**AWS Cloud Practitioner Essentials**

**MODULE 1: INTRODUCTION TO AMAZON WEB SERVICES**

**Introduction**

Client-server model

Client: something that the person interacts with to make requests

**Cloud Computing**

Cloud computing is the on-demand delivery of IT resources over the internet with pay-as-you-go pricing

Cloud-based Deployment

* Run all parts of the application in the cloud
* Migrate existing applications to the cloud
* Design and build new applications in the cloud
* In a cloud-based deployment model, you can migrate existing applications to the cloud, or you can design and build new applications in the cloud. You can build those applications on low-level infrastructure that requires your IT staff to manage them. Alternatively, you can build them using higher-level services that reduce the management, architecting, and scaling requirements of the core infrastructure.

On-premises Deployment

* Deploy resources by using virtualization and resource management tools
* Increase resource utilization by using application management and virtualization technologies
* On-premises deployment is also known as a *private cloud deployment*. In this model, resources are deployed on premises by using virtualization and resource management tools.

Hybrid Deployment

* Connect cloud-based resources to on-premises infrastructure
* Integrate cloud-based resources with legacy IT applications
* In a hybrid deployment, cloud-based resources are connected to on-premises infrastructure.

Benefits of cloud computing

* Trade upfront expense for variable expense
* Stop spending money to run and maintain data centers
* Stop guessing capacity
* Benefit from massive economies of scale
* Increase speed and agility
* Go global in minutes

**MODUEL 2: COMPUTE IN THE CLOUD**

**Introduction**

EC2 runs on top of physical host machines managed by AWS using virtualization technology. When you spin up an EC2 instance, you aren't necessarily taking an entire host to yourself. Instead, you are sharing the host with multiple other instances, otherwise known as virtual machines. And a hypervisor running on the host machine is responsible for sharing the underlying physical resources between the virtual machines. This idea of sharing underlying hardware is called multitenancy. The hypervisor is responsible for coordinating this multitenancy and it is managed by AWS. The hypervisor is responsible for isolating the virtual machines from each other as they share resources from the host. This means EC2 instances are secure. Even though they may be sharing resources, one EC2 instance is not aware of any other EC2 instances also on that host. They are secure and separate from each other.

**Amazon EC2 Instance Types**

Combinations of CPU, GPU, Memory, Network etc

* General purpose
* Compute optimized
* Memory optimized
* Accelerated computing
* Storage optimized

**Amazon EC2 Pricing**

* On-Demand: ideal for short-term
* Saving Plans: commit to a consistent amount of compute usage for a 1-year or 3-year term
* Reserved Instances: billing discount applied to the use of On-Demand; Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term, and Scheduled Reserved Instances for a 1-year term
* Spot Instances: ideal for workloads with flexible start and end times; use unised Amazon EC2 computing capacity (if capacity is no longer available for Spot Instance increases, the instance is iterrupted)
* Dedicated Hosts: physical servers with Amazon EC2 instance capacity that is fully dedicated to your use

**Scaling Amazon EC2**

Beginning with only the resources you need and designing your architecture to automatically respond to changing demand by scaling out or in

Amazon EC2 Auto Scaling

**Directing traffic with Elastic Load Balancing**

Where to route the requests?

Elastic Load Balancing

* Regional construct
* A load balancer acts as a single point of contact for all incoming web traffic to your Auto Scaling group
* Stop the flow and terminate all existing process. Then finally terminates extra instances.

**Messaging and queuing**

Message Queue Buffer between two applications

* Monolithic x Miscroservices w. integrations (SNS, SQS)
* Tightly coupled architecture: if a single component fails or changes, it causes issues for other components or even the whole system
* Payload: data contained within a message
* Amazon SQS: send, store, receive message between software components at any volume
* Amazon SQS queues: where messages are placed until they are processed
* Amazon SNS: send out messages to services and also send out notifications to end users (pub/sub model), using SNS topics. Subscribers can be web servers, email address, AWS Lambda functions etc

**Additional compute services**

AWS Serverless computing: you cannot see or access the underlying infrastructure or instances that are hosting the application

* AWS Lambda: upload the code and wait for some trigger, the code is runned in Lambda function automatically
* Amazon ECS, Amazon EKS: docker container orchestration tools with host in EC2 or AWS Fargate (serverless docker management platform)

**MODULE 3: GLOBAL INFRASTRUCTURE AND RELIABILITY**

**AWS global infrastructure**

Regions > Data centers

Factors to choose regions: compliance, proximity, feature availability, pricing

Availability Zones: multiple data centers in one region

**Edge locations**

Cache of copy of data / when you have data in LATAM and customers are in Asia but you cant move datacenter to Asia

* Amazon CloudFront: CDN; Edge locations; global content delivery service
* Amazon Route 53: route customers request for low latency
* AWS Outposts: service that enables you to run infrastructure in a hybrid cloud approach.

**How to provision AWS resources**

* By API
* AWS Management Console (Browser), CLI, SDKs etc
* AWS Elastic Beanstalk: provide code and configuration setting
* AWS CloudFormation: Infrastructure as code tool used to define a wide variety of AWS resources (templates ex. JSON) (Storage, Database, Machine Learning, Analytics)

**MODULE 4: NETWORKING**

**Connectivity to AWS**

Gateway or Virtual Private Gateway or AWS Direct Connect -> VPC

Subnet: chunks of Ips

* Amazon VPC: provide an isolated section of the AWS Cloud / control access to the gateways
* Subnet: section of VPC thar can contain resources such as ED2

**Subnets and network access control lists**

* Packet is checked by Network access control list (Network ACL) if the packed is allowed to go to subnet
* Multiple EC22 can be in same subnet -> Security Group for each instance
* Security Group (Stateful) x VPC - Network ACL (Stateless) //remembers

