Notes

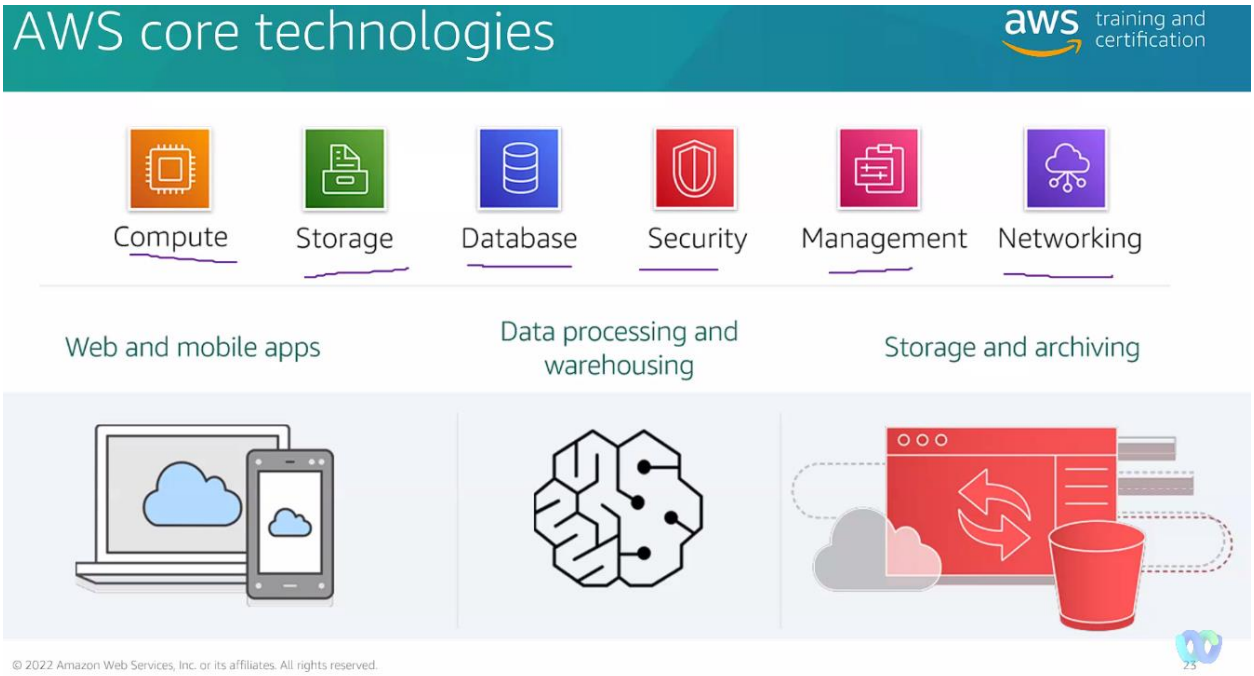
Resources

- Database links- Databases https://aws.amazon.com/products/databases/
- RDS https://aws.amazon.com/rds/
- Aurora https://aws.amazon.com/rds/aurora/
- DynamoDB https://aws.amazon.com/dynamodb/
- Database Migration https://aws.amazon.com/dms/
- Managed Blockchain https://aws.amazon.com/managed-blockchain/
- Elasticache https://aws.amazon.com/elasticache/
- DynamoDB Accelerator (DAX) https://aws.amazon.com/dynamodb/dax/
- Case Studies https://aws.amazon.com/solutions/case-studies/

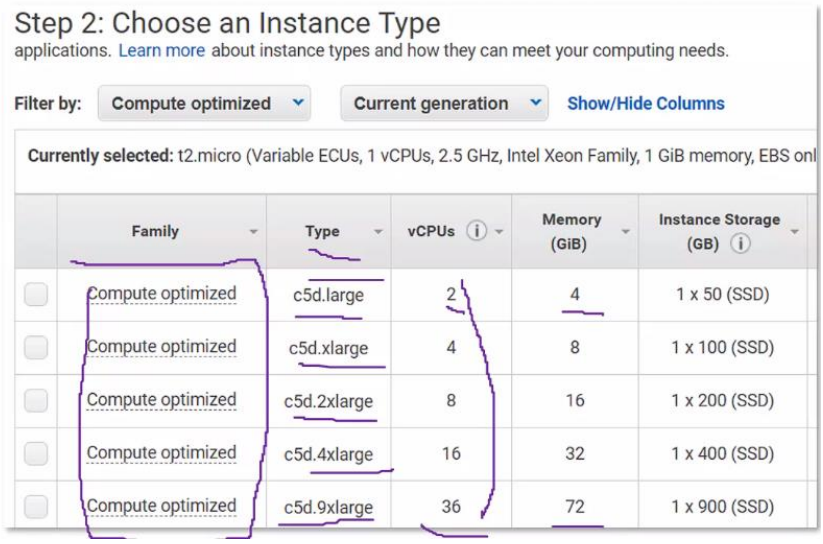
Module 2: Introduction to AWS Core Technologies

