - http://aws.amazon.com/compliance/

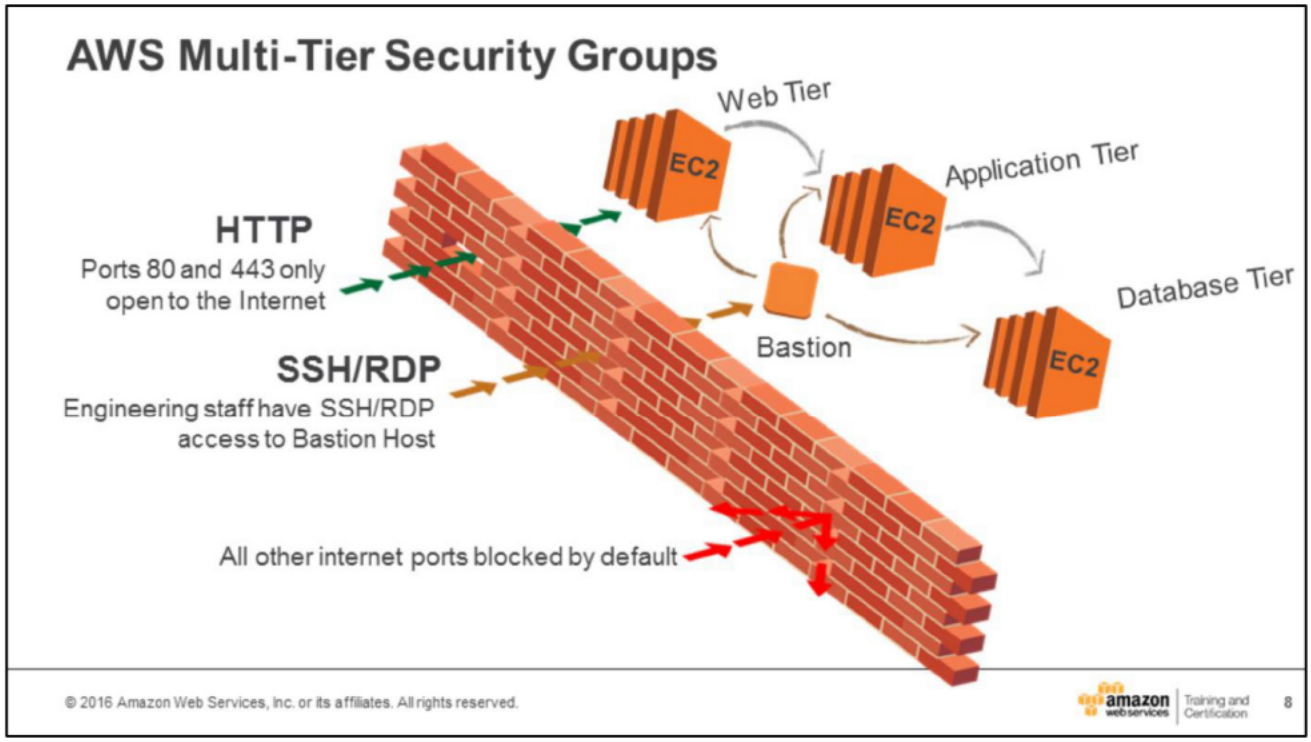


AWS provides customer access points, also called API endpoints, that allow HTTPS access so that you can establish secure communication sessions with your AWS services including SSL and TLS. SSL encrypts the transmission, protecting each request or the response from being viewed in transit.



AWS also provides security groups, which act like built-in firewalls for your virtual servers. You can control how accessible your instances are by configuring security group rules--from totally public to completely private, or somewhere in between. And when your instances reside within a Virtual Private Cloud (VPC) subnet, you can control egress as well as ingress traffic.

Security Groups can also be used by AWS services such as Amazon RDS, Amazon Redshift, Amazon EMR and Amazon ElastiCache.

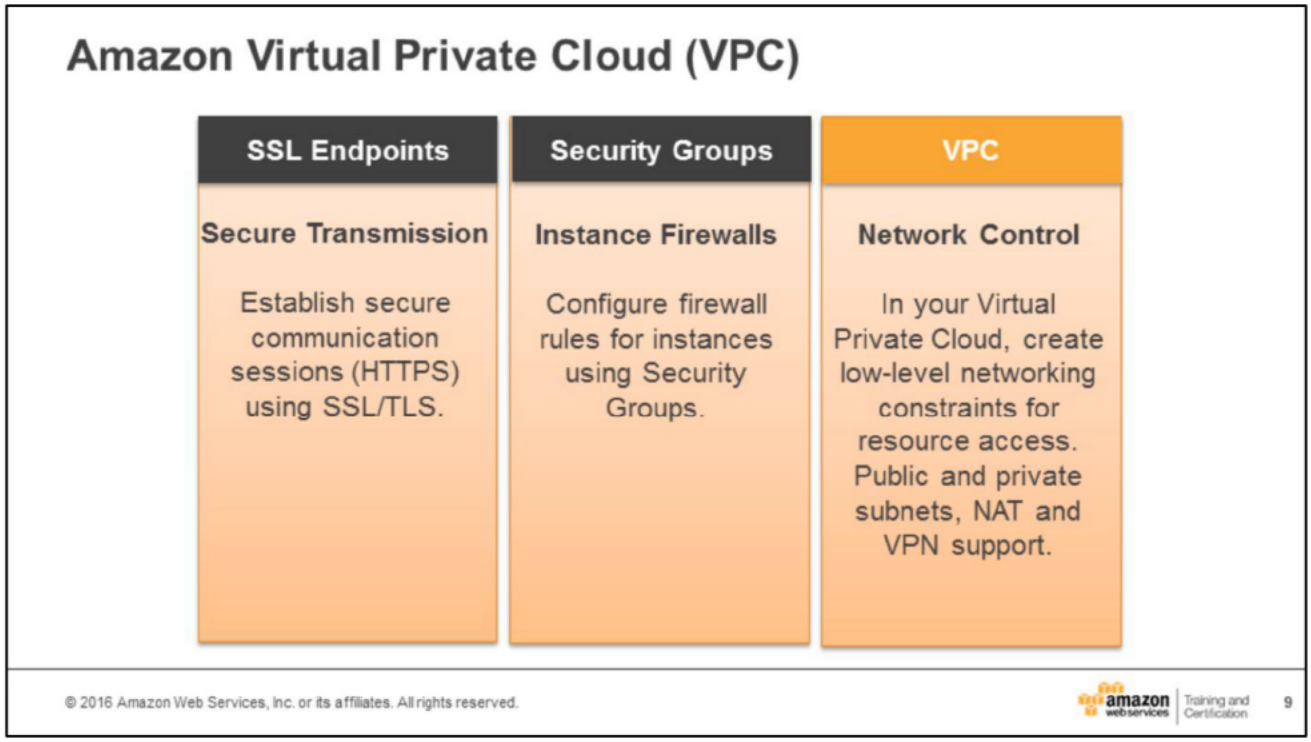


You can set up security group rules for your EC2 instances to create a traditional multi-tiered web architecture:

The web tier security group can accept traffic on port 80/443 from anywhere on the Internet if you select source 0.0.0.0/0. Alternatively, it might make more sense to only accept traffic from a load balancer so that individual clients cannot overload a single server and the load balancer can perform its job.

Similarly, the app tier can only accept traffic from the web tier, and the DB tier can only accept traffic from the app tier.

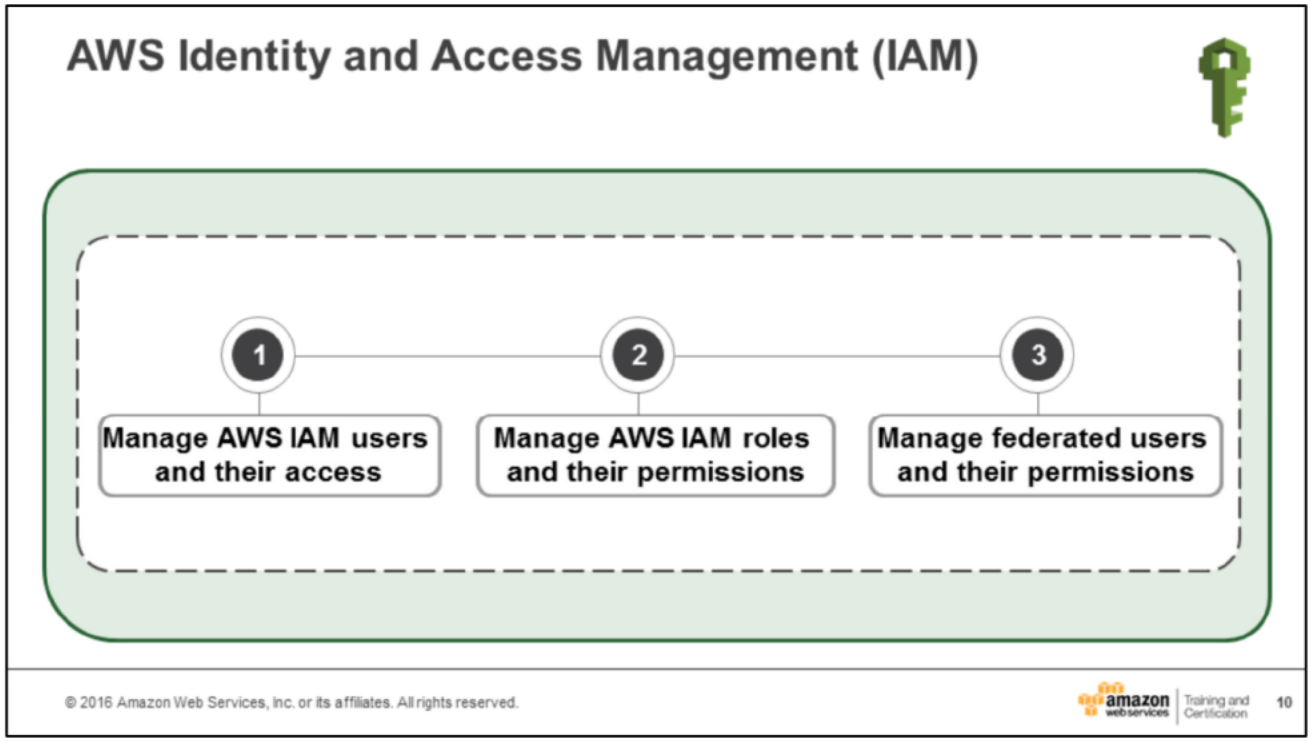
Lastly, we have also added a set of rules to allow remote administration over SSH port 22. We have restricted remote access by funneling all traffic through the app tier and allowing access only from a specific IP. After you use SSH to access an app tier server, you can then connect to machines on the web and DB security groups.



The Amazon Virtual Private Cloud (VPC) service allows you to add another layer of network security to your instances by creating private subnets and even adding an IPsec VPN tunnel between your network and your VPC. Amazon VPC allows you to define your own network topology, including definitions for subnets, network access control lists, Internet gateways, routing tables, and virtual private gateways. The subnets that you create can be defined as either private or public.

For more information, see:

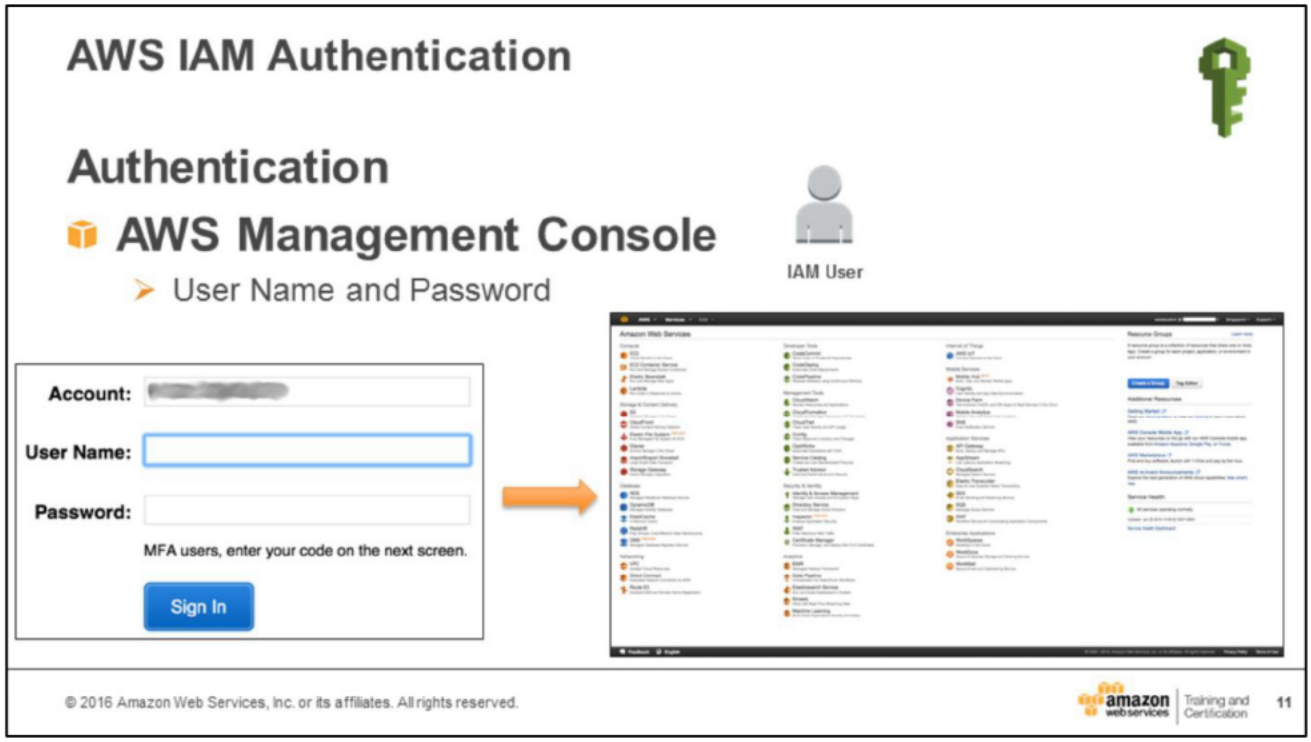
Amazon VPC: http://aws.amazon.com/vpc/



Using IAM you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources. You can use existing corporate identities to grant secure access to AWS resources, such as Amazon S3 buckets, without creating any new AWS identities.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-services-that-work-with-iam.html



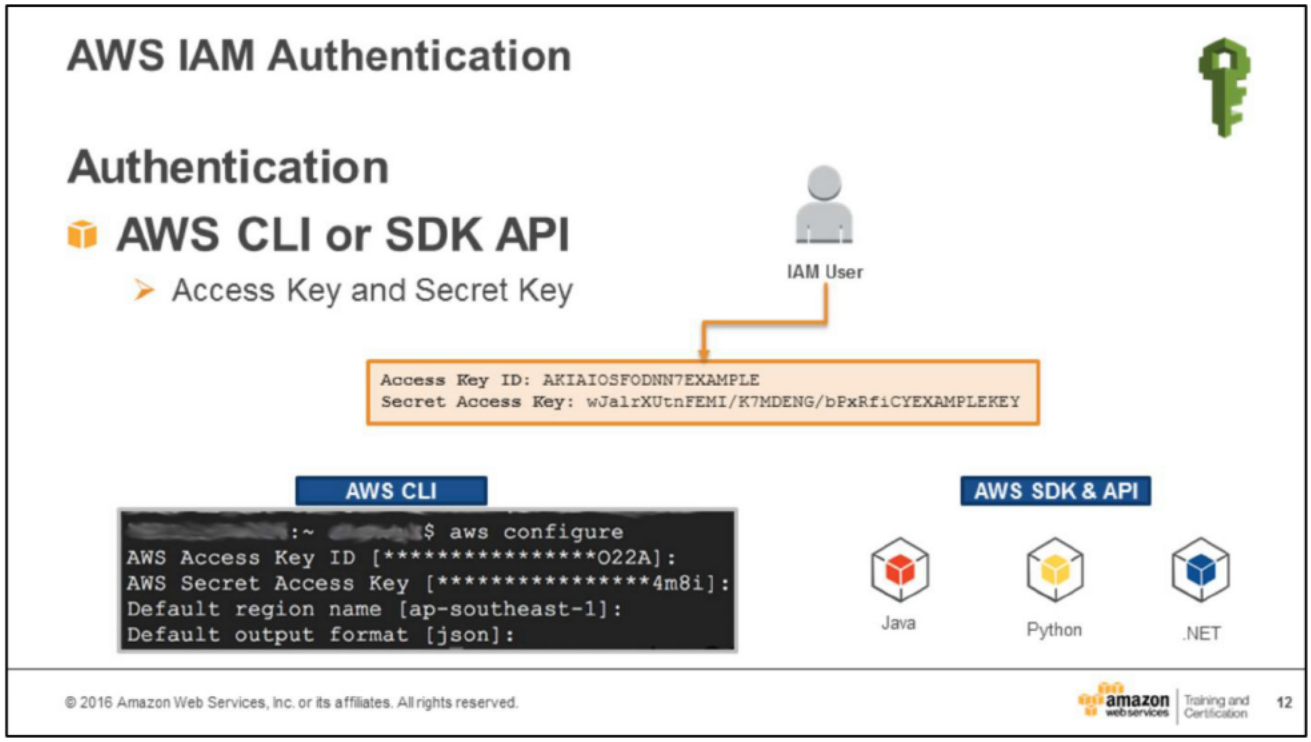
AWS services and resources can be accessed using the AWS Management Console, AWS CLI or through SDKs and APIs from a wide range of supported platforms. Users and systems have to be authenticated before they can access AWS services and resources.

The AWS Management Console provides a web-based way to administer AWS services. If you're the account owner, you can sign in to the console directly using the Root Account. It is, however, advisable to create individual IAM users for each user and login using individual credentials.

IAM is a complimentary service.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/console.html



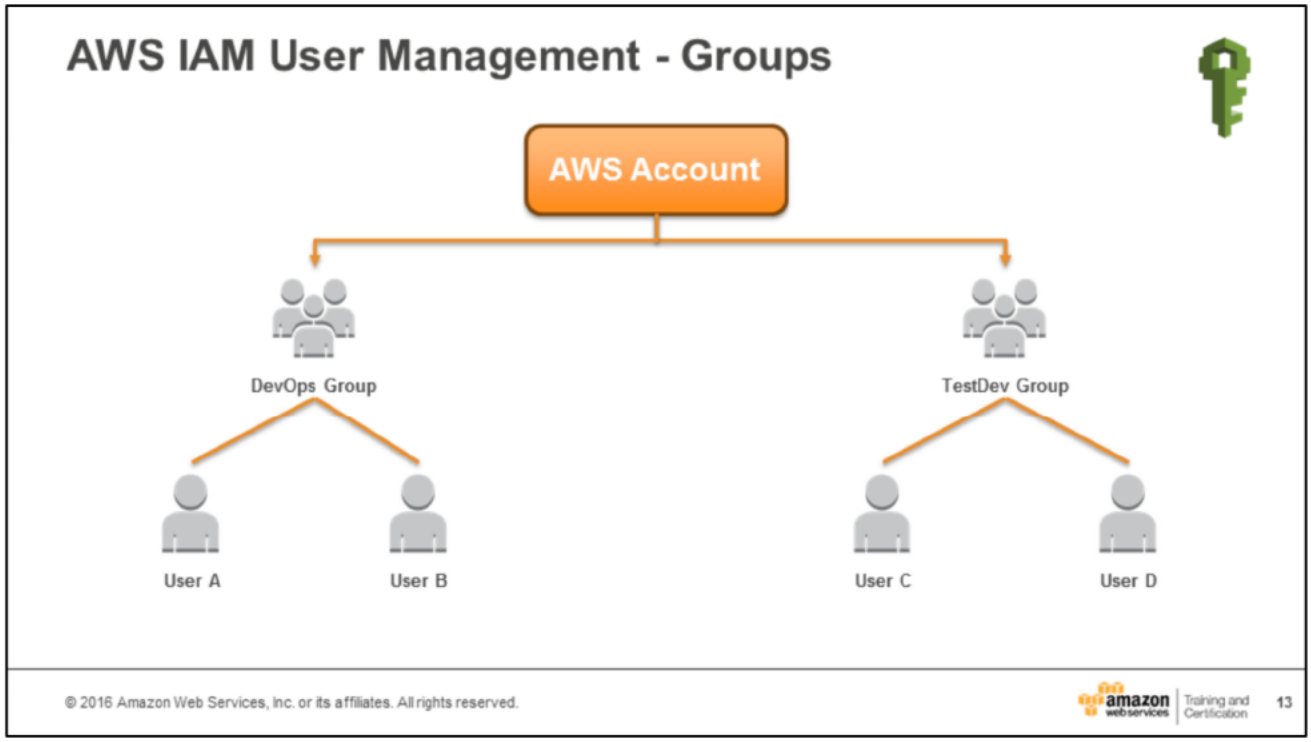
The AWS Command Line Interface is a unified tool to manage your AWS services. With AWS CLI, you can control multiple AWS services from the command line and automate them through scripts.

AWS CLI is supported on Windows, Linux, OS X, and Unix platforms.

AWS offers support for a wide variety of programming platforms including .NET, Java, Python etc.

For more information, see:

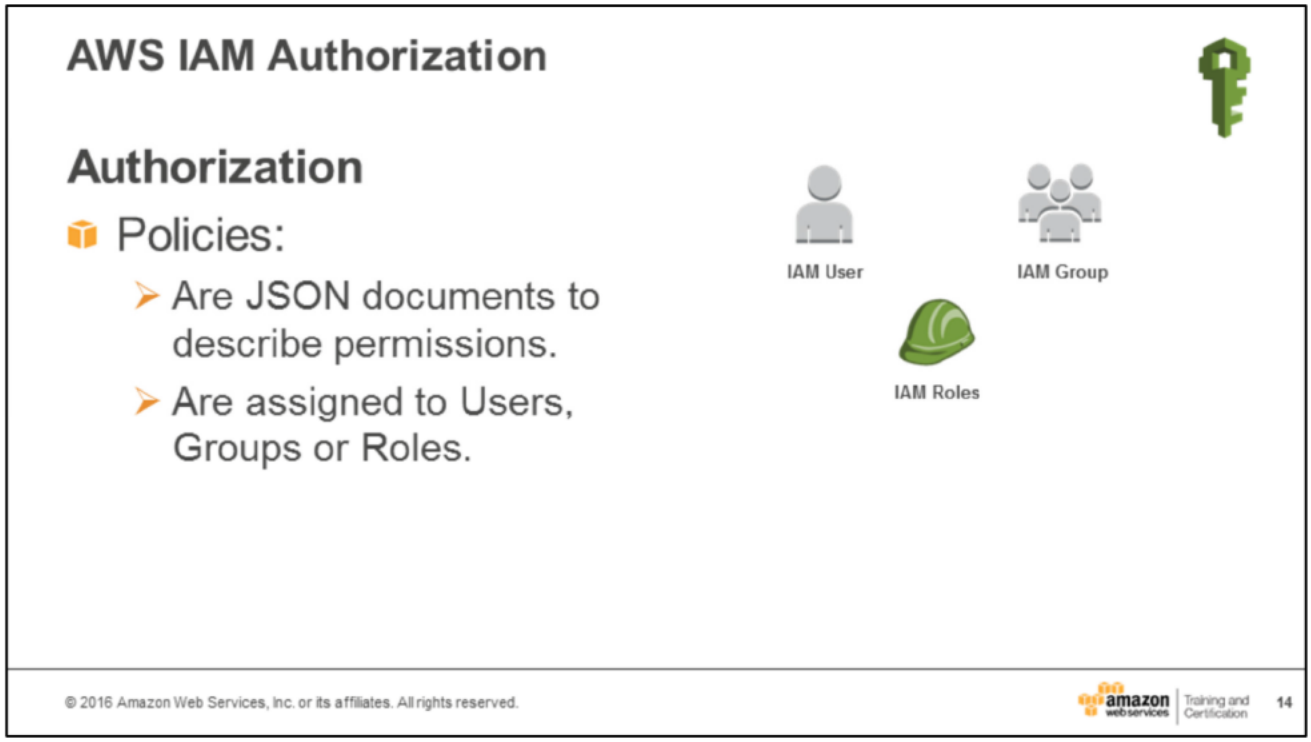
http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html



As the number of users managing your AWS environment increases, it is helpful to manage permissions for multiple IAM users using IAM groups.

For more information, see:

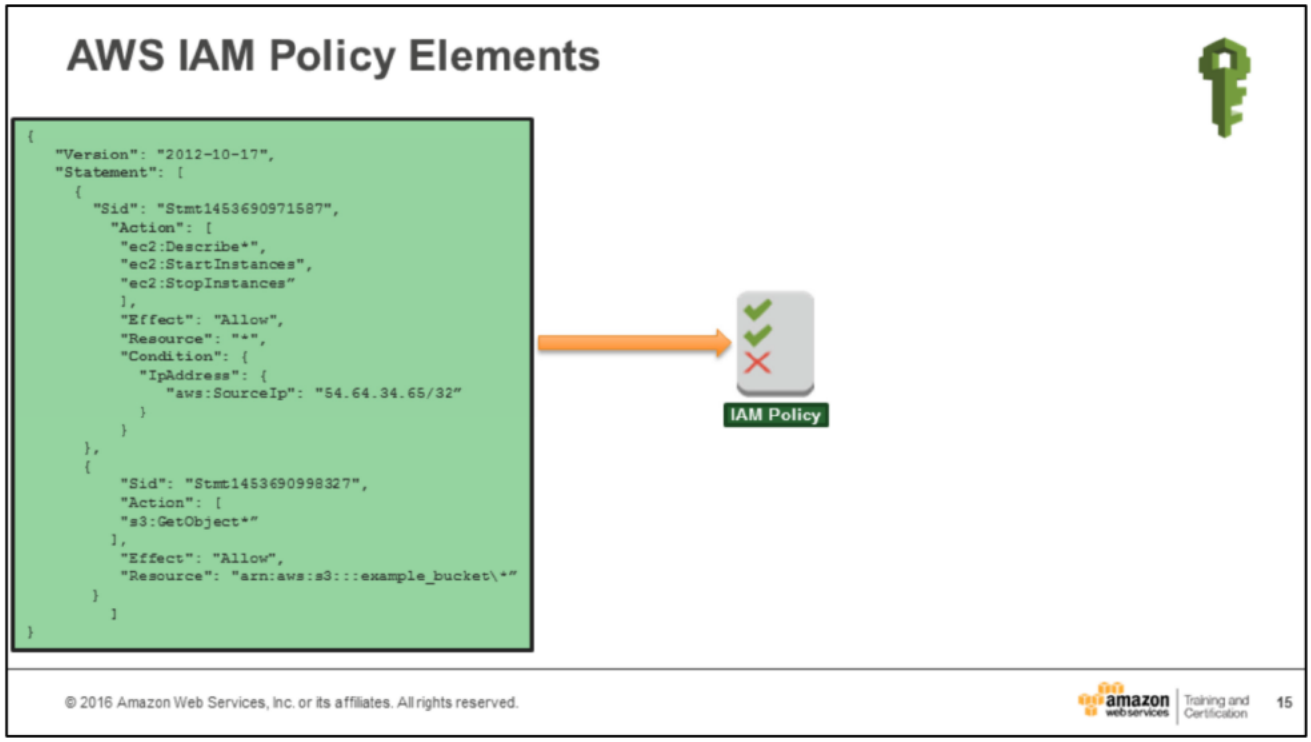
IAM Groups: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html



After a user or system has been authenticated, they have to be authorized to access AWS services. To assign permissions to a user, group, role, or resource, you create a policy, which is a document that explicitly lists permissions.

An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have any credentials (password or access keys) associated with it. Instead, if a user is assigned to a role, access keys are created dynamically and provided to the user.

Policies and Roles will be covered in more detail in the following slides.



Policies are documents that are created using JavaScript Object Notation (JSON). A policy consists of one or more statements, each of which describes one set of permissions.

An IAM policy may consist of :

- **Version**
- **Id**
- **Statement**
- **Sid**
- **Effect**: Defines what the effect will be when the user requests access—either allow or deny. Because the default is that resources are denied to users, you typically specify that you will allow users access to resource.
- **Principal**
- **NotPrincipal**
- **Actions**: Defines what actions you want to allow. Each AWS service has its own set of actions. Any actions that you do not explicitly allow are denied.
- **NotAction**
- **Resources**: Defines which resources you allow the action on. Users cannot access any resources that you have not explicitly granted permissions to.

- **NotResource**
- **Condition**
- **Supported Data Types**

AWS Policy Generator: You can use the AWS Policy Generator to generate policies at ease.

AWS Policy Validator: Policy Validator automatically examines your existing IAM access control policies to ensure that they comply with the IAM policy grammar.

AWS Policy Simulator: The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify. The simulator uses the same policy evaluation engine that is used during real requests to AWS services.

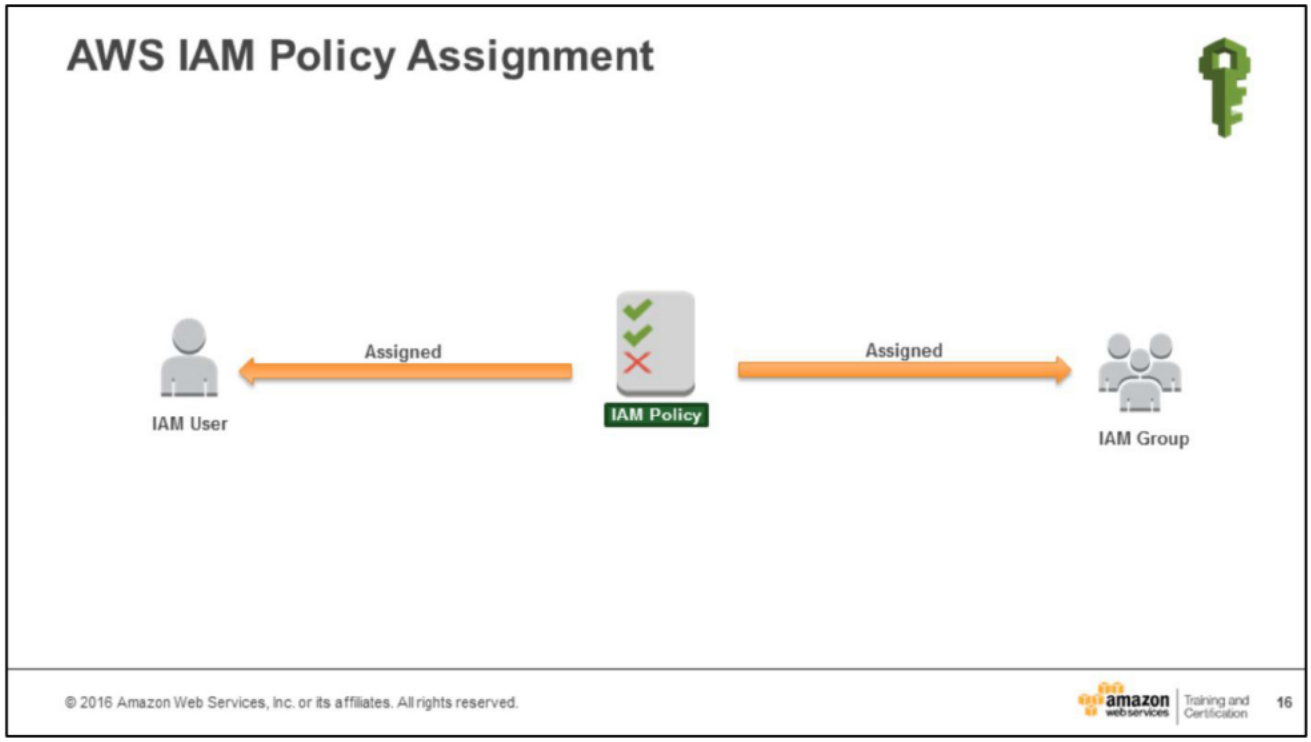
Managed policies – Are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies apply only to identities (users, groups, and roles) - not resources. You can use two types of managed policies:

- **AWS managed policies** – Managed policies that are created and managed by AWS. If you are new to using policies, it is recommended that you start by using AWS managed policies.
- **Customer managed policies** – Managed policies that you create and manage in your AWS account. Using customer managed policies, you have more precise control over your policies than when using AWS managed policies.

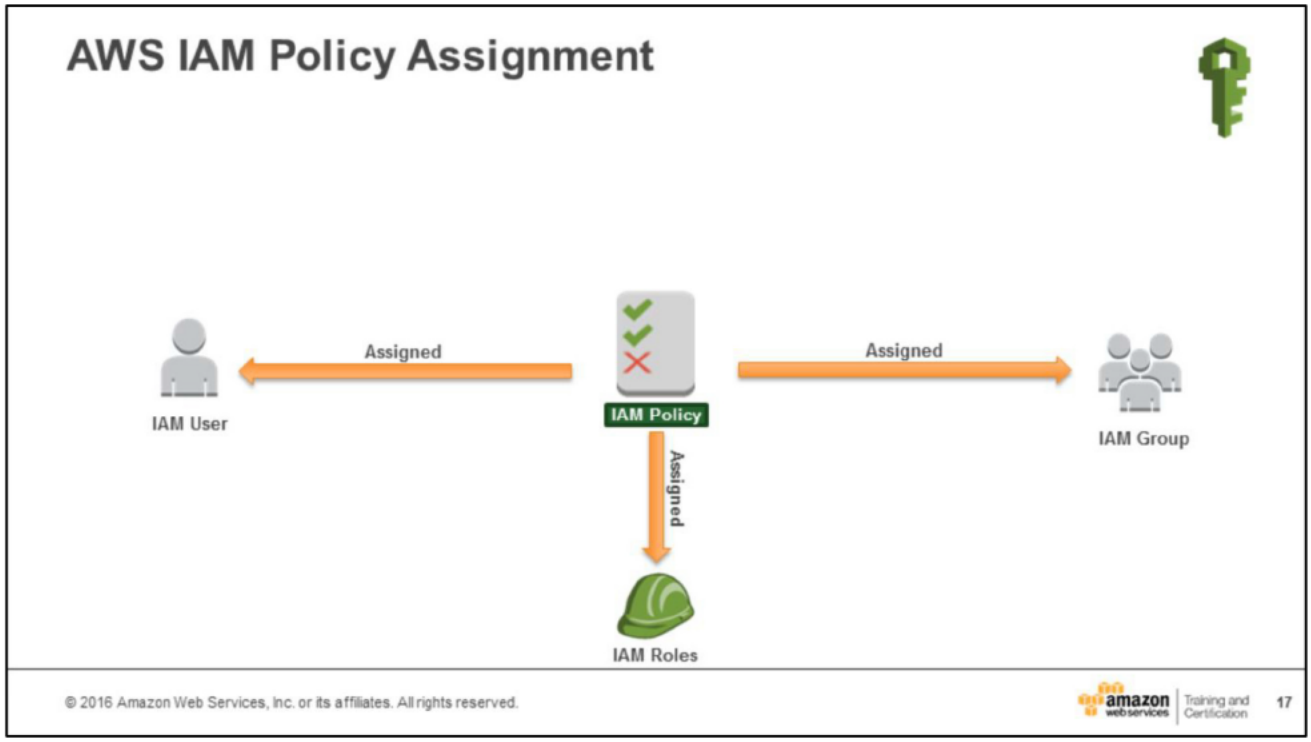
Inline policies – Policies that you create and manage, and that are embedded directly into a single user, group, or role.

For more information, see:

- AWS Policy Generator - http://awspolicygen.s3.amazonaws.com/policygen.html
- Access the Policy Simulator - https://policysim.aws.amazon.com/home/index.jsp
- Overview of IAM Policies - http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html
- IAM Policy Elements Reference - http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html



IAM Policies are assigned to IAM users and Groups. These users are bound by the permissions defined in the IAM Policy.

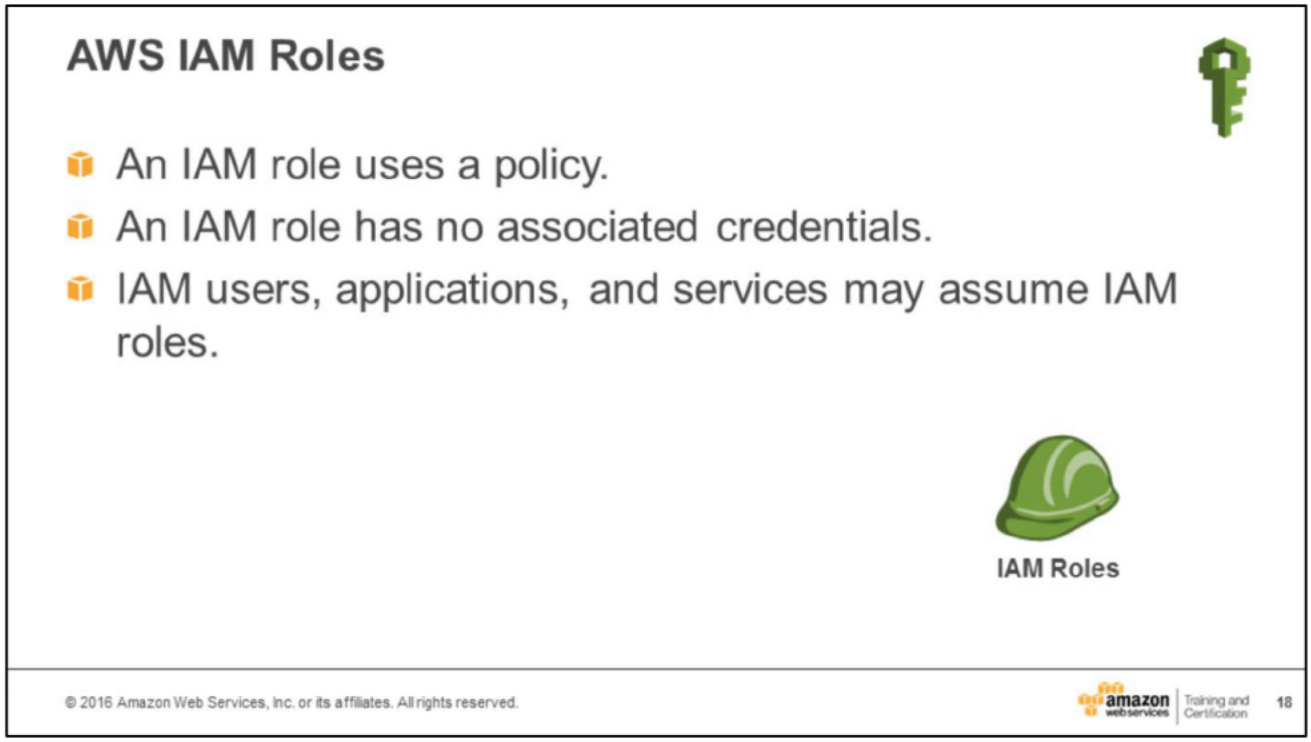


IAM Policies may also be assigned to an IAM Role.

An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have any credentials (password or access keys) associated with it. Instead, if a user is assigned to a role, access keys are created dynamically and provided to the user.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html

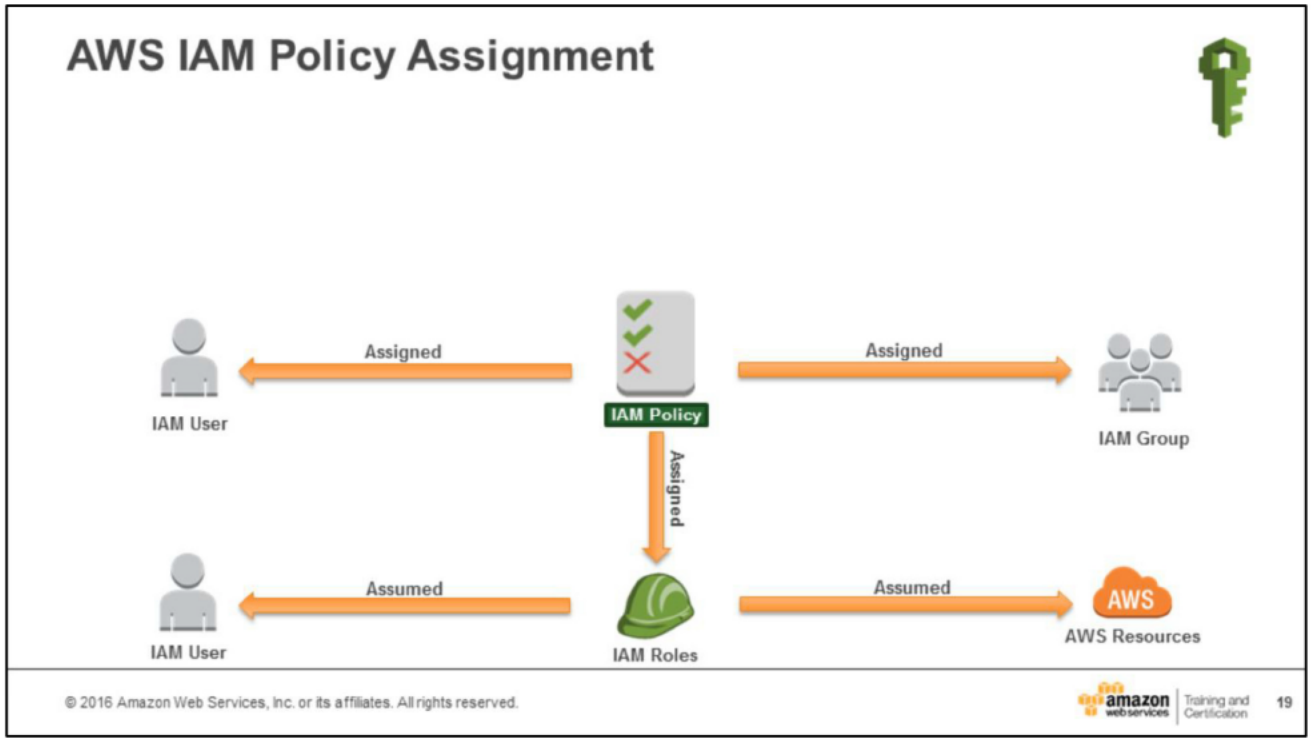


IAM Policies may also be assigned to an IAM role.

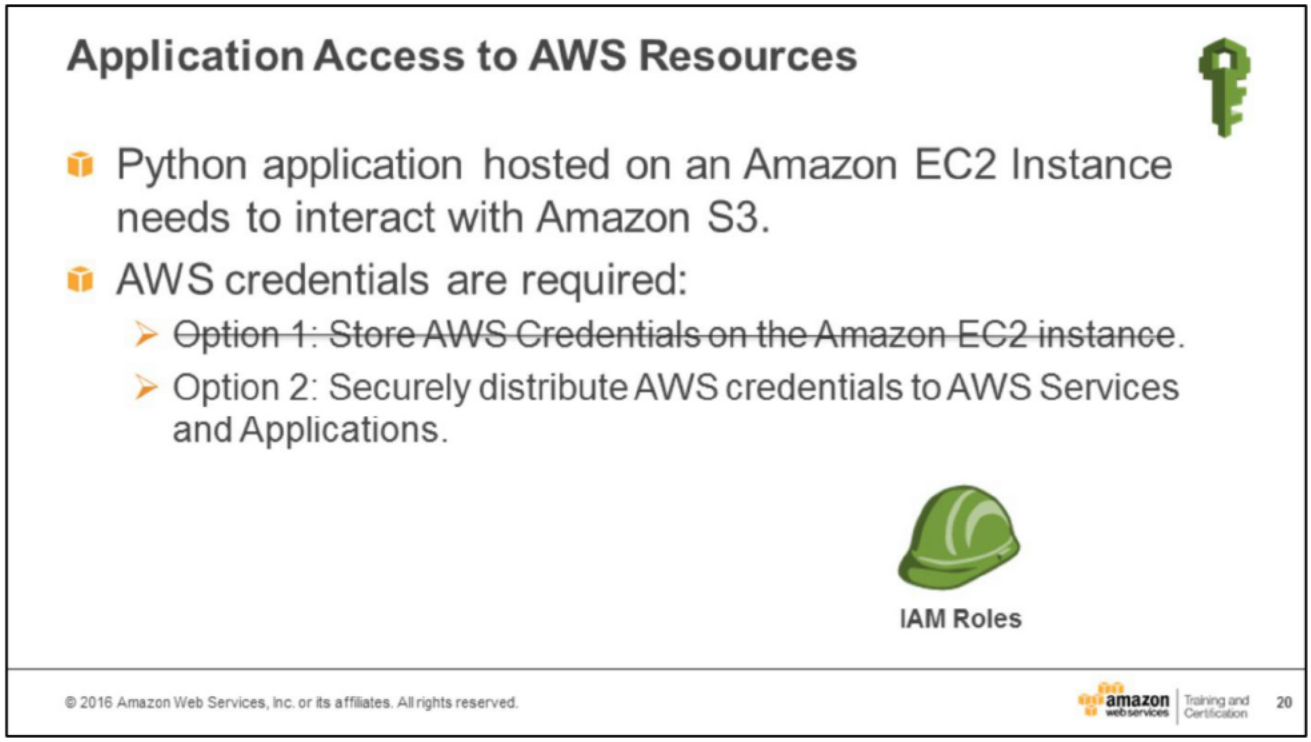
An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have any credentials (password or access keys) associated with it. Instead, if a user is assigned to a role, access keys are created dynamically and provided to the user.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html



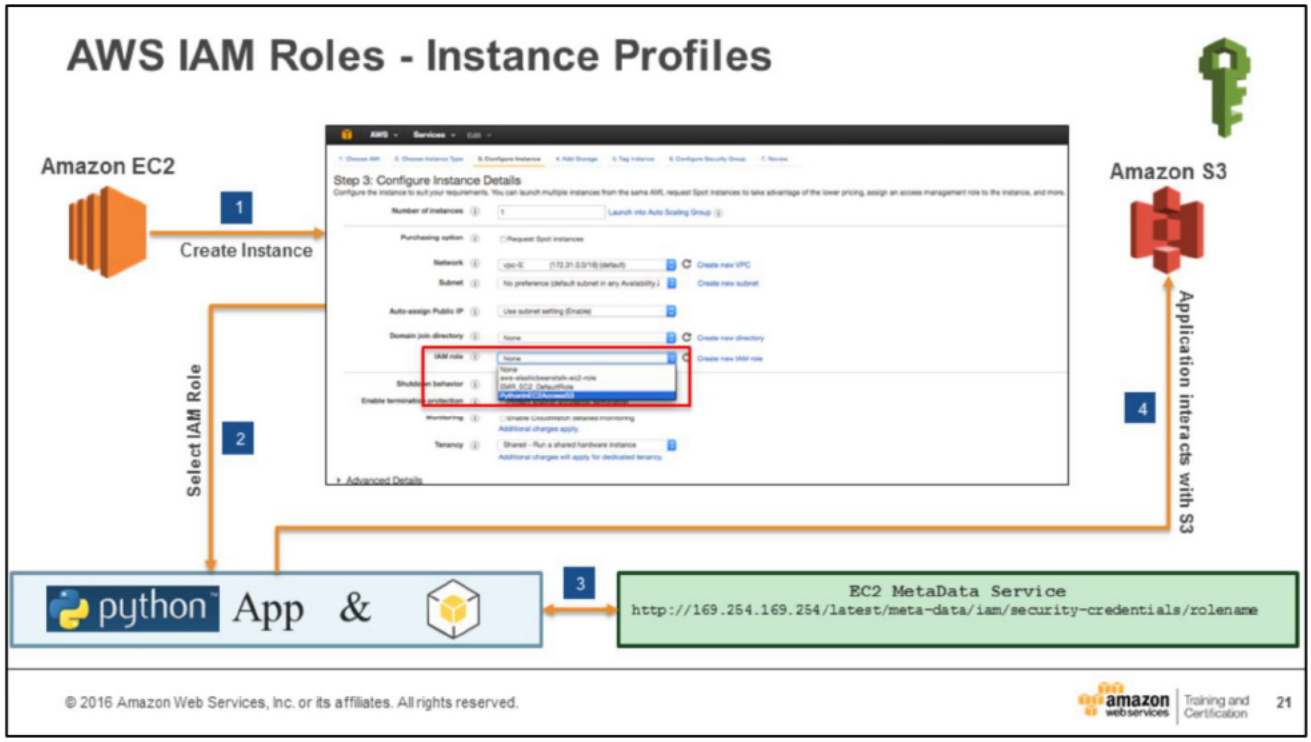
You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources.



