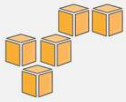


CCA Unit 2 – Getting Started with AWS

CCA 2.01: AWS Compute Storage & Networking

CCA 2.01 AWS Compute, Storage, and Networking
CCA 2.02 AWS Security, Identity, and Access Management
CCA 2.03 AWS Database Options
CCA 2.04 AWS Elasticity and Management Tools

amazon web services | Academy

This module begins CCA Unit 2 – Getting Started with AWS. Before starting Unit 2, you should have completed Unit 1.

Unit 2 also includes the following labs:

CCA Lab-01 - Creating an EC2 instance with Microsoft Windows

CCA Lab-02 - Build Your VPC and Launch a Web Server

CCA Lab-03 - Working with EBS

CCA Lab-04 - Introduction to AWS IAM

CCA Lab-05 - Build Your Database Server and Interact with Your Database using an Application

CCA Lab-06 - Scale and Load Balance Your Architecture

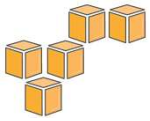
We'll start now with the first module, CCA 2.01: AWS Infrastructure.

What's In This Module?

- Global Infrastructure Review
- Compute Services Using Amazon EC2
- Virtual Private Networks (VPNs)
- Storage Services
  - Amazon Simple Storage Service (S3)
  - Edge Locations, Route 53, CloudFront

This module covers…

- Global Infrastructure Review
- Compute Services Using Amazon EC2
- Virtual Private Networks (VPNs)
- Storage Services
  - Amazon Simple Storage Service (S3)
  - Edge Locations, Route 53, CloudFront



Part 1

Global Infrastructure Review

amazon web services | Training and Certification

Part 1: Global Infrastructure Review

Regions and Availability Zones

Categorize each of the following:

	Region	Availability Zone
Associated with a global/geographic location:	✓	
Consists of clustered data centers:		✓
Connected by a low-latency link:		✓
Isolated from failures:	✓	✓
For high availability, provision EC2s across multiple…		✓
An EC2 can be used only in the ____ where you created it:	✓	

Recall from Unit 1 –

What is a Region? Geographic locations that contain multiple Availability Zones (AZs).

What is an AZ? Availability Zones consist of data centers clustered in a region. Each Availability Zone is engineered to be isolated from failures in other Availability Zones. 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. 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.

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/

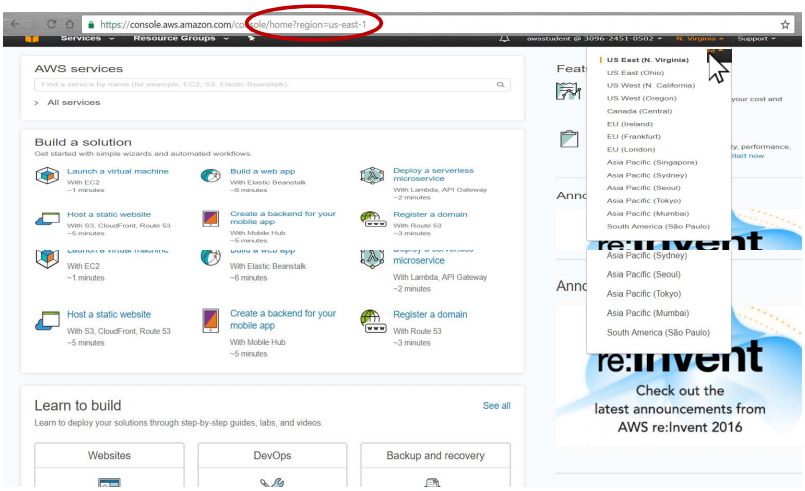
Regions and Availability Zones

Q: When you launch an EC2 instance, what happens if you don't select a Region?

A: The instance will launch in the region where you're logged-in to the management console.

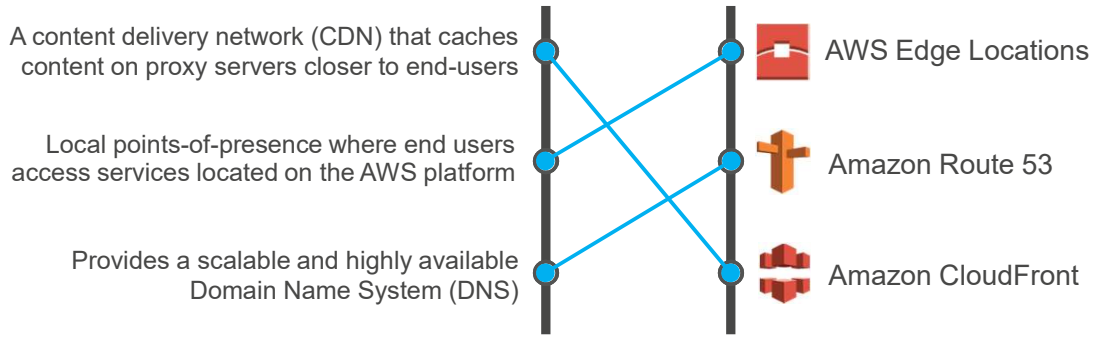
Q: When you launch an EC2 instance, what happens if you don't select an AZ?

A: AWS will choose one for you



Edge Locations

Match the following:

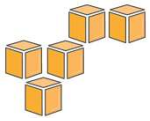


We also introduced AWS Edge Locations, Amazon Route 53 and Amazon CloudFront.

An **edge location** is where end users access services located on the AWS platform. They are located in most of the major cities around the world and provide a point-of-presence using CloudFront (CDN) to distribute content to end users with reduced latency. An AWS edge location hosts a robust content delivery network that can be used to deliver entire web sites and dynamic, static, and streaming content. Requests for content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

Route 53 provides a scalable and highly available Domain Name System (DNS). (Side-note: The name is a reference to TCP or UDP port 53, where DNS server requests are addressed.)

CloudFront is a content delivery network (CDN) providing a globally-distributed network of proxy servers which cache content closer to end-users to minimize rates downloading content.



Part 2

Amazon Elastic Compute Cloud (EC2)

amazon web services | Training and Certification

Part 2: Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2)

Amazon EC2

- **Resizable** compute capacity
- Complete control of your computing resources
- **Reduced time required** to obtain and boot new server instances

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

Amazon EC2 is designed to make web-scale computing easier for developers. With Amazon EC2's simple web service interface, you can 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, which helps 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 Facts

- **Scale capacity** as your computing requirements change
- Pay only for capacity that **you actually use**
- Choose **Linux** or **Windows**
- Deploy across **AWS Regions** and **Availability Zones** for reliability
- Use **tags** to help manage your Amazon EC2 resources

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 systems as you need.

You can 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.

For more information, see:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html

Launching an Amazon EC2 Instance via the Management Console

1. **Determine the AWS Region** in which you want to launch the Amazon EC2 instance.
2. **Launch** an Amazon EC2 instance from a pre-configured Amazon Machine Image (AMI).
3. **Choose an instance type** based on CPU, memory, storage, and network requirements.
4. **Configure** network, IP address, security groups, storage volume, tags, and key pair.

Before you create your first Amazon EC2 instance, decide which region you want to have that instance in. The AMI comes pre-installed with many AWS API tool and with 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.

Amazon Machine Image (AMI) Details

An AMI includes the following:

- A template for the **root volume** for the instance (for example, an operating system, an application server, and applications).
- **Launch permissions** that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the **volumes to attach** to the instance when it is launched.

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.

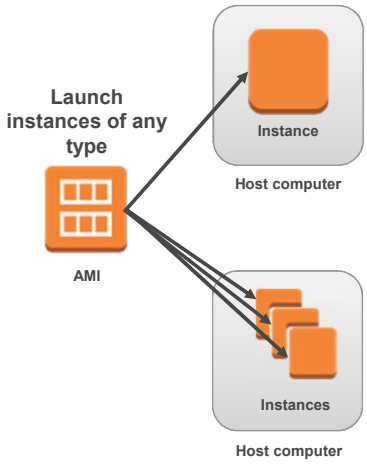
You can create your own AMI by customizing the instance that you launch from a public AMI and then saving the configuration as a custom AMI for your own use. You can also buy, share, and sell AMIs.

For more information, see:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

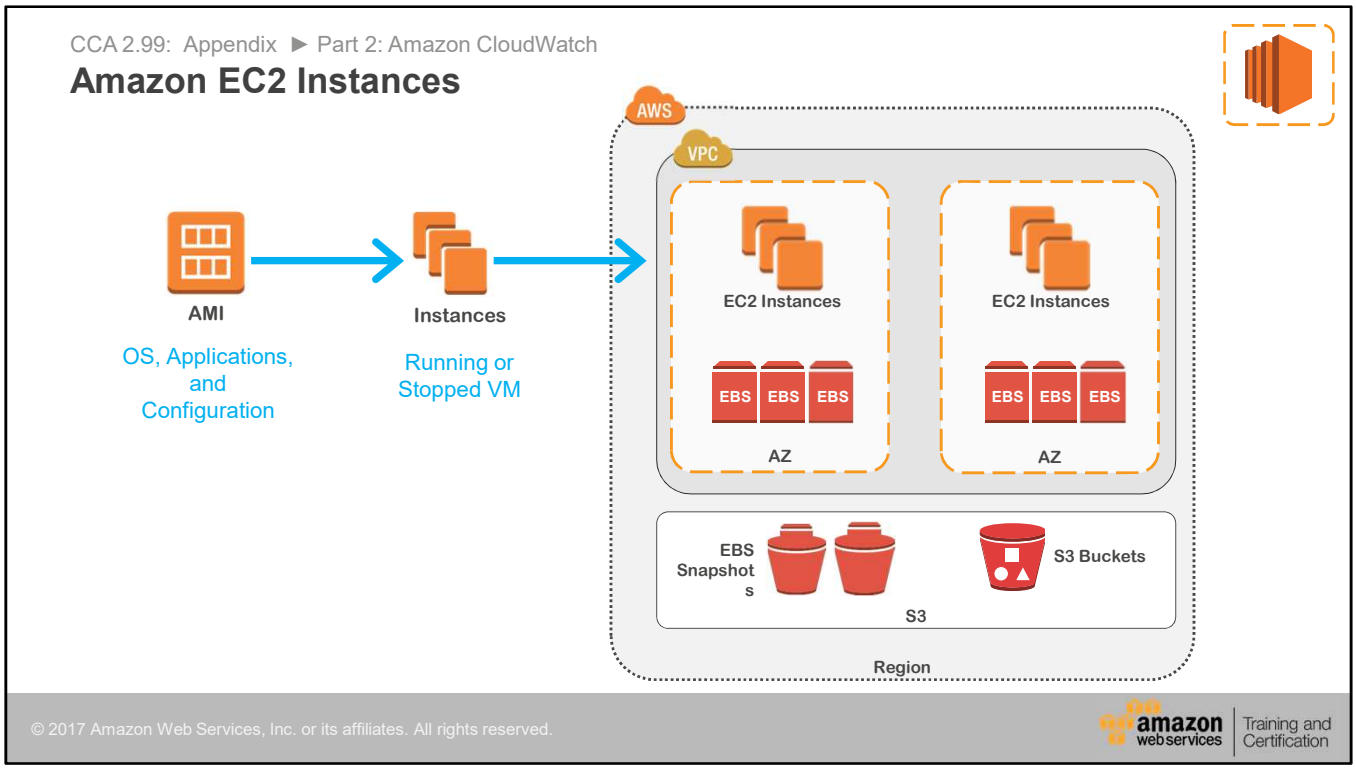
Instances and AMIs

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).

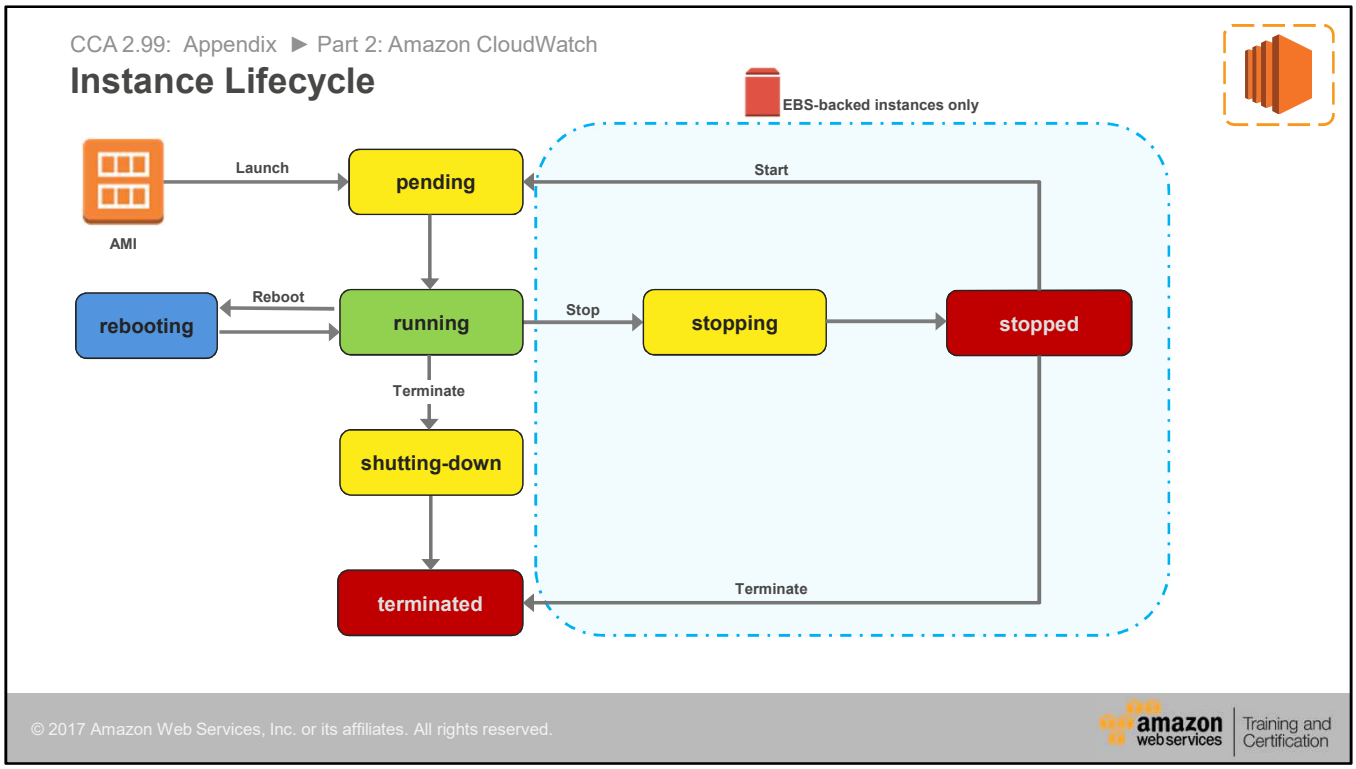


You can launch multiple instances of different types from a single AMI when launching an EC2 instance. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on 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 within a region. You can configure security and network access on your Amazon EC2 instance.

Customers can deploy to multiple Availability Zones 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 (EBS) volumes in each Availability Zone. Determine whether you want to run in multiple locations, use 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, such as instance-hours or data transfer.



The slide shows the lifecycle of an instance launched from an AMI. Note that you can only stop and start instances that are 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. When 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. When 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 instead of 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 that you no longer need an instance, you

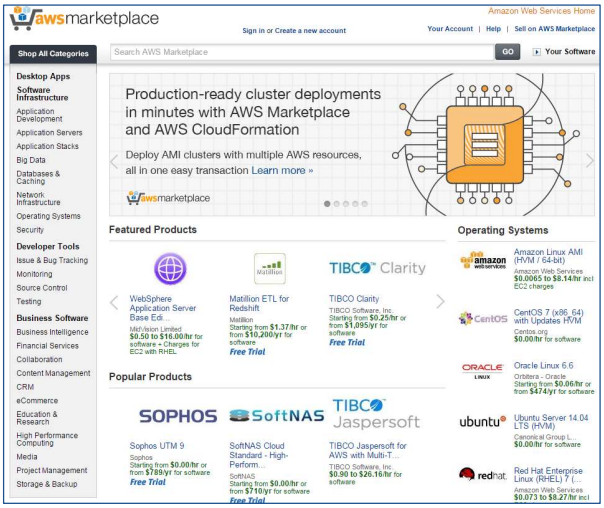
can terminate it. The instance will enter the shutting-down state. You will stop incurring charges as soon as the instance enters shutting-down or terminated states.

- **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 Marketplace – IT Software Optimized for the Cloud

- Online store to discover, purchase, and deploy IT software on top of the AWS infrastructure.
- Catalog of **2700+** IT software solutions including Paid, BYOL, Open Source, SaaS, and free-to-try options.
- Pre-configured to operate on AWS.
- Software checked by AWS for security and operability.
- Deploys to AWS environment in minutes.
- Flexible, usage-based billing models.
- Software charges billed to AWS account.

Includes AWS Test Drive.
https://aws.amazon.com/marketplace



AWS and Oracle have worked together to offer customers convenient options for deploying enterprise applications in the cloud. Customers can build enterprise-grade solutions hosted by Amazon Web Services using database and middleware software by Oracle, and can also launch entire enterprise software stacks from Oracle on Amazon EC2.

New and existing SAP customers can confidently deploy their SAP solutions on SAP-certified Amazon EC2 instances in production environments because 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. Available AMIs allow 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
- Expedited deployment cycles
- Optimized software capacity
- Matched spend to actual usage
- Trust vetted and scanned products

AWS Test Drive provides a private IT sandbox environment that contains preconfigured server based solutions. In under an hour, and using a step-by-step lab manual and video, you can launch, sign in, and learn about these popular third-party IT solutions powered by AWS and AWS CloudFormation.

For more information, see: https://aws.amazon.com/marketplace

Choosing the Right Amazon EC2 Instance

AWS uses Intel® Xeon® processors to provide customers with high performance and value. EC2 instance types are optimized for different use cases, workload requirements and come in multiple sizes.

Consider the following when choosing your instances:

- Core count
- Memory size
- Storage size and type
- Network performance
- CPU technologies

Choosing the right EC2 instance type matters.
AWS has a wide variety of EC2 compute instances. Most are based on Intel Xeon processors.
Intel® Xeon processors are compatible with the broadest range of ecosystem software, which helps developers to develop their application rapidly. IT also values Xeon processors for higher performance and reliability.

Each instance type or family is optimized for different workloads or use cases. Within each type or family we have multiple sizes: Large, XLarge, 2XLarge, etc.

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 Intel processor technologies.

Larger instances are better for workloads that scale.

Get the Intel® Advantage

Intel's Haswell microarchitecture on new X1, C4, D2, and M4 instances, with custom Intel® Xeon® v3 processors, provides new features:

Haswell microarchitecture can boost existing applications performance by **30% or more** for better workload performance and faster response times.

Newer Hardware Assisted technologies, such as Intel® AVX2.0 instructions, can double the floating-point performance for compute-intensive workloads and provide additional instructions for compression and encryption

In 2016, AWS launched the C4, D2, and M4 instances based on Intel's latest 22nm Haswell microarchitecture. They all use custom Intel® Xeon® v3 processors designed and built especially for AWS.

X1 is based on Intel E7 enterprise class CPU.

Haswell can boost application performance, which helps workloads to perform better or deliver faster response time. Haswell instances also include newer hardware-assisted technologies for workload acceleration, such as AVX 2.0, which doubles the floating point performance compared to AVX 1.0 for high performance compute (HPC) and financial modeling kinds of workload.

Intel® Processor Technologies

Intel® AVX: Provides dramatically better performance for highly parallel HPC workloads such as *life science engineering, data mining, financial analysis*, or other technical computing applications. AVX also enhances *image, video, and audio* processing.

Intel® AES-NI: Enhance your security with these new encryption instructions that reduce the performance penalty associated with encrypting/decrypting data.

Intel® Turbo Boost Technology: Provides more computing power when you need it with performance that adapts to spikes in your workload.

Intel Transactional Synchronization (TSX) Extensions: Enable execution of transactions that are independent to accelerate throughput.

P state & C state control: Gives you the ability to individually tune each cores performance & sleep states to improve application performance.

In addition to the new X1, C4, D2, and M4 instance families based on Haswell, with its host of new microarchitecture features, our other EC2 instances are based on Intel Xeon processors. These Xeon processors have important technology features that you should be aware of.

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 penalty you would pay by using encryption.

Intel® Turbo Boost Technology automatically gives you more computing power when your workloads are not fully using all CPU cores. Think of it as automatic overclocking when you have thermal headroom.

Intel Transactional Synchronization (TSX) Extensions enable execution of transactions that are independent in order to accelerate throughput. Intel® TSX delivers outstanding performance gains for multi-threaded applications that are developed using a coarse-grained lock (a single lock for the entire hash table) that behaves like a fine-grained lock (multiple locks for smaller table sections). Coarse-grained locks are easier to use, easier to understand, and easier to debug. In summary, Intel TSX provides a set of instruction set extensions that allow programmers to specify regions of code for transactional synchronization.

Programmers can use these extensions to achieve the performance of fine-grained locking while actually programming using coarse-grained locks.

P state and C state control on the 8xlarge (when a user has the whole 2S machine) provides the ability to individually tune each core's performance and sleep states to improve application performance.

AWS EC2 Instances with Intel® Technologies

AWS Instance Type	High Memory X1	Compute-Optimized C4	Storage-Optimized D2	General Purpose M4	Memory-Optimized R3	IO-Optimized I2	Graphics-Optimized G2	Burstable Performance T2
Intel Processor	Intel Xeon E7-8880 v3	Custom Intel Xeon E5-2666 v3	Custom Intel Xeon E5-2676 v3	Custom Intel Xeon E5-2676 v3	Intel Xeon E5-2670 v2	Intel Xeon E5-2670 v2	Intel Xeon E5-2670	Intel Xeon Family
Intel AVX	AVX 2.0	AVX 2.0	AVX 2.0	AVX 2.0	Yes	Yes	Yes	Yes
Intel AES-NI	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Intel Turbo Boost	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Intel TSX	Yes	No	No	No	No	No	No	No
Per core P- and C-state control	No	Yes (8xlarge only)	No	No	No	No	No	No
SSD Storage	EBS Optimized by default	EBS Optimized by default	No	EBS Optimized by default	Yes	Yes	Yes	EBS only

This chart provides you with a quick reference to the individual Intel® technologies that were explained on the previous slide and to which EC2 instance contains each Intel technology.

For each instance, a few key technologies are identified to end customers because they have significant impact on the performance of some workloads. More details are discussed later in the presentation, but for now, be aware that different instances have different instructions and capabilities that will affect performance beyond just frequency.

Current Generation Instances

Instance Family	Some Use Cases
General purpose (t2, m4, m3)	• Low-traffic websites and web applications • Small databases and mid-size databases
Compute-optimized (c4, c3)	• High performance front-end fleets • Video-encoding
Memory-optimized (r3)	• High performance databases • Distributed memory caches
Storage-optimized (i2, d2)	• Data warehousing • Log or data-processing applications
GPU instances (g2)	• 3D application streaming • Machine learning

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 several 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/

Instance Metadata

Is **data** about your **instance**.

Can be used to **configure or manage** a running instance.

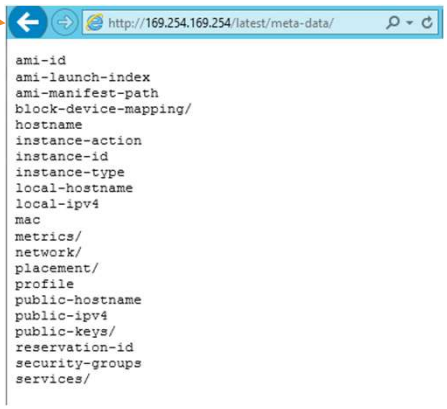
Instance metadata is divided into categories.

For more information, see:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html#instancedata-data-categories

Retrieving Instance Metadata

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```



On a Linux instance, you can use:

```
$ curl http://169.254.169.254/latest/meta-data/
$ GET http://169.254.169.254/latest/meta-data/
```

All metadata is returned as text (content type text/plain).

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 that are used to retrieve instance metadata and user data.

For more information, see:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

Instance User Data

Can be passed to the instance **at launch**.

Can be used to perform common **automated configuration tasks**.

Runs scripts after the instance starts.

Instance user data can be used to build more generic AMIs that can be modified by configuration files supplied at launch time. It is important to note that, 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.

Adding User Data

You can specify user data when launching an instance.

User data can be:

- Linux script – executed by **cloud-init**
- Windows batch or PowerShell scripts – executed by **EC2Config** service

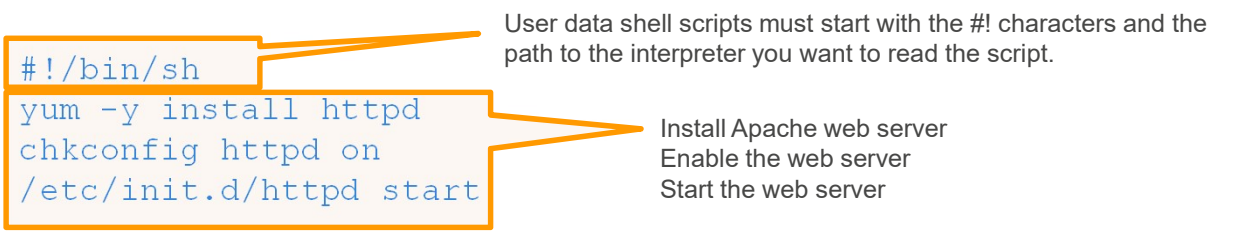
User data scripts run once per instance ID by default.

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, such as Amazon EC2.

User data information:

- User data is treated as opaque data: what you give is what you get back. The instance is responsible for interpreting user data.
- 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.

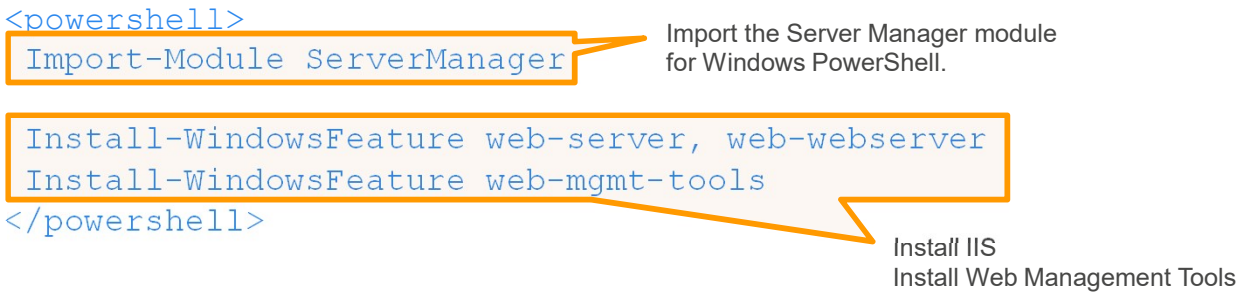
User Data Example Linux



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

User Data Example Windows



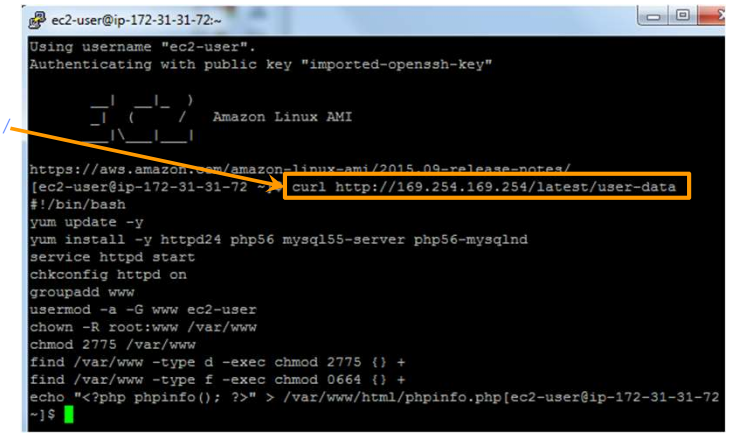
You can send user data to a Windows instance with a PowerShell script (shown in slide) or with a set of Windows batch commands.