- On Premises IT
    - Servers
    - Storage
    - Databases
    - Applications
    - All accesses through corporate network
- Internet
    - Servers
    - Storage
    - Databased
    - Applications
    - All access through the internet
- Cloud computing
    - On demand delivery of compute power, storage, applications, and other IT resources via internet
    - Pay as you go pricing
    - Benefits
        - Agility
            - Fast idea to implementation
            - Gives ability to quickly test and deploy
        - Elasticity
            - Don’t need to provision extra resources for the “just in case” situation
            - Easy to scale up/down in/out
            - Can do it almost instantly
        - Cost saving
            - Variable expenses
                - Spend exactly what you need instead of overspending
        - Deploy globally in minutes
            - Global infrastructure to deploy around the world in minutes
- Regions are made up of 3 or more availability zones which is made up of data centres

    - Interconnectivity with isolation
- AWS global infrastructure other options
    - Local zones
        - Near high pop or industrialized areas
        - Can connect to when need even lower latency
        - Extension of region that is close to high pop areas
    - Wavelength zones
    - Direct Connect Locations
        - Corporate Datacentre to AWS
        - Not the internet
    - Edge locations
        - For content distribution
        - Nothing to do with region/availability zone
    - Regional edge caches



-
- Compute Services
    - Can develop deploy run and scale workloads in AWS Cloud
    - Amazon Elastic Compute Cloud (EC2)
        - It is a virtual machine
        - Secure, Resize compute capacity.
        - Benefits
            - Elasticity in minutes through auto scaling
            - Control
                - Ability modify settings
                - Start/stop etc
            - Flexibility
                - Multiple instance types

        - Divided into General purpose, Compute optimized, Memory optimized, Accelerated Computing, and Storage optimized
            - Each of these have different families and within those are different generations
    - OS types
    - Different sizes
    - 

- Integrated
    - Integrated with all other AWS services
- Reliable
    - High availability
- Secure
- Cost effective
- Easy

- Amazon EC2 Auto scaling
    - Increase or decrease number of instances automatically based on user requirements
    - You have to tell auto scaling group when to scale out
        - Through network/CPU/memory utilization
    - You have to also tell when autoscaling group when to scale in
        - Through network/CPU/memory utilization
    - Has minimum # of instances, Maximum # of instances, and Desired Capacity # of instances (Optional)
- Elastic Load Balancing
    - Distribute incoming traffic
    - Single point of contact
    - Increases availability and fault tolerance
    - Evenly distribute workload among resources
    - Types
        - Application load balancer

        - App layer
    - Network load balancer
        - Connection/transport layer
    - Gateway Load Balancer
        - Gateway load balancer
- Elastic Container Service
    - For Docker
- Elastic Kubernetes Service
    - For Kubernetes
- AWS Lambda
    - Serverless computing where the user doesn't manage/maintain servers.
    - User only worries about the code
- Amazon Machine Images
    - Specify which AMI is used for each instance
    - Can be acquired through the AWS marketplace
    - Essentially has an Operating system

