AWS Core Services and Concepts

Here is a comprehensive roadmap drawing on the information provided in the sources, designed to be neat and easy to understand:

AWS Core Services and Concepts Roadmap

This roadmap outlines key AWS services and fundamental concepts, including EC2, Load Balancing, IAM, S3, networking, security, databases, analytics, machine learning, and serverless computing.

I. EC2 (Elastic Compute Cloud)

- **EC2 Instances and Instance Storage**
    - **EC2 Instance Types**
        - **General Purpose**: Balance of compute, memory, and networking.
        - **Compute Optimized**: High-performance, CPU-bound applications (e.g., batch processing, media transcoding, high-performance web servers, scientific modelling, machine learning, dedicated gaming servers).
        - **Storage Optimized**: High, low-latency access to large data (e.g., OLTP systems, relational and NoSQL databases, in-memory DB cache, data warehousing, distributed file systems).
        - **Accelerated Computing**: Utilizes hardware accelerators like GPUs (e.g., graphics processing, machine learning).
        - **Memory Optimized**: Fast, in-memory processing of large data sets (e.g., high-performance databases, business intelligence, real-time big data analytics).
    - **EC2 Pricing Models**
        - **On-Demand** – Pay per hour or second with no long-term commitments.
        - **Reserved Instances** – Commit to a 1- or 3-year term in exchange for significant hourly rate discounts (up to ~75%).
        - **Savings Plans** – Flexible commitment to spend (e.g. $10/hour) for 1 or 3 years on any compute usage (EC2, Fargate, Lambda) with similar discounts to Reserved Instances.
        - **Spot Instances** – Bid on unused EC2 capacity at discounts of up to 90%—but AWS can reclaim the instance with a two-minute warning.
    - **Spot Fleet**: A set of Spot instances + On-demand instances.
    - **Placement Groups**: Influence the placement of interdependent EC2 instances.
        - **Cluster Placement Groups**: Instances in close proximity within a single Availability Zone for low-latency network performance (HPC applications).

        - **Spread Placement Groups**: Instances across distinct underlying hardware to reduce correlated failures (critical applications requiring instance isolation).
        - **Partition Placement Groups**: Instances across multiple partitions within an Availability Zone for isolation (large distributed and replicated workloads like Hadoop, Cassandra, Kafka).
    - **EC2 Hibernate**: Preserves RAM state into EBS.
    - **EC2 Instance Storage**
        - **EBS (Elastic Block Store)**: Persistent, network-attached block storage. Data persists when instance is stopped or terminated; experiences network latency.
            - **EBS Volume Types**: General Purpose SSD (gp3 & gp2), Provisioned IOPS SSD (io1/io2, io2 Block Storage), Hard Disk Drive (Throughput Optimized HDD (stl), Cold HDD (scl)).
            - **EBS Multi-attach**: Supported by io1/io2 family.
        - **EC2 Instance Store**: Temporary, block-level storage physically attached to the host. Offers very low latency and high I/O performance (local cache). Data is lost when instance is stopped, terminated, or fails.
        - **Amazon EFS (Elastic File System)**: Managed NFS mountable on many EC2 Instances across multiple AZs (e.g., content management, web serving, data sharing); Linux compatible only.
    - **Launch Templates**: Specifies settings for building EC2 instances, replacing the need to reconfigure via the EC2 wizard.
    - **Launch Configurations**: An outdated instance configuration template for Auto Scaling Groups.
    - **User Data**: Can be included in launch templates or configurations.
    - **Networking Configurations**: Can be included in templates.

II. Elastic Load Balancers (ELB)

- **Types of Load Balancers**
    - **Classic Load Balancer (CLB)**: Basic load balancing at transport (TCP/SSL) and application (HTTP/HTTPS) layers for older applications.
    - **Application Load Balancer (ALB)**: Operates at the application layer (HTTP/HTTPS) with advanced routing based on content (e.g., host-based, path-based routing). Ideal for microservices, container-based, and modern web applications.
        - **Listener**: Checks for client connection requests using configured protocol and port.
        - **Target Group**: Routes requests to registered targets (e.g., EC2 instances).
        - **String/Parameter Routing**.