In the example above a custom application written in Python and hosted on an Amazon EC2 instance needs to interact with objects stored in an Amazon S3 bucket. Applications may access AWS resources in multiple ways. One way is to embed your AWS access key ID and secret access key in the application code or in a Config file supported by the application. However, doing so may compromise the user's credentials. Changing or rotating the user's credentials would require an update in the code each time. This approach is not secure and feasible in many cases. The alternate and secure option is to use an IAM role to pass temporary security credentials as part of an instance profile.

For more information, see:

- Using Instance Profiles - http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html
- IAM Roles for Amazon EC2 - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html



An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.

In the example, an IAM role named PythonInEC2AccessS3 is created by an IAM user. The role grants access to an Amazon S3 bucket.

Step 1: An application developer selects the PythonInEC2AccessS3 role while creating the Amazon EC2 instance. The instance would host a Python application which would need access to an Amazon S3 bucket. Note: An IAM role may be associated with an EC2 instance only during creation. The policy associated with the role may be modified at any time. A user launching an EC2 instance also needs appropriate permissions to associate an IAM role to the EC2 instance.

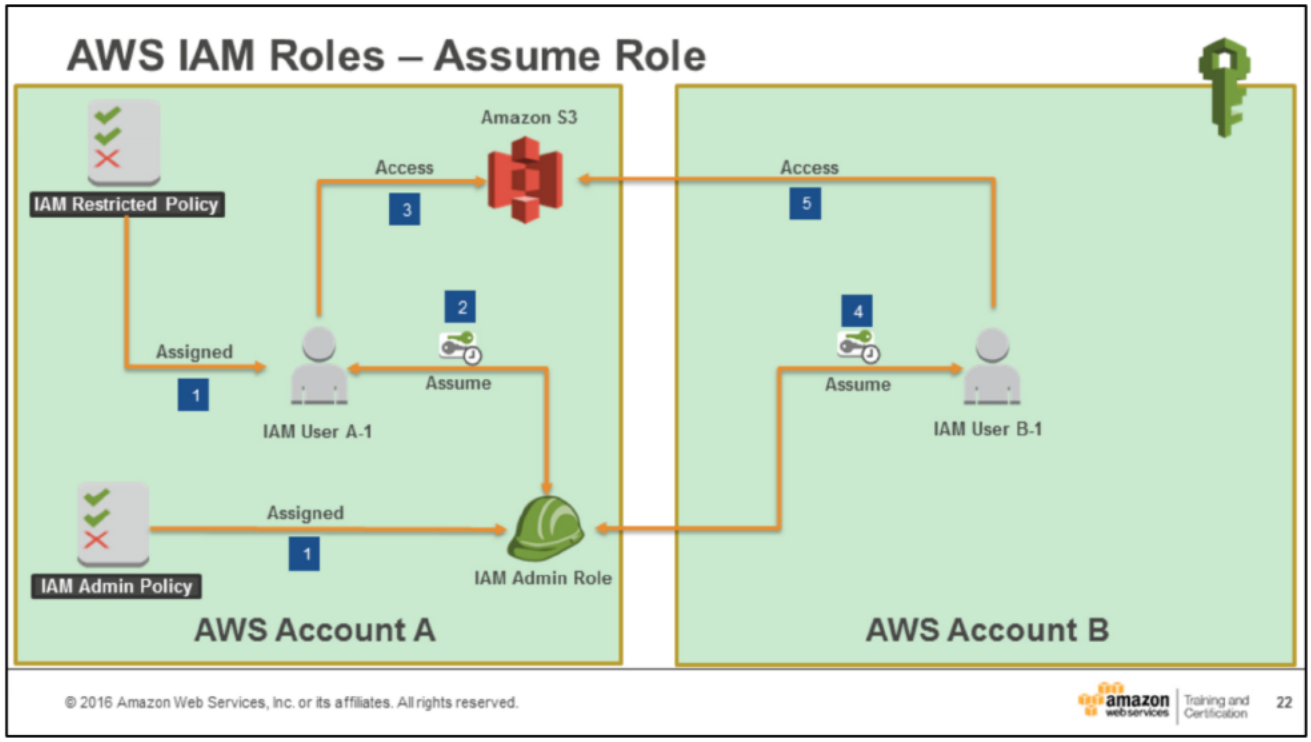
Step 2: Post instance creation the Python application is installed on the EC2 instance. AWS SDK for Python (Boto3) is also installed on the instance. The application tries to access an Amazon S3 bucket. However AWS credentials are not available on the instance.

Step 3: The Python application then uses the EC2 metadata service to gain access to Temporary Security Credentials. Temporary Security Credentials will be discussed later.

Step 4: The application interacts with the Amazon S3 bucket specified in the *PythonInEC2AccessS3* role.

For more information, see:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html#instance-metadata-security-credentials



IAM roles may also be associated with users.

In the above example, there are two AWS accounts A and B. *IAM User A-1* is part of *Account A* and *IAM User B-1* is part of *Account B*.

Step 1: An IAM policy named *IAM Admin Policy* with access to an Amazon S3 bucket is associated to an IAM role named *IAM Admin Role*. User A-1 has an IAM policy with restricted access. This is done as *User A-1* does not normally need administrative privileges. However, User A-1 may sometimes have to perform tasks that require administrative privileges.

Step 2: When required *User A-1* assumes the *IAM Admin Role.* Doing so gives *User A-1* access to the S3 bucket. A user who assumes a role temporarily gives up his or her own permissions and instead takes on the permissions of the role. When the user exits, or stops using the role, the original user permissions are restored. It is therefore helpful to use IAM roles instead of changing the user's policies each time a change is required.

Note: *User A-1*'s policy must contain permissions to assume the role.

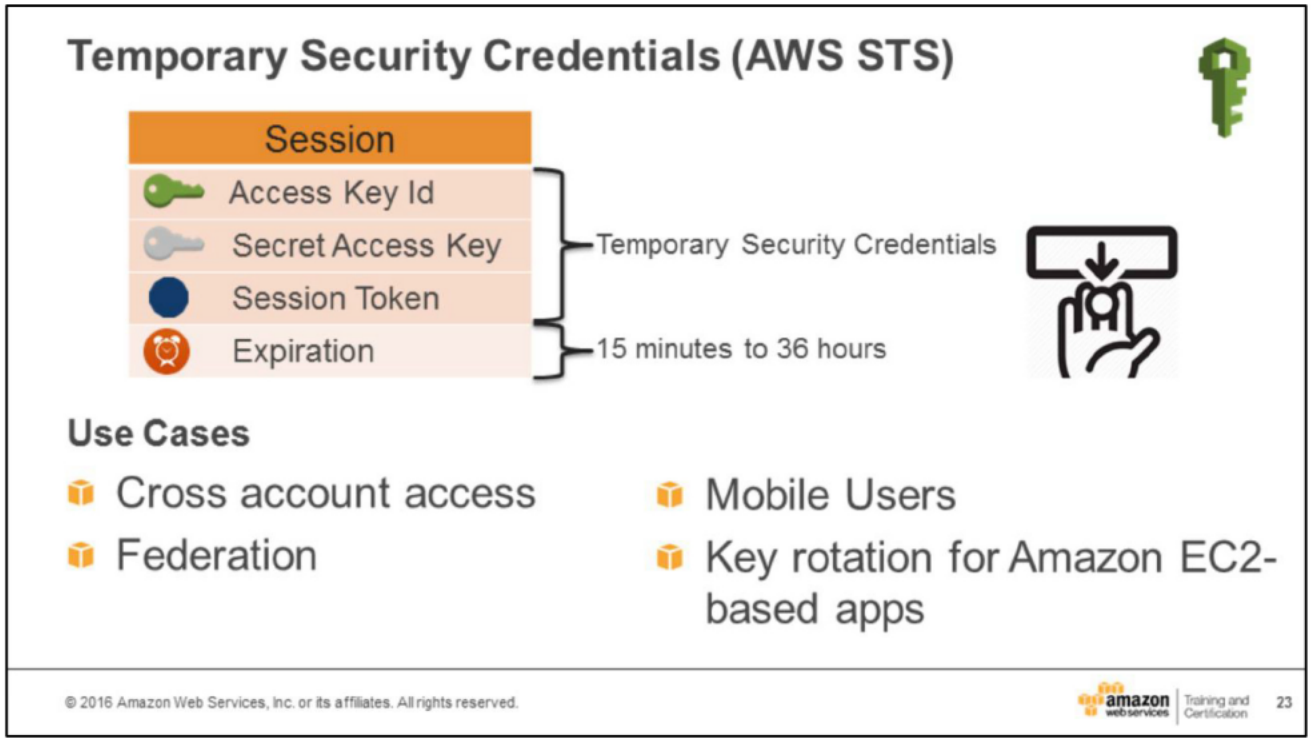
Step 3: *User A-1* gains access to the Amazon S3 bucket.

Step 4: With IAM roles, you can establish trust relationships between your trusting account and other AWS trusted accounts. The trusting account owns the resource to be accessed and the trusted account contains the users who need access to the resource. *User B-1* from *Account B* assumes the *IAM Admin Role* from *Account A*.

Step 5: *User B-1* gains access to the Amazon S3 bucket owned by *Account A*.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-user.html

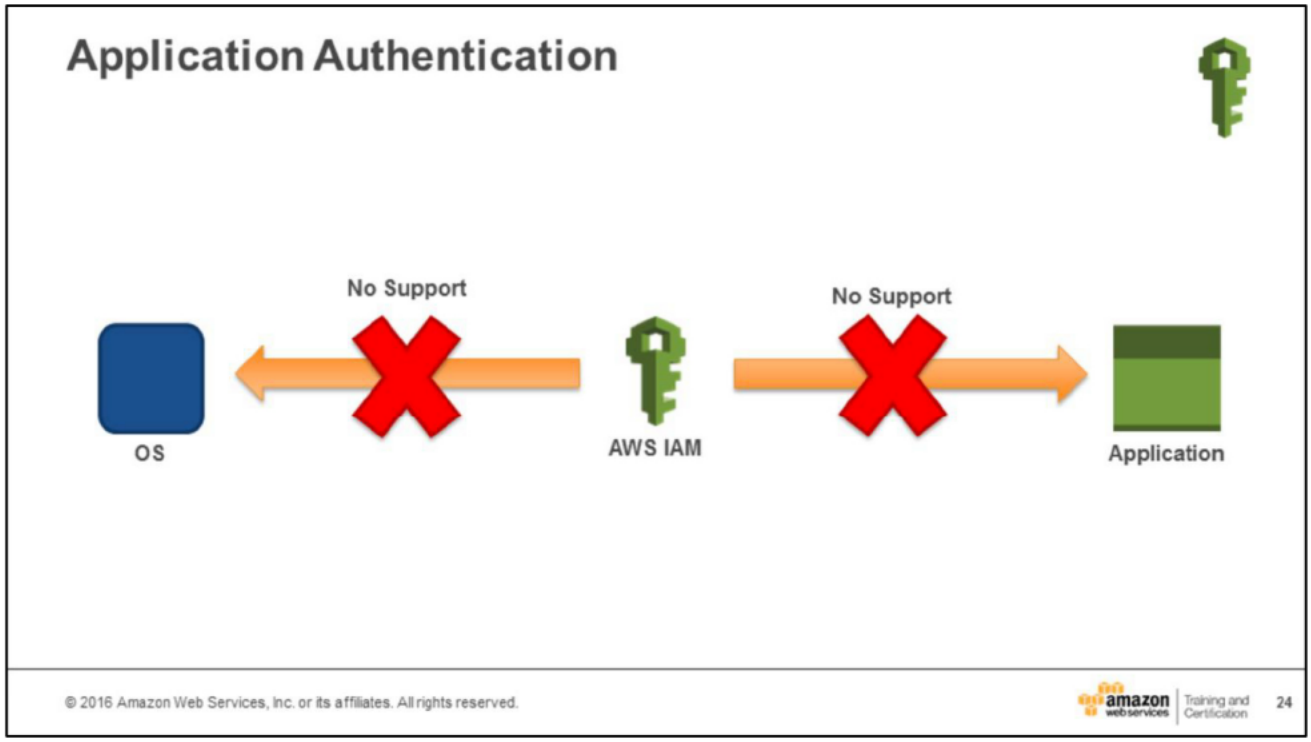


AWS Security Token Service (AWS STS) provides trusted users with temporary security credentials that can control access to your AWS resources. These credentials are short-term and work almost identically to the long-term access key credentials. These credentials are generated dynamically and provided to the user when requested.

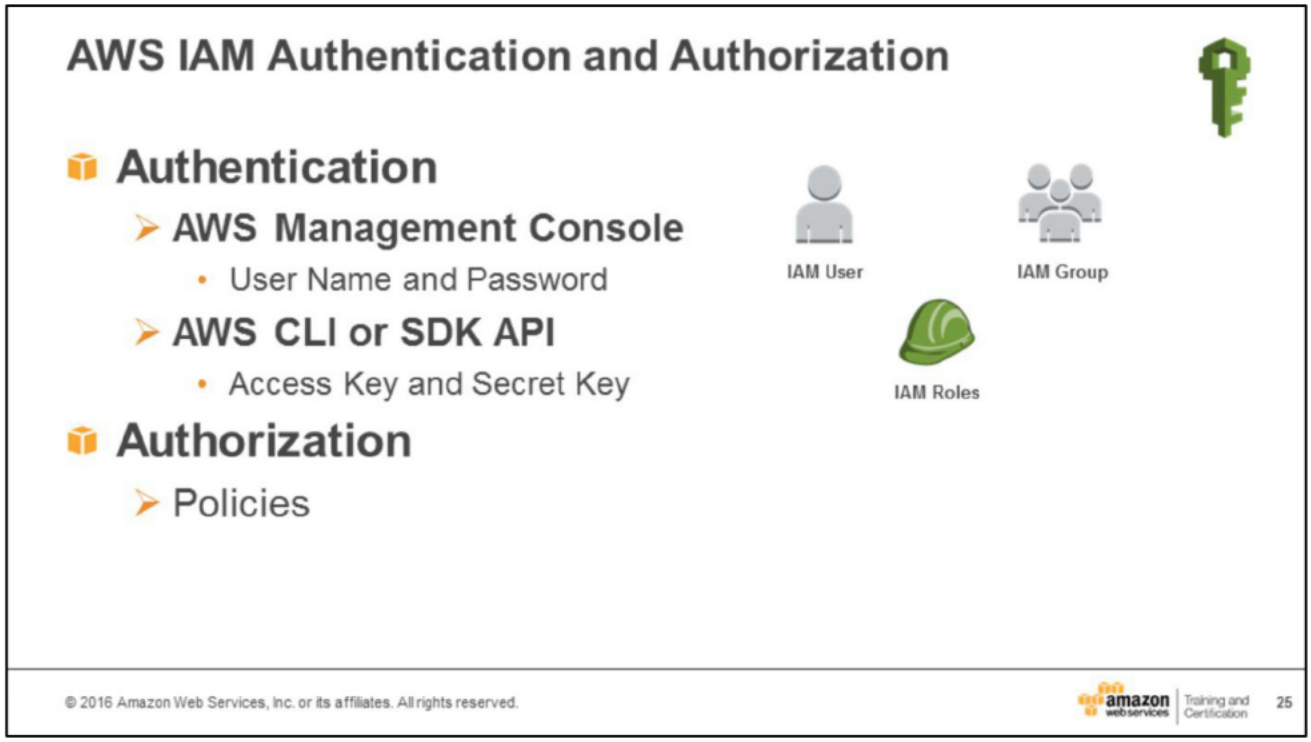
A session established with AWS STS consists of an access key ID, secret access key, a session token, and an expiration time. The expiration time could last between 15 minutes to 36 hours. The keys are used to sign API requests and pass in the token as an additional parameter, which AWS uses to verify that the temporary access keys are valid.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html



AWS IAM is not appropriate for OS and application authentication.

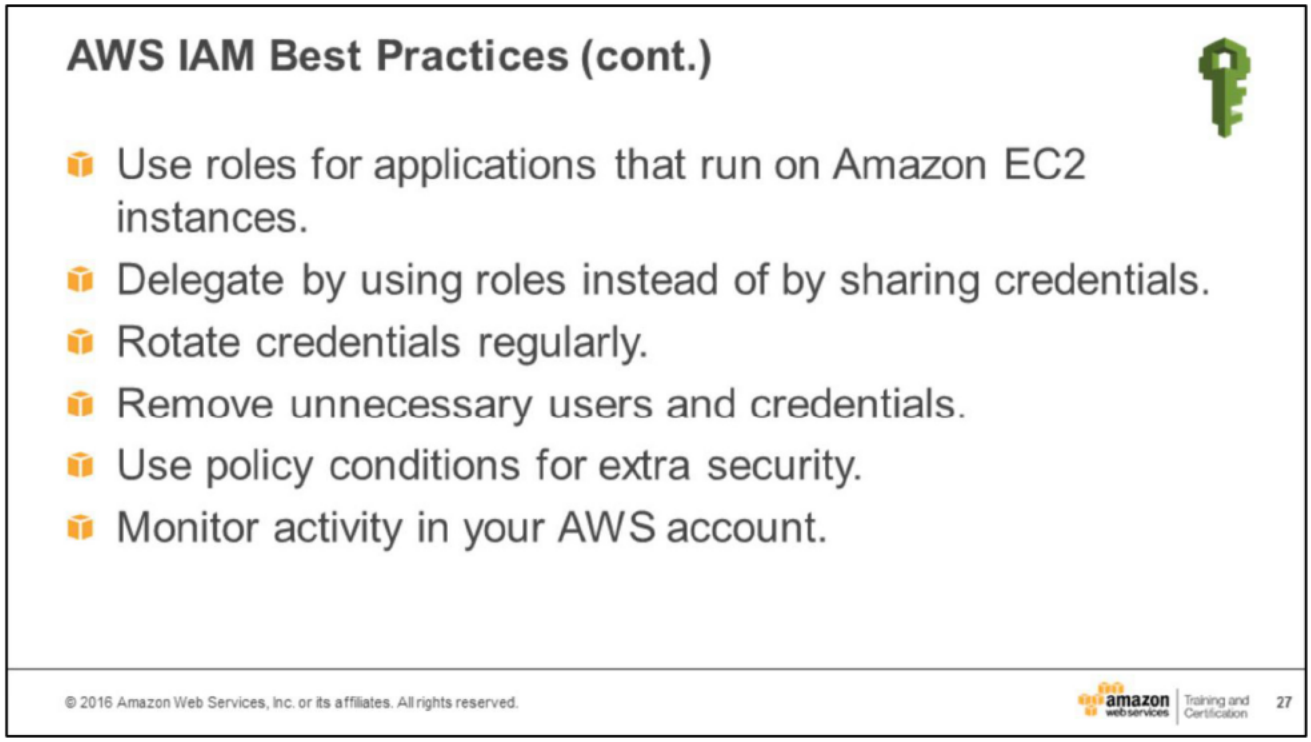


IAM is a powerful service to authenticate and authorize users and AWS resources.

AWS IAM Best Practices

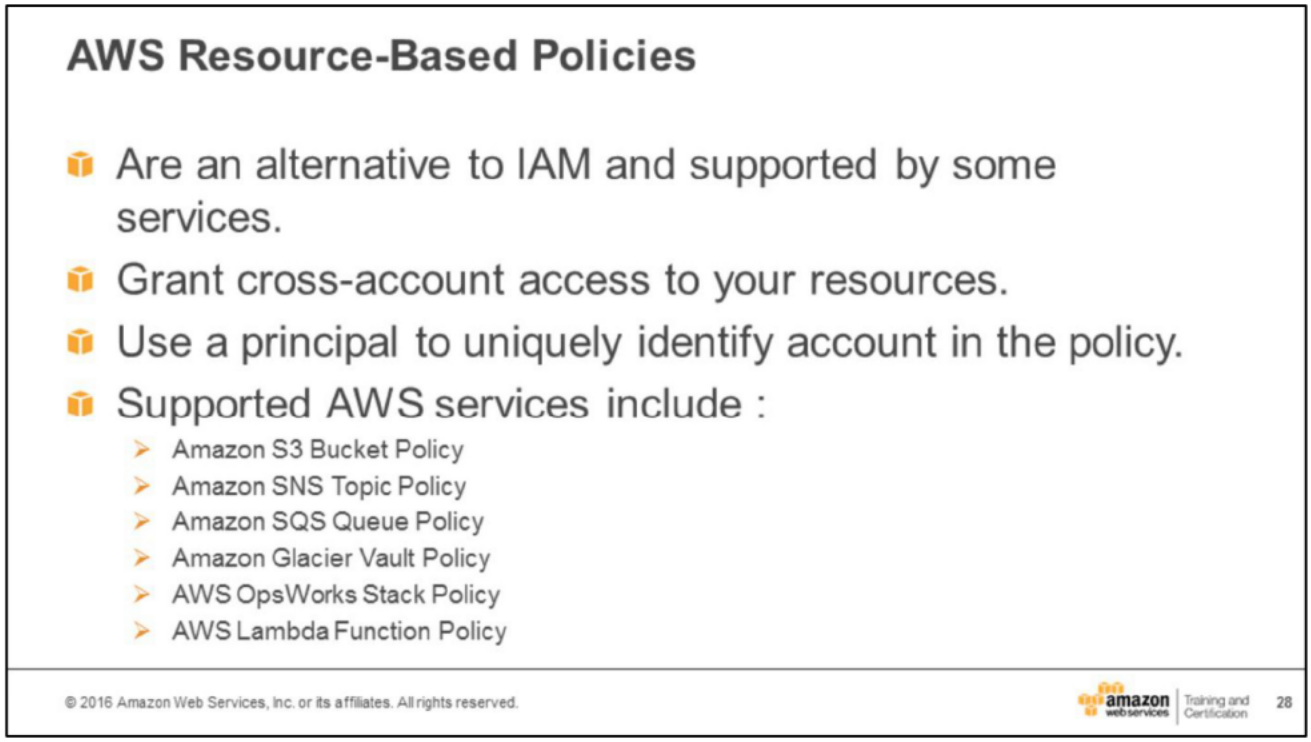
- Delete AWS account (root) access keys.
- Create individual IAM users.
- Use groups to assign permissions to IAM users.
- Grant least privilege.
- Configure a strong password policy.
- Enable MFA for privileged users.

The slide shows some best practices to follow with IAM.



For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

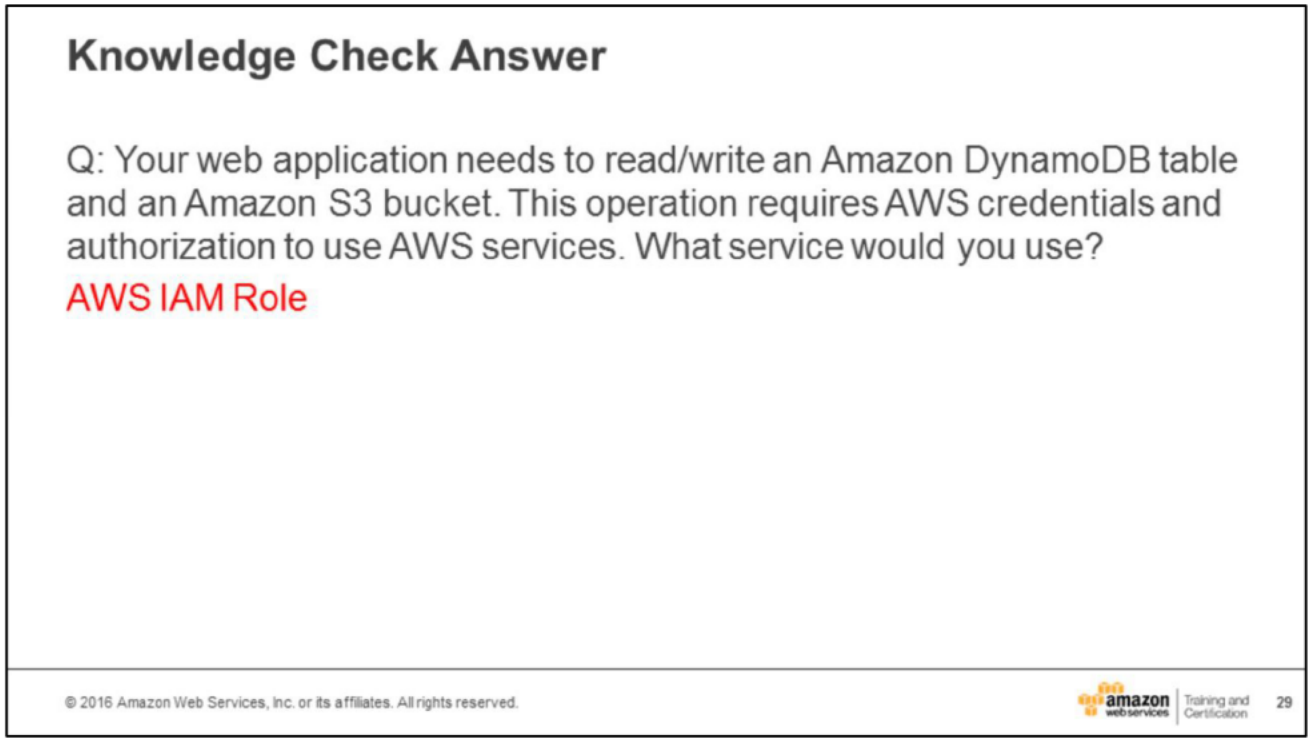


For some AWS services, you can grant cross-account access to your resources. To do this, you attach a policy directly to the resource that you want to share, instead of using a role as a proxy. The resource that you want to share must support resource-based policies. Unlike a user-based policy, a resource-based policy specifies who (in the form of a list of AWS account ID numbers) can access that resource. Cross-account access with a resource-based policy has an advantage over a role. With a resource that is accessed through a resource-based policy, the user still works in the trusted account and does not have to give up his or her user permissions in place of the role permissions. In other words, the user continues to have access to resources in the trusted account at the same time as he or she has access to the resource in the trusting account. This is useful for tasks such as copying information to or from the shared resource in the other account.

Principal: This element defines an account in a policy. In a resource-based policy, the principal may refer to the same account or another account.

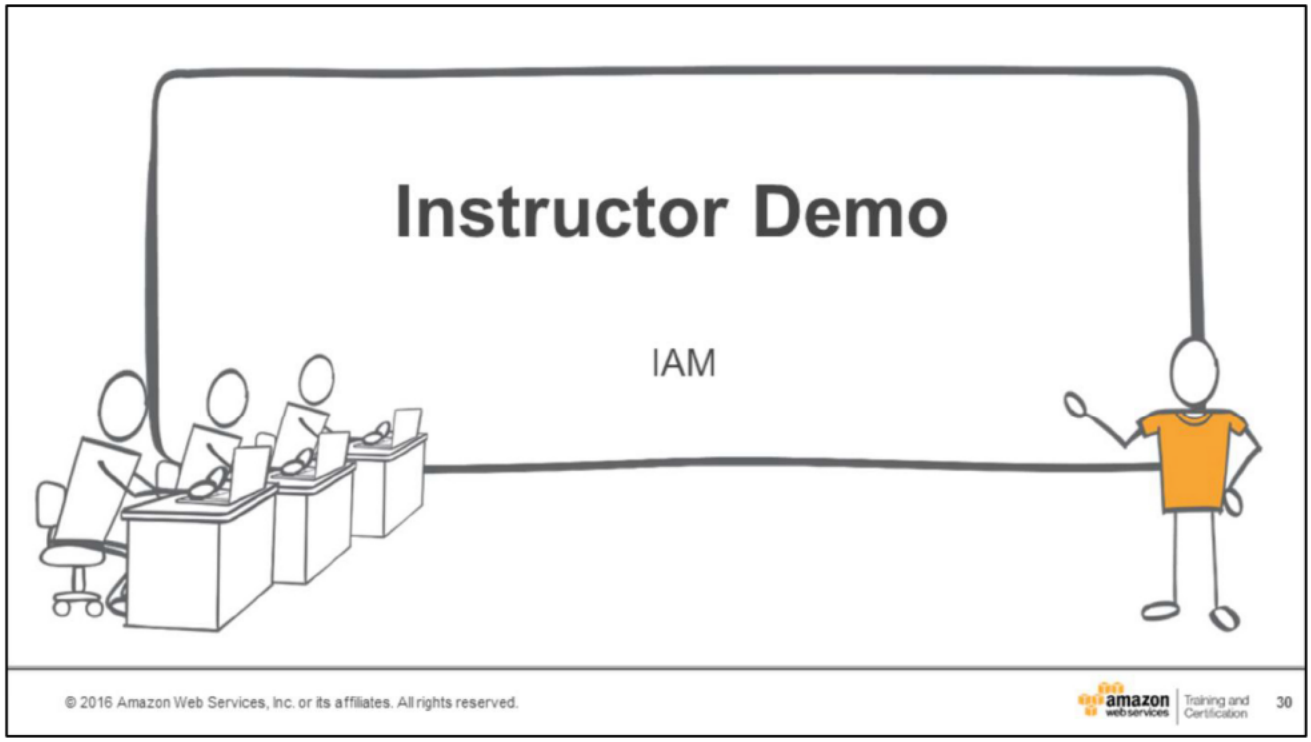
For more information, see:

How IAM Roles Differ from Resource-based Policies - http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_compare-resource-policies.html

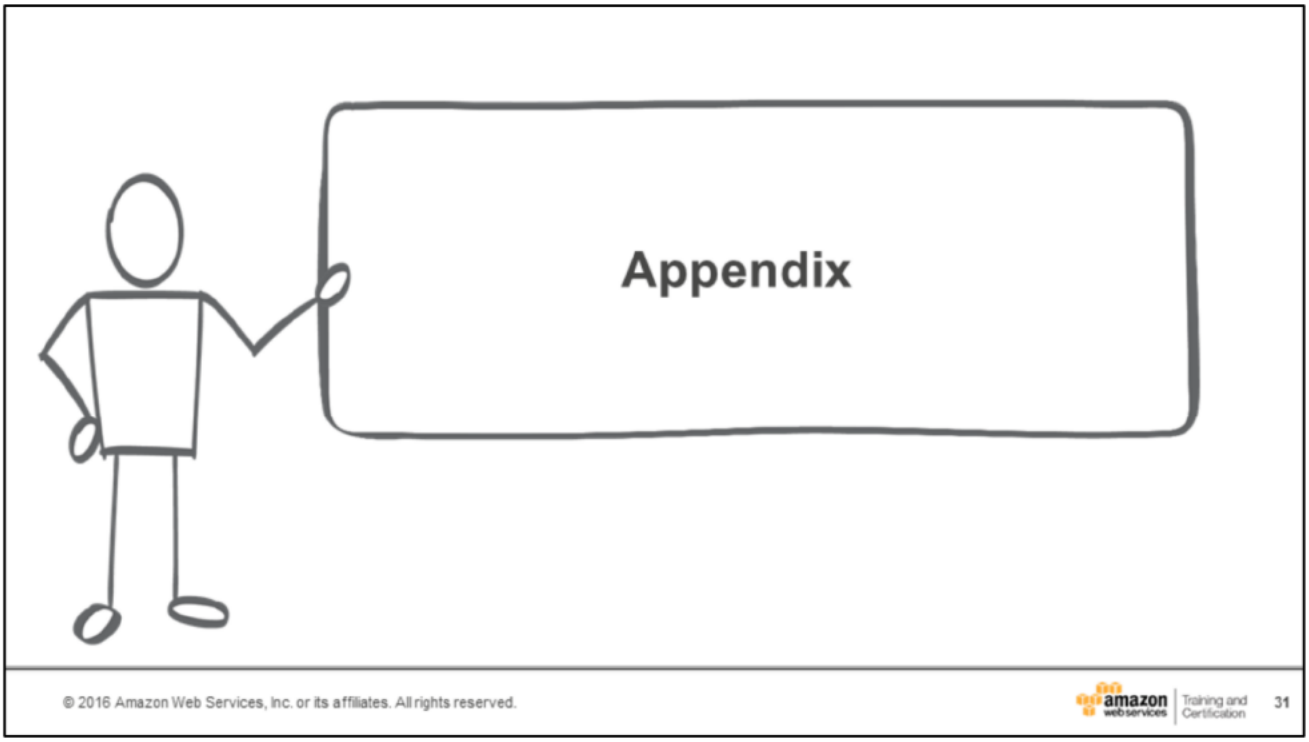


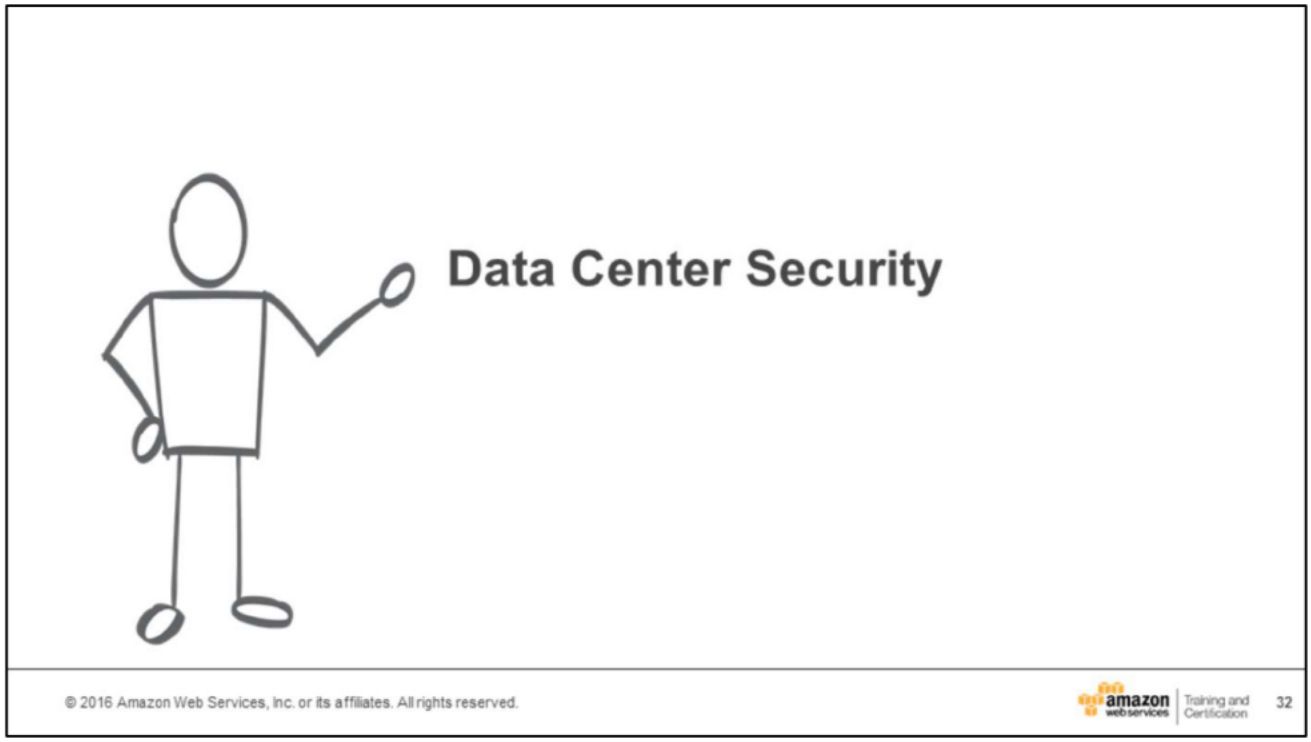
You have reached the end of this training module. In summary, you have learned the key fundamental elements of AWS deployment management products and services.

Test out some of your new skills!



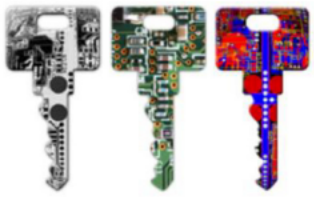
The instructor will demonstrate how to create a user, role, group, and policy using the IAM console.


Appendix
© 2016 Amazon Web Services, Inc. or its affiliates. All rights reserved.
amazon web services | Training and Certification
31



Understand common methods used to secure data centers.

Physical & Environmental Security



- Lock your data center.
- Only provide access to those who need it.
- Keep track of access.
- Mount servers on racks with locks.
- Have redundant utilities.
- Build your data center with security in mind.

Physical and environmental security is important in an on-premises data center. Remember to lock your data center, only provide access to people who need it, and keep track of the access. Servers should be mounted on racks with locks. Ideally, data centers should have redundant utilities (electricity, water, voice, data, HVAC).

Establishing and maintaining a secure computing environment is more difficult the more interconnected networks become. The slide shows common methods used to security your network and ensure confidentiality, integrity, and availability of systems and data.

Identification & Authentication: The careful user of user accounts is an important aspect of security in your network. Least privilege access should be provided and user accounts should be properly maintained to avoid unauthorized access to resources.

Firewalls: A firewall is a security router that sits between the Internet and your network. The firewall's purpose is to act as a security guard between the Internet and the network it is guarding.

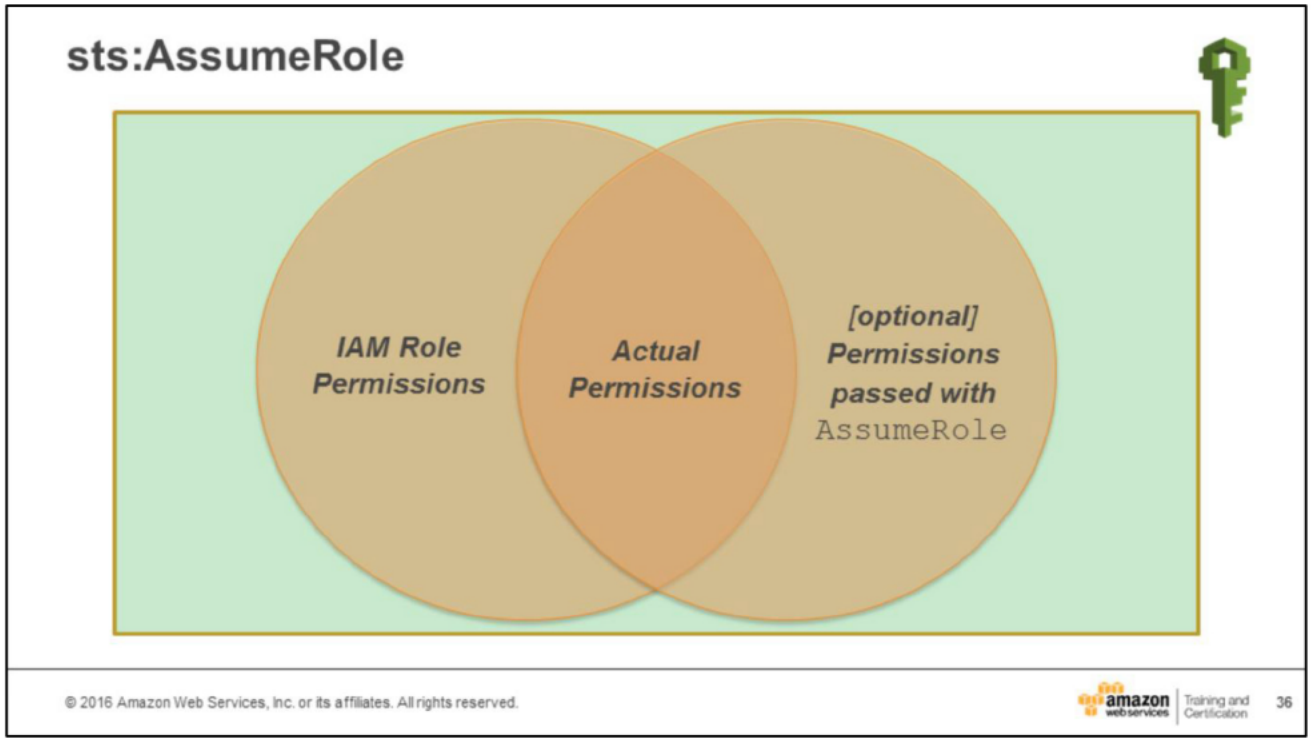
Patching: Maintaining operating system updates (patching) is another important aspect of network security. Software patches are minor updates that fix bugs and address security flaws that can be abused by hackers. Keeping your operating systems up to date with the latest patches can help facilitate a safer environment for your network.

Virus Protection: Antivirus programs can help detect and remove known threats to clients on your network.

Encryption: Encryption processes use numeric keys and algorithms to scramble data when it is sent over the network or saved on disk. This helps prevent unauthorized users from reading and accessing confidential or sensitive data. Effective encryption is important in the effective use of a virtual private network (VPN).

