Exam Readiness: AWS Certified Solutions Architect – Associate

The Exam

- What to Expect
    - Define a solution using architectural design principles based on customer requirements
    - Provide implementation guidance based on best practices
- Multiple Choice
    - Single and Multi Selection
    - No penalty
- 65 questions
- 130 minutes

Domain 1: Design Resilient Architectures (30%)

Choose reliable/resilient storage

- EC2 Instance Store
    - Ephemeral volumes (temporary)
    - Only certain EC2 instances have Instance store
    - Fixed capacity
    - Disk type and capacity depends on EC2 instance type
    - Application-level durability
    - Use Case: Caching or storing temporary data
- Elastic Block Storage (EBS)
    - Different types available
        - SSD for random access, HDD for sequential access
        - General Purpose SSD
        - Provisioned IOPS SSD (for critical applications)
        - Throughput optimized HDD (streaming, big data, data warehouse)
        - Cold HDD (infrequently accessed data, cheapest)
    - Encryption
    - Snapshots
    - Provisioned capacity
    - Independent lifecycle than EC2 instance
    - Multiple volumes striped to create larger volumes
    - Use case: durable attached storage to EC2 instance
- Elastic File System (EFS)
    - Shared file Storage
    - Petabyte-scale file system
    - Elastic capacity
    - Supports NFS 4.0 and 4.1
    - Compatible with Linux based AMIs
    - EFS is mounted in targets in one VPC where the instances can connect to

- Amazon S3
    - Consistency model (distributed system)
        - Strong consistency for new objects
        - Eventual consistency for updates
    - Multiple Storage classes & durability models available
        - Standard, Standard-IA
    - Encryption (at rest): SSE-S3, SSE-KMS, SSE-C
    - Encryption (in transit): HTTPS
    - Versioning
    - Access control with IAM
    - Multi-part upload
    - Internet API accessible
    - Virtually unlimited capacity
    - Regional Availability
    - Highly durable – 99.999999999%
- Amazon Glacier
    - Data backup and archive storage
    - Vaults and archives
    - 3 retrieval options: expedited, standard, bulk
    - Encryption
    - Amazon S3 object lifecycle policy to move data automatically
    - Regionally Available
    - Highly durable – 99.999999999%
- Decoupling Server and Services
    - For health of components
        - With Simple Queue Service (SQS)
    - For scalability
        - With SQS or Load Balancer
    - For identity of components
        - With Elastic IP Address or Load Balancer
- High Availability
    - Loosely coupled systems are more fault tolerant
    - High Availability: System still operates but can degenerate
    - Fault tolerance: user does not recognize any impact
- Amazon CloudFormation
    - Infrastructure as Code
    - Declarative programming language(JSON) for deploying AWS resources
    - Templates and stacks to provision resources
    - AMI IDs differ across regions and mappings should be used
- AWS Lambda
    - Fully managed compute service that runs stateless code in response to event or timed trigger
    - Serverless
    - Print statements are outputted in CloudWatch logs
    - RTO – Recovery Time Objective (Time to restore System)
    - RPO – Recovery Point Objective (Time/Data which will be lost by restoring Backup)

<u>Domain 2: Define Performant Architectures (28%)</u>

- Choose performant storage and databases
    - EBS
        - SSD vs. HDD
        - Perfmormant option vs. Standard option
    - Move static content (CSS, JS, etc.) to S3
        - Create a Bucket in Region and upload any number of objects
        - Bucket name becomes subdomain of URL
        - Bucketnames are globally unique
        - Pricing: Storage in GB/month, Transfer out of region, API requests
            - Free: Transfer in to Amazon S3, Transfer out from S3 to CloudFront or in same region
            - S3 Standard vs. S3 Infrequently accessed data
            - S3 Lifecycle policies to move data between S3 options automatically
    - Performant storage on databases
        - Amazon Relational (RDS) vs. DynamoDB vs. Redshift
        - RDS:
            - Complex transactions/queries
            - Medium to high query/write rate
            - No more than a single worker node
            - Read Replicas can be used to share load (Aurora, MariaDB, MySQL, PostgreSQL)
        - DynamoDB
            - Allocates resources based in throughput capacity requirements →Massive read/write rates
            - Sharding
            - Simple GET/PUT requests and queries
- Apply caching
    - Caching in CloudFront → Data is cached in Edge Location close to users
    - Caching with ElastiCache for Database Cache
        - Memcached: Multithreading, Low maintenance, easy horizontal scaling
        - Redis: Support for data structures, Persistence, Atomic operations, Pub/sub messaging, Read replicas/failover, Cluster mode/sharded cluster
    - CloudFront
        - For static and dynamic content
        - S3, EC2, ELB and HTTP origins
        - Protect private content
        - Supports SSL
        - Improve Security with AWS Shield and AWS WAF (Firewall)
- <u>Auto Scaling (horizontal scaling)</u>
    - Vertical Scaling
        - Change in specifications of instances (CPU, memory)
    - Horizontal Scaling
        - Change in number of instances

