AWS Cloud.

Date : ...../...../.....

Client Server Modal → many clients using services from centralized server.

cloud Computing → (1) Computing over internet
(2) Pay as you go model.

Virtual Server → Amazon Elastic cloud Compute. (EC2)

Pay for you mud → Pay as you go model.

Deployment Models - (1) cloud based
(2) on premises
(3) Hybrid.

Mutitenancy → Sharing underlying hardware b/w virtual Machine's (Hypervisor's are used for this). coordination, isolation of machine's from each other).

Vertical Scaling → Increasing power.

Load Balancing → Routes req to multiple Server/ instances evenly to properly distribute traffic.

Date : ...../...../.....

* Elastic Load Balancing → AWS Load balancer.
- Automatically distributes incoming traffic across multiple resources.
- Load balancer is single point of contact for incoming traffic.
- Elastic Load balancing & AWS EC2 Autoscaling are seperate Services. (they work together).

* Tightly Coupled Architecture → If application's communicate directly.
If Single Component fails all System goes down.

* Loosely coupled Architecture → Single failure won't Cause Cascading failures.

Amazon SQS, SNS (Queue, Notification)

↳ Send
Store
receive

payload → data contained within a message.

SQS Queue → Msgs are placed here until they are processed.

SNS → Send msgs to Servies & end users.
(Message channel) publish/Subscribe (pub/sub) model.

Date : ...../...../.....

* Monolithic Applications ⇒ - Application with tightly Coupled Components.
Components can be database, Servers, UI, logic etc.

- Single point of failure.

* Microservice Approach ⇒

- Helps maintain Availability.
- Components are loosly Coupled.

* Serverless ⇒ You cannot see or access the underlying infastructure.

* AWS Lambda ⇒ Serverless Compute. (Serverless)
- Add code to lambda function. and when we call/trigger code it run's in auto managed environment.
- Suitable for quick processing
- Not for dup learning.

Date : ...../...../.....

* AWS Container Service (ECS).
* ~~AWS~~ Amazon Elastic Kubernetes Service (EKS).

This are Container Orchestration tool's
↑
Docker

Container ⇒ Package for your code.
Cluster ⇒ Multiple EC2 instances.

* AWS Fargate ⇒ Serverless Compute Platform for ECS/EKS.

* Interacting with AWS Services ⇒
  → AWS Management Console. (browser based).
  → AWS CLI
  → AWS SDK's
  → various other tools. (AWS Elastic BeanStalk.)
  (AWS Cloudformation)
  ↓
  Infastructure as a code

Region → Availability Zones.

Provide code & config settings Beanstalk will deploy resources necessary.

* Cloudfront ⇒ CDN, edge location's.

* Amazon Route 53 ⇒ DNS.
* AWS Outposts ⇒ Install DataCenter at your location.
  isolated with your own building.

Date : ...../...../.....

* Edge location ⇒ It is a site that Amazon Cloudfront uses to store cached copies of your content closer to your Customer's for faster delivery.

* Amazon Virtual Private cloud (VPC)

→ public/private grouping of resources (App's, databases) are called Subnets.

→ Allows to define private IP Range.

→ Subnets ⇒ chunks of IP Address that allows to group resources. (Section of VPC)

→ virtual private gateway ⇒ To acess private resources in VPC.

→ AWS Direct Connect ⇒ private data Connection b/w data center & a VPC.

→ Internet Gateway ⇒ To allow public traffic from internet to acess your VPC.

→ Network Access control list ⇒ only allows valid packets into private Subnets. (Virtual firwall)

→ Security Group ⇒ Stateful (has a memory)
Network ACL ⇒ Stateless, checks both in/out traffic.

Date : ...../...../.....

*AWS Route 53 → - DNS of AWS

- Translates website names to IP.
- Routing Policies
  - ↳ latency based
  - ↳ Geo location
  - ↳ Geo proximity routing
  - ↳ weighted round robin.
- Register Domain name

* AWS Cloud front → - CDN

- Serves data faster based on geographic loc" & edge location's

Date : ...../...../.....

* Storage & Databases ⇒

* Amazon Elastic Block Store (EBS) ⇒ Virtual hard drives called (Block Storage).

volumes.