For more information, see:

- AssumeRole - http://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html
- GetFederationToken - http://docs.aws.amazon.com/STS/latest/APIReference/API_GetFederationToken.html

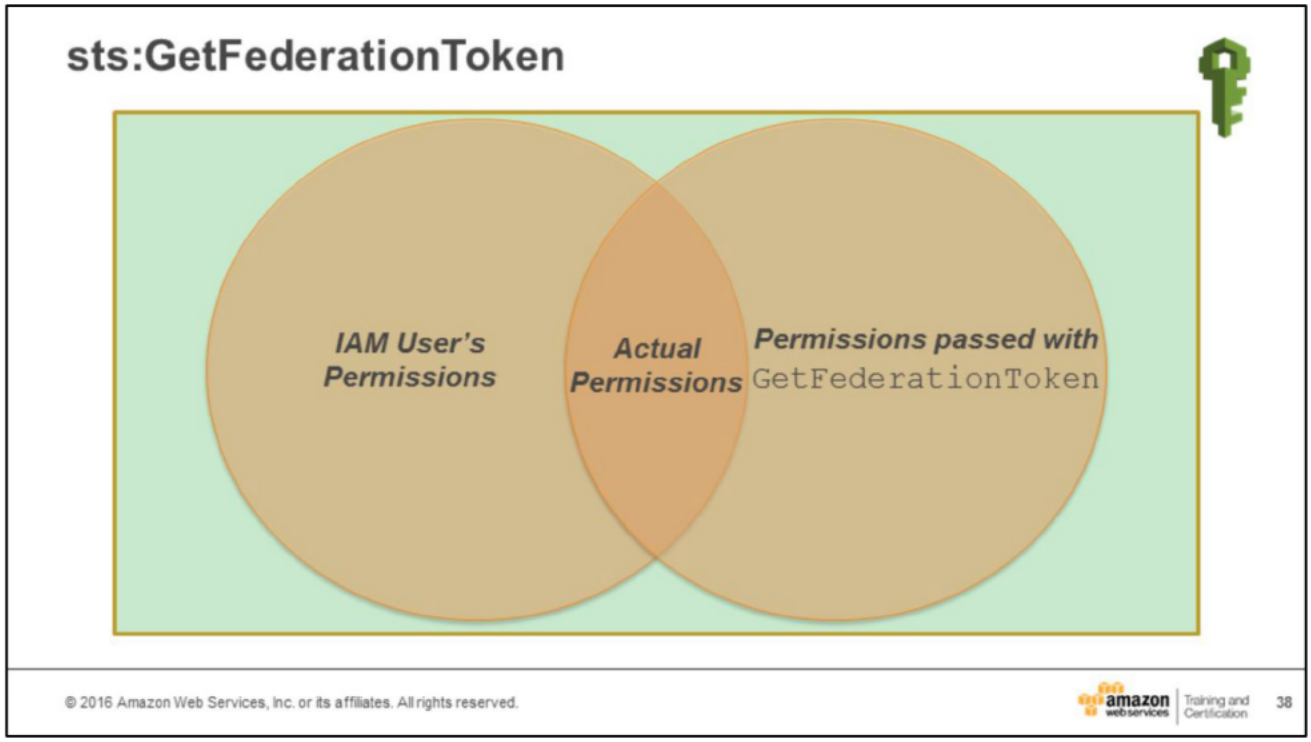


For more information, see:

http://blogs.aws.amazon.com/security/post/Tx1DM54S2Q7TC8U/Understanding-the-API-options-for-securely-delegating-access-to-your-AWS-account

For more information, see:

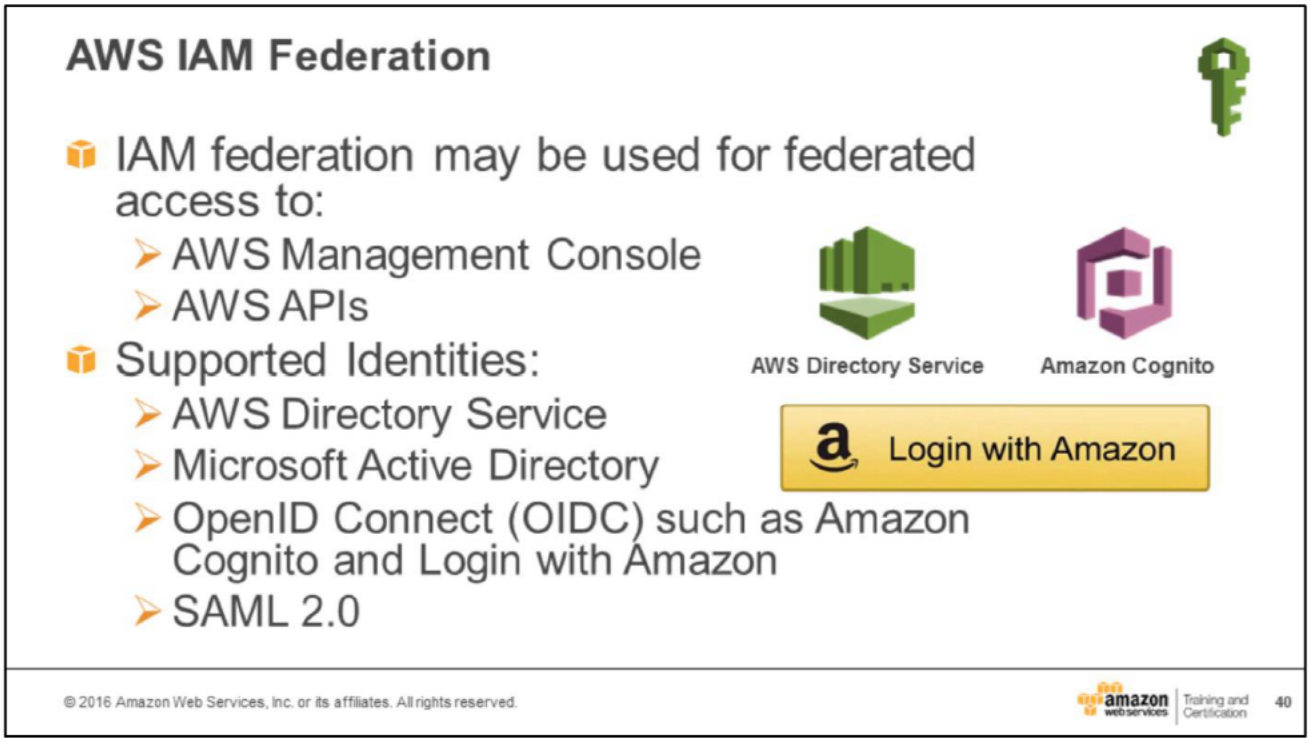
- AssumeRole - http://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html
- GetFederationToken - http://docs.aws.amazon.com/STS/latest/APIReference/API_GetFederationToken.html
- Creating a URL that Enables Federated Users to Access the AWS Management Console (Custom Federation Broker) - http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_enable-console-custom-url.html



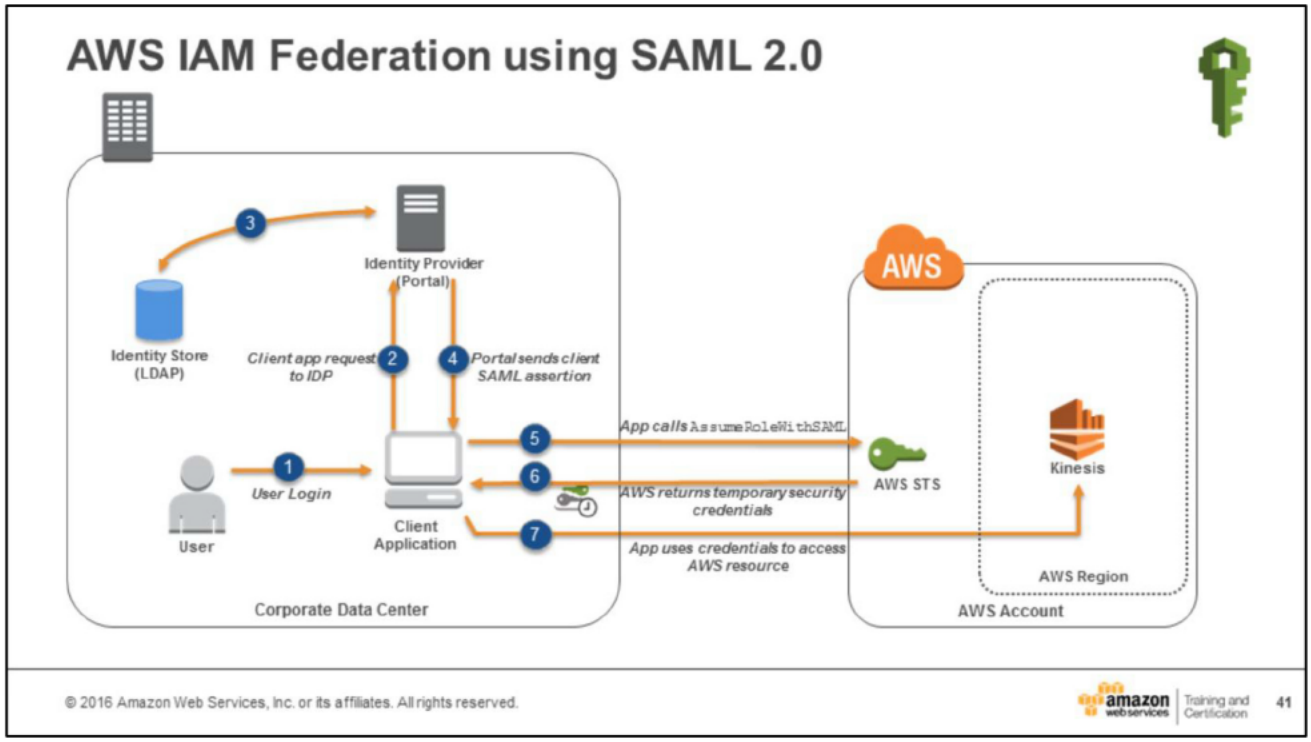
For more information, see:

http://blogs.aws.amazon.com/security/post/Tx1DM54S2Q7TC8U/Understanding-the-API-options-for-securely-delegating-access-to-your-AWS-account

IAM roles may be used in many scenarios to allow AWS services and resources to gain access to resources in the same or another account. Some uses cases are mentioned above.



AWS IAM supports identity federation for delegated access to the AWS Management Console or AWS APIs. With identity federation, external identities (federated users) are granted secure access to resources in your AWS account without having to create IAM users. These external identities can come from your corporate identity provider (such as Microsoft Active Directory or from the AWS Directory Service) or from a web identity provider, such as Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect (OIDC) compatible provider.



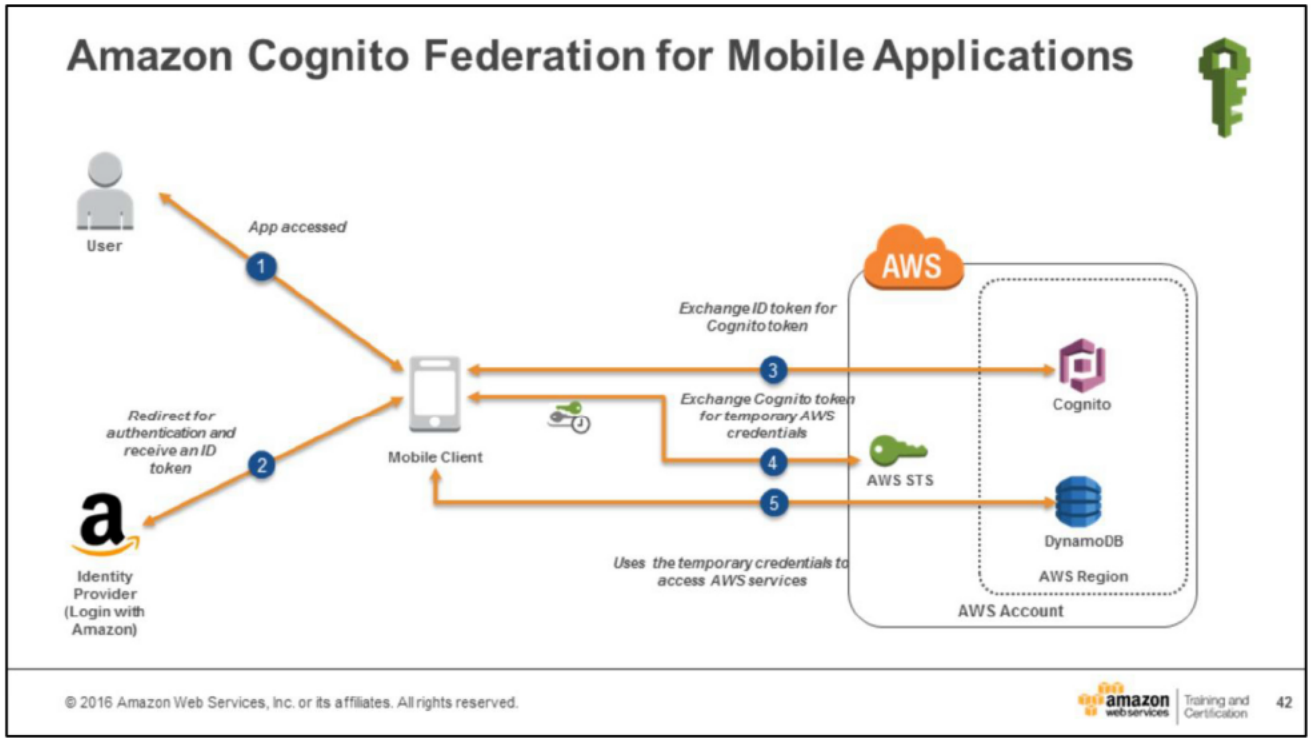
If you already manage user identities outside of AWS, you can use IAM *identity providers* instead of creating IAM users in your AWS account. With an identity provider (IdP), you can manage your user identities outside of AWS and give these external user identities permissions to use AWS resources in your account. This is useful if your organization already has its own identity system, such as a corporate user directory. It is also useful if you are creating a mobile app or web application that requires access to AWS resources.

AWS supports identity federation with SAML 2.0 (Security Assertion Markup Language 2.0), an open standard that many identity providers (IdPs) use.

Example: You want to provide a client application access to an Amazon Kinesis Stream. Developers could build an application that users can use to access the Kinesis Stream. The end users however do not directly need access to the AWS account. Instead, the application communicates with an identity provider (IdP) to authenticate the user. The IdP gets the user information from your organization's identity store (such as an LDAP directory) and then generates a SAML assertion that includes authentication and authorization information about that user. The application then uses that assertion to make a call and get temporary security credentials. The app can then use those credentials to access the Amazon Kinesis Stream.

For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html



The preferred way to use web identity federation is to use Amazon Cognito. For example, Adele the developer is building a game for a mobile device where user data such as scores and profiles is stored in Amazon S3 and Amazon DynamoDB. Adele could also store this data locally on the device and use Amazon Cognito to keep it synchronized across devices. She knows that for security and maintenance reasons, long-term AWS security credentials should not be distributed with the game. She also knows that the game might have a large number of users. For all of these reasons, she does not want to create new user identities in IAM for each player. Instead, she builds the game so that users can sign in using an identity that they've already established with a well-known identity provider, such as Login with Amazon, Facebook, Google, or any OpenID Connect (OIDC)-compatible identity provider. Her game can take advantage of the authentication mechanism from one of these providers to validate the user's identity.

To enable the mobile app to access her AWS resources, Adele first registers for a developer ID with her chosen IdPs. She also configures the application with each of these providers. In her AWS account that contains the Amazon S3 bucket and DynamoDB table for the game, Adele uses Amazon Cognito to create IAM roles that precisely define permissions that the game needs. If she is using an OIDC IdP, she also creates an IAM OIDC identity provider entity to establish trust between her AWS account and the IdP.In the app's code, Adele calls the sign-in interface for the IdP that she configured previously. The IdP handles all the details of letting the user sign in, and the app gets an OAuth access token or OIDC ID token from the provider. Adele's app can trade this authentication information for a set of temporary security credentials that consist of an AWS access key ID, a secret access key, and a session token.

The app can then use these credentials to access web services offered by AWS. The app is limited to the permissions that are defined in the role that it assumes.The above figure shows a simplified flow for how this might work, using Login with Amazon as the IdP. For Step 2, the app can also use Facebook, Google, or any OIDC-compatible identity provider.

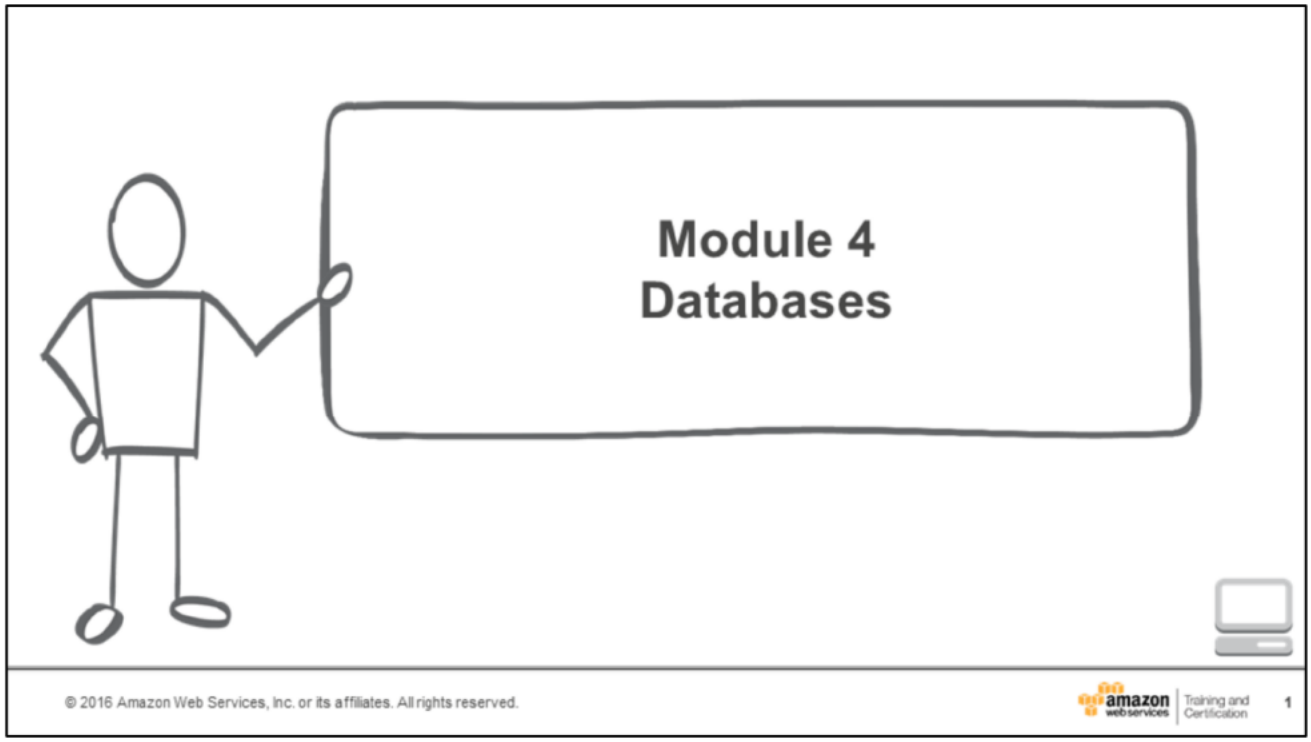
For more information, see:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc_cognito.html

AWS CloudTrail

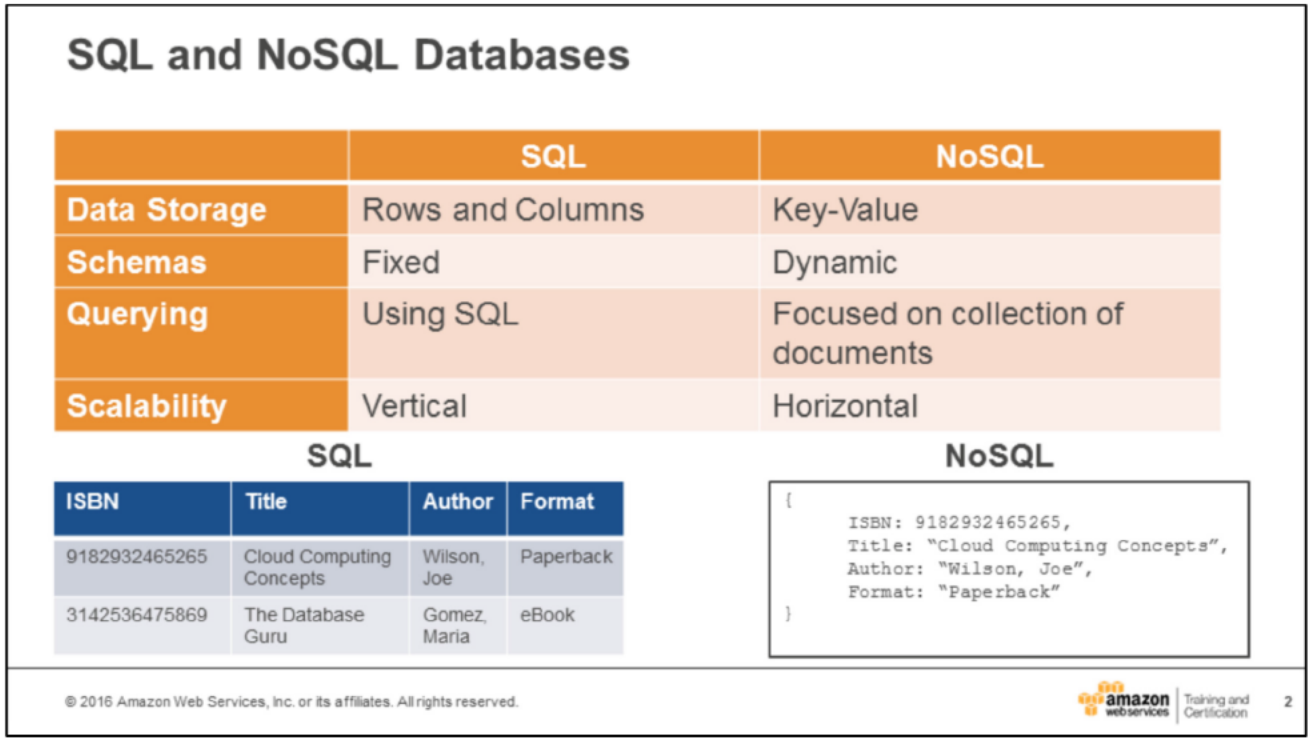
- Records AWS API calls for accounts.
- Delivers log files with information to an Amazon S3 bucket.
- Makes calls using the AWS Management Console, AWS SDKs, AWS CLI and higher-level AWS services.

AWS CloudTrail is a web service that records API calls to supported AWS services in your account and delivers log files to you.



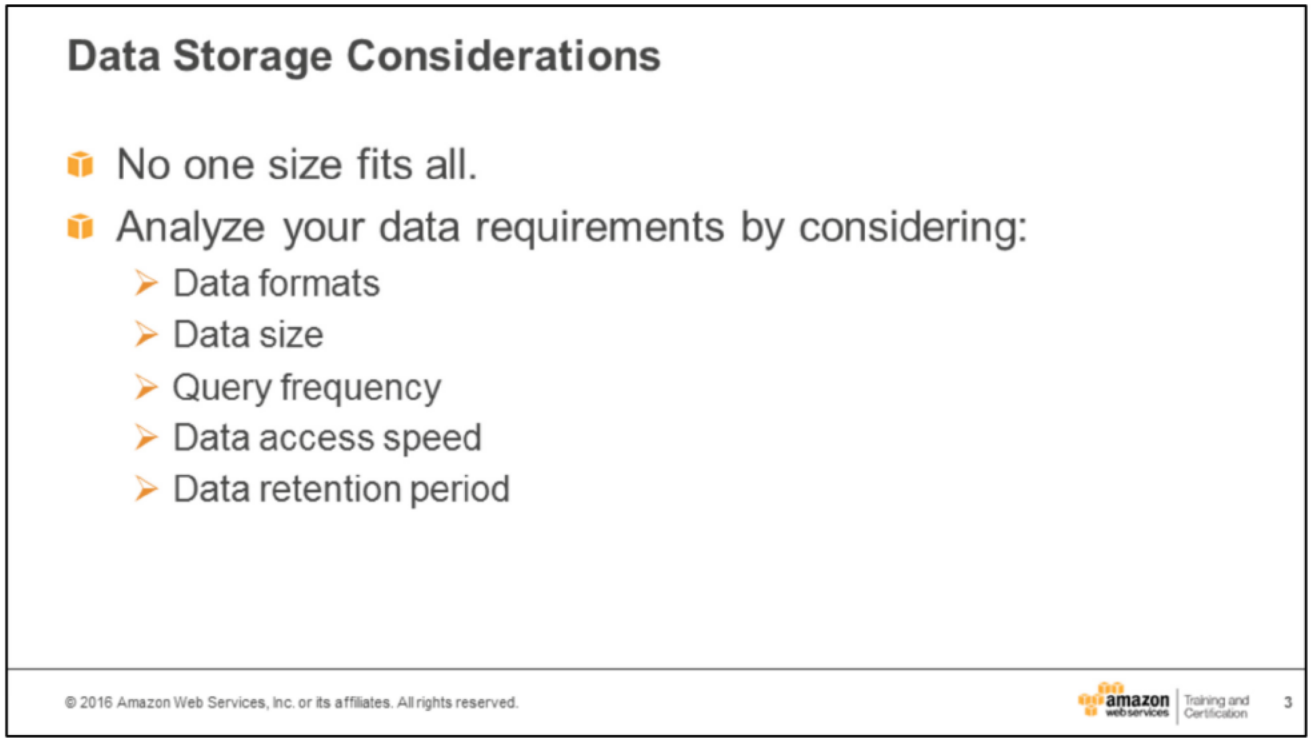
Understand the concepts of fundamental AWS database services including:

- Amazon Relational Database Service (RDS)
  - DB Instances
  - Security Groups
  - DB Parameter Groups
  - DB Option Groups
  - RDS Interfaces
- Amazon DynamoDB
  - DynamoDB Data Model
  - Supported Operations
  - Provisioned Throughput
  - Accessing DynamoDB

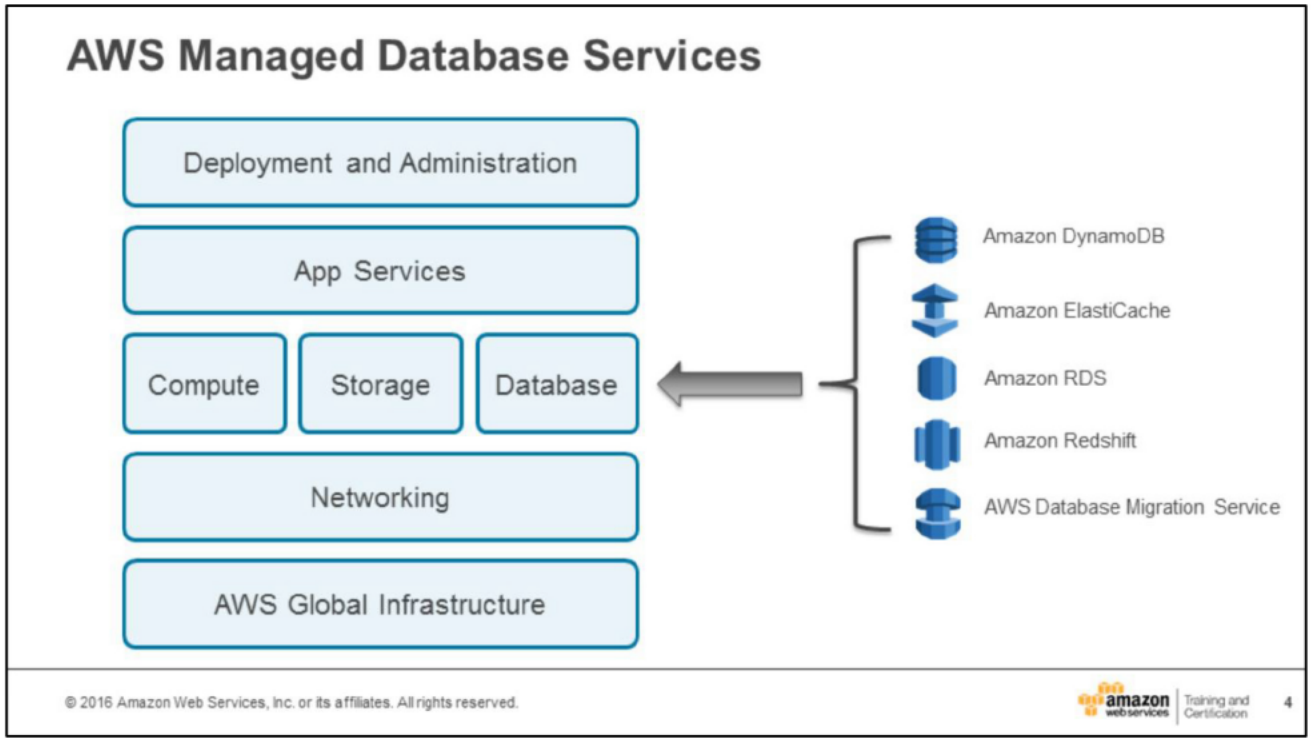


A SQL database stores data in rows and columns. Rows contain all the information about one entry and columns are the attributes that separate the data points. A SQL database schema is fixed – columns must be locked before data entry. Schemas can be amended if the database is altered entirely and taken offline. Data in SQL databases is queried using SQL (Structure Query Language), which can allow for complex queries. SQL databases scale vertically, by increasing hardware power.

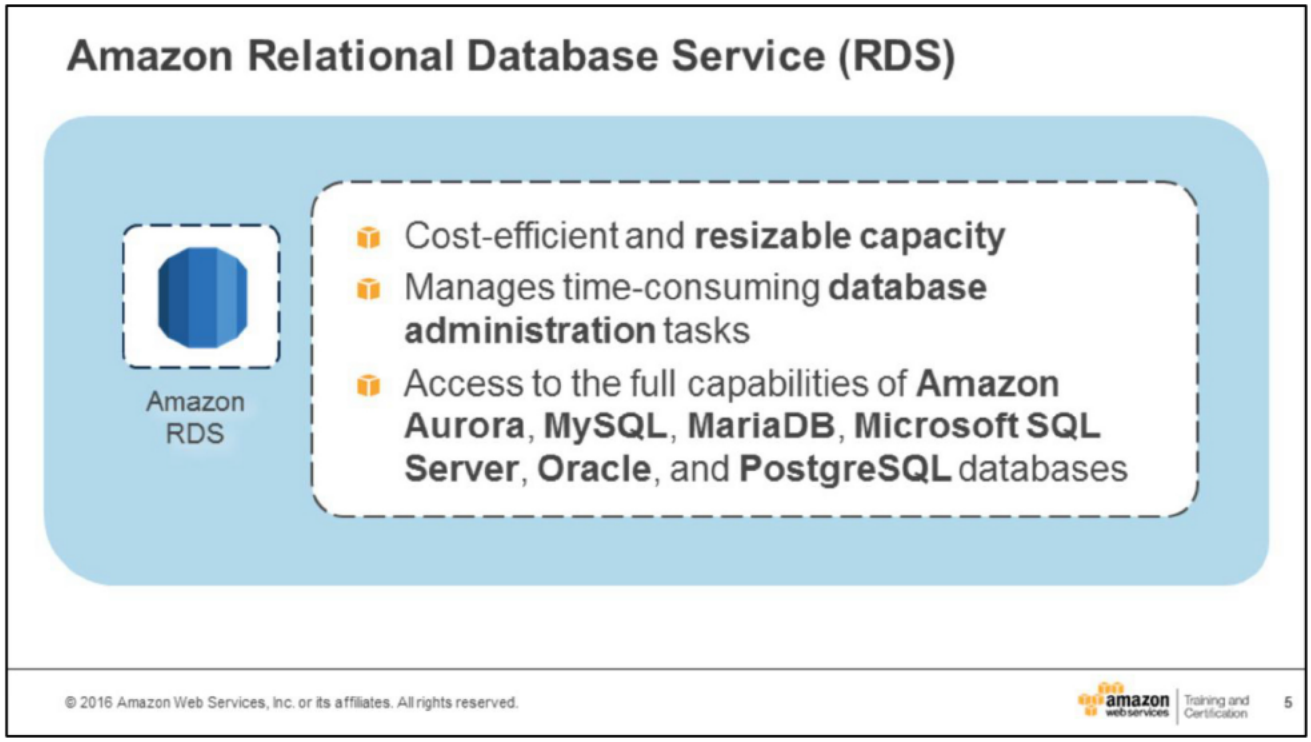
NoSQL databases store data using one of many storage models including key-value pairs, documents, and graphs. NoSQL schemas are dynamic and information can be added on the fly. Each 'row' doesn't have to contain data for each 'column'. Data in NoSQL databases is queried by focusing on collections of documents. NoSQL databases scale horizontally, by increasing servers.



No one size fits all when considering database types. You must take into consideration your data requirements such as data formats, data size, the frequency of your queries, how quickly you need your data, and for how long you need to keep it.

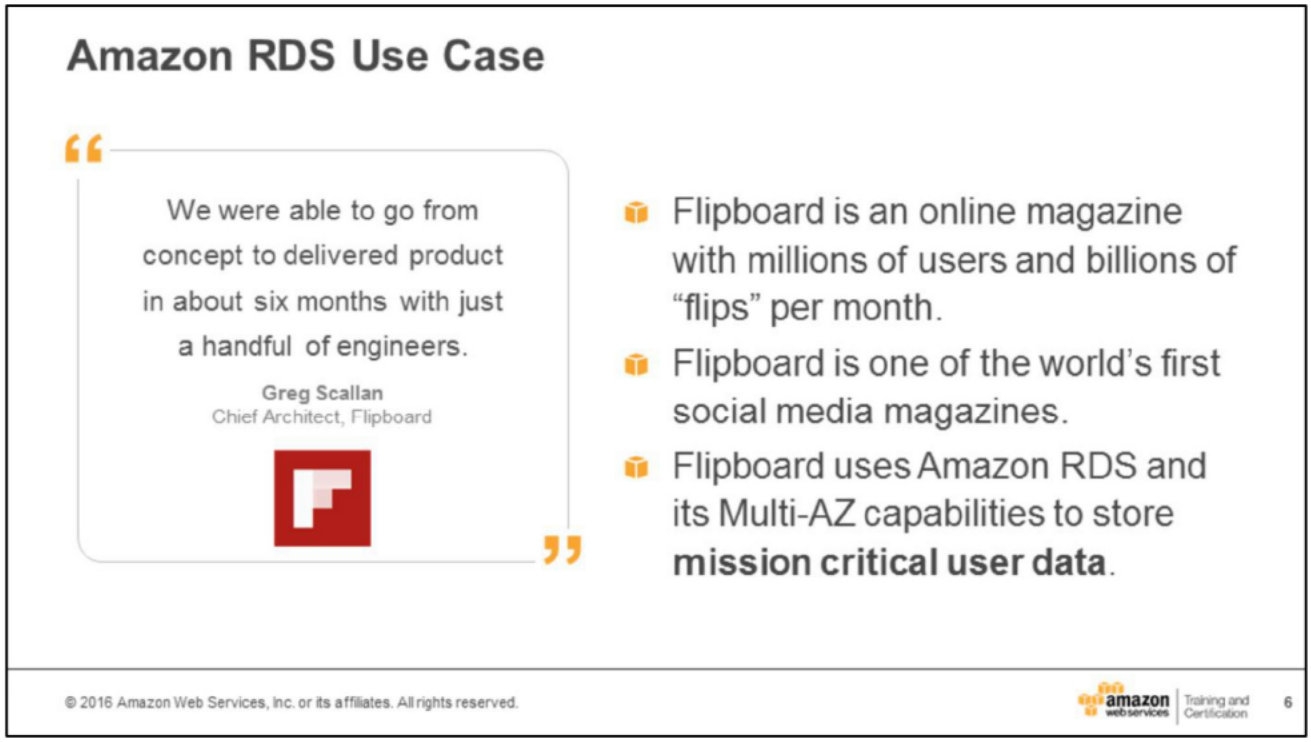


This slide shows the database services AWS provides.



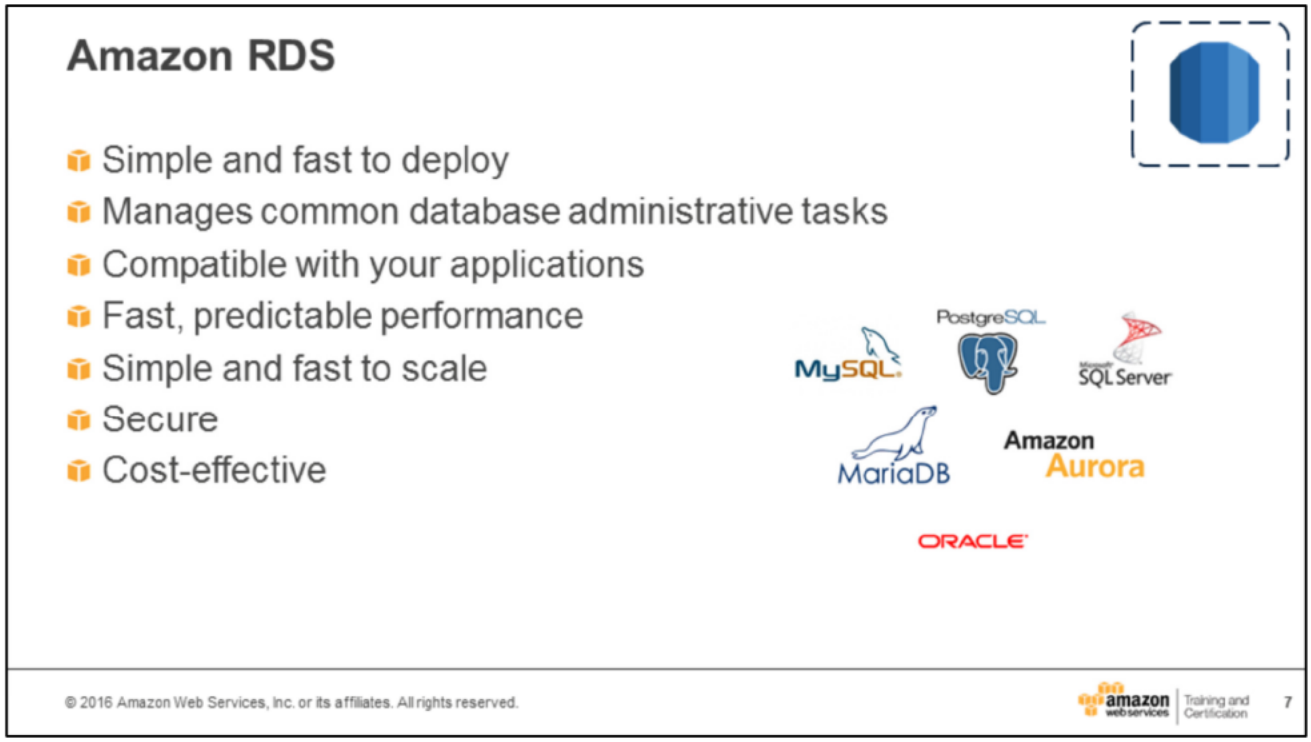
With Amazon RDS, you can access the full capabilities of a familiar MySQL, MariaDB, Microsoft SQL Server, Oracle, or PostgreSQL database. In addition, Amazon RDS for MySQL provides two distinct, but complementary, replication features: Multi-AZ deployments and read replicas that can be used in conjunction with each other to gain enhanced database availability, protect your latest database updates against unplanned outages, and scale beyond the capacity constraints of a single DB instance for read-heavy database workloads.

Amazon Aurora is a MySQL-compatible relational database engine that is part of Amazon RDS.



For more information, see:

https://aws.amazon.com/solutions/case-studies/flipboard/

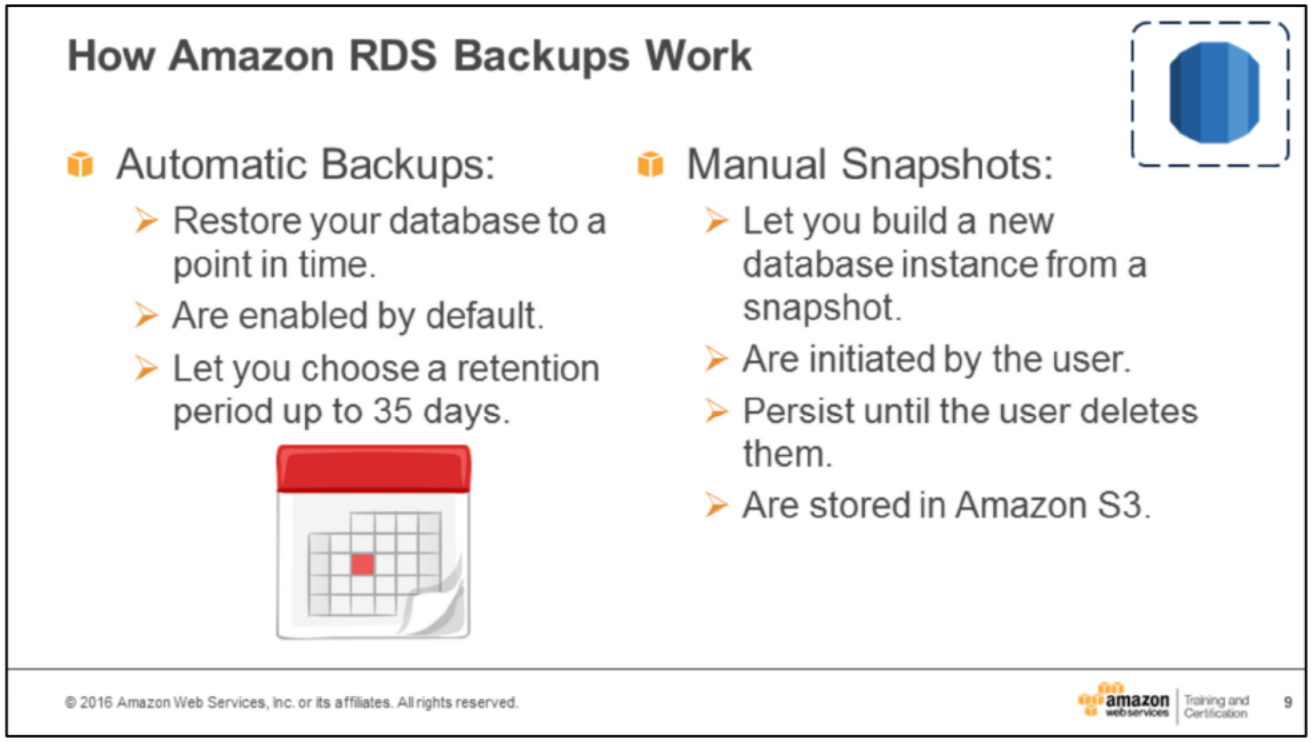


Amazon RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business. Amazon RDS gives you access to the full capabilities of a MySQL, Oracle, SQL Server, or Amazon Aurora database engine. This means that the code, applications, and tools you already use today with your existing databases can be used with Amazon RDS. Amazon RDS automatically patches the database software and backs up your database, storing the backups for a user-defined retention period and enabling point-in-time recovery. You benefit from the flexibility of being able to scale the compute resources or storage capacity associated with your relational database instance via a single API call.

DB Instances

- DB Instances are the basic building blocks of Amazon RDS.
- They are an isolated database environment in the cloud.
- They can contain multiple user-created databases.

The basic building block of Amazon RDS is the DB instance. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance. You can create and modify a DB instance by using the AWS Management Console, Amazon AWS command line interface, or the Amazon RDS API.



When automated backups are turned on for your DB Instance, Amazon RDS automatically performs a full daily snapshot of your data (during your preferred backup window) and captures transaction logs (as updates to your DB Instance are made). When you initiate a point-in-time recovery, transaction logs are applied to the most appropriate daily backup in order to restore your DB Instance to the specific time you requested. Amazon RDS retains backups of a DB Instance for a limited, user-specified period of time called the retention period, which by default is one day but can be set to up to thirty five days.

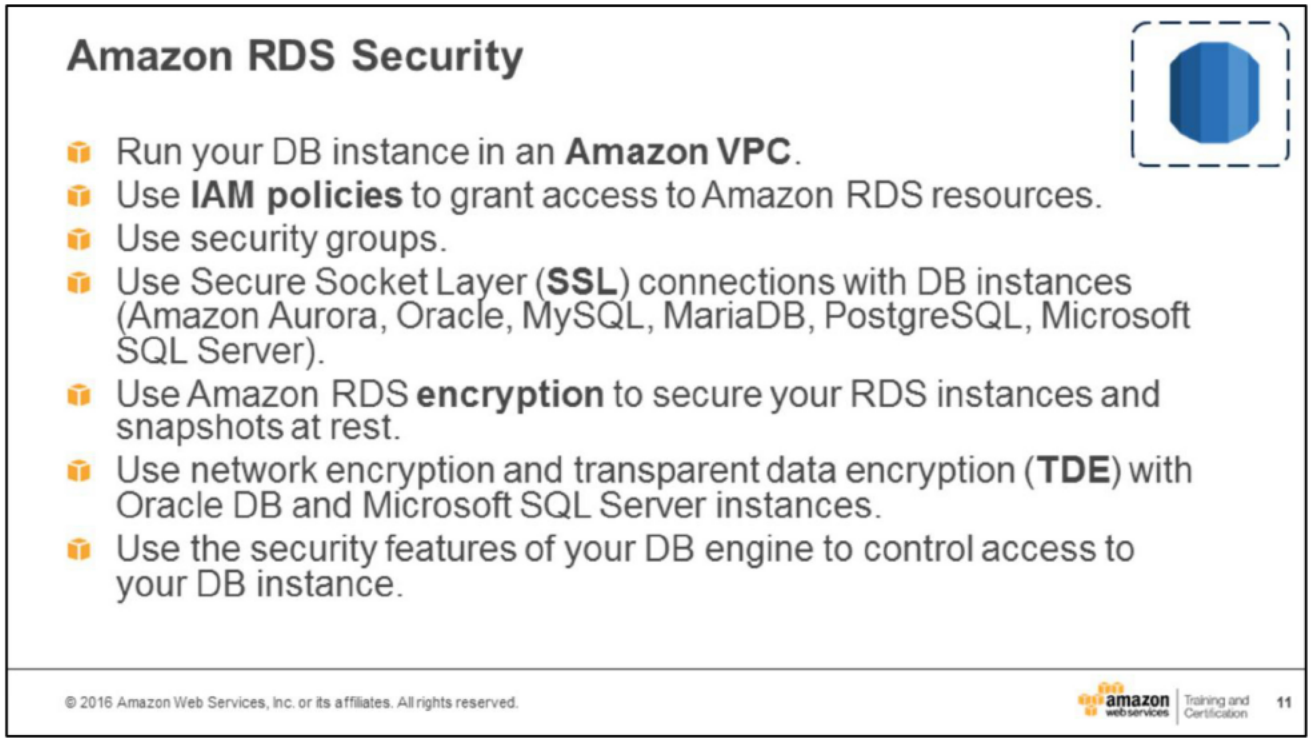
Manual database snapshots are user-initiated and enable you to back up your DB Instance in a known state as frequently as you want, and then restore to that specific state at any time. DB Snapshots can be created with the AWS Management Console or CreateDBSnapshot API and are kept until you explicitly delete them with the Console or DeleteDBSnapshot API.