        - **HTTPS Listener**: Requires SSL/TLS certificate deployment on the load balancer.
        - **Deregistration Delay**: Keeps existing connections open for de-registered/unhealthy instances.
        - **Sticky Session**: Redirects client to the same instance.
        - **Health Checks**: Used to route traffic to healthy instances/targets.
    - **Network Load Balancer (NLB)**: Operates at the transport layer (TCP/UDP/TLS) for extreme performance and low latency. Handles volatile traffic patterns.
    - **Gateway Load Balancer (GWLB)**: Operates at the network layer (Layer 3) to deploy, scale, and manage third-party network virtual appliances (e.g., firewalls, IDS/IPS).

III. High Availability & Scalability

- **Vertical Scaling**: Increases resources of a single machine (e.g., faster CPUs, memory, storage); easier but may involve downtime.
- **Horizontal Scaling**: Increases capacity by adding more computers and distributing load; enables parallel execution, improves performance, scalability, and reliability.
- **Auto Scaling Group (ASG)**: Automatically adjusts the number of EC2 instances based on application needs to maintain availability and capacity.
    - **Instance Warm-Up**: Time for new instances to become fully operational before handling requests during scale-out.
    - **Instance Cool-down**: Time delay after an instance starts to prevent alarms from adding more instances prematurely.
- **Warm Standby** – Reduced-capacity active environment in another region.

🔄 **What is Warm Standby in AWS?**

Warm Standby is a **disaster recovery (DR) strategy** where a **scaled-down, functional version** of your production environment is **always running** in a different AWS region or account. In the event of a failure, it can be **quickly scaled up** to full capacity.

🧠 **Analogy:**

Think of it like having a **spare tire partially inflated** — it's not the same as your main tire, but it's ready to go with a bit of inflation. You won't be stranded, and you can be up and running much faster than starting from scratch.

✅ **Example in AWS:**

Assume your production environment includes:

- 10 EC2 instances
- RDS database
- Application Load Balancer
- Auto Scaling groups

In a **Warm Standby** setup:

- You run a **smaller version** (e.g., 2 EC2 instances) in a **different region**.
- The **RDS instance is fully synced** and running.
- **Route 53** is configured for DNS failover.
- If a disaster occurs, you **scale out** the EC2 instances using Auto Scaling, and traffic is routed via Route 53 to the standby environment.

⚙️ **Key AWS Services Used:**

- **Amazon EC2** – minimal app servers always running
- **Amazon RDS / DynamoDB** – continuously replicated data
- **Amazon S3** – backup storage
- **Auto Scaling** – to scale up resources during failover
- **Route 53** – for health checks and DNS routing

📈 **Pros:**

- **Faster recovery** than Pilot Light or Backup & Restore
- Environment is already running and tested
- Less effort to activate in a disaster
- Ideal for applications requiring **low RTO (Recovery Time Objective)**

📉 **Cons:**

- **Higher cost** than Pilot Light — you pay for running infrastructure (even at reduced capacity)
- More maintenance needed to keep the standby environment updated

---

🆚 Warm Standby vs. Pilot Light:

Feature	Warm Standby	Pilot Light
Running Environment	Partial & functional	Core components only (e.g., DB)
RTO	Faster (minutes)	Slower (tens of minutes to hours)
Cost	Moderate	Low
Use Case	Low downtime apps	Cost-sensitive, less critical apps

---

- **Pilot Light** – Minimal core services running in another region, with scale-up templates ready.

In AWS, the **Pilot Light** strategy is a **disaster recovery (DR) approach** used to **minimize downtime and data loss** while keeping infrastructure costs low.

🔥 What is "Pilot Light" in AWS?

Definition:
The **Pilot Light** is a **minimal version of a working environment** that's always running in the cloud. It includes the **core components** of your infrastructure (like databases, AMIs, or configuration) needed to restore full functionality **quickly** in the event of a disaster.

