AWS CERTIFIED CLOUD PRACTITIONER CERTIFICATION GUIDE COMPLETE 2018 CLF-C01 EXAM STUDY GUIDE By: Todd Montgomery

Introduction

This book will introduce you to, and explain, all of the necessary information to help you to prepare and pass the AWS Certified Cloud Practitioner certification exam. The material is presented in a condensed “what you need to know” format that covers the services and topics outlined in the AWS Cloud Practitioner exam blueprint.

Throughout the book, web links are provided to AWS white papers and other informative documents that provide additional information on AWS and, more specifically, the topics covered in the Practitioner exam.

By mastering the topics covered in this guide, you will have all of the necessary technical information to pass the current version of the Cloud Practitioner exam.

Disclaimer:

The Author is an independent content developer not associated or affiliated with the vendor mentioned throughout this book. The names, titles and awards of the certification exams mentioned in this book are the trademarks of Amazon Web Services or AWS. I mention these names and/or the relevant terminologies only for describing the relevant exam objectives and knowledge. I develop study material entirely on my own without endorsement from AWS. This material is fully copywrited.

We are not affiliated with Amazon Web Services and this document is not endorsed by AWS. The AWS certification exams are the property of Amazon Web Services.

All rights reserved. The book or any portion thereof may not be reproduced or used in any mannermanner whatsoever without the express written permission of the author except for the use of brief quotations in a book review.

Although the author has have made every effort to ensure that the information in this book was accurate at press time, the author does not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

About the Author

Todd Montgomery has over 35 years of IT experience and has held a number of roles with equipment vendors, large enterprises, Systems Integrators, The Department of Defense, and as a published author of Cloud computing and data center books. He holds many industry certifications including AWS cloud practitioner, Solutions Architect Associate, Developer Associate and Sysops Administrator Associate. He also hosts the Austin, Texas AWS certification meetup group. Currently he is writing books on AWS certifications and is consulting on network automation, security and analytics for Cisco Systems. Todd lives in Austin, TX and can be contacted at toddmont@thegateway.net

Also by Todd Montgomery:

CompTIA Cloud+ Study Guide: Exam CV0-002 2nd Edition CompTIA Cloud+ Study Guide: Exam CV0-001 1st Edition CCNA Cloud Complete Study Guide: Exam 210-451 and Exam 210-455 1st Edition CCNA Data Center: Introducing Cisco Data Center Technologies Study Guide: Exam 640-916 1st Edition AWS Certified Solutions Architect Associate Certification guide

Description

This book is written to help you gain the knowledge to pass the Certified Cloud Practitioner exam from Amazon Web Services. In this course, you will learn the core cloud concepts and AWS services including security, architecture, pricing, and support.

This is not a technical course that teaches you how to configure and deploy AWS services, but rather, it will teach you what the cloud is and what some of the many services are and how they are used. This will allow you to view the cloud conceptually.

Intended Audience

The cloud practitioner certification is designed for people that are in non-technical roles but need or desire an understanding of cloud computing in general and Amazon Web Services in particular. This would include such on-IT professionals such as C-level executives, Business, Finance, Legal, Sales, Marketing, Project managers, AWS Academy students and other IT-related professionals.

Course Objectives

This book covers the objectives outlined in the AWS cloud practitioner exam blueprint:

* AWS Cloud concepts and its global infrastructure
* Key services on the AWS platform and their common uses
* Cloud architectural principles, AWS basics
* Overview of the basic security and compliance of the AWS platform and the shared security model
* Billing, account management, and pricing options
* Documentation and white papers
* AWS Cloud value compared to on-premise data centers
* Deploying and operating in the AWS Cloud

Prerequisites

It is recommend (but not required) that students of this book have the following background knowledge:

* General IT technical knowledge
* General IT business knowledge

Course Outline

This course covers the following concepts and follows the AWS certified cloud practitioner exam blueprint:

Cloud Concepts

* Introduction to the Cloud
* Introduction to the AWS Cloud

Core Services

* Overview of Services and Categories
* The AWS Global Infrastructure
* Amazon VPC
* Security Groups
* Compute Services
* Storage Services
* Database Solutions

Security

* Introduction to AWS Security
* The AWS Shared Responsibility
* Responsibility Model
* Access Control and Management
* Security Compliance Programs
* Security Resources

Architecting

* Introduction to the Well-Architected Framework
* Reference Architecture: Fault Tolerance and High Availability
* Reference Architecture: Web Hosting

Pricing and Support

* AWS Pricing
* Pricing Details
* The TCO Calculator
* Support Plans

Cloud Practitioner: additional materials

* links that enhance concepts learned in this book

Introduction

This course is designed for students that wish to gain a basic instruction and understanding of AWS cloud services.

We will give you a detailed overview of cloud concepts required to ace the AWS Practitioner exam, including, services, security, pricing, architecture and support.

When completed, you will understand:

* What the cloud is and the basic AWS cloud infrastructure
* Understand the core services on the AWS cloud and what they are used for
* Understand the AWS architecture and their principals, the security and compliance in the AWS cloud including the shared security model outlined by AWS
* Pricing, billing, account administration and the various pricing models offered on the AWS platform
* Learn about AWS documentation and how to get assistance for technical questions
* Learn the value of the public cloud and more specifically, the value of AWS public cloud services
* Deploying and operating services in the AWS cloud

Fundamentals of Cloud Computing

* This section covers some common terms and concepts in cloud computing so you can become familiar with the basic terms used throughout this book
* Cloud computing is fundamentally a data center shared by many different users
* Cloud computing uses a utility model of pricing where you only pay for what you use
* Expenses are shared across many companies and you benefit from economy of scale
* You do not have any upfront equipment, software, facilities or labor costs to operate in the public cloud. There are not capital expenditures, only operational.
* The public cloud offers many different services including computing, storage, networking, security, databases and many other offerings
* Cloud data centers are usually accessed over the internet
* Companies no longer need to build their own information technology infrastructure, they can simply “rent” services from a public cloud company
* There are no longer any long procurement and installation cycles as was common with private data centers
* With public cloud offerings, large and complex operations can be provisioned in minutes or hours
* Amazon Web Services, or AWS, is Amazons public cloud product
* AWS has cloud data centers located thought the world
* People use the cloud constantly, however they may not be aware that they are doing so

Cloud Terms and concepts

* There are three main service categories of cloud computing:

Infrastructure as a service (IaaS): Cloud provider offers basic platforms, and you manage the operating systems and applications

Platform as a Service (PaaS): The cloud provider supplies all the underlying infrastructure and the operating systems (servers) and you are responsible only for the applications you install on the servers

Software as a Service (SaaS): The cloud provider supplies the underlying compute infrastructure, servers and applications for a complete offering

* Elasticity: A term used to describe the clouds ability to dynamically add and remove services, and capacity, such as web servers, to meet the current demand on your deployment
* Compute capacity is added and subtracted to meet current demand is said to be elastic
* Scalability: The ability of the cloud to add capacity as required, this can be compute capacity that gets added by deploying additional servers or migrating to a single larger server, adding additional storage, deploying services in another part of the world, as required and on demand
* Fault tolerance: Resiliency to failures, cloud systems are designed and deployed to withstand failures and not interrupt your operations
* High availability: Systems that are designed to always be available with redundant systems to remain operational in the event of component or system failures

Cloud Benefits

* Shared online storage of files and backing up data
* Accessing mobile applications from your smart phones, laptops and tablets
* Accessing your information from multiple devices
* Leverage cloud fault tolerance in the cloud to reduce or eliminate the loss of data due to local failures of a PC, tablet or phone
* For enterprises, dynamic compute capacity on demand
* You pay only for what you use in the cloud
* No upfront costs to deploy datacenter equipment
* Ability to access advanced applications and services that were traditionally only available to very large corporations that could afford the costs of deploying these technologies
* Take advantage of the pricing cloud providers due to economy of scale
* Start small and grow as demand increases
* No equipment obsolescence
* No need to predict growth and purchase equipment to meet that load since the cloud scales as needed
* Leverage the clouds high availability, elasticity, global footprint, and advanced features

Core services

* This section will provide you with a basic high level overview of some of the most common services in the AWS cloud, this is not exhaustive as AWS offers over 100 different services, but rather the most common
* We will explore each of these in greater detail in the rest of this book
* IAM, Identity and Access management

Allows you to define what can be connected where and what they are allowed to do

Assigns security rights to users, groups and objects

How AWS users are created and managed

* Storage

Simple Storage Services (S3), Amazons primary cloud based storage offering

* Networking services
* Virtual Private Cloud (VPC)

your private compute area in the cloud

* Compute Services

EC2 AWS virtual computers

Lambda serverless computing

* Elastic Load Balancing (ELB)

balances web traffic across multiple servers

* Autoscaling

allows for dynamic and automated increasing and decreasing of AWS resources based on demand

* Content Delivery (CloudFront)

edge locations, data cached at locations

around the world for faster response time

* Domain Name Services(DNS)

Route 53 is the AWS DNS service offering

Route 53 maps domain names to IP addresses and has advanced networking features

* Monitoring and logging

CloudWatch

CloudTrail

* Alerting and Notification services

Simple Notification Services (SNS)

Text messages and email notifications based on events

* Database offerings

Data caching via ElastiCache

Data warehousing via Redshift

NoSQL service via DynamoDB

RDS, Relational Database Services

SQS, Simple Queuing Services

Aurora, MySQL and PostgreSQL-compatible relational database

Services and categories

* Services are another name for applications and other offerings in AWS
* All services are presented and managed in the web based AWS Management Console at https://aws.amazon.com
* Documentation and explanations are on the main page and you do not need to create an account or log into the service to learn about the many services offered by AWS
* AWS offers a free tier account and it is highly advised that you take advantage of this offer to explore AWS for free
* Exam Tip: Know the names and uses of the primary services covered in this book

Global Infrastructure

Three main components to the AWS global infrastructure

Regions

Availability Zones

Cloudfront Edge locations

* Geographical regions host two or more availability zones
* Regions are located around the world and are groups of AWS data centers
* As of 2020 there are 24 regions with more being added
* Regions are isolated from each other, a failure or service disruption in one region will not affect other regions
* Deploying critical applications in multiple regions ensures that your data is globally available at all times
* Data can be easily replicated in a region for high availability and durability.
* Availability Zones
* Data centers inside of an AWS region
* A region is comprised of two or more availability zones
* Each availability zone consists of two or more data centers
* Each availability zone is physically and logically distinct and isolated data center separated from each other and connected with high speed, low latency redundant network connections
* AZ isolation is a physically separation, 10 to 100s of miles apart but are still in the same geographic region of the earth
* AZ isolation protects them from a failure in one availability zone from effecting other
* AZs It is a best practice to provision your deployment in multiple availability zones for fault tolerance and high availability
* Each AZ has its own separate power systems, cooling, backup generators, and networking connections
* Regions enable redundant and highly available cloud operations, if one AZ were to fail, the other AZs in the region can take the workload and keep your site operational
* 77 Availability Zones with more being added on a regular basis Edge Locations

149 edge locations (and counting)

11 regional edge caches

65 cities across 29 countries

* The AWS Content Delivery Network (CDN) called CloudFront
* CloudFront AWS content close to customers with multiple locations spread throughout the world
* Requests for AWS content are routing to the nearest CloudFront edge location where a copy of the data is stored (known as caching) for low latency responses
* Exam Tip: Know the Regions are geographical areas of data centers and that multiple availability zones make up a region. Edge locations are independent of Regions and Availability Zones
* https://aws.amazon.com/about-aws/global-infrastructure/
* <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>
* https://aws.amazon.com/cloudfront/details

Virtual Private Cloud: VPC

* Virtual Private Clouds are a core AWS service
* A private network inside of the Amazon public cloud
* Acts as a private data center
* VPC hides most of the complexity of setting up your own network
* The VPC is logically isolated from the rest of the AWS cloud
* A VPC is your private section of AWS where you control the resources that you place inside
* The VPC owner has complete control of the VPC and places AWS resources into the VPC
* The VPC owner controls who has access to the resources inside of the VPC
* You control your configurations and decide what to isolate or expose outside of the VPC
* Many layers of security controls allows you to define what traffic is allowed into and out of your VPC
* You can deploy AWS services inside your VPC to build your own network and applications such as compute, databases and storage
* When you create an AWS account, a default VPC in each region is created for you and you are allowed to create additional VPCs with a limit of 5 VPCs per region
* Exam tip: A Virtual Private Cloud is a carved out portion of the AWS cloud for your exclusive use and is not shared with other organizations unless configured to do so
* https://aws.amazon.com/vpc/