Manual database snapshots are kept in Amazon Simple Storage Service (Amazon S3). There is no additional charge for backup storage up to 100% of your consumed database storage for an active DB Instance.

Cross-Region Snapshots

- Are a copy of a database snapshot stored in a different AWS Region.
- Provide a backup for disaster recovery.
- Can be used as a base for migration to a different region.

Cross-region snapshot copy is available for all Amazon RDS engines. You can copy snapshots of any size. Copies can be moved between any of the public AWS Regions, and you can copy the same snapshot to multiple regions simultaneously by initiating more than one transfer. There is no charge for the copy operation itself; you pay only for the data transfer out of the source region and for the data storage in the destination region.



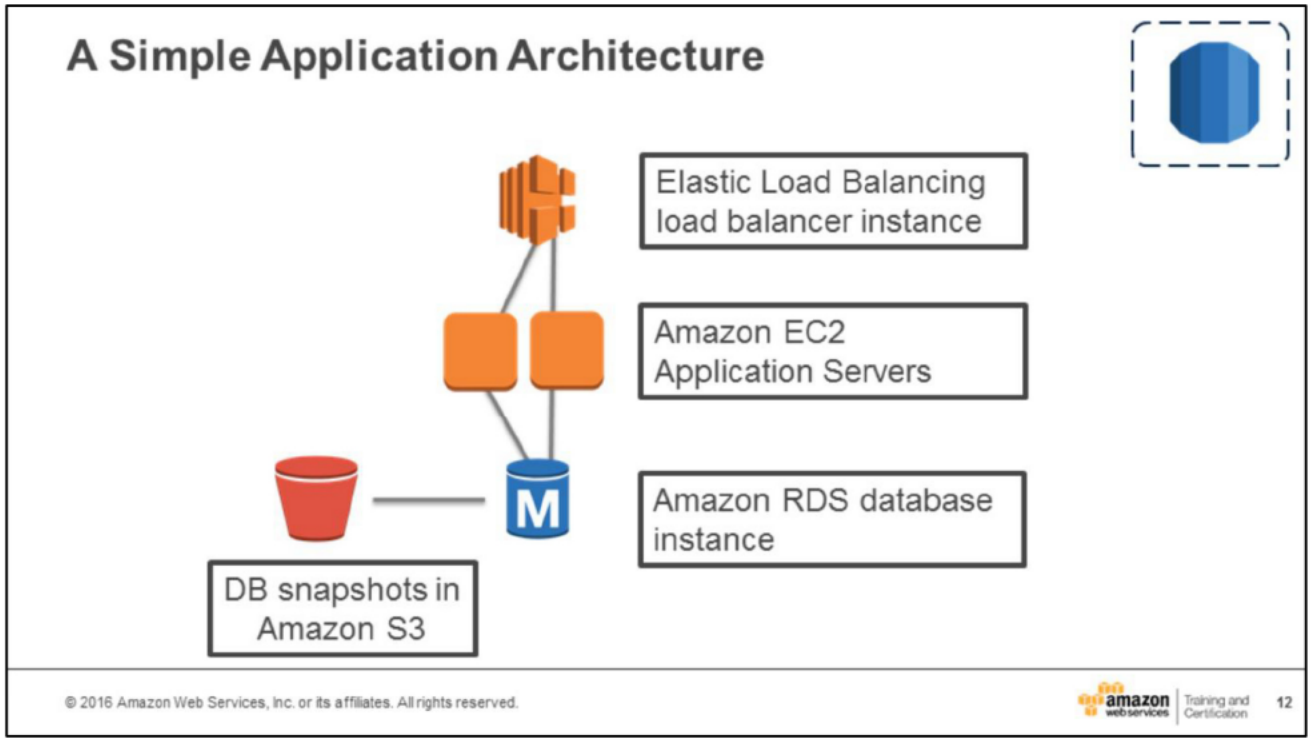
You can manage access to your Amazon Relational Database Service (Amazon RDS) resources and your databases on a DB instance. The method you use to manage access depends on what type of task the user needs to perform with Amazon RDS.

- Run your DB instance in an Amazon Virtual Private Cloud (VPC) for the greatest possible network access control.
- Use AWS Identity and Access Management (IAM) policies to assign permissions that determine who is allowed to manage RDS resources. For example, you can use IAM to determine who is allowed to create, describe, modify, and delete DB instances, tag resources, or modify DB security groups.
- Use security groups to control which IP addresses or EC2 instances can connect to your databases on a DB instance. When you first create a DB instance, its firewall prevents any database access except through rules specified by an associated security group.
- Use Secure Socket Layer (SSL) connections with DB instances running the MySQL, MariaDB, PostgreSQL, or Microsoft SQL Server database engines.
- Use Amazon RDS encryption to secure your RDS DB instances and snapshots at rest. Amazon RDS encryption uses the industry standard AES-256 encryption algorithm to encrypt your data on the server that hosts your RDS DB instance.
- Use network encryption and transparent data encryption with Oracle DB instances.

- Use the security features of your DB engine to control who can log in to the databases on a DB instance, just as you would if the database was on your local network.

For more information, see:

- Using Amazon RDS with Amazon Virtual Private Cloud (VPC) - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.html
- Setting up an IAM user - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SettingUp.html#CHAP_SettingUp.IAM
- Using SSL with a DB instance - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.SSL.html
- Encrypting Amazon RDS Resources - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html
- Oracle NNE - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.html#Appendix.Oracle.Options.NetworkEncryption
- Oracle TDE - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.html#Appendix.Oracle.Options.AdvSecurity

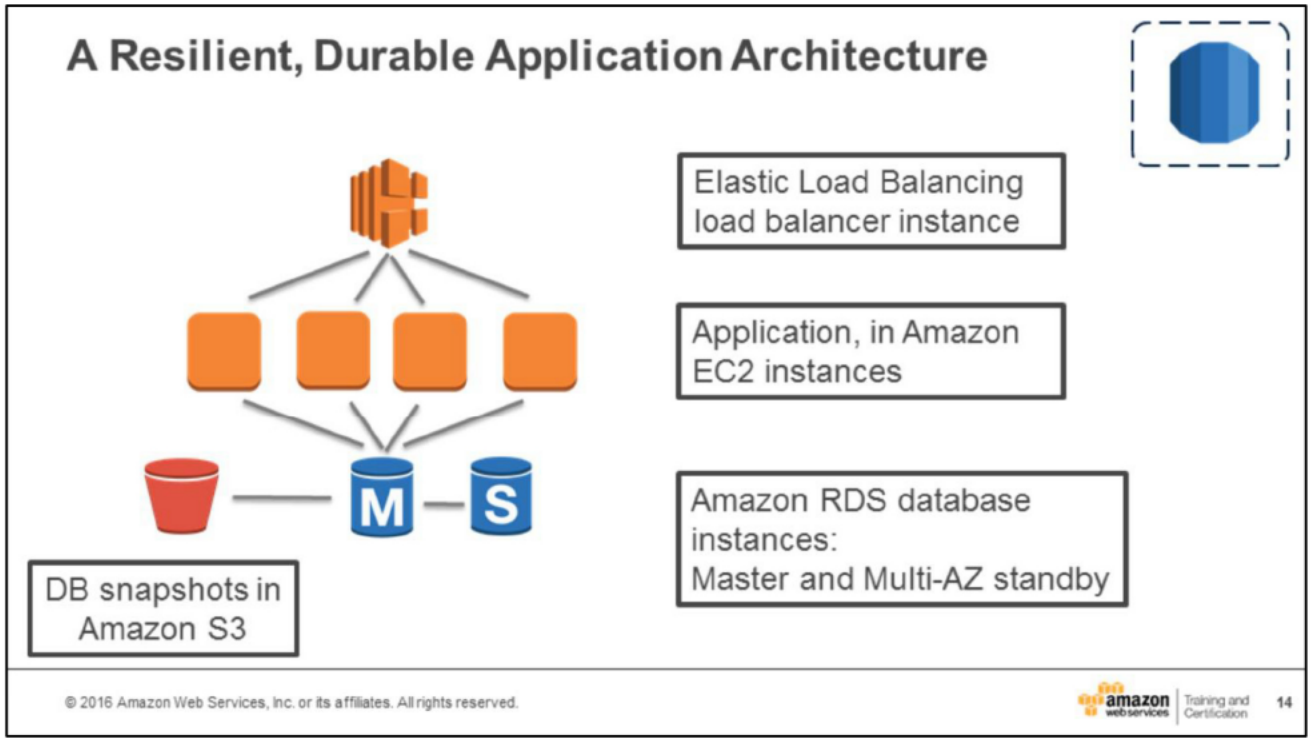


The slide shows a simple application stack with an application running in an Amazon EC2 instance supported by a master database running in an Amazon RDS database instance. Presenting the application behind an Elastic Load Balancer allows for compute resiliency and scaling features such as Auto Scaling and ELB groups to be adopted in the future.

Multi-AZ RDS Deployment

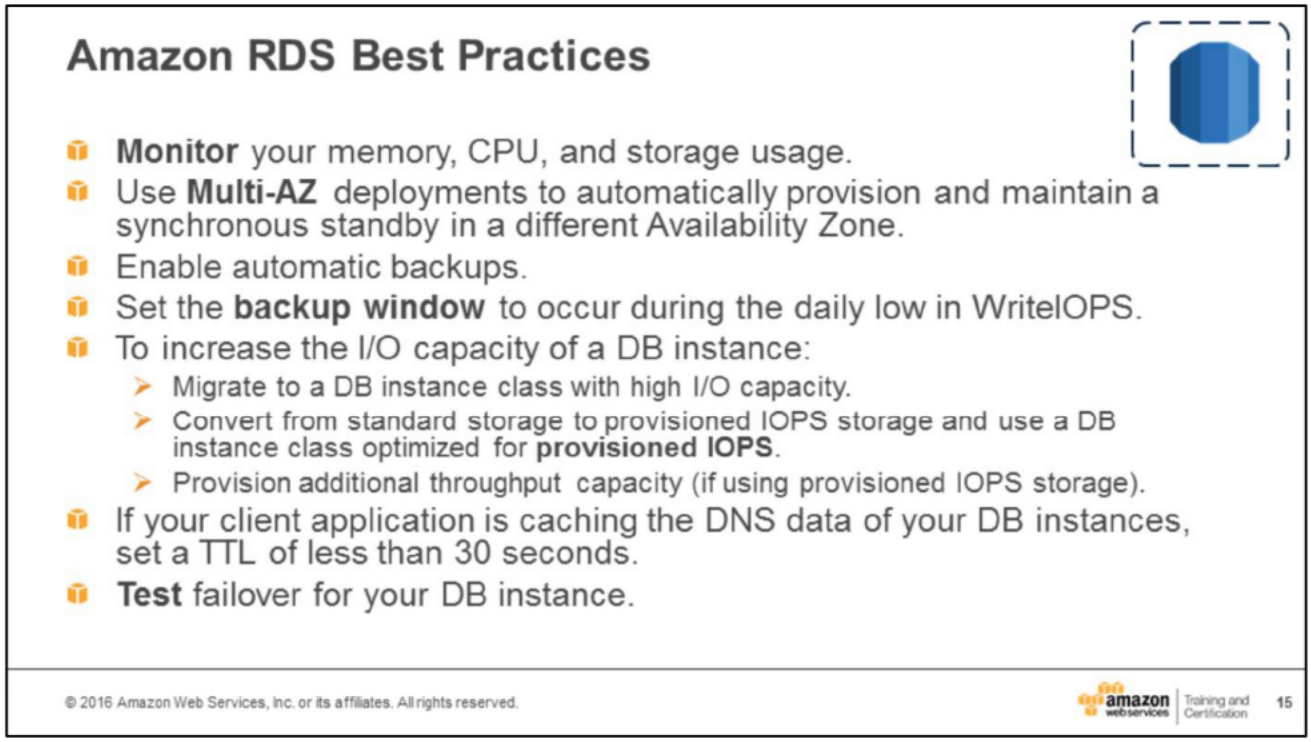
- With Multi-AZ operation, your database is synchronously replicated to another AZ in the same AWS Region.
- Failover automatically occurs to the standby in case of master database failure.
- Planned maintenance is applied first to standby databases.

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable. In case of an infrastructure failure (for example, instance hardware failure, storage failure, or network disruption), Amazon RDS performs an automatic failover to the standby, so that you can resume database operations as soon as the failover is complete. Since the endpoint for your DB instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention.



The slide shows an application stack that uses AWS reliability and durability features. An ELB group of Amazon EC2 instances supports the application logic. The instances use a Multi-AZ Amazon RDS deployment. In the event of infrastructure failure, the database fails over to a standby instance. The application logic retries its database connections, to the same endpoint as before, and the service resumes using the new master. Meanwhile, a new standby is instantiated.

In addition to Amazon RDS's automatic backups, the database snapshot feature is used to ensure that backups are durably retained. You can create a new database instance from a database snapshot whenever you want.

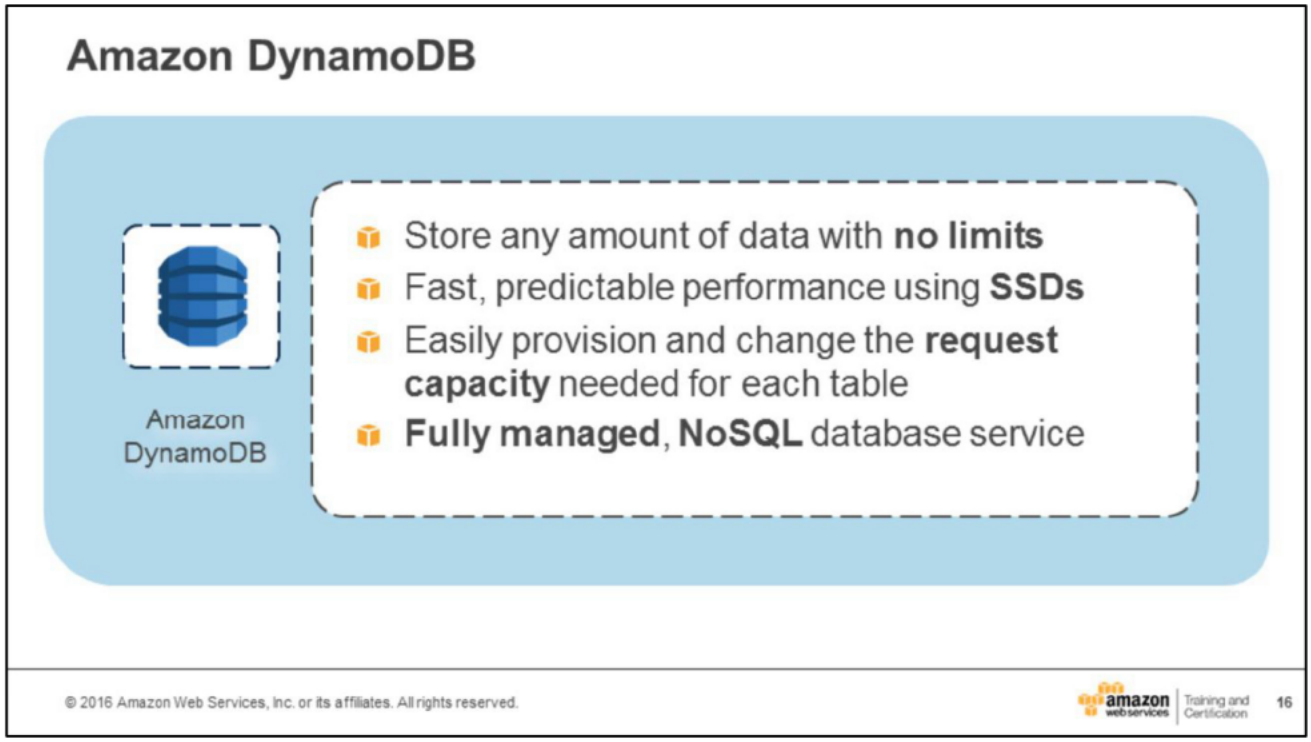


- Monitor your memory, CPU, and storage usage. Amazon CloudWatch can be set up to notify you when usage patterns change or when you approach the capacity of your deployment, so that you can maintain system performance and availability.
- Use Multi-AZ deployments to automatically provision and maintain a synchronous standby replica in a different Availability Zone.
- Enable automatic backups and set the backup window to occur during the daily low in WriteIOPS.
- On a MySQL DB instance:
    - Do not create more than 10,000 tables using provisioned IOPS (IOPS are input/output operations per second) or 1000 tables using standard storage. Large numbers of tables will significantly increase database recovery time after a failover or database crash. If you need to create more tables than recommended, set the innodb_file_per_table parameter to 0.
    - Avoid tables in your database growing too large. Underlying file system constraints restrict the maximum size of a MySQL table file to 2 TB. Instead, partition your large tables so that file sizes are well under the 2 TB limit. This approach can also improve performance and recovery time.

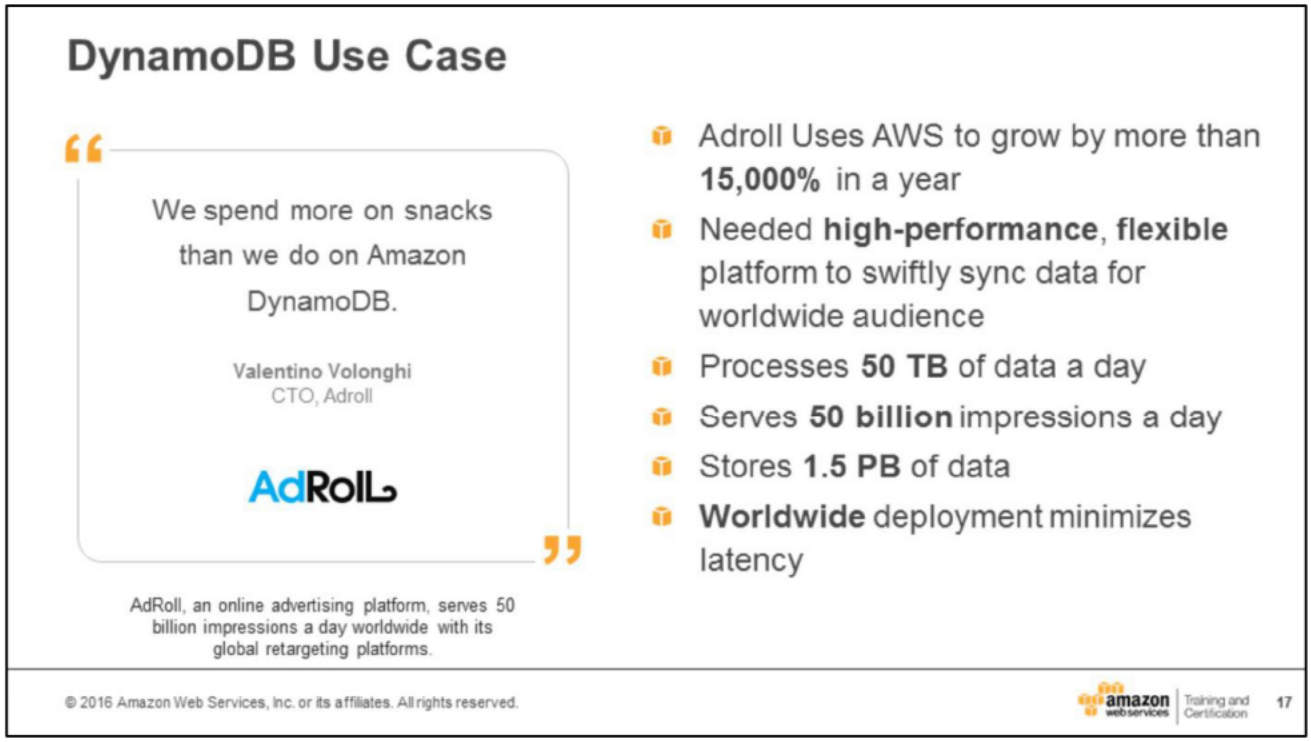
- If your database workload requires more I/O than you have provisioned, recovery after a failover or database failure will be slow. To increase the I/O capacity of a DB instance, do any or all of the following:
    - Migrate to a DB instance class with high I/O capacity.
    - Convert from standard storage to provisioned IOPS storage, and use a DB instance class that is optimized for provisioned IOPS.
- If you are already using provisioned IOPS storage, provision additional throughput capacity.
- If your client application is caching the DNS data of your DB instances, set a time to live (TTL) of less than 30 seconds. Because the underlying IP address of a DB instance can change after a failover, caching the DNS data for an extended time can lead to connection failures if your application tries to connect to an IP address that no longer is in service.
- Test failover for your DB instance to understand how long the process takes for your use case and to ensure that the application that accesses your DB instance can automatically connect to the new DB instance after failover.

For more information, see:

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.html

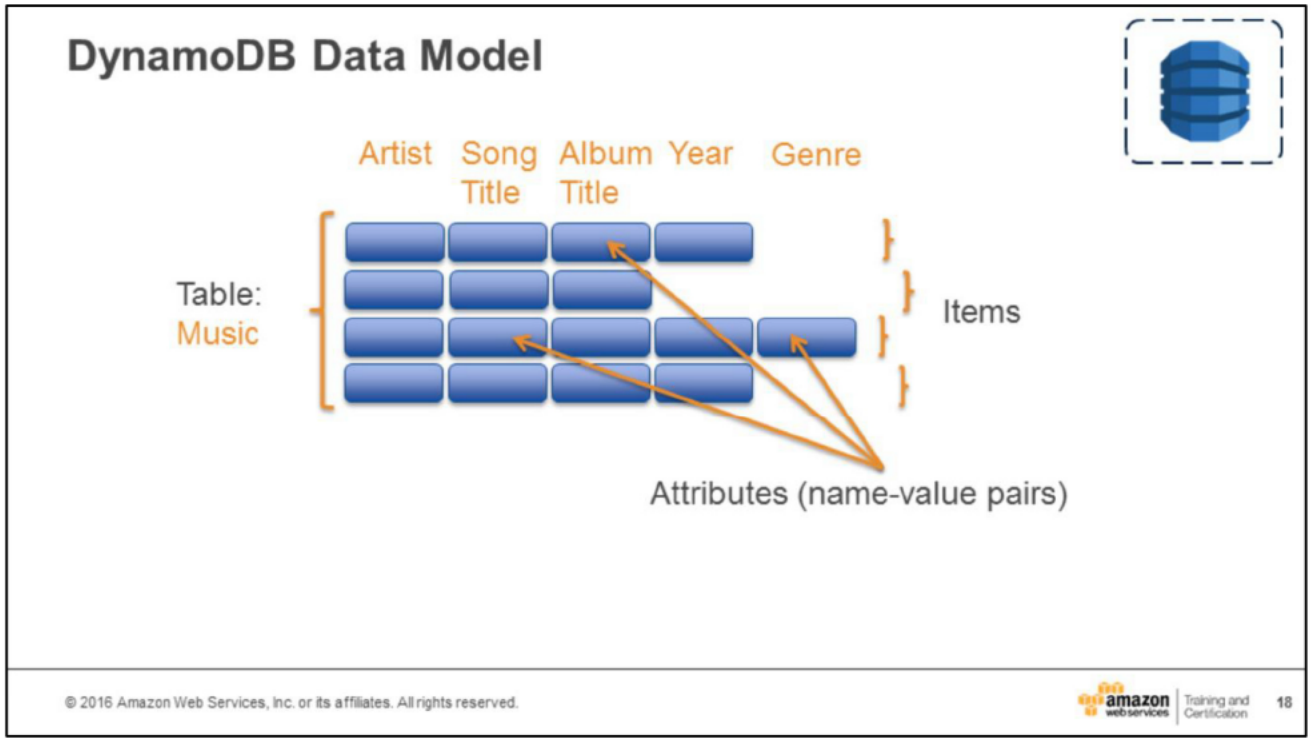


Amazon DynamoDB is a fully-managed NoSQL database service that offers high performance, predictable throughput and low cost. It is easy to set up, operate, and scale. With Amazon DynamoDB, you can start small, specify the throughput and storage you need, and easily scale your capacity requirements in seconds, as needed. It automatically partitions data over a number of servers to meet your requested capacity. In addition, Amazon DynamoDB automatically replicates your data synchronously across multiple Availability Zones within an AWS Region to ensure high availability and data durability.



For more information, see:

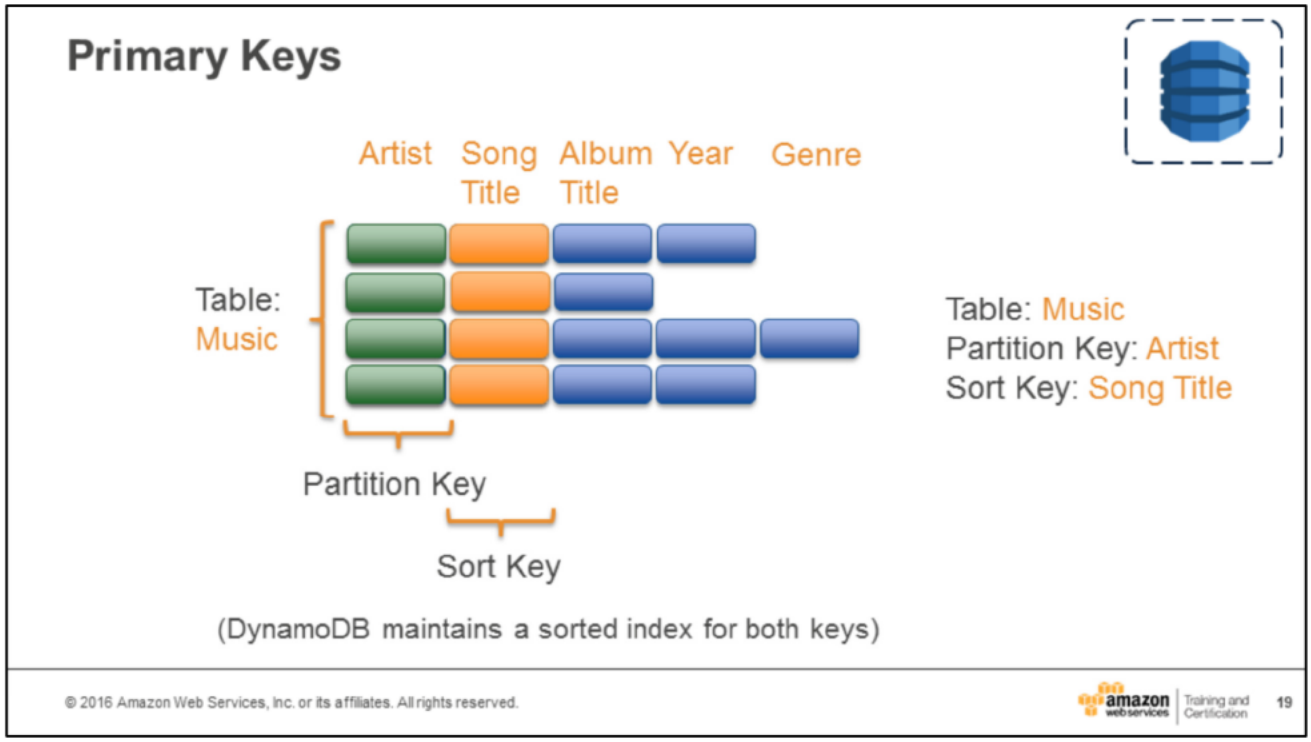
https://aws.amazon.com/solutions/case-studies/adroll/



In Amazon DynamoDB, a *table* is a collection of *items* and each item is a collection of *attributes*. Each attribute in an item is a name-value pair. An attribute can be a scalar (single-valued), a JSON document, or a set.

For more information, see:

http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.html



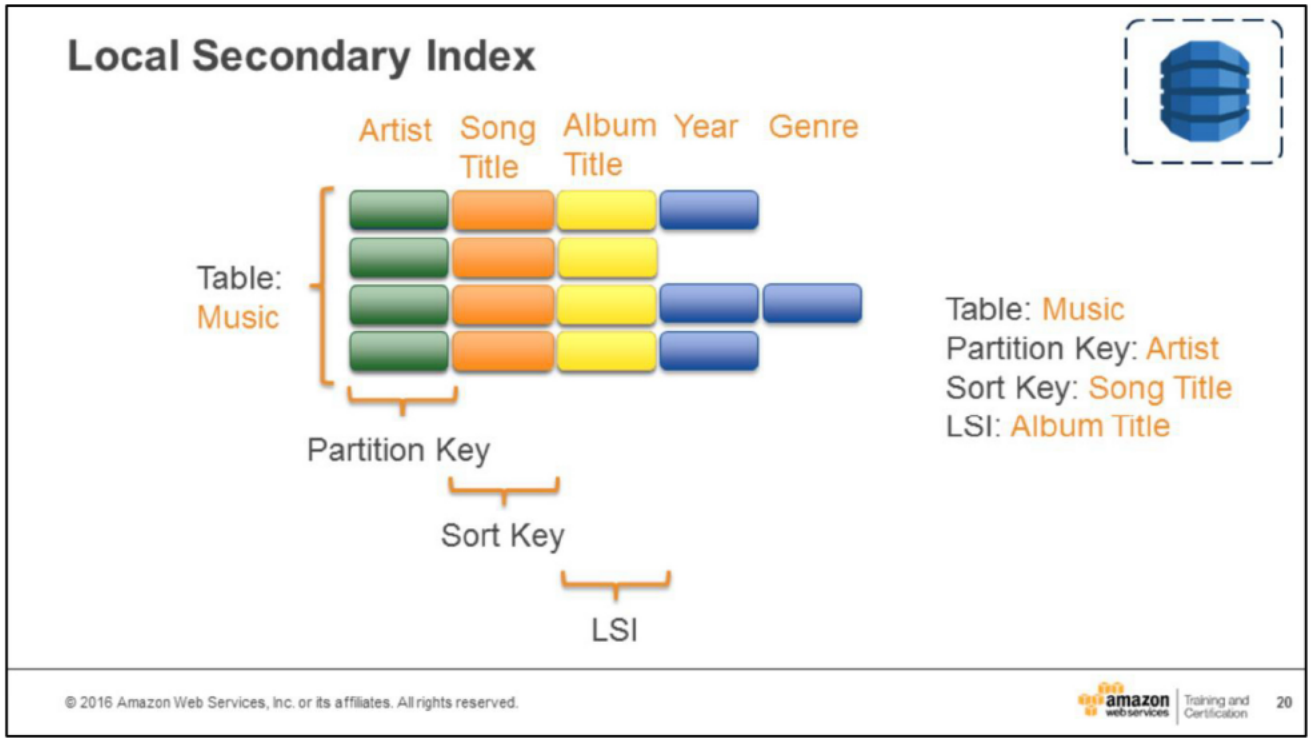
When you create a table, in addition to the table name, you must specify the primary key of the table. As in other databases, a primary key in DynamoDB uniquely identifies each item in the table, so that no two items can have the same key. When you add, update, or delete an item in the table, you must specify the primary key attribute values for that item.

DynamoDB supports two different kinds of primary keys:

1. Partition Key—A simple primary key, composed of one attribute known as the partition key. DynamoDB uses the partition key's value as input to an internal hash function; the output from the hash function determines the partition where the item is stored. No two items in a table can have the same partition key value.
2. Partition Key and Sort Key—A composite primary key, composed of two attributes. The first attribute is the partition key, and the second attribute is the sort key. DynamoDB uses the partition key value as input to an internal hash function; the output from the hash function determines the partition where the item is stored. All items with the same partition key are stored together, in sorted order by sort key value. It is possible for two items to have the same partition key value, but those two items must have different sort key values.

For more information, see:

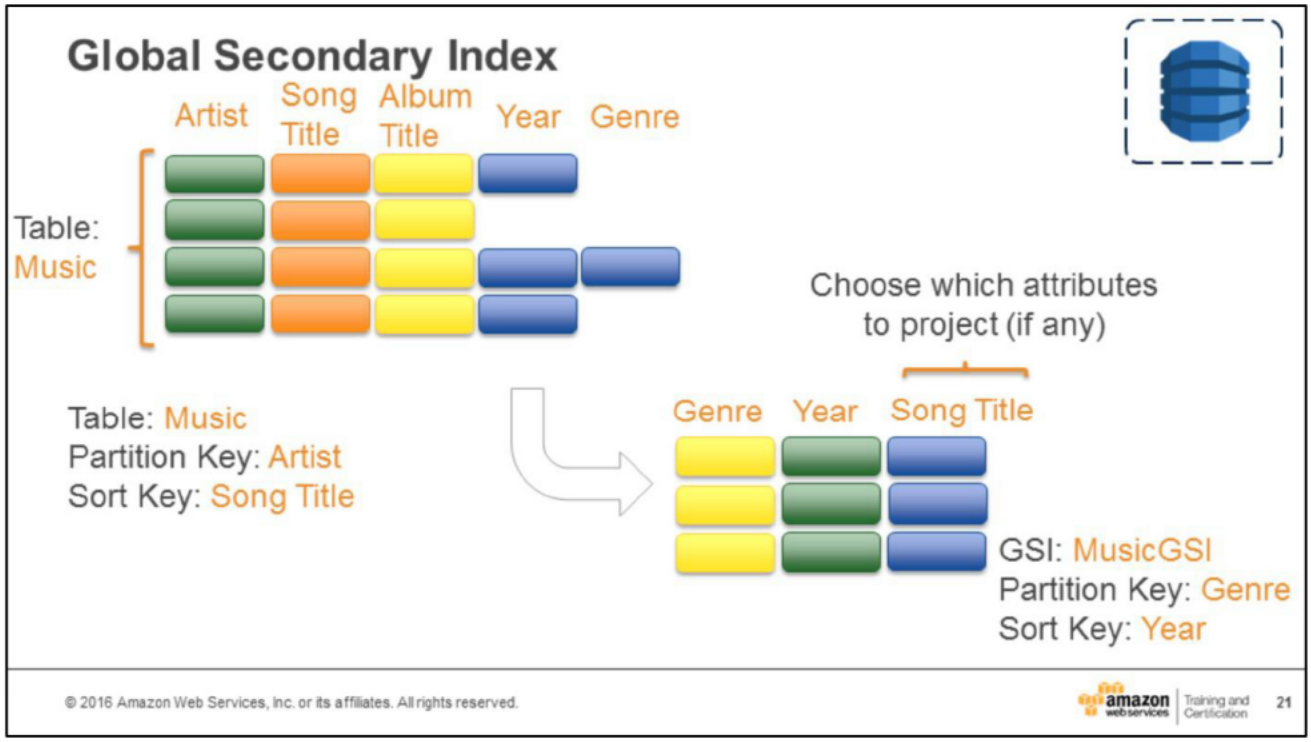
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.html



If you want to read the data using non-key attributes, you can use a secondary index to do this. A local secondary index is an index that has the same partition key as the table, but a different sort key.

For more information, see:

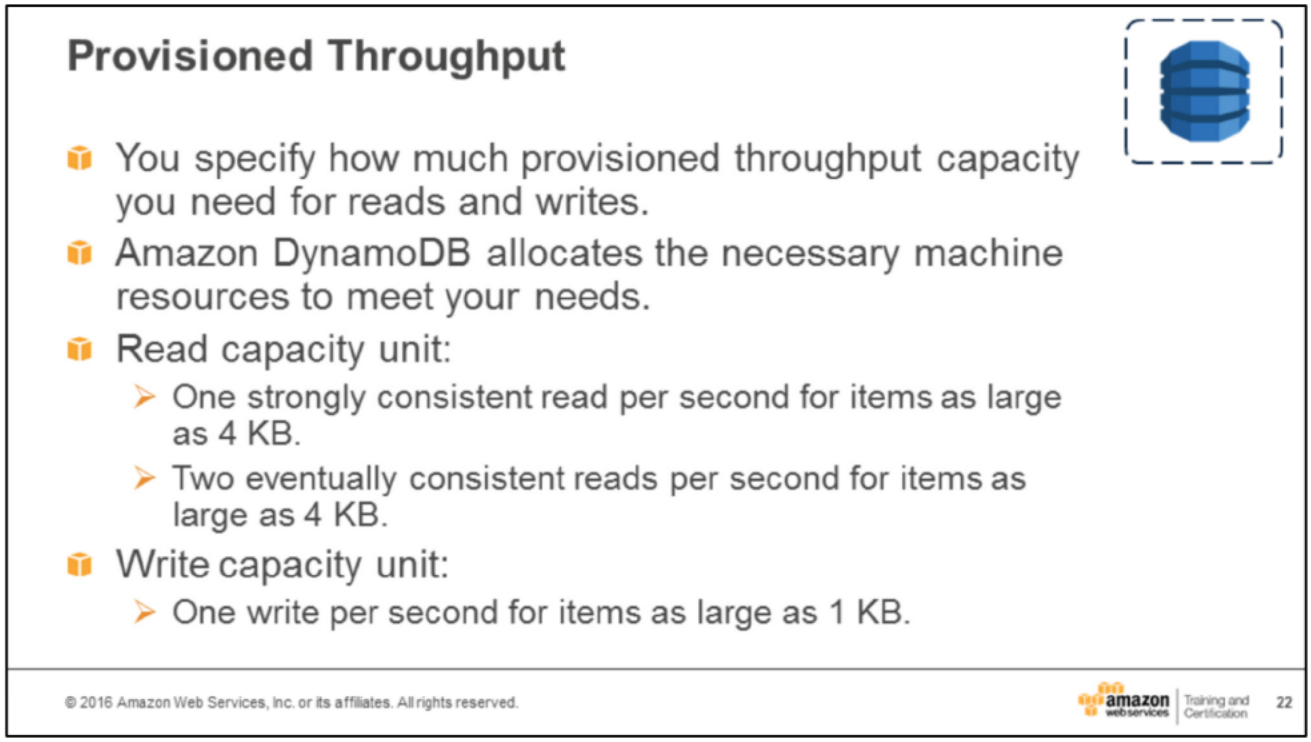
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LSI.html



A Global secondary index is an index with a partition key and sort key that can be different from those on the table. They can be thought of as “pivot charts” for your table.

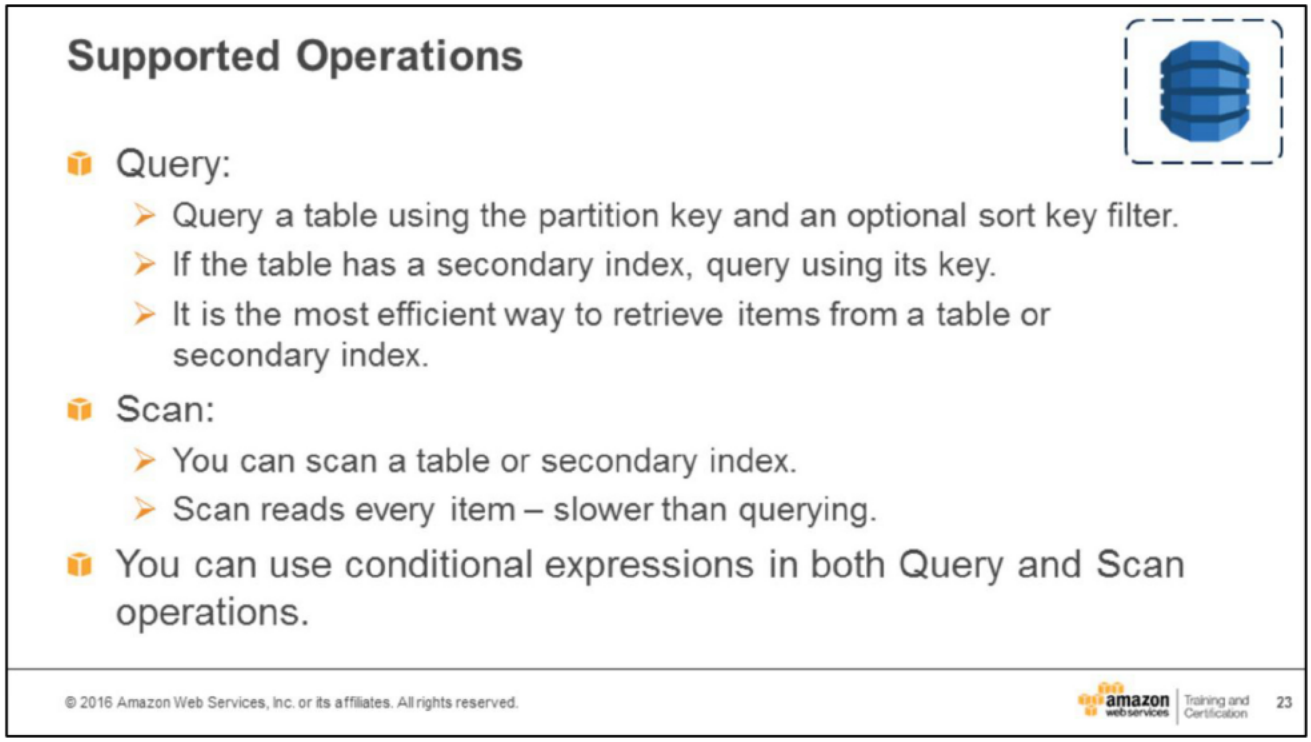
For more information, see:

http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html



When you create or update a table, you specify how much provisioned throughput capacity you need for reads and writes. Amazon DynamoDB will allocate the necessary machine resources to meet your throughput needs while ensuring consistent, low-latency performance.

A unit of *read capacity* represents one strongly consistent read per second (or two eventually consistent reads per second) for items as large as 4 KB. A unit of *write capacity* represents one write per second for items as large as 1 KB.



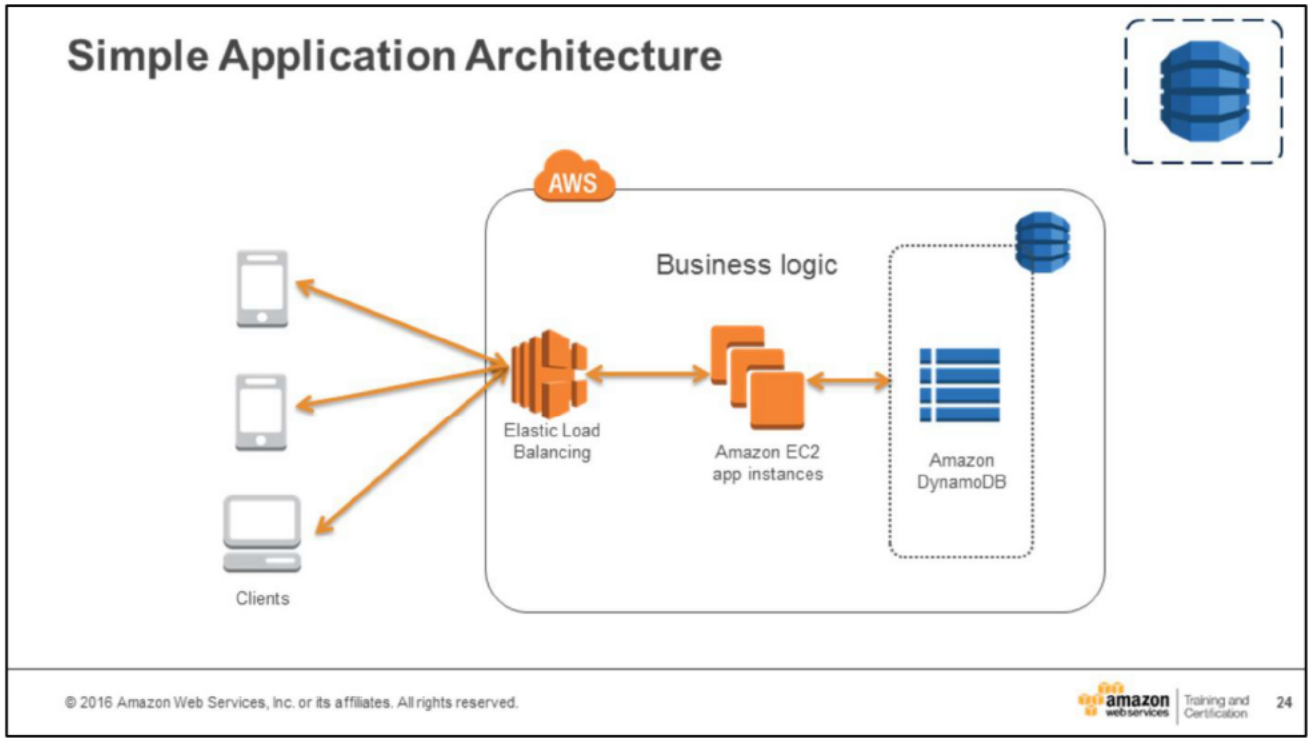
The Query operation enables you to query a table using the partition key and an optional sort key filter. If the table has a secondary index, you can also Query the index using its key. You can query only tables that have a composite primary key (partition key and sort key). You can also query any secondary index on such tables. Query is the most efficient way to retrieve items from a table or a secondary index.

Amazon DynamoDB also supports a Scan operation, which you can use on a table or a secondary index. The Scan operation reads every item in the table or secondary index. For large tables and secondary indexes, a Scan can consume a large amount of resources; for this reason, we recommend that you design your applications so that you can use the Query operation mostly, and use Scan only where appropriate.