🧠 Analogy:

Think of it like the **pilot light in a gas heater** — it's a small, continuously burning flame that can **quickly ignite** the full flame when needed.
Similarly, in AWS, the pilot light setup keeps only **essential services "warm" (ready to go)**, so you can rapidly scale up everything else if your primary environment goes down.

✅ Example in AWS:

Let's say your production environment runs:

- EC2 for web/app servers
- RDS for database
- S3 for backups
- Auto Scaling & Load Balancers

In a **pilot light DR strategy**, you would:

- Keep the **RDS database and backups** always running or frequently updated.
- Store EC2 **AMI snapshots** and **infrastructure as code (IaC)** templates like CloudFormation or Terraform.

- During a disaster, spin up EC2 instances from AMIs, attach them to the existing RDS, and route traffic through Route 53.

⚙️ **Key AWS Services Used:**

- **Amazon RDS / DynamoDB** – for always-available data layer
- **Amazon S3** – for AMIs, backups, and static assets
- **AWS CloudFormation / Terraform** – to launch infrastructure quickly
- **Amazon EC2** – for on-demand server instances
- **Route 53** – for DNS failover

📈 **Pros:**

- **Cost-effective**: You don't pay for idle full-scale infrastructure.
- **Fast recovery**: You can bring everything online quickly when needed.
- **Good for critical apps** that don't need immediate failover (within minutes to hours is okay).

❗ **When NOT to Use:**

- Not ideal for **mission-critical apps** requiring **zero downtime or instant failover** — for that, use **multi-site active/active** or **warm standby**.

- **Backup and Restore** – Snapshots/AMI-based recovery with defined RTO/RPO.
    - **Scaling Options**
        - **Manual Scaling**: You specify the exact number of instances.
        - **Simple Scaling**: Actions based on CloudWatch alarms (e.g., add instances when CPU utilization breaches a threshold).
        - **Step Scaling**: Defines scaling adjustments based on the size of the alarm breach.
        - **Target Tracking Scaling**: Defines a target value for a metric (e.g., average CPU utilization of 50%), and ASG adjusts instances to maintain it.
        - **Scheduled Scaling**: Scales ASG based on a predefined schedule.
    - **ASG + SQS Integration**: EC2 instances can scale based on the number of messages in an SQS queue.

IV. IAM (Identity and Access Management)

- **Core Components of AWS IAM**
    - **Users**: Represent physical people or applications interacting with AWS.
    - **Groups**: Collections of IAM users assigned permissions collectively, simplifying management.
    - **Roles**: Provide permissions to AWS services and applications, assumed by trusted entities without specific credentials.
    - **Policies**: Define permissions (allowed/denied actions on resources) in JSON format. IAM enforces the principle of least privilege.
- **Securing AWS Root Account**: Enable MFA, create an admin group, create user accounts, and add users to the admin group.
- **IAM Federation**: Combines existing user accounts with AWS (e.g., PC login credentials with SAML standard via Active Directory).
- **AWS Shared Responsibility Model**: AWS is responsible for security *of* the cloud (global infrastructure); customers are responsible for security *in* the cloud (applications, data, configurations, IAM policies).
- **IAM Policy Evaluation Logic** – How AWS decides allow vs. deny.
- **AWS Single Sign-On (SSO)** – Centralized user portal for multiple AWS accounts and business applications.
- **AWS Secrets Manager** – Securely store, rotate, and audit database credentials and API keys.
- **AWS Systems Manager Parameter Store** – Hierarchical storage for config data and secrets, with optional KMS encryption.

V. S3 (Simple Storage Service)

- **Features**: Object-based storage, unlimited total data volume and objects, objects up to 5TB, stores files in buckets (key-value pair), data spread across multiple AZs for availability/durability, Strong Read-After-Write Consistency.
- **Working with S3**: Universal Namespace (globally unique bucket names), S3 URLs, HTTP 200 status code on successful upload.
- **Characteristics**: Tiered Storage, Lifecycle Management, Versioning.
- **Securing Your Data**: Server-Side Encryption, Access Control Lists (ACLs), Bucket Policies. Buckets are private by default; public access requires explicit configuration via bucket policies or object ACLs.
- **Hosting Static Websites**: S3 can host static websites, scaling automatically to meet demands, but not dynamic websites requiring database connections.
- **Versioning**: Enabled at bucket level; unversioned files get Version ID 'null'.