Retrieving User Data

To retrieve user data, use the following URI:
`http://169.254.169.254/latest/user-data`

On a Linux instance, you can use:

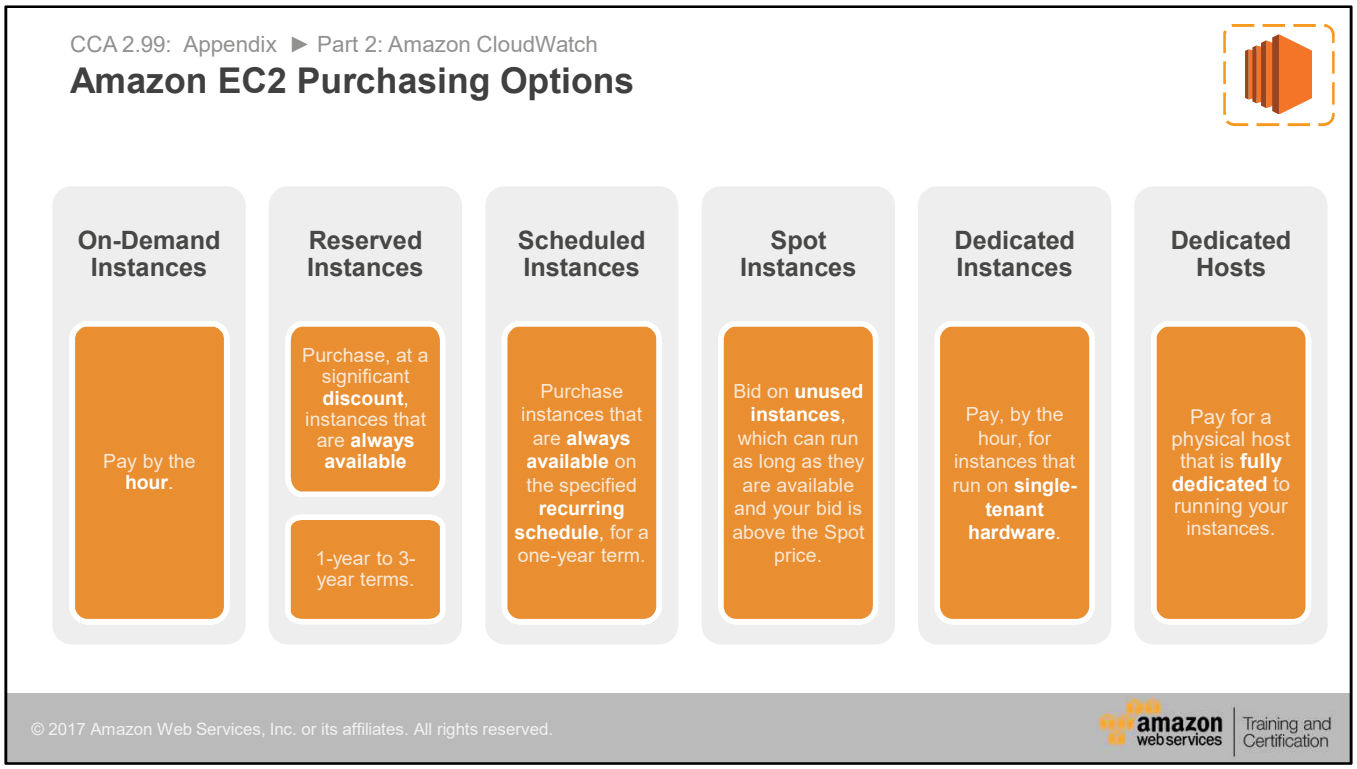
```
$ curl http://169.254.169.254/latest/user-data/
$ GET http://169.254.169.254/latest/user-data/
```



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. They have 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 provide 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, which allows 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, and Linux Enterprise Server. 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.

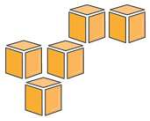
Dedicated instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. Your Dedicated Instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances

For more information, see:

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

CCA Lab-01:

Creating an EC2 instance with Microsoft Windows



Part 3

Networking – Amazon VPC

amazon web services | Training and Certification

Part 3: Networking – Amazon VPC

Amazon Virtual Private Cloud (VPC)

Amazon VPC

- Provision a **private, isolated virtual network** on the AWS cloud.
- Have complete control over your virtual networking environment.

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.

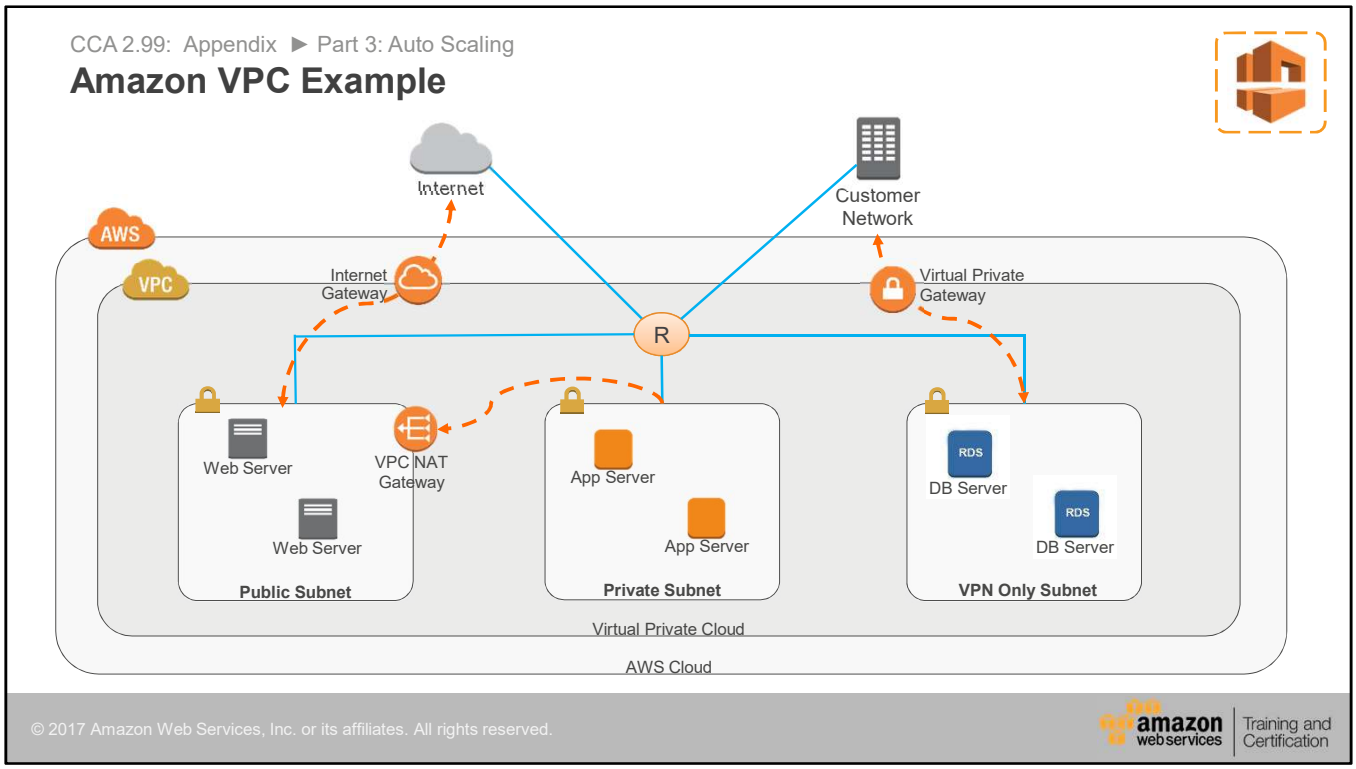
VPCs and Subnets

- A **subnet** defines a range of IP addresses in your VPC.
- You can launch AWS resources into a subnet that you select.
- A **private subnet** should be used for resources that won't be accessible over the Internet.
- A **public subnet** should be used for resources that will be accessed over the Internet.
- Each subnet must reside entirely within one Availability Zone and cannot span zones.

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).

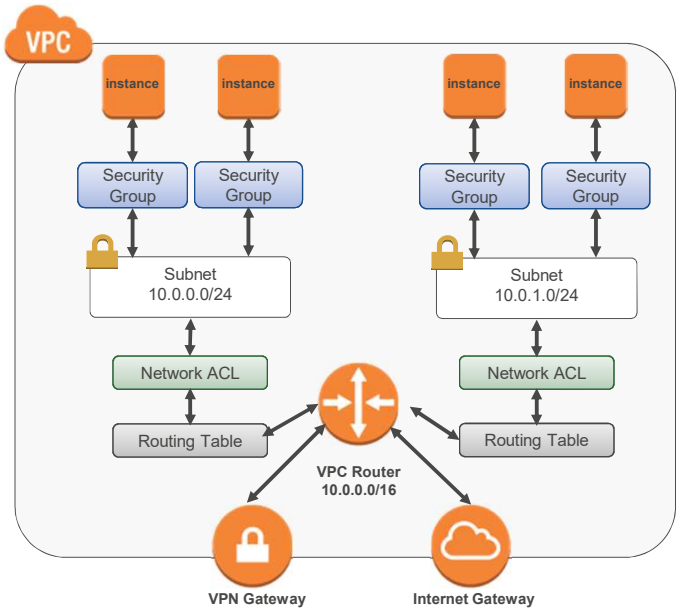
A private subnet has no route to an Internet gateway (i.e., for a database server only accessed within the VPC).



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, which allows you to leverage the AWS cloud as an extension of your corporate data center.

Security in Your VPC

- Security groups
- Network access control lists (ACLs)
- Key Pairs



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

- Security groups act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.
- Network access controls lists (ACLs) act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level.
- Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. Public-key cryptography uses a public key to encrypt a piece of data, and the recipient uses the private key to decrypt the data. The private and public keys are known as a *key pair*. To log in to your instance, you must create a key pair, specific the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. Linux instances have no password, and you use a key pair to log in using SSH. Windows instances require a key pair to obtain the administrator password to log in using RDP.

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

CCA Lab-02

Build your VPC and Launch a Web Server

(45 minutes)

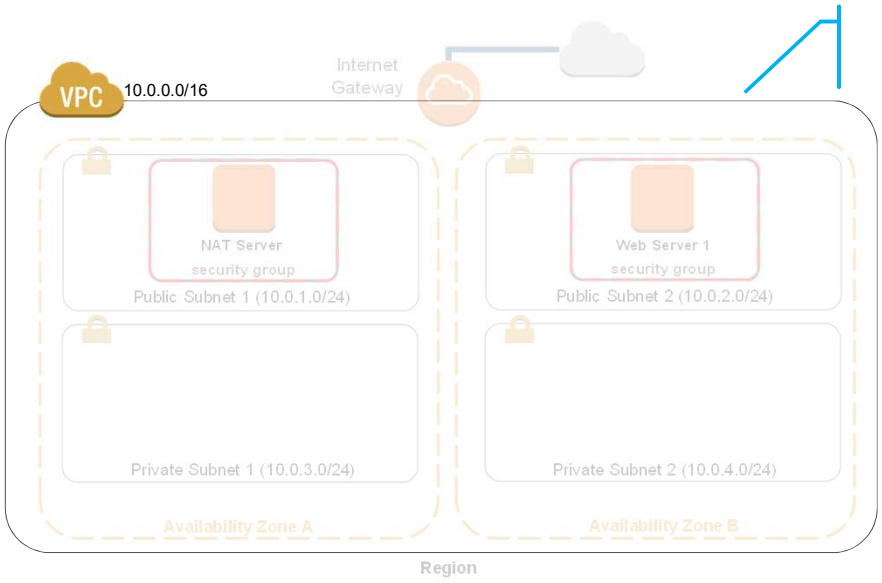
Lab 2: Overview

For more information, see:

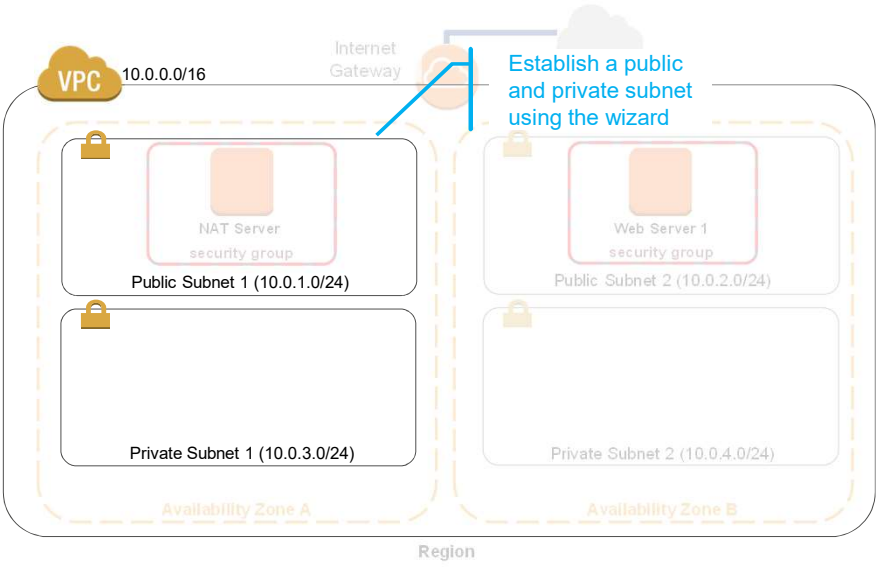
AWS Web Hosting Best Practices:
https://media.amazonwebservices.com/AWS_Web_Hosting_Best_Practices.pdf

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

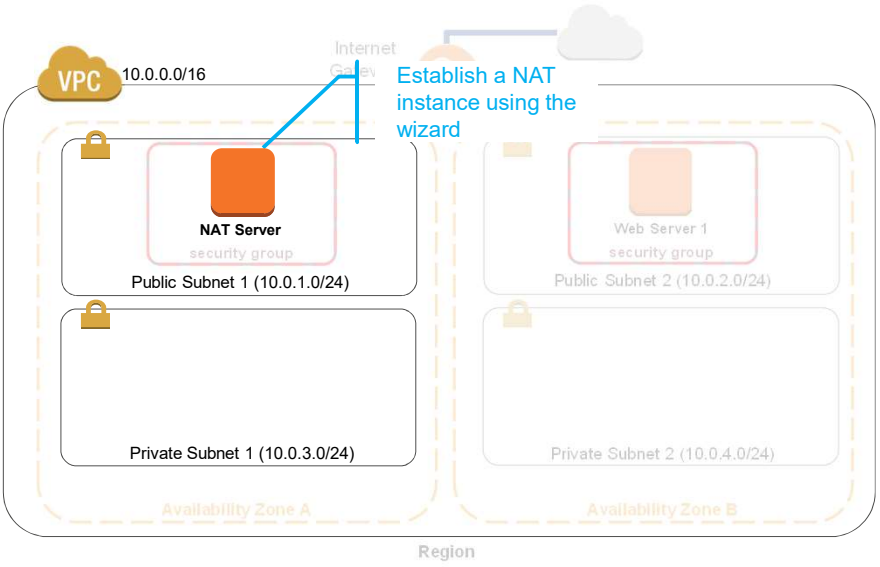
Lab 2: Build your VPC and Launch a Web Server



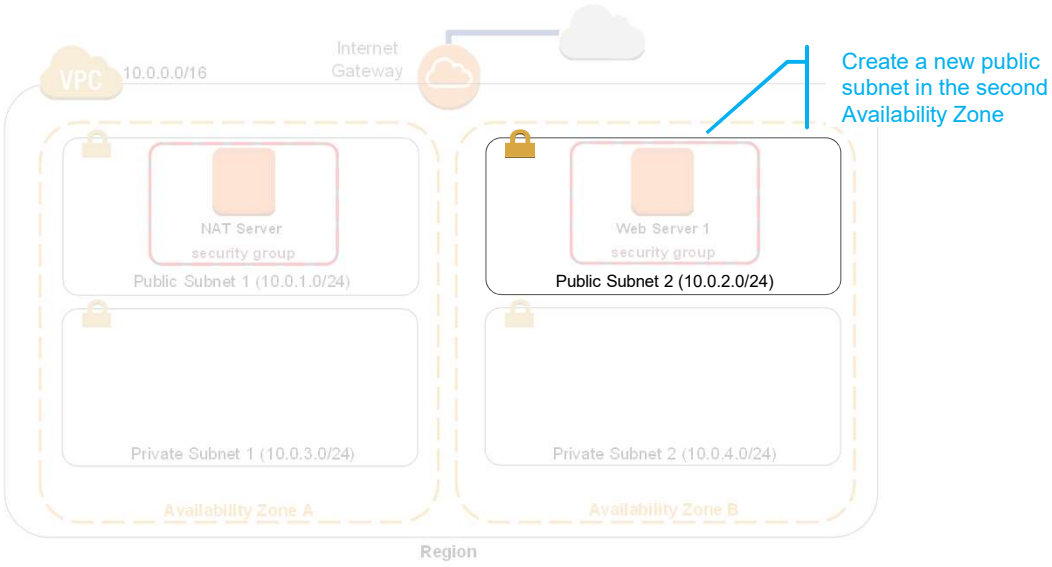
Lab 2: Build your VPC and Launch a Web Server



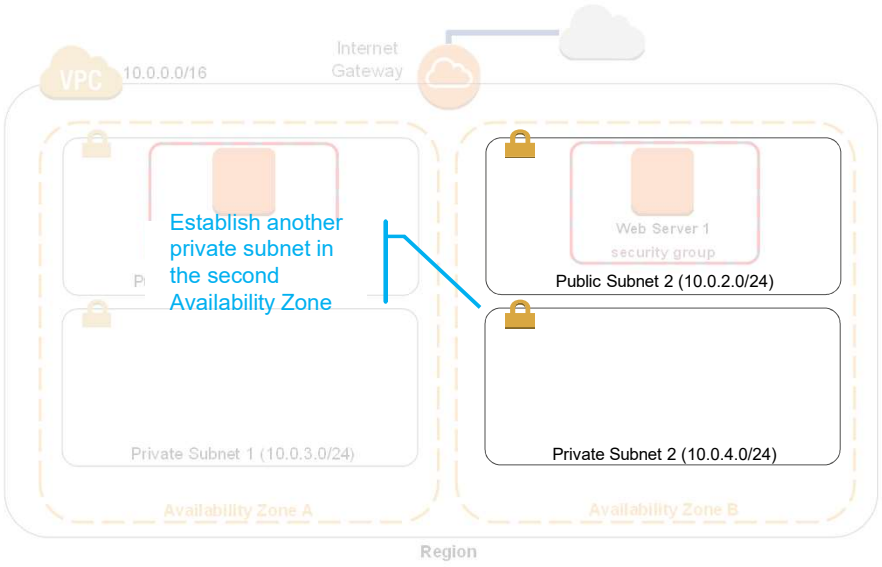
Lab 2: Build your VPC and Launch a Web Server



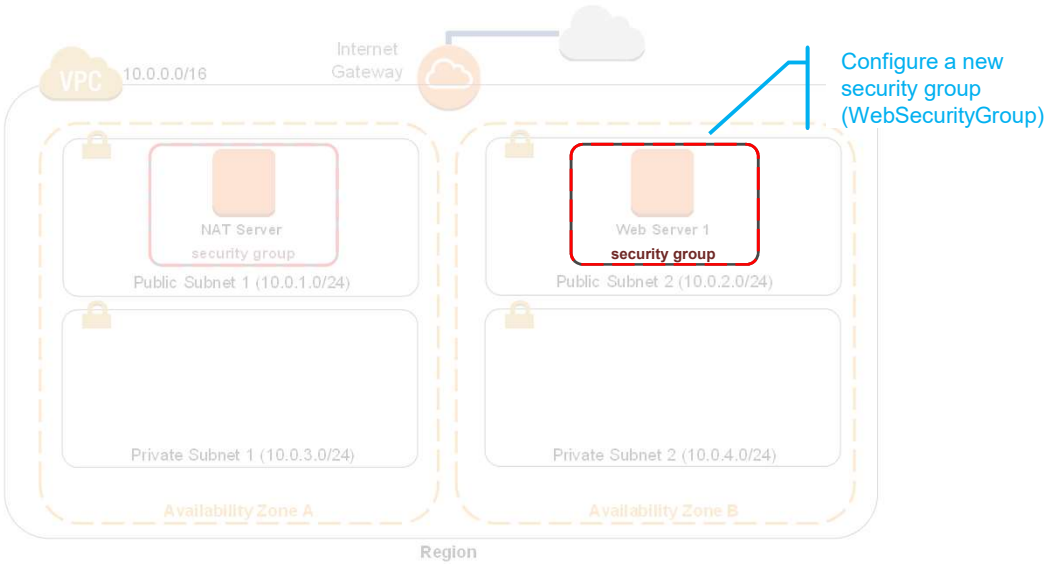
Lab 2: Build your VPC and Launch a Web Server



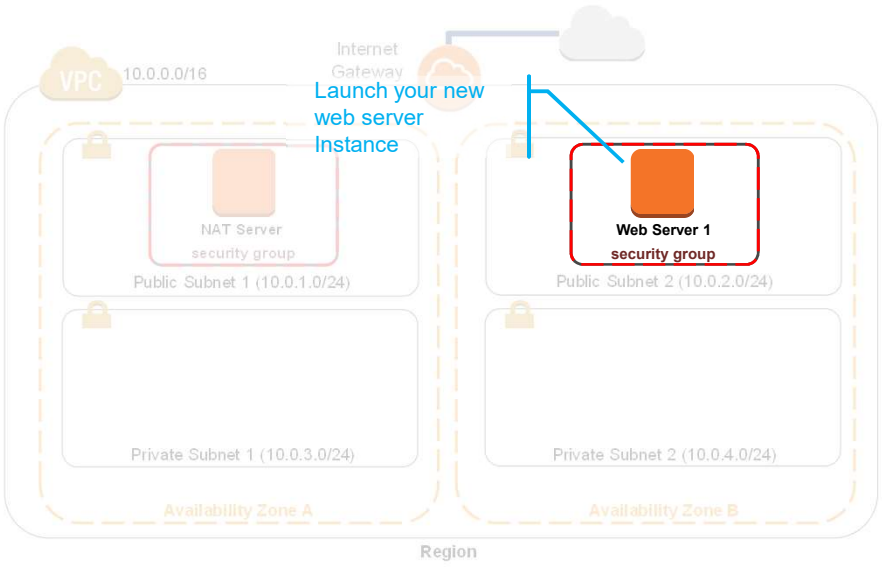
Lab 2: Build your VPC and Launch a Web Server



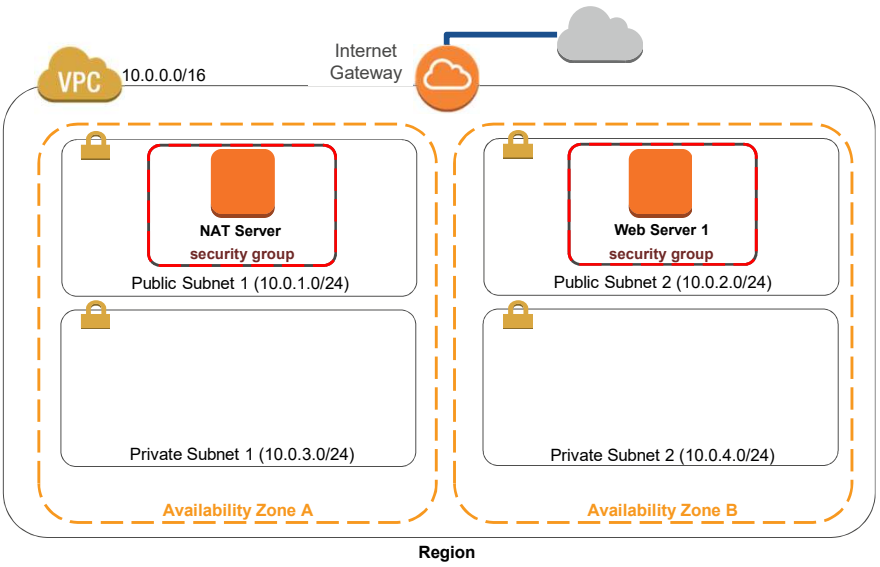
Lab 2: Build your VPC and Launch a Web Server

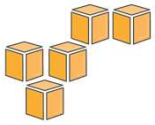


Lab 2: Build your VPC and Launch a Web Server



Lab 2: Build your VPC and Launch a Web Server





Part 4

Storage Services

- Amazon Simple Storage Service (S3)
- Amazon Glacier
- Amazon Elastic Block Store (EBS)
- EC2 Instance Storage

amazon web services | Training and Certification

Part 4: Storage Services

- Amazon Simple Storage Service (S3)
- Amazon Glacier
- Amazon Elastic Block Store (EBS)
- EC2 Instance Storage

Amazon Simple Storage Service (S3)

Amazon S3

- Storage for the Internet
- Natively online, HTTP access
- Storage that allows you to store and retrieve **any amount of data**, any time, from anywhere on the web
- **Highly scalable**, reliable, fast and durable

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/

Amazon S3 Facts

- Can store an **unlimited number of objects** in a bucket
- Objects can be **up to 5 TB**; no bucket size limit
- Designed for **99.999999999%** durability and **99.99%** availability of objects over a given year
- Can use **HTTP/S** endpoints to store and retrieve any amount of data, at any time, from anywhere on the web
- Is highly scalable, reliable, fast, and inexpensive
- Can use optional server-side **encryption** using AWS or customer-managed provided client-side encryption
- Auditing is provided by access logs
- Provides standards-based **REST** and SOAP interfaces

Here are some facts about Amazon S3. There is a 100-bucket limit per account. You can store an 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.

Common Use Scenarios

- Storage and backup
- Application file hosting
- Media hosting
- Software delivery
- Store AMIs and snapshots

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.

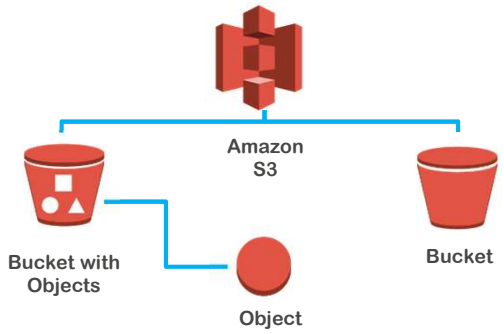
Publishing Content Using Amazon S3 and BitTorrent: You can direct your clients to your BitTorrent-accessible objects by giving them the .torrent file directly or by publishing a link to the BitTorrent URL of your object.

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 Concepts



- Amazon S3 stores data as objects within **buckets**
- An object is composed of a file and optionally any **metadata** that describes that file
- You can have **up to 100 buckets** in each account
- You can **control access** to the bucket and its objects

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 permissions on the object and add 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.