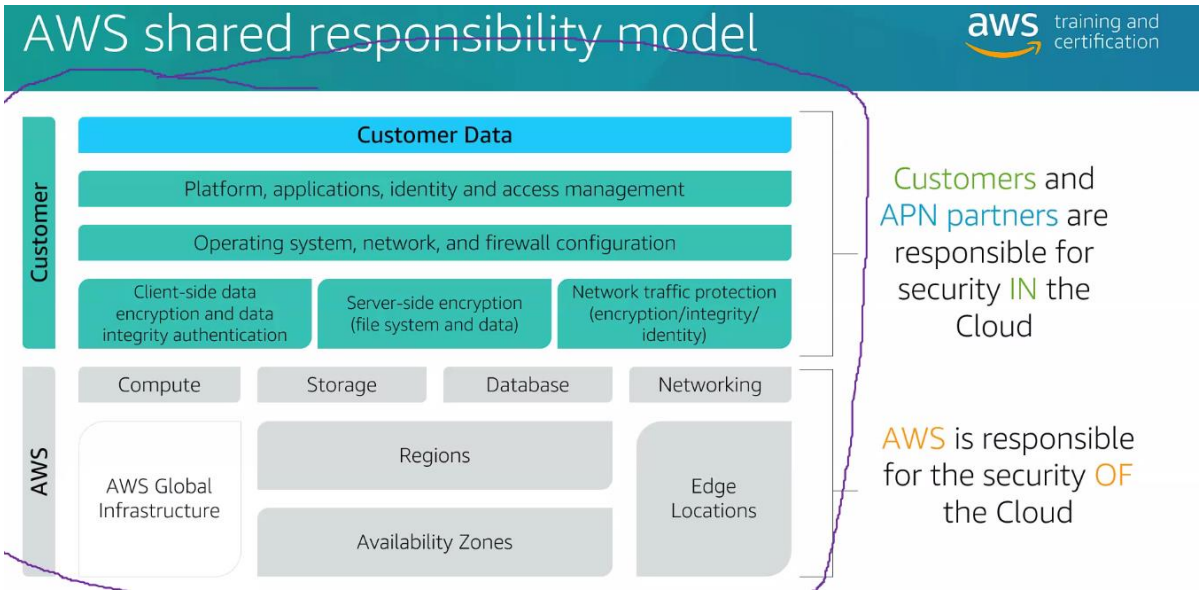
- Storage Services
    - Amazon Elastic Block Store
        - Hard drive for EC2 instance
        - Persistent block level storage
        - Portable
            - Can disconnect and connect to any EC2 instance you want
        - Replicated through entire availability zone
        - Create point-in-time snapshots
    - Amazon Simple Storage Service (S3 Bucket)
        - Store/Retrieve any amount of data
        - Billed per object not per bucket
        - Durable, scalable object storage
        - Data is treated as Objects
        - Fast data retrieval
        - Common uses
            - Data lakes
            - Backup and storage
            - Media hosting Application hosting
            - Software Delivery
        - Storage classes (sorted from High retrieval to Low retrieval)
            - Amazon S3 Standard
                - Multiple copies in each availability zone
            - Amazon S3 Standard – Infrequent Access
                - Multiple copies in each availability zone
            - Amazon S3 One Zone – Infrequent Access
                - Multiple copies in one availability zone
            - S3 Glacier Storage classes
        - Intelligent tiering

            - Automatically moves objects between tiers based on access patterns
    - Amazon Simple Storage Glacier
        - Data archiving and backup
    - Glacier Deep Archive
        - Data archiving and backup
        - Longer time to retrieve than regular glacier
    - AWS storage gateway
        - Connect on-premises to cloud storage
    - Amazon Elastic File System
        - It's the network shared drive
    - Amazon FSx
        - Similar to EFS but can be used for windows etc.
- Database Services
    - Amazon Relational Database Service
        - SQL database
        - Cost efficient and resizable capacity
    - Amazon DynamoDB
        - No SQL
        - Serverless key-value database
        - Fast and predictable performance
    - Amazon Elasti-Cache
        - Fast, managed information retrieval
        - Add caching layers to improve database read times
    - Amazon Redshift
        - Query and analyze data across a data warehouse
    - Amazon Document DB
        - Run mongo dB workloads in document database service
    - Amazon Neptune
        - Run applications that use highly connected datasets
    - Amazon QLDB
        - Review a complete history of changes to your application data
    - Amazon Managed Block chain
        - run a decentralized database
    - Amazon DynamoDB Accelerator
        - Improve DynamoDB response times from single digit milliseconds to microseconds
    - Amazon Aurora
        - Store data in an enterprise class relational database
        - Used instead of MySQL or postgres sql
        - 6 copies of data across 3 availability zones
        - Reduce database costs by eliminating unnecessary input/output operations
        - Backup to S3 bucket
            - opt in/out and does cost extra
    - AWS Database migration service