- **Replication**:
    - **CRR (Cross Region Replication)**: For compliance, low-latency access, and replication across accounts.
    - **SRR (Same Region Replication)**: Live replication between production and test accounts.
    - Both require versioning enabled on source and destination buckets. Existing objects can be replicated using S3 Batch Replication.
- **S3 Storage Classes**: Optimized for different access patterns and costs.
    - **S3 Standard**: For frequently accessed data, high durability, availability, performance (e.g., dynamic websites, content distribution, data analytics).
    - **S3 Intelligent-Tiering**: Automatically moves data between frequent and infrequent access tiers based on changing access patterns, optimizing costs.
    - **S3 Standard-Infrequent Access (S3 Standard-IA)**: For less frequently accessed data requiring rapid access, lower storage costs with retrieval fees (e.g., disaster recovery, backups).
    - **S3 One Zone-Infrequent Access (S3 One Zone-IA)**: Stores data in a single Availability Zone for lower costs but lower availability and durability (e.g., secondary backup copies, easily reproducible data).
    - **S3 Glacier Instant Retrieval**: Low-cost archive storage with millisecond retrieval, suitable for data accessed once per quarter (minimum 90-day storage duration).
    - **S3 Glacier Flexible Retrieval** (formerly S3 Glacier): Very low-cost archive storage for rarely accessed data with retrieval times of minutes to hours (e.g., backups, long-term archiving, minimum 90-day storage duration).
    - **S3 Glacier Deep Archive**: Lowest-cost storage class for long-term retention with retrieval times of hours to days (e.g., regulatory compliance, digital preservation, minimum 180-day storage duration).
- **S3 Lifecycle Rules**: Automate object transitions between storage classes for cost-effectiveness.
- **S3 Analytics**: Provides recommendations for S3 Lifecycle Rules by analyzing optimal object movement between tiers.
- **S3 Event Notification**: Notifies about S3 bucket events (e.g., object creation, removal, restore); integrates with SNS, SQS, Lambda.
- **S3 Performance**:
    - **Multi-Part Upload**: Recommended for files >100 MB, mandatory for files >5 GB.
    - **S3 Transfer Acceleration**: Increases transfer speed by utilizing Edge Locations.
- **S3 Select**: Retrieves a subset of data from S3, supporting .csv format.

- **S3 Batch Operations**: Performs operations on existing S3 objects with a single request (e.g., modify metadata, copy objects, encrypt un-encrypted objects, restore from Glacier).
- **S3 Encryption**:
    - **Server-Side Encryption (SSE)**: Data encrypted after being received by server, decrypted before sending.
        - **SSE-S3 (with Amazon S3 Managed Key)**: Keys managed by AWS; AES-256 encryption; default for new buckets/objects.
        - **SSE-KMS (with KMS Key stored in AWS KMS)**: Keys managed by AWS KMS.
        - **SSE-C (with Customer Provided Keys)**: Keys fully managed by customer outside AWS; requires HTTPS and CLI.
    - **Client-Side Encryption**: Data encrypted by client, never decrypted by server, decrypted by receiving client.
- **CORS (Cross-Origin Resource Sharing)**: Defines how client web applications in one domain interact with resources in another domain.
- **S3 Access Logs**: Logs all access to S3 buckets (authorized or denied) for audit purposes; target bucket must be in the same region.
- **S3 Pre-Signed URL**: Allows temporary, specific access to a file.
- **S3 Glacier Vault Lock**: Adopts WORM (Write Once Read Many) model; policies can be locked for future edits.
- **S3 Object Lock**: Adopts WORM; blocks object version deletion for a specific time.
    - **Retention Mode**: **Compliance** (object version cannot be deleted by any user, including root; retention period cannot be changed), **Governance** (only specific users can overwrite/delete or alter).
    - **Retention Period**: Protects object for fixed period, can be extended.
    - **Legal Hold**: Protects object indefinitely, independent of retention period; can be freely placed and removed.
- **S3 Access Points**: Create access policies for specific prefixes within S3 buckets; require VPC Endpoints for access.
- **S3 Object Lambda**: Changes objects before retrieval (e.g., redacting sensitive info for reports).