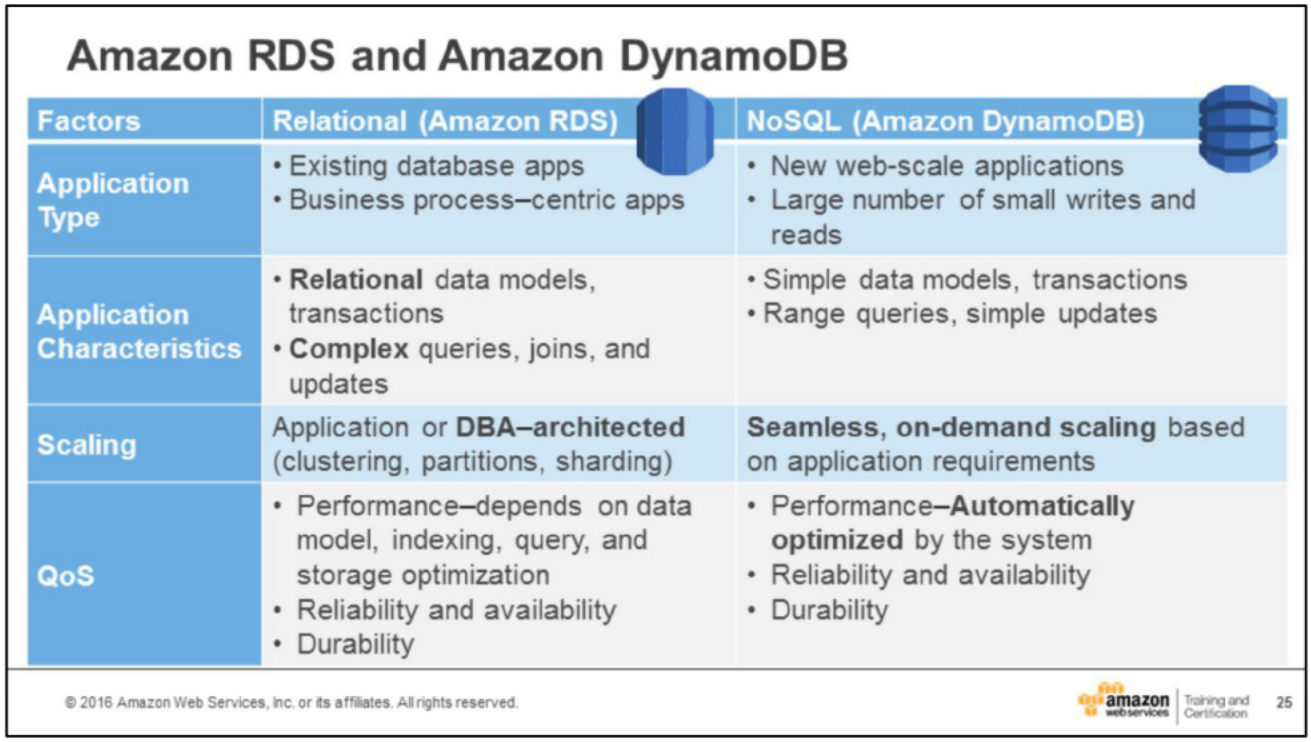
You can use conditional expressions in both the Query and Scan operations to control which items are returned.

For more information, see:

Query and Scan Operations in DynamoDB - http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/QueryAndScan.html



The slide shows a simple application architecture using Amazon DynamoDB to store data processed by applications on Amazon EC2 instances.



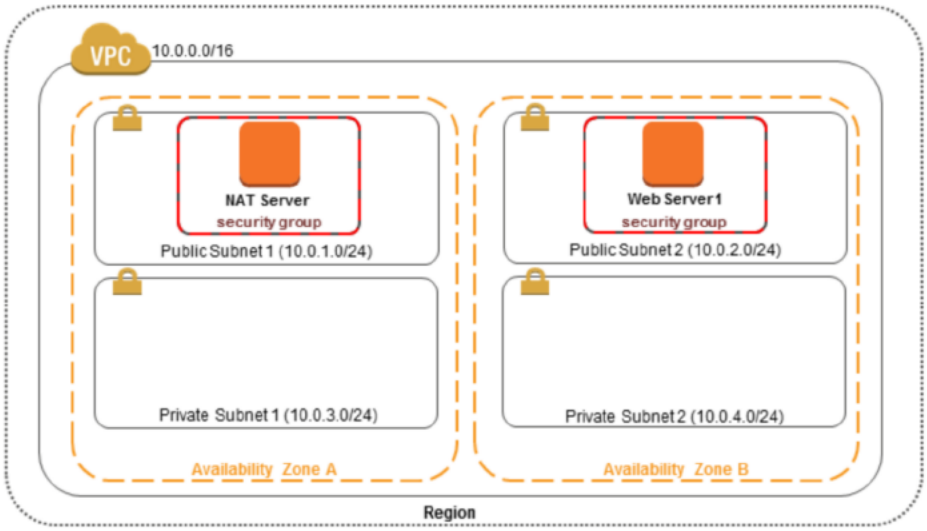
One type does not fit all. The choice depends on several factors. You can use both relational and NoSQL databases in one application, depending on requirements. This table provides a side-by-side comparison of relational or non-relational database.

Database Considerations

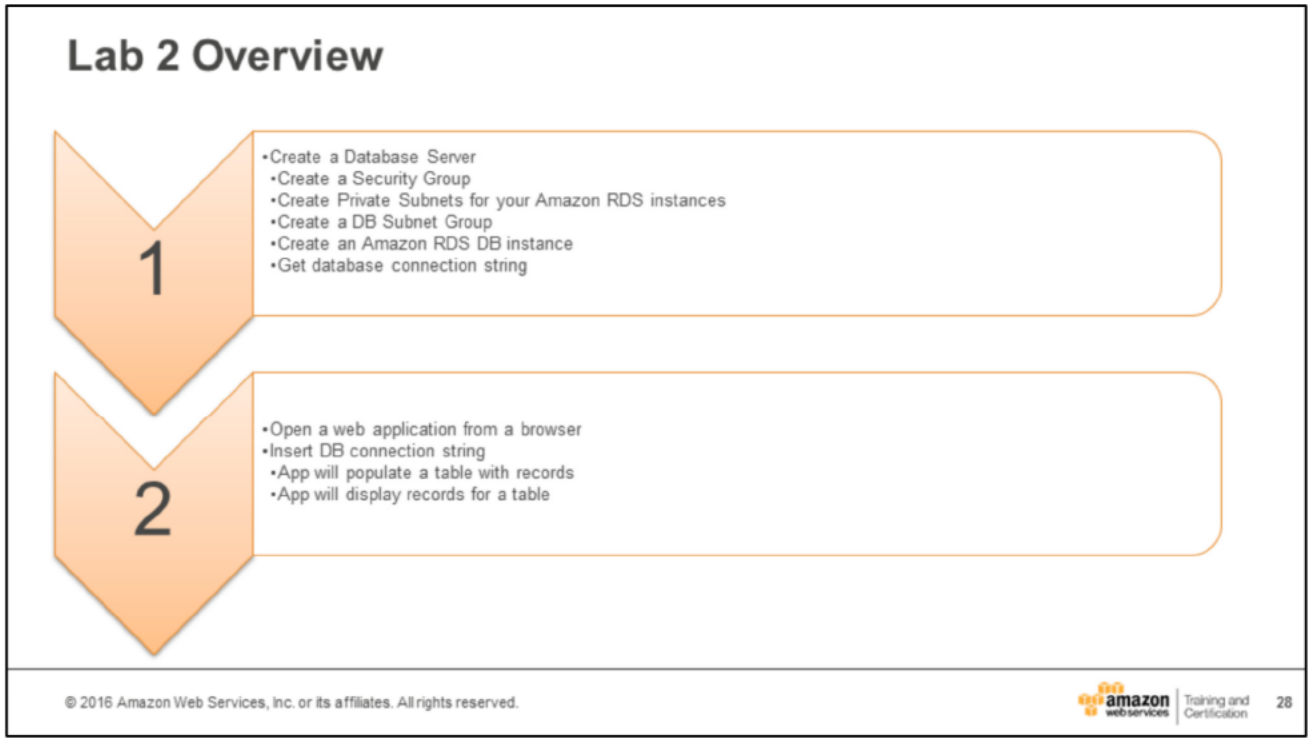
If You Need	Consider Using
A relational database service with minimal administration	**Amazon RDS** • Choice of Amazon Aurora, MySQL, MariaDB, Microsoft SQL Server, Oracle, or PostgreSQL database engines • Scale compute and storage • Multi-AZ availability
A fast, highly scalable NoSQL database service	**Amazon DynamoDB** • Extremely fast performance • Seamless scalability and reliability • Low cost
A database you can manage on your own	Your choice of **AMIs** on Amazon EC2 and Amazon EBS that provide scale compute and storage, complete control over instances, and more.

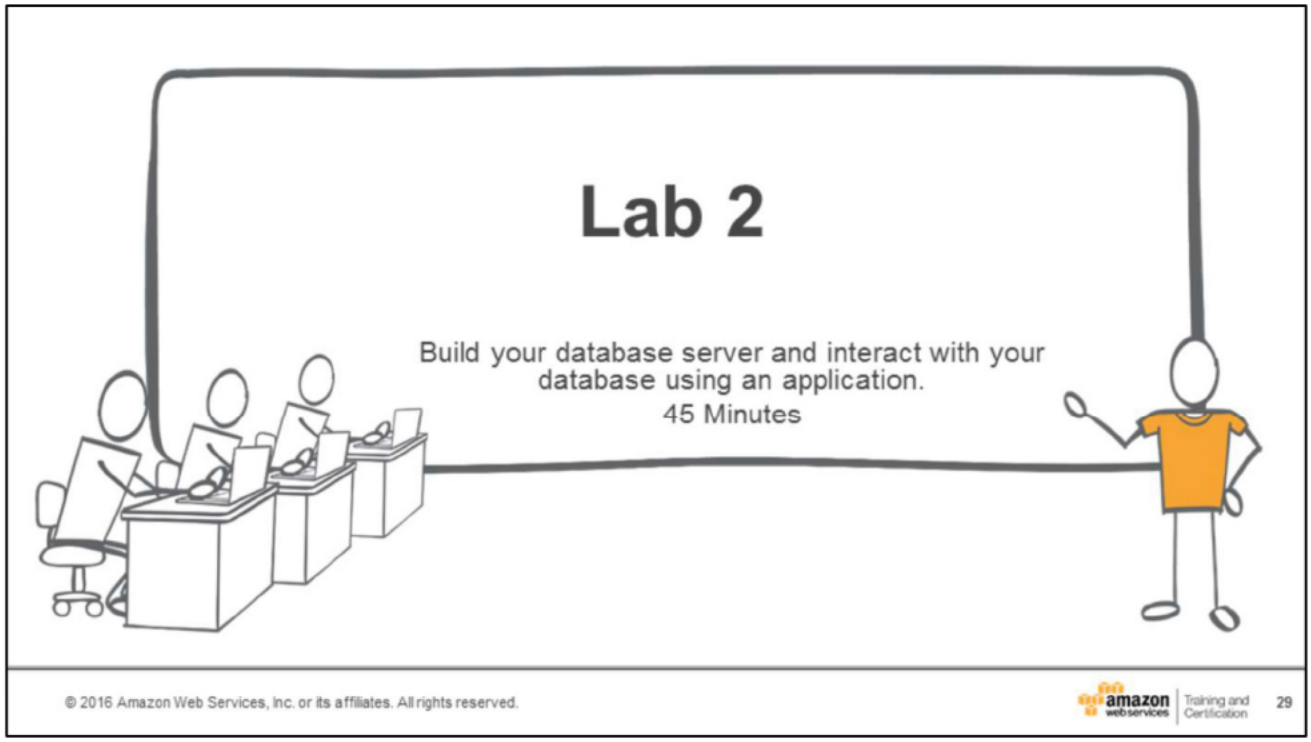
AWS provides a number of database alternatives for developers. You can run fully managed relational and NoSQL services, or you can operate your own database in the cloud on Amazon EC2 and Amazon EBS. If you need a relational database service with minimal administration, consider using Amazon RDS. If you need a fast, highly scalable NoSQL database service, consider using Amazon DynamoDB. If you need a relational database you can manage on your own, consider using your choice of relational AMIs.

What You're Starting With – Lab 1 Review

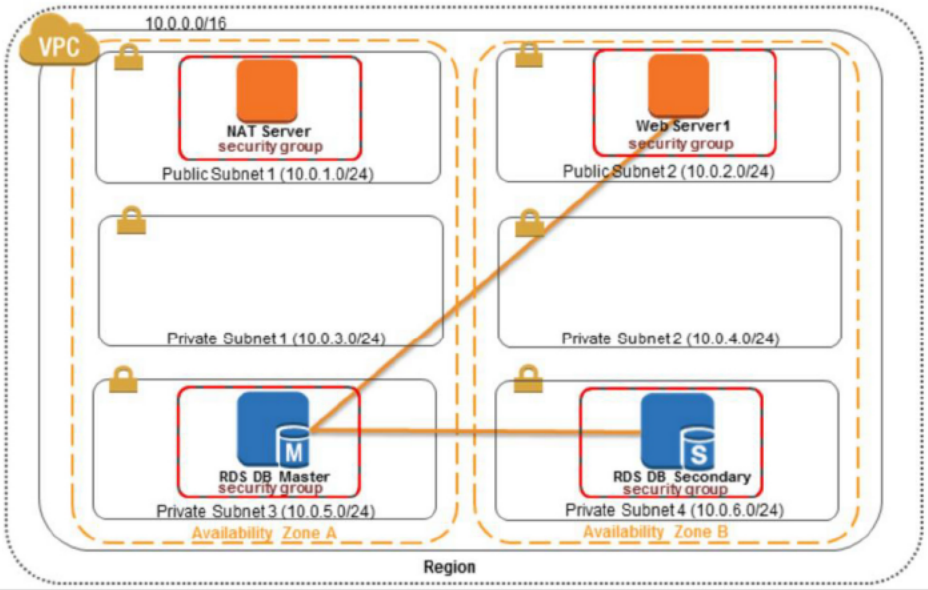


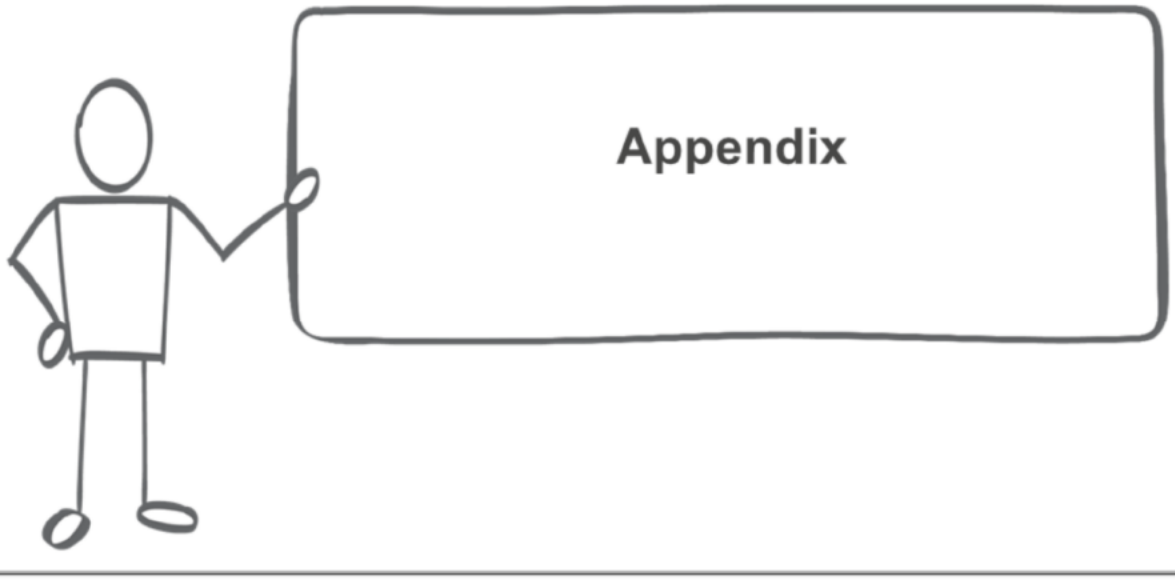
amazon web services | Training and Certification 27

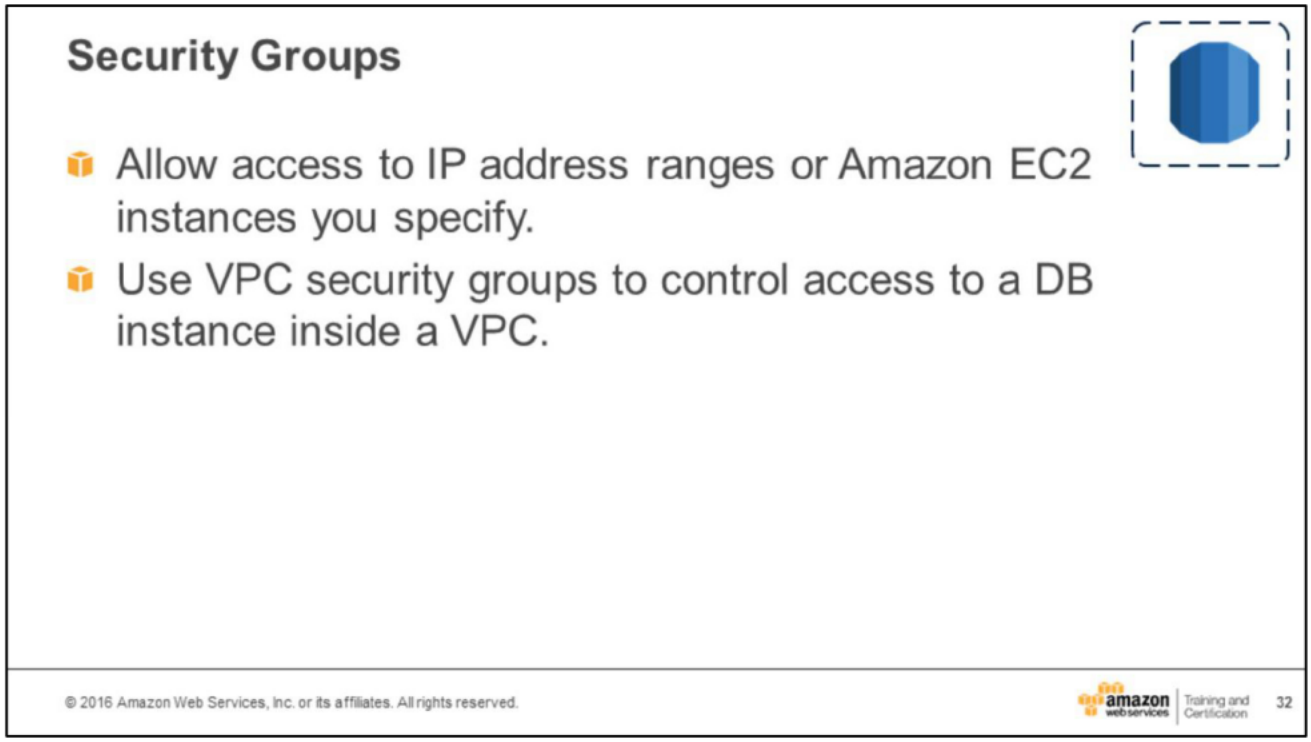



Lab 2
Build your database server and interact with your database using an application.
45 Minutes
© 2016 Amazon Web Services, Inc. or its affiliates. All rights reserved.
amazon web services | Training and Certification
29

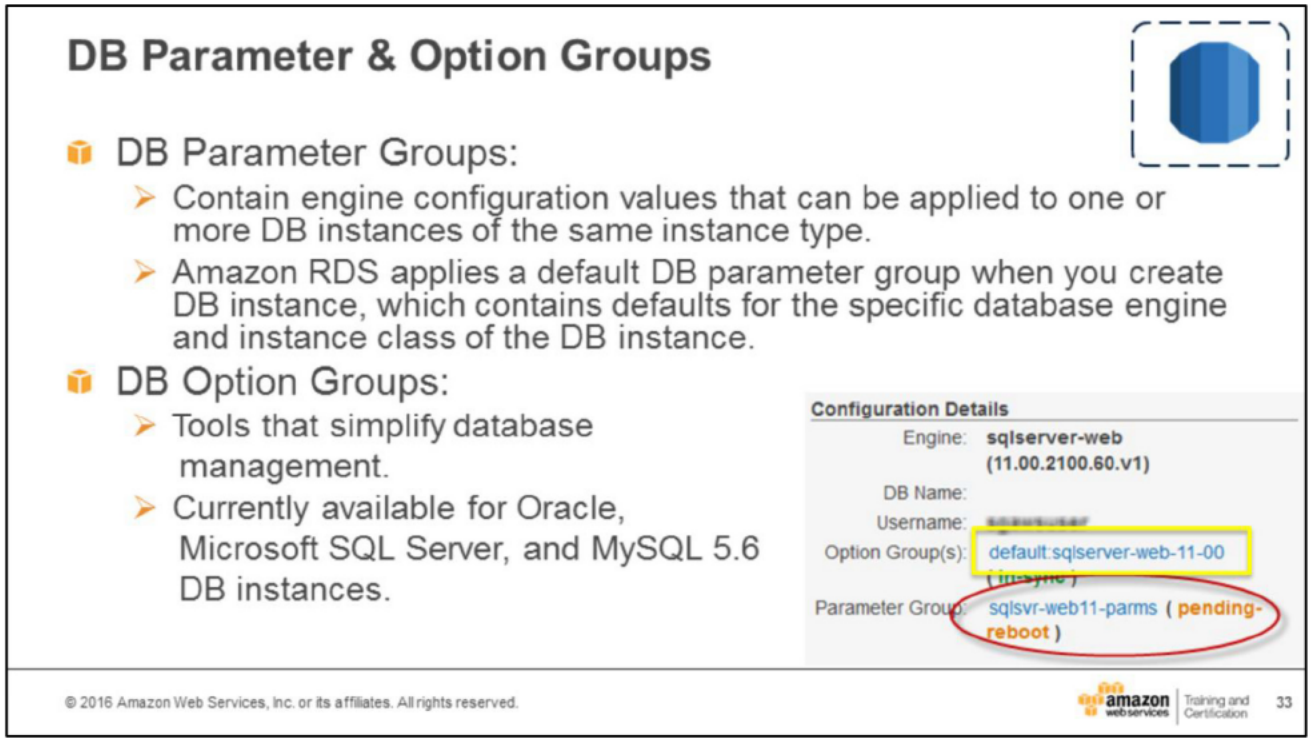
Lab 2 – What You Created







A VPC security group controls access to a DB instance inside a VPC.

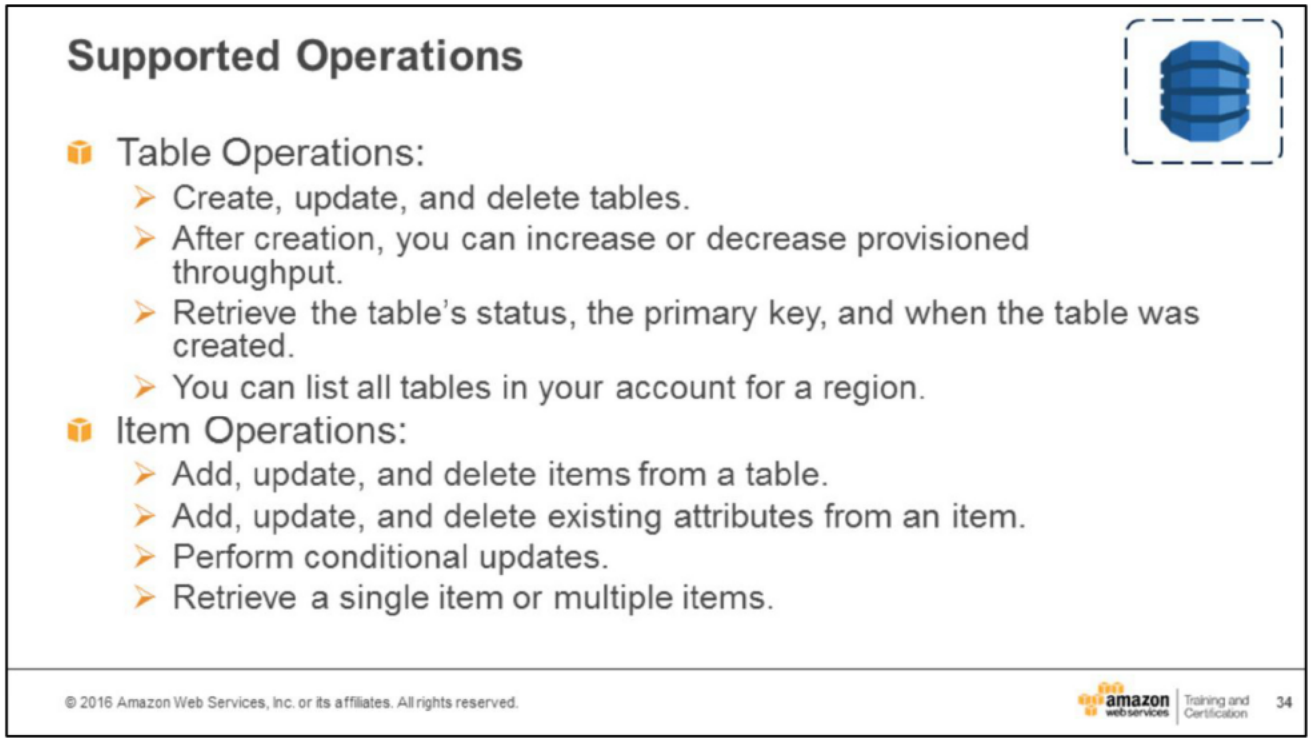


You manage the configuration of a DB engine by using a DB parameter group. A DB parameter group contains engine configuration values that can be applied to one or more DB instances of the same instance type. Amazon RDS applies a default DB parameter group if you don't specify a DB parameter group when you create a DB instance. The default group contains defaults for the specific database engine and instance class of the DB instance.

Some DB engines offer tools that simplify managing your databases and making the best use of your data. Amazon RDS makes such tools available through option groups.

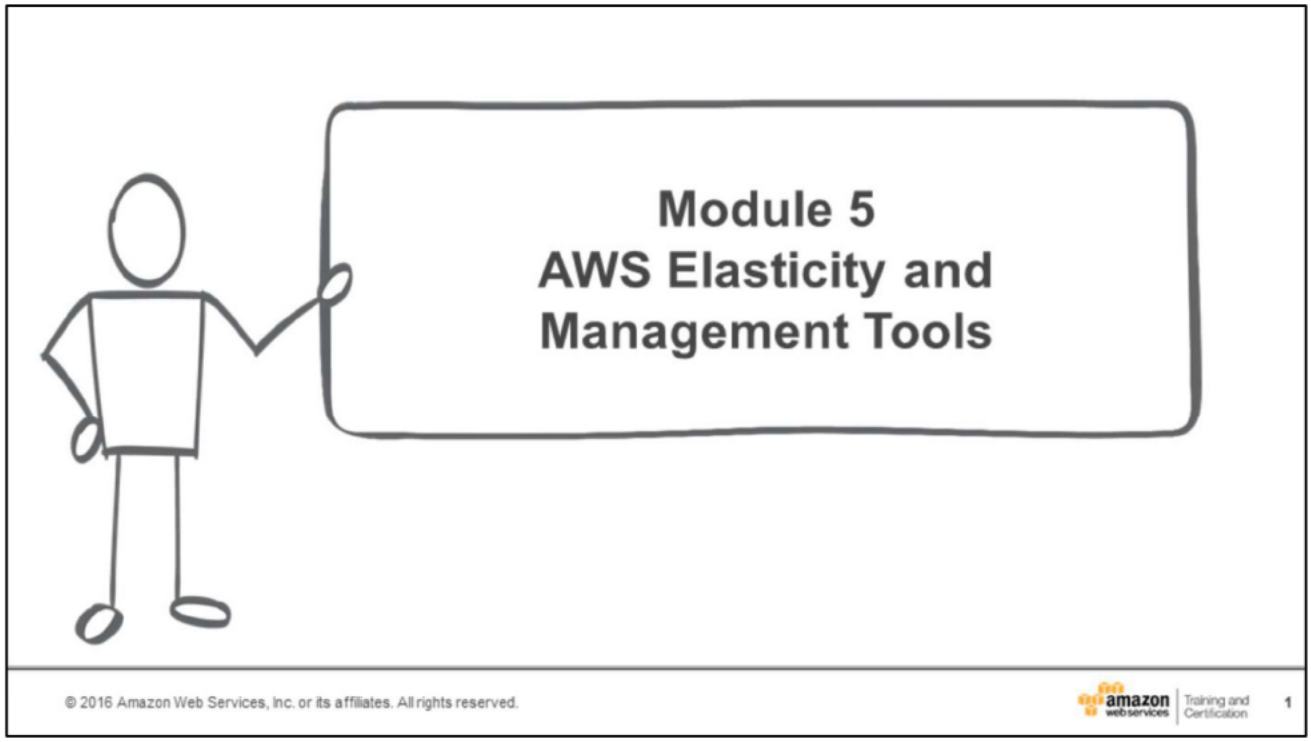
For more information, see:

- Oracle Database Engine option groups - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.html
- SQL Server option groups - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.SQLServer.Options.html
- MySQL option groups - http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.MySQL.Options.html

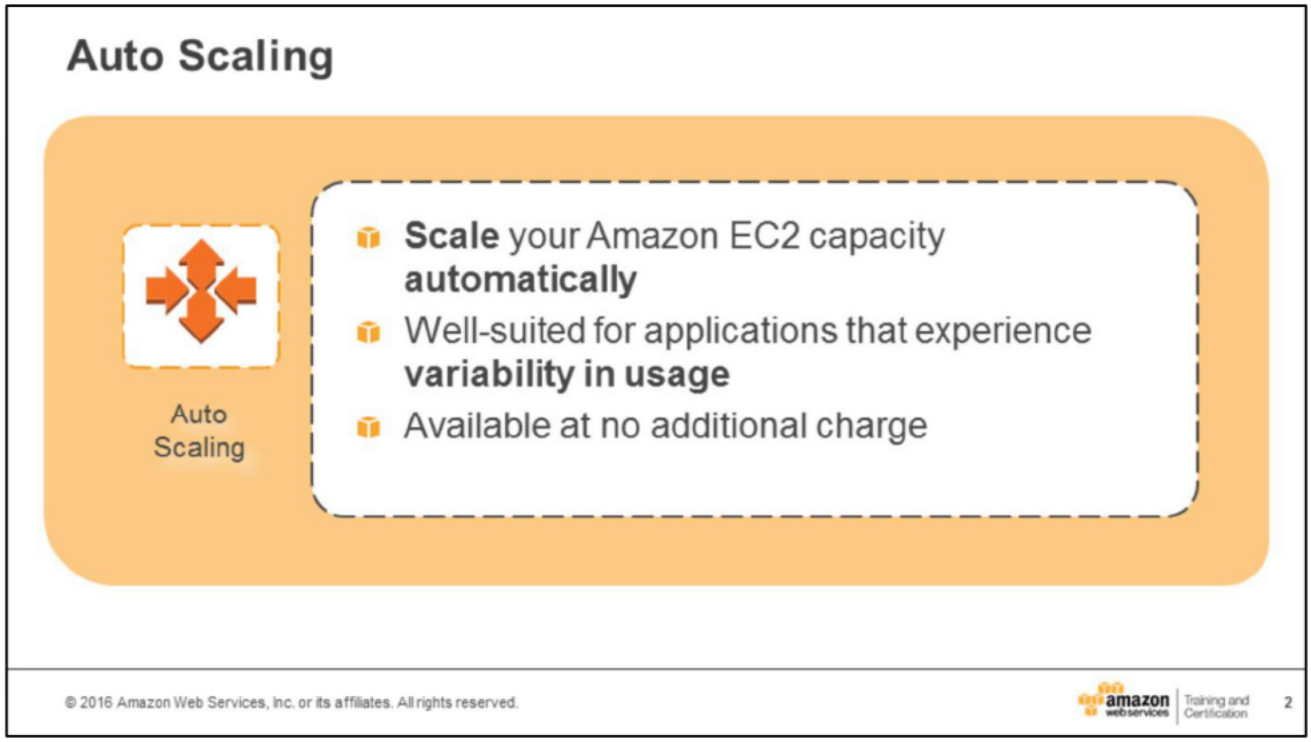


Amazon DynamoDB provides operations to create, update, and delete tables. After the table is created, you can use the UpdateTable operation to increase or decrease a table's provisioned throughput. Amazon DynamoDB also supports an operation to retrieve table information (the DescribeTable operation), including the current status of the table, the primary key, and when the table was created. The ListTables operation enables you to get a list of tables in your account in the region of the endpoint you are using to communicate with Amazon DynamoDB.

Item operations enable you to add, update and delete items from a table. You can update existing attribute values, add new attributes, and delete existing attributes from an item. You can also perform conditional updates. For example, if you are updating a price value, you can set a condition so the update happens only if the current price is $15. You can retrieve a single item or multiple items from a table.



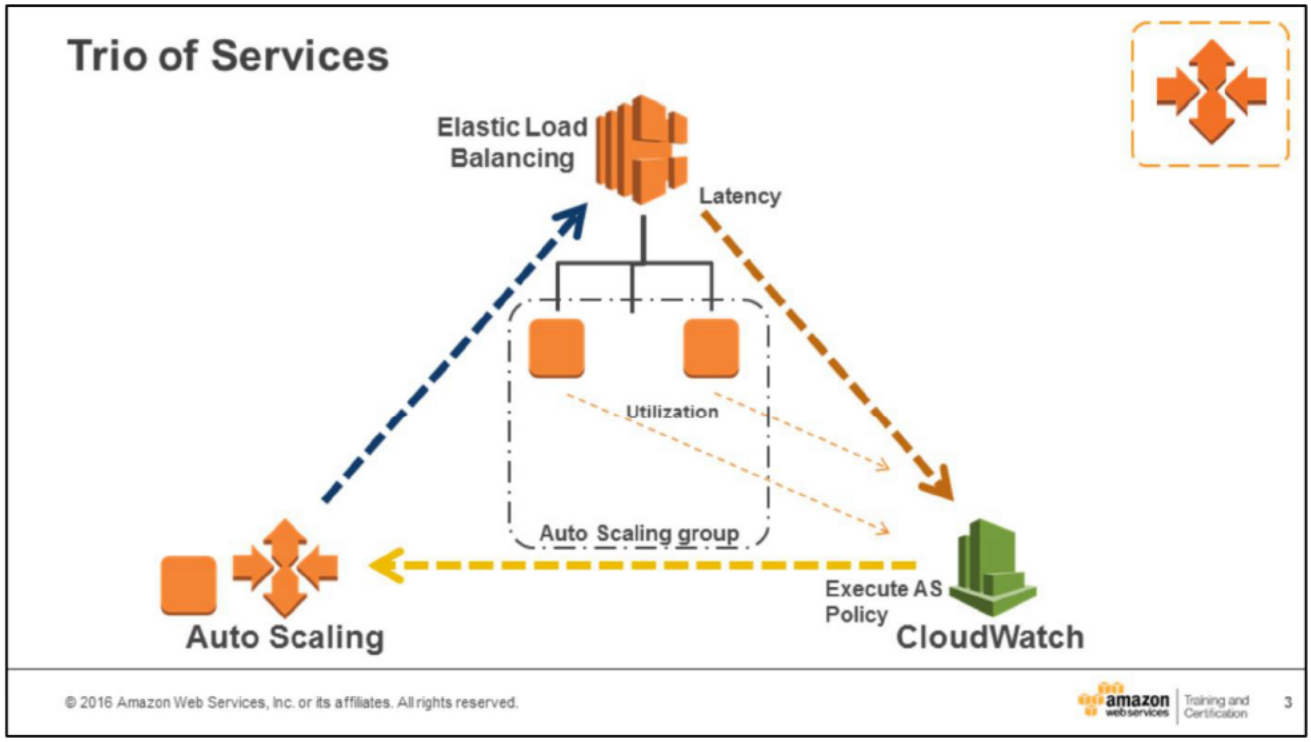
This module describes the fundamental elements of AWS Management Tools.



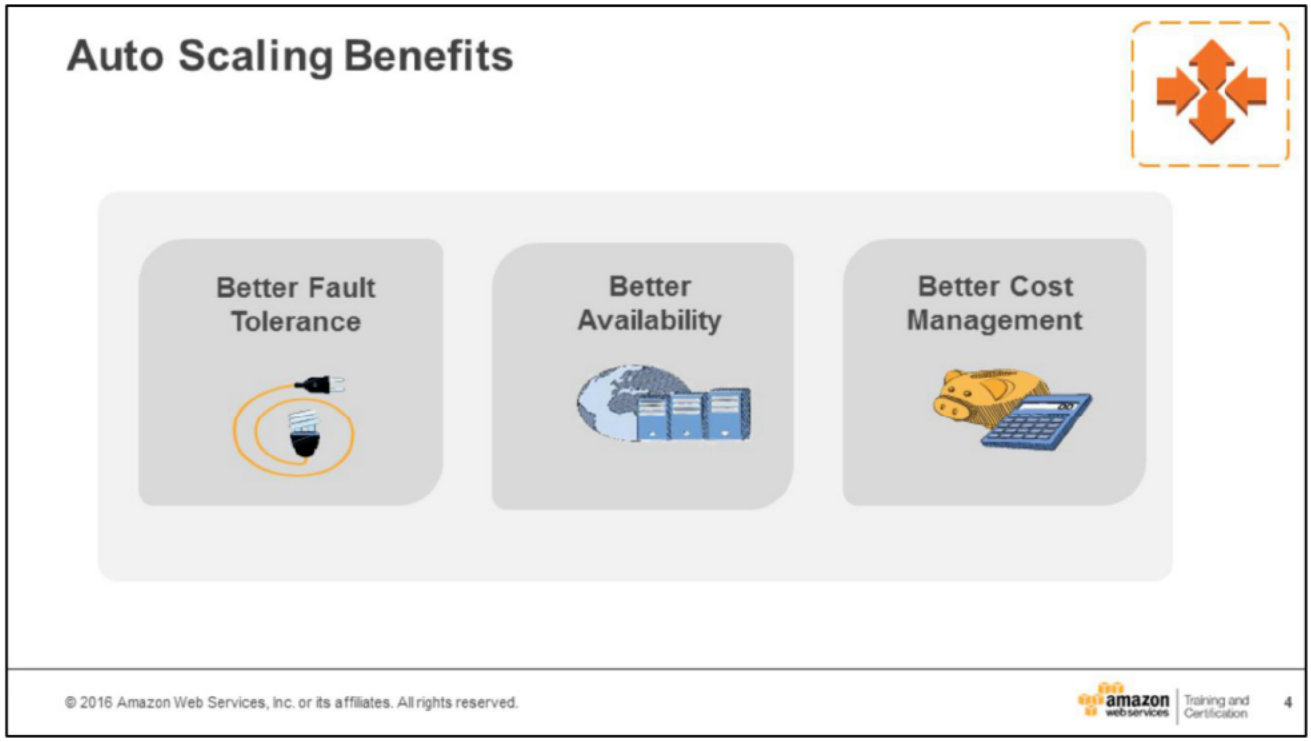
Understand Auto Scaling concepts including:

- Launch Configurations
- Auto Scaling Groups
- Scaling Plans
- Auto Scaling Lifecycle
- Auto Scaling Limits

Auto Scaling helps you ensure that you have the correct number of EC2 instances available to handle the load for your application. Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage.

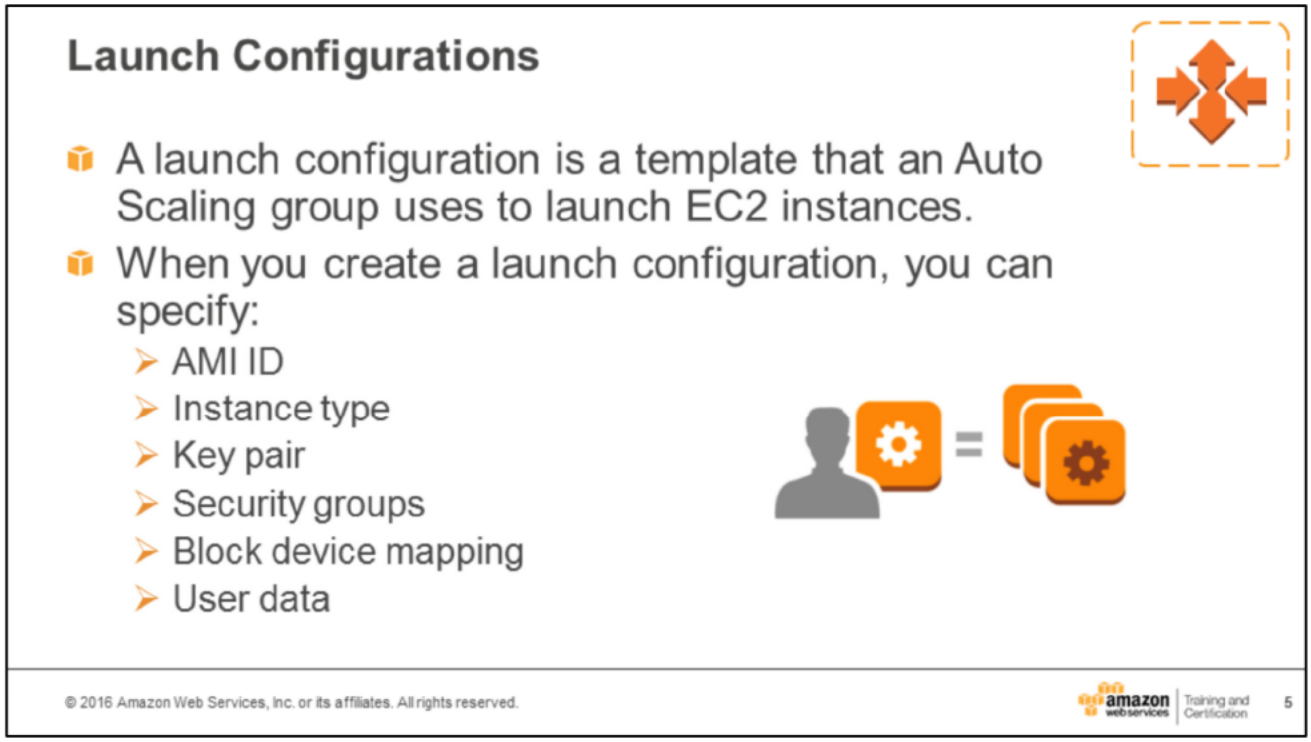


Auto Scaling works as a triad of services working in sync. Elastic Load Balancing and EC2 instances feed metrics to Amazon CloudWatch. Auto Scaling defines a group with launch configurations and Auto Scaling policies. Amazon CloudWatch alarms execute Auto Scaling policies to affect the size of your fleet. All of these services work well individually, but together they become more powerful and increase the control and flexibility our customers demand.



Adding Auto Scaling to your application architecture is one way to maximize the benefits of the AWS cloud. When you use Auto Scaling, your applications gain the following benefits:

- Better fault tolerance: Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
- Better availability: Auto Scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demands.
- Better cost management: Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are actually needed and terminating them when they aren't needed.



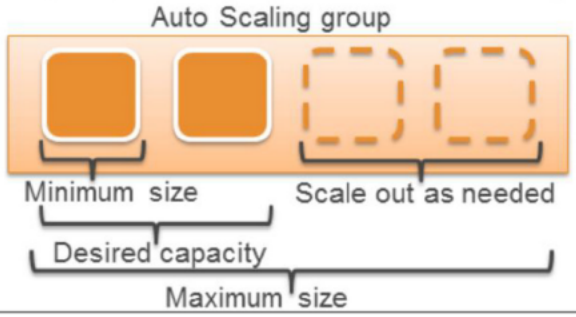
When you create an Auto Scaling group, you must specify a launch configuration. You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. If you want to change the launch configuration for your Auto Scaling group, you must create a new launch configuration and then update your Auto Scaling group with the new launch configuration. When you change the launch configuration for your Auto Scaling group, any new instances are launched using the new configuration parameters, but existing instances are not affected.