VPC Features

* VPC’s reside inside of AWS regions and availability zones
* You can have many VPC’s per AWS account, including up to 5 VPCs per region
* Network subnets are created inside of each VPC, public subnets have internet access and private subnets do not access the internet (unless configured to do so)
* VPC’s can span multiple availability zones inside of a region
* VPC’s cannot span regions, they are region specific
* Route tables inside of a VPC control the traffic flow between subnets
* Internet gateways, (IGW) connects a VPC to the internet
* NAT (Network Address Translation) gateways enables private subnets to access the internet
* Access control lists restrict and allow the flow to/from each subnet Security groups restrict and allow the flow of traffic in and out of individual EC2 compute instances
* VPC Configuration example:

Create the VPC

Define a IP address block

Define subnets in availability zones

Define an internet gateway

Define security groups and network access control lists

Create compute, storage and other services inside of the VPC

Security Groups

* Controls access to resources and virtual servers in a VPC
* Acts as a built in virtual firewall
* Filters traffic to your instances, you define what you want to allow access from
* Configure security groups rules to define what traffic is allowed in and out of your instance
* Blocks malicious, unwanted and unauthorized traffic from accessing your virtual server
* For example, you can only allow web traffic from the internet to connect to your EC2 web server
* You can use multiple security groups in a VPC
* AWS console management, under EC2 instances and network and security tab on the left
* By default, all outbound traffic is allowed and all inbound traffic is denied
* Security groups are stateful, this means they keep track of the state of each flow and if the security group allowed traffic out, it will automatically allow return traffic back into the server instance
* Create rules to modify the default behavior of a security group
* Exam Tip: Security groups act as firewalls for virtual servers inside of a VPC
* https://docs.aws.amazon.com/vpc/latest/userguide/VPC\_SecurityGroups.html

Compute Services

* Servers in the cloud
* Large selection of instance types from large to small and instances that are designed for specific use cases
* AWS offers a huge selection of virtual servers and operating systems
* Scales to meet your workload
* Add and remove compute services as needed (elastic/scalability)
* EC2, Elastic Compute Cloud are AWS hosted virtual servers
* Lambda runs code without servers, you pay for the compute time only
* LightSail offers a complete server deployment solution based on your requirements
* Elastic Container Services (ECS) is a container service that supports Docker compute containers and is fully managed by AWS
* Exam Tip: Know that the AWS compute servers include EC2 and Lambda serverless offerings
* https://aws.amazon.com/products/compute/
* https://aws.amazon.com/lambda/serverless/
* https://docs.aws.amazon.com/ecs/

Elastic Compute Cloud: EC2

* Elastic Compute Cloud (EC2) is the server based compute resources in the AWS cloud
* Servers in the AWS cloud
* Virtualized servers are software representations of real servers
* Many different server types to select from to meet your requirements
* EC2 individual processes are called instances in the console
* EC2 is like an advanced home computer that you use, only it is in the cloud
* Provides scalable compute capacity in AWS
* Eliminates the upfront investment in servers
* You are only charged for the computing resources you use
* Deploys in seconds or minutes
* Examples:

Application server

Database server

E-Mail server

File Server

Media Server

Web Server

..many more

* EC2 includes the following components:

Operating systems (Linux or Windows) called

Amazon Machine Images (AMIs)

CPUs

RAM memory

Hard drives for storage (usually Elastic Block Storage EBS)

Network cards for network connectivity

Firewall for security (security groups)

* Servers in the cloud that can be scaled up and down based on current demands
* Servers and EC2 are the same, EC2 is the AWS marketing name, also commonly referred to as instances by AWS
* Pay as you go, you are only charged for the time that the instances are active
* AWS offers a very large selection of EC2 instances to meet any requirements
* You select what region and Availability Zone to run your instances in for global hosting
* Huge selection of hardware and software available on AWS
* AMI is the Amazon Machine Image with is the operating system you choose to run on your EC2 instance, some are free and others are chargeable
* AMIs provide the information needed to launch and instance
* AMIs are the similar to the “Gold Images” used in enterprise computing
* There is a large selections of AMI’s in both Windows and Linux variants
* AMI’s are the software (operating system)and machine instance types are the hardware (server)
* The Instance type are the hardware that is virtualized and runs in the AWS cloud, there are over 70 different instance types offered by AWS, families include:

Accelerated computing

Bare Metal

Compute optimized

General purpose

GPU optimized

I/O optimized Memory optimized

Programmable/FPGA (Field Programmable Gate Arrays)

Storage optimized

More are being added regularly

* Elasticity refers to the ability to add and remove instances in near real time based on workloads
* When configuring a EC2 instance you define the VPC, subnet and availability zone that it will run in
* EC2’s are placed in availably zones in a region that you select
* You configure networking, security and storage for each instance
* A security key pair is created and used to securely connect to the instance remotely
* Use the AWS web console to create, manage and monitor your EC2 instance
* Instances are only billed when they are running and to the nearest second of operation
* Instance purchasing options:
* On-Demand is the most expensive and flexible method to purchase an instance, you pay for the time it is running and it can be created and terminated at any time
* Reserved instances, purchased for a set time period of either one or three years, pay upfront or over time, receive a big discount over on-demand pricing (up to 75% less), you own it for the time period you selected if you use it or not, use reserved instances for your baseline compute requirements, cost effective for EC2 utilizations in the 90-100% range and expected to remain at that load for a full year
* Spot instances, you bid for instances in an online auction environment and can use the instance as long as you are the winning bidder, you use excess AWS compute capacity, charged for the time you use it, AWS can terminate it at any time if there is a higher bid, used for workloads that can support variable availability of EC2 instances that can support interruptions
* EC2 instances are secured by Security Groups and Network Access Control Lists
* Exam Tip: EC2 allows the customer full administrative privileges to operating system and the ability to define CPU, Memory, and Networking options
* <https://aws.amazon.com/ec2>

Lambda

* Lambda is a AWS managed serverless architecture
* Serverless means you do not have to deploy or manage any servers, AWS does it all for you
* Lambda instances are launched by pre-defined events
* Upload your code and AWS handles the server based operations of setting up, configuring and monitoring the compute services
* Lambda is integrated with many AWS services
* A service that allows you to run code inside of AWS without managing any service
* Pricing is granular to the sub-second level
* There is no need to provision any instances or manage any servers with the Lambda services, AWS handles all of the server management, you just provide the code to run
* Lambda only runs code when needed and scales automatically
* Since Lambda is on demand, you save money since you are only charged for the time your code is executing
* Scales to thousands of requests per second
* You pay for the compute services you use, if your application is not running you do not pay for lambda
* You are charged to the millisecond of compute time
* There are no servers to manage
* Fast loading and scaling
* On-demand computing and a very low price point
* Good for workloads that are infrequently needed, intermittent or variable
* AWS provides all server and operating system support, you provide the application code you wish to run
* NodeJS, C#, Java, Python are supported languages with more being added
* Run Lambda jobs in response to events such as a file being placed into storage or an alarm on a system
* Events are defined that launch code that runs on Lambda to perform a function you wish to occur
* When the job is done, lambda shuts down and goes away
* Lambda supports serverless and micro-service applications
* Lambda instance constraints:
* Disk space is limited to 512 MegaBytes
* Execution time is a maximum of 5 minutes
* Event request body is limited to 128 KiloBytes
* Memory from 128 to 1536 MegaBytes
* Response body payload is a maximum of 6 MegaBytes
* Billing is based on the number of times Lambda is triggers and per 100 milliseconds of execution time
* Configure Lambda in AWS and upload your code to run on it and configure trigger events
* In the AWS console under compute
* Exam Tip: Know that Lambda is serverless
* https://aws.amazon.com/lambda/

Elastic Beanstalk

* AWS service that allows a developer to quickly deploy applications in the cloud
* Pre-configured templates of common cloud deployments allows for a quick deployment of your application in the Amazon cloud Allows users to get up and running with limited knowledge of AWS administration and deployments
* Platform as a Service PaaS creates the whole infrastructure for you
* The underlying complexity of the system is managed by AWS
* You control what instance type, database and scaling you want and Elastic Beanstalk takes care of the deployment for you
* Provides the Host, Language interpreter, Operating System, Web server, Application server and more, you supply the application code
* Additional feature include AWS managing all updates, provides log files, creates web servers and load balancers
* Platforms supported include:
* Containers, Docker and Kubernetes
* Go
* Java Server Edition
* Java with Tomcat web server
* .NET for Microsoft IIS web servers on Windows
* Node.js
* Packet builder
* PHP
* Python
* Ruby
* You create your code and use Elastic Beanstalk to deploy it
* Exam Tip: Know that Elastic Beanstalk is Pre-configured templates of common cloud deployments allows for a quick deployment of your application in AWS
* https://aws.amazon.com/elasticbeanstalk/

Elastic Load Balancer

* Load balancers evenly distribute incoming traffic across multiple EC2 instances that are associated with the load balancer process
* Provides access to your services such as web servers using a single entry point that connects to multiple back end servers
* Load balancers can be configuring for public internet facing or for the internal load balancing of private servers
* Allows for application decoupling between layers of services
* Enable high availability and fault tolerance
* Enables elasticity and scalability
* Load balancers can load balance across multiple availability zones
* By distributing connections to servers in more than one availability zone, high availability can be achieved
* Detects unhealthy EC2 servers connected to them and stops forwarding traffic to them until they pass health checks
* Balances traffic to back end servers using metrics such as round robin, least connections and response times
* Load balancers are commonly used to distribute the workload across two or more internet facing web servers
* For example, 500 web users accessing your AWS web site that has a load balancer and 2 EC2 web servers will balance across them with 250 users connected to one web server and the other 250 users connected to the other web server using a round robin algorithm
* Load balancing increases the fault tolerance of your applications
* The load balancer health checks the servers to ensure they are active before sending connections to them
* If they are unhealthy, the server is taken out of rotation
* Sticky connections nails a user’s connection to a single server for the duration of the connection Use DNS (Domain Name System) to point your domain to the load balancers public address, the load balancer then connects to multiple servers behind it
* Monitoring in AWS allows you to see HTTP responses, the number of healthy and unhealthy connection and availability zone usage
* Load balancers can be integrated with other AWS services that allow additional servers to be added an subtracted for scalability to meet current workloads, this is referred to as auto scaling
* Load balancing is configured in the EC2 section of the AWS console
* Exam Tip: Load balancers evenly distribute incoming traffic across multiple EC2 servers that are associated with the load balancer process
* https://aws.amazon.com/elasticloadbalancing/

Application Load Balancer

* Takes incoming traffic and spreads the connections between multiple servers
* Application Load Balancers include routing and visibility features targeted at the application level
* Works with EC2 virtual servers, microservices and containers.
* Support for container load balancing, IPV6, Web Application Firewall, Health checks, Dynamic IP ports, deletion protection, request traces
* Listener is the public IP that accepts incoming traffic
* Targets are servers hosting a particular application, can be EC2 or a micro-service
* Target groups are groups of servers that are load balanced and define the health checking Support HTTP, HTTPS2, HTTPS, WebSockets
* Supports CloudWatch Metrics
* Generates log files of connection details
* Health checks validates that the servers are active and able to accept connections
* Load balancers are configured in the EC2 console
* Exam Tip: Know that Application load balancers operate at the application level and distribute traffic across multiple servers
* https://aws.amazon.com/elasticloadbalancing/