VI. AWS CloudFront & Global Accelerator

- **CloudFront (Content Delivery Network - CDN)**: Improves read performance by caching content at Edge Locations (216 PoPs globally).
    - **CloudFront Origin**: S3 bucket (distributing/caching files, enhanced security with OAC), Custom Origin (HTTP) like ALB, EC2, S3 static website, or any HTTP backend.
    - **CloudFront Geo Restriction**: Allows or blocks content access for specific countries.

    - **CloudFront Price Classes**: All (best performance), 200 (most regions excluding expensive), 100 (least expensive regions).
    - **CloudFront Invalidation**: Refreshes cached content at Edge Locations to access the latest version from origin.
- **Global Accelerator**: Lowers latency for global users by routing traffic through AWS network using Anycast IP (sends traffic to closest edge location). Performs health checks and re-routes traffic if a server fails.

VII. AWS Storage Extras

- **AWS Snow Family**: Secure, portable devices for edge computing and data migration.
    - **Data Migration**:
        - **Snowcone & Snowcone SSD**: 8TB HDD / 14TB SSD devices for edge computing and data transfer; can be sent back offline or connect to DataSync.
        - **Snowball Edge**: Physical data transport solution for TBs/PBs of data; alternative to network transfer; pay per data transfer.
            - **Storage Optimized**: 80 TB HDD.
            - **Compute Optimized**: 42 TB SSD or 28TB NVMe.
            - Use cases: large cloud migration, disaster recovery.
        - **Snowmobile**: Transfers exabytes of data (100 PB capacity per unit); for transfers >10 PB; high security.
    - **Edge Computing**: Snowcone/Snowcone SSD (2 CPUs, 4GB RAM), Snowball Edge Compute Optimized (104 vCPUs, 416 GiB RAM), Snowball Edge Storage Optimized (40 vCPUs, 80 GiB RAM); can run EC2 Instances & Lambda functions.
    - **AWS OpsHub**: GUI to manage Snow devices.
    - **Glacier Integration**: Data cannot be transferred directly to Glacier, only to S3 then to Glacier via Lifecycle Policy.
- **Amazon FSx**: Fully managed service to launch third-party high-performance file systems on AWS.
    - **FSx For Windows (File Server)**: Supports SMB protocol and Windows NTFS; mountable on Linux EC2; scales to 10s GB/s, millions of IOPS, 100s PB; accessible from on-premises (VPN/Direct Connect); configurable for Multi-AZ; daily backups to S3.

VIII. AWS Integration & Messaging

- **Application Communication Patterns**: Decoupling applications using SQS (queue model), SNS (pub/sub model), or Kinesis (real-time streaming).
- **Amazon SQS (Simple Queuing Service)**: Fully managed service for decoupling applications.

    - **SQS - Standard Queue**: Unlimited throughput and messages, default 4-day retention (max 14 days), low latency, 256 KB message limit.
        - **Security**: HTTPS API, KMS keys, Client-Side Encryption, IAM policies, SQS Access Policies.
        - **Message Visibility Timeout**: Message invisible to other consumers for 30 seconds (default) after polling.
        - **Long Polling**: Consumer receives messages in real-time within a wait time (1-20 sec) if no messages in queue.
    - **SQS - FIFO (First In First Out)**: Exactly-once send, messages processed in order.
- **Amazon SNS (Simple Notification Service)**: Managed service for message delivery from publishers to subscribers via topics.
    - **Publishing Methods**: Topic Publish (SDK), Direct Publish (mobile apps SDK).
    - **SNS + SQS (Fan Out)**: Push once to SNS, receive in all subscribed SQS queues; supports FIFO topics; JSON policies for message filtering; cross-region with SQS.
- **Amazon Kinesis**: Collects, processes, analyzes streaming data in real-time (e.g., application logs, metrics, clickstreams, IoT telemetry).
    - **Kinesis Data Streams**: Captures, processes, stores data streams; 1-365 day retention; data immutable once inserted; responsible for consumer creation and scaling.
        - **Capacity Modes**: Provisioned Mode (choose/scale shards manually), On-demand Mode (scales automatically based on throughput peaks).
        - **Partition Key**: Determines which shard a data record belongs to.
    - **Kinesis Data Firehose**: Loads data streams into AWS data stores (S3, Redshift, Elasticsearch, Splunk); near real-time, fully managed, serverless, automatic scaling.
    - **Kinesis Data Analytics**: Analyzes data streams with SQL and Apache Flink; real-time analytics; automatic scaling.
    - **Kinesis Video Streams**: Captures, processes, stores video streams.
- **Amazon MQ**: Managed message broker service for RabbitMQ, ActiveMQ, supporting open protocols (MQTT, AMQP, STOMP, Openwire, WSS) for traditional applications.