For more information, see:

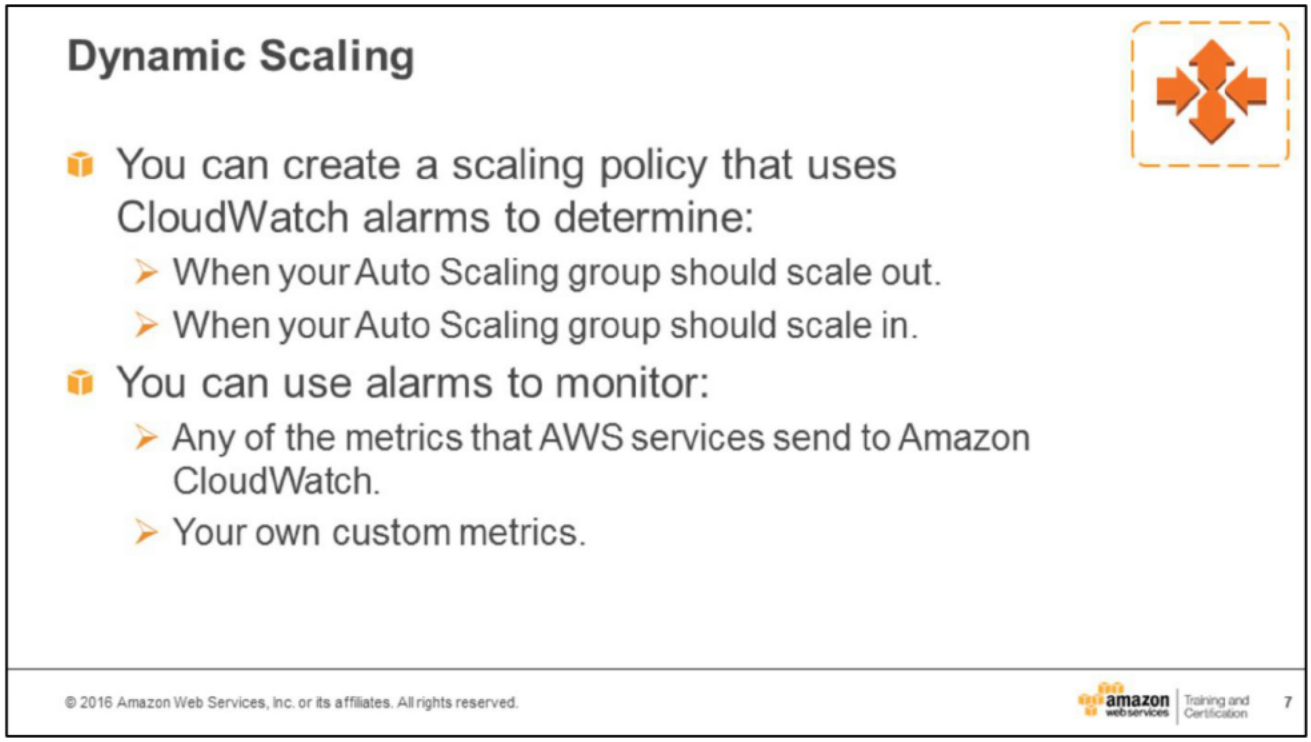
http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/LaunchConfiguration.html

Auto Scaling Groups

- Contain a collection of EC2 instances that share similar characteristics.
- Instances in an Auto Scaling group are treated as a logical grouping for the purpose of instance scaling and management.



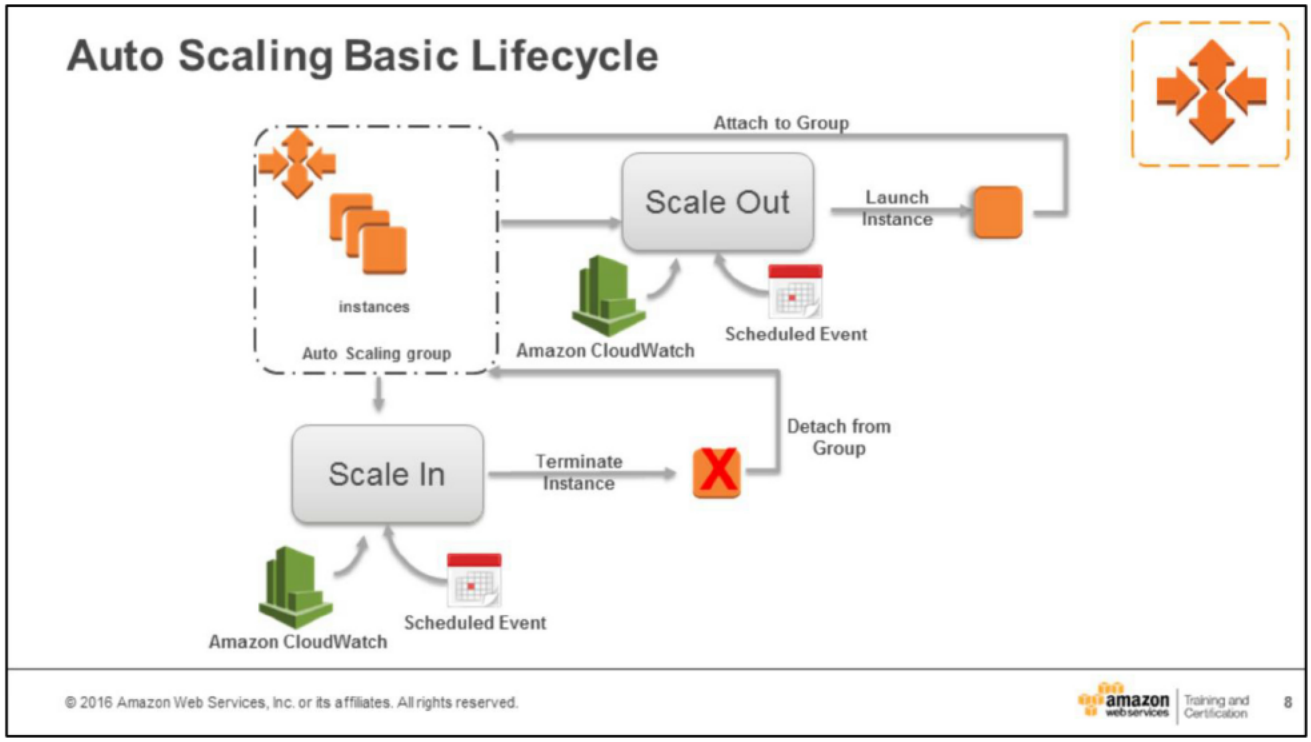
You can create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Auto Scaling can launch or terminate instances as demand on your application increases or decreases.



Each CloudWatch alarm watches a single metric and sends messages to Auto Scaling when the metric breaches a threshold that you specify in your policy.

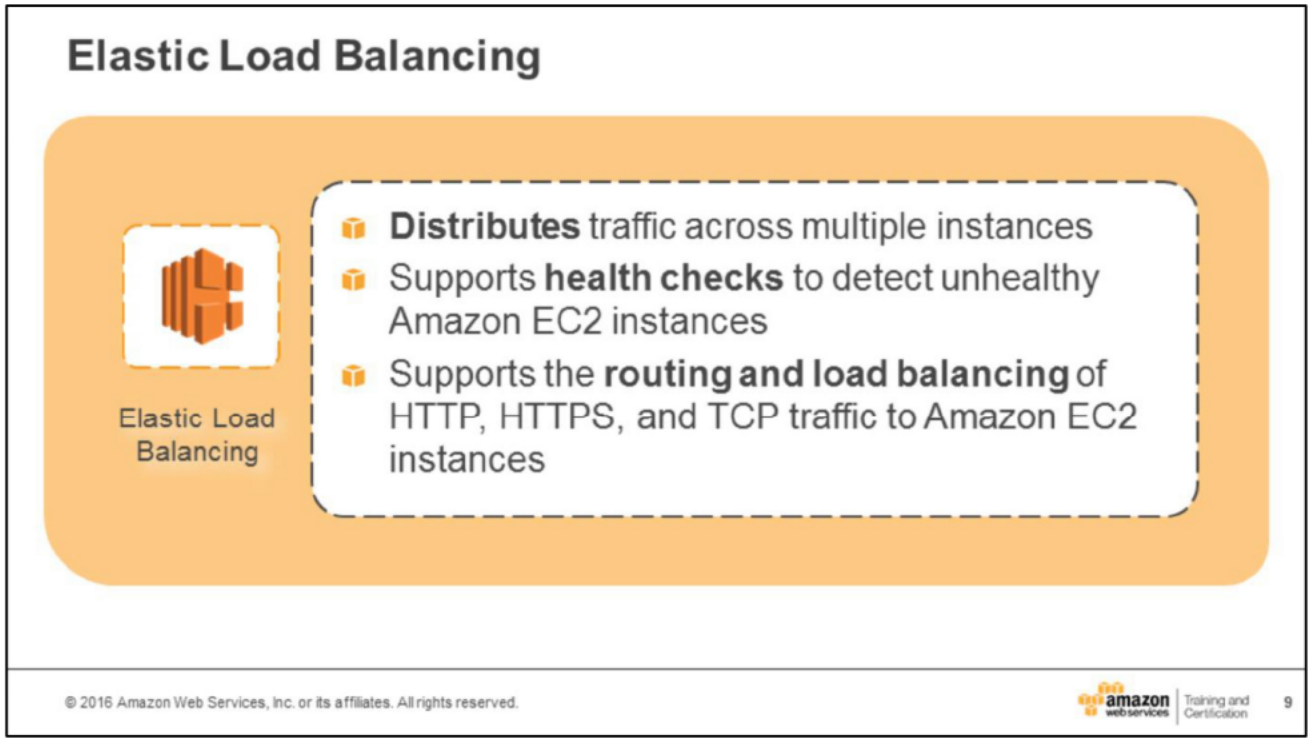
For more information, see:

- http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/as-scale-based-on-demand.html
- http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/policy_creating.html



The slide shows the basic lifecycle of instances within an Auto Scaling Group.

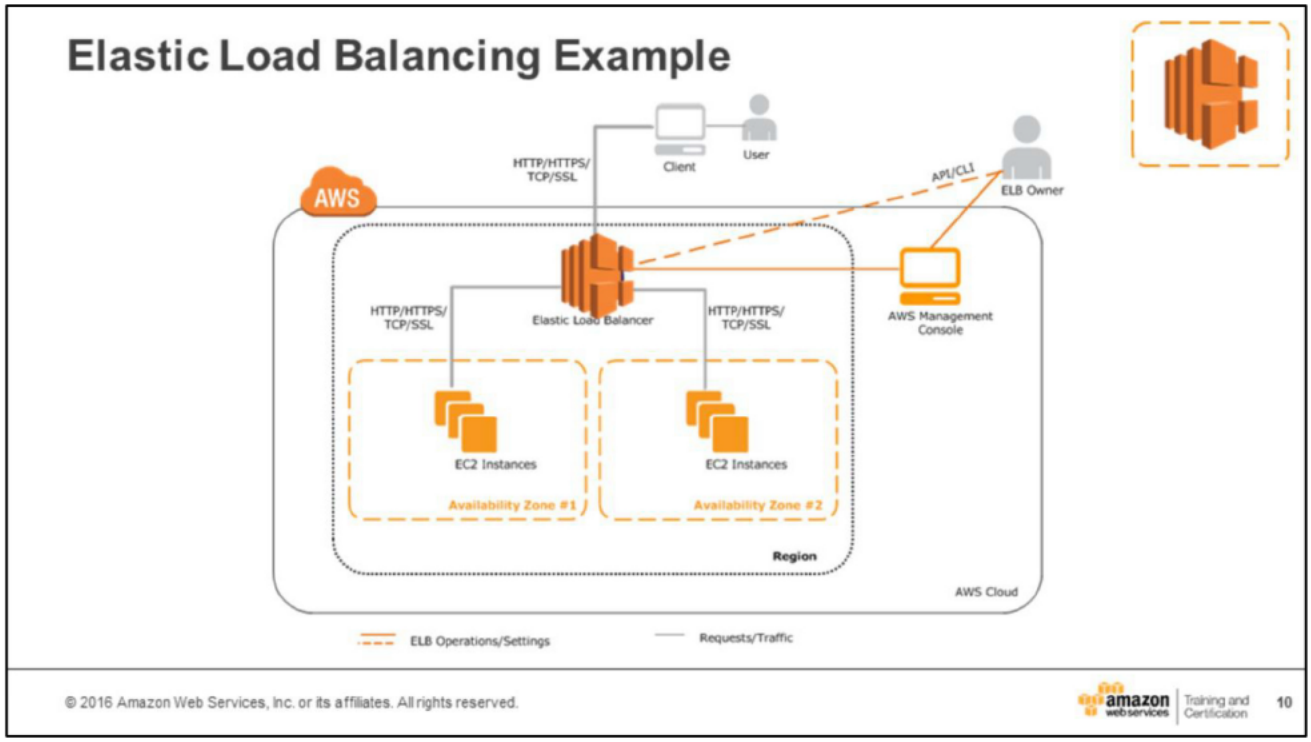
1. The Scaling Group has a desired capacity of three instances.
2. A CloudWatch alarm triggers scaling events and policies scale the group at specific dates and times.
3. The scaling policy launches an instance and attaches it to the Auto Scaling Group.
4. A health check fails and triggers an alarm similar to scaling out.
5. The instance is terminated.
6. The instance is detached from the Auto Scaling Group.



Understand Elastic Load Balancing (ELB) concepts including:

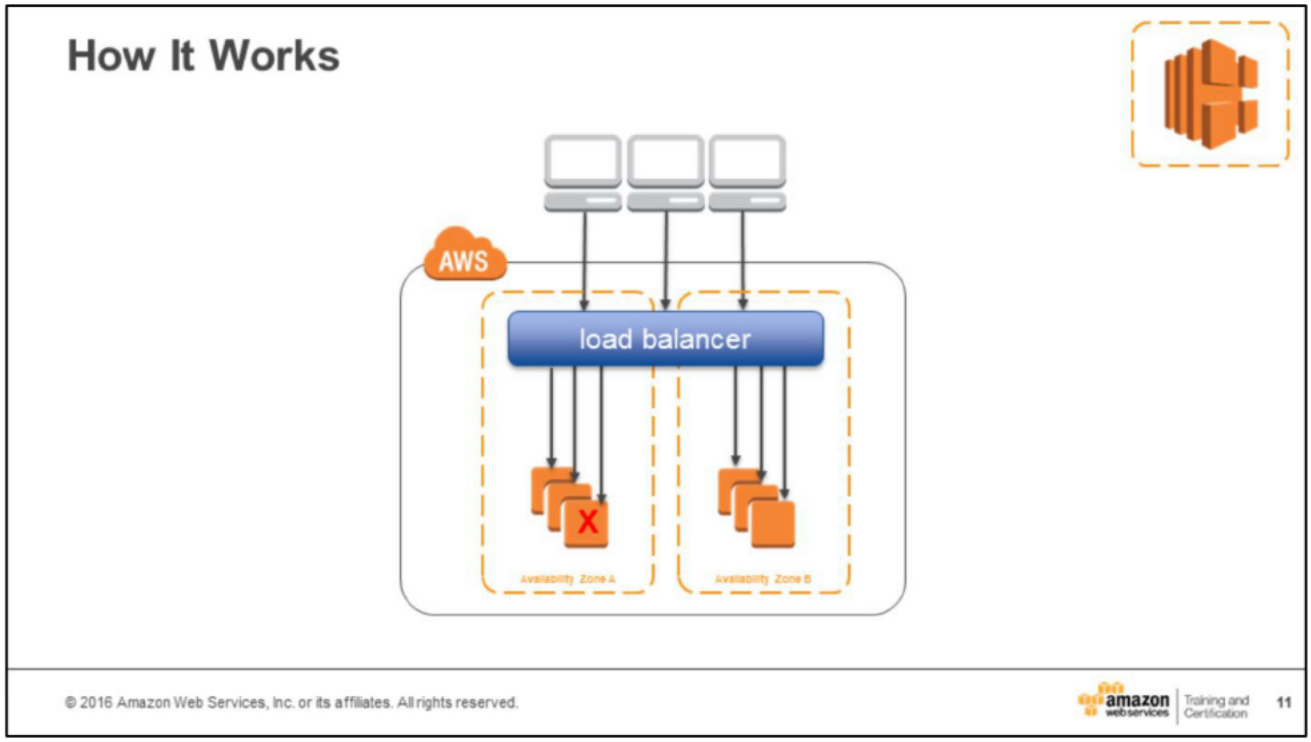
- Request Routing
- Internet-facing vs. internal vs. http load balancers
- Back-end instances
- Listeners

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. You can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for even more consistent application performance.

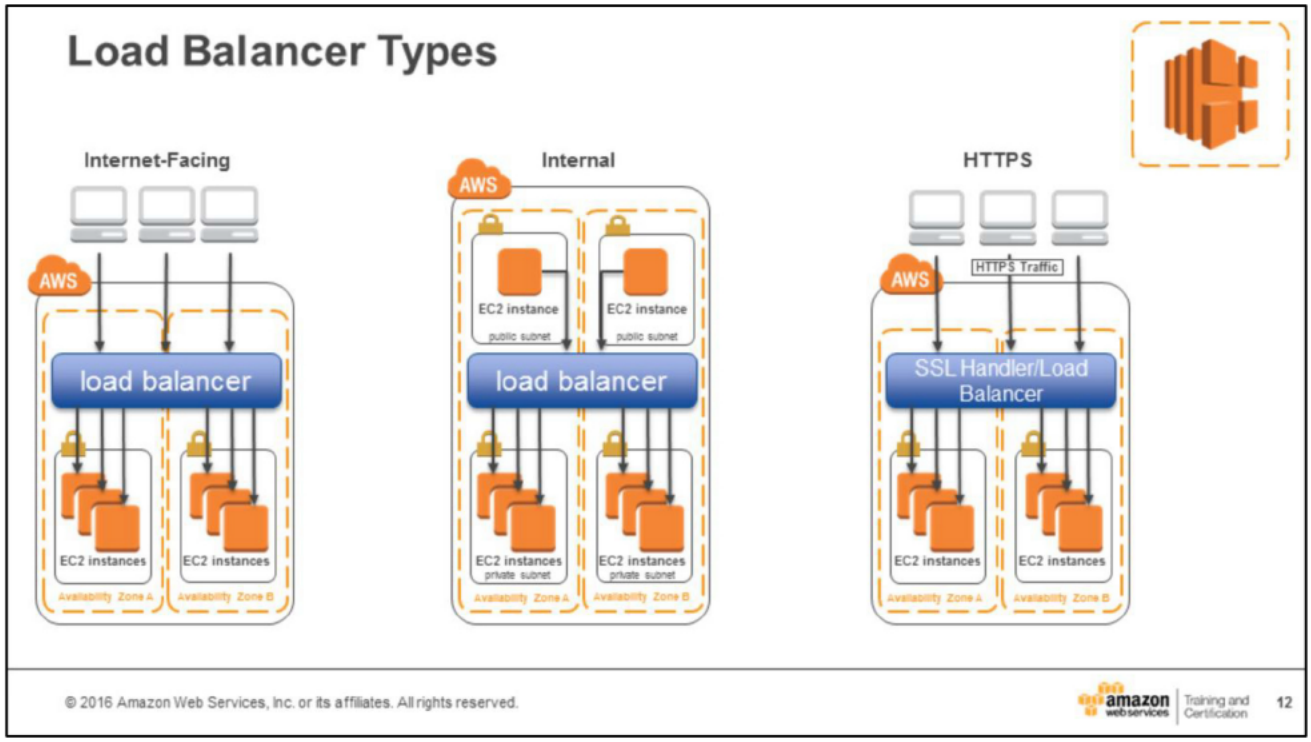


Elastic Load Balancing automatically scales its request handling capacity in response to incoming traffic. The diagram shows how the various components of Elastic Load Balancing work together. You can access and work with your load balancer using one of the following interfaces:

- AWS Management Console—A simple web browser interface that you can use to create and manage your load balancers without using additional software or tools.
- Command Line Interfaces —A Java-based command line client that wraps the SOAP API.



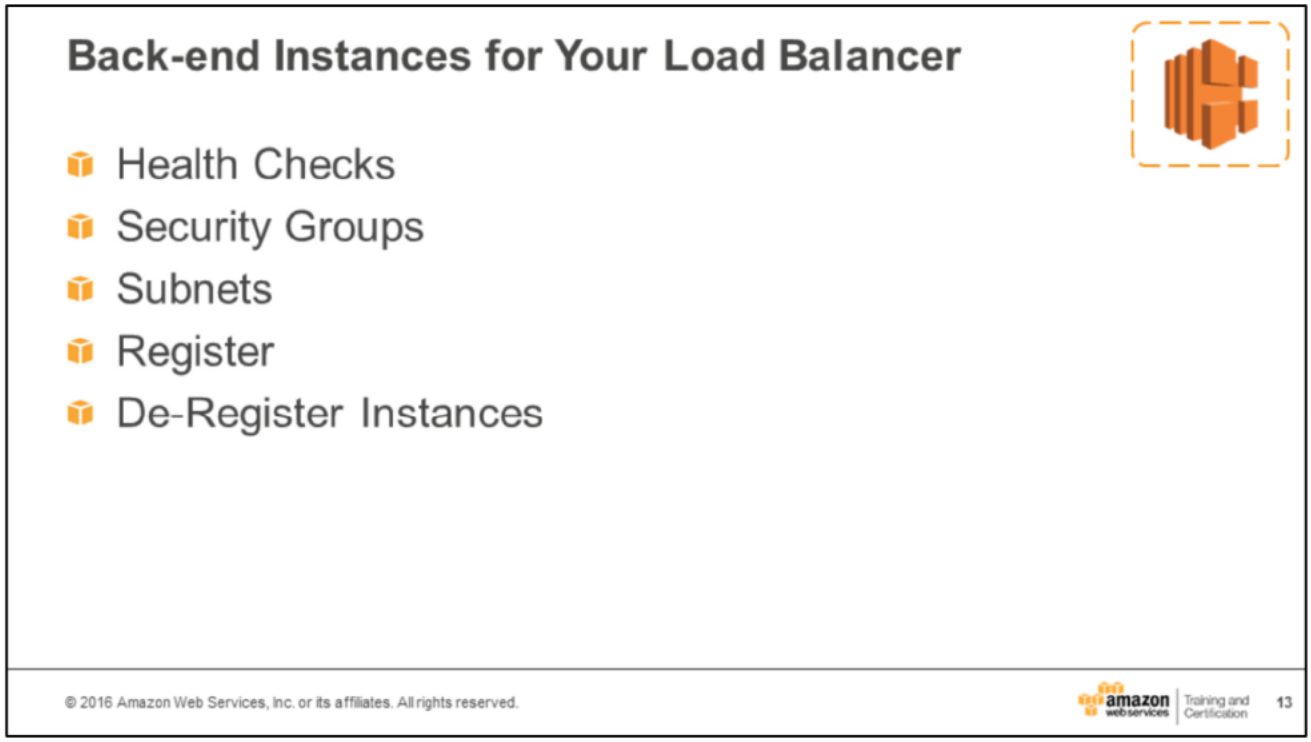
A load balancer accepts incoming traffic from clients and routes requests to its registered EC2 instances in one or more Availability Zones. The load balancer also monitors the health of its registered instances and ensures that it routes traffic only to healthy instances. When the load balancer detects an unhealthy instance, it stops routing traffic to that instance, and then resumes routing traffic to that instance when it detects that the instance is healthy again.



Internet-facing load balancer: An Internet-facing load balancer takes requests from clients over the Internet and distributes them across the EC2 instances that are registered with the load balancer.

Internal load balancer: An internal load balancer routes traffic to your EC2 instances in private subnets. The clients must have access to the private subnets.

HTTPS load balancer: You can create a load balancer that uses the SSL/TLS protocol for encrypted connections (also known as SSL offload). This feature enables traffic encryption between your load balancer and the clients that initiate HTTPS sessions, and for connections between your load balancer and your back-end instances.



After you've created your load balancer, you must register your EC2 instances with the load balancer. You can select EC2 instances from a single Availability Zone or multiple Availability Zones within the same region as the load balancer. Elastic Load Balancing routinely performs health checks on registered EC2 instances, and automatically distributes incoming requests to the DNS name of your load balancer across the registered, healthy EC2 instances.

Health checks are periodic pings, attempted connections, or requests sent to EC2 instances by your load balancer to check the availability of your EC2 instances. The load balancer performs health checks on all registered instances, whether the instance is in a healthy state or an unhealthy state. The load balancer routes requests only to the healthy instances. When the load balancer determines that an instance is unhealthy, it stops routing requests to that instance. The load balancer resumes routing requests to the instance when it has been restored to a healthy state.

A security group acts as a firewall that controls the traffic allowed to and from one or more instances. When you launch an EC2 instance, you can associate one or more security groups with the instance. For each security group, you add one or more rules to allow traffic. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances associated with the security group. You must ensure that the security groups for your instances allow the load balancer to communicate with your back-end instances on both the listener port and the health check port. In a VPC, your security groups and network access control lists (ACL) must allow traffic in both directions on these ports.

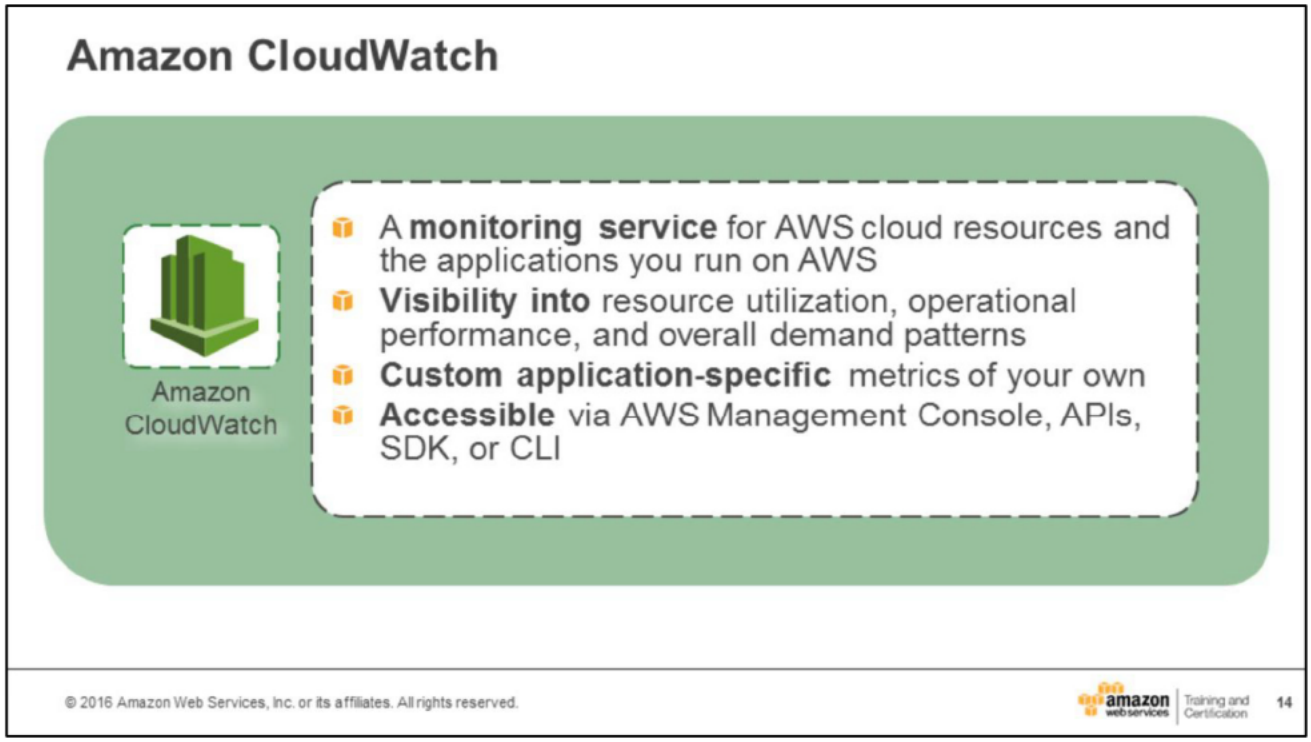
When you attach a subnet to your load balancer, Elastic Load Balancing creates a load balancer node in the Availability Zone. Load balancer nodes accept traffic from clients and forward requests to the healthy registered instances in one or more Availability Zones. For load balancers in a VPC, we recommend that you attach one subnet per Availability Zone for at least two Availability Zones. This improves the availability of your load balancer. Note that you can modify the subnets attached to your load balancer at any time.

Registering an EC2 instance adds it to your load balancer. The load balancer continuously monitors the health of its registered instances, and routes requests to the healthy registered instances. If demand on your instances increases, you can register additional instances with the load balancer to handle the demand.

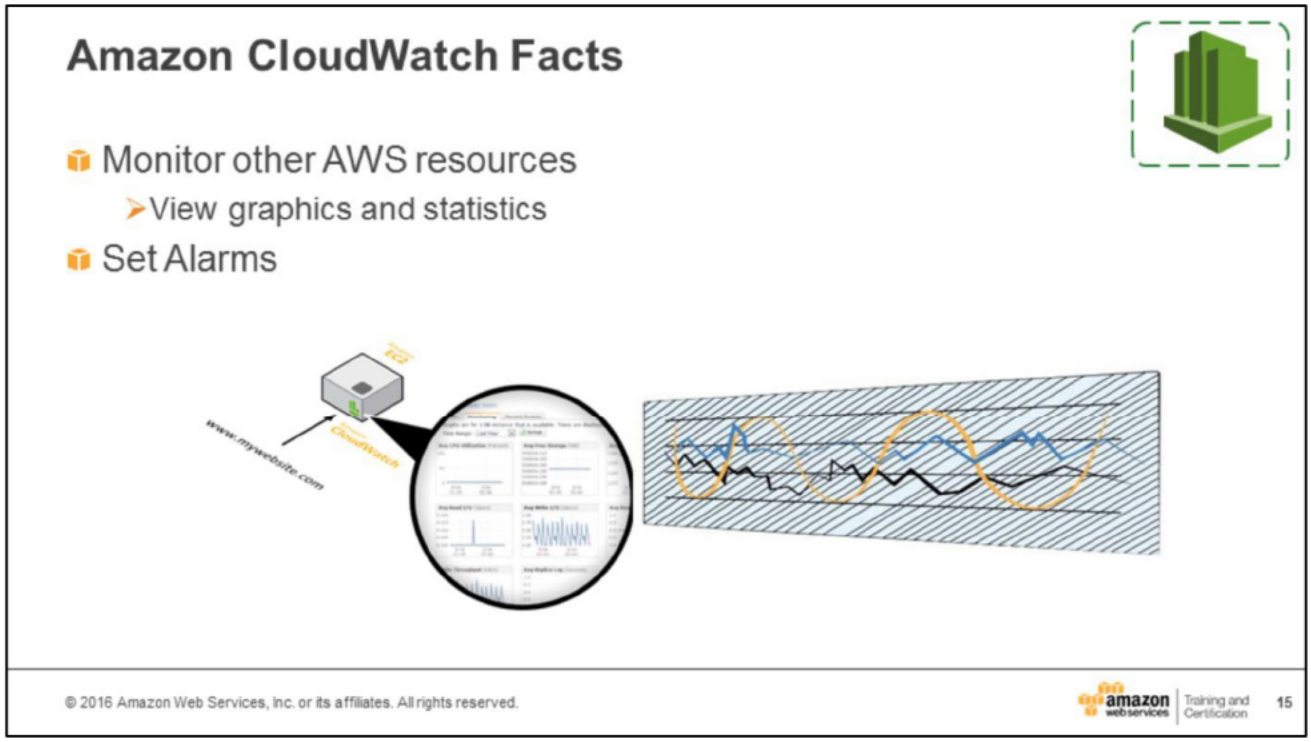
De-registering an EC2 instance removes it from your load balancer. The load balancer stops routing requests to an instance as soon as it is de-registered. If demand decreases, or you need to service your instances, you can de-register instances from the load balancer. A de-registered instance remains running, but no longer receives traffic from the load balancer, and you can register it with the load balancer again when you are ready.

For more information, see:

http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-backend-instances.html



CloudWatch lets you view graphs, set alarms to troubleshoot, spot trends, and take automated action based on the state. It is accessible via the AWS Management Console, APIs, SDK or CLI. You can customize with your own metrics or use a sample template found online.

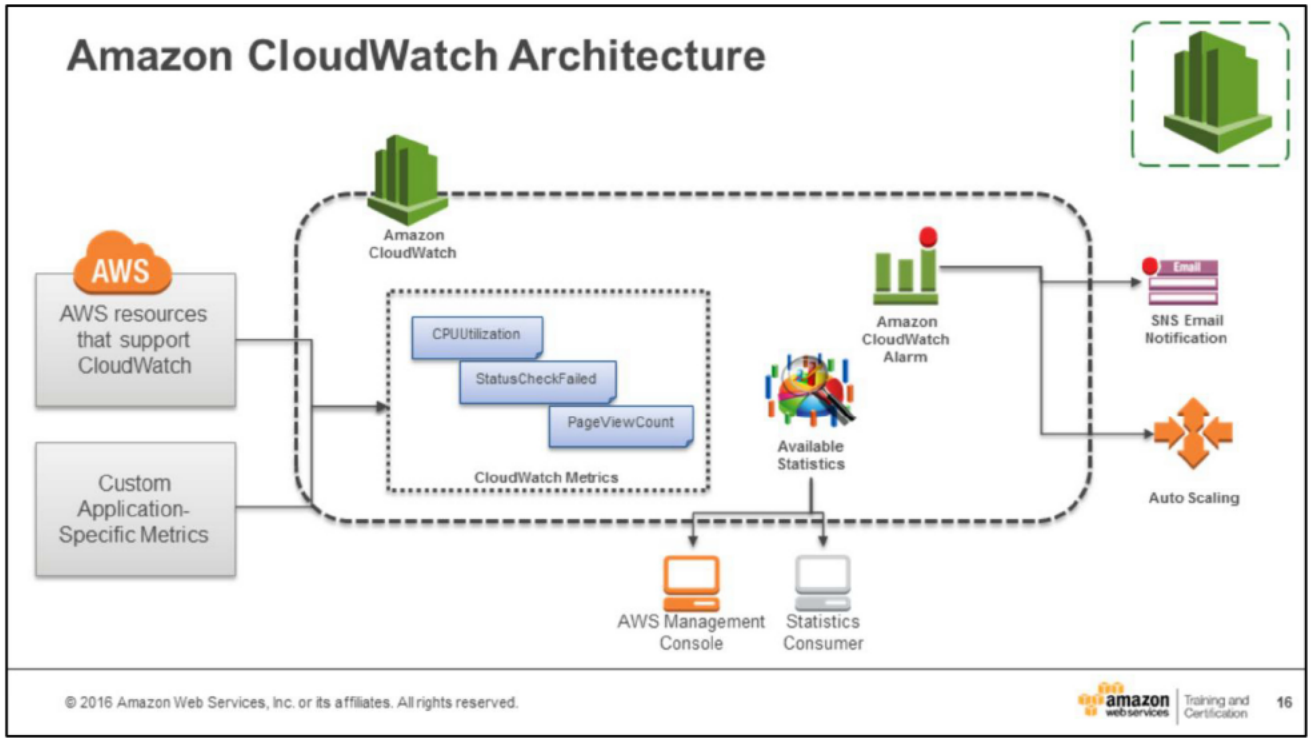


For Amazon EC2 instances, Amazon CloudWatch basic monitoring collects and reports metrics for CPU utilization, data transfer, and disk usage activity from each Amazon EC2 instance at a five-minute frequency. Amazon CloudWatch detailed monitoring provides these same metrics at one-minute intervals, and also enables data aggregation by Amazon EC2 AMI ID and instance type.

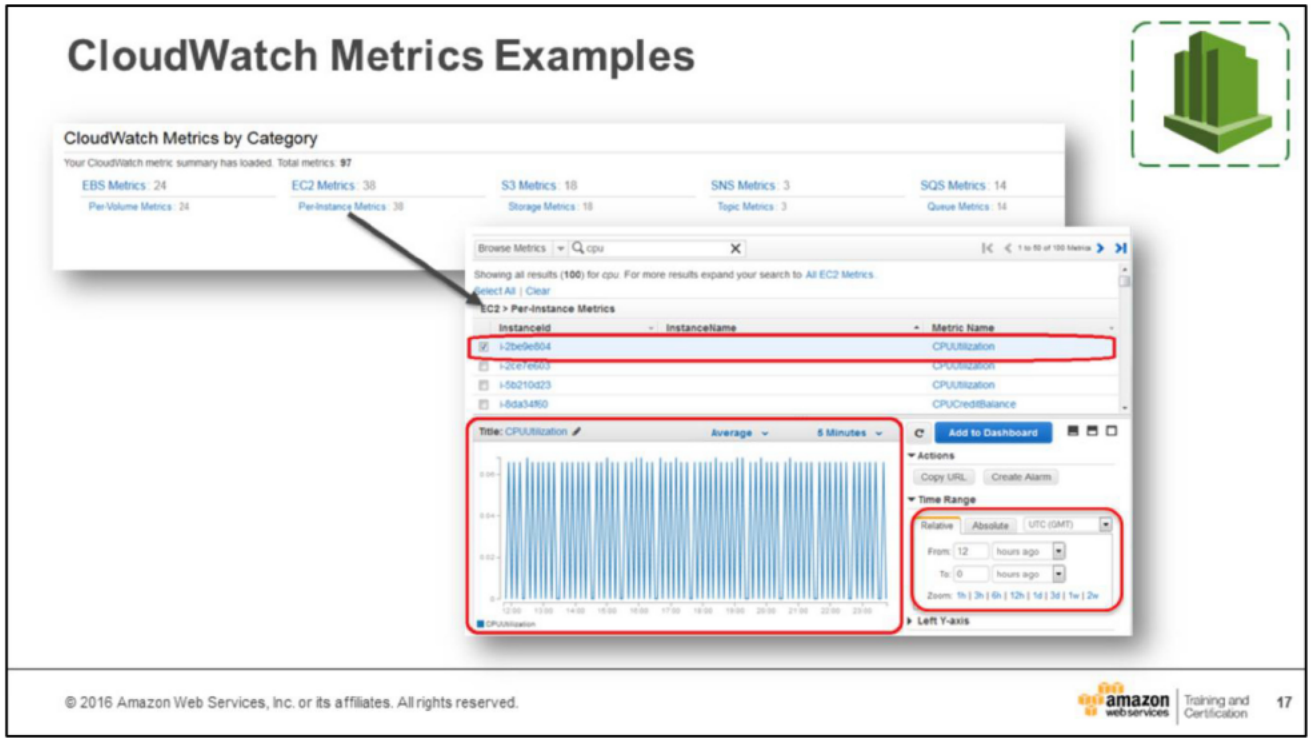
Set alarms on any of your metrics to receive notifications. You can also use Auto Scaling to add or remove Amazon instances.

For more information, see:

- http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/AlarmThatSendsEmail.html
- https://aws.amazon.com/cloudwatch/details/#other-aws-resource-monitoring
- https://aws.amazon.com/blogs/aws/new-cloudwatch-events-track-and-respond-to-changes-to-your-aws-resources/

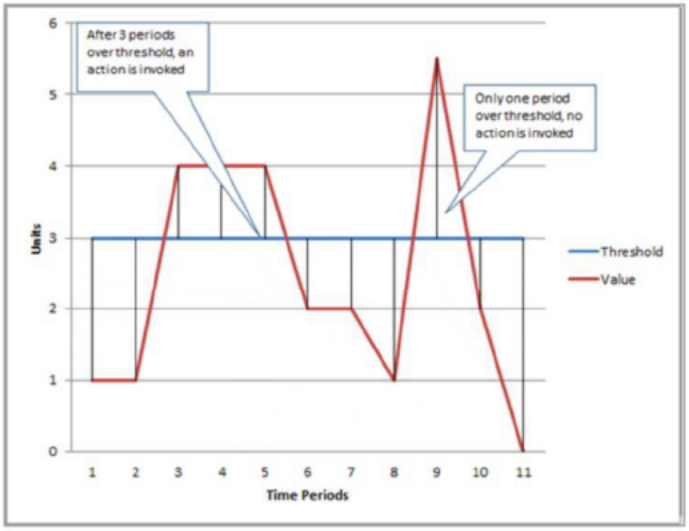


Amazon CloudWatch is a metrics repository. AWS products put metrics into the repository and you retrieve statistics based on the metrics. Statistics can be graphically presented in the CloudWatch console.



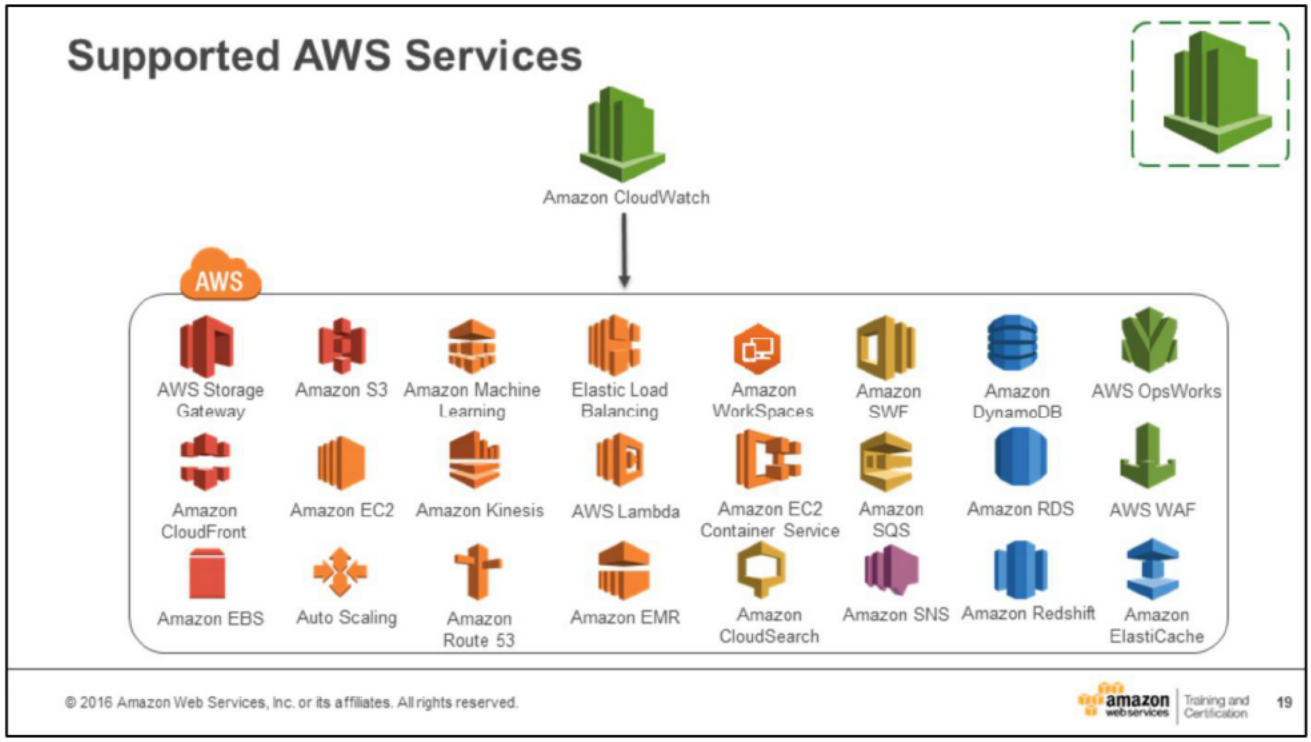
The slide shows screenshots from the Amazon CloudWatch Console. In the example, the user has selected an EC2 per-instance metric of CPUUtilization.

CloudWatch Alarms



You can create a CloudWatch alarm that sends an Amazon Simple Notification Service (SNS) message when the alarm changes state. An alarm watches a single metric over a time period you specify, and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state, the state must have changed and been maintained for a specified number of periods.

In the slide, the alarm threshold is set to 3 and the minimum breach is 3 periods. The alarm invokes its action only when the threshold is breached for 3 consecutive periods. In the figure, this happens with the third through fifth time periods, and the alarm is triggered. At period six, the value dips below the threshold, and the state is set to OK. Later, during the ninth time period, the threshold is breached again, but not for the necessary three consecutive periods. Consequently, the alarm's state remains OK.



The slides shows AWS services that Amazon CloudWatch collects metrics from.

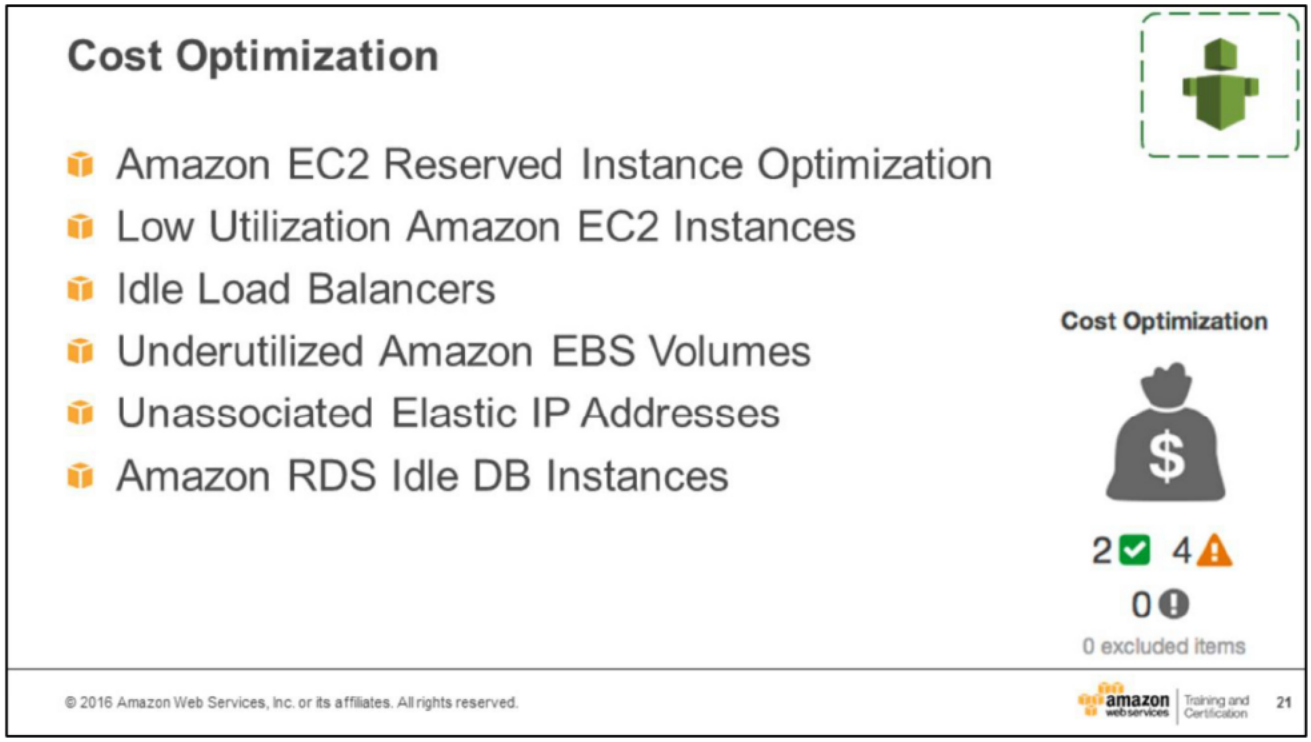
For more information, see:

http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/supported_services.html

The status of checks provided by AWS Trusted Advisor is shown by using color coding on the dashboard page:

- Red: action recommended
- Yellow: investigation recommended
- Green: no problem detected

For each check, you can review a detailed description of the recommended best practice, a set of alert criteria, guidelines for action, and a list of useful resources on the topic.