Auto Scaling

* Using automation in the AWS Cloud to dynamically add and remove services based on current workload
* Automates the process of scaling up and down (adding and removing) EC2 compute instances to meet current demands
* Auto scaling ensures that your applications are fault tolerant by making sure there is adequate compute capacity and replacing failed EC2 instances
* Auto Scaling enables scalability and elasticity
* If the EC2 instanced reach a certain threshold where they cannot handle the traffic load, Auto Scaling will take action and add additional compute instances, it will then monitor the workload and remove the instances when the load decreases
* Enables the current number of EC2 servers to be available to match your applications workload As the workload increases, auto scaling adds EC2 servers and removes them when the demand decreases
* You monitor the performance and load on your deployment using the AWS CloudWatch service, CloudWatch then notifies the autoscaling service to add or remove servers based on your workload at the current time
* Autoscaling matches compute capacity to your current workload
* This allows for paying only for what you need
* You define the conditions to add and remove resources
* EC2 provisioning is automated using autoscaling
* Scaling out is adding services and scaling in removes instances
* To configure autoscaling:
* Create a launch configuration (what) consisting of AMI’s, Instance types, security groups and roles
* Create an autoscaling group (where) VPC and subnets, load balancers, minimum and maximum number of instances, and the desired capacity
* Define an autoscaling policies (when) scheduled such as the day and time each week, on-demand based on conditions such as CPU load on servers, scale-out and scale-in policies when to add and remove servers
* An example would be monitor of each instances CPU, if the load goes above 90% for 5 minutes, then add two more serves and when the CPU load then drops below 50% for 5 minutes, remove servers one at a time down to the desired instance count parameter you defined
* Auto scaling makes sure you have the processing power to handle the load of your applications
* Groups are created that are collections of EC2 instances
* Groups make sure you never go below a defined minimum of instances, you can also define the maximum number of instances to define and minimum and maximum capacity
* Scaling policies define the levels to trigger an auto scaling event
* Scaling uses the CloudWatch service that can monitor, for example, the CPU load on a EC2 instance, if the CPU load goes above a defined threshold of, say, 80% for a period of time, then add more instances .
* Then, monitor the CPU load and if it drops below a threshold for a period of time, then remove the EC2 instances
* Elastic load balancers are also aware of servers being added and removed by Auto Scaling and configure themselves appropriately
* Scalability is also referred to as elasticity
* Auto scaling works without human intervention after it is configured
* Exam Tip: Know that as the workload increases, Auto Scaling adds EC2 servers and removes them when the demand decreases
* https://aws.amazon.com/autoscaling/

Simple Storage Services: S3

* Simple Storage Services is Amazons primary storage service
* S3 is a fully managed AWS storage service Object storage in the AWS cloud (Objects is the term used for files)
* Online mass storage in the cloud that is accessible over the internet by any device
* Cloud storage, S3 is the primary AWS storage service
* AWS manages S3 so there is no customer overhead managing storage
* S3 storage units are called buckets
* Buckets are region specific
* Any type of file or data can be stored in S3
* Objects are downloaded using a URL that is assigned to each object
* Highly scalable, reliable and fast online storage
* S3 stores a virtually unlimited number of objects, trillions of objects can be stored and supports millions of puts per second
* You can access S3 storage from anywhere
* Extensive security controls and management options
* Due to the massive scale of S3, Amazon has low costs for the service and passes the savings onto their customers
* S3 is not associated with any server, it is a standalone storage service
* Objects stored can be any type of data such as images, videos, data, pictures, music, web pages and logs
* Each object can be up to 5 Terabytes in size
* There is no limit to the amount of data that can be stored in S3, it is a Petabyte scale service with an almost unlimited amount of capacity
* Supports internet access with a browser using HTTP or HTTPS
* Security policies are provided by bucket policies, Identity and Access Management and Access Control Lists
* S3 data can be encrypted at rest and in transit
* Data objects are automatically replicated in two or more facilities in a region to protect against data loss
* S3 automatically scales to meet your storage requirements, it is elastic
* Amazon only bills for the storage you use and the class of service you select
* S3 has highly granular and detailed security services to protect your data down to the object level
* Integrates with many other AWS services
* Supports mass storage for Big Data applications
* AWS makes it easy to get data into and out of S3
* Extensive management options
* Supports transfer acceleration that enable fast and secure communications over long distances from the client to the S3 bucket
* Transfer Acceleration uses the CloudFront globally distributed edge data centers, data is routed to S3 over optimized internal data connections
* S3 is accessed through the AWS management console, the command line interface (CLI), REST endpoints with HTTP/HTTPS or the AWS provided software development kit (SDK)
* Bucket names resemble the DNS (Domain Name System) format and must be compliant with the DNS naming convention and globally unique in AWS, you cannot use a bucket name that any other user worldwide is using
* Bucket names can only contain lower case characters, numbers and hyphens, a maximum of 32 characters and no IP address formats
* S3 is widely used by the many services available in AWS and is considered a core component to AWS
* Common S3 use cases are:
* Application data
* Backups
* Big Data staging and long term storage
* Client data access
* Cross region replication of data
* Data transfer acceleration of large files of long distances
* Disaster recovery
* Large volume data import and export into and out of AWS
* Media files such as videos and images
* Web hosting (static)
* An almost limitless list of any data that needs to be stored in the Amazon cloud
* In the AWS console, all configuration and management is in the S3 section under storage
* Pricing for S3 is detailed here: https://aws.amazon.com/s3/pricing/
* Exam Tip: Know the S3 is an object based storage system and uses buckets to store data
* https://aws.amazon.com/s3/

S3 Storage Classes

* S3 has different levels of storage for different storage requirements
* A storage class is assigned to each S3 object
* Current S3 storage classes include:
* S3 Standard
* Reduced Redundancy Storage (RRS)
* Infrequent Access (IA)
* Glacier (for archival of data)
* Attributes in each storage class include:
* Availability of Objects (data)
* Cost
* How often the object is accessed
* Object Durability
* Standard is the default S3 class
* Each object must belong to one of the S3 storage classes
* Objects can be moved from one class to another either manually or by lifecycle policies that move them based on time
* Durability is the percent of time over one year that a file in S3 will not be lost, For example, the eleven-nines rating of 99.99999999% means that there is a 0.0000000001% chance a file will be lost in S3
* Another way of calculating this is to say that if you have 10,000 files stored in S3 standard, you may lose one file every 10 million years!
* Availability of Objects is the percentage over one year that an object (or file) stored in S3 will be accessible
* For the S3 standard availability of 99.99% there is 0.01% chance you will not be able to access it in S3 over a one year timeframe or one hour for every 10,000 hours
* Each object you store in S3 is backed up in multiple data-centers in a region
* Storage classes can be selected in the upload process, by time (or more commonly called lifecycle polices), or programmatically
* Objects can be made available to the public with Internet URL links
* Objects can be versioned with backup copies of all older versions retained when this service is enabled
* Exam Tip: Know the different storage classes of S3 Standard, Reduced Redundancy Storage (RRS), Infrequent Access (IA) and Glacier (for archival of data)
* https://aws.amazon.com/s3/storage-classes/

S3 Standard

* AWS managed service for general purpose cloud data storage
* Designed for files accessed on a daily basis
* S3 standard is the default S3 storage class
* Durability is eleven-nines or 99.999999999%
* Availability is 99.99% yearly
* The most expensive of the S3 storage offerings
* Very low latency and high throughput
* Data is stored in multiple data centers on multiple storage arrays
* Encryption for data in transit and data at rest
* Exam Tip: Know that S3 standard is for general purpose storage and has 11 nines of durability and 99.99% availability
* https://aws.amazon.com/s3/storage-classes/

S3 Reduced Redundancy Storage: RRS

* S3 Reduced Redundancy is used for data that is not critical and is reproducible
* RRS costs less than S3 standard
* Durability is less than standard at 99.99%%
* Availability is 99.99% yearly
* Very low latency and high throughput
* Not stored in multiple locations (less redundancy)
* Encryption for data in transit and data at rest is supported
* Good for reproducible or non-critical object storage
* Exam tip: Know that reduced redundancy is used for replaceable user data stored in S3
* https://aws.amazon.com/s3/reduced-redundancy/

S3 Infrequent access

* For data that is not accessed very frequently, you can save money by choosing this class
* Durability is the same as Standard at eleven-nines 99.999999999% yearly
* Availability is the same as Standard at 99.9% yearly, lower than standard and RRS classes
* Less cost than standard S3 access
* Data is stored in multiple data centers on multiple storage arrays
* Supports lifecycle policies to move data into IA from Standard and then on to Glacier for archival
* Exam tip: Used for data that is accessed less frequently than standard storage but needs to be highly available, less cost due to very limited reads and write operations
* https://aws.amazon.com/s3/storage-classes/

Glacier

* AWS managed service for data archival
* Glacier is extremely low cost long term storage
* Each object or file can be up to 40 Terabytes in size
* There is no limit to the volume of data that can be stored in Glacier
* For the long term storage of data, for example, for regulatory reasons
* Part of AWS storage services
* Very low cost storage of cold data that is infrequently accessed
* 99.999999999% durable
* Data is stored in multiple data centers on multiple storage arrays
* Archive is any storage object with its own unique identifier and an optional description
* Vaults are archive containers, you specify a name and the region to create the vault in
* Vaults can be locked so they cannot be altered
* Access is controlled by policies called vault policies, this determines rights for user access
* Glacier is managed by:
  + The AWS console
  + RESTful Web services
  + Java or .NET software development kits
  + S3 lifecycle policies
* Migrate data into Glacier using:
  + AWS Glacier API (Application Programmable Interface)
  + AWS Glacier SDK (Software Development Kit)
  + S3 Lifecycle policies
  + UNSUPPORTED from the AWS console
* S3 lifecycle policies allow data to be transferred automatically from S3 to Glacier based on metric such as the amount of time the data has been in S3, versions or date ranges
* Glacier retrieval is not immediate:
  + Bulk takes 5-12 hours
  + Standard takes 3-5 hours
  + Expedited takes 1-5 minutes
* Retrieval options are priced on the retrieval times where bulk is the cheapest, then standard and expedited costs the most
* Retrieval requests and data transfer costs are higher than S3
* All data stored in Glacier is encrypted be default using AES-256 type encryption
* Glacier manages all encryption keys
* Access is controlled using AWS Identity and Access Management (IAM)
* Managed in the storage area of the AWS console
* Select the region where you want to store your data in a vault
* Exam Tip: Glacier is used for long term archival of data
* https://aws.amazon.com/glacier/

Elastic Block Store: EBS

* Storage in the AWS cloud that uses standard magnetic or solid state storage drive options
* EBS is a fully managed AWS block file storage service
* EBS storage volumes are attached to EC2 instances much like a local hard drive in your laptop
* EBS provides persistent storage for EC2 instances, persistent means that even if the instance is stopped or terminated, the storage remains and there is no loss of data
* The EBS data is copied in the same Availability Zone for durability and availability
* EBS volumes can be encrypted at no additional cost
* Volumes can be increased or decreased in size and changed to different types. They are elastic
* EBS can be converted from magnetic to solid state and back, and the storage size can be increased between 1 G to 16TB to meet your changing storage needs
* Volume resizing can be performed live without having to stop the instances
* Choose the EBS volume type for your storage, performance and cost requirements
* EBS supports point in time snapshots for backup, sharing and replication
* EBS creation and configuration is performed in the EC2 section of the AWS console on the left panel Elastic Block Store section
* Volumes must be created in the same availability zone as the EC2 instance it is attached to
* After the volume is created, it gets attached to an EC2 instance as local storage
* Pricing is based on the volume size in gigabytes
* Exam Tip: Elastic block store, or, EBS is a fully managed AWS block file storage service
* https://aws.amazon.com/ebs/  
    
  Snowball
* Snowball is a petabyte-scale data transport hardware appliance that uses secure enclosures shipped to you from Amazon to transfer large amounts of data into AWS
* Using Snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns
* Transferring data with snowball is simple, fast, secure, and can be as little as one-fifth the cost of high-speed internet
* Upload data to AWS bypassing the internet or direct connections, useful in transferring large amount of data into AWS
* Rent snowball, you cannot own one, it is a loaner from AWS used to load up your data and ship it to them, they will then transfer the data into S3
* Exam Tip: Snowball is a hardware appliance shipped to you to load Terabytes of data onto and then you ship it back to AWS which then loads it into their storage systems
* https://aws.amazon.com/snowball/  
    