        - Migrate relational databases, nonrelational databases and other types of data stores

          Example

        -
    - AWS database Services VS EC2 Hosted
        - AWS database
            - Designed to make it easy to set up manage and maintain
            - Push button high availability
            - Automatic backup/recovery
            - Scale up or down based upon pattern
        - EC2
            - Having operating system access
            - Need features not available on the managed service
            - Better for complete control
- Networking Services
    - Amazon Virtual Private Cloud (VPC)
        - Networking layer for AWS resources
        - Build a virtual network in the cloud
        - Virtual network dedicated to a customer's AWS account
        - Subnet
            - Range of IP addresses in a VPC
        - Securing a VPC
            - Network Access Control Lists
                - Control traffic at the subnet level
                - Firewalls at the subnet level
                - Will not remember packet
            - Security groups
                - Control traffic at instance level
                - Firewall at instance level
                - Will remember packet
            - Flow logs
                - Captures network flow information
            - Host-based firewalls
                - Operating system firewalls
    - Amazon Route 53
        - Route end users to internet applications
- Cloud security on AWS
    - It is the highest priority

- Inherits benefits from AWS data center and network architecture
- Similar to on-premises data centres but without need to maintain facilities and hardware
- Can be automated
- Inherits all the best practices of AWS
- 

- The responsibility is not Static can change based on how many services you use from AWS

- IAM (identity and access management)
  - Securely manage access to AWS services and resources
  - Fine-grained access control to AWS resources
  - Multi-factor Authentication
  - Ability to analyze access
  - Integration with corporate directories
- AWS Cloud compliance
  - One way street AWS must tell its customers but customers do not
  - Sharing information
    - Industry certifications
    - Security and control practices
    - Compliance reports directly under NDA
  - Assurance programs
    - Certifications and attestations
    - Laws regulations and privacy
    - Alignments and frameworks

Module 3

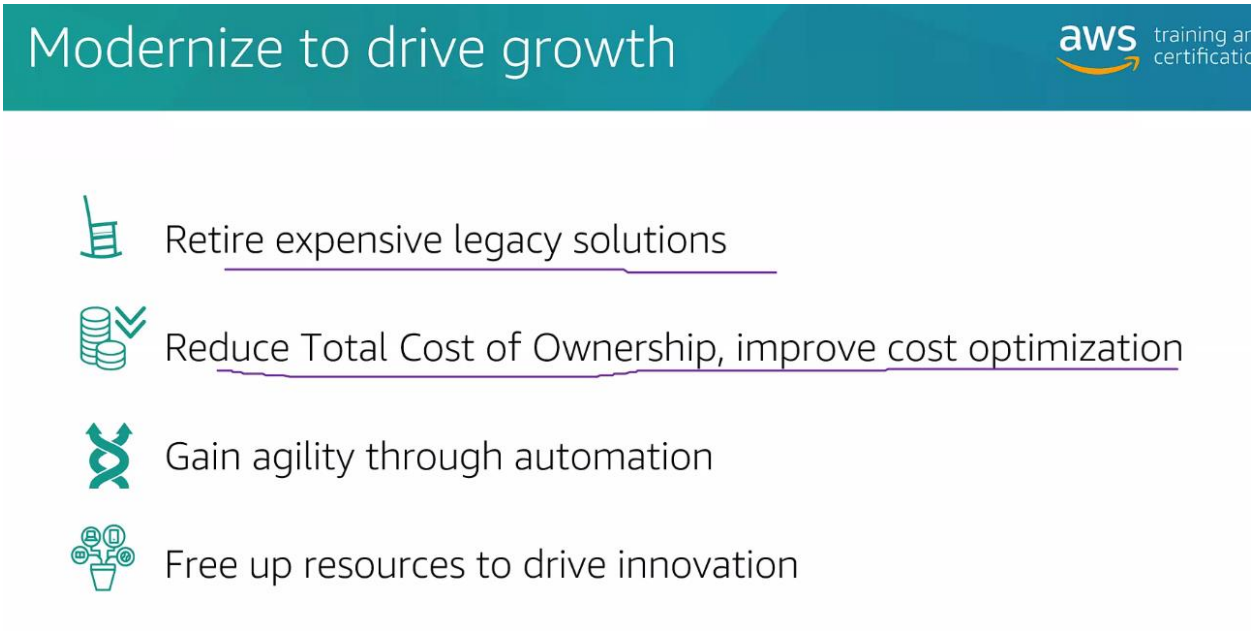
- AWS solutions
  - Machine learning
  - Analytics and data lakes
  - Internet of things