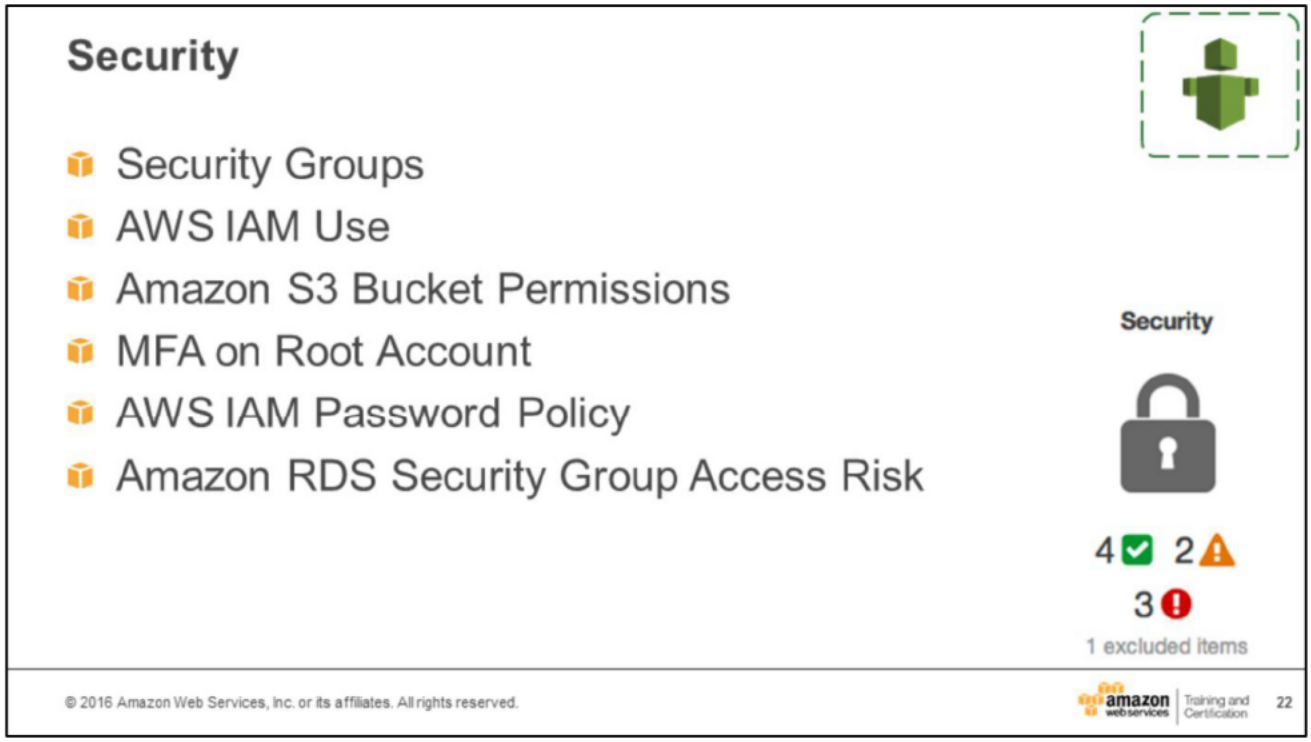


AWS Trusted Advisor helps you save money on AWS by checking for unused and idle resources and making commitments to reserved capacity.

The following cost optimization checks are available with Trusted Advisor:

- Amazon EC2 Reserved Instance Optimization: Checks your Amazon Elastic Compute Cloud (Amazon EC2) computing consumption history and calculates an optimal number of Partial Upfront Reserved Instances. Recommendations are based on the previous calendar month's hour-by-hour usage aggregated across all consolidated billing accounts.
- Low Utilization Amazon EC2 Instances: Checks the Amazon EC2 instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was 10% or less and network I/O was 5 MB or less on 4 or more days.
- Idle Load Balancers: Checks your Elastic Load Balancing configuration for load balancers that are not actively used.
- Underutilized Amazon EBS Volumes: Checks Amazon Elastic Block Store (Amazon EBS) volume configurations and warns when volumes appear to be underused.
- Unassociated Elastic IP Addresses: Checks for Elastic IP addresses (EIPs) that are not associated with a running Amazon EC2 instance.

- Amazon RDS Idle DB Instances: Checks the configuration of your Amazon Relational Database Service (Amazon RDS) for any DB instances that appear to be idle. If a DB instance has not had a connection for a prolonged period of time, you can shut down the instance to reduce costs. If persistent storage is needed for data on the instance, you can use lower-cost options such as taking and retaining a DB snapshot.

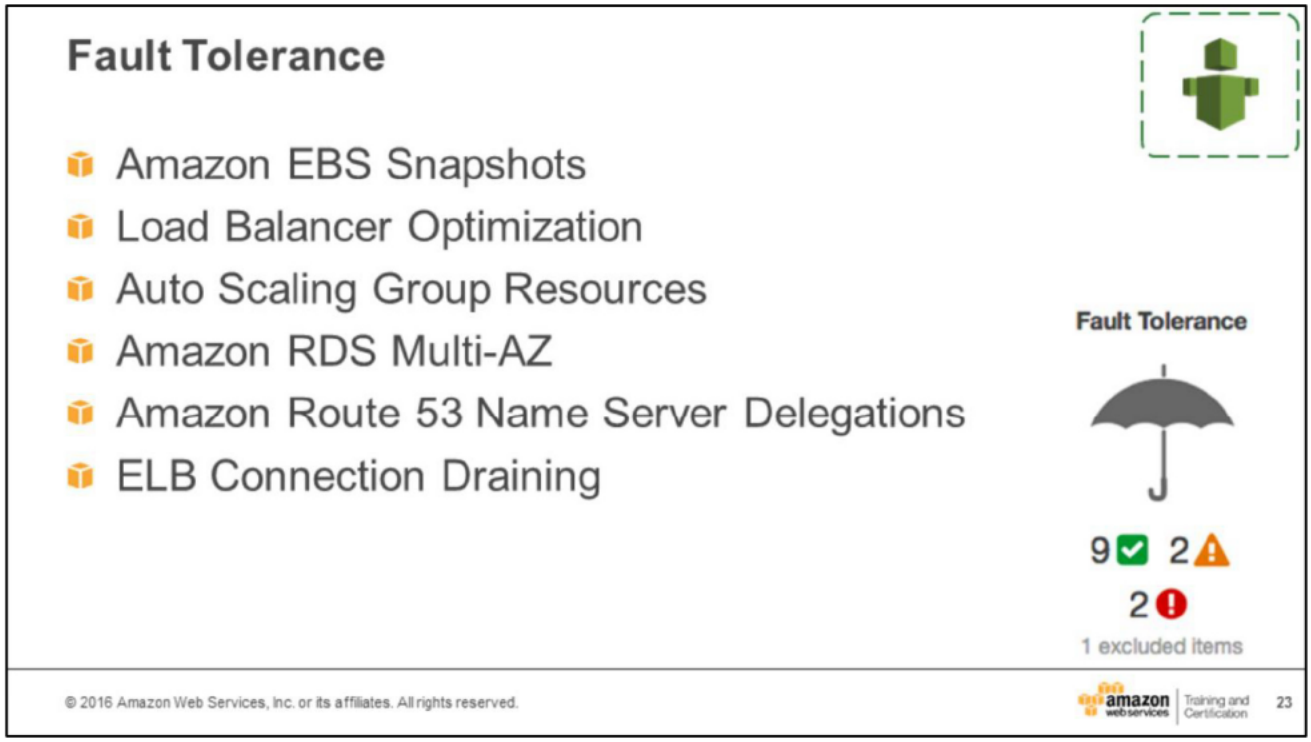


AWS Trusted Advisor helps you improve the security of your application by closing gaps, enabling various AWS security features, and examining your permissions.

The following security checks are available with Trusted Advisor:

- Security Groups - Specific Ports Unrestricted: Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports. Unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data). The ports with highest risk are flagged red, and those with less risk are flagged yellow. Ports flagged green are typically used by applications that require unrestricted access, such as HTTP and SMTP.
- Security Groups - Unrestricted Access: Checks security groups for rules that allow unrestricted access to a resource. Unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data).
- IAM Use (Free!): Checks for your use of AWS Identity and Access Management (IAM).
- Amazon S3 Bucket Permissions : Checks buckets in Amazon Simple Storage Service (Amazon S3) that have open access permissions. This check examines explicit bucket permissions, but it does not examine associated bucket policies that might override the bucket permissions.
- MFA on Root Account: Checks the root account and warns if multi-factor authentication (MFA) is not enabled.

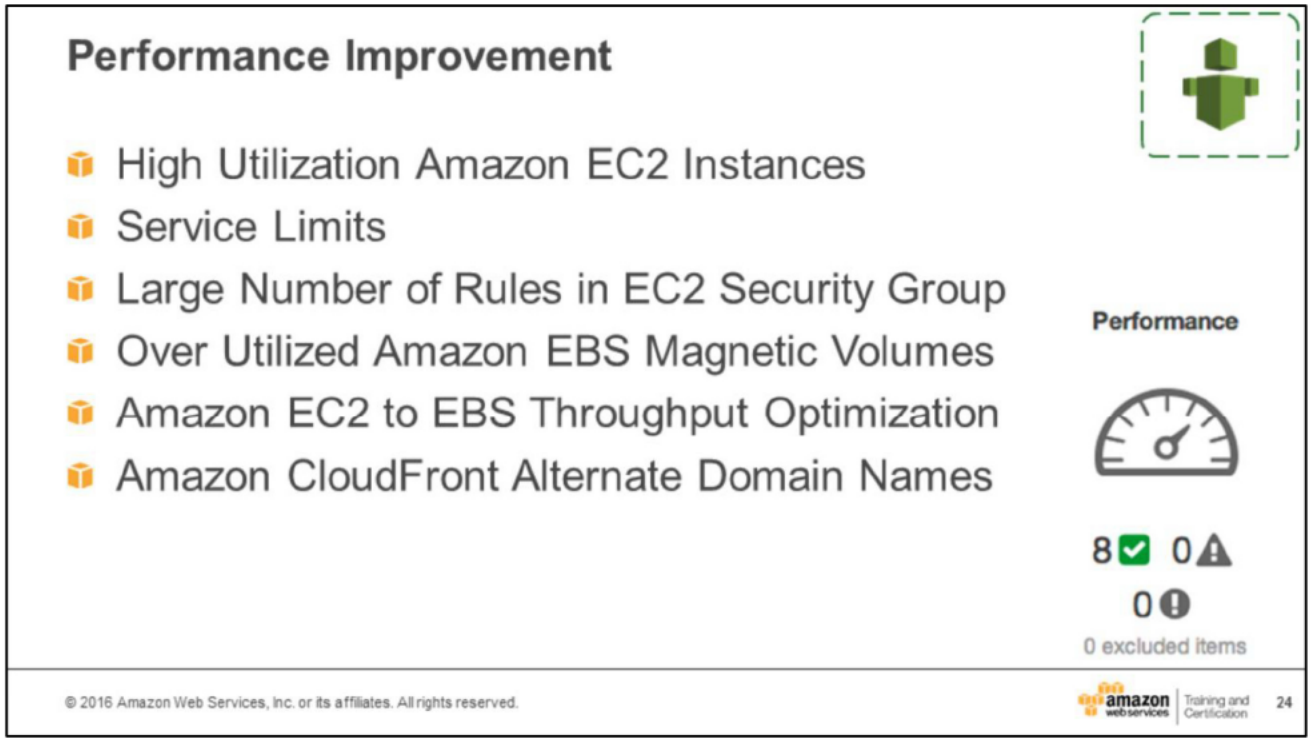
- IAM Password Policy: Checks the password policy for your account and warns when a password policy is not enabled, or if password content requirements have not been enabled.
- Amazon RDS Security Group Access Risk: Checks security group configurations for Amazon Relational Database Service (Amazon RDS) and warns when a security group rule might grant overly permissive access to your database.



AWS Trusted Advisor helps you increase the availability and redundancy of your AWS application by taking advantage of auto scaling, health checks, multi AZ, and backup capabilities.

The following fault tolerance checks are available with Trusted Advisor:

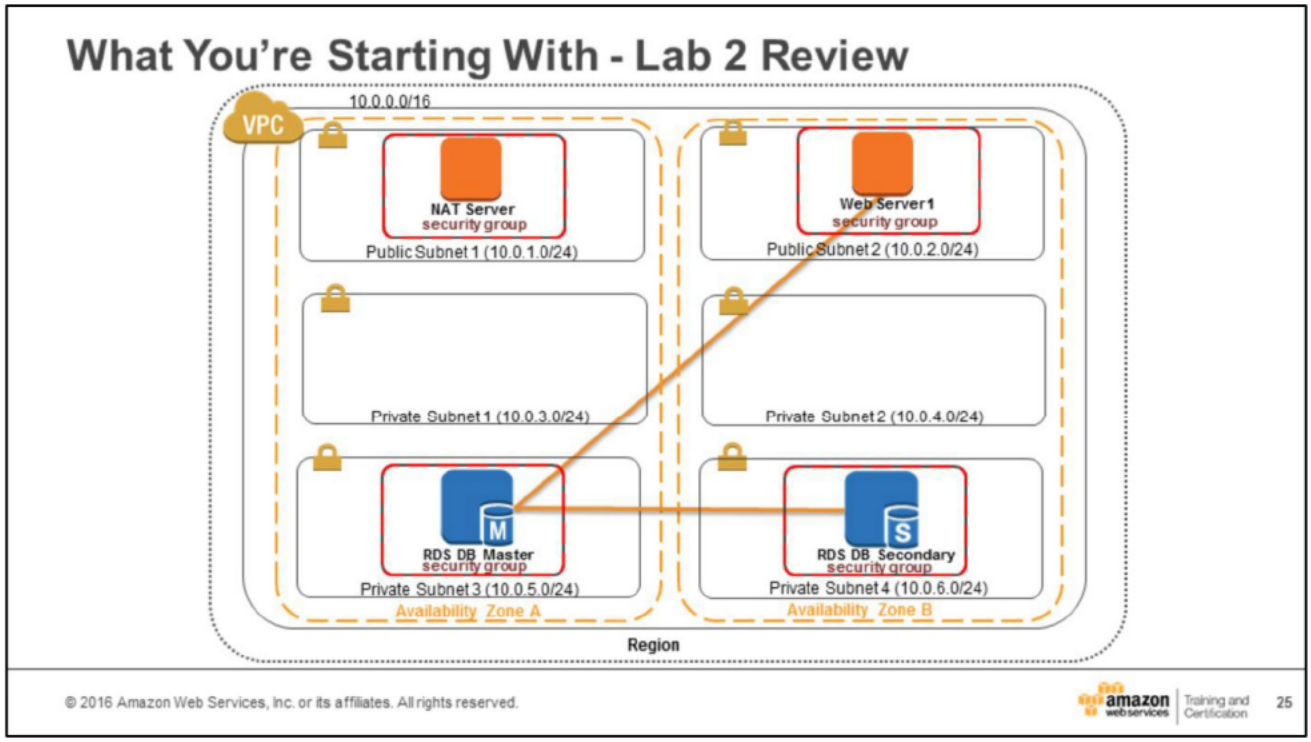
- Amazon EBS Snapshots: Checks the age of the snapshots for your Amazon Elastic Block Store (Amazon EBS) volumes (available or in-use).
- Load Balancer Optimization: Checks your load balancer configuration.
- Auto Scaling Group Resources: Checks the availability of resources associated with launch configurations and your Auto Scaling groups.
- Amazon RDS Multi-AZ: Checks for DB instances that are deployed in a single Availability Zone.
- Amazon Route 53 Name Server Delegations: Checks for Amazon Route 53 hosted zones for which your domain registrar or DNS is not using the correct Route 53 name servers.
- ELB Connection Draining: Checks for load balancers that do not have connection draining enabled.

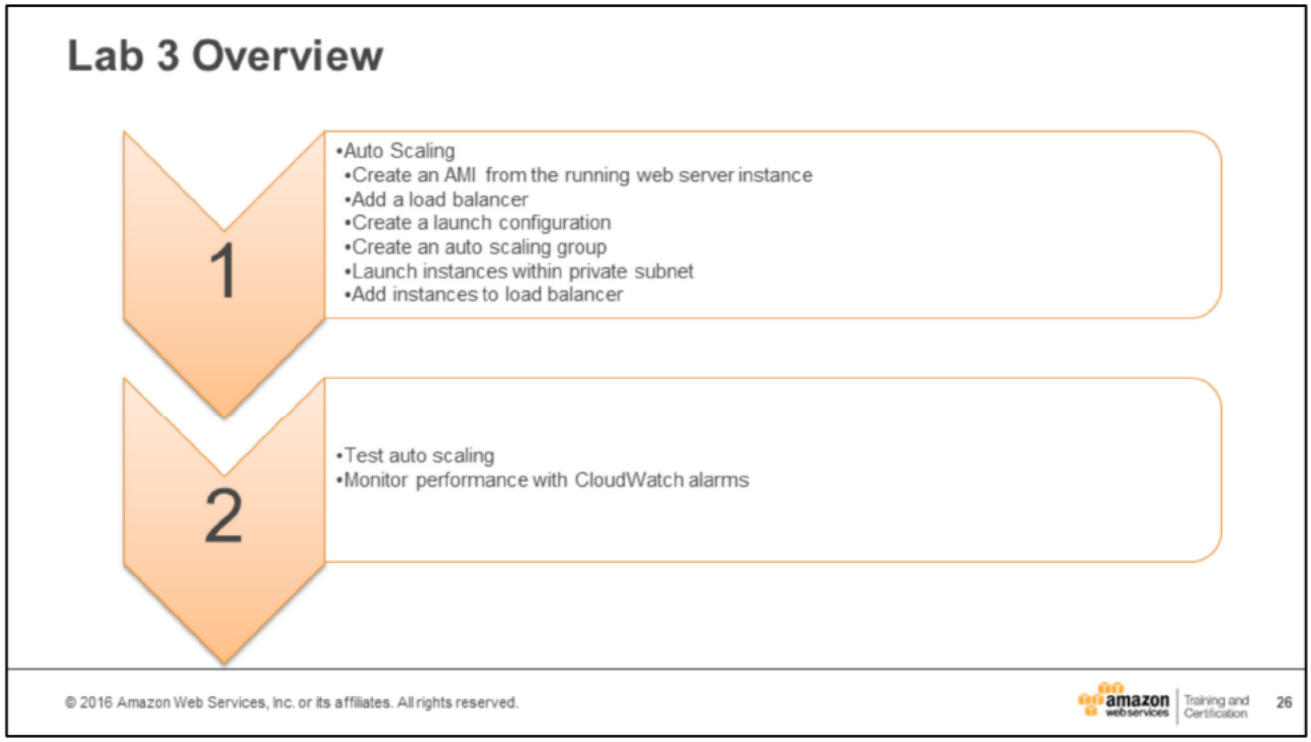


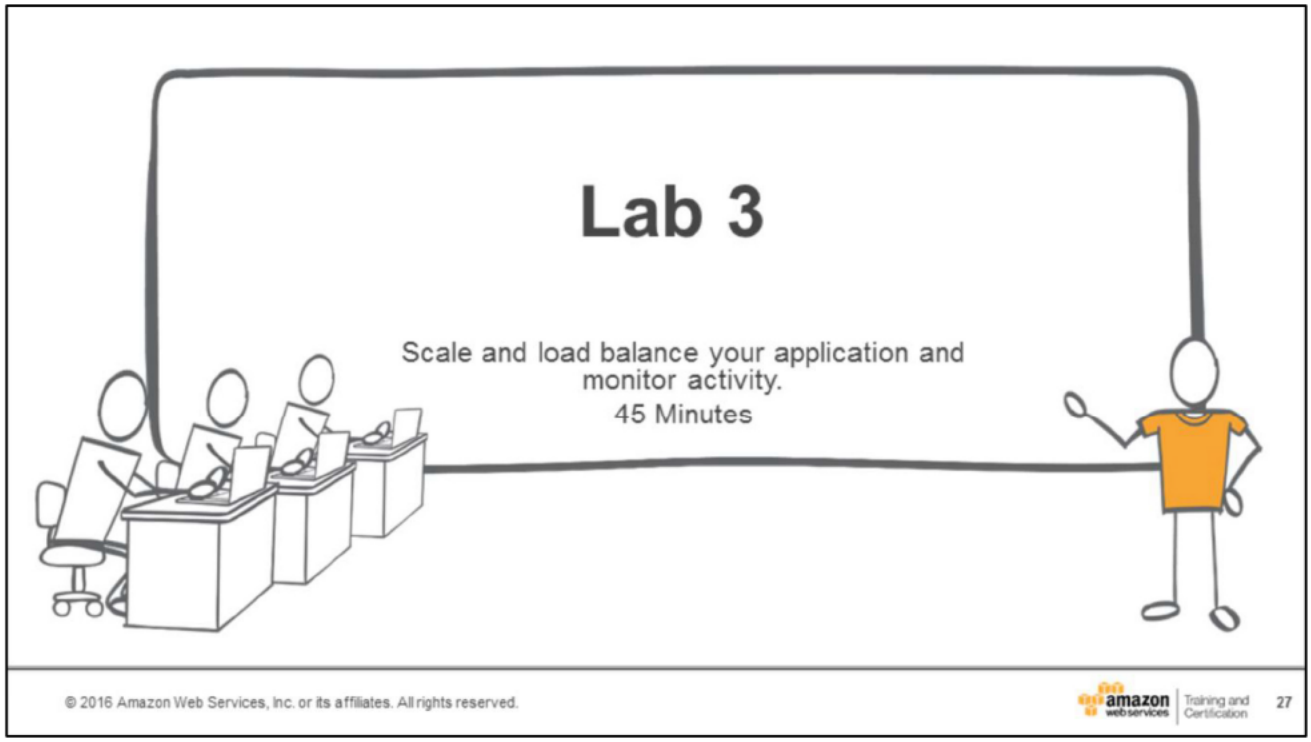
AWS Trusted Advisor helps you improve the performance of your service by checking your service limits, ensuring you take advantage of provisioned throughput, and monitoring for over utilized instances.

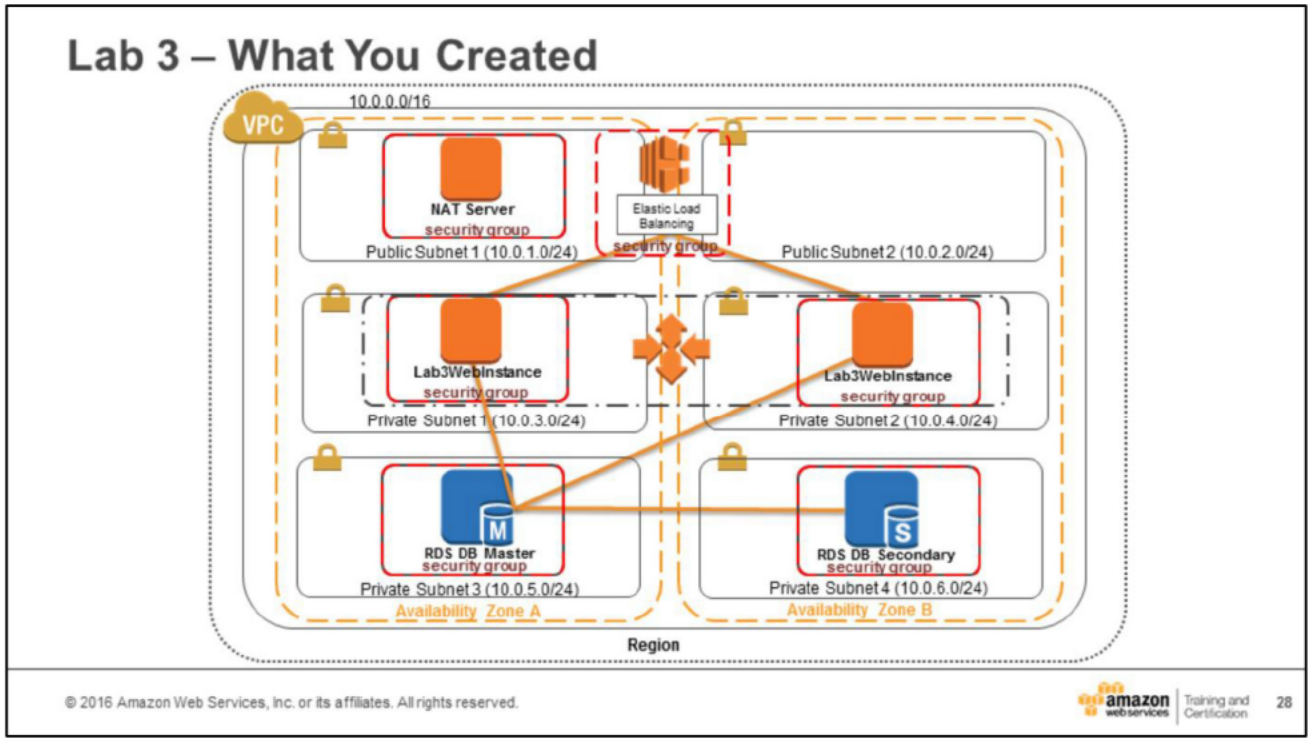
The following performance improvement checks are available with Trusted Advisor:

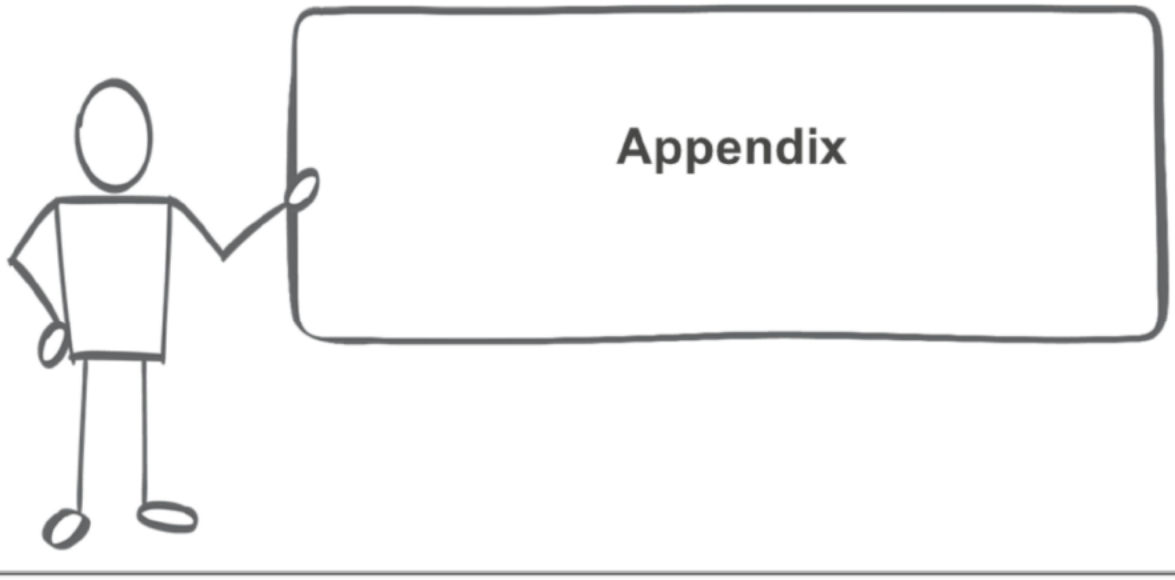
- High Utilization Amazon EC2 Instances: Checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was more than 90% on 4 or more days.
- Service Limits: Checks for usage that is more than 80% of the service limit.
- Large Number of Rules in EC2 Security Group: Checks each Amazon EC2 security group for an excessive number of rules.
- Over Utilized Amazon EBS Magnetic Volumes: Checks for Amazon Elastic Block Store (EBS) Magnetic volumes that are potentially overutilized and might benefit from a more efficient configuration.
- Amazon EC2 to EBS Throughput Optimization: Checks for Amazon EBS volumes whose performance might be affected by the maximum throughput capability of the Amazon EC2 instance they are attached to.
- CloudFront Alternate Domain Names: Checks Amazon CloudFront distributions for alternate domain names with incorrectly configured DNS settings.

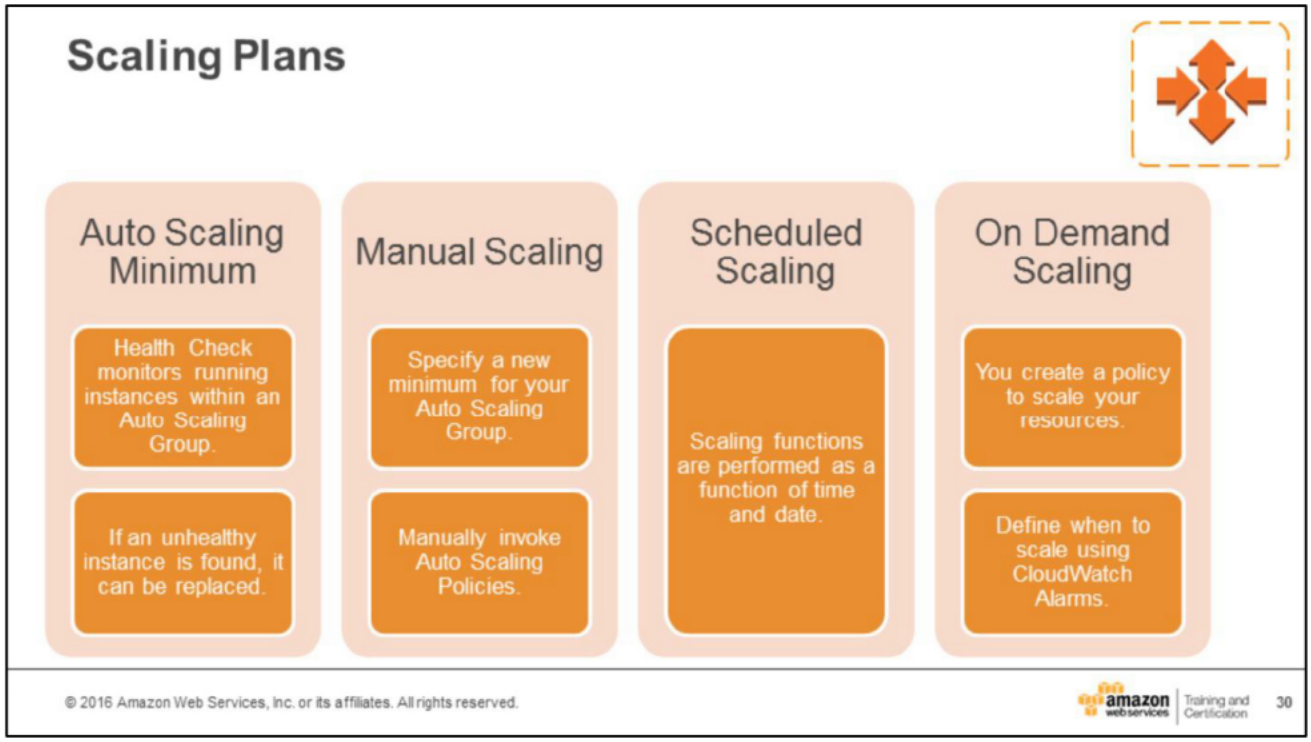

What You're Starting With - Lab 2 Review
VPC
10.0.0.0/16
NAT Server
security group
Public Subnet 1 (10.0.1.0/24)
Web Server 1
security group
Public Subnet 2 (10.0.2.0/24)
Private Subnet 1 (10.0.3.0/24)
Private Subnet 2 (10.0.4.0/24)
M
RDS DB Master
security group
Private Subnet 3 (10.0.5.0/24)
S
RDS DB Secondary
security group
Private Subnet 4 (10.0.6.0/24)
Availability Zone A
Availability Zone B
Region
© 2016 Amazon Web Services, Inc. or its affiliates. All rights reserved.
amazon web services | Training and Certification
25





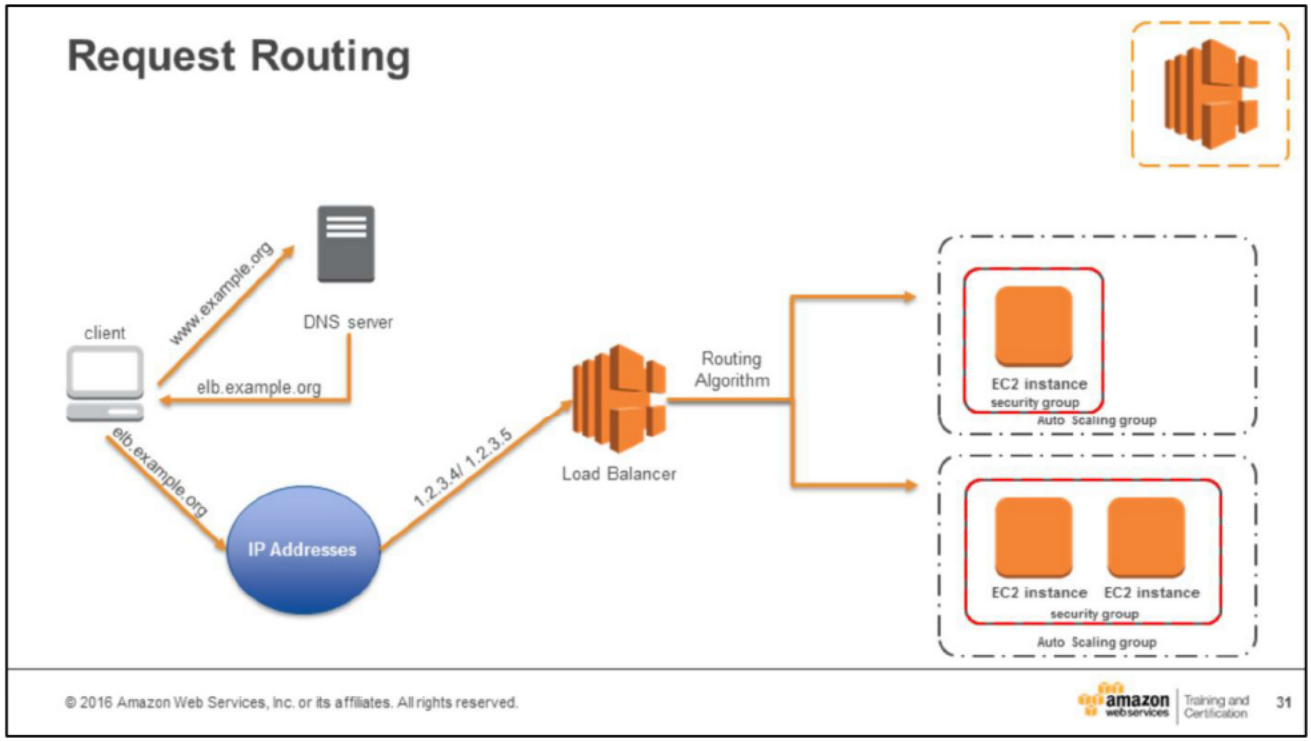






For more information, see:

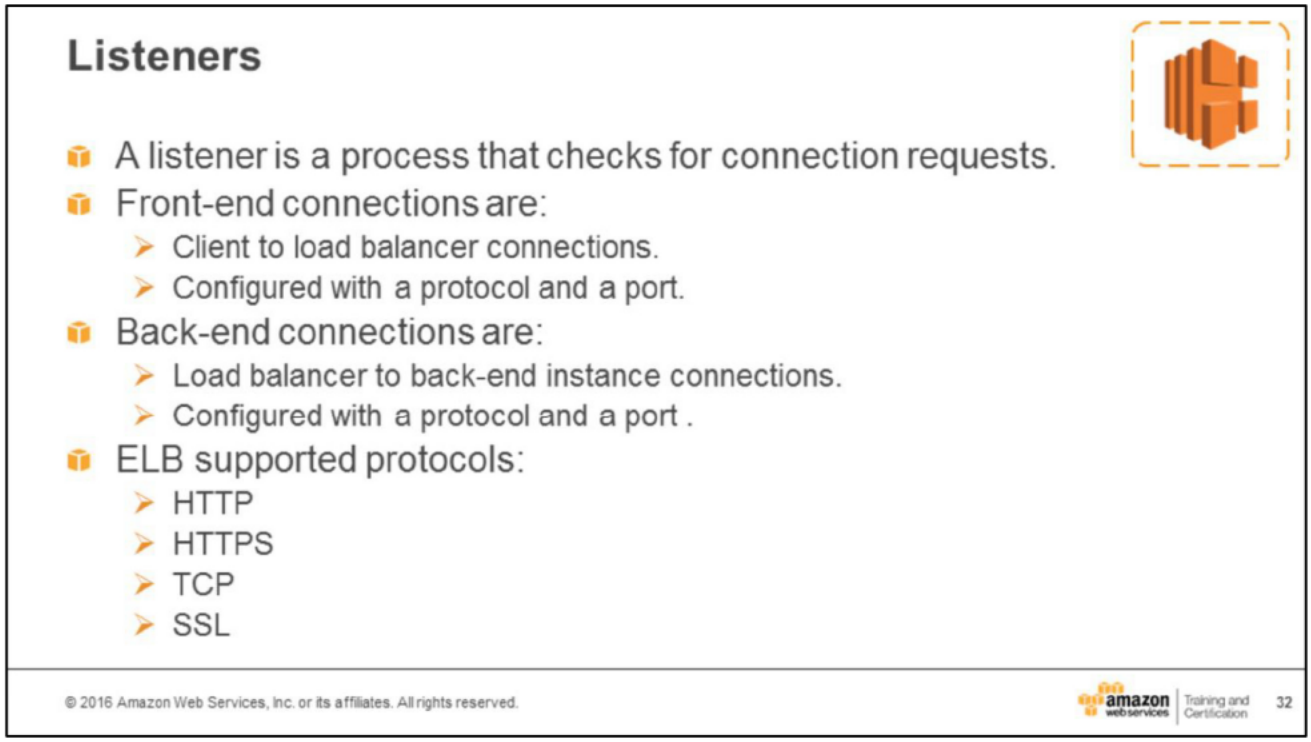
http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/scaling_plan.html#scaling_typesof



Before a client sends a request to your load balancer, it resolves the load balancer's domain name using a Domain Name System (DNS) server. The DNS entry is controlled by Amazon, because your instances are in the amazonaws.com domain. The Amazon DNS servers return one or more IP addresses to the client. These are the IP addresses of the load balancer nodes for your load balancer. As traffic to your application changes over time, Elastic Load Balancing scales your load balancer and updates the DNS entry. Note that the DNS entry also specifies the time-to-live (TTL) as 60 seconds, which ensures that the IP addresses can be remapped quickly in response to changing traffic.

The client uses DNS round robin to determine which IP address to use to send the request to the load balancer. The load balancer node that receives the request uses a routing algorithm to select a healthy instance. It uses the round robin routing algorithm for TCP listeners, and the least outstanding requests routing algorithm (favors the instances with the fewest outstanding requests) for HTTP and HTTPS listeners.

The cross-zone load balancing setting also determines how the load balancer selects an instance. If cross-zone load balancing is disabled, the load balancer node selects the instance from the same Availability Zone that it is in. If cross-zone load balancing is enabled, the load balancer node selects the instance regardless of Availability Zone. The load balancer node routes the client request to the selected instance.



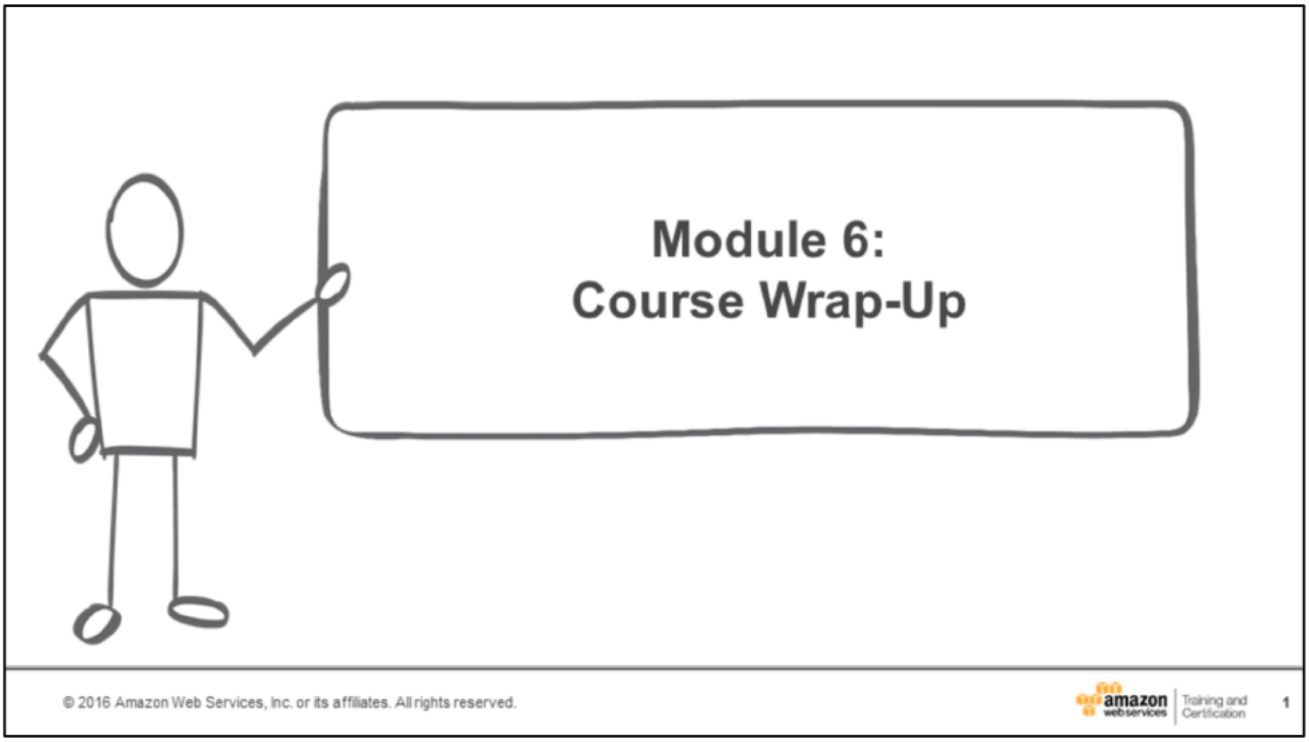
Before you start using Elastic Load Balancing, you must configure one or more listeners for your load balancer. A listener is a process that checks for connection requests. It is configured with a protocol and a port for front-end (client to load balancer) connections, and a protocol and a port for back-end (load balancer to back-end instance) connections.