  Relational Database Services: RDS
* Amazons managed database offerings that installs, manages and scales common relational database products
* RDS is a family of relational databases on the market that you can choose from
* RDS is very cost competitive to hosting your own database
* RDS is a AWS service that makes it easy to set up, operate and scale cloud based databases
* RDS is very cost effective and operates on a pay-as-you-go model
* Relational databases, sometimes called SQL “SeeQuell”
* RDS is a group of SQL, or, relational databases and not a single product
* Capacity it resizable
* AWS takes care of all of the database management and underlying systems that support the RDS service including patching and backups
* Features high performance, high availability, security and compatibility
* Features resizable capacity that scales to meet your needs at any point in time
* Security and compatibility of databases hosted in the AWS cloud
* AWS RDS offers the following managed services:
* Backup and restore of databases
* Compatibility
* High database availability
* Hardware provisioning
* Installation of all software and maintenance patches
* Security of data
* Server maintenance
* Operating system and underlying server hardware maintenance
* Resizing of capacity based on workload
* Database instance class is the CPU, Memory and Networking components
* Database instance storage includes magnetic, General purpose solid state drives (SSD), Provisioned IOPS (High input/output)
* Database engines include:
* Amazon Aurora
* MariaDB
* Microsoft SQL server (multiple options)
* MySQL
* Oracle (multiple options)
* PostgreSQL
* RDS runs inside your VPC
* RDS supports high availability by configuring a multi Available Zone deployment
* Multi-AZ replicates your database to another Availability Zone and manages all failover operations automatically with no data loss due to synchronous data replication
* Multiple read replicas spread the load to multiple database copies that are in read only mode to greatly enhance performance and offload reads from the master
* Read replicas are asynchronously updated
* Read replicas can be promoted to a master
* Read replicas can be placed in a different region to reduce global access times and enhance disaster recovery
* RDS is used in web, mobile, e-commerce and gaming to name a few uses
* RDS features:
* Ability to use highly scalable storage and compute capacity
* Administration is easy and straightforward using the AWS console, API’s and CLI
* Can scale automatically using automation tools provided by AWS
* Elastic with the ability to grow and shrink capacity of your database
* High availability/disaster recovery, RDS is durable
* Low-cost including no licensing limitations
* Monitoring and logging included
* RDS is a fully managed AWS offering
* Security is included and is compliant with most regulatory attestation
* Very high throughput, RDS is a high performance database offering
* Under the Databases section in the console
* Database Migration Service, or DMS, is a AWS database migration serves used to import data into RDS quickly and securely
* Exam tip: Memorize the databases in the RDS family and that Aurora is the Amazon offering
* https://aws.amazon.com/rds/

Aurora Part of the RDS (Relational Data Base) services

* Amazons MySQL Database managed service
* Aurora is an exclusive Amazon product offering
* AWS manages all of the resources for Aurora including underlying hardware, storage, high availability, database backups, scaling, patching of OS and database and optimization
* Tools include database migrations and schema conversions to import data into Aurora
* Fully featured, Aurora as a high end commercial database
* Fully automated with management, monitoring and scaling included
* Very high performance
* Highly available
* You only pay for what you use
* Over 5 times the performance and throughput of a standard MySQL database product
* Stores six copies of the data in three availability zones
* Backups are continuous to S3 storage
* Supports up to 15 read replicas for extremely high read performance
* Aurora is designed for crash recovery usually under one minute
* Buffer cache is available at initialization to allow for high performance when the database starts, there is no need to slowly populate the cache
* In the AWS console under the RDS section of the database group
* Aurora has the ability to expand the storage size automatically
* Supports operations in multiple availability zones
* Supports security and encrypts the data
* Backup retention periods from 1 day to 35 days
* Exam Tip: Know that Aurora is the name of the AWS designed SQL database
* https://aws.amazon.com/rds/aurora/

DynamoDB

* Amazons managed NoSQL database offering that installs, manages and scales its NoSQL service called DynamoDB
* DynamoDB is an exclusive Amazon product offering
* AWS manages all of the resources for DynamoDB and automatically stores your data in multiple facilities in a región
* Fault tolerance is built into the DynamoDB service
* Very cost competitive compared to hosting your own NoSQL database
* Features resizable capacity that scales to meet your needs at any point in time
* Securely hosted in the AWS cloud
* Unlimited storage, store as many items as needed
* Users create tables and items, DynamoDB automatically creates partitions
* Items stored in DynamoDB can have different attributes, this is different than the requirements of a standard SQL database
* You can add attributes later, after you have created the database tables
* Storage scales with your needs
* All data is stored on solid state drives
* Very low latency reads
* Highly scalable throughput for both read and write operations, both manual and autoscaling options are available
* DynamoDB is used in many applications including
* Online gaming
* IoT (Internet of Things)
* Common web applications
* Mobile phone applications
* Storage of application data
* Good for high throughput and applications that are sensitive to delay/latency
* Performance of throughput and latency is consistent and predictable as the database grows
* Tables are partitioned and indexed using primary keys
* Tables can be queried using the keys or scanned over the complete table
* Queries and much more efficient and preferable over scans
* All objects in a table must have either a single of compound key
* Single keys use the partition key and the attributes
* Compound keys use a partition and a sort key and then attributes
* Created in the management console under the Database section
* Single digit millisecond latency (very fast)
* Automatically scaling of throughput capacity
* Used when data is unstructured and can change
* Fully managed service with no provisioning of underlying services
* Document and key value store models
* DynamoDB can be used in place of:
* MongoDB
* CassandraDB
* Oracle NoSQL
* Exam tip: Know that DynamoDB is a NoSQL database
* <https://aws.amazon.com/dynamodb/>

ElastiCache

* ElastiCache caches (stores locally) frequently accessed data that speeds up and increases the performance of, web applications running on AWS
* An in-memory data store or cache in the AWS cloud
* ElastiCache improves application performance by retrieving information from a fast, managed, secure in-memory data storage service bypassing the need to access data from slower disk based applications such as databases
* ElastiCache comes in two offerings:
* Redis: a high performance open source, in-memory data storage and cache
* MemcacheD: a memory object caching system that is a widely deployed application
* Caches, or stores in RAM memory frequently accessed data so it can be served very quickly
* Reduces latency for retrievals (for example, from a webserver)
* Exam Tip: know that Redis and MemcacheD are the two in-memory engines that ElastiCache offers
* https://aws/amazon.com/elasticache/

Redshift

* Amazons big data offering that installs, manages and scales a fully managed data warehouse
* RedShift is a fully managed AWS data warehouse service
* Simple and cost effective services that is used to analyze all your data with standard SQL and business intelligence applications
* Adds data analytics capabilities to applications
* Redshift is an exclusive Amazon product offering
* AWS manages all of the resources for Redshift
* Fully automated with management, monitoring and scaling included
* Analyzes data using SQL and business intelligence tools
* Runs complex queries against large data stores that can be in the Petabytes and Exabytes.
* Stores data on high performance disk drives that are local for increased performance
* Massively parallel queries are supported
* Uses columnar storage
* Data is compressed by Redshift
* Security is management by AWS with strong encryption of the data and rest and in flight
* Consists of a leader node and multiple compute nodes
* You only pay for what you use
* Significantly lower cost than traditional data warehouse offerings
* Can run queries using Exabytes of data that is stored in AWS S3 storage services
* Redshift can be scaled up or down to meet your requirements
* Cost effective, no large upfront investment of large data warehouse clusters
* Allows small accounts to access powerful business intelligence applications that were formally only available to only large enterprises due to costs and complexity
* AWS manages the service to remove the deployment and maintenance complexity from customers
* No need to manage the Redshift clusters, you can concentrate on your big data requirements and not be concerned with the underlying infrastructure
* Scales as your demand grows
* Redshift is configured and managed in the AWS console under the database section
* Enables running complex analytics queries on petabytes of structured data
* Features sophisticated query optimization, columnar storage on fast local storage
* Supports massively parallel query executions
* Exam Tip: know that Redshift is the AWS data warehouse offering that handles petabytes of data
* https://aws.amazon.com/redshift/

Elastic Map Reduce: EMR

* Managed Hadoop framework
* Big Data processing
* Process large amounts of data
* Hadoop is the application that is managed by AWS and marketed as Elastic Map Reduce (EMR)
* Hadoop is an open source, Java software framework that supports data-intensive distributed applications running on large clusters of commodity hardware
* Data is loaded into S3, EMR processes the data and then stores it back into S3
* Clusters can expanded or contracted as needed (Elastic)
* EMR can launch inside a VPC
* Multiple pricing options
* Exam Tip: AWS Big Data application that analyzes and processes a large number of datasets with AWS managing all of the applications
* <https://Aws.amazon.com/elasticmapreduce>

Route 53: Domain Name Services / DNS

* Route 53 is the AWS service for Domain Name Services
* Maps domain names to IP addresses (human readable to machine readable)
* Your client computer reaches out to Route 53 (DNS) to resolve a DNS name like www.tipofthehat.com to that servers actual IP address like 192.168.22.148
* You configure and operate your web domains and applications running on AWS
* Route 53 offers the following advanced services:
* Domain registration
* DNS name resolution
* Traffic flow management (send users to the best location)
* Health checking
* Failover automatically if a site goes down
* Integrated with other AWS services such as, S3, ELB, CloudFront
* AWS is a domain registrer that enables you to claim a new domain name such as tipoftthehat.com, you can search and purchase domains from Route 53
* DNS services that offers a 100% uptime global query service for your domain name
* Route 53 DNS servers are located at many different locations worldwide for fast response times
* DNS servers update records at regular intervals
* Route 53 health checking tests the availability of your services before responding to DNS requests to validate its reachability, if it is not reachable, Route 53 can send your traffic to another region or availability zone in AWS for fault tolerance
* Reroutes traffic around failed hosts for disaster recovery
* Exam Tip: Route53 is the Amazon offering for Domain Name Services and resolves host names to IP addresses
* https://aws.amazon.com/route53/

CloudFormation

* Infrastructure as code
* AWS service that you use to model and set up AWS services
* Automatically deploys complete AWS deployments based code you create that specifies how your deployment is configured
* Uses configuration templates that describe all of the AWS resources you wish to deploy
* CloudFormation provisions the resources for you
* There is no need to create and configure resources individually
* Exam Tip: Know that CloudFront automatically deploys resources for you by using “Infrastructure as code”
* https://aws.amazon.com/cloudfront/

CloudFront

* AWS’s content delivery network
* Places copies of your data (called caches) at edge locations worldwide for fast local access across the globe
* Uses the AWS Edge data centers are located in cities around the world
* Edge locations are separate from regions and availability zones
* Edge locations are directly connected to the AWS global network
* Supports Lambda compute services at the edge
* Route 53 routes requests to CloudFront which then finds the nearest edge location to you and sends you there
* CloudFront greatly reduces the load on your hosting deployment since the data is served remotely from edge locations
* Eliminates the need for all requests to come back to your AWS origin region and site
* When a request is made anywhere in the world, CloudFront will either service it from its local edge location cache, or if it is a first time request, it will fetch the data from your server or storage and then serve it out from the nearest edge location, keeping a copy local for any future requests
* Stores both static and dynamic content such as .html, .css and image files
* In the AWS console under the networking and content delivery section
* CloudFront has built in DDoS (distributed denial of service) protection
* CloudFront adds a level of security by lessening the impact of denial of services attacks by using AWS shield Web Application Firewall
* Web Shield mitigates DDoS (Distributed Denial of Service) attacks along with CloudFront
* CloudFront is a network of edge data centers that are not complete facilities like those in a availability zone, edge locations act as content caching for local connections and as an entry point into the AWS cloud
* Edge locations also support Lambda serverless compute instances
* Content origins can be S3 stored data or your EC2 instances running applications such as a web server
* CloudFront allows fast response, for example, of a user in Cape Town South Africa access data locally instead of having to go all the way to your web server in Northern Virginia, USA
* You set up your site in one region and AWS can disperse your content to edge locations all over the world using CloudFront
* In the AWS console under Networking and Content Delivery section
* Exam tip: Cloudfront is a global AWS content delivery service
* https://aws.amazon.com/cloudfront/

CloudWatch

* AWS’s service that offers monitoring of your operations
* CloudWatch is a network management tool hosted by AWS for AWS service monitoring
* Monitors resources and applications in real time that are running in AWS
* Cloudwatch collects and tracks the performance metrics of your AWS services
* Collects and tracks metrics (operational data) that you wish to monitor
* Collects and monitors log files, sets alarms, and automatically reacts to changes in your AWS resources
* Sends alerts and notifications or can automatically take action and make changes if a metric is out of normal parameters based on your defined criteria or rules
* CloudWatch monitors services provisioned in your account such as your EC2 instances or S3 buckets
* For example, CloudWatch can monitor your EC2 instances memory and CPU utilization over a period of time and display it on a web graph and also initiate an autoscaling event
* Alarms can be configured that can alert you to anomalies or to kick off automation events
* CloudWatch also collects and stores log files from your defined services
* CloudWatch dashboards are graphical web pages that gives you an overview of your operations, you can define what you want displayed in your dashboards
* In the console under Management and Governance
* Alarms can be triggered for billing thresholds exceeded, number of objects in an S3 bucket, Storage read write volumes and many other events
* Create and monitor customer defined metrics
* Alarms and events trigger actions for automation, this is a huge value of CloudWatch
* Exam Tip: Know that CloudWatch is the network management for AWS service monitoring
* <https://aws.amazon.com/cloudwatch>