Object Keys

An object key is the unique identifier for an object in a bucket:

http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.html

Bucket

Object/Key

Because the combination of a bucket, key, and version ID uniquely identifies 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.

Amazon S3 Security

- You can **control access** to buckets and objects with:
  - Access Control Lists (ACLs)
  - Bucket policies
  - Identity and Access Management (IAM) policies
- You can upload or download data to Amazon S3 via **SSL** encrypted endpoints.
- You can **encrypt data** using AWS SDKs.

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 data to and download data from 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 such as the Amazon S3 Encryption Client to encrypt data before

uploading to Amazon S3.

Alternatively, they 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.

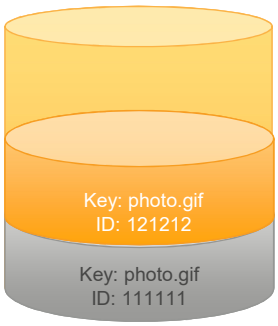
Amazon S3 SSE uses one of the strongest block ciphers available: 256-bit Advanced Encryption Standard (AES-256). With Amazon S3 SSE, every protected object is encrypted with a unique encryption key. This object key itself is then encrypted with a regularly rotated master key. Amazon S3 SSE provides additional 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 can also encrypt 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

Versioning is a method 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).

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

Amazon S3 Object Lifecycle

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:

- Log files
- Archive documents
- Digital media archives
- Financial and healthcare records
- Raw genomics sequence data
- Long-term database backups
- Data that must be retained for regulatory compliance

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/

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. Because 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/

Amazon S3

Instructor Demo

- Create a S3 bucket
- Set access control with a bucket policy
- Enable versioning and server-side encryption
- Upload an object
- Set the object lifecycle

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

Amazon Glacier

- Long term low-cost archiving service
- Optimal for infrequently accessed data
- Designed for 99.999999999% durability
- Three to five hours' retrieval time
- Less than $0.01 per GB/month (depending on region)

For more information about Glacier pricing, see:
https://aws.amazon.com/glacier/pricing/

Glacier and S3 Storage Classes

Storage Class	Durability	Availability	Other Considerations
Amazon S3 Standard	99.999999999%	99.99%	
Amazon S3 Standard - Infrequent Access (IA)	99.999999999%	99.9%	• Retrieval fee associated with objects • Most suitable for infrequently accessed data
Glacier	99.999999999%	99.99% (once restored)	• Not available for real-time access • Must restore objects before you can access them • Restoring objects can take 3-5 hours

Each object in Amazon 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 acces: 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, thus reducing cost.

For more information, see:

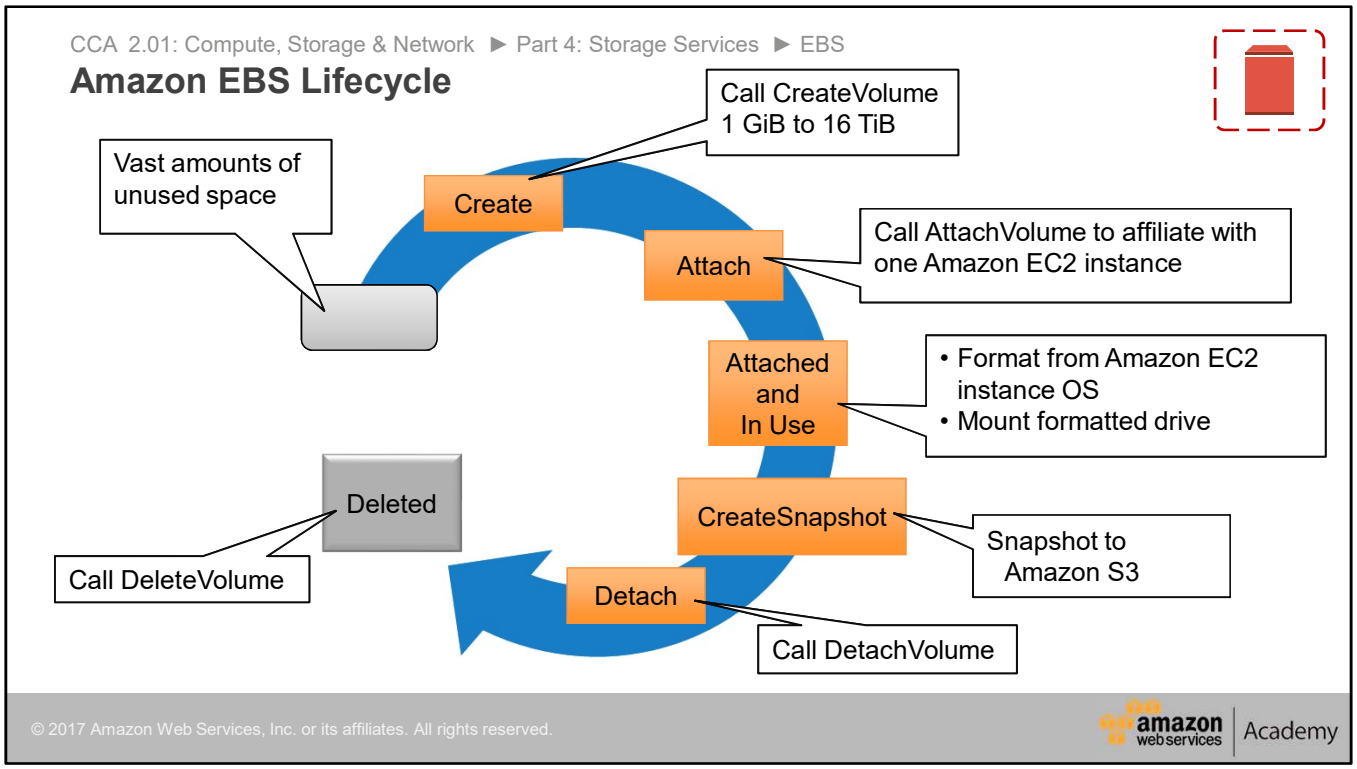
- http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html
- https://aws.amazon.com/s3/storage-classes/

Amazon Elastic Block Store (EBS)

Amazon EBS

- **Persistent block level storage** volumes offer consistent and low-latency performance.
- Stored data is automatically replicated within its Availability Zone.
- Snapshots are stored durably in Amazon S3.

Amazon Elastic Block Store, also known as Amazon EBS, provides persistent block-level storage volumes for use with Amazon EC2 instances and offers 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 EBS Volume Types

SSD-backed volumes

- Optimized for **transactional** workloads that involve **frequent read/write** operations with **small I/O** size.
- Dominant in **IOPS** performance.

HDD-backed volumes

- Optimized for **large streaming** workloads.
- Dominant in **throughput** (measured in MiB/s).

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volumes types fall into two categories: solid-state drives (SSD) and hard disk drives (HDD).

For more information, see:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

Amazon EBS Volume Types

	SSD		HDD	
Volume Type	**General Purpose SSD (gp2)**	**Provisioned IOPS SSD (io1)**	**Throughput Optimized HDD (st1)**	**Cold HDD (sc1)**
Description	Balances price and performance for a wide variety of **transactional** loads.	**Highest-performance** SSD volume designed for **mission-critical** applications.	**Low-cost** HDD designed for **frequently accessed**, **throughput-intensive** workloads.	**Lowest cost** HDD designed for less frequently accessed workloads.
Volume Sizes	1 GiB – 16 TiB	4 GiB – 16 TiB	500 GiB – 16 TiB	500 GiB – 16 TiB
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

For more information, see:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

Amazon EBS Facts

- EBS is recommended when data must be **quickly accessible** and requires **long-term persistence**.
- You can launch your EBS volumes as **encrypted** volumes – data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted.
- You can create **point-in-time snapshots** of EBS volumes, which are persisted to Amazon S3.

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 and 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.

Amazon EBS Use Cases

OS:	Use for boot/root volume, secondary volumes
Databases:	Scales with your performance needs
Enterprise applications:	Provides reliable block storage to run mission-critical applications
Business continuity:	Minimize data loss and recovery time by regularly backing up using EBS Snapshots
Applications:	Install and persist any application

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
- Review Pricing Calculator online
- Pricing is available as: Storage or IOPS

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

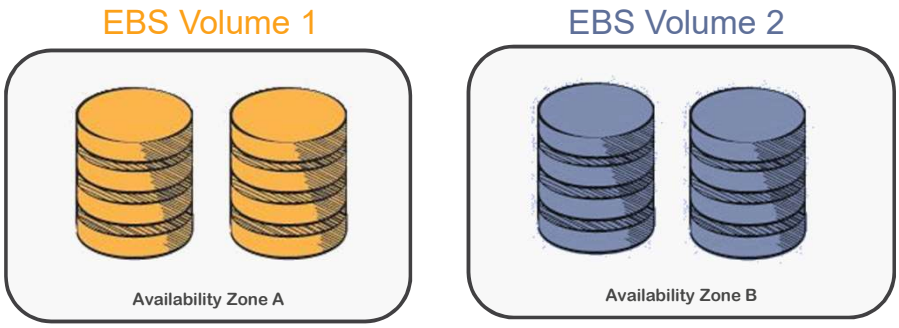
Amazon EBS pricing is based on allocated storage, whether you use it or not. This is unlike Amazon S3, whose pricing is based on space actually in use. Prices may vary based on region or for IOPS.

For more information, see:

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

Amazon EBS Scope

Amazon EBS volumes are in a single Availability Zone:



Volume data is replicated across multiple servers in an Availability Zone.

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% and 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%, making EBS volumes 10 times more reliable than typical commodity disks.

Because 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.

If you are 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.

Amazon EBS and Amazon S3

	Amazon EBS	Amazon S3
Paradigm	Block storage with file system	Object store
Performance	Very fast	Fast
Redundancy	Across multiple servers in an Availability Zone	Across multiple facilities in a Region
Security	EBS Encryption – Data volumes and Snapshots	Encryption
Access from the Internet?	No (1)	Yes (2)
Typical use case	It is a disk drive	Online storage

(1) Accessible from the Internet if mounted to server and set up as FTP, etc.
(2) Only with proper credentials, unless ACLs are world-readable

This table illustrates the 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.

Amazon EC2 Instance Storage

- Local, complimentary **direct attached block storage**
- Includes availability, number of disks, and size **based on EC2 instance type**
- Optimized for **up to 365,000 Read IOPS** and 315,000 First Write IOPS
- SSD or magnetic
- Has **no persistence**
- **Automatically deletes** data when an EC2 instance stops, fails or is terminated

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

Amazon EBS vs. Amazon EC2 Instance Store

Amazon EBS

- Data stored on an Amazon EBS volume can persist independently of the life of the instance.
- Storage is **persistent**.

Amazon EC2 Instance Store

- Data stored on a local instance store persists only as long as the instance is alive.
- Storage is **ephemeral**.

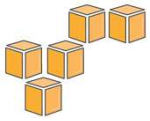
Use the local instance store only for temporary data. For data that requires a higher level of durability, use Amazon EBS volumes or back up the data to Amazon S3. If you are using an Amazon EBS volume as a root partition and want your Amazon EBS volume to persist outside of the life of the instance, set the Delete on termination flag to “No”.

Instance Lifecycle – 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**	
Public IP address	No change	**New address assigned**	
Elastic IP addresses (EIP)	EIP remains associated with the instance	EIP remains associated with the instance	EIP is **disassociated** from the instance
Instance store volumes	Preserved	Erased	Erased
EBS volume	Preserved	Preserved	Boot volume is **deleted by default**
Billing	Instance billing hour doesn't change	**Stops incurring charges** as soon as state is changed to *stopping*	**Stops incurring charges** as soon as state is changed to *shutting-down*

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



In review…

- Compute Services Using Amazon EC2
- Virtual Private Networks (VPNs)
- Storage Services
  - Amazon Simple Storage Service (S3)
  - Edge Locations, Route 53, CloudFront

Knowledge Assessment

amazon web services | Training and Certification

In review…

- Compute Services Using Amazon EC2
- Virtual Private Networks (VPNs)
- Storage Services
  - Amazon Simple Storage Service (S3)
  - Edge Locations, Route 53, CloudFront

To complete this module, please remember to finish the corresponding knowledge assessment.

Knowledge Check

Q What AWS service would help support your web application to **offload serving static assets** and **store user uploaded images and video** off-instance?

Amazon S3

Q How would an EC2 instance find its private and public IP addresses?

Retrieve the instance metadata. http://169.254.169.254/latest/meta-data/

Q What acts as an additional layer of security at the subnet level in a VPC?

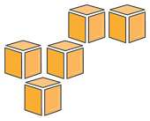
Network ACLs

T/F S3 limits the amount you can store

False

CCA Lab-03

Working with EBS



Up Next…

LAB 03 - Working with EBS

CCA 2.02 - AWS Security, Identity, and Access Management

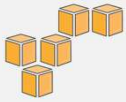
amazon web services | Training and Certification

If you haven't completed it already, please do **Lab 3, Working with EBS**. Refer back to the Welcome module for instructions on accessing the lab environment.

Be sure to complete the lab before continuing with **CCA 2.02** covering AWS security including Identity and Access Management (IAM).

© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Do not speak over this slide – just let it play for 8 seconds.



CCA Unit 2 – Getting Started with AWS

CCA 2.02: AWS Security and IAM

CCA 2.01 AWS Compute, Storage, and Networking

CCA 2.02 AWS Security, Identity, and Access Management

CCA 2.03 AWS Database Options

CCA 2.04 AWS Elasticity and Management Tools

amazon web services | Academy

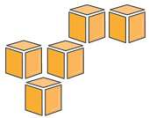
Welcome to Module CCA 2.02 –AWS Security, Identity, and Access Management (IAM)

What's In This Module?

- AWS Shared Responsibility Model
- AWS Identity and Access Management (IAM)
- AWS CloudTrail

This module covers…

- AWS Shared Responsibility Model
- AWS Identity and Access Management (IAM)
- AWS CloudTrail

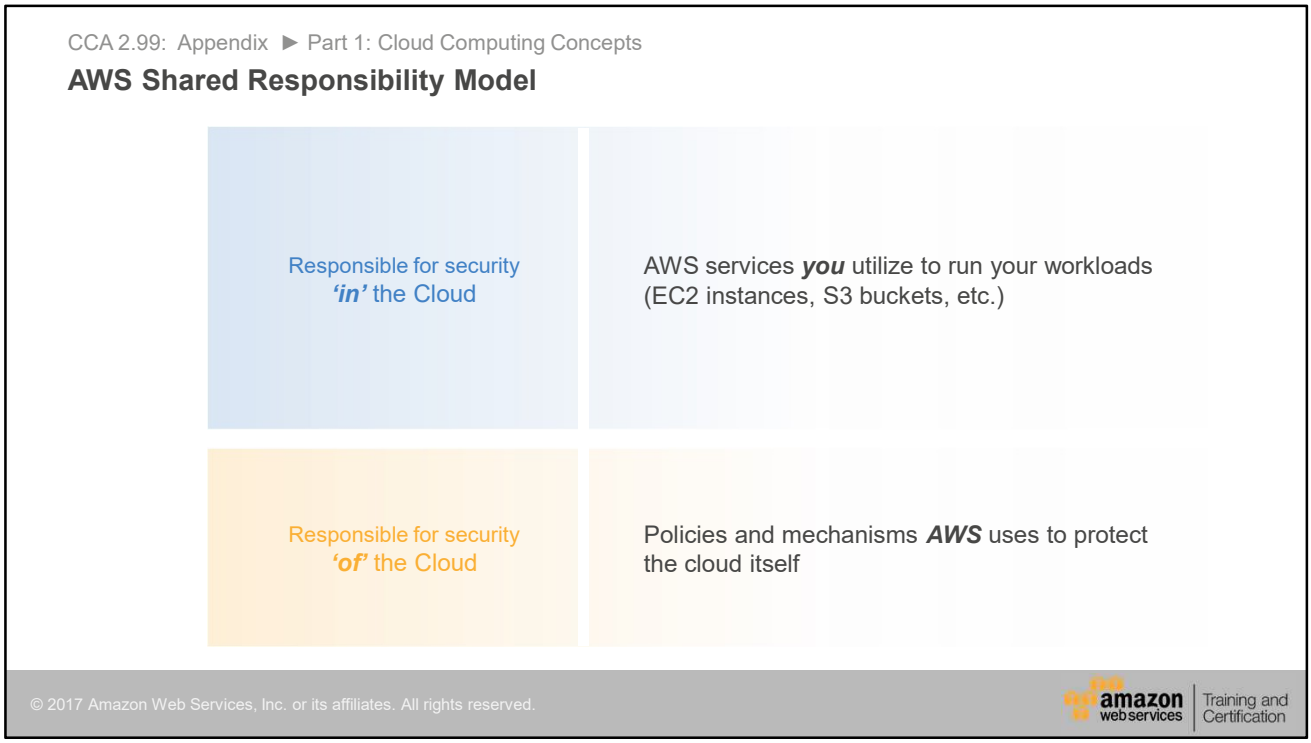


Part 1

AWS Shared Responsibility Model

amazon web services | Training and Certification

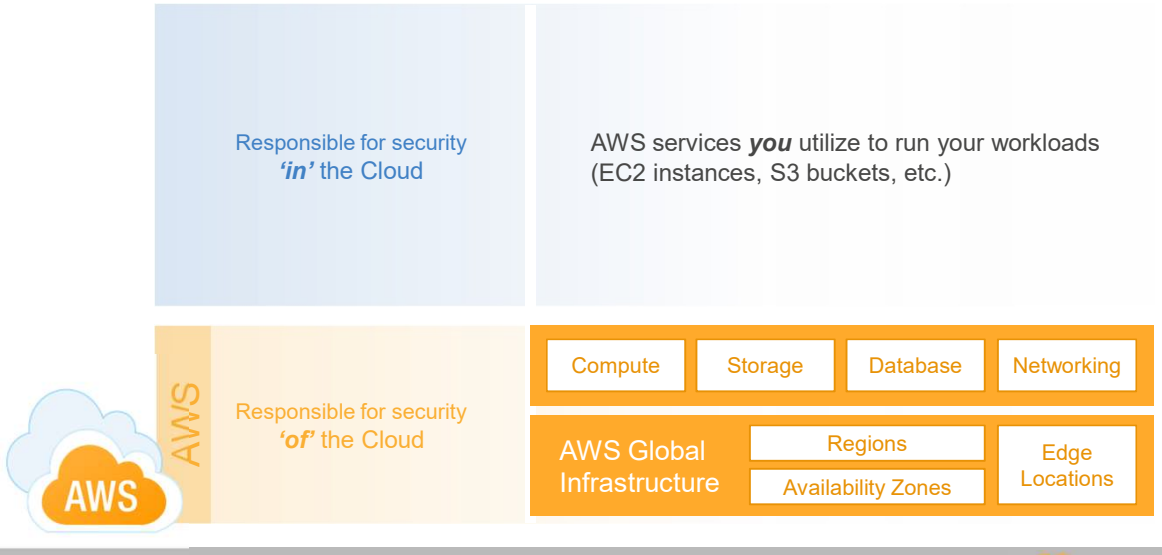
Part 1: AWS Shared Responsibility Model



With respect to your AWS infrastructure, we distinguish between:

- Security **in** the cloud - the AWS services you utilize to run your workloads (EC2 instances, S3 buckets, etc.) , and
- Security ***of*** the cloud - the policies and mechanisms AWS uses to protect the cloud itself

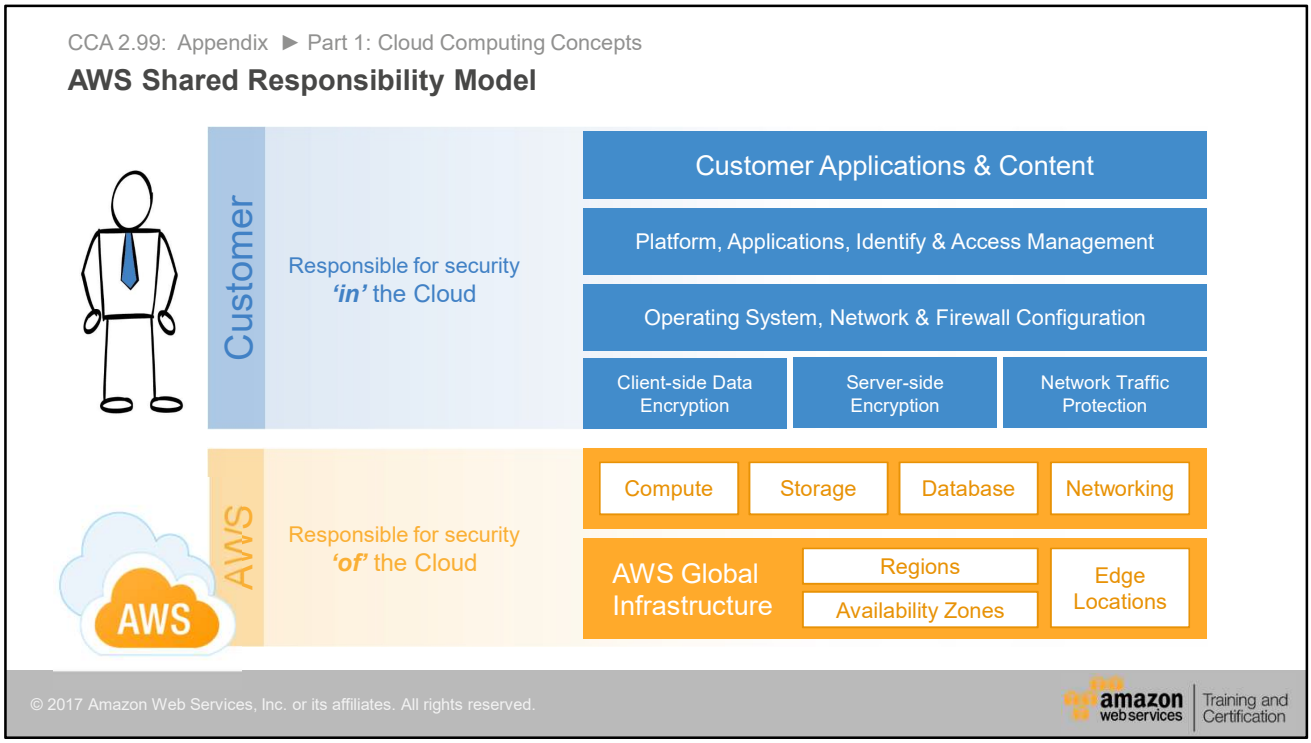
AWS Shared Responsibility Model



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. The responsibility of AWS 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 and 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, and the configuration of the AWS-provided firewall (called a security group) on each instance.

When you use any of the AWS managed services such as 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 such as 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.

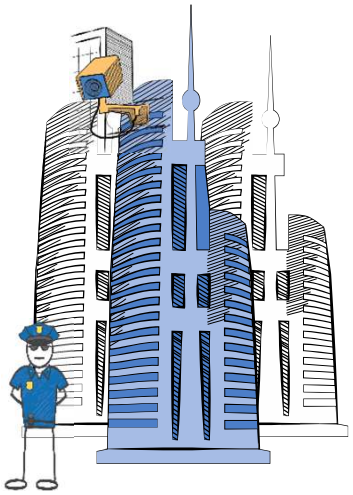
For more information, see:

- 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



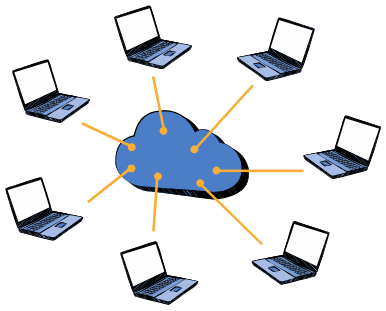
One of the main security responsibilities of AWS is the physical security of the data centers that house the AWS Cloud infrastructure. Amazon has many years of experience designing, constructing, and operating large-scale data centers.

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 highly available, redundant, and 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: http://aws.amazon.com/security/

Certifications and Accreditations

ISO 9001, ISO 27001, ISO 27017, ISO 27018, IRAP (Australia), MLPS Level 3 (China), MTCS Tier 3 Certification (Singapore) and more …

amazon web services | Training and Certification

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.

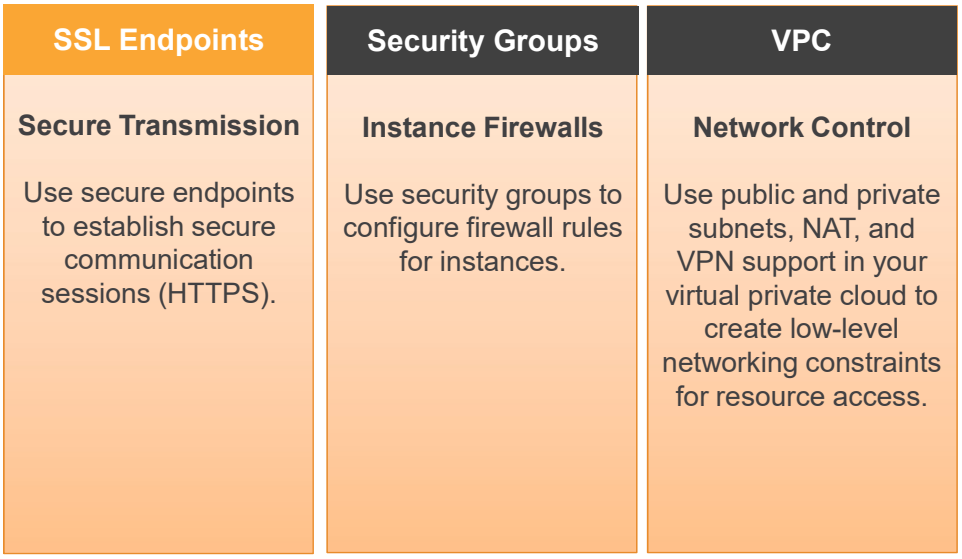
In addition, AWS has achieved ISO 9001, ISO 27001, ISO 27017, and ISO 27018 certifications, has been successfully validated as a Level 1 service provider under the Payment Card Industry (PCI) Data Security Standard (DSS), and currently offers HIPAA Business Associate Agreements to covered entities and 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, and DoD SRG Levels 2 and 4 are some of the other certifications AWS has received.