- Complex read/write - Solutions. change solution.

- Snapshots ⇒ Incremental backup.

- Instance store ⇒ Provides temporary block level Storage for an Amazon EC2.

It has Same lifespan as EC2 instance (i.e when Stoped or terminated data willbeloss)

* AWS S3 (Simple Storage Service) ⇒ Store and retrieve unlimited amount of data.

e.g. When edits are done need to reupload file update not Supported.

- data → objects (data, metadata, key)
- objects are stored in buckets.
- max size of object 5TB
- Static website hoisting.

- S3 standard, S3 Standard IA (Infrequent Acess), S3 Glacier Flexible Retrival

Date : ...../...../.....

* Amazon Elastic File System (EFS) ⇒

- Shared file system.
- Multiple instances can access the data in EFS.
- when large no. of services & resources need access the same data at same time.
- EFS grow's & shrink's automatically.

* Amazon Relational Database Service (RDS) ⇒

- Automated Paching
- Backups
- Redundancy
- Failover
- Disaster recovery.

(Engine) like SQL, Oracle etc

- Amazon Aurora
  - ↳ Postgres, SQL
  - ↳ data replication
  - ↳ continious S3 backup
  - ↳ upto 15 replicas.
  - ↳ faster than SQL
  - ↳ less ip/op costs

- This db uses SQL to store and query data.

* Amazon Dynamo DB ⇒ "NO SQL Database".

- Data organized into key-value pairs.
- Dynamo DB is serverless
- Auto Scaling

Date : ...../...../.....

* Amazon Redshift ⇒ - Data warehouse
- Handle large datasets.
- used for big data Analytics.
- collects data from diff. sources & understand trend & relationship across your data.

* Amazon Database Migration Service (DMS) ⇒

- Homogenous Databases ⇒ Same type
e.g. MySQL → AWS RDS MySQL

- Hetrogenous Migration ⇒ Diff types

Step ① → Convert Schema, datatypes, code ~~to~~ using AWS Schema Conversion tool.

Step ② → use DMS for Migration.

Other usecases of DMS ⇒ - Development & test database migrⁿ.
- Database Consolidation.
- Continious db replication.

Database Consolidation ⇒ Combining multiple db into single db.

Date : ...../...../.....

* Additional Database Services ⇒

1. Amazon Document DB (MongoDB Compatible).
2. Amazon Neptune ⇒ Graph database.
3. Amazon Managed Blockchain.
4. Amazon Quantum ledger Database (AWS QLDB).

- Database Accelerator ⇒ Amazon Elasticache.
  for Dynamo DB ⇒ DAX (Dynamo DB Accelerator)

* SECURITY ⇒

Shared responsibility Principle

- AWS & owner are responsible for keeping resources secure.

- Customer responsibility ("Security in cloud")
- AWS (security of the cloud)

Customer's → - Customer data, Platform, Applications, Identity Access Management (IAM), OS, Network & firewall Config.
Client side & Server Sider Encryption, Network traffic

Date : ...../...../.....

AWS → Software, Compute, Storage, database, Networking, Hardware, Regions, Availability Zones, Edge locations.

* User Permissions and Access →

- Root user ⇒ - AWS Account owner.
  - Access & control any services
  - Multifactor Auth (MFA) (for security)

- AWS Identity and access Management (AWS IAM) →
  - ↳ create IAM user.
  - ↳ we have to give permission for this user
  - ↳ give only required acces (Least Privilage Principal)

- IAM Groups ⇒ Grant Permissions in groups.
- Policies ⇒ Set of permission which are attached to user's.

- Roles ⇒ - AWS Identifis.
  - Associated permissions.
  - No username & password.

Date : ...../...../.....

* AWS organizations ⇒ - A central location to manage multiple AWS Accounts.

- Centrally control permissions using service control policies (SCP's).

- Organizational Units (OU's) to make it easier to manage accounts with Similar business or Security requirements.

  ↳ Isolate workload

* Compliance →

AWS Artifacts → - Access to Compliance reports by third party Services / auditor's.