    - Serverless computing
    - Containers
    - Enterprise applications
    - Storage
    - Windows workloads
- Migration
    - 7 R's
        - Rehost
            - Lift and Shift
            - Recreate on premises network only hosted on AWS
            - Automating with tools such as AWS application Migration Service
        - Relocate
            - Hypervisor level lift and shift
            - Migration specific to VMware Cloud on AWS
        - Replatform
            - Lift tinker and Shift
            - Retaining core architecture
            - Making targeted AWS cloud optimizations
        - Refactor
            - Modernize
            - Re-imagining how the application is architected and developed
            - Using cloud native features like containers and serverless
            - Monolithic -> microservice
        - Retire
            - Shutting off non-useful applications
            - Reducing spend, management, and security
        - Retain/Revisit
            - Keep certain applications on-premises
        - Repurchase
            - Moving workflows to Software as a Service
            - Example moving from license to software
                - Microsoft word to Google docs
- Cloud architecture Best practices
    - Design for failure and nothing fails
        - Avoid single points of failures
        - Use multiple instances
        - Use multiple availability zones
        - Separate single server into multiple tiered application
        - Use multi-az (availability zone) feature for amazon RDS
    - Building security into every layer
        - Encrypt data at rest and in transit
        - Enforce principle of least privilege in IAM
        - Configure both security groups and network ACLs
        - Consider advanced security features and services

    - Leverage different storage options
        - Move static web assets to amazon S3
        - Use amazon CloudFront to serve globally
        - Store session state in amazon DynamoDB
        - Use amazon elasti-cache between hosts and databases
    - Implement elasticity
        - Auto scaling
        - Architect resiliency to reboot and relaunch
        - Leverage managed services like S3 and DynamoDB
    - Think parallel
        - Scale horizontally not vertically
        - Decouple compute from session/state
        - Use elastic load balancing
        - Size your infrastructure appropriately
    - Loose coupling sets your free
        - Instead of single, ordered workflow, use multiple queues
        - Use existing services
    - Do not fear constraints
        - Rethink traditional constraints
        - More RAM?
        - Better input/output operations per second for databases?
        - Response to failure?
- The Well-Architected Framework
    - A framework for ensuring infrastructures are
        - Secure
        - High performing
        - Resilient
        - Efficient
        - Sustainable
    - Practices developed through reviewing customer's architectures on AWS
    - Systematic approach for evaluating and implementing architectures
    - Well-Architected (WA) tool in the console
    - 6 pillars
        - Operational excellence
        - Security
        - Reliability
        - Performance efficiency
        - Cost optimization
        - Sustainability
- Keys to consistent results
    - Prepare
    - Anticipate
    - Differentiate
    - Stay on message

- Do not
    - Over promise
    - Use acronyms or technical jargon
    - Focus on technology
    - Focus on the short or midterm
    - Read the slides
- POC fundamentals
    - Practical example of solution
    - Evaluation mechanism
    - Educational tool
- Building a POC
    - Customer agrees to POC
    - Take the feedback form the solution proposal
    - Develop the POC
    - Collect the following information
        - Application code
        - Databases
        - Data
- Resources for POC
    - Building on AWS
    - Migrating Data
    - Validating and Testing
    - Cost of running POC
- Partner opportunity acceleration program
    - Aka POA
    - Accelerate sales cycles and customer adoption
    - Have the POA program coinvest with you by providing
        - Aws promotional credits
        - Cash reimbursements
    - Be aware of expectations and requirements
    - Contact AWS business contact for details
- Implementation
    - Assessment
        - Identify readiness
        - Identify potential business outcome
    - Readiness and Planning
        - Analyze environment
        - Determine migration strategies
        - Create a well-architected landing zone
    - Migration and Modernization
        - Design, migrate, and validated each application
        - Automatic or manual
        - Migrate data
        - Modernize

- Going to production
    - Best practices
        - Involve AWS account team (Solutions architect or technical account manager)
        - Customer specific regulatory requirements
        - AWS support level