IX. Container on AWS

- **AWS ECS (Elastic Container Service)**: Launches Docker containers on AWS ECS Clusters.
    - **EC2 Launch Type**: Provision and maintain underlying EC2 infrastructure; EC2 instances run ECS Agent.
    - **Fargate Launch Type**: Serverless; no infrastructure to provision/manage; simply create task definitions and scale by increasing tasks.

Sure! Here's a complete breakdown of **AWS Fargate** in simple terms:

🚀 What is AWS Fargate?

AWS Fargate is a **serverless compute engine for containers**. It lets you run containers (like Docker) **without managing servers or clusters**. You just define **what** to run (your container image, CPU/memory, and networking), and Fargate takes care of **how** and **where** it's run.

🧠 Analogy:

Think of AWS Fargate as **Uber for containers**. You don’t own the car (server), you just tell it where to go (your container/task), and it arrives ready — no need to maintain the car, fill fuel, or schedule repairs.

🛠️ How It Works

You define:

- A **container image**
- Required **CPU and memory**
- **Networking settings**
- Task or service behavior

Fargate:

- **Launches the container** for you
- **Manages scaling and availability**
- **Automatically provisions and patches infrastructure**

📦 Fargate Runs With:

Service	Description
Amazon ECS	Elastic Container Service — tightly integrated with Fargate
Amazon EKS	Elastic Kubernetes Service — also supports Fargate profiles for pods

---

📊 Key Features

- 🧺 **No server provisioning** — fully serverless
- 📦 **Per-second billing** — pay only for CPU/RAM used
- 🔒 **Built-in isolation** — each task runs in its own kernel-virtualized environment
- 📈 **Auto-scaling** — based on demand

- 🔐 **IAM integration** — secure role-based access per task
- 🌍 **VPC-native networking** — supports private subnet deployments

---

✅ **Benefits**

Benefit	Description
Simplicity	No EC2 instance or AMI management
Efficiency	Auto-scales based on task size
Security	Isolated execution environment
Cost Control	Pay only for what you use

Faster Deployment Launch containers in seconds

💡 **Real-World Use Case:**

Imagine you're deploying a **REST API** that scales up when users hit it and scales down when idle:

- Package your API as a **Docker container**
- Define the CPU and memory (e.g., 0.5 vCPU and 1 GB RAM)
- Use **Fargate with ECS** to run the container
- Attach an **Application Load Balancer (ALB)** in front of it

→ No need to manage or patch any servers!

🧪 **Fargate vs EC2 (in ECS)**

Feature	ECS with EC2	Fargate
Infrastructure	You manage EC2 instances	Fully managed by AWS
Scaling	Manual or auto-scale EC2 fleet	Scales automatically per task
Pricing	Pay for full EC2 uptime	Pay per task per second
Use case	More control or special hardware	Simpler deployments, no ops

---

🔐 **Fargate Security**

- Supports **IAM roles per task**
- **Encrypts data at rest and in transit**
- **Runs each task in an isolated environment**
- Integrated with **AWS VPC** (private subnets, security groups)

🏁 **Common Use Cases:**

- Microservices architecture
- Event-driven processing (e.g., image resizing)
- CI/CD pipelines (e.g., task per build/test job)
- Serverless APIs (with ALB + ECS/Fargate)
- ML inference (short-lived containers with GPU tasks via EKS)

- **IAM Roles for ECS**:
  - **EC2 Instance Profile**: Used by ECS agent (EC2 Launch Type only) for API calls to ECS, sending logs to CloudWatch, pulling Docker images from ECR.
  - **ECS Task Role**: Allows each task to have specific permissions for calling other AWS services (e.g., S3, SQS).