CloudTrail

* AWS service that logs all changes in your account
* Keeps a record of all actions taken by services such as IAM, Management console, CLI, SDK or changes made by other services
* Any change made in AWS creates a record that is written to CloudTrail
* CloudWatch gives you a complete history of events in your AWS account
* All AWS services use API (Application Programmable Interfaces) underneath the hood. API’s also allow developers to work with AWS resources programmatically and for machine to machine communications. CloudWatch records all API calls to AWS
* Tracks all actions taken by users
* Continuously monitors all API activity related to the AWS infrastructure, keeps a long term record of all events and can be used for governance, compliance, operational auditing, and risk auditing of your account
* Simplifies security, analysis, resource change tracking, and troubleshooting
* Allows for record keeping and forensics of all AWS operations
* Helpful for troubleshooting
* Used for security analysis
* Tracks changes in resources
* Logs all changes, that at their root are API (Application Programmable Interfaces) interactions
* In the console under Management and Governance
* Easy to enable, just click a button
* Exam Tip: Use CloudTrail when you want to log all actions taken in your AWS account
* https://aws.amazon.com/cloudtrail

Simple Notification Services: SNS

* Simple Notification Services, SNS automates the sending of notifications via e-mail or text (and others) when events occur in your AWS account
* SNS is an AWS service that people and devices can subscribe to and receive alerts of AWS events
* Sends automated or manual notifications of events in your AWS account
* Sends to many different devices such as email, text, HTTP, Lambda and others
* Integrated in with many AWS services that can notify SNS to send out alerts based on triggers you define
* There are two parts:
* Publishers (producers)
* Subscribers (consumers)
* Publishers send messages to subscribers asynchronously (it’s a “push” service)
* Publishers produce and send the message based on topics
* Publishers can be humans, alarms, events that send the alarm to SNS to be sent out to the subscribers
* A topic is a logic endpoint and communications portal
* Topics are a grouping of like events
* Topics is where you assign labels to, and group endpoints that receive the messages
* There can be many topics created in SNS for different areas such as billing, storage, compute or any other events in AWS
* Subscribers can be such endpoints as:
* HTTP/HTTPS web services
* Email addresses
* Text messages
* SQS queues
* Applications
* JSON
* Lambda functions
* Subscribers consume (receive) the messages sent out by the Publisher based on topics
* Subscribers must be associates with a topic to receive the messages
* Builds on top of services such as CloudWatch that can be configured to trigger SNS messages when triggers are met
* An example of SNS in the real world would be if an EC2 web server instance went down, CloudWatch would send the details to SNS that would send you an e-mail, text message and run a Lambda instance to restart the EC2 server
* SNS Publishers send notifications to every device that is subscribed to the Topic
* The notifications contain a great deal of information on the alarm event
* In the AWS console under messaging
* Exam Tip: If you need text or email notifications of events, use SNS
* https://aws.amazon.com/sns/

Simple Queuing Services: SQS

* Simple Queuing Service is a AWS managed distributed message queuing systems
* SQS is a web service that gives you access to a managed message queue that can be used to store messages while waiting for other computers and applications to process them
* A distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component thereby decoupling the applications from each other
* SQS decouples the applications and acts as a buffer should the receiving process get behind in processing the data
* A queue is a temporary repository for messages that are awaiting processing
* Application look in the SQS Queue for jobs to run, they poll the queue for waiting messages
* Decouples applications so there are no dependencies between the applications since there is a queue in the middle that hold messages between applications
* Using Amazon SQS you can decouple the components of an application so the run independently, with SQS easing message management between components
* The SQS queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing
* This means the queue resolves capacity issues that arise of the producer is producing work faster than the consumer can process it, or if the producer of consumer are only intermittently connected to the network
* SQS is a pull based system only, it is based on polling the queue, (not a push like SNS)
* A requesting server sends the message to be performed, and a worker server receives the message, locks it, performs the task and then deletes the message from the queue when completed
* Amazon SQS can also be viewed as a class of temporary data storage for many classes of applications
* Exam Tip: SQS is a message store used to decouple distributed systems
* https://aws.amazon.com/sqs/

Trusted Advisor

* An AWS tool to keep track of your cloud resources and helps configure resources to follow best practices
* Helpful in identifying unused services such as EC2 and storage as well as resources that are not optimized
* Trusted advisor evaluates four different categories of your AWS deployment:
* Cost optimization
* Fault Tolerance
* Performance
* Security
* Ensures AWS best practices are followed
* For example, you can use trusted advisor to identify EC2 instances that are not being utilized fully, EC2 reserved instances, underutilized storage volumes such as EBS or EFS
* Compares your account deployment to AWS best practices and compiles a report of recommendations
* Supports automation and interoperability with other AWS services such as CloudWatch, Lambda and others
* Trusted Advisor is under Management Tools in the AWS console
* AWS service that automates account evaluations and suggest ways that you can optimize your configurations
* Online tool that can advise you on ways to reduce your AWS costs, increase performance and improve security
* Provides real time feedback that helps you provision your systems that follow AWS best practices
* The level of detail and support Trusted Advisor provides varies by the level of support plan that you have (Basic, Developer, Business or Enterprise)
* Services available in all support plans:
* Security group port checks
* IAM (Identity and Access Management) usage
* Check to see if Multi-Factor Authentication (MFA) is enabled on the IAM root user account
* Elastic Block Storage (EBS) public snapshots
* Relational Database Services (RDS) snapshots
* Service limits being approached
* Added to Business and Enterprise support plans:
* API programmatic access to refresh and retrieve Trusted Advisor results
* complete trusted advisor checks
* notifications to remain up to date with weekly resource deployments
* Exam Tip: Know that Trusted Advisor is the AWS tool to keep track of your cloud resources and helps configure resources to follow best practices
* https://aws.amazon.com/premiumsupport/trustedadvisor/

AWS Inspector

* Analyzes the behavior of resources in the AWS cloud and identifies potential security issues
* You define a collection of resources as a target for an assessment
* Create an assessment template
* Launch a security assessment that runs against the target
* Exam Tip: AWS Inspector is an AWS service that uses pre-defined security templates that check for vulnerabilities
* https://aws.amazon.com/inspector/

AWS customer support

* Account support plans offered by AWS
* Basic - included with every account
* Developer - Starts at $29.00/month
* Business - Starts at $100.00/month
* Enterprise - Starts at $15,000.00/month
* Each higher level support plan increases the response time and availability of AWS support
* Support is incremental with Enterprise offering the highest level of priority to technical support and customer service
* Basic: No customer support representative access, email support
* Developer: Business hours access to Cloud Support Associate representatives
* Business: 24x7 access to Cloud Support Engineers
* Enterprise: 24x7 access to Senior Cloud Support Engineers
* All plans includes trusted advisor checks
* Exam Tip: Know the different support plans offered and the different service levels
* https://www.amazonaws.cn/en/support/compare-plans/

AWS Security overview

* Security is an integral part of the AWS offering
* Protects the confidentiality, integrity and availability of your cloud deployment
* Security increases trust and confidence by both AWS and its customer base
* The exam covers AWS security products and some of their features, built in controls and an overview of Amazons security model known as shared responsibility
* AWS offers an extensive list of security services to meet the needs of even the most security sensitive applications hosted in AWS
* AWS security offerings saves customers money by eliminating capital expenses of an advanced security infrastructure
* AWS security is engineered for resiliency, high availability and built in safeguards at large scale
* There are constant AWS innovations and new offerings from AWS
* You only pay for the security services you need
* AWS Security solutions:
* Auditing
* Adherence to industry and governmental compliance regulations such as HIPPA and PCI
* Configuration management
* Data security
* Denial of service protection
* Encryption at rest and in transit
* Firewall services
* Identity and access management
* Intrusion detection
* Inventory management
* Key management with flexible array of options
* Logging
* Monitoring
* Standardization using template and definition utilities
* …and more
* A solid security plan lowers the cost of operations
* Enables you to add security services and your needs evolve
* AWS provides Governance attestations and features
* Compliance includes choosing AWS services that meet the compliance requirements and then auditing the deployment to ensure all requirements are being met
* Leverages Amazon’s large enterprise engineering resources
* Wide range of security tools from both AWS and the business partners
* Use Identity and Access Management (IAM), to define privileged between users and services
* Multifactor authentication is used for additional login security
* Corporate user directories integration and confederations
* Open API documentation allows for integrating your security tools with the AWS security offerings
* API logging for change and audit records, who, what when and where the API calls happened
* Logging aggregation
* Automatic alerting to email, text, queuing and Lambda services
* AWS security services reduces your risk profile
* AWS marketplace offers a wide array of security products from AWS partners that complement the AWS offerings
* AWS marketplace bundles vendor application software and licenses for a charge, such as VPN, Firewall, web servers from software vendors
* Exam Tip: Know the suite of AWS security offerings
* https://aws.amazon.com/security/

Account Compromise remediation

* If you think there has been unauthorized access to your account, do the following:
* Change the AWS root account password
* Change all IAM user‘s passwords
* Delete or rotate all access keys being used for API authentications
* If you did not create resources that are now showing in your account, delete them
* Contact AWS support and report a security breach, open a support ticket
* Monitor messages from the AWS support groups
* Exam Tip: Change all access credentials if your account has been compromised
* https://aws.amazon.com/premiumsupport/knowledge-center/potential-account-compromise/

Shared Responsibility Model

* The security of the data and applications in the AWS cloud are shared between AWS and your organization
* AWS defines what their security responsibilities are and what yours are
* AWS is responsible for security OF the cloud, which is the global infrastructure that supports all of the AWS services
* Customers are responsibility of security IN the cloud
* Security is based on best practices and international standards
* AWS is responsible for all facilities in Regions, Availability Zones and Edge locations, compute, storage, database network services, and hardware
* AWS provides reports and audits to the public, however data center visits are not allowed
* Responsible for the security of managed services such as Database and Big Data offerings including OS and application version patching, network security configurations and disaster recovery/failover
* Managed services customer responsibilities include access controls and user account credentials
* Customers are responsible for, and have, complete control over their content in the AWS cloud:
* Access rights granting and revocation (Full IAM control)
* Data format and structure
* Encryption options
* What to store
* Where the content is located (regions)
* Which services to use
* Who is allowed access to the resources and content
* Shared responsibility model changes based on services selected
* AWS services catalog allows users to publish approved services inside their organization
* Exam Tip: Know what AWS is responsible for and what the customer is responsible for
* https://aws.amazon.com/compliance/shared-responsibility-model/

Access control: Identity and Access Management IAM

* Identity and Access Management is referred to in AWS as IAM
* A very critical component of AWS
* Controls access to all AWS resources
* IAM is where you create accounts and control access to AWS services and resources
* For the exam it is important to understand what IAM is and what it is used for, not how to configure it
* Authentication identifies users
* Authorization determines what they can do, how they can use the resources
* IAM controls access to:
* Applications
* Compute
* Databases
* Storage
* …and all other AWS services
* IAM defines:
* Users
* Groups
* Access policies
* User credentials
* Multi-Factor Authentication (MFA)
* Password policies
* Permissions
* Roles
* API cryptographic keys for command line interface (CLI) Access
* Permissions allow and deny access to AWS resources by attaching IAM policies
* IAM is in the AWS console under security, identity and compliance section
* A role is a trusted entity, not necessarily a user but a device or service that is granted access to other services. For example, a compute instance can be assigned a role to access storage services
* When a user is first created, it is not allowed to any services except the ability to log into AWS
* Users must be explicitly granted access to services in the IAM console by assigning policies or permissions
* For example, your IT department may allow you access to compute and storage services for you to do your job
* Roles are created and permissions assigned to the roles to define what actions can be allowed of denied by services that assume the role
* IAM supports user federations and their permissions, such as using a Microsoft Active Directory service to user accounts
* Your AWS account has a root user created when you create your account, this has complete access to the AWS console and services, the root is the email address and password used when creating the account
* IAM users get assigned either management console access or by using programmatic Access
* User passwords are administered in IAM and also allows the operator to create password polices such as how long it is and how often it must be changed
* Multi-factor authentication (MFA) is a higher level of security that uses another device that after you enter you enter your username and password, you must enter a changing code. This code is presented on the MFA device which may be a key fob or smartphone application or other types of output devices
* MFA is used to protect your account against unauthorized access and is managed in IAM
* IAM policies explicitly list permissions and can be assigned to either users, groups or roles
* IAM policies grant user access to the AWS management console an programmatic access to services
* IAM is in the console under the Security, Identity and Compliance section
* In the AWS shared responsibility model, the user has full IAM control and AWS cannot access it
* The root account provides unrestricted access and should never be used in normal operations
* It is a good practice to enable multifactor authentication where you enter a username/password and then a randomly generated one time number string
* Give users only the permissions needed, this is known as the principle of least Access
* Put users in groups and assign permissions to the groups for ease of management, this saves the work of having to assign polices to every user, just make a group, assign the policies and put users into the group
* Users can be assigned policies at the user and group level at the same time
* Create a password policy that defines strong passwords and rotation intervals
* Roles are rights assigned to AWS services such as EC2 or S3 to allow them access to other resources, for example, it an EC2 instance needs to access data stored in an S3 bucket, a role needs to be assigned that grants it permission to access that data in S3
* Roles grants services permissions to communicate and share data with each other. By default they are denied and must be explicitly granted
* Polices are attached to roles and then the role is granted to the service
* Test questions that reference “access”, “groups” or “rights” it may very well be about IAM
* https://aws.amazon.com/iam/