**Global networking**

* Amazon Route 53 = DNS (can define routing policies)

**MODULE 5: STORAGE AND DATABASES**

**Instance stores and Amazon Elastic Block Store (Amazon EBS)**

* Block-level storage: saved as block in disk: db, software, files
* Instance store volumes: attached to instance (stop instance = delete volume)
* EBS: block-level storage volumes that can be used with EC2; persistent; snapshot

**Amazon Simple Storage Service (Amazon S3)**

Data stored as objects, buckets, versions; web enabled

* S3 Standard: high availiability
* S3 Standard-Infrequent Access
* S3 One Zone-Infrequent Access
* S3 Intelligent-Tiering: ideal for unknown or changing access patterns
* S3 Glacier: min to hour
* S3 Glacier Deep Archive: within 12 hours

**Amazon Elastic File System (Amazon EFS)**

|  |  |
| --- | --- |
| EBS | EFS |
| Single Availability Zones | Multiple Availability Zones |
| To attach an EC2, must be in same Availability Zone | Anywhere |
| Cannot scale | Can scale |

**Amazon Relational Database Service (Amazon RDS)**

* Lift-and-shift migration
* RDS comes with management services
* Amazon Aurora: enterprise-class relational db: mysql, postgre +) services

**Amazon DynamoDB**

* Serverless db, automatic scaling
* NRDB, NoSQL db
* RDS: full control / DynamoDB: more easy

**Amazon Redshift**

* Data warehouses can be used for big data analytics
* Used with API

**AWS Database Migration Service**

Amazon DMS

* Source db remains fully operational during the migration
* The source and target db don’t have to be of the same type
* Database consolidation & replication

**Additional database services**

* Amazon DocumentDB: document dv service that supports MongoDB workloads
* Amazon Neptune: graph database service (You can use Amazon Neptune to build and run applications that work with highly connected datasets, such as recommendation engines, fraud detection, and knowledge graphs)
* Amazon Quantum Ledger Database (Amazon QLDB) is a ledger database service. You can use Amazon QLDB to review a complete history of all the changes that have been made to your application data.
* - Amazon Managed Blockchain is a service that you can use to create and manage blockchain networks with open-source frameworks. Blockchain is a distributed ledger system that lets multiple parties run transactions and share data without a central authority.
* Amazon ElastiCache is a service that adds caching layers on top of your databases to help improve the read times of common requests. It supports two types of data stores: Redis and Memcached.
* Amazon DynamoDB Accelerator (DAX) is an in-memory cache for DynamoDB. It helps improve response times from single-digit milliseconds to microseconds.

**MODULE 6: SECURITY**

**Shared responsibility model**

* AWS: Physical, network, hypervisor
* Customers: security in the cloud / AWS: security of the cloud

**User permissions and access**

* AWS account root user
* IAM – Identity and Access Management
  + Default: no permissions
* IAM groups / policies
* Roles

**AWS Organizations**

**Compliance**

* Adequar a legislacao tipo LGPD
* AWS Artifact: provides on-demand access to AWS security and compliance reports and select online agreements
  + AWS Artifact Agreements, AWS Artifact Reports

**Denial-of-service attacks (DDoS)**

* AWS se vira (AWS Shield) e vc se vira tb

**Additional security services**

* AWS Key Management Service (KMS): cryptographic keys
* AWS WAF: monitor network request
* Amazon Inspector: verify security issues
* GuardDuty: intelligent threat detection for infrastructure and resources

**MODULE 7: MONITORING AND ANALYTICS**

**Amazon CloudWatch**

* Visibilities in dashboard in real time
* CloudWatch Alarm

**AWS CloudTrail**

* Records all API calls for account in S3

**AWS Trusted Advisor**

* Automated advisor – cost optimization, performance, security, fault tolerance, service limits

**MODULE 8: PRICING AND SUPPORT**

**AWS Free Tier**

* Always free, 12 months, trials

**AWS pricing concepts**

* Pay for what you use
* Pay less when you reserve
* Pay less with volume-based discounts when you use more
* AWS Pricing Calculator

**Billing dashboard**

**Consolidated billing**

Organization

**AWS Budgets**

* Create budgets to plan your service usage, service costs, and instance reservations
* Can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount

**AWS Cost Explorer**

* Visualize, understand, and manage the AWS costs and usage over time

**AWS Support plans**

* Basic, Developer, Business (all AWS Trusted Advisor), Enterprise (has TAM)

**AWS Marketplace**

* Software listing s from independent software vendors

**MODULE 9: MIGRATION AND INNOVATION**

**AWS Cloud Adoption Framework (AWS CAF)**

* Cloud Adoption Framework with multiple perspectives: business, people governance, platform, security, operations

**Migration strategies**

* Rehosting,
* Replatforming: moving to a different product
* Refactoring or re-architecturing
* Repurchasing: traditional license to a software-as-a-service model
* Retaining: keeping applications that are critical for the business in the source environment
* Retiring: removing applications that are no longer needed

**AWS Snow Family**

* Collection of physical devices that help to physically transport up to exabytes of data into and out of AWS
* AWS Snowcone (8TB) < AWS Snowball (80TB) < AWS Snowmobile (100PB)

**Innovation with AWS**

* SageMaker: quickly build, train, and deploy machine learning models
* Textract: machine learning service that automatically extracts text and data from scanned documents
* Lex: build conversational interfaces using voice and text
* DeepRacer: autonomous 1/18 scale race car that you can use to test reinforcement learning models

**MODULE 10: THE CLOUD JOURNEY**

**The AWS Well-Architected Framework**

* Operational excellence
* Security
* Reliability
* Performance efficiency
* Cost optimization

Benefits of the AWS Cloud