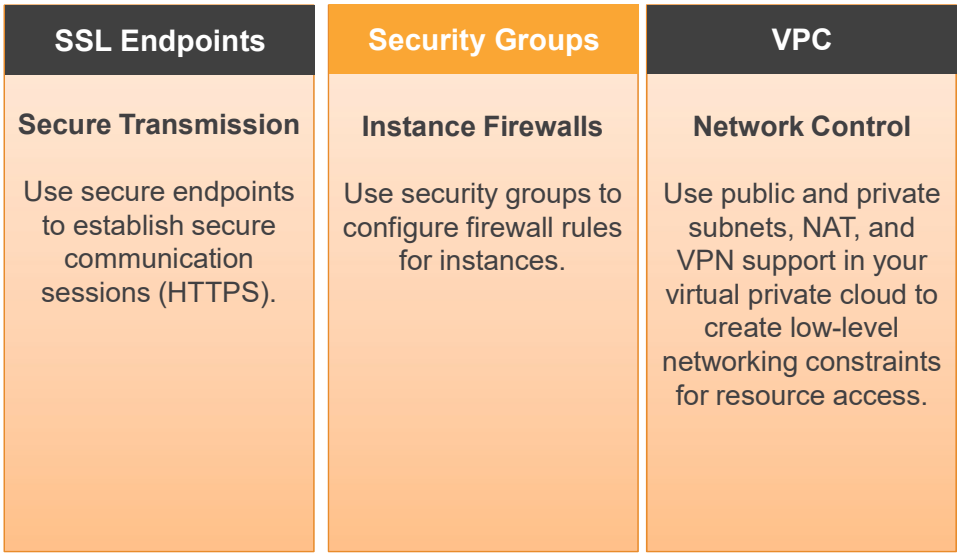
For more information, see: http://aws.amazon.com/compliance/

SSL Endpoints



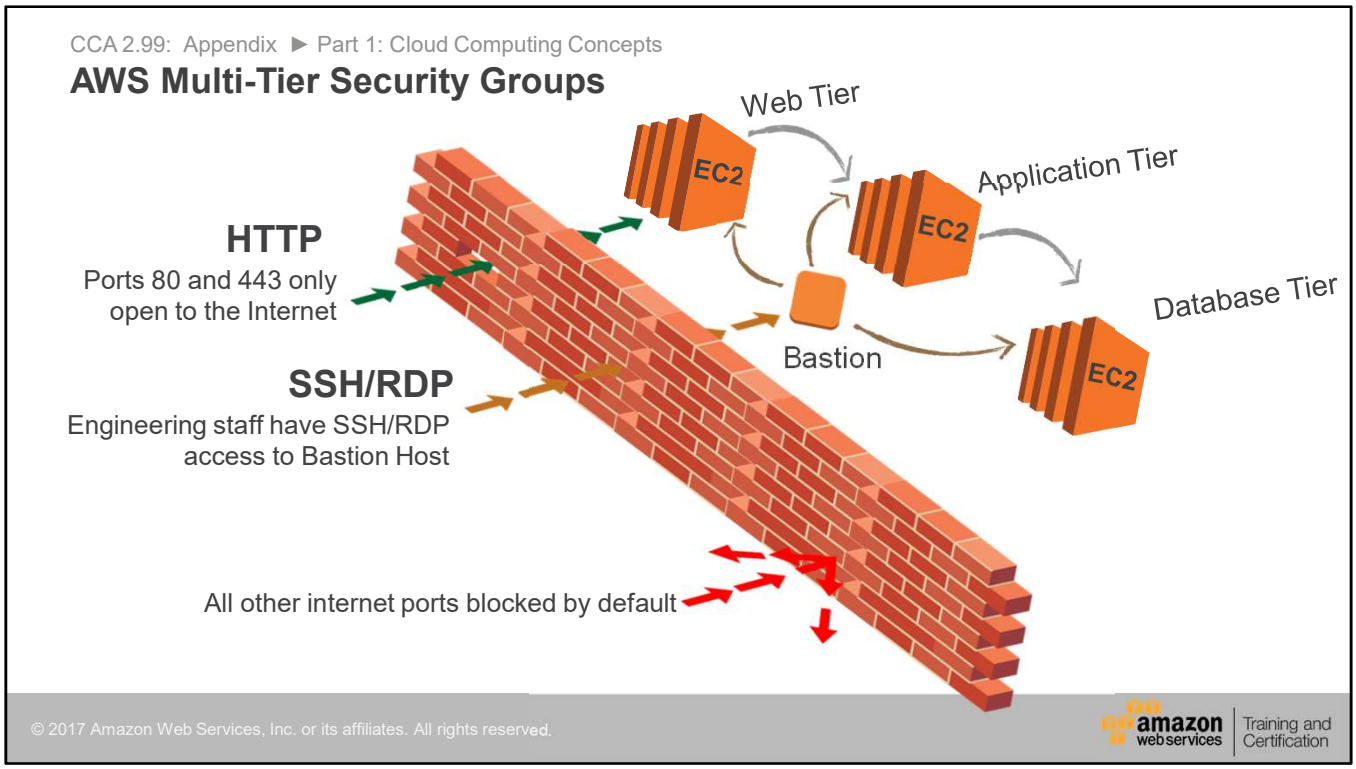
AWS provides customer access points, also called API endpoints, that allow HTTPS access so that you can establish secure communication sessions with your AWS services, including SSL and TLS. SSL encrypts the transmission, protecting each request or the response from being viewed in transit.

Security Groups



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.



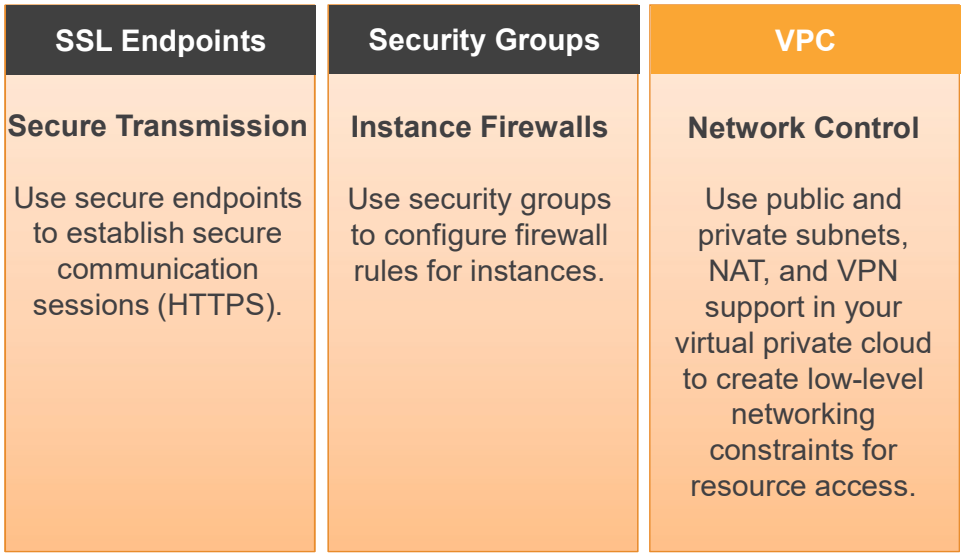
You can set up security group rules for your EC2 instances to create a traditional multi-tiered web architecture:

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.

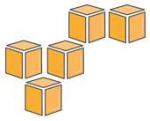
Lastly, we have also added a set of rules to allow remote administration over SSH port 22. We have restricted remote access by funneling all traffic through the app tier and allowing access only from a specific IP. After you use SSH to access an app tier server, you can then connect to machines on the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

Amazon Virtual Private Cloud (VPC)



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



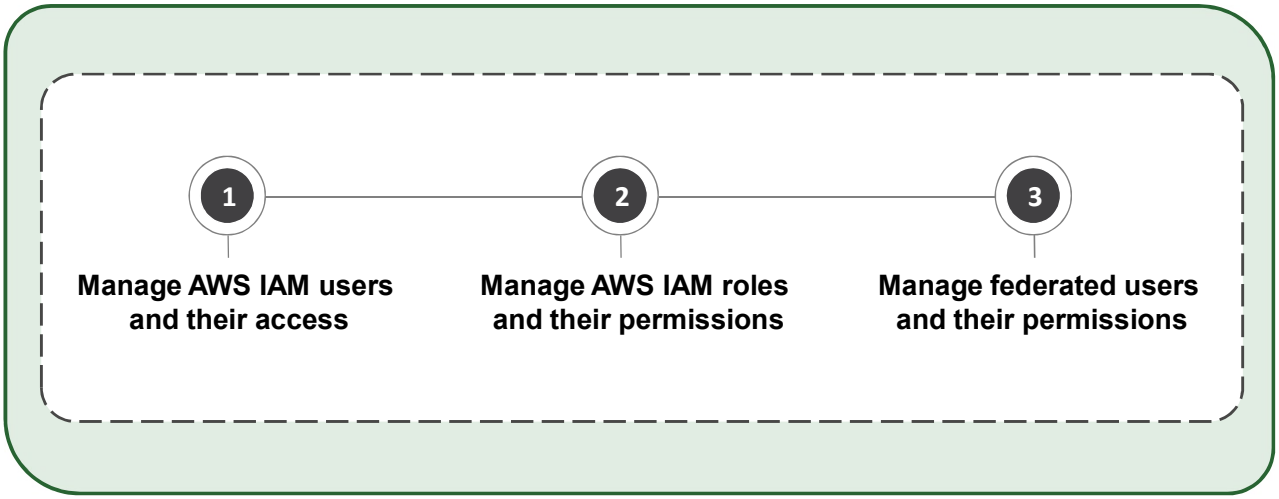
Part 2

AWS Identity and Access Management (IAM)

amazon web services | Training and Certification

Part 2: AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM)



Using AWS 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.

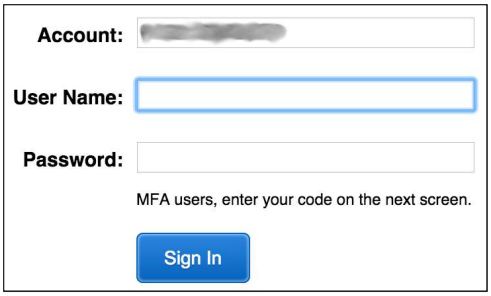
For more information, see:

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

AWS IAM Authentication

- **Authentication**
- **AWS Management Console**
  - User Name and Password

IAM User



amazon web services | Training and Certification

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 sign in using individual credentials.

IAM is a complimentary service.

For more information, see:
http://docs.aws.amazon.com/IAM/latest/UserGuide/console.html

AWS IAM Authentication

- **Authentication**
- **AWS CLI or SDK API**
  - Access Key and Secret Key

Access Key ID: AKIAIOSFODNN7EXAMPLE
Secret Access Key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

AWS CLI

```
[redacted]:~ [redacted]$ aws configure
AWS Access Key ID [****************O22A]:
AWS Secret Access Key [****************4m8i]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

AWS SDK & API


Java


Python


.NET

amazon web services | Training and Certification

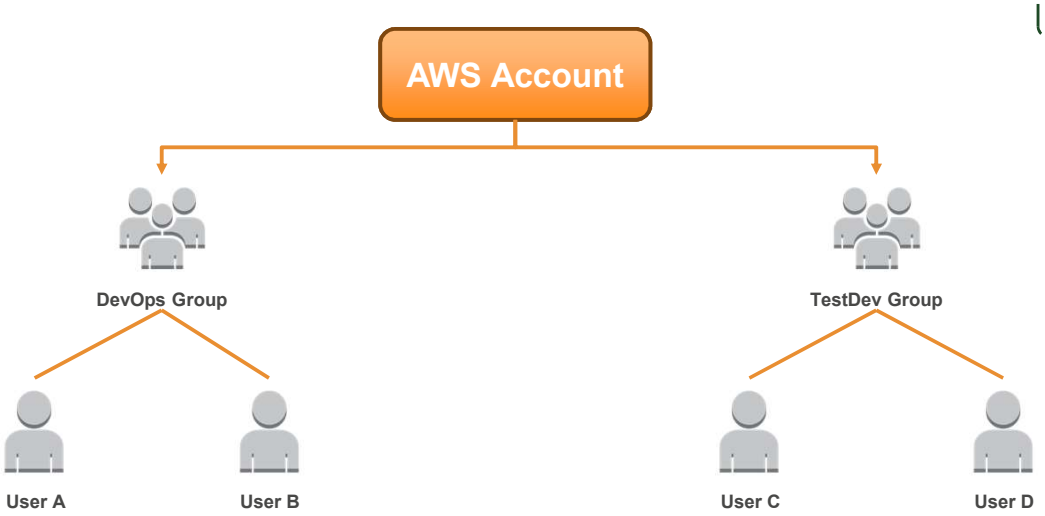
The AWS Command Line Interface is a unified tool to manage your AWS services. With AWS CLI, you can control multiple AWS services from the command line and automate them through scripts.

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, and Python.

For more information, see: http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

AWS IAM User Management - Groups



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:

For more information, see:
http://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

AWS IAM Authorization

Authorization

Policies:

- ✓Are JSON documents to describe permissions.
- ✓Are assigned to users, groups or roles.

IAM User

IAM Group

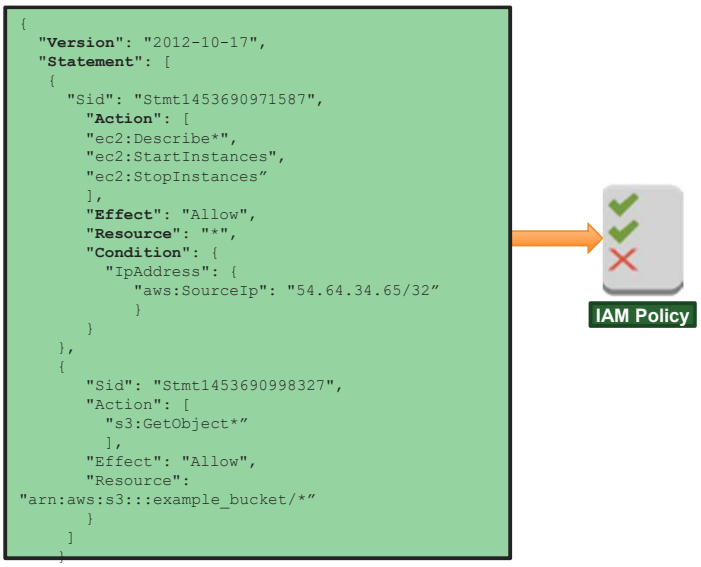
IAM Roles

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 are covered in more detail in the following slides.

AWS IAM Policy Elements



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*. The default is that resources are denied to users, so 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 easily generate policies.

AWS Policy Validator: The 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: 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, we 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.

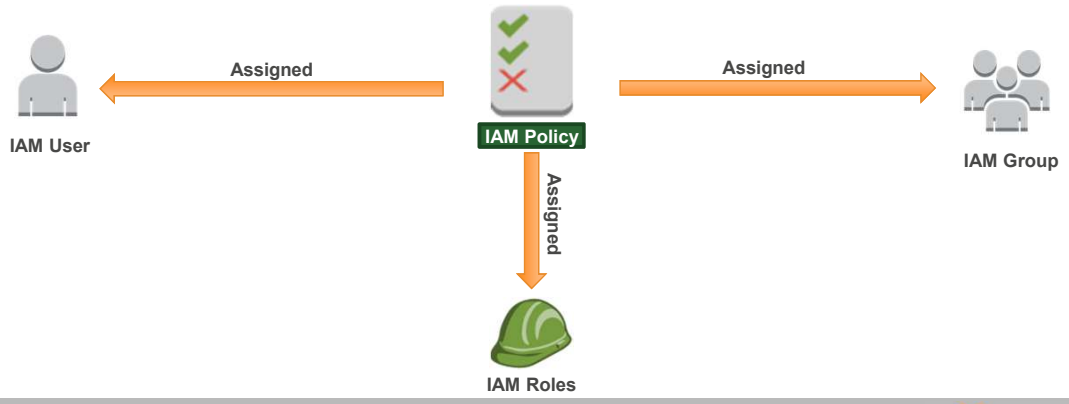
For more information, see:

- 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

AWS IAM Policy Assignment

IAM policies are assigned to IAM users and groups. These users are bound by the permissions defined in the IAM policy.

AWS IAM Policy Assignment



IAM policies may also be assigned to an IAM role.

For more information, see:
http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html

AWS IAM Roles

IAM Roles

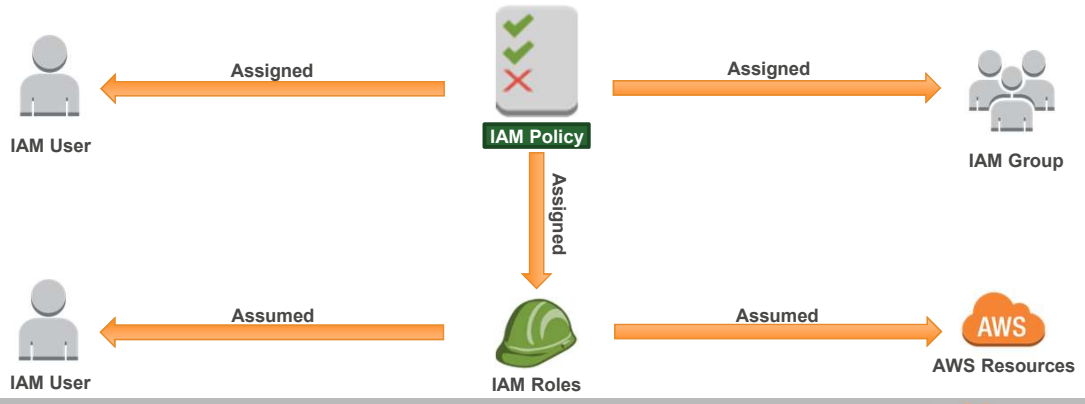
- An IAM role uses a **policy**.
- An IAM role has **no associated credentials**.
- IAM users, applications, and services may assume IAM **roles**.

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. 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

AWS IAM Policy Assignment



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

Example: Application Access to AWS Resources

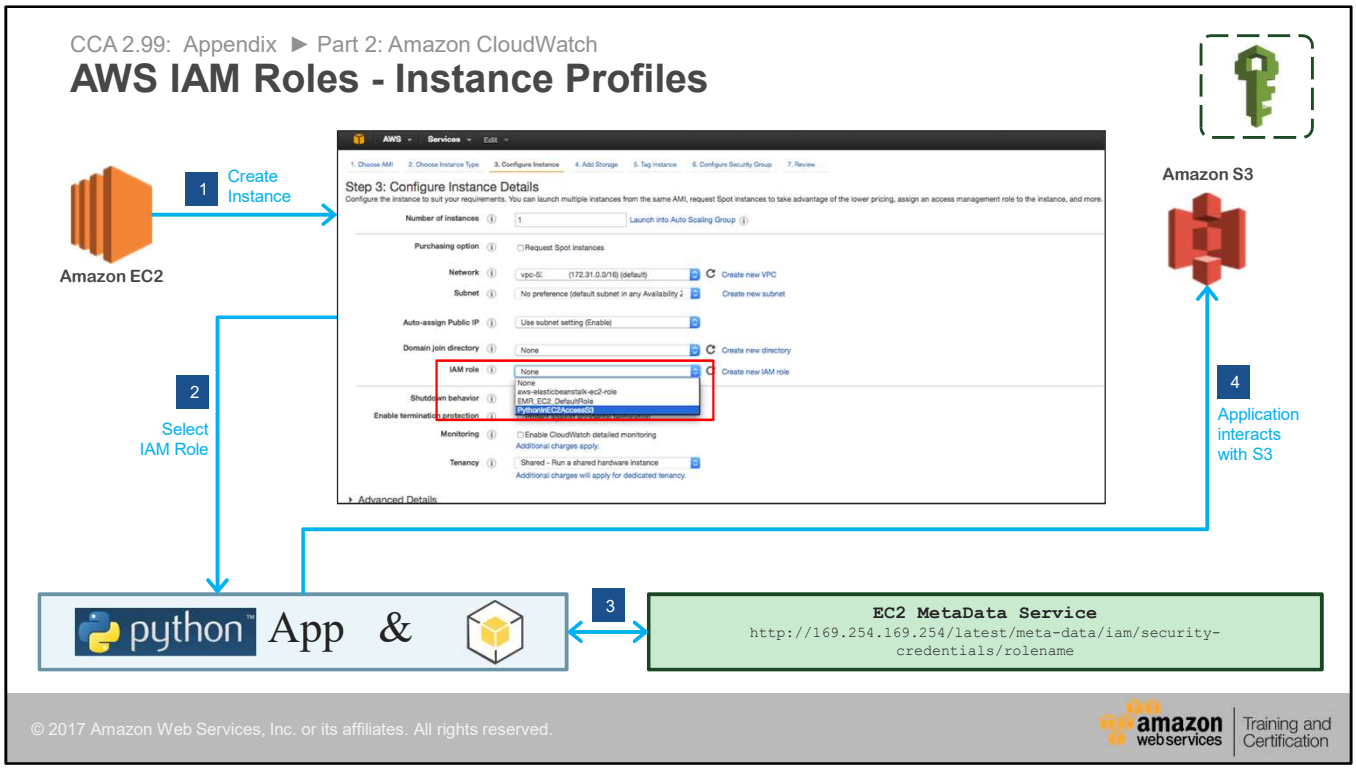
- Python application hosted on an Amazon EC2 Instance needs to interact with Amazon S3.
- AWS credentials are required:
  - Option 1: Store AWS Credentials on the Amazon EC2 instance.
  - Option 2: Securely distribute AWS credentials to AWS Services and Applications.

IAM Roles

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 alternative 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.

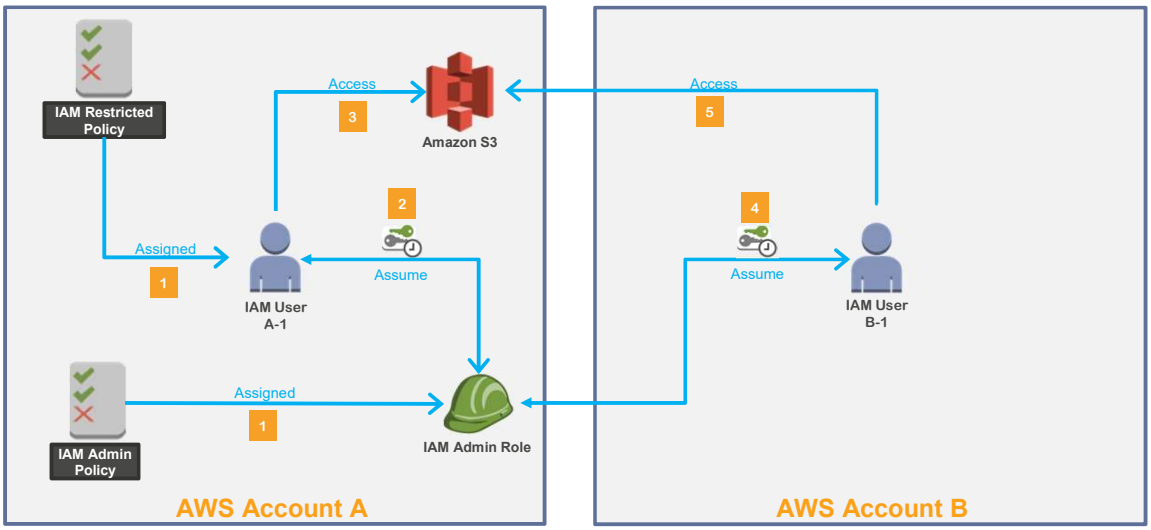
1. An application developer selects the PythonInEC2AccessS3 role while creating the Amazon EC2 instance. The instance will host a Python application that will need access to an Amazon S3 bucket.

Note An IAM role may be associated with an EC2 instance only during creation. The policy associated with the role can be modified at any time. A user launching an EC2 instance also needs appropriate permissions to associate an IAM role to the EC2 instance.

2. 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.
3. The Python application uses the EC2 metadata service to gain access to temporary security credentials.
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

AWS IAM Roles – Assume Role



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*.

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 because *User A-1* does not normally need administrative privileges. However, User A-1 may sometimes have to perform tasks that require administrative privileges.
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.

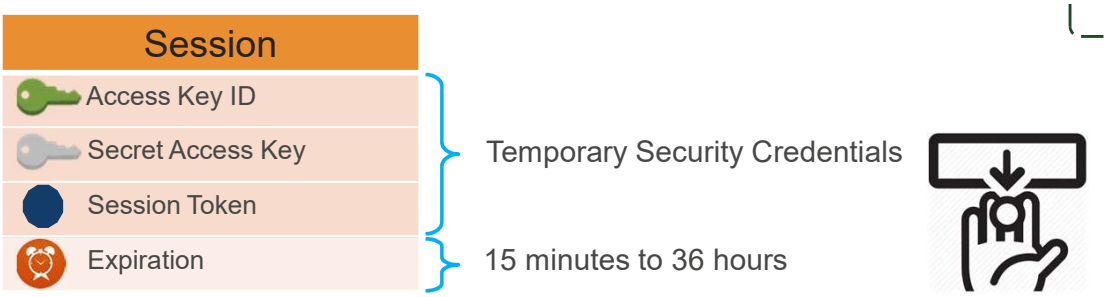
Note *User A-1*'s policy must contain permissions to assume the role.

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

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

Temporary Security Credentials (AWS STS)



Use Cases

- Cross account access
- Federation
- Mobile Users
- Key rotation for Amazon EC2-based apps

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, a secret access key, a session token, and an expiration time. The expiration time could be between 15 minutes and 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

Application Authentication

AWS IAM is not appropriate for OS and application authentication.

AWS IAM Authentication and Authorization

Authentication:

- **AWS Management Console**
  User Name and Password
- **AWS CLI or SDK API**
  Access Key and Secret Key

Authorization: Policies

IAM User

IAM Group

IAM Roles

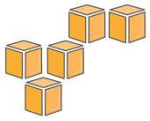
IAM is a powerful service to authenticate and authorize users and AWS resources.

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.
- Use **roles for applications** that run on Amazon EC2 instances.
- Delegate by **using roles** instead of by sharing credentials.
- **Rotate credentials** regularly.
- **Remove unnecessary** users and credentials.
- Use **policy conditions** for extra security.
- **Monitor activity** in your AWS account.

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



Part 3

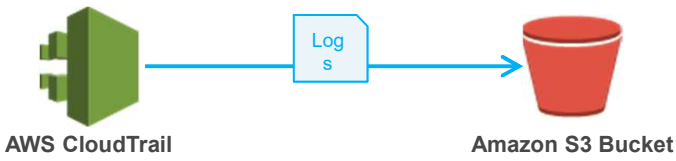
AWS CloudTrail

amazon web services | Training and Certification

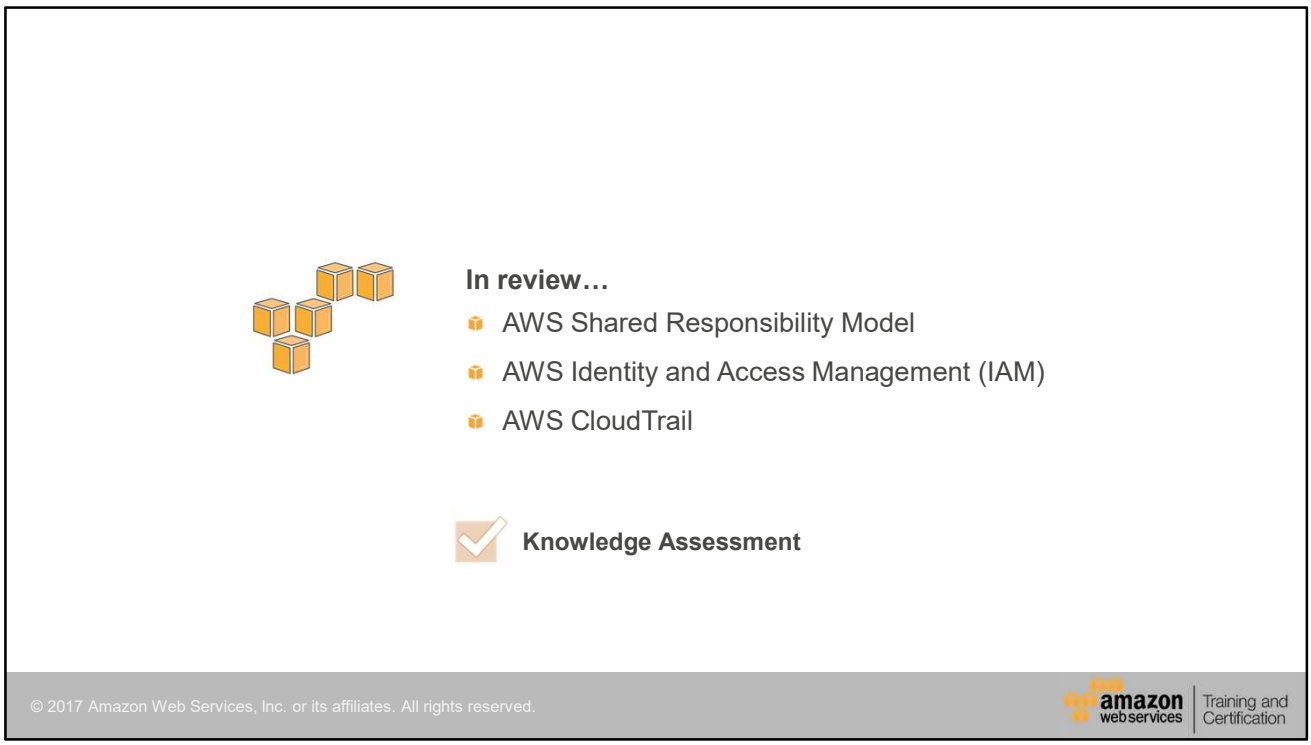
Part 3: AWS CloudTrail

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.



In review…

- AWS Shared Responsibility Model
- AWS Identity and Access Management (IAM)
- AWS CloudTrail

To complete this module, please remember to finish the corresponding knowledge assessment.

Knowledge Check

Q

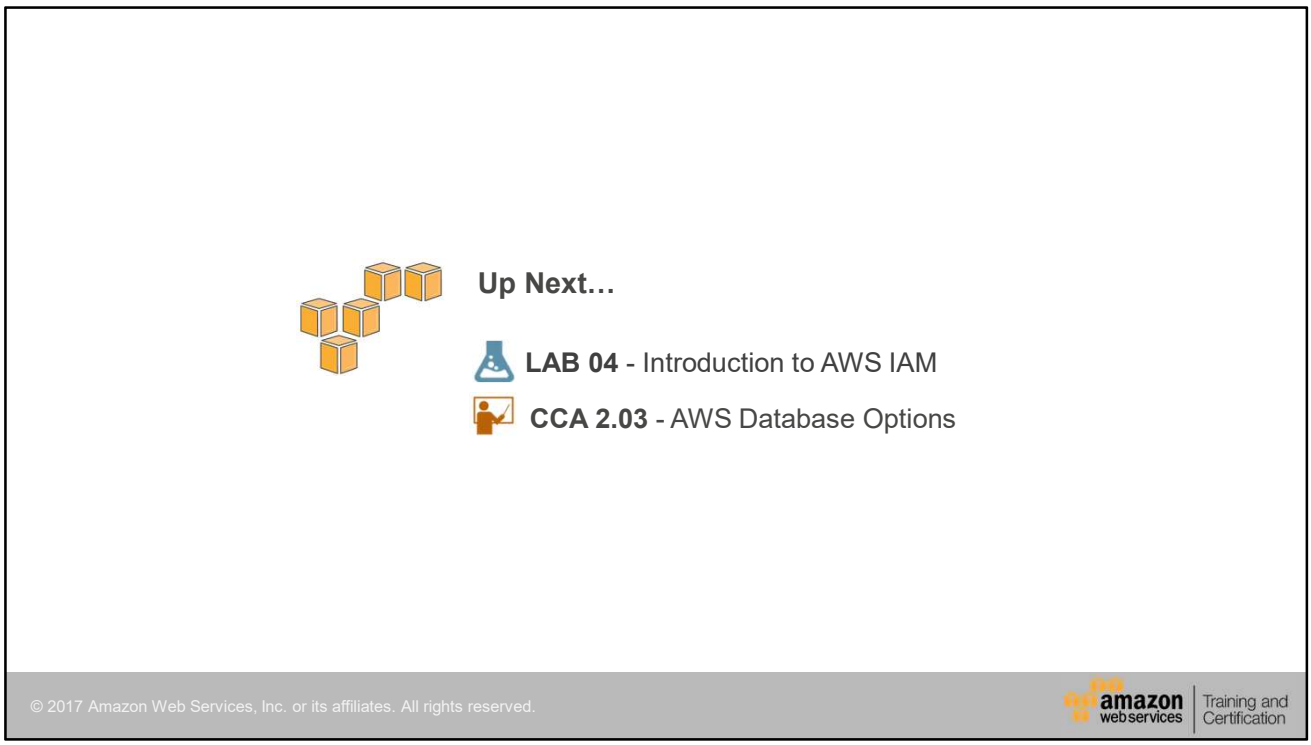
Your **web application** needs to **read/write** an Amazon DynamoDB table and an Amazon S3 bucket. This operation requires **AWS credentials** and **authorization to use AWS services**. What IAM entity should be used?

You have reached the end of this training module. Test out some of your new skills!

Instructor Demo

Using the IAM console…
- Create a user
- Create a role
- Create a group
- Create a policy

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

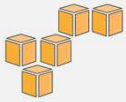


If you haven't completed it already, please do **Lab 4, Introduction to AWS IAM**. Refer back to the Welcome module for instructions on accessing the lab environment.

Be sure to complete the lab before continuing with **CCA 2.03** covering AWS database options.

© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Do not speak over this slide – just let it play for 8 seconds.



CCA Unit 2 – Getting Started with AWS

CCA 2.03: AWS Database Options

CCA 2.01 AWS Compute, Storage, and Networking

CCA 2.02 AWS Security, Identity, and Access Management

CCA 2.03 AWS Database Options

CCA 2.04 AWS Elasticity and Management Tools

amazon web services | Academy

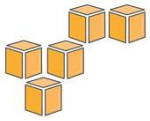
Welcome to Module CCA 2.03 –AWS Database Options

What's In This Module?

- Databases Overview and Considerations
- Amazon Relational Database Service (RDS)
- Amazon DynamoDB
- Choosing a Database Service

This module covers…

- Databases Overview and Considerations
- Amazon Relational Database Service (RDS)
- Amazon DynamoDB
- Choosing a Database Service



Part 1

Database Overview

amazon web services | Training and Certification

Part 1: Database Overview

SQL and NoSQL Databases

	SQL	NoSQL
Data Storage	Rows and Columns	Key-Value
Schemas	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scalability	Vertical	Horizontal

ISBN	Title	Author	Format
9182932465265	Cloud Computing Concepts	Wilson, Joe	Paperback
3142536475869	The Database Guru	Gomez, Maria	eBook