Security compliance

* The AWS shared responsibility model outlines the shared control between AWS and the customer
* AWS controls the data centers and resources
* Customer control their deployments in the AWS cloud
* AWS offers a controlled platform that is highly secured and then adds a large number of security features that you can use
* Customers are responsible for configuring their services in a secure manner
* AWS does not see, and is not responsible for, customer security configurations
* AWS communicates openly about its security posture and control practices
* AWS meets a large number of industry and governmental certifications, regulations and controls
* AWS provides third party attestations
* Provides security documents to customers under NDA (non-disclosure agreements)
* Certifications and attestations are performed by third party independent audits of AWS
* Customers are ultimately responsible for complying with all laws, regulations and privacy requirements that pertain to their operations on the AWS platform
* AWS offers support for customer compliance
* Compliance frameworks are offered for specific areas such as healthcare or financial transactions
* AWS provides white papers, examples, playbooks and general information about its security programs to assist customers in creating their compliance and security posture
* Three main risk and compliance pillars
* Risk management
* Control and environment
* Information Security (InfoSEC)
* Risk management is the identification and management of risk factors in the AWS cloud
* Risk management is reviewed and evaluated bi-annually by AWS
* Ongoing internal and external risk assessments are being performed on the AWS cloud
* Internal security policies, employee training, security audits, that asses confidentiality, integrity availability and policy conformance are ongoing at AWS
* AWS scans all services, endpoints and internet facing services for security vulnerabilities
* If vulnerabilities are discovered, AWS will notify the interested parties for remediation
* AWS hires independent outside security organizations to perform security assessments
* Customers are responsible for meeting their own compliance requirements
* Do not perform scans on your deployment without first obtaining AWS’s permission
* Amazon is responsible for securing their control environment, polices, control activities and operational system, this includes the systems, people and processes to deliver AWS services
* AWS supports all leading practices and is a member of many cloud development groups
* Information security programs protect the confidentiality, integrity and availability of customer data
* Exam Tip: Customers are responsible for the entire environment they run in the AWS cloud, they must understand the compliance requirements they are required to meet and implement a control environment to meet and verify those requirements
* www.aws.amazon.com/compliance

Security Resources

* AWS publishes a large amount of security content including:
* Advisories
* Attestations
* Blogs
* Certifications
* Internal AWS security practices
* Technical bulletins on security threats
* Web publications
* Whitepapers
* Additional information provided under non-disclosure agreements (NDA)
* Offers professional services and support to all customers
* There is a large ecosystem of third party security and consulting services available
* Trusted advisor outlines best practices, inspections, increases reliability and enhances performance (see trusted advisor section)
* AWS account representatives are direct AWS contacts that can assist in deployments and give security guidance
* AWS support services:
* Direct support and help desk
* Many offerings, including 15 minute support windows with 24x7 global operational support
* Support delivered by chat, email or pone
* Technical account manager can be dedicated to customers for enhanced support services
* Professional services and AWS partners:
* Assist customers in all aspects of AWS including security consulting and outsourcing
* Assists in meeting all the security requirements the customer may have
* Develop security policies and implementation
* Direct and partner consultants located worldwide for country local support
* Offer audits and remediation services
* Follow industry best practices
* Penetration testing and remote monitoring services
* Uses proven design architectures
* Work directly with AWS and partner experts
* The compliance solutions guide has a wealth of information on meeting security requirements and is on the AWS website given above
* Understand the shared responsibility model given above
* Exam Tip: Read over the links provided to get a good understanding of the various AWS security domains
* AWS publishes a great security blog that keep you updated date on latest security developments <https://aws.amazon.com/blogs/security>
* Read AWS customer case studies to learn how other companies have deployed their fleet of services in the AWS cloud and their security architecture: <https://aws.amazon.com/solutions/case-studies>
* AWS published Frequently Asked Questions (FAQ’s) on all services that provided valuable information on each AWS offering: <https://aws.amazon.com/faqs>
* For complete information on AWS security go here: <https://aws.amazon.com/security>
* For complete information on AWS compliance go here: <https://aws.amazon.com/compliance>

A Well Architected Framework

* Created by AWS as an example of best practices on AWS by referencing thousands of deployments
* Allows for customers to reference consistent and validated design best practices
* Help customers design efficient, resilient, high-performance and secure networks in the AWS cloud
* By following the well architected framework, you can design and deploy services quickly and implement best practices
* Risks can be lowered or eliminated
* Allows customers to make informed decisions on performance, availability, security and costs of deployments
* Advantages of using the well architected framework
* Be able to make more informed decisions
* Build and Deploy applications faster
* Learn and implement AWS best practices
* Lower or mitigate risks
* Five pillars of the well architected framework:
* Cost optimization
* Operational excellence
* Performance efficiency
* Reliability
* Security
* Operational excellence:
* All changes should be well documented, planned, frequent but small increments and reversible
* Compile and update documentation with all changes using automation where posible
* Constantly monitor and modify operational procedures for continuous improvements
* Improve supporting processes and procedures in the cloud
* Learn from past experiences
* Perform operations in code for automation and virtualization using scripting
* Plan and research failures and engineer to remove or mitigate them
* Run and monitor systems to deliver business value
* Security:
* Assessments of all risks
* Centralize the management of privilege assignments
* Design and apply security at all layers and services of your deployment using a defense in depth approach
* Encrypt and protect all data both at rest and in motion
* Implement a separation of duties plan
* Implement strong auditing, recording and tracking of operations. Monitor all alerts and automate responses in real-time
* Plan and prepare for security breaches and failures
* Plan for security detection, investigation and recovery of events
* Plan to reduce or eliminate these security risks
* Protects all information, systems and other assets
* Use automation to implement security best practices
* Use least privilege policies (users granted permissions to access only the resources they need to do their assigned job), implement a well-planned identity posture and foundation
* Reliability:
* Automated changes with automation and implement a change management plan to reduce risk and increase reliability
* Design to automatically recover from reliability events using monitoring and automation features built into the AWS cloud
* Disaster recover should be based on launching resources in separated AWS regions
* Implement scalability to dynamically scale in and out to meet your cloud resource demands
* Plan and test recovery from failures and disruptions
* Plan for events that disrupt your operations
* Plan to recover from failures of infrastructure and systems in the AWS cloud
* Scale out (horizontally) with multiple smaller systems instead of one large instance to reduce the impact of a single point of failure
* Understand your resource utilization and system utilization and automate capacity management procedures to properly scale your deployment to meet current and anticipated workloads
* Performance:
* Expand globally, place data and services closer to users worldwide by deploying in multiple AWS regions and using the CloudFront content distribution network
* Investigate and explore new and different AWS services offerings to increase the performance of your deployment, compare services for the most efficient deployments
* Implement serverless architectures to remove the requirement for you to manage servers in the cloud and to reduce operational expenses
* Maintain these efficiencies as your ongoing demand and service technologies are changing
* Meet your requirements with efficient use of your cloud resources
* Use AWS managed expertise of advanced technologies instead of developing them in-house for faster deployments
* Cost Optimization:
* Constantly monitor and optimize your operations to keep expenses to a mínimum
* Cost optimization based on the consumption model where you only pay for what you use
* Create a model of consuming only the resources that you require at any point in time
* Identify and address resources that are not optimized for your requirements
* Private data center operations are extremely expensive, investigate what can be migrated to the AWS public cloud to reduce these expenses
* Managed services and monitoring using AWS tools and third party services can reduce your cost of operations
* Monitor and measure the efficiency of your operations, know the costs associated with the service you deliver
* Reduce or eliminate unnecessary costs in the cloud
* Exam tip: assume everything will fail, build loosely coupled components where one service or application does not have to depend on the other to opérate
* aws.amazon.com/architecture/well-architected

Reference Architecture: Fault Tolerance

* Fault tolerance is at the system level and refers to the ability of a device to remain in operation even if a competent, such as a power supply, fails
* Fault tolerant systems are highly redundant
* High availability refers to systems that are always online and functional
* High availability reduces or eliminates downtime automatically without operator intervention
* High availability systems often use two separate systems in active/active or active/standby configurations and back each other up in case of one failing, the other takes the workload
* AWS is designed for fault tolerant and high availability operations
* Significant cost savings are realized by using AWS fault tolerant and highly available architectures that you leverage and configure to meet your architecture instead of the high capital expenditures required to deploy these systems in a private data center
* Traditional data center fault tolerant and high availability systems:
* Extremely expensive to implement and opérate
* Used mainly for critical systems due to high cost
* AWS fault tolerant and high availability systems:
* Available on servers, Availability Zones, Regions, services, storage
* AWS has a long list of services that are design to be fault tolerant, you do not have to implement anything special to take advantage of this
* Flexible and elastic services, pay for what you use leverage AWS support expertise and design practices
* Much lower cost than private data centers
* Services such as autoscaling, CloudWatch, Route 53 (DNS), load balancers and floating IP addresses (Elastic IPs) allow you to take advantage of fault tolerant designs
* With the right design many other offerings support high availability such as EC2 servers, virtual Private Clouds (VPC), Caching systems, Direct and VPN network connections
* Fault tolerant services offered by AWS
* Simple Queue Service (SQS) decouples applications by temporally caching jobs delivered by one application to be processed by another application
* Simple Storage Services (S3) highly available and durable static cloud based storage
* The variety of AWS Database services are designed for high availability and offer scalable fault tolerant deployment options, services include, RDS, DynamoDB and SimpleDB, these include multiple replicates, automated backups and multi-AZ designs with auto-failover and data recovery
* Decoupling an application reduced the inter-dependencies so a failure of one component does not affect other components
* Decoupling allows applications to be managed and maintained separately
* If components are tightly coupled together then one component failure will affect the others
* Exam tip: If an EC2 web server crashes but your system is still service web pages from another server, you have a fault tolerant system
* AWS documentation on fault tolerance and high availability is product specific and can be found by searching https://aws.amazon.com

Reference Architecture: Web Hosting

* AWS has many different options to host web sites
* Web hosting is highly scalable and reliable with options to deploy worldwide, closer to your customers
* Very low cost managed web services are offered as well as design your own options
* Compute, storage, database, networking and application services are part of the reference architecture
* Autoscaling dynamically adds and removers servers to match your current workload, no more wasted capacity
* Load balancing spreads the web site traffic across multiple web servers and monitors the health of each server
* AWS offers many advanced services such as mobile, content distribution, domain name services, traffic distribution, authentication, video, audio, translations and much more
* AWS Web hosting product offerings:
* Autoscaling
* CloudFront (Content Distribution)
* Database offerings (RDS and DynamoDB)
* Elastic Compute Cloud (EC2) and Application servers
* Elastic Load Balancing (ELB)
* ElastiCache
* Firewalls WAF and Shield
* Simple Storage Services (S3)
* Route 53 (Domain Name Services / DNS)
* Virtual Private Cloud (VPC)
* AWS web offerings are all virtualized into code for cost efficiency and flexibility
* Security is built in at every stage of your web deployment on AWS
* Highly available architecture using built in durability of all services placed into multiple Regions and Availability Zones as well has CloudFront edge content distribution centers worldwide
* Exam Tip: Know the services that make up a standard web deployment
* AWS documentation on web hosting is product specific and can be found by searching https://aws.amazon.com/websites/