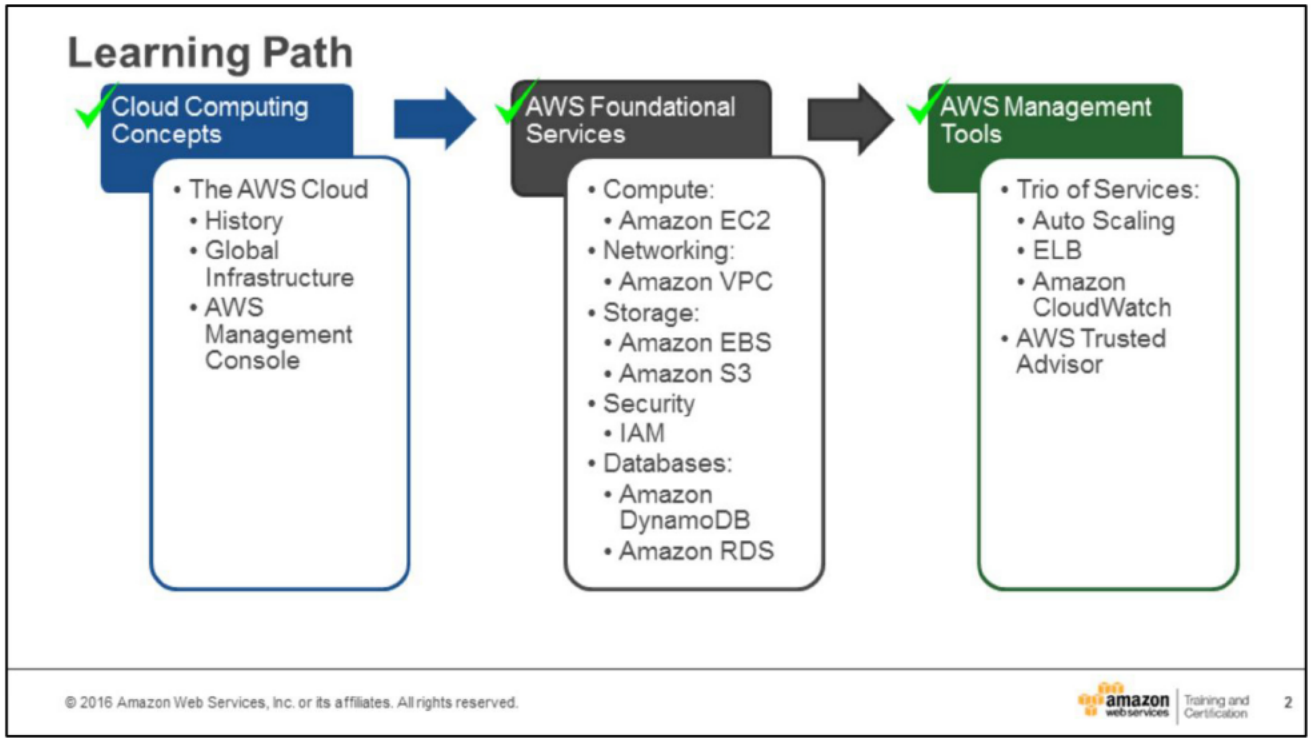
Elastic Load Balancing supports the following protocols:

- HTTP
- HTTPS (secure HTTP)
- TCP
- SSL (secure TCP)

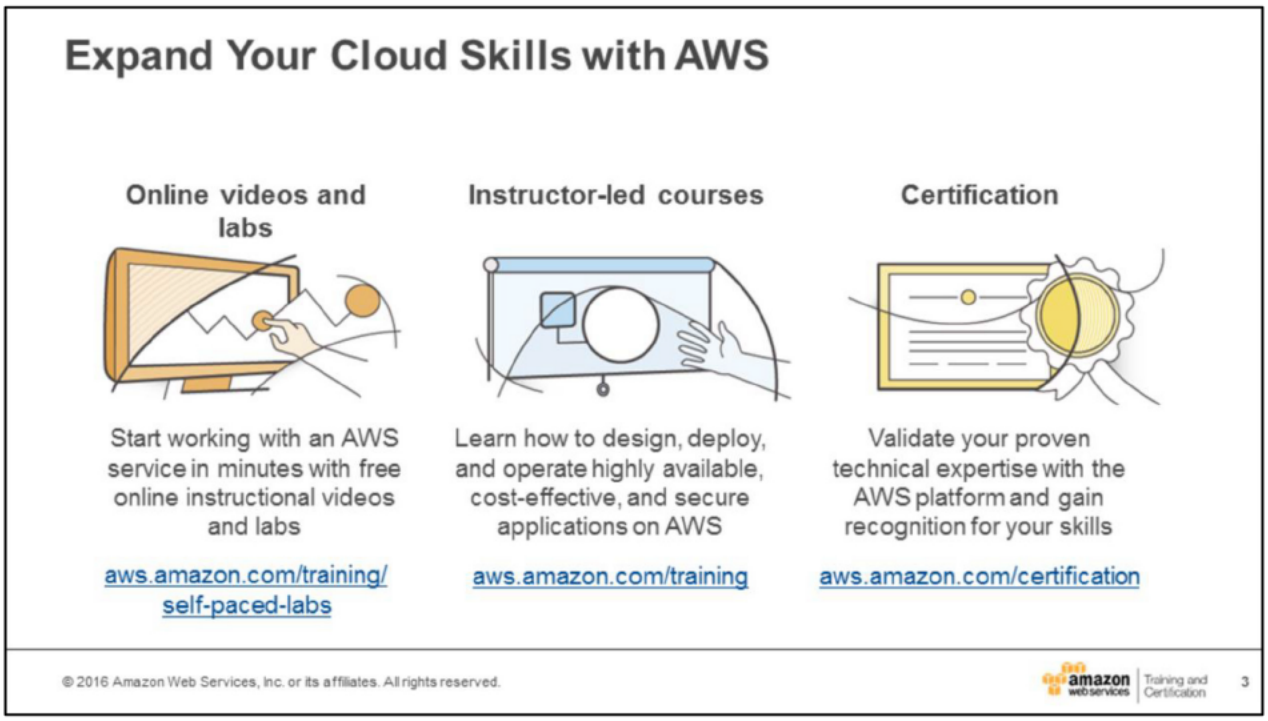
The HTTPS protocol uses the SSL protocol to establish secure connections over the HTTP layer. You can also use the SSL protocol to establish secure connections over the TCP layer.

If the front-end connection uses TCP or SSL, then your back-end connections can use either TCP or SSL. If the front-end connection uses HTTP or HTTPS, then your back-end connections can use either HTTP or HTTPS.





The slide shows the learning path of concepts we have covered today.



AWS Training and Certification is an organization dedicated to expanding and deepening knowledge of AWS, as well as driving proliferation in the usage of AWS services. Our programs are designed for customers, partners and AWS employees.

We are continuously rolling out new and updated courses, training labs, and certifications to our customers and partners.

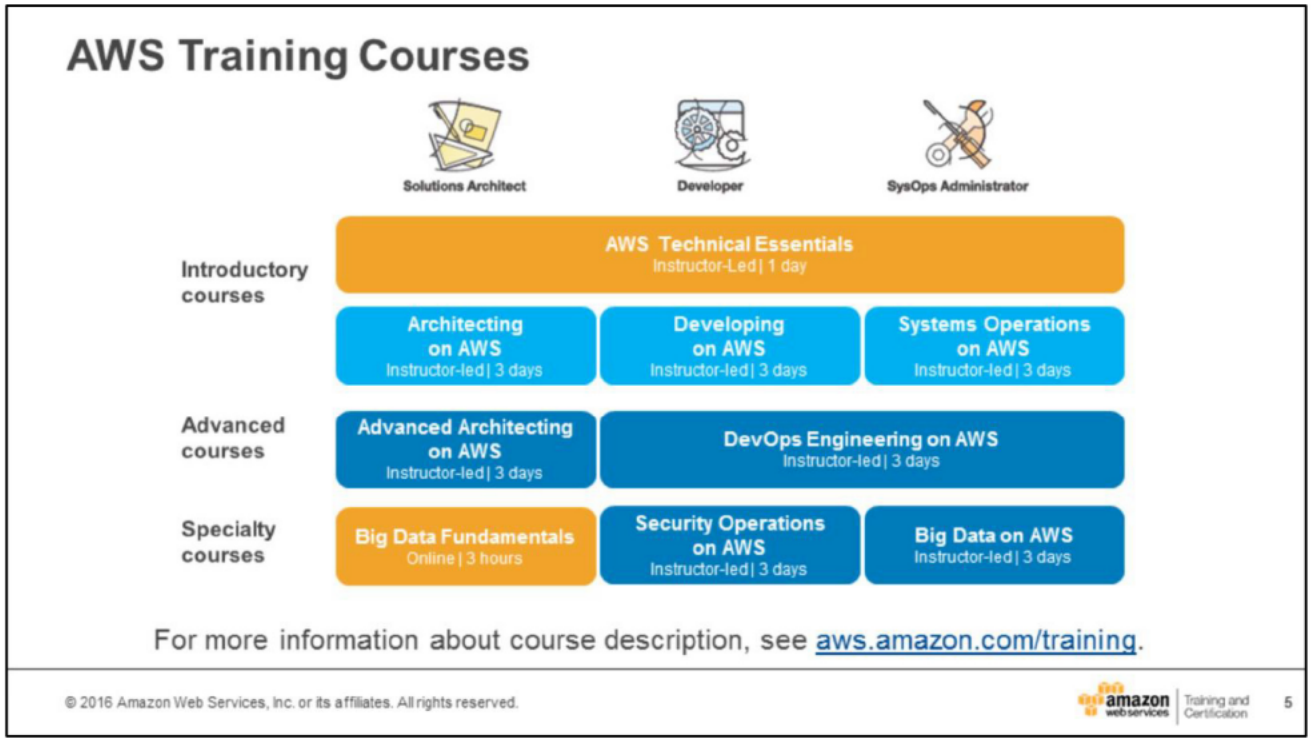
Self-Paced Labs

- Learn an individual AWS Service topic
- Follow a Learning Quest by AWS Service Area or Use Case
- Practice working with AWS as you prepare for an exam

For more information, see aws.amazon.com/training/self-paced-labs/.

amazon web services | Training and Certification 4

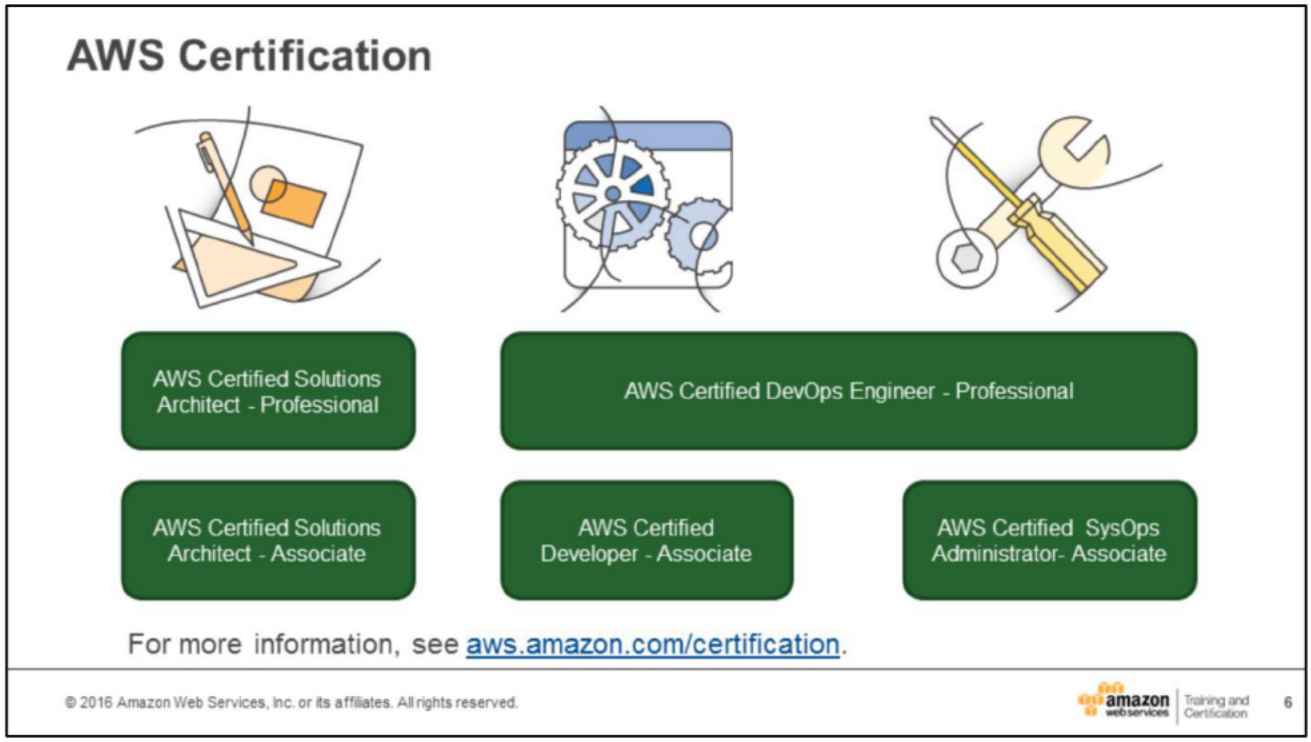
Designed by AWS subject matter experts, self-paced labs provide an opportunity to use the AWS console with step-by-step instructions, giving you hands-on practice to help you gain confidence working with AWS.



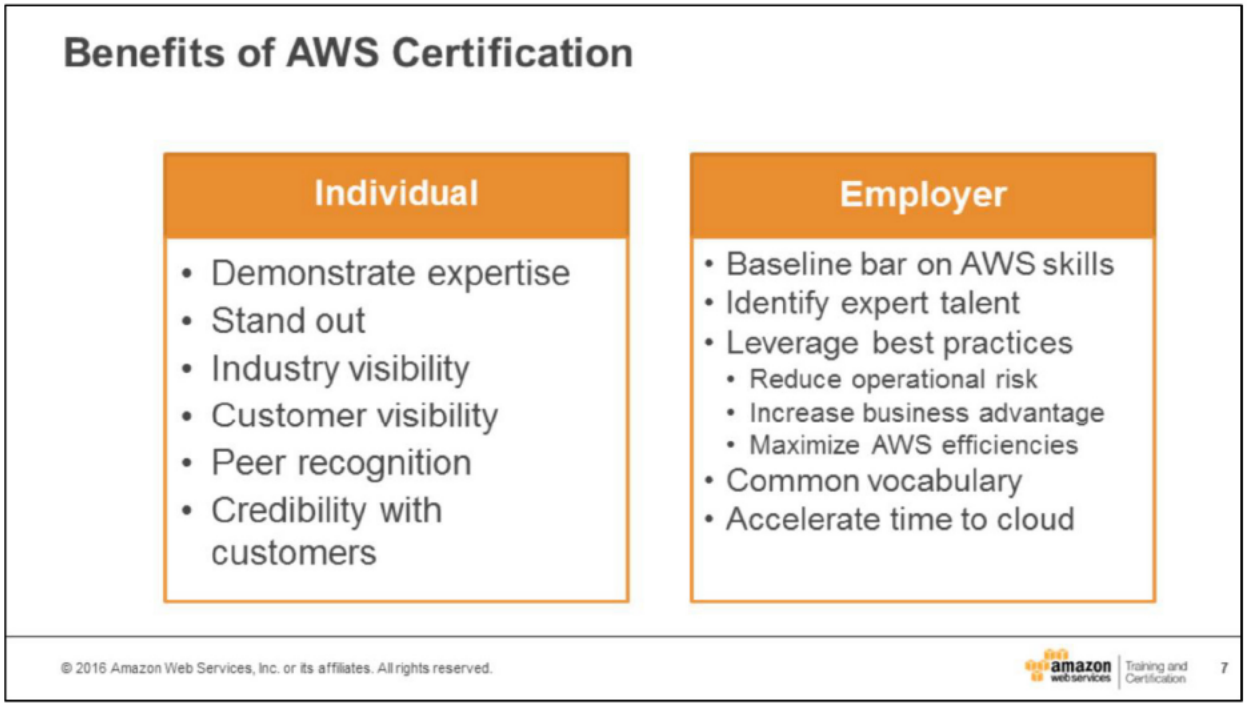
Leverage AWS Training to prepare for certification.

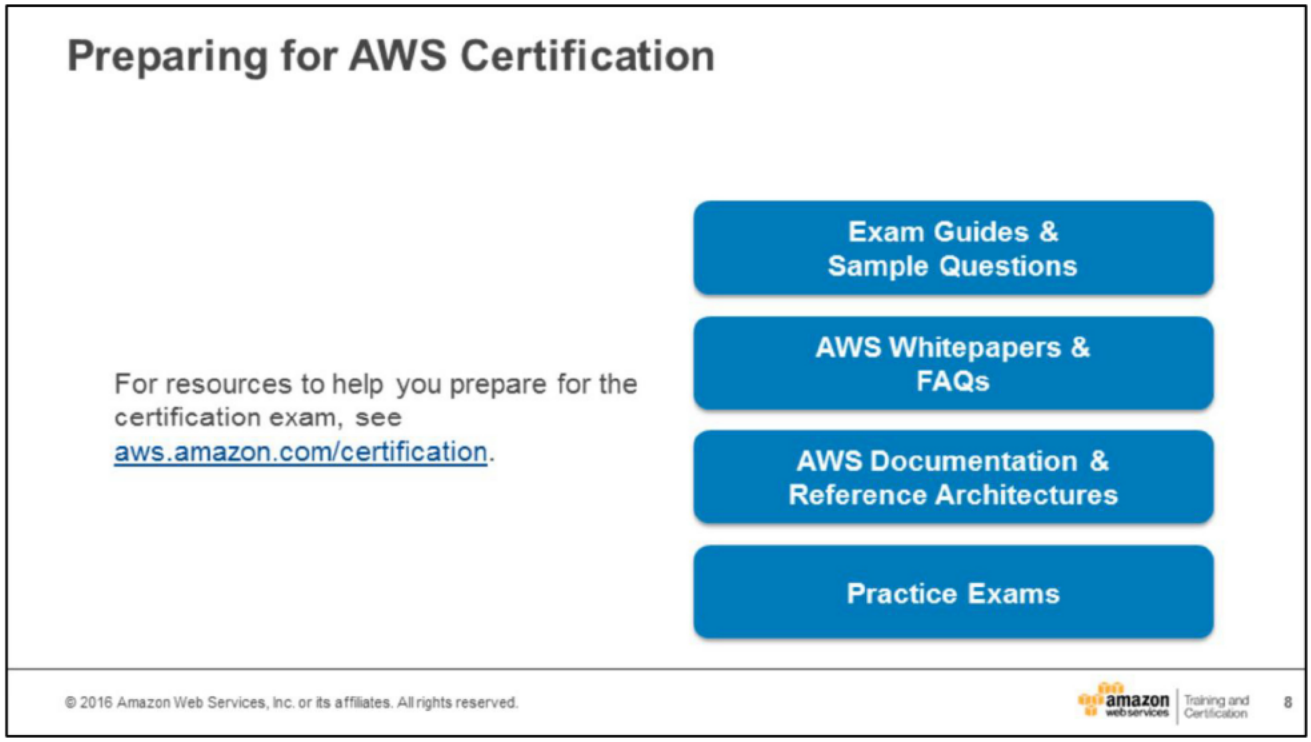
- Classes are role-based and aim to be progressive as individuals gain deeper experience working with AWS services.
    - **AWS Technical Essentials** is our one day foundational training on AWS services.
    - **Architecting on AWS** is a three-day course designed to teach Solutions Architects to design and use AWS services for common IT applications.
    - **Advanced Architecting on AWS** is a course that covers building more complex solutions which incorporate data services, infrastructure configuration management, and security on AWS.
    - **Developing on AWS** is a course designed to help individuals design and build secure, reliable and scalable AWS-based applications.
    - **Systems Operations on AWS** a course designed to help individuals operate highly available and scalable infrastructure on the AWS platform.
    - **DevOps Engineering on AWS** is a three-day advanced level course that demonstrates how to use the most common DevOps patterns to develop, deploy, and maintain applications on AWS.
    - **Big Data on AWS** is a three-day course that introduces you to cloud-based big data solutions and Amazon Elastic MapReduce (EMR), the AWS big data platform.

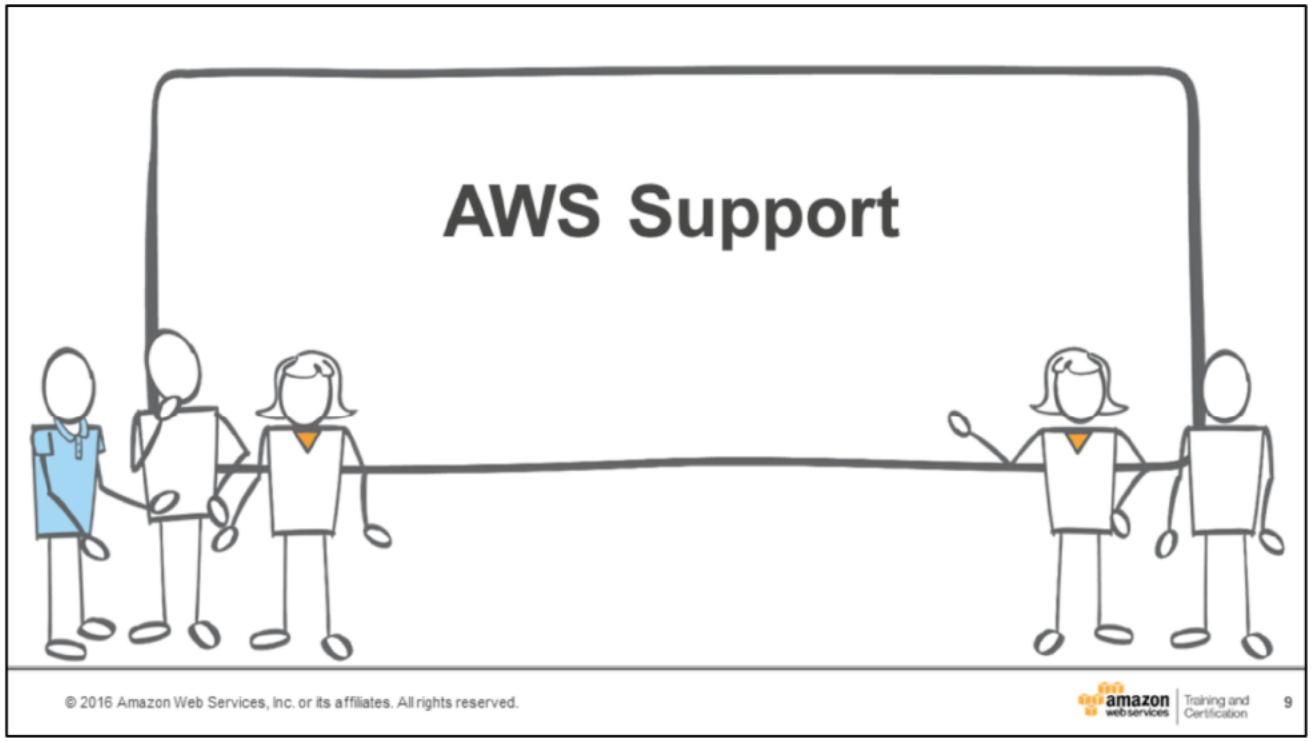
- **Security Operations on AWS** is a three-day advanced level course that demonstrates how to efficiently use AWS security services to stay secure and compliant.

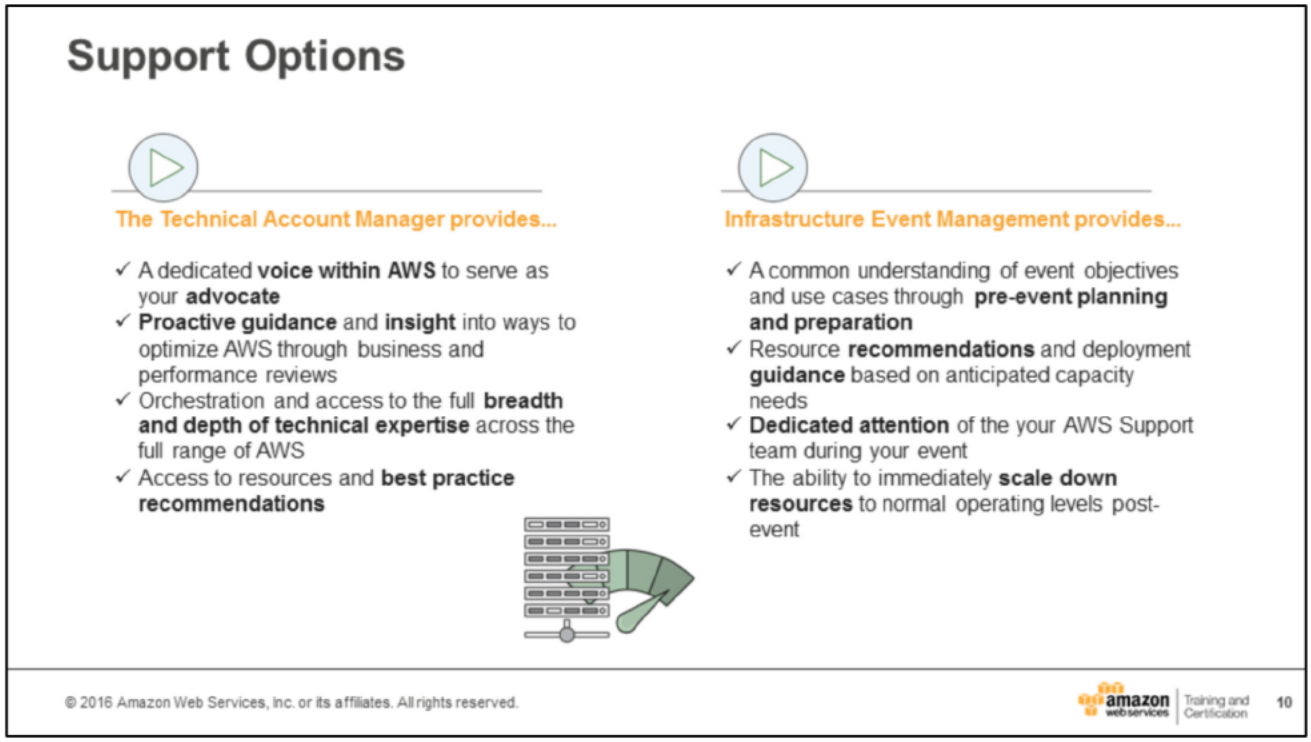


AWS certifications validate technical skills and knowledge necessary for designing, deploying, and operating solutions on the AWS platform. Earning certification helps you gain credibility for your proven experience working with AWS, as well as contributes to your organization's proficiency with AWS-based applications.





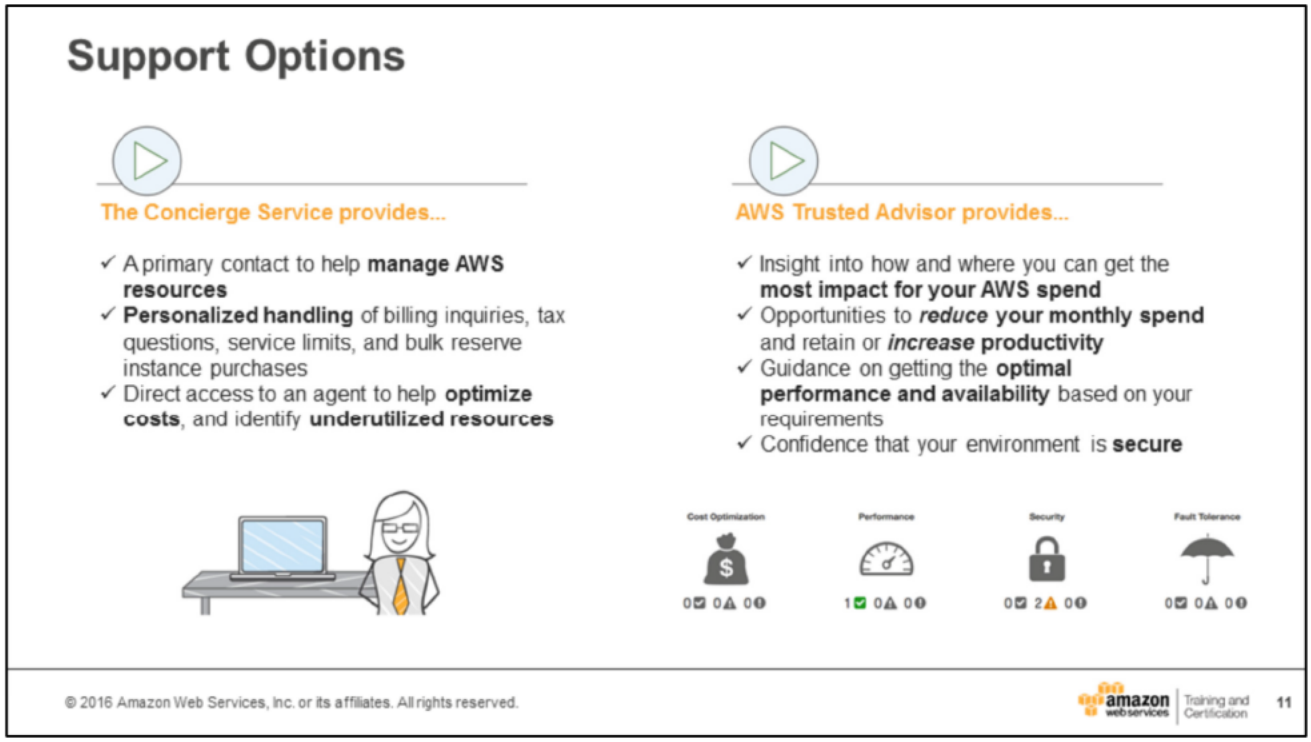




Your TAM is your primary point of contact who provides guidance and advocacy to proactively optimize your environment. The TAM sets the suite of AWS Support offerings and services in motion to prevent issues from arising - well beyond break/fix.

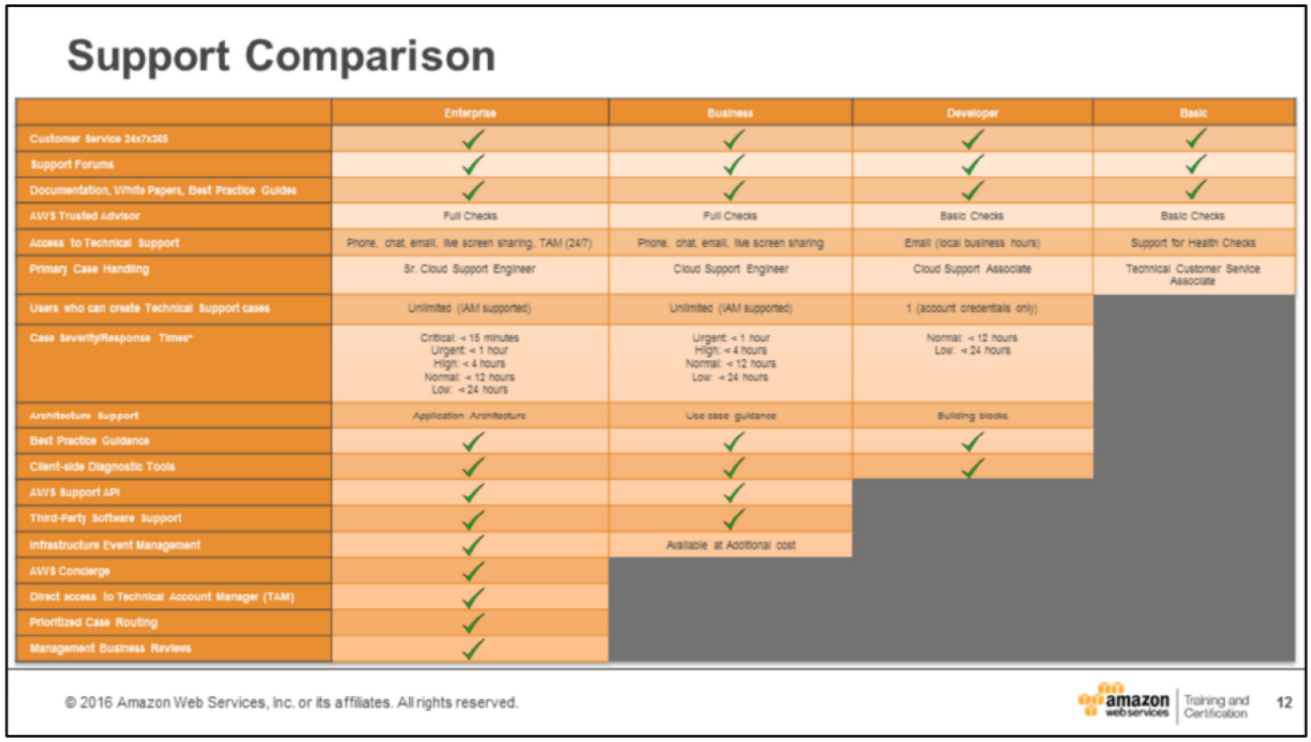
For more information, see:

https://aws.amazon.com/premiumsupport/



The Concierge Agent is a senior customer service professional who is familiar with you and your environment, and is able to quickly and efficiently address billing and account inquiries - freeing up your time to run your business.

AWS Trusted Advisor is an online resource to help you reduce cost, increase performance and fault tolerance, and improve security by optimizing your AWS environment. Developed using best practices, Trusted Advisor provides real time guidance for specific services.



Support Comparison

	Enterprise	Business	Developer	Basic
Customer Service 24x7x365	✓	✓	✓	✓
Support Forums	✓	✓	✓	✓
Documentation, White Papers, Best Practice Guides	✓	✓	✓	✓
AWS Trusted Advisor	Full Checks	Full Checks	Basic Checks	Basic Checks
Access to Technical Support	Phone, chat, email, live screen sharing, TAM (24/7)	Phone, chat, email, live screen sharing	Email (local business hours)	Support for Health Checks
Primary Case Handling	Sr. Cloud Support Engineer	Cloud Support Engineer	Cloud Support Associate	Technical Customer Service Associate
Users who can create Technical Support cases	Unlimited (IAM supported)	Unlimited (IAM supported)	1 (account credentials only)	
Case Severity/Response Times*	Critical: < 15 minutes Urgent: < 1 hour High: < 4 hours Normal: < 12 hours Low: < 24 hours	Urgent: < 1 hour High: < 4 hours Normal: < 12 hours Low: < 24 hours	Normal: < 12 hours Low: < 24 hours	
Architecture Support	Application Architecture	Use case guidance	Building blocks	
Best Practice Guidance	✓	✓	✓	
Client-side Diagnostic Tools	✓	✓	✓	
AWS Support API	✓	✓		
Third-Party Software Support	✓	✓		
Infrastructure Event Management	✓	Available at Additional cost		
AWS Concierge	✓			
Direct access to Technical Account Manager (TAM)	✓			
Prioritized Case Routing	✓			
Management Business Reviews	✓			

amazon web services | Training and Certification 12

Case Severity & Response Times

	Critical	Urgent	High	Normal	Low
Enterprise Plan (24 x 7)	15 minutes or less	1 hour or less	4 hours or less	12 hours or less	24 hours or less
Business Plan (24 x 7)		1 hour or less	4 hours or less	12 hours or less	24 hours or less
Developer Plan (Business hours)				12 hours or less	24 hours or less

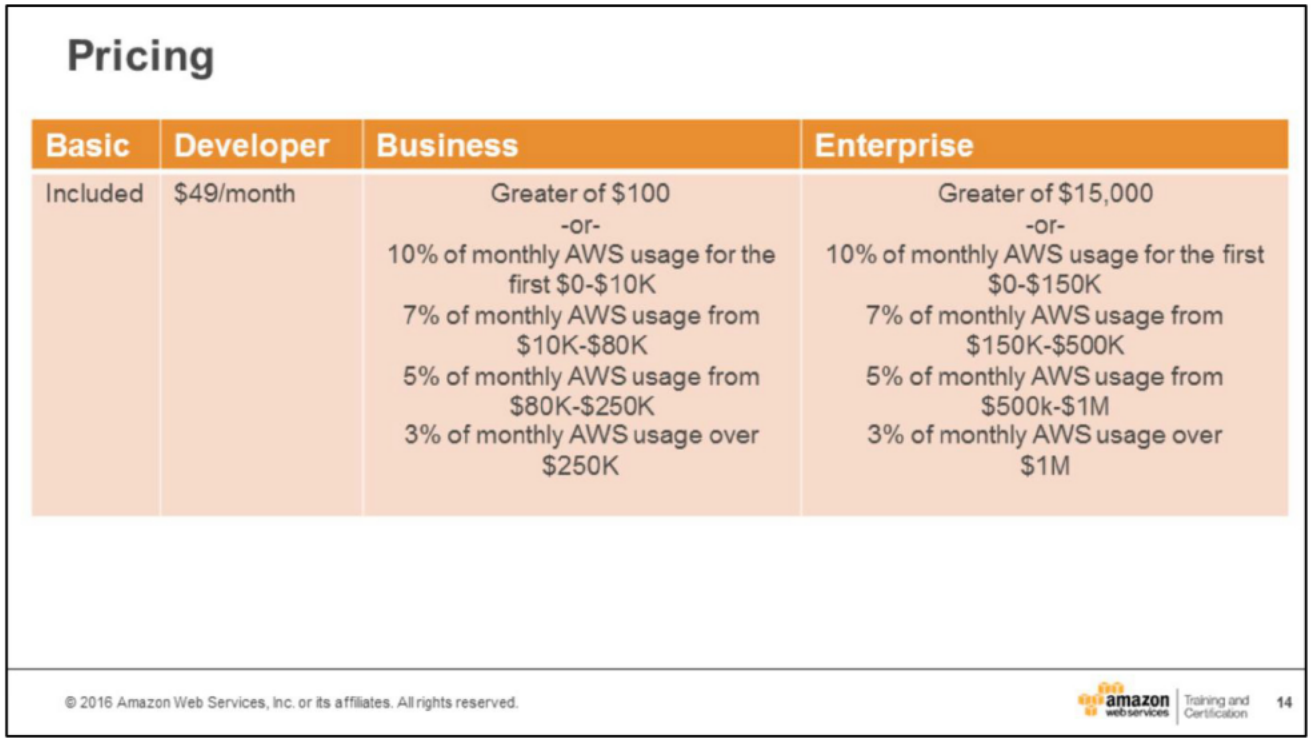
Critical: Your business is at risk. Critical functions of your application are unavailable.

Urgent: Your business is significantly impacted. Important functions of your application are unavailable.

High: Important functions of your application are impaired or degraded.

Normal: Non-critical functions of your application are behaving abnormally, or you have a time-sensitive development question.

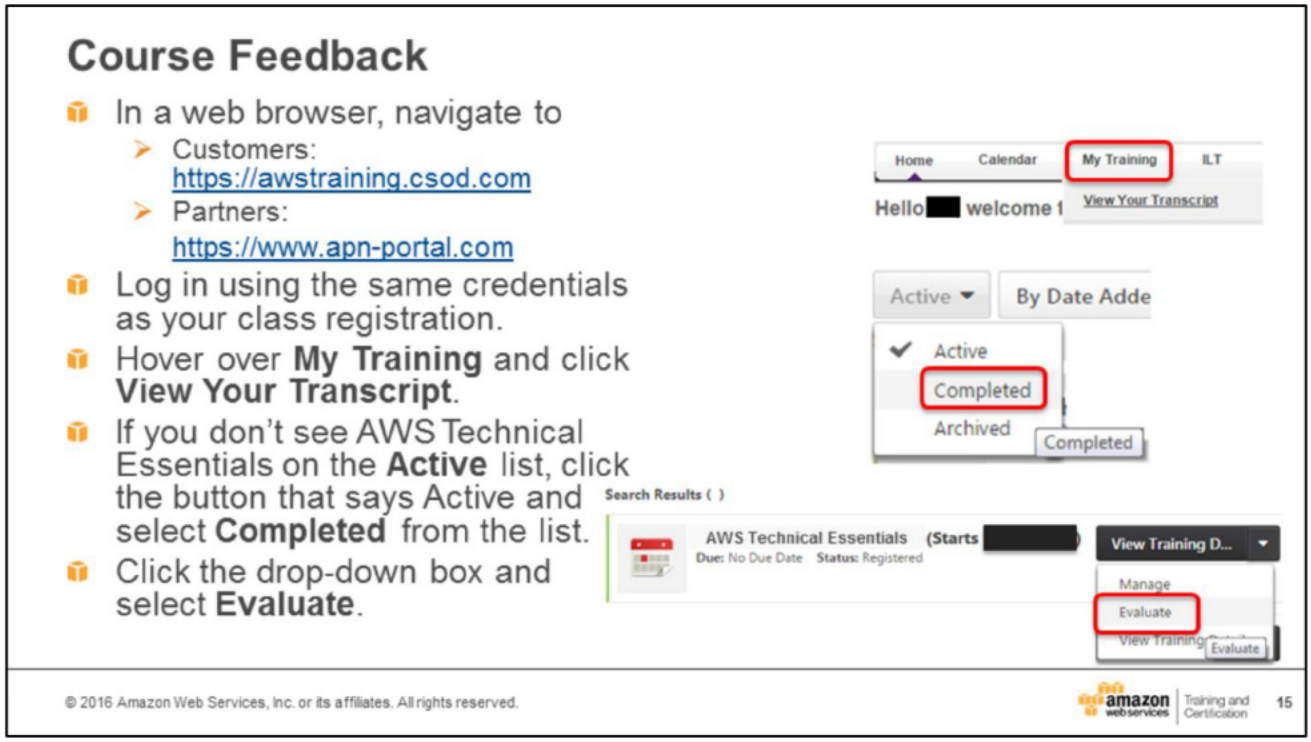
Low: You have a general development question, or you want to request a feature.



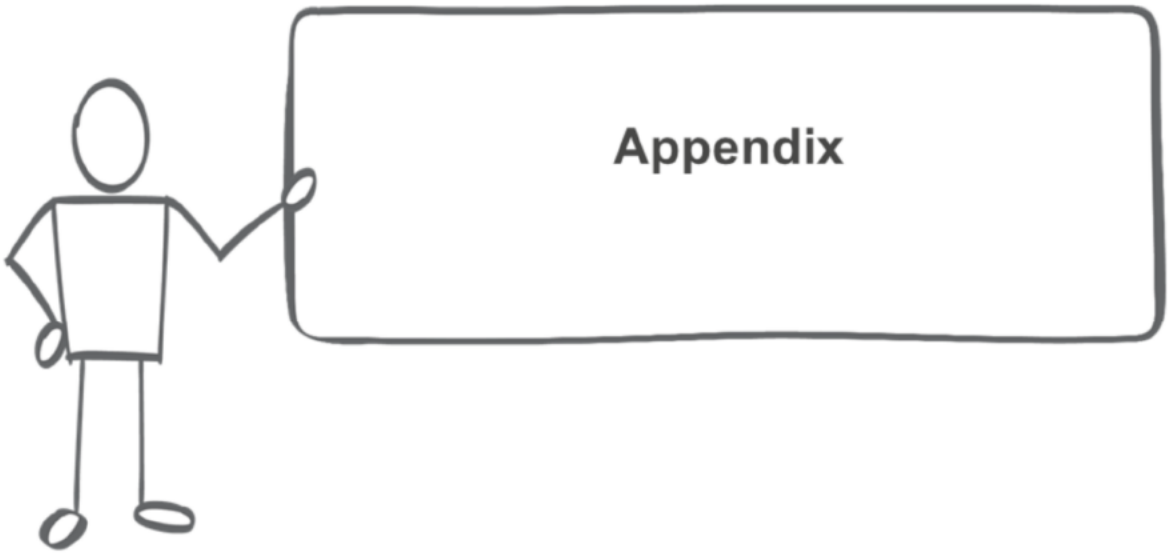
All AWS Support tiers include an unlimited number of support cases, with no long-term contracts. Also, with the Business and Enterprise-level tiers, as your AWS charges grow, you earn volume discounts on your AWS Support costs.

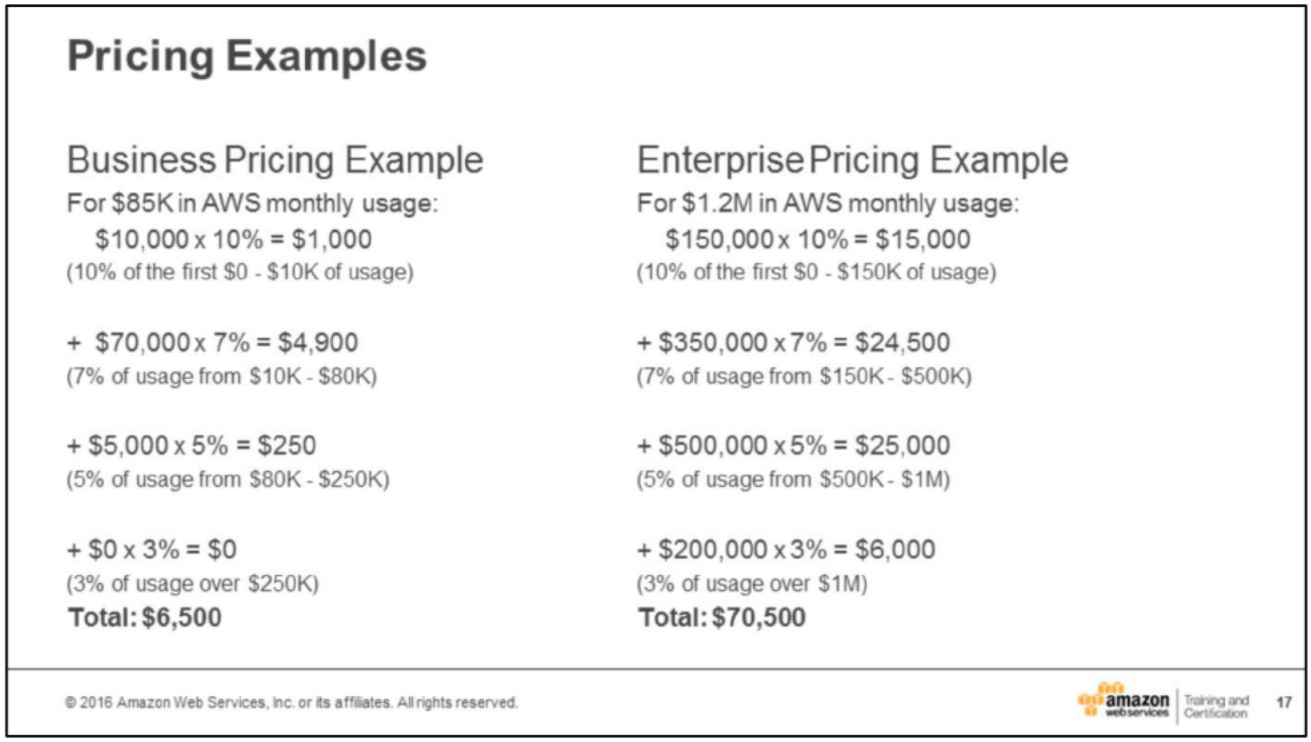
For more information on estimating your deployment's AWS Support cost, see:

http://calculator.s3.amazonaws.com/index.html



You will also receive an email with a link to the course evaluation upon completion of the course.





Examples of business and enterprise pricing are shown on the slide.