```
{
   ISBN: 9182932465265,
   Title: "Cloud Computing Concepts",
   Author: "Wilson, Joe",
   Format: "Paperback"
}
```


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 rapidly. 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.

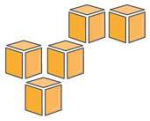
Data Storage Considerations

No one size fits all

Analyze your data requirements by considering:

- ✓ Data formats
- ✓ Data size
- ✓ Query frequency
- ✓ Data access speed
- ✓ Data retention period

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.



Part 2

Amazon Relational Database Service (RDS)

amazon web services | Training and Certification

Part 2: Amazon Relational Database Service (RDS)

Amazon Relational Database Service (RDS)

Amazon RDS

- Cost-efficient and **resizable capacity**
- Manages time-consuming **database administration** tasks
- Access to the full capabilities of databases:
  Amazon Aurora • MySQL • MariaDB • Microsoft SQL Server • Oracle • PostgreSQL

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.

Amazon RDS

- Simple and **fast to deploy**
- Manages common database administrative tasks
- **Compatible** with your applications
- Fast, predictable performance
- Simple and **fast to scale**
- Secure
- Cost-effective

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, which allows you 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, the Amazon AWS command line interface, or the Amazon RDS API.

How Amazon RDS Backups Work

Automatic Backups:

- Restore your database to a point in time.
- Are enabled by default.
- Let you choose a retention period up to 35 days.

Manual Snapshots:

- Let you build a new database instance from a snapshot.
- Are initiated by the user.
- Persist until the user deletes them.
- Are stored in Amazon S3.

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.

Amazon RDS Security

- Run your DB instance in an **Amazon VPC**.
- Use **IAM policies** to grant access to Amazon RDS resources.
- Use **security groups**.
- Use Secure Socket Layer (**SSL**) connections with DB instances (Amazon Aurora, Oracle, MySQL, MariaDB, PostgreSQL, Microsoft SQL Server).
- Use Amazon RDS **encryption** to secure your RDS instances and snapshots at rest.
- Use network encryption and transparent data encryption (**TDE**) with Oracle DB and Microsoft SQL Server instances.
- Use the security features of your DB engine to **control access** to your DB instance.

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.

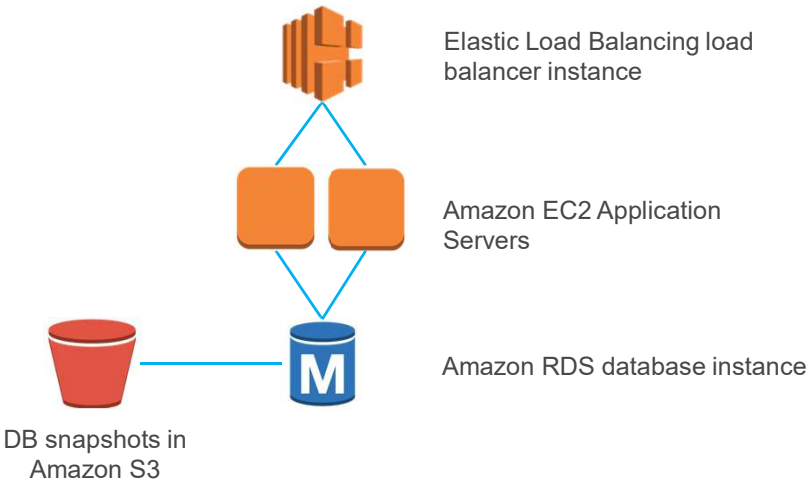
- Run your DB instance in an Amazon virtual private cloud (VPC) for the greatest possible network access control.
- Use AWS Identity and Access Management (IAM) policies to assign permissions that determine who is allowed to manage RDS resources. For example, you can use AWS IAM to determine who is allowed 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

A Simple Application Architecture



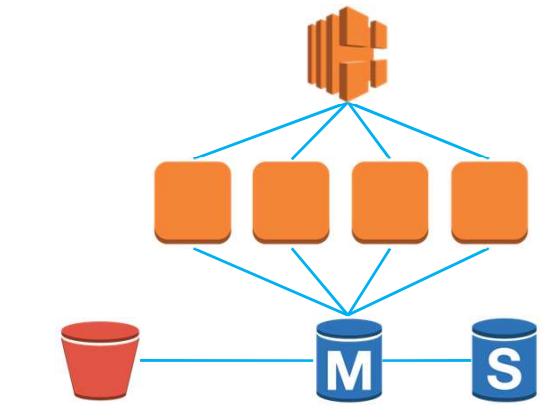
The slide shows a simple application stack with an application running in an Amazon EC2 instance supported by a master database running in an Amazon RDS database instance. Presenting the application behind an elastic load balancer allows for compute resiliency and scaling features such as Auto Scaling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

Multi-AZ RDS Deployment

- With **Multi-AZ** operation, your database is **synchronously replicated to another Availability Zone** in the same AWS Region.
- **Failover** to the standby **automatically** occurs 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 Availability Zone 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. Because 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.

A Resilient, Durable Application Architecture



Elastic Load Balancing load balancer instance

Application, in Amazon EC2 instances

Amazon RDS database instances: Master and Multi-AZ standby

DB snapshots in Amazon S3

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.

For more information, see:
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html

Amazon RDS Best Practices

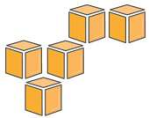
- **Monitor** your memory, CPU, and storage usage.
- Use **Multi-AZ** deployments to automatically provision and maintain a synchronous standby in a different Availability Zone.
- Enable **automatic backups**.
- Set the **backup window** to occur during the daily low in WriteIOPS.
- To increase the I/O capacity of a DB instance:
  - 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 optimized for **provisioned IOPS**.
  - Provision additional throughput capacity (if using provisioned IOPS storage).
- If your client application is caching the DNS data of your DB instances, set a TTL of less than 30 seconds.
- **Test** failover for your DB instance.

- 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 (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 is no longer 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



Part 3

Amazon DynamoDB

amazon web services | Training and Certification

Part 3: Amazon DynamoDB

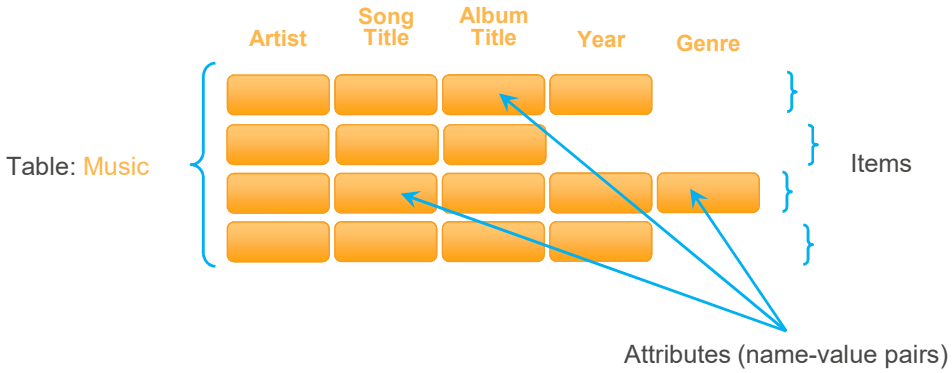
Amazon DynamoDB

Amazon DynamoDB

- Allows you to store any amount of data with **no limits**.
- Provides fast, predictable performance using **SSDs**.
- Allows you to easily provision and change the **request capacity** needed for each table.
- Is a **fully managed, NoSQL** database service.

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 multiple 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.

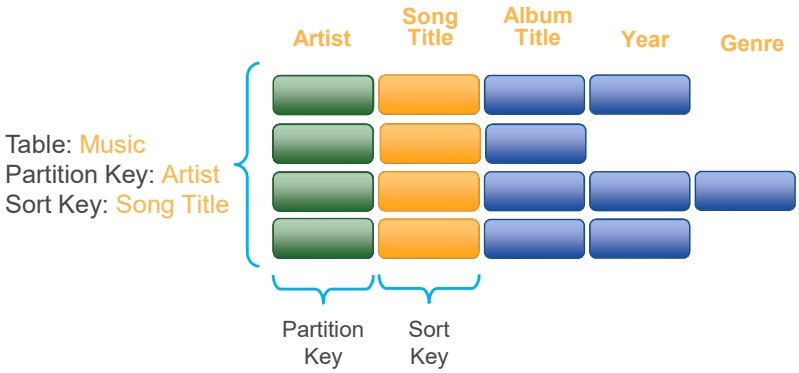
DynamoDB Data Model



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

Primary Keys



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

Provisioned Throughput

You specify how much **provisioned throughput capacity** you need for reads and writes.

Amazon DynamoDB allocates the necessary machine resources to meet your needs.

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.

Supported Operations

Query:

- Query a table using the partition key and an optional sort key filter.
- If the table has a secondary index, query using its key.
- It is the **most efficient way to retrieve items** from a table or secondary index.

Scan:

- You can scan a table or secondary index.
- Scan reads every item – **slower than querying**.

You can use conditional expressions in both Query and Scan operations.

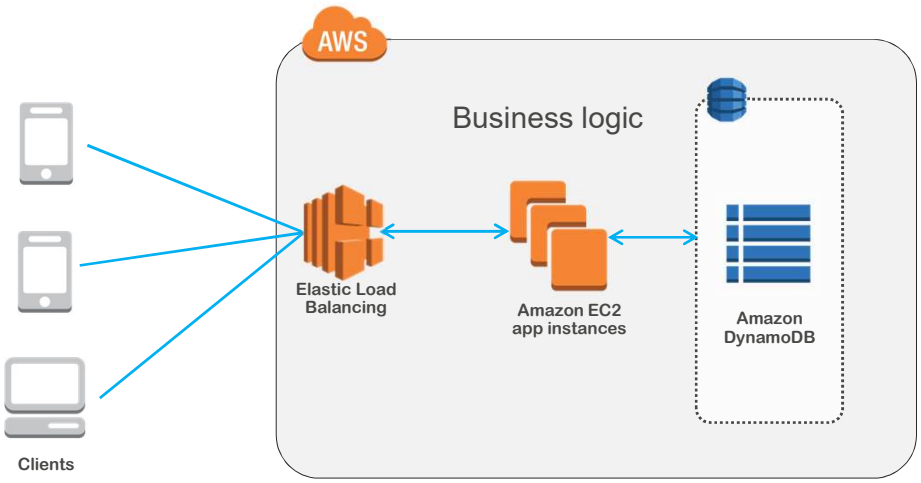
The Query operation enables you to query a table using the partition key and an optional sort key filter. If the table has a secondary index, you can also query the index using its key. You can query only tables that have a composite primary key (partition key and sort key). You can also query any secondary 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.

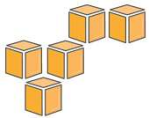
You can use conditional expressions in both the Query and Scan operations to control which items are returned.

For more information, see:
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/QueryAndScan.html

Simple Application Architecture



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



Part 4

Choosing a Database Service

amazon web services | Training and Certification

Part 4: Choosing a Database Service

Amazon RDS and Amazon DynamoDB

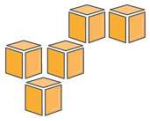
Factors	Relational (Amazon RDS)	NoSQL (Amazon DynamoDB)
Application Type	• Existing database apps • Business process–centric apps	• New web-scale applications • Large number of small writes and reads
Application Characteristics	• **Relational** data models, transactions • **Complex** queries, joins, and updates	• Simple data models, transactions • Range queries, simple updates
Scaling	Application or **DBA–architected** (clustering, partitions, sharding)	**Seamless, on-demand scaling** based on application requirements
QoS	• Performance–depends on data model, indexing, query, and storage optimization • Reliability and availability • Durability	• Performance–**Automatically optimized** by the system • Reliability and availability • Durability

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 and non-relational databases.

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 several 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.



In review…

- Databases Overview and Considerations
- Amazon Relational Database Service (RDS)
- Amazon DynamoDB
- Choosing a Database Service

Knowledge Assessment

amazon web services | Training and Certification

In review…

- Databases Overview and Considerations
- Amazon Relational Database Service (RDS)
- Amazon DynamoDB
- Choosing a Database Service

To complete this module, please remember to finish the corresponding knowledge assessment.

Knowledge Check

Q What are the basic building blocks of Amazon Relational Database Service (RDS)?

DB Instances

T/F Amazon DynamoDB allows you to store any amount of data with no limits.

True

T/F Scan is the most efficient way to retrieve items from a DynamoDB table.

False

Q You are creating a resilient, durable application using Amazon RDS. In addition to Amazon RDS's automatic backups, what feature should you use to ensure that your backups are durable retained?

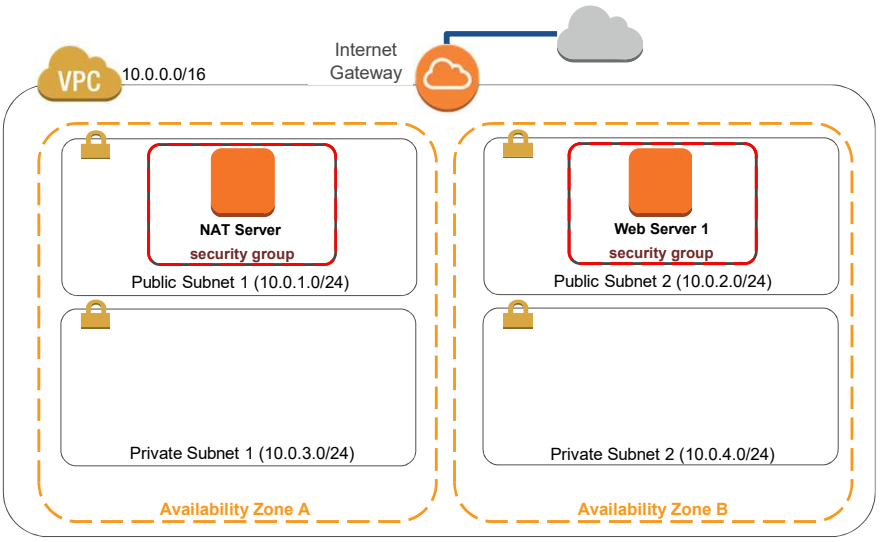
Manual Snapshots

CCA Lab-05

Build Your Database Server and Interact with Your Database using an Application

(Approx. 45 minutes)

What You're Starting With



Lab 5 Overview

1

- Create a **database server**
- Create a **security group**
- Create a **DB subnet group**
- Create an **Amazon RDS DB instance**