Pricing of AWS services

* As a general rule, you pay for:
* Compute resources
* Data Transferred out
* Storage consumed
* You generally pay only for what you use
* No charge for data transferred into AWS
* No charge for data transferred in the same AWS región
* AWS pricing is based on the resources consumed for each service
* This is often referred to as Utility Pricing or pay-as-you-go
* Each service is priced independently of the other AWS offerings, many are free
* There are no upfront costs with AWS services
* Charges stop when you stop or terminate the service or feature
* There are no long term contract to agree to use AWS
* AWS offers volume discounts, the more you use, the lower the Price
* No termination charges are applied to AWS services
* Pricing is less if you reserve AWS resources or consume more of them (quantity discounts)
* AWS is constantly reducing their pricing as they become more efficient and larger, leveraging economy of scale in their hyper scale data centers
* No more costly upfront expenditures to build, run, staff and maintain your own data center.
* With AWS you only pay for what you use and costs are dispersed between thousands of customers
* Variable costs are lower in AWS than in private data centers
* Pay for the services only when needed, all services are on demand
* Access extremely high costs services such as Big Data and Artificial Intelligence applications that were previously too expensive except for only the largest companies in the world are now available to everyone at a reasonable Price
* Cost effectively and quickly adapt to business needs as they change
* Eliminates long procurement cycles for equipment in a private data center
* Reserved capacity for storage, database and compute services allow savings as high as 75% compared to on demand capacity offerings from AWS
* Reserved capacity options: (The larger the up-front payment, the greater the savings)
* All up-front payments
* Partial up-front payments
* No up-front payments
* Volume discounts are offered by AWS so you pay less when you use more of their services
* Savings are incurred as your usage increases
* Pricing is tiered for many services such as data transferred out (data transfer in is free) of S3 and EC2, the more you use the les the service costs
* There are extensive storage tiers and lifecycle management offering designs to lower the pricing of storage based on your access and retrieval performance requirements
* AWS operational costs are low based on their efficiency and scale, these costs are passed along to their customers
* AWS offers custom pricing for unique customer and high volume requirements
* AWS offers a free tier for one year that give you time to work with and experiment on AWS
* Many services and offerings are free of charge such as VPC, Elastic Beanstalk, CloudFormation, OpsWorks, Identify and Access Management (IAM), Autoscaling and more
* No long term contracts are required
* Services can be stopped and started at anytime
* EC2 (Elastic Compute Cloud) pricing:
* AMI type (Operating system) Linux or Windows options with varied pricing depending on features and licensing options
* Detailed monitoring, CloudWatch enhanced monitoring, fixed monthly rate
* Elastic IP addresses from the time they are allocated until they are deallocated
* Instance configuration, capacity, number of cores, región
* Instance type, hundreds of instance types available of varying capacity
* Load balancing, number of hours and the amount of data
* Number of EC2 instances
* Operating system and application software packages varies on selections, any licensing fees
* Pay for only the capacity used, charged when running
* Per second billing based on an hourly rate
* Purchasing option selected: reserved, spot or on-demand pricing
* Region you select can effect EC2 pricing
* There is a charge for data transferred into and out of an EC2 instance
* <https://aws/amazon.com/ec2/pricing/>
* S3 Storage (Simple Storage Services) pricing:
* Charged per GB
* Data transfer out of the S3 region
* Number of requests, get, put and copy requests
* Price varies by region and storage class
* Storage class, standard, infrequent access, glacier effects pricing
* Storage quantity, amount of storage used
* <https://aws/amazon.com/s3/pricing/>
* EBS Storage (Elastic Block Storage) pricing:
* Data Transfer, inbound data transfer is free and outbound data transfer is charged on a tier depending on the región
* IOPS, input output operations per second, included with general purpose, Magnetic based on requests and Provisioned IOPS is charged by the amount you provision
* Pricing based on Storage class, general purpose (SSD), Provisioned IOPS (SSD), Magnetic Hard Drives
* Snapshots, charged for backup images you create and store in S3, charged in Gigabytes per month
* Volumes charged by gigabytes per month provisioned
* RDS (Relational Database Services) pricing:
* Additional storage billed by gigabyte per month
* Clock hours of server time charged when resources are running launch to termination of data base instances
* Data transfer costs, inbound is free and outbound is a tiered pricing model
* Database characteristics, capacity of database, engine, size, memory class
* Database purchase type, priced per hour, reserved or on demand options
* Database requests, the total number of input and output requests to the database
* Deployment type and configuration, single or multiple availability zones, storage, I/O charges vary
* Number of instances running, multiple databases and be running at the same time and charged per database
* Provisioned storage, backup storage is not charged for active databases, billed for storage when databases are terminated
* Cloudfront Pricing:
* Data transfer out of CloudFront edge locations
* Number of requests, the type of requests and the geographic región
* Traffic distribution costs, varies by edge location
* Autoscaling pricing
* Free however, you are charged for the services that are enabled by autoscaling
* A free tier is offered by AWS that allows you to evaluate many services for one year free of charge <https://aws.amazon.com/free/>
* Exam Tip: Know that AWS uses the pay-as-you-go pricing model
* <https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf>
* White Paper: AWS pricing <https://d1.awsstatic.com/whitepapers/aws_pricing_overview.pdf>
* https://aws.amazon.com/pricing

AWS Organizations

* AWS Organizations is a way to manage billing and access to multiple AWS accounts using one consolidated user interface
* This allows you to centrally management multiple account in one AWS console
* Allows you to create additional accounts using API’s for automation
* Simplifies and consolidates billing or an organization that owns many AWS accounts
* AWS organizations is a no-charge service
* Organizations Features:
* Automation of account creation
* Centrally manage access policies in many AWS account
* Consolidated billing
* Controls access to your AWS services
* Configured in the AWS console under your account pull down tab in the upper left hand corner
* Policy based management for multiple AWS accounts
* Organizations allows you to create groups of accounts and then apply policies to them
* Consolidated billings shares volume discounts of services across all of your AWS accounts
* Exam Tip: Know that Organizations are used to consolidate accounts
* https://aws.amazon.com/organizations/

Total Cost of Ownership (TCO) Calculator

* The Total Cost of Ownership calculator is a free AWS tool that allows for estimated cost savings comparisons between the AWS public cloud and your on-premise data center
* Provides insight into costs and savings of deploying services in the AWS cloud
* Show costs savings by avoiding upfront capital expenses associated with purchasing compute resources compared to the AWS pay-as-you-go model
* Provides direction on pricing guidance for saving costs in AWS
* A cost approximation tool that is offered free of charge
* You input data into the calculator and the service generates a cost report
* You can add and change parameters to tune the output and do comparisons
* The number of servers being migrated is a key component in the TCO calculator
* A tool that estimates the cost savings of using AWS vs. using an on-premises data center
* The total Cost of Ownership Calculator does the following:
* Allows you to modify requirements and the TCO assumptions
* Looks for, and recommends cost saving and provides an estimate
* Provides detailed reporting of costs for AWS services
* Run the tool from here: <https://awstcocalculator.com>
* Basic and Advanced options such as:
* Country
* Currency
* Environment
* Region
* Server type and quantity
* Storage
* Workload types
* …many other options
* After you enter the values shown above, you click calculate and the TCO tool gives you a cost breakdown and comparison to on-premise and AWS services costs
* The change input selection takes you back to the original input page so you can make changes and run the tool again
* The calculator breaks down costs and shows charts to illustrate your anticipated costs of services running in the AW cloud by compute, servers, storage, networking and other services
* Reports can be downloaded or stored in S3 for further analysis
* Exam tip: Know that pricing analysis comparing on-premise and AWS costs is called the TCO or Total Cost of Ownership Calculator
* https://aws.amazon.com/tco-calculator/

Cost Explorer

* Detailed reports of your AWS costs
* The Cost Explorer is a free tool
* Enables you to view charges of your cost (helps you analyze your expenditures)
* Shows costs and billing activity for the past 13 months
* Can be used to forecast expenses three months into the future
* Identifies AWS spending patterns and trends
* Show services usage and usage per location
* In the AWS Console under AWS Cost management
* Recommends what reserved instances to purchase
* Exam Tip: Know that cost explorer is sued to exam your past, current and estimate future expenses of you AWS account
* https://aws.amazon.com/aws-cost-management/aws-cost-explorer/

AWS Support offerings

* Support offered by AWS that includes tools and expertise
* Complete support for all AWS services
* Support plans are designed to meet all customer needs
* Plan, deploy and optimize services
* For planning and proactive support there are AWS technical account managers that are a customer’s primary contact for complete lifecycle support
* Support assists in implementing best practices with trusted advisor offerings that is an online tool that evaluates your deployment and recommends changes to optimize services and configurations
* The support concierge offers account assistance for billing and your account all nontechnical issues you may have and is available at the Enterprise level
* AWS support plan offerings:
* Basic support
* Developer support
* Business support
* Enterprise support
* Support is linked on the main AWS page on the upper left hand corner pulldown menú
* Complete explanations, options and pricing are linked off of this site
* Read the explanations of the different support plans and know the characteristics of each for the exam, for example a developer support plan offers email access during normal business hours and allows only one contact to open tickets
* Exam Tip: review and memorize the options offered for each type of support plan
* https://console.aws.amazon.com/support/home

AWS VPC Security

* This section describes the differences between security groups and Network Access Control Lists (NACL)
* Every instance (EC2) has a security group that controls traffic into and out of each EC2 instance
* Security groups define traffic by IP range, protocol and port number (TCP/IP values)
* By default a security group will deny everything incoming and you need to configure each SG to allow needed traffic into your compute instance
* However, a security group, by default, allows all traffic to exit an instance unless you change the configuration
* IP subnets (groups of devices such as EC2 instances) have network access control lists also, these work in parallel to security groups, these are more often called NACL’s
* NACL’s filter on IP range, port and protocol, open (permit any/any) be default
* NACL’s and security groups have a fundamental difference in their operations
* NACL’s are stateless which means that you have to allow traffic in and out with a rule for each direction
* Security groups are stateful which is to say, traffic allowed in one direction is automatically allows back the other direction, only a rule in one direction is required
* Security groups can restrict traffic between instances in the same subnet where a NACL only is in effect for traffic entering and exiting the subnet
* For security configurations in AWS it is suggested to use security groups first and then NACL’s for backup and override purposes
* Exam Tip: Understand the differences between security groups and Network Access Control Lists
* <https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html>

Security Groups

* Secures your application and services inside of AWS
* Security groups protect individual instances
* Security groups are assigned to every instance inside of a VPC
* Security groups act as a firewall wrapped around an EC2 instance
* Configuration changes are applied immediately by AWS
* Customers must define the security group name, description and rules
* The default configuration is to block all incoming traffic to the EC2 instance, you must configure what traffic is allowed in and from where
* You must explicitly configure what is allowed in by defining the IP address (or range), protocol and port number
* Security groups prevent malicious activity by defining what devices are allowed to connect to others such as a web server open to the internet but a layer of security that application servers can only talk to the web server and database server. The database server would then be configured to only allow communications (via the security groups) from the application servers
* Exam Tip: Security groups act as a firewall wrapped around an EC2 instance
* https://docs.aws.amazon.com/vpc/latest/userguide/VPC\_SecurityGroups.html