- **ECS-Load Balancer Integration**.
- **ECS-Data Volumes (EFS)**: Mount EFS files onto ECS tasks; works with both EC2 and Fargate (Fargate + EFS = Serverless); provides persistent multi-AZ shared storage.
- **ECS-Autoscaling**:
  - **ECS-Service Auto Scaling**: Automatically scales desired number of ECS tasks using Application Auto Scaling (based on CPU/Memory utilization, ALB Request Count per Target); supports Target, Step, and Scheduled Scaling.
  - **EC2 Launch type-Autoscaling EC2 instances**: Accommodates ECS Service Scaling by adding more underlying EC2 instances; ASG scales based on CPU Utilization.
  - **ECS Cluster Capacity Provider**: Automatically provisions and scales infrastructure for ECS Tasks, paired with ASG to add instances when capacity is missing.

- **Amazon ECR (Elastic Container Registry)**: Stores and manages Docker Images; private and public registries; fully integrated with ECS; backed by S3; access controlled by IAM.
- **Amazon EKS (Elastic Kubernetes Service)**: Open-source system for automatic deployment, scaling, and management of containerized applications; supports EC2 worker nodes or Fargate for serverless containers.
    - **Node Groups**.
    - **Storage Support**: Amazon EBS, EFS (works with Fargate), FSx for Luster, FSx for NetApp ONTAP.
- **AWS App Runner**: Fully managed service to easily deploy and scale web apps from source code or container images; offers automatic builds, deployment, scaling, high availability, load balancing, encryption, and database/cache/message queue connectivity.

X. AWS Database

- **Database Types**:
    - **RDBMS**: RDS, Aurora.
    - **NoSQL**: DynamoDB (JSON), ElastiCache (Key-Value Pair), Neptune (graphs), DocumentDB (MongoDB), Keyspaces (Apache Cassandra).
    - **Data Warehouse**: Redshift, Athena, EMR.
    - **Search**: OpenSearch.
    - **Ledger**: Amazon Quantum Ledger Database (QLDB).
    - **Time Series**: Amazon Timestream.
- **Amazon RDS (Relational Database Service)**: Managed service for relational databases.
    - **Supported Engines**: PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server, Aurora.
    - **Storage Auto Scaling**: Dynamically increases storage for unpredictable workloads.
    - **Read Replicas**: Up to 5 replicas; ASYNC replication; no network cost within AZ.
    - **Multi AZ (Disaster Recovery)**: Used as standby DB; SYNC replication; helps for AZ failures; maintains same connection string.
    - **RDS Custom**: Managed Oracle and Microsoft SQL Server DB with OS and database customization; provides full admin access to underlying OS/DB.
    - **Security**: IAM, Security Groups, SSL in transit, KMS.
    - **Backups**: Automated backup with point-in-time restore (up to 35 days); manual DB snapshots.
    - **Maintenance**: Managed and scheduled (with downtime).
    - **Authentication**: IAM authentication, integration with Secret Managers.
    - **Use Cases**: Store relational datasets (OLTP), perform SQL queries, transactions.