-
- Containers
    - Use cases
        - Microservices
        - Batch processing
        - Machine learning
        - Hybrid applications
        - Application migration to the cloud
        - PaaS
- Serverless

Serverless

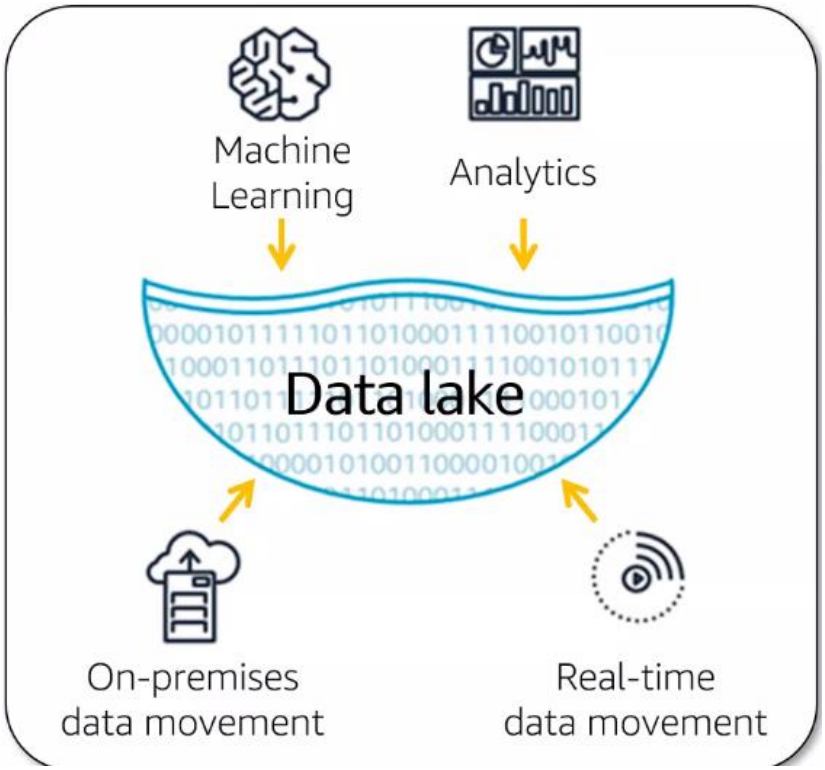
aws training and certification

Category	AWS Fully Managed Services for Serverless
Compute	Lambda Lambda@Edge Fargate
Storage	Amazon S3 Amazon EFS
Data stores	DynamoDB Amazon Aurora Serverless
API proxy	Amazon API Gateway
Application integration	Amazon Simple Notification Service (Amazon SNS) Amazon Simple Queue Service (Amazon SQS) AWS AppSync Amazon EventBridge
Orchestration	AWS Step Functions
Analytics	Amazon Kinesis Amazon Athena
Developer tooling	Various tools and services

Benefits

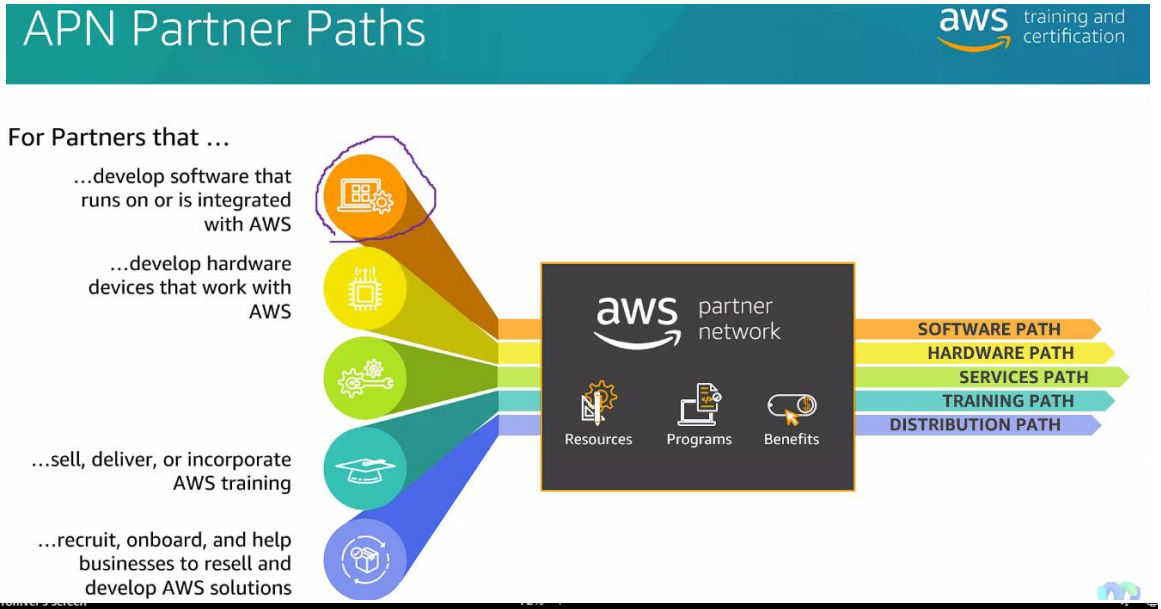
- No provisioning, maintaining, and administering servers
- AWS handles fault tolerance and availability
- Focus on product innovation