- Get database connection string

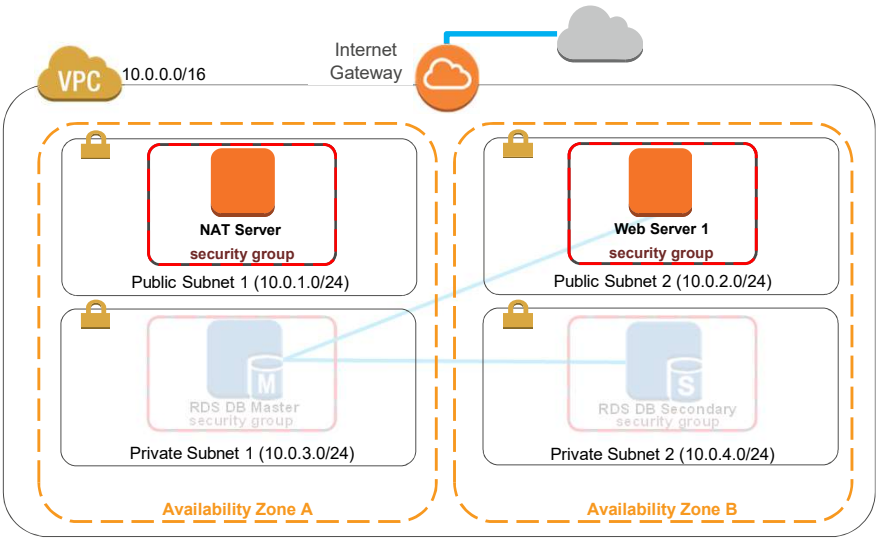
While waiting for the database to start, grab a coffee!

2

- Open a web application from a browser
- Insert DB connection string
  - App will populate a table with records
  - App will display records for a table

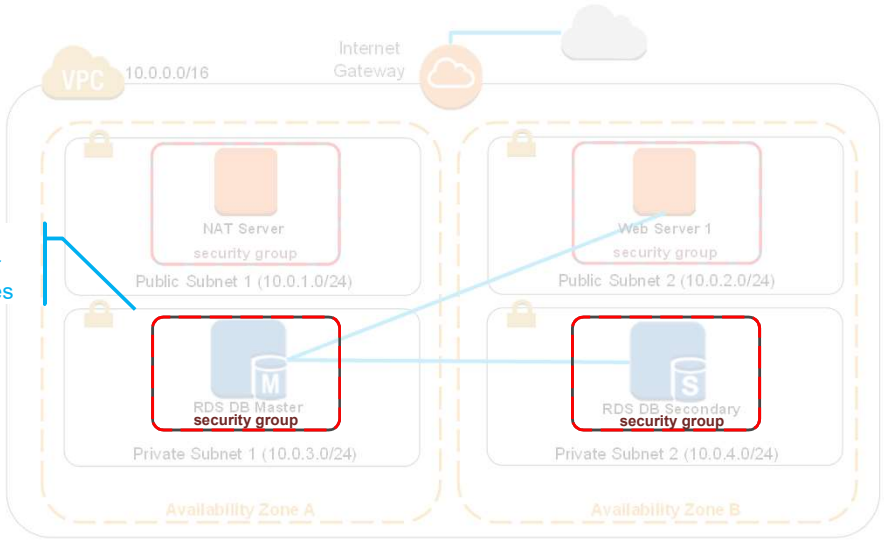
Lab 2 – Build Your Database Server and Connect to It

Your lab environment starts with the following resources

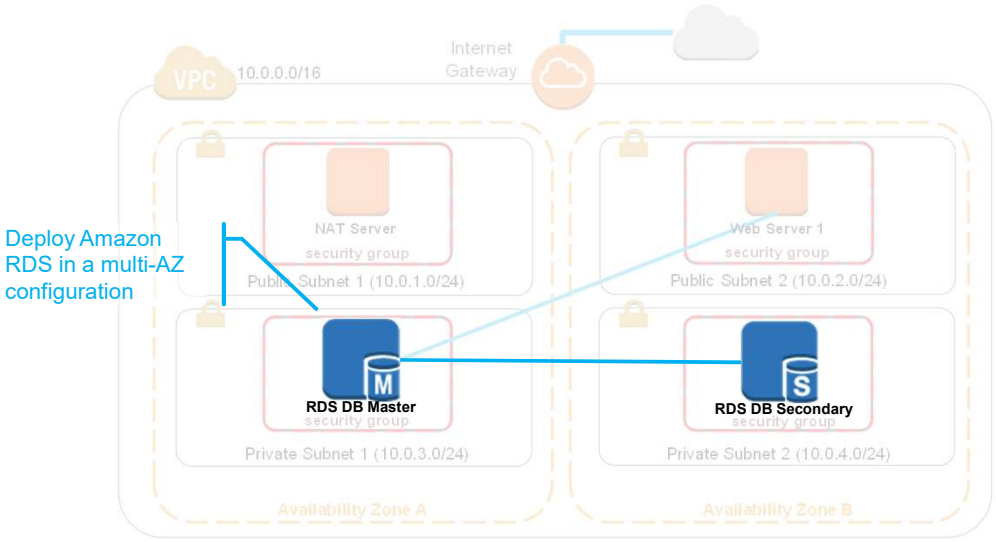


Build Your Database Server and Connect to It

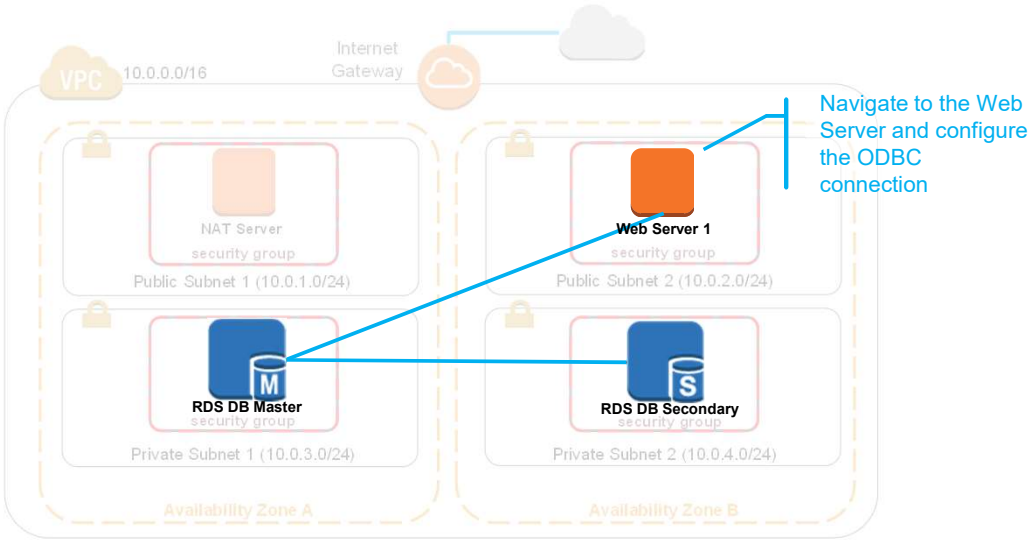
You will create a security group for the RDS instances



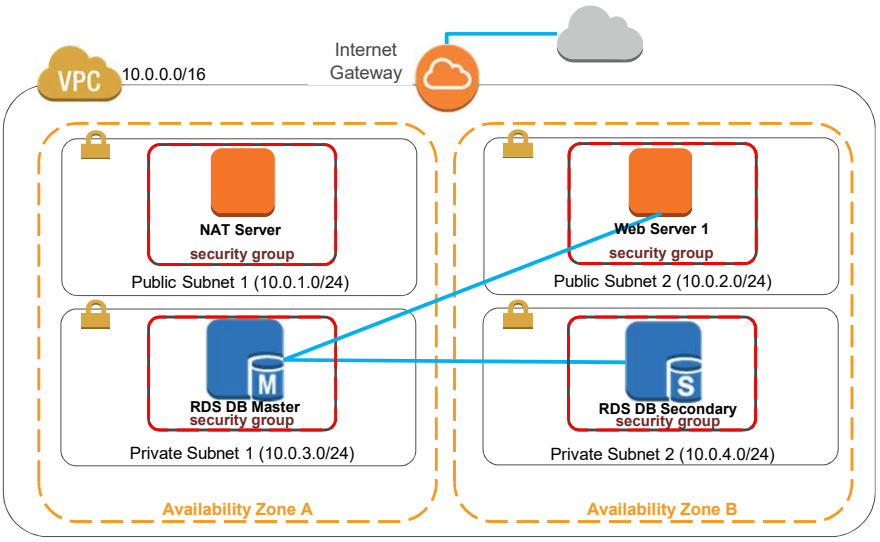
Build Your Database Server and Connect to It

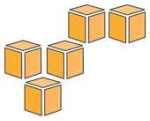


Build Your Database Server and Connect to It



Build Your Database Server and Connect to It





Up Next…

LAB 05 - Build Your Database Server and Interact with Your Database using an Application

CCA 2.04 - AWS Elasticity and Management Tools

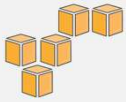
amazon web services | Training and Certification

If you haven't completed it already, please do **Lab 5,** where you will build a database server that interacts with an application. Refer back to the Welcome module for instructions on accessing the lab environment.

Be sure to complete the lab before continuing with **CCA 2.04** covering elasticity with AWS Auto Scaling and AWS management tools.

© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Do not speak over this slide – just let it play for 8 seconds.



CCA Unit 2 – Getting Started with AWS

CCA 2.04: AWS Elasticity and Management Tools

CCA 2.01 AWS Compute, Storage, and Networking
CCA 2.02 AWS Security, Identity, and Access Management
CCA 2.03 AWS Database Options
CCA 2.04 AWS Elasticity and Management Tools

amazon web services | Academy

Welcome to Module CCA 2.03 –AWS Database Options

What's In This Module?

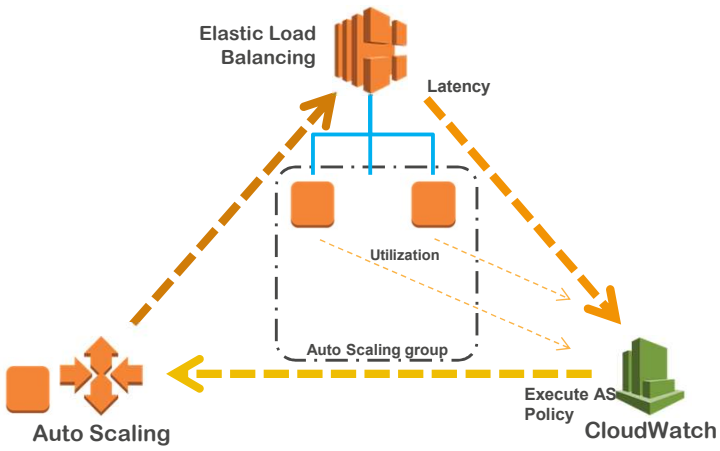
- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling
- AWS Trusted Advisor

This module covers…

- AWS Shared Responsibility Model
- AWS Identity and Access Management (IAM)
- AWS CloudTrail

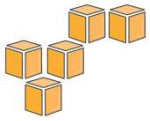
CCA 2.99: Appendix

Triad of Services



amazon web services | Training and Certification

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.



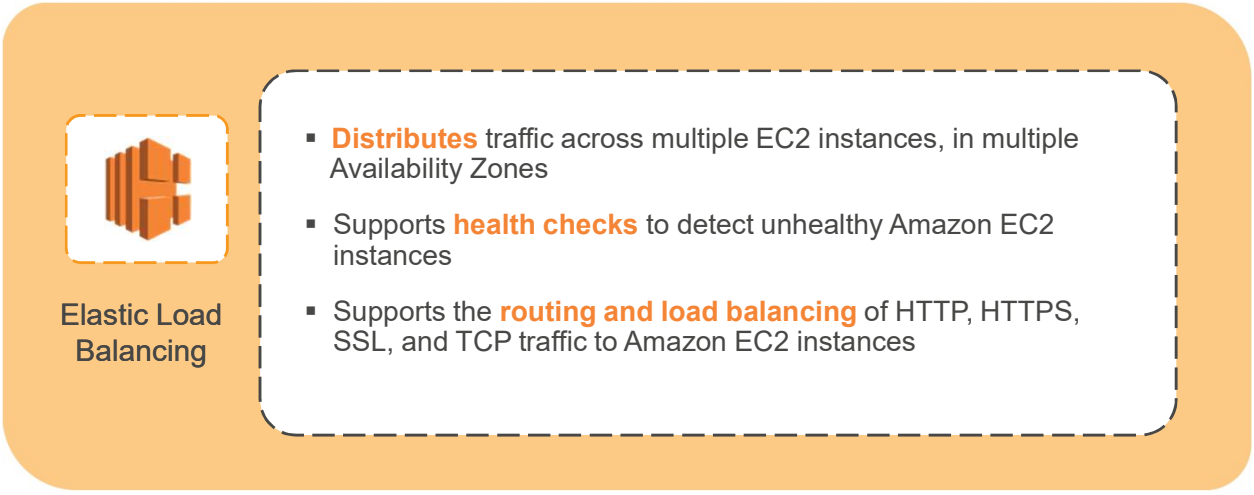
Part 1

Elastic Load Balancing

amazon web services | Training and Certification

Part 1: Elastic Load Balancing

Elastic Load Balancing



Understand Elastic Load Balancing (ELB) concepts including:

- Classic Load Balancer
- Application Load Balancer
- Load Balancer Comparison

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater fault tolerance in your applications, and seamlessly provides 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.

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.
- AWS SDKs: Language-specific APIs that take care of many of the connection

details, such as calculating signatures, handling request retries, and error handling.

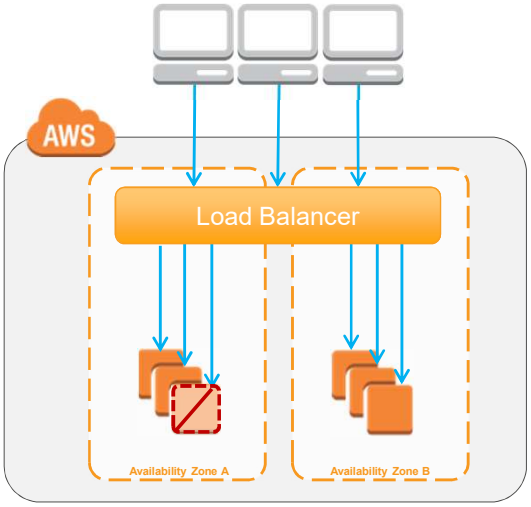
- Query API: Low-level API actions that you call using HTTPS requests.

For more information, see:

- AWS SDKs: https://aws.amazon.com/tools/#SDKs
- Application Load Balancer Query API: http://docs.aws.amazon.com/elasticloadbalancing/latest/APIReference/Welcome.html
- Classic Load Balancer Query API: http://docs.aws.amazon.com/elasticloadbalancing/2012-06-01/APIReference/Welcome.html

Classic Load Balancer - How It Works

Register instances with your load balancer.



A load balancer distributes incoming application traffic across multiple EC2 instances in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Your load balancer serves as a single point of contact for clients. This increases the availability of your application. You can add and remove instances from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing can scale to the vast majority of workloads automatically.

A listener checks for connection requests from clients, using the protocol and port that you configure, and forwards requests to one or more registered instances using the protocol and port number that you configure. You add one or more listeners to your load balancer.

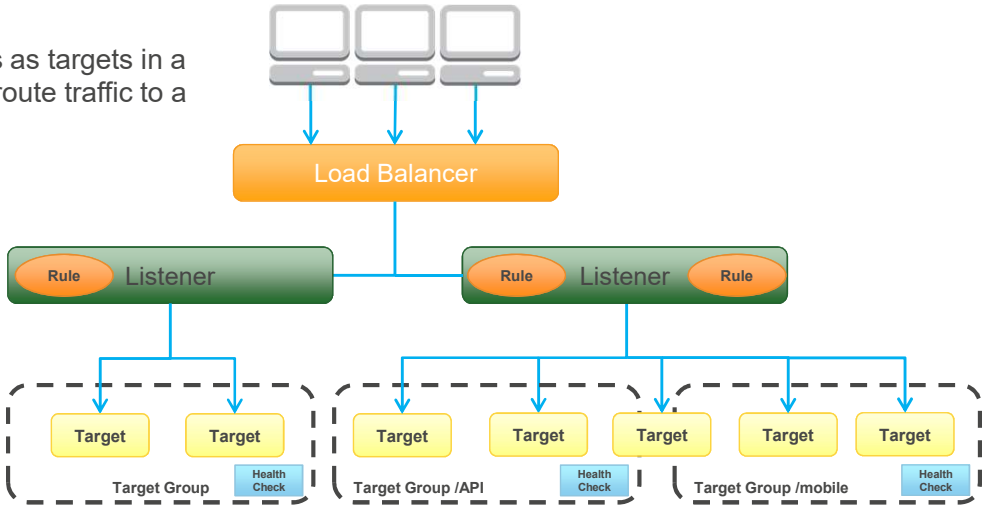
You can configure health checks, which are used to monitor the health of the of the registered instances, so that the load balancer can send requests only to the healthy instances.

By default, the load balancer distributes traffic evenly across the Availability Zones that you enable for your load balancer. To distribute traffic evenly across all registered instances in all enabled Availability Zones, enable *cross-zone load balancing* on your load balancer.

For more information, see:
http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html

Application Load Balancer – How It Works

Register instances as targets in a target group, and route traffic to a target group.



An Application Load Balancer (ALB) functions at the application layer (layer 7 of the OSI model). The ALB makes routing decisions based on the content of the application traffic in the HTTP messages.

With ALB, we require you to enable multiple Availability Zones.

A load balancer serves as the single point of contact for clients. You add one or more listeners to your load balancer.

A listener checks for connection requests from clients, using the protocol and port that you configure, and forwards requests to one or more target groups, based on the rules that you define. Each rule specifies a target group, condition, and priority. When the condition is met, the traffic is forwarded to the target group. You must define a default rule for each listener, and you can add rules that specify different target groups based on the content of the request (also known as *content-based routing*).

Each target group routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify. You can register a target with multiple target groups. You can configure health checks on a per target group basis. Health checks are performed on all targets registered to a target group that is specified in a listener rule for your load balancer.

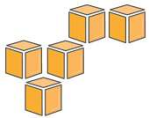
The slides shows the basic components of an ALB. Each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups. The listener on the right has been configured with a rule to route /API requests to one target and /mobile requests to another.

For more information, see:
http://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html

Load Balancer Comparison

	Classic Load Balancer	ALB
Includes support for:	✓ EC2-Classic ✓ VPC ✓ TCP and SSL listeners ✓ Sticky sessions	✓ Path-based routing ✓ Routing requests to multiple services on a single EC2 instance ✓ Containerized applications ✓ Monitoring the health of each service independently

ALB access logs contain additional information and are stored in compressed format, and ALB has improved load balancer performance.
For more information, see:
http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html



Part 2

Amazon CloudWatch

amazon web services | Training and Certification

Part 2: Amazon CloudWatch

Amazon CloudWatch

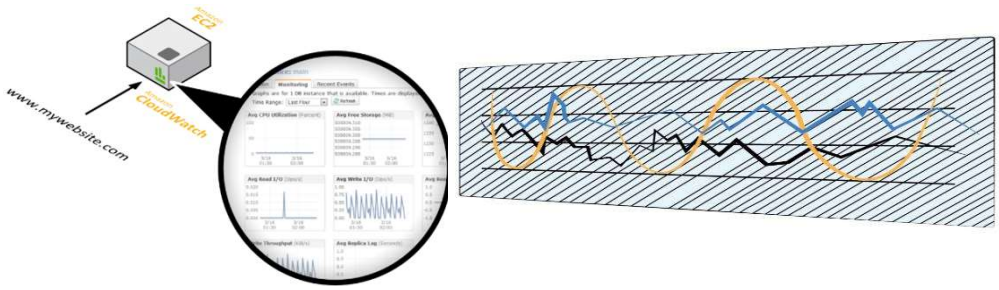
Amazon CloudWatch

- A **monitoring service** for AWS cloud resources and the applications you run on AWS
- **Visibility into** resource utilization, operational performance, and overall demand patterns
- **Custom application-specific** metrics of your own
- **Accessible** via AWS Management Console, APIs, SDK, or CLI

Amazon 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.

Amazon CloudWatch Facts

- Monitor other AWS resources
  View graphics and statistics
- Set Alarms



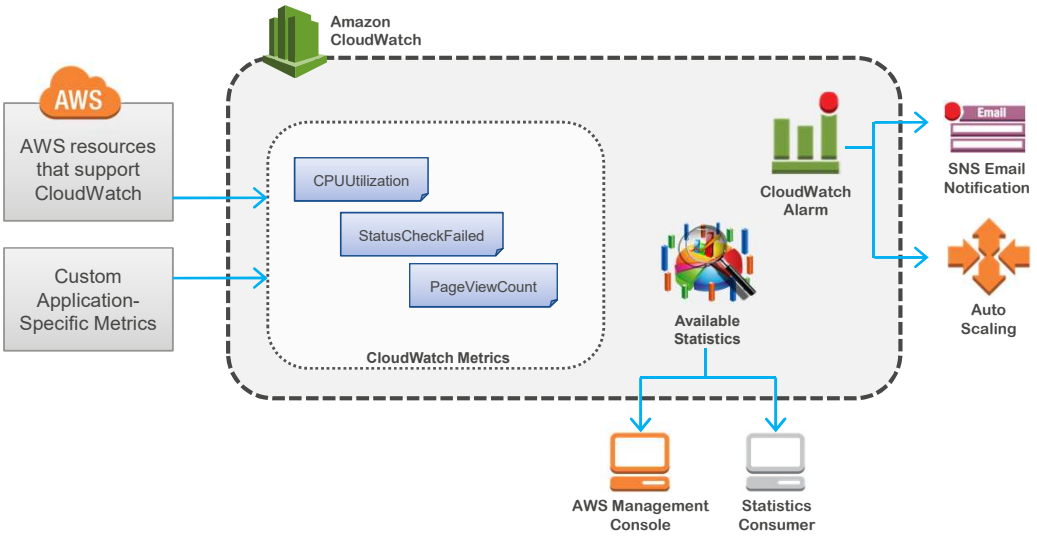
For Amazon EC2 instances, Amazon CloudWatch basic monitoring collects and reports metrics for CPU utilization, data transfer, and disk usage activity from each Amazon EC2 instance at a five-minute frequency. Amazon CloudWatch detailed monitoring provides these same metrics at one-minute intervals, and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

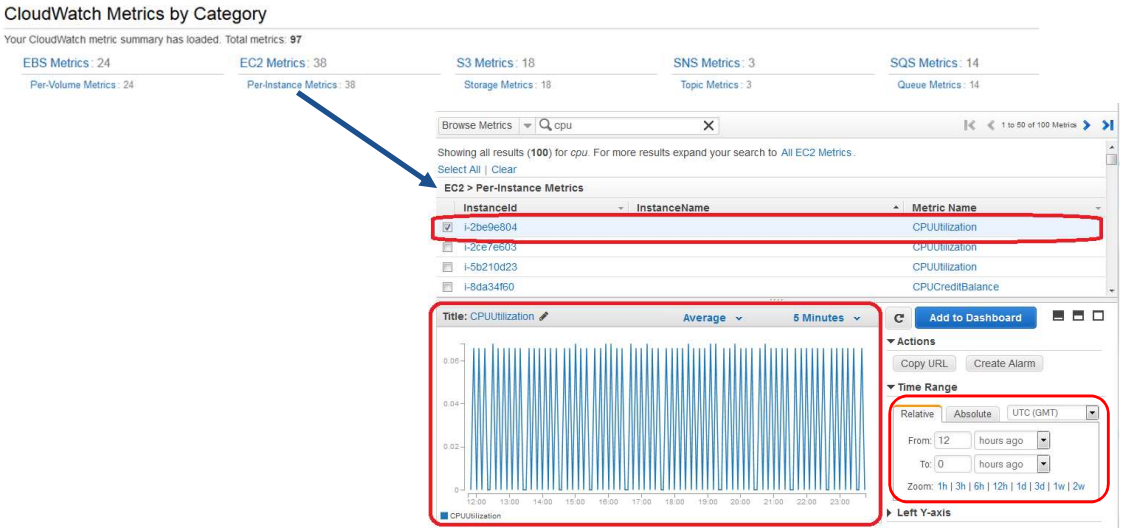
- http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/AlarmThatSendsEmail.html
- https://aws.amazon.com/cloudwatch/details/#other-aws-resource-monitoring
- https://aws.amazon.com/blogs/aws/new-cloudwatch-events-track-and-respond-to-changes-to-your-aws-resources/

Amazon CloudWatch Architecture

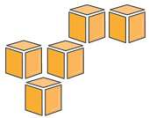


Amazon CloudWatch is a metrics repository. AWS products put metrics into the repository, and you retrieve statistics based on the metrics. Statistics can be graphically presented in the CloudWatch console.

CloudWatch Metrics Examples



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



Part 3

Auto Scaling

amazon web services | Training and Certification

Part 3: Auto Scaling

Auto Scaling

Auto Scaling

- **Scale** your Amazon EC2 capacity **automatically**
- Well-suited for applications that experience **variability in usage**
- Available at no additional charge

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 Benefits

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.

Launch Configurations

A **launch configuration** is a template that an Auto Scaling group uses to launch EC2 instances.

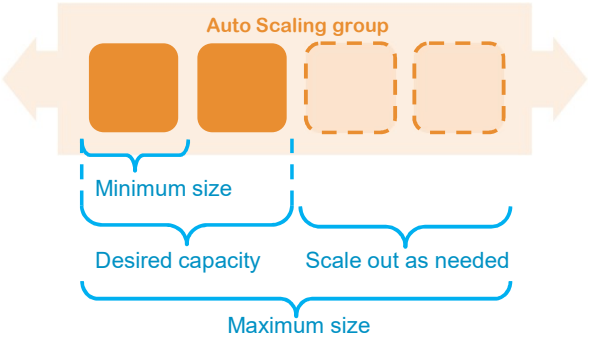
When you create a launch configuration, you can specify:

- AMI ID
- Instance type
- Key pair
- Security groups
- Block device mapping
- User data

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.