Shared Responsibility model

* The shared security model outlines who has what responsibility to secure your AWS deployment
* AWS will be responsible for certain aspects and you, the customer has responsibility for other, each is 100% responsible for their área
* As a general rule, AWS is responsible for security of the cloud and you are responsible for security in the cloud
* Defines what you and AWS are responsible for in meeting security and compliance needs
* AWS is responsible for, manages, and operates components from the host operating system and virtualization layer downwards to the actual physical security of their data centers hosting these services, they take responsibility for the physical hardware and buildings
* AWS is responsible for the installation, configuration and maintenance of the physical hardware located in every AWS data center
* AWS secures the data centers with guards, locks, cameras etc.
* AWS secures that virtualization and automation software systems
* The customer is responsible for management of the guest operating systems (this includes all system updates and patches), application software and security configurations in such services as IAM user account management, NACLs, security groups in VPCs, client and server side encryption and other application security parameters. This list is just an example to understand the demarcation of responsibilities between AWS and the customer
* AWS published a document outlined all responsivities in detail: <https://aws.amazon.com/compliance/shared-responsibility-model/>
* AWS is responsible for the physical data center facilities, the network and hypervisor
* Customers are responsible for the guest operating systems, application and user data
* If the application is offered as an AWS service, such as the database offerings, AWS will assume responsibility of the guest OS and application as well
* The customer is always responsible for the management and protection of their data
* Amazon tends to keep much of their information secret for enhanced security, however, you can reference https://aws.amazon.com/compliance to see the results of third party audits
* When running EC2, you are responsible for the operating system, applications and data
* Customers are responsible for the data security in their account
* AWS does not have visibility into your instances and data, AWS is restricted from accessing your systems hosted in their facilities
* Exam Tip: AWS is responsible for what is IN the datacenter and you are responsible for what is RUNNING in the data center
* https://aws.amazon.com/compliance/shared-responsibility-model/

Distributed Denial of Service (DDoS) and Penetration Testing

* When there is a malicious attack that floods your service with a massive number of connections that overwhelms your deployment and causes it to fail
* AWS builds into many of its services DDoS mitigation and protection:
* CloudFront
* Elastic Load Balancing (EBS)
* Route 53 DNS
* VPC (Virtual Private Clouds) and Security Groups
* WAF (Web Application Firewall)
* Penetration testing allows you to test your fleet for vulnerabilities, validate configurations and benchmark your security profile by simulating an attack
* You MUST get AWS permission prior to undertaking any penetration testing on their platform and test ONLY your instances and no other AWS customers deployments
* Penetration testing is allowed (with AWSs permission) on the following services:
* API gateway (programmatic interfaces)
* Aurora (database)
* CloudFront (content distribution)
* DNS Zones (Route 53 domain name services)
* EC2 (Elastic Compute)
* Lambda (serverless)
* LightSail (automated deployment)
* RDS (databases)
* Exam Tip: Know that you must have AWS permission before doing any testing in your account
* <https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/>
* https://aws.amazon.com/security/penetration-testing/

IAM Users, Groups, Roles: AWS Authentication and Authorization

* Use Identity and Access Management (IAM) configure and control access and security permissions in the AWS cloud
* IAM defines users are a permanent object, usually a human but it can also be an application or other service such as a computer
* A group is a number of users placed into a common configuration area, the users share the permissions assigned to the group
* Roles are a method of authentication that is assigned to users or groups, credentials in a role are temporary
* There are two steps, authentication and authorization
* Authentication determines the “who”
* Authorization is the “what can they do”
* Policies are the authorization piece and define what the users, groups and roles can do, what they can access and other details
* Policy documents are formatted in JSON and get attached to users, groups and roles
* Policy Documents are permit statements to AWS resources
* Policies are formal statements of permissions
* Explicit denies in a policy will always overwrite an allow statement
* You always have to explicitly allow a permission for it to be active
* There will always be an implicate deny at the end of a policy statement, if it is not explicitly permitted, then it is implicitly denied (it is a hidden blacklist at the end)
* All IAM actions can be logged in CloudTrail since everything uses and API call. This logging is for both successful and unsuccessful operations
* Exam Tip: Know that you use Identity and Access Management (IAM) to configure and control access and security permissions in the AWS cloud
* https://aws.amazon.com/iam/

Data Encryption

* AWS offers an extensive array of cryptography services and products
* Data can be encrypted at rest (storage) and in transit (over the network)
* Focus on the key management services AWS has available to you and what they do
* The symmetrical key is used to encrypt the data which is then put into storage, when it is read again, the key is used to unencrypt the data. Without the key, it is impossible to read the data
* AWS offers services to protect the keys
* There can be many different keys you use for different data sets
* If you have millions of objects in storage and encrypt each one using a unique key, you will need to keep track of millions of encryption keys. AWS can manage this for you
* There is a master key and the symmetrical data key, when combined together you derive an encrypted data key that can be used encrypt and unencrypt the data
* AWS key management services include:
* Cloud HSM (Hardware Security Module)
* KMS (Key Management System)
* Security is a very complex topic and not covered deeply in the practitioner exam, just be aware of it
* Exam Tip: Know that there is a master key and the symmetrical data key, when combined together you derive an encrypted data key that can be used encrypt and unencrypt the data
* <https://aws.amazon.com/cloudhsm/>
* https://aws.amazon.com/kms/

Encryption of Elastic Block Store (EBS) Volumes

* AWS offered encryption services to Elastic Block Storage volumes
* AWS manages the encryption of data stored in EBS, this is known as encrypted data at rest
* You configure server side encryption when you configure the EBS volumen
* Then the services will take call of all encrypt and decrypt operations in the background for you
* You can detach and then reattach a volume without compromising the encryption
* If you attached the EBS volume to another EC2 instance or create a snapshot, AWS will keep track of the keys and you will not lose the data
* This is done by the EBS volume transmitting a copy of the encrypted volume key to the EC2 instance which them gets the decrypt key from the AWS KMS service (Key Management Service)
* Exam Tip: AWS offers encryption services to Elastic Block Storage volumes
* https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

Networking

* Routing tables direct traffic to IP subnets
* A VPC can have multiple subnets and uses the routing table to know where they are and how to send traffic to them
* Routing tables have a “default” route to send traffic it otherwise does not know what to do with, this is almost always a default route to the internet and is shows as: 0.0.0.0/0 which means “any”
* Each VPC gets assigned a “master” or “global” block of addresses that later get separated into smaller networks called subnet. This large block is called a CIDR (Sigh-Der) block and is usually donated in the 10 series and as 10.0.0.0/16, however, you can choose others blocks to suit your needs
* Blocks of computers or applications get divided into subnets such as 10.0.1.1-255 and 10.0.2.1-255 and so on
* All subnets are a subset of the main CODR address block assigned to the VPC
* The routing tables directs traffic between these subnets as stated above
* All subnets must be in a unique address block
* Networking is the process of enabling communications inside the VPC and also, into and out of the VPC
* Network Access Control Lists (NACL) are filters that permit or deny traffic into and out of a subnet
* Security groups filter traffic at the server (EC2) level inside of a subnet
* Public subnets in a VPC allow traffic from the outside (usually the internet) to connect to devices in that subnet. Private subnets cannot be reached from the outside, only other subnets inside the VPC
* You can have multiple public and private subnets
* Traffic in and out of a VPC passes through routers that are called gateways
* An Internet Gateway (IGW) routes traffic to and from the internet from your VPC
* Packets from the public subnets use the Internet Gateway and default route table to access the internet
* Route tables can also use a “public” route table attached to the public subnets that uses the IGW
* Since there is no external route to the private subnets, users from the outside world cannot reach the servers on the private subnets
* There is a method for the internal servers on the private subnets to access the internet for patches and updates as well as other needs, this is through a Network Address Translation (NAT) device that allows one-way out traffic from the private subnet to the internet
* To manage servers and other devices inside of a subnet remotely (normally they cannot be reached from the internet since they are private), you set up what is called a “jump box” or in other parts of the world they are called “bastion hosts”. These reside in a public subnet that can be reached from the internet, you access the jump box and then start another session that goes out and connects to the servers in the private subnet. There is security set up that allows only the jump box to manage the hosts in the private subnets
* You “land” on the jump box remotely and then “jump” over to your databases, application servers etc. These servers have extensive security configurations that allow only the bastion host this level of heightened access to access these servers
* There can be multiple route tables inside of your VPC, there is usually one for the public subnets appropriately names the public route table and one for your internal subnets called the private route table
* Private subnets use the NAT gateway to access the public internet that allows servers inside the private subnets to reach out to the internet for patch updates and other requirements
* Other VPC networking options for networking include Virtual Private Networks (VPN) and direct connections to your private data center
* Direct Connect is a dedicated private connection from your data center to the AWS cloud that bypasses the internet
* Exam Tip: Know that Network Access Control Lists (NACL) are filters that permit or deny traffic into and out of a subnet and Security groups filter traffic at the server (EC2) level inside of a subnet
* <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>
* https://docs.aws.amazon.com/vpc/latest/userguide/VPC\_SecurityGroups.html

Whitepapers

* AWS offers an extensive collection of white papers all of their cloud offerings
* A huge collection of documents drafted by AWS that cover a wide range of topics specific to both AWS and Cloud computing in general
* For example, white papers include:
* Architecture
* Best practices
* Cloud computing economics
* DevOps
* Frameworks and methodologies
* Guides that explain underlying technologies
* How-to guides
* Introduction to AWS
* Operations
* Papers on all of the services offered
* Pricing and economics
* Security and compliance
* Serverless architectures
* Storage architectures
* White papers are extremely accurate and have been peer reviewed by AWS, outside analysists, customers and partners
* Exam Tip: Know that AWS white papers is a huge collection of documents drafted by AWS that cover a wide range of topics specific to both AWS and Cloud computing in general
* https://aws.amazon.com/whitepapers/

Service Documentation

* Documents from AWS that are specific to every service AWS offers
* Detailed technical information
* A massive collection of documents
* Highly technical and covers fine grained details of each individual service
* Explains how to deploy and use AWS features
* Examples and walkthroughs on how to deploy and operate AWS services
* Frequently asked question documents for services
* Deep dive information on AWS services and technologies
* Documents include:
* API reference
* Developer guides
* Frequently Asked Questions (FAQ)
* General resources
* Projects
* Quickstart guides
* Tutorials
* User guides
* Exam Tip: Know that there are documents from AWS that are specific to every service AWS offers
* https://aws.amazon.com/documentation/

Appendix A: About the AWS Certified Cloud Practitioner Certification exam

* Certification Name: AWS Certified Cloud Practitioner Certification
* Exam ID: CLF-C01
* Duration: 90 Minutes
* Pre-requisite: None
* Exam fee is $100.00 USD
* MuIti-choice / multi-answer format
* 65-70 questions
* Exam offered at designated testing centers worldwide
* Offered in English, Japanese, Korean and Chinese languages
* You must take the exam on site at the testing center
* Passing grade is on a curve based on average scores
* You will be given pass/fail status at the end of the exam
* Later you will be sent information by e-mail from AWS on how to download your certification certificate and access your records
* After passing the exam you are eligible to purchase AWS certified merchandise, they will send you a link to the store
* Schedule the exam using your AWS Certification account: https://www.aws.training/Certification

Appendix B: links to resources

* AWS Certified Cloud Practitioner Exam Blueprint <https://d1.awsstatic.com/training-and-certification/Docs%20-%20Cloud%20Practitioner/AWS%20Certified%20Cloud%20Practitioner_Exam_Guide_v1.4_FINAL.PDF>
* White Paper: Overview of AWS <https://docs.aws.amazon.com/aws-technical-content/latest/aws-overview/aws-overview.pdf?icmpid=link_from_whitepapers_page>
* White Paper: AWS Cloud Best Practices <https://d1.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf>
* White Paper: AWS Well-Architected Framework <https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf>
* White Paper: AWS pricing https://d1.awsstatic.com/whitepapers/aws\_pricing\_overview.pdf

Appendix C: AWS Free Tier

* AWS offers a one year free account that allows you to work with the main AWS services at no charge
* You will get 750 hours of EC2 usage per month, 1 GB of Quicksite, 750 hours of RDS, 5GB of S3 storage, 1 million Lambda request and many other services at no charge monthly for 12 months
* Go to: <https://aws.amazon.com>
* Click on the yellow signup icon on the upper left
* Click on create new account
* Fill out your email address, password and account name, address, phone number and other contact information
* Agree to the license agreement and the terms and conditions
* You must give a credit card number, however, if you stay inside the free tier you will never be charged
* Amazon will send you an authorization code to confirm you are who you claim to be
* After you are registered, you will be given your AWS account number that you use to log into the AWS console at <https://console.aws.amazon.com>
* Upper right of the screen allows you to choose the region you are working in
* Spend time exploring the console and all the services offered, its huge
* Use the free tier billing widget and add the link to the billing console
* Add billing alerts if you like to make sure you are never charged this is in your account information in the upper right hand corner
* https://aws.amazon.com/free/
* Montgomery, Todd. AWS CERTIFIED CLOUD PRACTITIONER CERTIFICATION GUIDE: COMPLETE 2018 CLF-C01 EXAM STUDY GUIDE (AWS Certification Guides) . TipOfTheHat. Edición de Kindle.