  - Launches or terminates instances, automatically registers new instances with load balancers
  - Across AZs
  - **Cloud watch launches alarm → triggers auto scale policies → scaling happens**
    - **Load Balancer is needed**
  - Auto Scaling launch configuration specifies EC2 instance size and AMI name
  - Auto scaling group
    - References launch configuration
    - Specifies min, max and desired size
    - May reference ELB
    - Health Check Type
  - Auto scaling policy
    - Specifies how much scale in or out
    - One or more may be attached to Auto scaling group
  - Cloud Watch monitors CPU, Network, Queue size, etc.
    - Cloud Watch Logs for logging
    - Default metrics and custom metrics possible
- Operational Excellence
  - Perform operations with code
  - Annotate documentation
  - Frequent small reversible changes
  - Refine operations procedures frequently
  - Anticipate failure
  - Learn from failures
  - AWS Config: Tracks resources
  - CloudFormation: Converts JSON/YAML into infrastructure
  - Trusted Advisor: Checks for best practices
  - AWS Inspector: Checks for security invulnerabilities
  - VPC Flow Logs: Logs network traffic (Layer 3 + 4 / IP)
  - Cloud Trail: Logs API Calls

Domain 3: Specify Secure Applications and Architectures (24%)

- Determine how to secure application tiers
    - Shared responsibility model
        - OS and above customer responsible, rest AWS
    - Principle of least privileged
        - Persons can perform all activities they nedd and no more
    - Identities AWS IAM
        - Centrally manage users and permissions
        - Create users, groups, roles and policies
        - Users: created within account
        - Roles: Temporary identities used by EC2, Lambda and external users
        - Federation: Users with AD identities with assigned IAM role
        - Web Identity Federation: Users with web identities from Amazon.com or other Open ID provider
        - Define permissions to control which AWS resources users can access
        - Integrates with AD and AWS Directory service
- Determine how to secure Data
- Define the networking infrastructure for a single VPC application
    - Virtual Private Cloud VPC
        - Organization: Subnets
            - Public Subnets:
                - To support inbound/outbound access to public internet
                - Include routing table entry to internet gateway
            - Private Subnets:
                - No routing table entry in internet gateway
                - Not directly accessible from public internet
                - For restricted outbound only internet access use jump box (NAT/proxy/bastion host)
        - Security: Security groups/ access control lists
            - Security Groups:
                - Specify port, protocol, source IP
                - Explicit Allow only
                - Stateful
                - Applied to ()ENIs
                - Associated with single VPC
            - Access control list:
                - Specify port, protocol, source IP
                - Explicit Allow or Deny
                - Stateless
                - Applied to subnets
                - Associated with single VPC
        - Network isolation: Internet gateways/virtual private gateways/NAT gateways
            - Use security groups to control traffic into, out of and between resources
            - Internet gateway: Connect to public internet

            - Virtual private gateway: Connect to VPN
            - AWS Direct Connect: Dedicated pipe
            - VPC peering: Connect to other VPCs
            - NAT gateways: Allow internet traffic from private subnets
        - Traffic direction: Routes
- Securing Data Tier
    - Data in transit
        - SSL over web
        - VPN for IPsec
        - IPsec over AWS direct Connect
        - Import/Export with Snow Family
    - Data at rest
        - S3
            - Private by default, requires credentials for access
            - Access over HTTP/S
            - Audit of access
            - Supports ACL
        - EBS
        - Server side encryption options:
            - Amazon S3-Managed Keys (SSE-S3)
            - KMS-Managed keys (SSE-KMS)
            - Customer-Provided keys (SSE-C)
        - Client side encryption options
            - KMS managed master encryption keys (CSE-KMS)
            - Customer managed master encryption keys (CSE-C)
    - Key management:
        - Key Management Service
            - Customer software-based key management
            - Integrated with many AWS services
            - Use from application
        - AWS CloudHSM
            - Hardware-based key management
            - Use from application
            - FIPS 140-2 compliance

Domain 4: Design Cost-optimized Architectures (18%)

- Pay as you go
- Pay when you reserve
- Pay less when buying bulk (volume discount)
- Cost factors: Compute, storage, and data transfer
- Determine how to design cost-optimized storage
    - Storage Class: Standard, IA, Glacier, etc.
    - Storage amount
    - Number of requests
    - Data Transfer
    - EBS: HDD vs. SDD (Volumes, IOPS, Snapshots, Data transfer)
- Determine how to design cost-optimized compute
    - Clock hours
    - Machine configuration
    - Machine purchase type
    - Number of instances
    - Load balancing
    - Detailed monitoring
    - Auto scaling
    - Elastic IP addresses
    - Operating systems and software packages
    - EC2: reserved instances, spot instances (with hibernate and spot block)
- Serverless Architecture
    - Lambda, S3, DynamoDB, API Gateway
    - CloudFront
        - Use cases:
            - Content static and dynamic
            - Origins: S3, EC2, Elastic Load Balancing, HTTP servers
        - Cost benefits:
            - No cost for data transfer between S3 and CloudFront
            - Can be used to reduce compute workload for EC2

Test Axioms

- Domain 1
  - Single AZ will never be the right answer
  - Using AWS managed services should always be preferred
  - Fault tolerant and high availability are not the same thing
  - Expect that everything will fail at some point and design accordingly
- Domain 2:
  - IAM roles are easier and safer than keys and passwords
  - Monitor metrics across the system
  - Automate responses to metrics where appropriate
  - Provide alerts for anomalous conditions
  - Provide alerts for anomalous conditions
- Domain 3:
  - Lock down root user
  - Security groups only allow, ACLs allow deny
  - Prefer IAM Roles to access keys
- Domain 4:
  - If you know it's going to be on, reserve it
  - Any unused CPU time is waste of money
  - Use the most cost-effective data storage service and class
  - Determine the most cost-effective EC2 pricing model and instance type for each workload