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**, Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

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.

Dynamic Scaling

You can create a scaling policy that uses **CloudWatch alarms** to determine when your **Auto Scaling** group should…

You can use alarms to monitor:

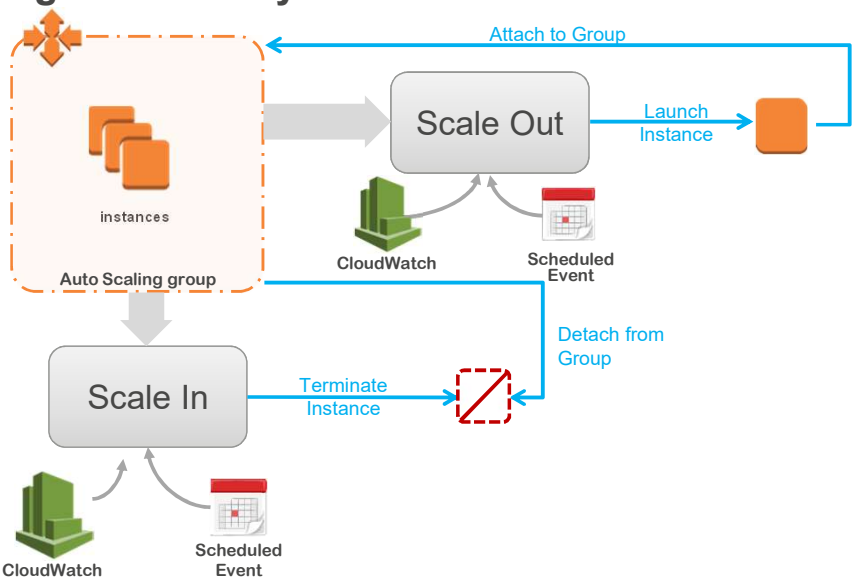
- Any of the metrics that AWS services send to CloudWatch
- Your own **custom metrics**

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:

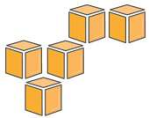
- http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/as-scale-based-on-demand.html
- http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/policy_creating.html

Auto Scaling Basic Lifecycle



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.



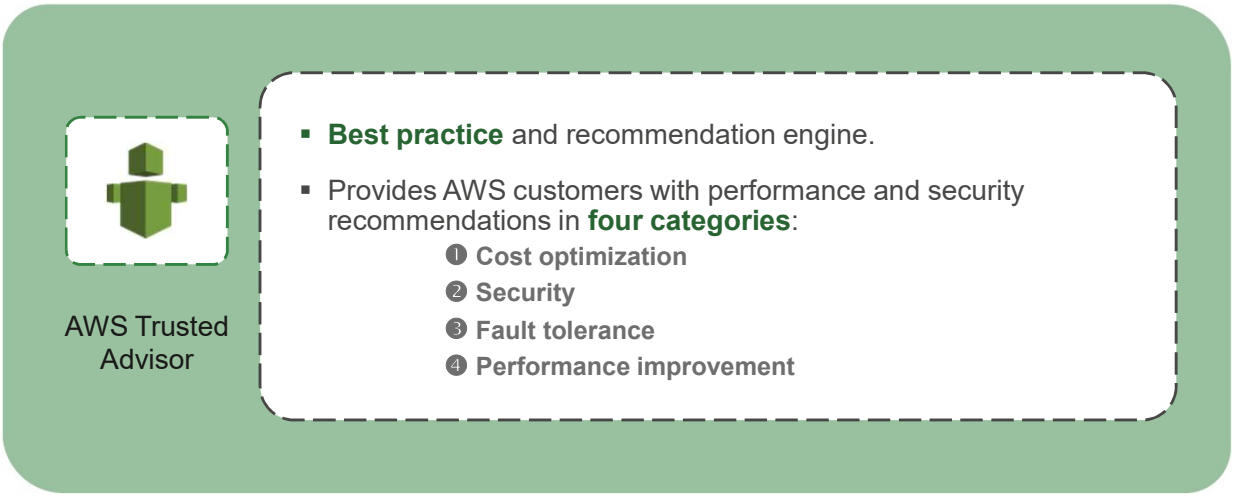
Part 4

AWS Trusted Advisor

amazon web services | Training and Certification

Part 4: AWS Trusted Advisor

AWS Trusted Advisor



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

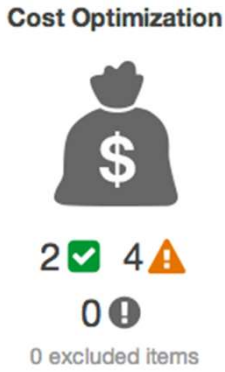
- Red: action recommended
- Yellow: investigation recommended
- Green: no problem detected

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.

Best Practices

Cost Optimization
Security
Fault Tolerance
Performance Improvement

- Amazon EC2 Reserved Instance Optimization
- Low-utilization Amazon EC2 Instances
- Idle load balancers
- Underutilized Amazon EBS volumes
- Unassociated Elastic IP addresses
- Amazon RDS idle DB instances



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 four 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.

Best Practices

Cost Optimization
Security
Fault Tolerance
Performance Improvement

- Security groups
- AWS IAM use
- Amazon S3 bucket permissions
- MFA on toot Account
- AWS IAM password policy
- Amazon RDS security group access risk

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.

Best Practices

Cost Optimization
Security
Fault Tolerance
Performance Improvement

- Amazon EBS Snapshots
- Load balancer optimization
- Auto Scaling Group Resources
- Amazon RDS Multi-AZ
- Amazon Route 53 name server delegations
- ELB connection draining

AWS Trusted Advisor helps you increase the availability and redundancy of your AWS application by taking advantage of autoscaling, 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.

Performance Improvement

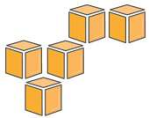
Cost Optimization
Security
Fault Tolerance
Performance Improvement

- High-utilization Amazon EC2 instances
- Service limits
- Large number of rules in EC2 security group
- Over-utilized Amazon EBS magnetic volumes
- Amazon EC2 to EBS throughput optimization
- Amazon CloudFront alternate domain names

AWS Trusted Advisor helps you improve the performance of your service by checking your service limits, ensuring that 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 four 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 over-utilized 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.



In review…

- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling
- AWS Trusted Advisor

Knowledge Assessment

amazon web services | Training and Certification

In review…

- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling
- AWS Trusted Advisor

To complete this module, please remember to finish the corresponding knowledge assessment.

Knowledge Check

T/F

Auto Scaling helps you ensure that you have the correct number of EC2 instances available to handle the load for your application.

True

Q

What feature would you use with an auto scaling policy to determine when your auto scaling group should scale out/in?

Amazon CloudWatch alarms

Q

You have an application composed of individual services and need to route a request to a service based on the content of the request. What type of load balancer should you use?

Application Load Balancer

Q

Which AWS service serves as a best practice and recommendation engine?

AWS Trusted Advisor

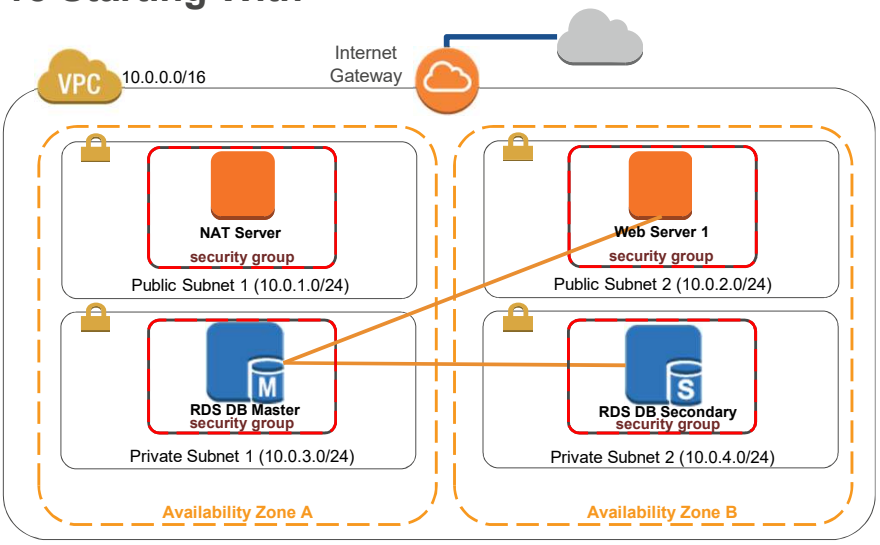
CCA Lab-06

Scale and Load Balance Your Architecture

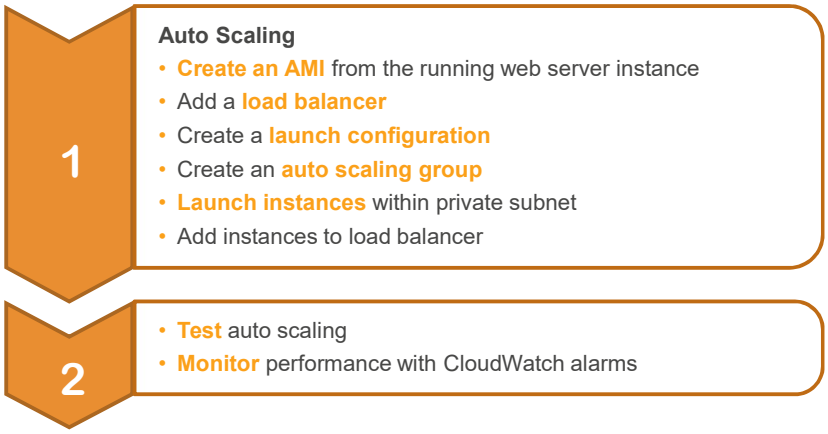
(Approx. 45 minutes)

amazon web services | Training and Certification

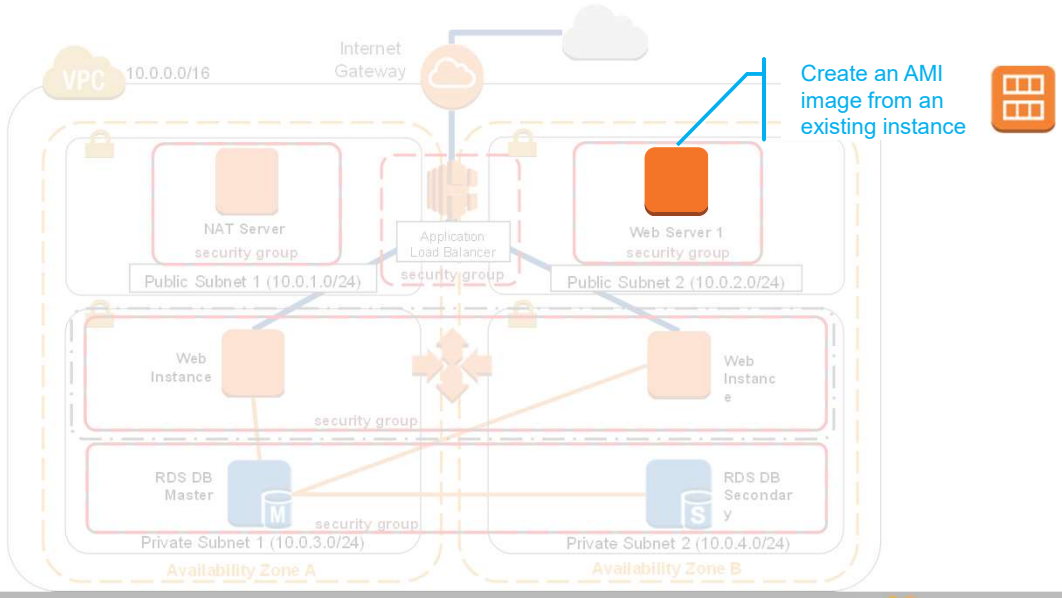
What You're Starting With



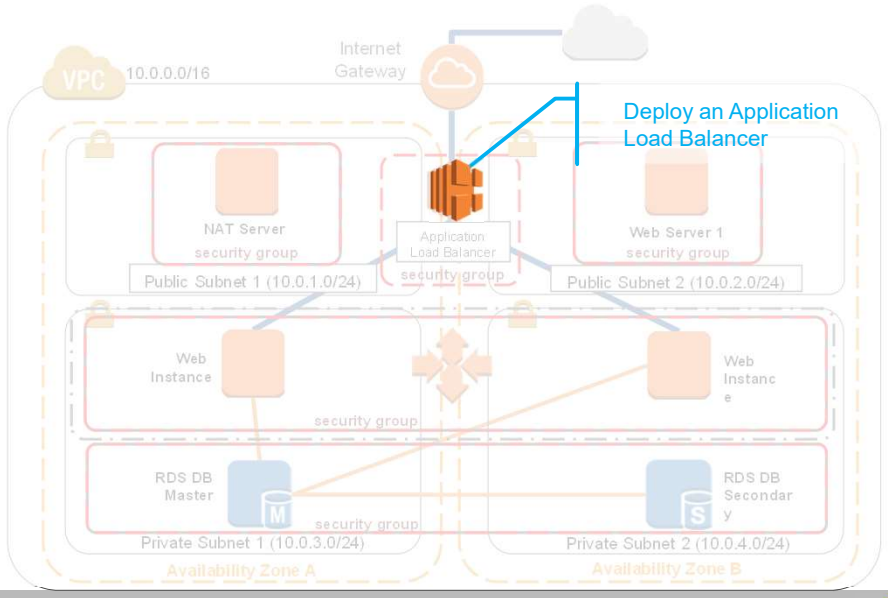
Lab 6 Overview



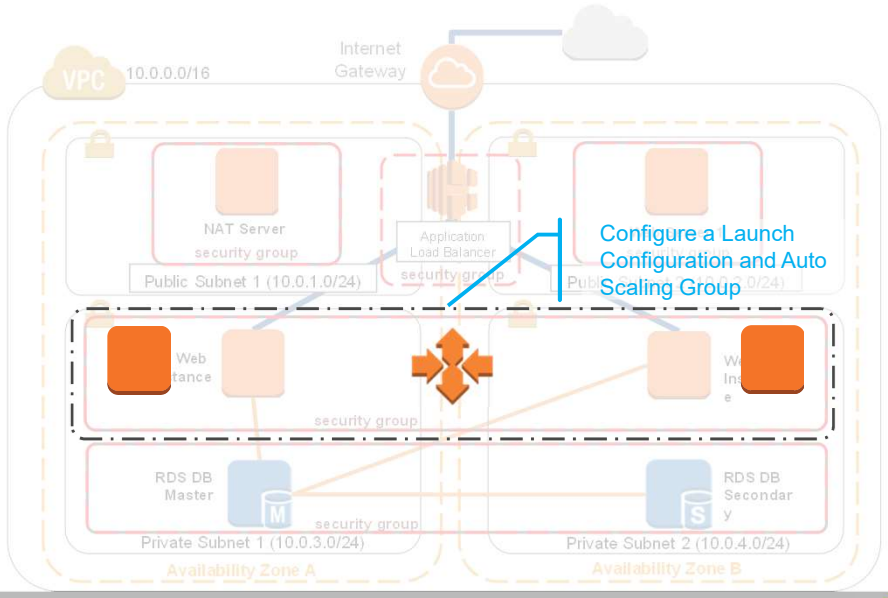
Scale and Load Balance your Architecture



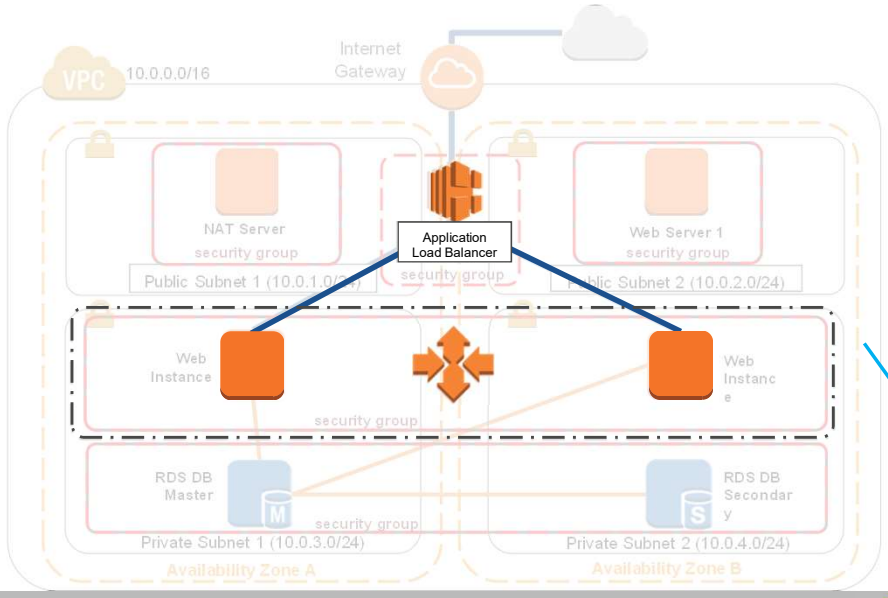
Scale and Load Balance your Architecture



Scale and Load Balance your Architecture

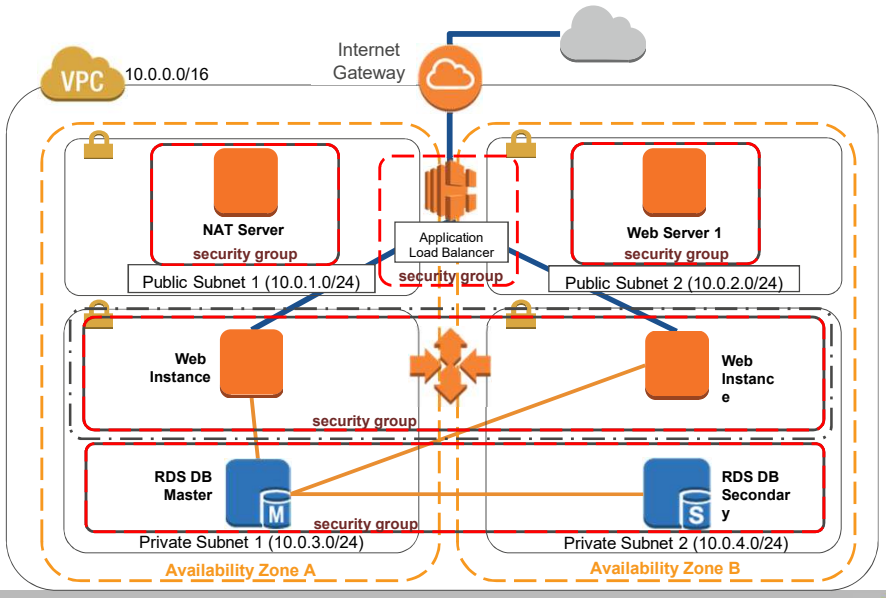


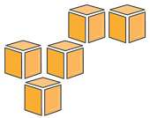
Scale and Load Balance your Architecture



Test and monitor your Auto Scaling Group

Scale and Load Balance your Architecture





Up Next…

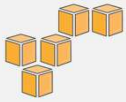
LAB 06 - Scale and Load Balance Your Architecture

UNIT 3: Architecting on AWS

amazon web services | Training and Certification

If you haven't completed it already, please do **Lab 5,** where you will build a database server that interacts with an application. Refer back to the Welcome module for instructions on accessing the lab environment.

Be sure to complete the lab before continuing with **CCA 2.04** covering elasticity with AWS Auto Scaling and AWS management tools.



CCA Unit 2 – Getting Started with AWS

Appendix

CCA 2.01 AWS Compute, Storage, and Networking

CCA 2.02 AWS Security, Identity, and Access Management

CCA 2.03 AWS Database Options

CCA 2.04 AWS Elasticity and Management Tools

amazon web services | Academy

Appendix

Appendix
AWS Introduction and History

Cloud Computing Concepts

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.

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.

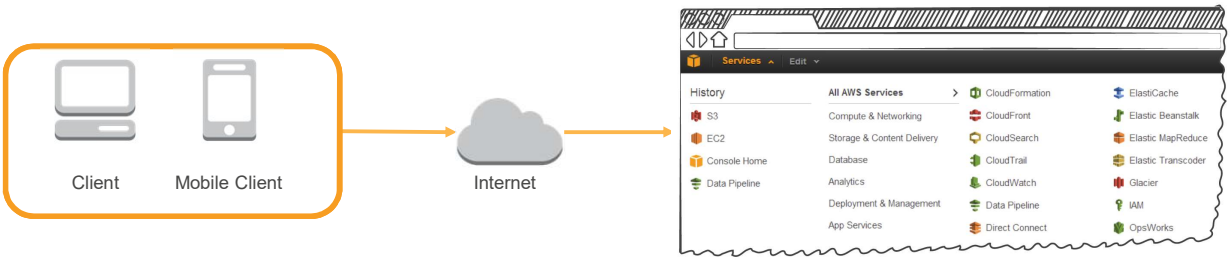
Word cloud created at: http://www.tagxedo.com/

Essential Characteristics of Cloud Computing

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

On-Demand Self Services & Broad Network Access

- User provisions computing resources as needed.
- User interacts with cloud service provider through an online control panel.
- Clear solutions are available through a variety of network-connected devices and over varying platforms.

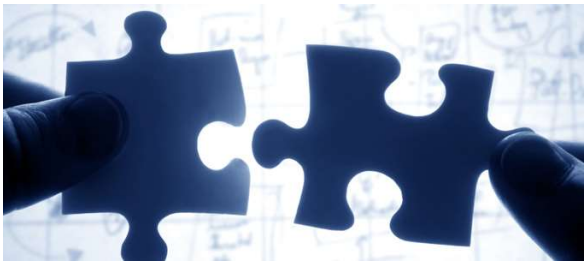


On-demand self services are an essential characteristic of cloud computing. The user provisions computing resources as needed and interacts with the cloud service provider through on online control panel, such as the AWS Management Console.

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 that 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 users' demands.

Measured Service

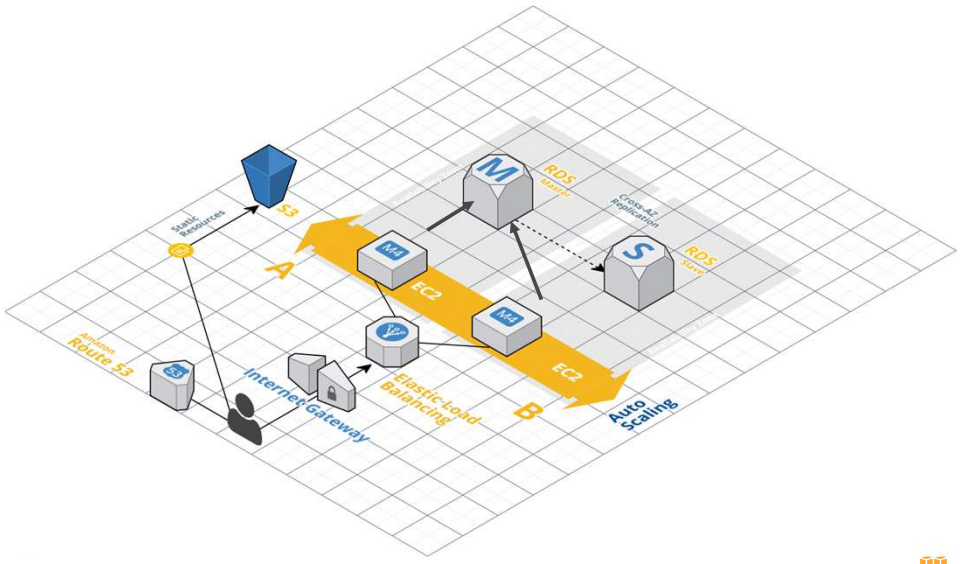
Pay for services as you go.

Electrical services analogy

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.

What Does My AWS Cloud Look Like?



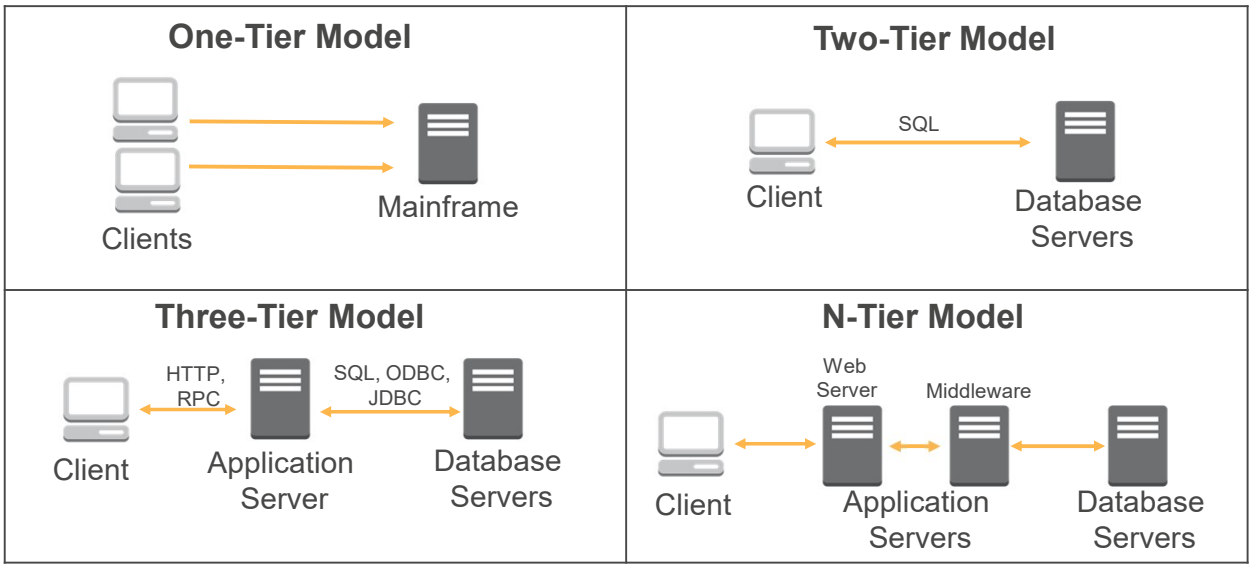
The slide shows the architecture of a highly scalable and reliable web application on AWS. This is an example of one possible configuration among millions.

For more information, see: http://docs.aws.amazon.com/gettingstarted/latest/wah-linux/web-app-hosting-intro.html

2 Appendix
AWS Foundational Services

Data Center Design Models

Application Design Model



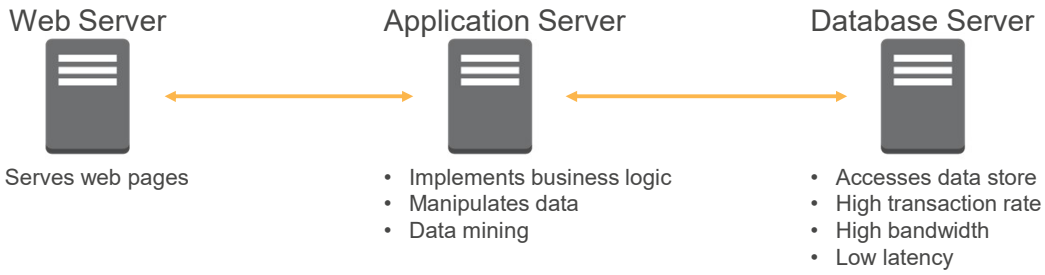
The one-tier model applies to clients (no local processing or storage) connected to a mainframe. This design is usually used for monolithic applications such as kiosks. This model has limited scalability and lacks flexibility.

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 is 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.

Web Services Model



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

Amazon EC2

AMI Types - Storage for the Root Device

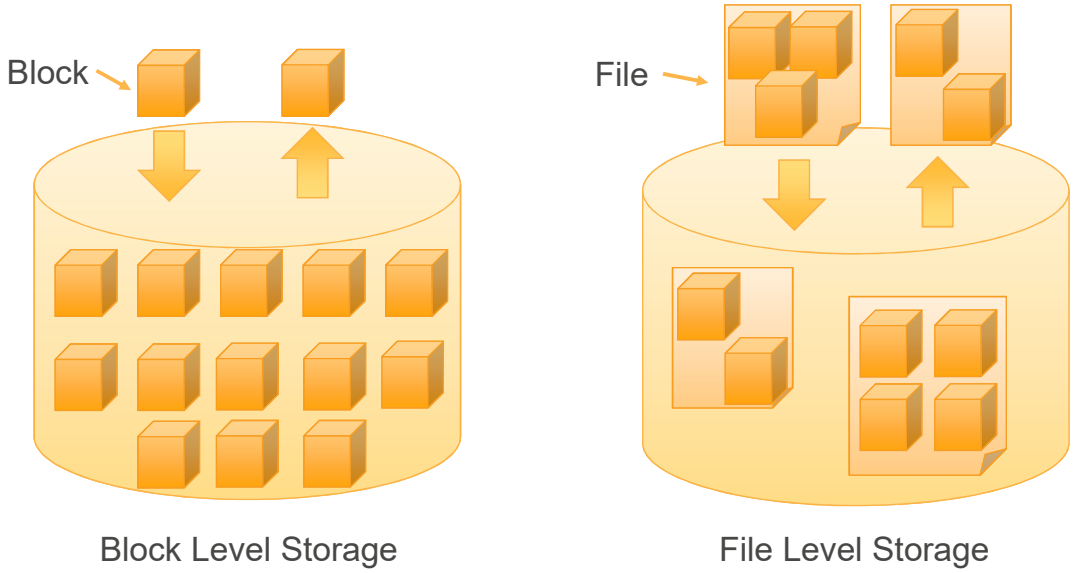
Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually < 1 minute	Usually < 5 minutes
Size limit	16 TiB	10 GiB
Data persistence	The root volume is deleted when the instance terminates. Data on any other Amazon EBS volumes persists after the instance is terminated.	Data on any 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, the root device for an instance is an EBS volume created from an EBS snapshot. When an AMI is instance-store backed, the root device for the instance was created from a template stored in Amazon S3. Key differences between the two categories of AMIs are shown in the slide.

Storage Concepts and Solutions

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

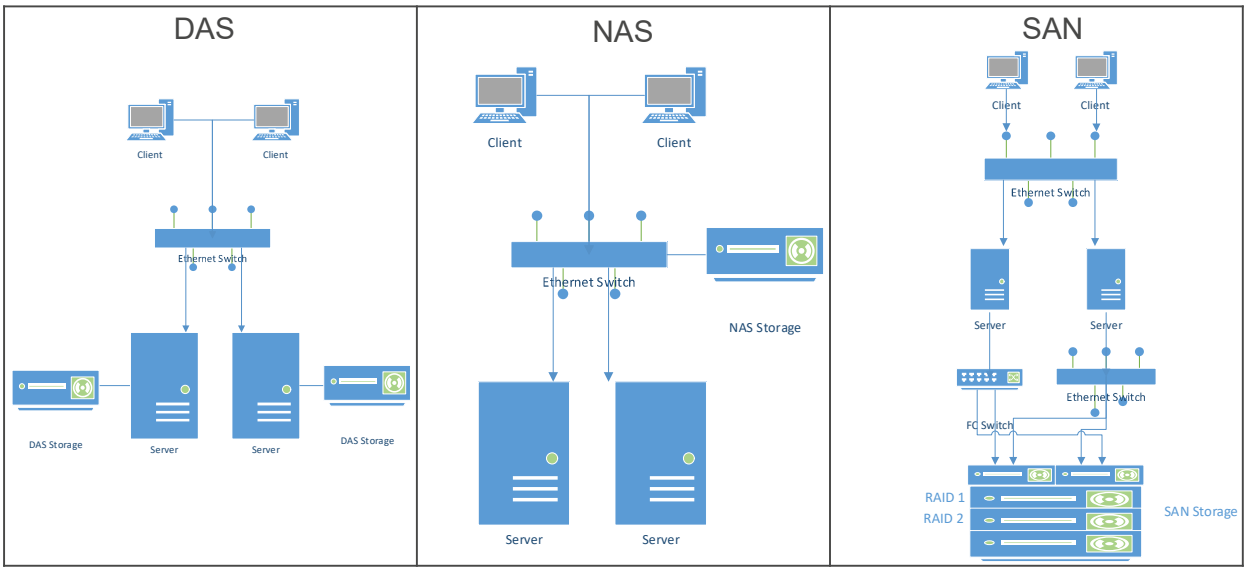
Block and File Level Storage



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 because they do not have to back up 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 is ideal choice.

Storage Technologies



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

Amazon S3 Buckets

- Organize the Amazon S3 namespace at the highest level.
- Identify the account responsible for storage and data transfer charges.
- Play a role in access control.
- Serve as the unit of aggregation for usage reporting.
- Have globally unique bucket names, regardless of the AWS region in which they were created.

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.

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

Amazon S3 Objects

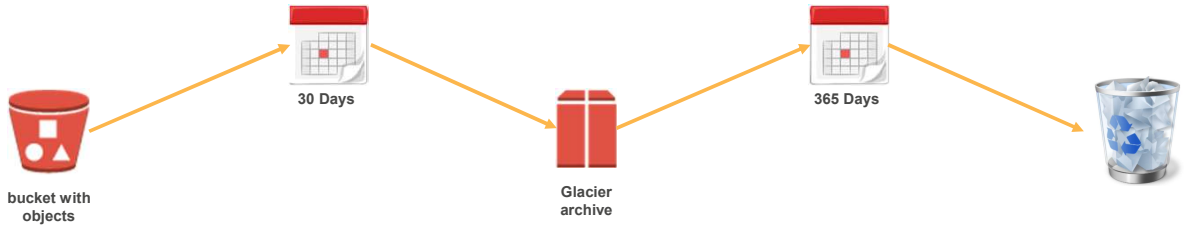
- Objects are the fundamental entities stored in Amazon S3.
- When using the console, you can think of them as files.
- **Objects consist of data and metadata**. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object.
  - Default metadata such as the date last modified