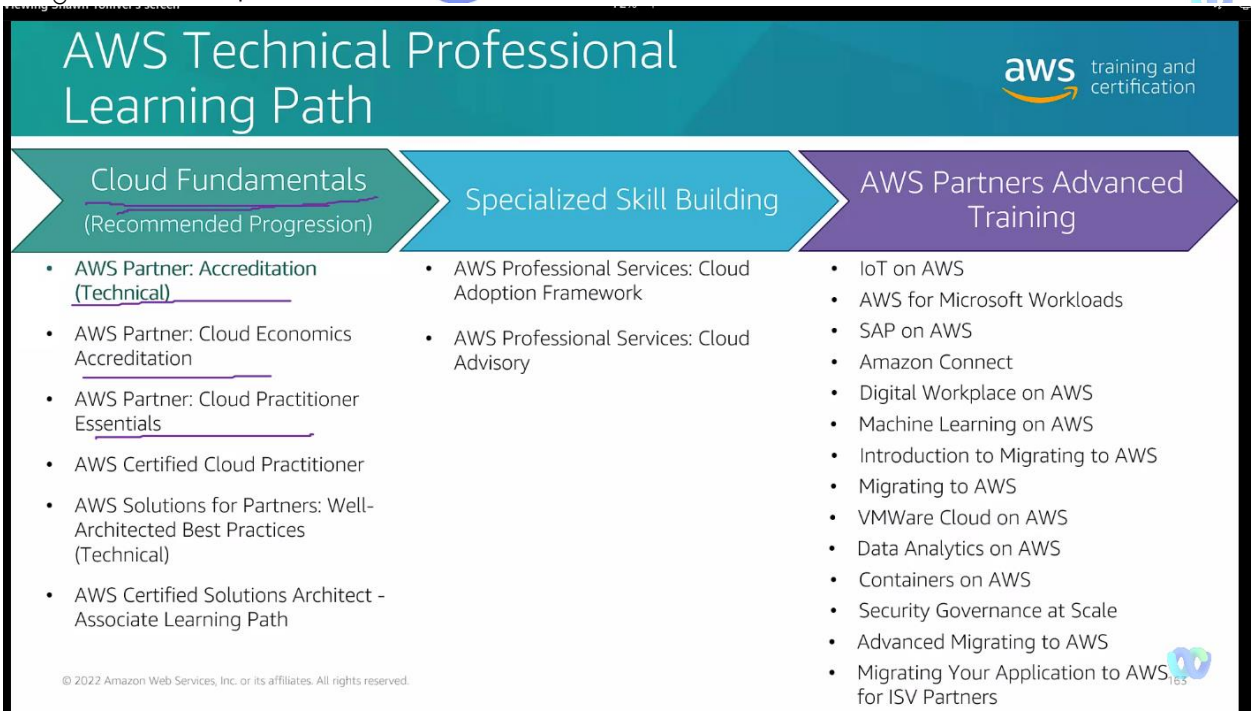
- Analytics and data lakes
  - Value
    - Improved customer interactions
    - Improved research and development innovation choices
    - Increase operational efficiencies



Module 6

- AWS partner network
  - APN benefits
    - Deliver more innovation
    - Get trained and certified
    - Promote your business
    - Highlight your expertise
    - Maximize opportunities with AWS
    - Increase visibility to AWS field teams and AWS customers
    - Save time and money with tools and resources



  - 

-