- on demand access to AWS Security & Compliance reports.
- Sections → (1) AWS Artifact Agreements.
  (2) AWS Artifact Reports.

* DDoS ⇒ (Distributed Denial of Service)
  - Make application unavailable to user's.
  - Shutdown system by overwhelm load. / unbearable load.

Date : ...../...../.....

- Security groups are solution to this problem. (UDP) Flood.
- Load Balancer takes care of Slowris attack

- AWS Shield with AWS WAF ⇒ Recognize threats, filters traffic.
  - Two levels of Protection → (1) Standard
    - no cost, common attacks.
  - (2) Advanced → Paid, detailed attack diagnostics.

- Encryption ⇒ Securing a message or data in a way that only authorized parties can acces it.

- Amazon Inspecter ⇒ Automated Security Investigation/ assesment.

- Amazon Guarduty ⇒ Analyses Streams of metadata.

- AWS Key Management Service (KMS) ⇒ Enables you to perform enc. operations through the use of Cryptographic Keys.

- AWS WAF (Web Application Firewall) ⇒ monitor network req.
- works with cloudfront & load balancer
- It uses web acces control list (ACL) to block or allow traffic.

Date : ...../...../.....

* Monitoring & Analytics ⇒

- Monitoring ⇒ observing systems, collecting metrics and then using data to make decision's.

- Amazon cloudwatch ⇒ - Monitor AWS Infastructure in realtime.
  - Gain visibility for apps.
  - cloudwatch alarm - generates alert after hitting certain specified limits.
  - acess metrics from central location.
  - Can be integrated with SNS.
  - Dashboard

- AWS CloudTrail ⇒ - API Auditing Tool.
  - Every req get logged in engine with req. details.
  - Saves logs in S3 buckets.
  - History of user activity.

- AWS Trusted Advisor → - Inspects AWS environment and provides realtime recommendation's.

Date : ...../...../.....

- Evaluats based on five Pillars → ① Cost optimization
② Performance
③ Security
④ Fault Tolerance
⑤ Service limits.

* Pricing and Support ⇒

* AWS Free Tier

* Consolidated billing → Manage multiple accounts from central location.

* AWS Budgets → Set custom budgets & when budget exceeds we will get alert.

* AWS Support → - Basic Support is free.
- AWS Developer (Email Support)
- AWS Business Support (phone access)

* AWS Marketplace → one click deployment, use third party services.

Date : ...../...../.....

* Migration and Innovation →

- AWS cloud Adoption framework → Helps organization migrate quickly to cloud.

- Six core perspectives of cloud adoption framework.
  1. Business Perspective (Roles → Business Mng., Fin, Budget, Strategy)
  2. People (HR, Staffing, People managers)
  3. Governance (CIO, Program manager, Business Anly, Portfolio Mng)
  4. Platform (CTO, IT Managers, Solution Architect).
  5. Security (CISO, IT Security Managers/Analysts)
  6. operations (IT operations/Support Managers)

Technical

- Six R's of Migration → choose best option based on time and cost.

1. Rehosting → "Lift-and-shift" involves moving app's without changes.

2. Re Platforming → "lift-tinker-shift"
   - making cloud optimizations without changing core architecture.

Piyush. S

Date : ...../...../.....

③ Refactoring / Re-architecturing → - Reimagining how app is architectured & developed.
- Strong business need to add features, scale.

④ Repurchasing → Moving from traditional license to SaaS Model.

⑤ Retaining → - Keeping apps that are critical for the busines in Source environment.
- Might include major refactoring before migrn.

⑥ Retiring → Removing apps that are no longer needed.

- AWS Snowcone ⇒ Small & secure edge computing and data transfer device.

- AWS Snowball ⇒ ① Snowball edge storage optimized - for large scale data migrn.

② Snowball edge Compute optimized ⇒ Provides powerful Computing resources for use cases such as machine learning, etc.

Date : ...../...../.....

- AWS Snowmobile → It is an exabyte scale data transfer service used to move large amount of data to AWS.

* AWS Well Architected Framework →

- Based on Six pillar's →

1. operational excellence
2. Security
3. Reliability
4. Performance efficiency
5. Cost optimization
6. Sustainability