  - Standard HTTP metadata such as Content-Type
  - Custom metadata at the time the object is stored
  - A key that uniquely identifies as object within its bucket

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

Amazon S3 + Amazon Glacier

S3 Lifecycle policies allow you to delete or move objects based on age and set rules per S3 bucket.



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

EBS Performance

EBS Magnetic
- 40-200 IOPS

EBS General Purpose SSD
- SSD backed
- 3 IOPS / GB
- Burstable to 3,000 IOPS and up to 10,000 IOPS

EBS Provisioned IOPS SSD
- SSD backed
- Up to 20,000 IOPS consistently
- Up to 320 MB/s throughput

Amazon EBS volume types are shown in the slide.

For more information, see:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

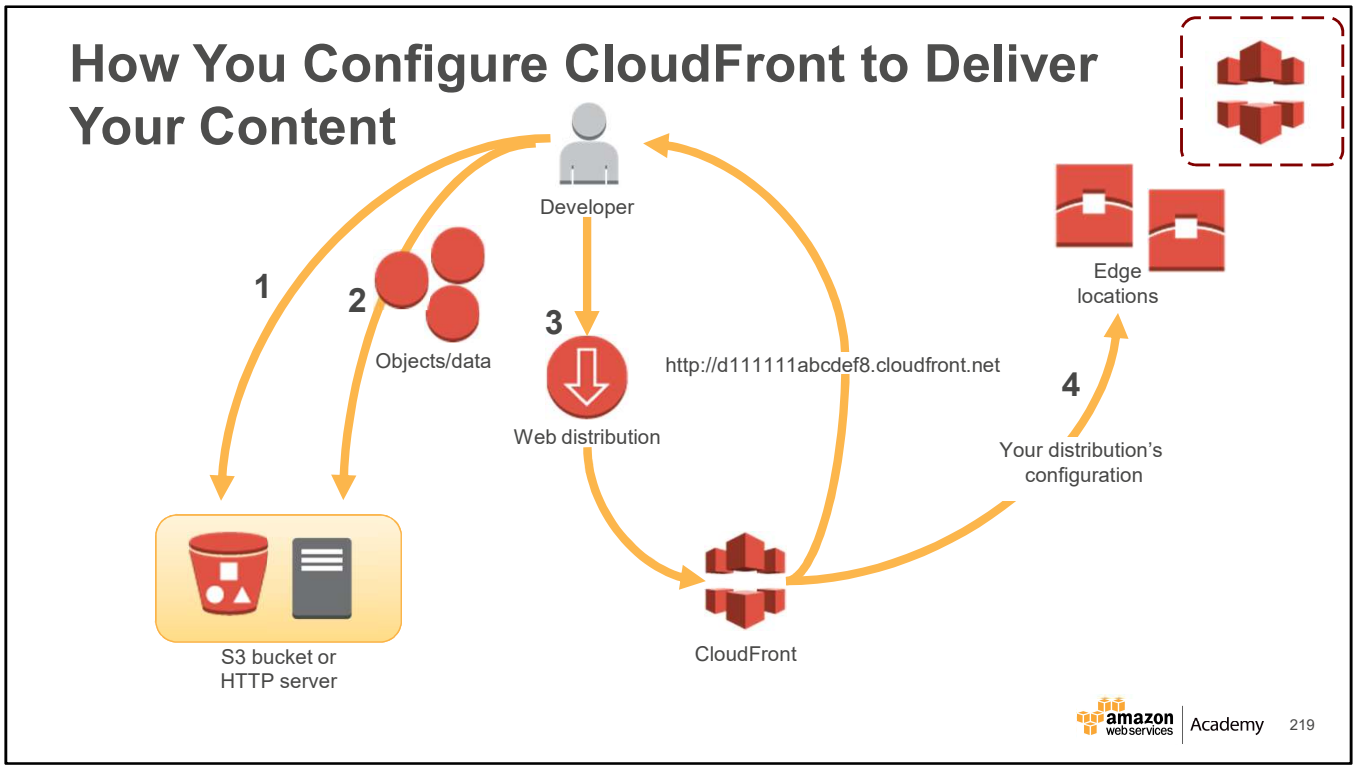
Amazon CloudFront

Amazon CloudFront

Amazon CloudFront

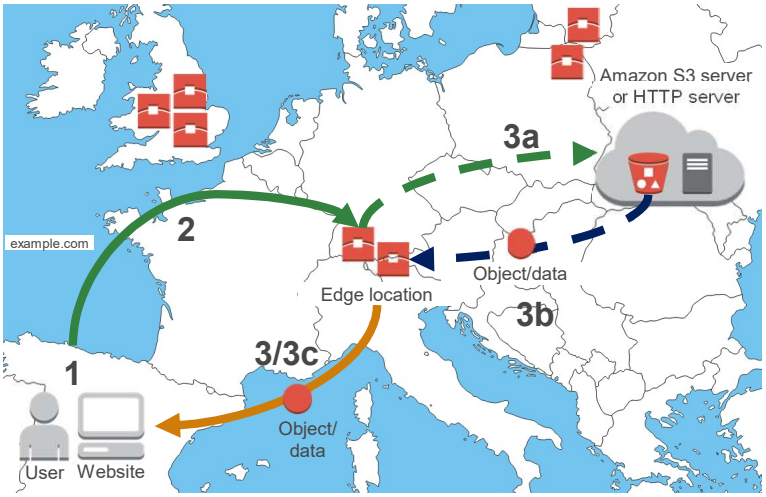
- Easy and cost effective way to **distribute content** to end users
- **Low latency, high data transfer speeds**
- Deliver your entire website, including static, dynamic, and streaming content using a global network of edge locations

Amazon CloudFront integrates with other AWS 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 to 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, 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.

How CloudFront Delivers Content to Your Users



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.

Networking Concepts

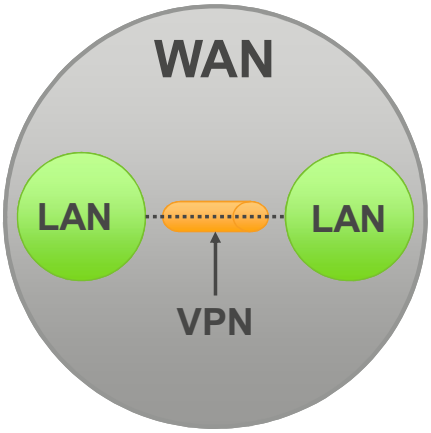
Understand typical networking devices, protocols, and services.

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.

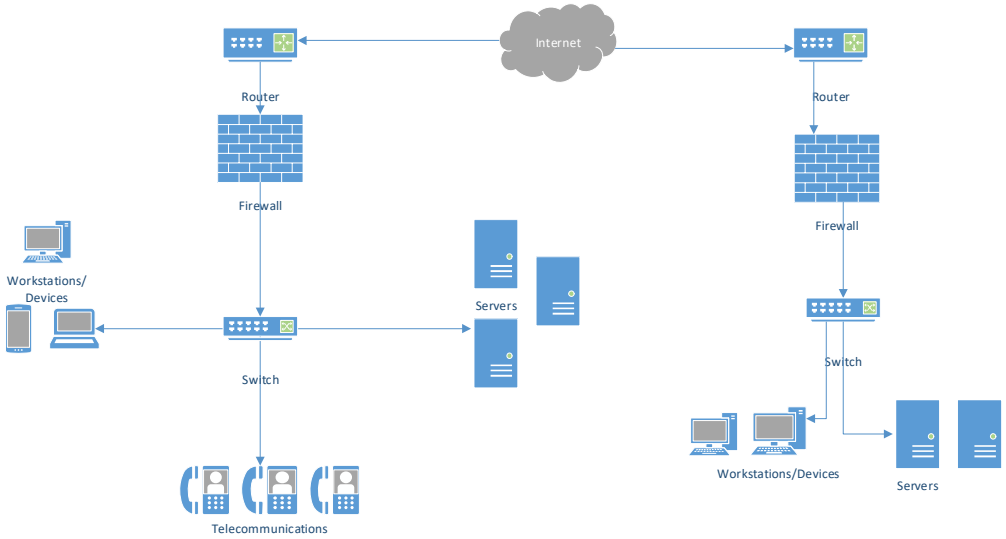
A virtual private network (VPN) allows users to securely connect to a network through the Internet and remotely access network resources.

Physical vs. Logical Topology

- A physical topology defines how the systems are physically connected.
- A logical topology defines how the systems communicate across the physical topologies.

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.

Physical Network Hardware/Devices



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 cards. 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.

Amazon VPC

Networking in Your VPC

You can use the following components to configure networking in your VPC:

- IP addresses
- Elastic network interfaces
- Route tables
- Internet gateways
- Network Address Translation (NAT)
- Dynamic Host Configuration Protocol (DHCP) options sets
- Domain Name System (DNS)
- VPC peering
- VPC endpoints
- VPC flow logs

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:

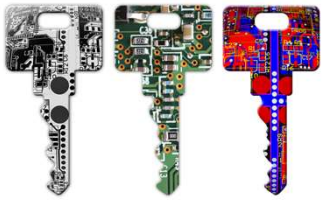
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Networking.html
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html

Appendix
Security, Identity, and Access Management

Data Center Security

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).

Network Security

- Identification & Authentication
- Firewalls
- Patching
- Virus Protection
- Encryption

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

Identification and 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).

AWS IAM
Advanced Concepts

AWS Resource-Based Policies

- Are an alternative to IAM and supported by some services.
- Grant cross-account access to your resources.
- Use a principal to uniquely identify accounts in the policy.
- Supported AWS services include :
  - Amazon S3 Bucket Policy
  - Amazon SNS Topic Policy
  - Amazon SQS Queue Policy
  - Amazon Glacier Vault Policy
  - AWS OpsWorks Stack Policy
  - AWS Lambda Function Policy

For some AWS services, you can grant cross-account access to your resources. To do this, 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:
http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_compare-resource-policies.html

Access to AWS Resources

Temporary Security Credentials

- Security Token Service
- `AssumeRole`
- `AssumeRoleWithSAML`
- `AssumeRoleWithWebIdentity`

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

AWS Services support for IAM Roles

- AWS CLI on Amazon EC2
- AWS CloudTrail logs to Amazon S3
- Amazon Elastic Transcoder access to Amazon S3
- AWS Elastic Beanstalk access to AWS services
- AWS Lambda code access to AWS services
- Many more …

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 in the slide.

Appendix
Databases

Security Groups

Allow access to IP address ranges or Amazon EC2 instances you specify.

Use VPC security groups to control access to a DB instance inside a VPC.

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

DB Parameter & Option Groups

DB parameter groups:

- Contain engine configuration values that can be applied to one or more DB instances of the same instance type.
- Are applied by Amazon RDS by default when you create DB instance, which contains defaults for the specific database engine and instance class of the DB instance.

DB option groups:

- Tools that simplify database management.
- Currently available for Oracle, Microsoft SQL Server, and MySQL 5.6 DB instances.

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

Supported Operations

Table Operations:

- Create, update, and delete tables.
- After creation, you can increase or decrease provisioned throughput.
- Retrieve the table's status, the primary key, and when the table was created.
- List all tables in your account for a region.

Item Operations:

- Add, update, and delete items from a table.
- Add, update, and delete existing attributes from an item.
- Perform conditional updates.
- Retrieve a single item or multiple items.

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.

Local Secondary Index

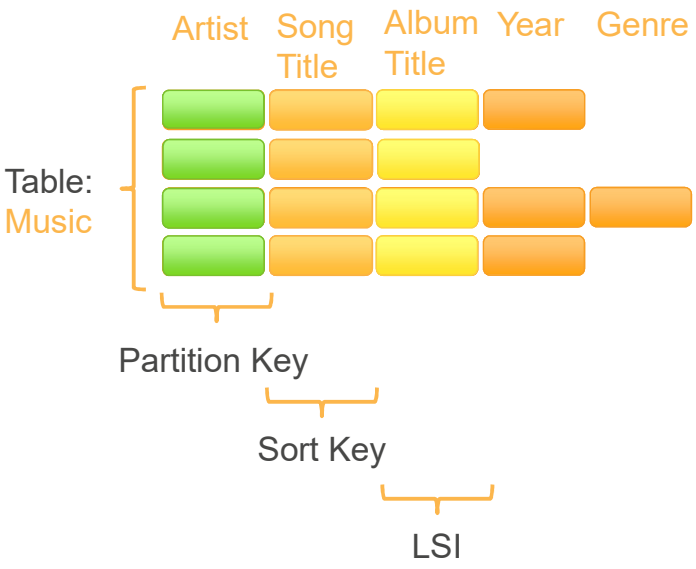
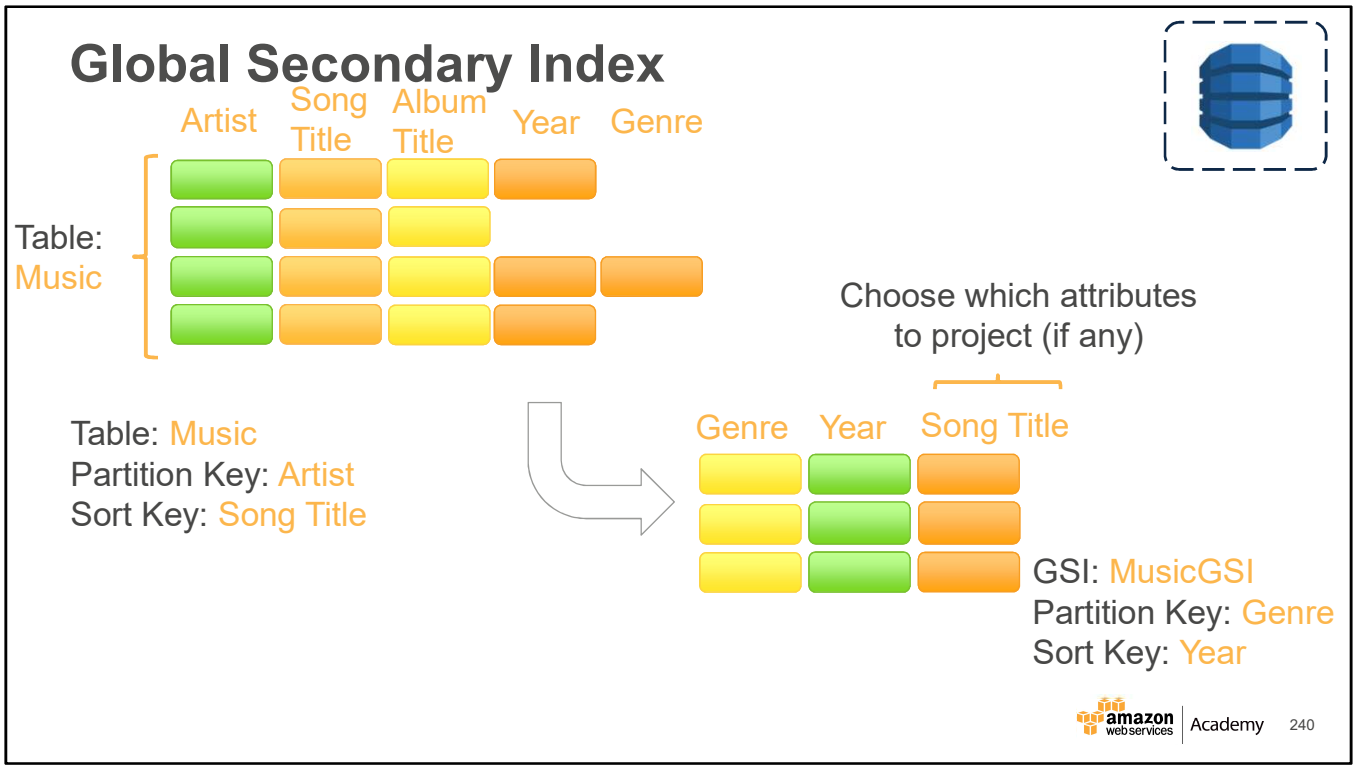


Table: Music
Partition Key: Artist
Sort Key: Song Title
LSI: Album Title

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. The global secondary index must be created at the time the table is created.

For more information, see:
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

Appendix
AWS Elasticity and Management Tools

AutoScaling
Advanced Concepts

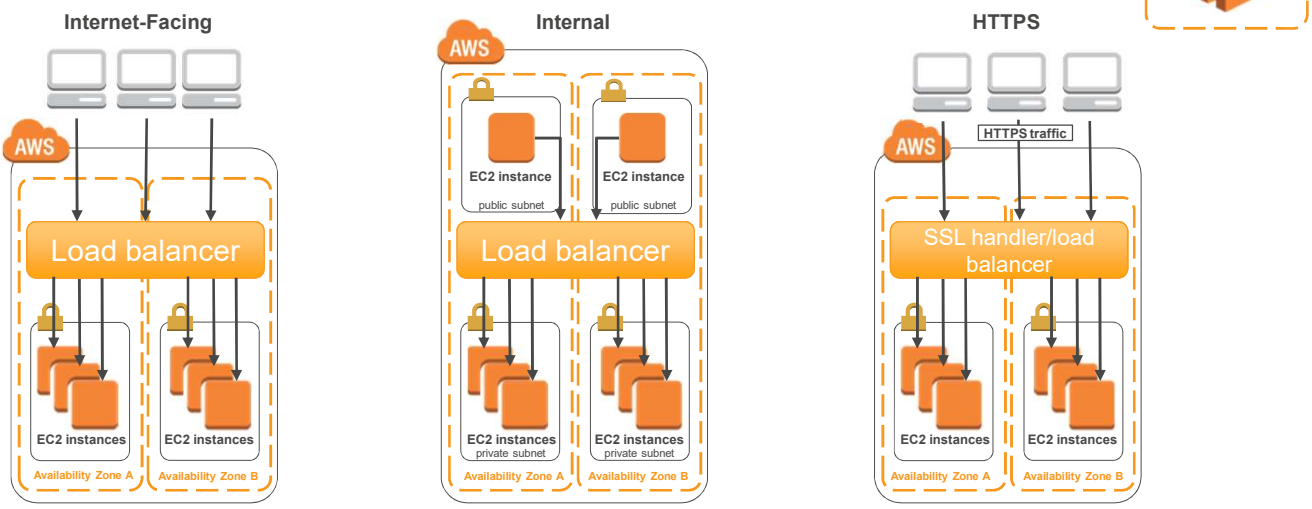
Scaling Plans

Auto Scaling Minimum	Manual Scaling	Scheduled Scaling	On Demand Scaling
Health Check monitors running instances within an Auto Scaling group.	Specify a new minimum for your Auto Scaling group.	Scaling functions are performed as a function of time and date.	You create a policy to scale your resources.
If an unhealthy instance is found, it can be replaced.	Manually invoke Auto Scaling policies.		Define when to scale using CloudWatch Alarms.

For more information, see:
http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/scaling_plan.html#scaling_typesof

Elastic Load Balancing Advanced Concepts

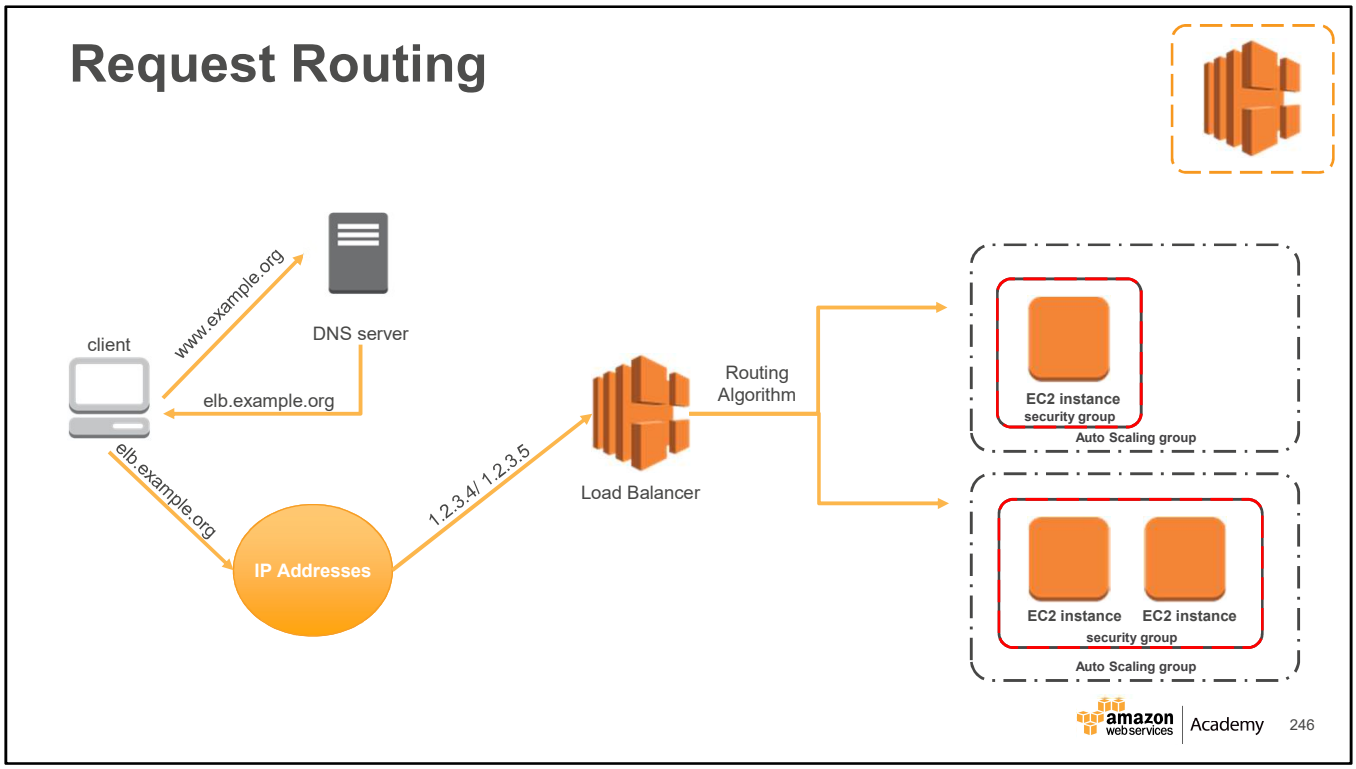
Load Balancer Types



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.



Before a client sends a request to your load balancer, the client 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.

Listeners

- A listener is a process that checks for connection requests.
- Front-end connections are:
  - Client to load balancer connections.
  - Configured with a protocol and a port.
- Back-end connections are:
  - Load balancer to back-end instance connections.
  - Configured with a protocol and a port .
- ELB supported protocols:
  - HTTP
  - HTTPS
  - TCP
  - SSL

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

Back-end Instances for Your Load Balancer

- Health checks
- Security groups
- Subnets
- Register
- De-register 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.

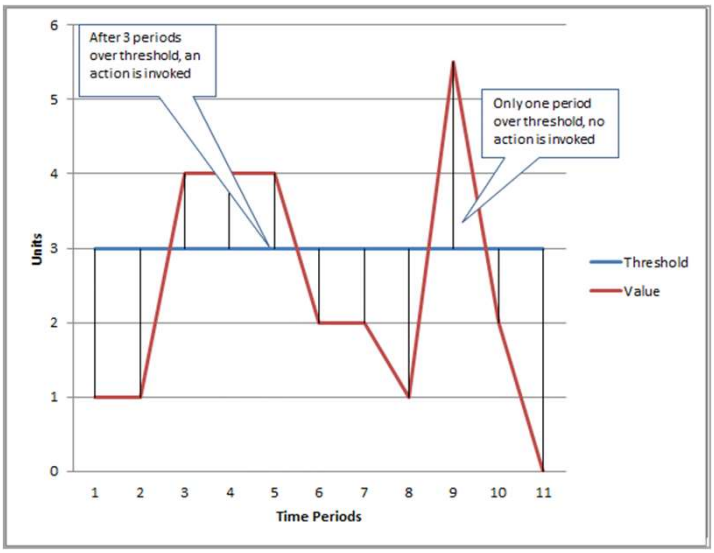
Registering an EC2 instance adds it to your load balancer. The load balancer continuously monitors the health of its registered instances and routes requests to the healthy registered instances. If demand on your instances increases, you can register additional instances with the load balancer to handle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

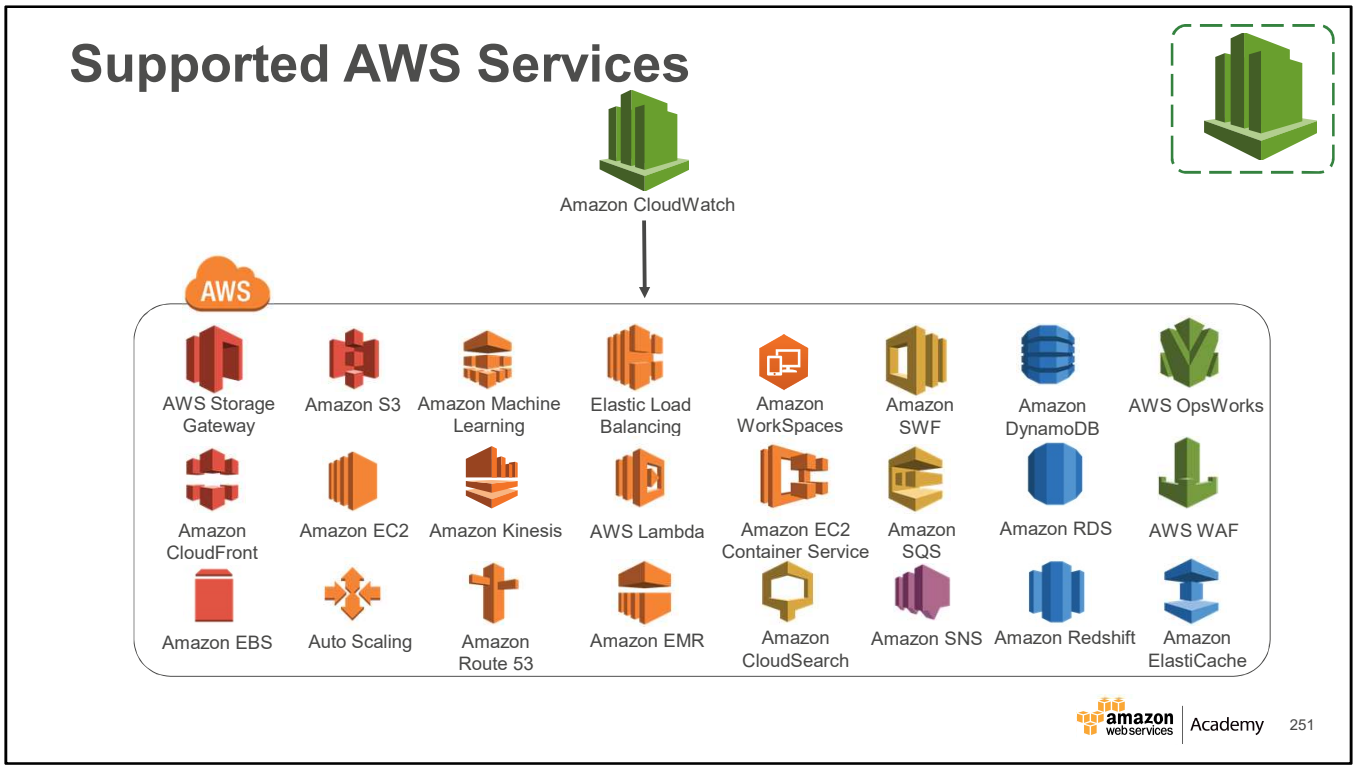
CloudWatch Advanced Concepts

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 three periods. The alarm invokes its action only when the threshold is breached for three 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

Appendix
Course Wrap-Up

AWS Support

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.

Pricing

Basic	Developer	Business	Enterprise
Included	$29/month -or- 3% of monthly AWS spend	Greater of $100 -or- 10% of monthly AWS usage for the first $0-$10K 7% of monthly AWS usage from $10K-$80K 5% of monthly AWS usage from $80K-$250K 3% of monthly AWS usage over $250K	Greater of $15,000 -or- 10% of monthly AWS usage for the first $0-$150K 7% of monthly AWS usage from $150K-$500K 5% of monthly AWS usage from $500k-$1M 3% of monthly AWS usage over $1M

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

Pricing Examples

Business Pricing Example

For $85K in AWS monthly usage:

$10,000 x 10% = $1,000
(10% of the first $0 - $10K of usage)

+ $70,000 x 7% = $4,900
(7% of usage from $10K - $80K)

+ $5,000 x 5% = $250
(5% of usage from $80K - $250K)

+ $0 x 3% = $0
(3% of usage over $250K)
Total: $6,500

Enterprise Pricing Example

For $1.2M in AWS monthly usage:

$150,000 x 10% = $15,000
(10% of the first $0 - $150K of usage)

+ $350,000 x 7% = $24,500
(7% of usage from $150K - $500K)

+ $500,000 x 5% = $25,000
(5% of usage from $500K - $1M)

+ $200,000 x 3% = $6,000
(3% of usage over $1M)
Total: $70,500

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