- **Amazon Aurora**: Compatible API for PostgreSQL/MySQL; separates storage and compute.
    - **Performance**: 5x over MySQL on RDS, 3x over Postgres on RDS.
    - **Storage**: Grows automatically in 10GB increments up to 128TB; 6 copies of data across 3 AZs (highly available, self-healing, auto-scaling).
    - **Replicas**: Can have 15 replicas (vs MySQL's 5).
    - **Cost**: 20% costlier than RDS.
    - **Aurora Serverless**: Automated instantiation and auto-scaling based on actual usage; good for infrequent, intermittent, or unpredictable workloads; pay-per-second.
    - **Aurora Multi-Master**: Immediate failover for write nodes.
    - **Global Aurora**: Cross-region Read Replicas for disaster recovery; up to 5 secondary regions; decreased latency.
    - **Machine Learning Integration**: Perform ML via SQL using SageMaker & Comprehend.
    - **Database Cloning**: Create new cluster from existing one, faster than snapshot restore.
    - **Use Cases**: Same as RDS but with less maintenance, more flexibility, performance, and features.
- **ElastiCache**: Managed Redis and Memcached; in-memory databases with high performance, low latency.
    - **Features**: Multi-AZ with Auto-Failover (Redis), Read Replicas (scale reads, high availability), Backup and restore, real-time data update. Clustering (Redis), Multi-AZ, Read Replicas (sharding).
    - **Use Cases**: Key/value store, frequent reads/less writes, cache DB query results, store website session data; cannot use SQL.
- **DynamoDB**: Managed Serverless NoSQL database with millisecond latency.
    - **Capacity Modes**: Provisioned Capacity (specify reads/writes per second, plan capacity, pay for RCU/WCU, can auto-scale) and On-Demand Capacity (reads/writes auto-scale, no planning, pay-per-use, more expensive, great for unpredictable workloads).
    - **Features**: Highly available, Multi-AZ by default, decoupled read/writes, transaction capability.
    - **DAX Cluster**: DynamoDB Accelerator; in-memory cache for microsecond read latency.
    - **Event Processing**: DynamoDB Streams integrate with AWS Lambda or Kinesis Data Streams for real-time changes.
    - **Backups**: Automated backups up to 35 days with Point-In-Time Recovery (PITR) to a new table; on-demand backups for long-term retention.

    - **Data Management**: Export to S3 without RCU, import from S3 without WCU; great for rapidly evolving schemas.
    - **Time to Live (TTL)**: Automatically deletes items after an expiry timestamp (e.g., reduce stored data, regulatory obligations, web session handling).
    - **Use Cases**: Serverless application development, distributed serverless cache.
- **DocumentDB (MongoDB compatible)**: Managed NoSQL database for JSON data; similar deployment concepts to Aurora; fully managed, highly available with 3 AZ replication; storage auto-grows up to 64TB; scales for millions of requests.
- **Neptune (Graph Database)**: Fully managed graph database; highly available across 3 AZs with up to 15 read replicas; for highly connected datasets (e.g., social networks, knowledge graphs, fraud detection, recommendation engines).
- **Keyspaces (Apache Cassandra compatible)**: Managed, serverless, scalable, highly available, fully managed by AWS; scales tables automatically; 3x replication across AZs; uses CQL (Cassandra Query Language); single-digit millisecond latency; On-demand or provisioned capacity; encryption, backup, PITR; use cases: IoT device info, time-series data.
- **Amazon QLDB (Quantum Ledger Database)**: Fully managed, serverless, highly available ledger database with 3 AZ replication; reviews history of all changes (immutable, cryptographically verifiable).
- **Amazon Timestream**: Fully managed, fast, scalable, serverless time series database; auto-scales capacity; stores/analyzes trillions of events; much faster/cheaper than relational DBs; tiered storage (recent data in memory, historical in cost-optimized); built-in analytics functions; encryption in transit/rest; use cases: IoT apps, operational apps, real-time analytics.

XI. Data and Analytics

- **Athena**: Serverless query service to analyze data stored in S3 using standard SQL; supports CSV, JSON, ORC, Avro, Parquet; commonly used with Amazon QuickSight for reporting/dashboard; use cases: BI, analytics, query VPC flow logs, ELB logs, CloudTrail trails.
- **Redshift**: Data warehouse (OLAP), based on PostgreSQL but not for OLTP; 10x better performance than other data warehouses, scales to PBs; columnar storage, parallel query engine; SQL interface; integrates with QuickSight.
    - **Redshift Cluster**: Leader Node (query planning, results aggregation) and Compute Node (performing queries, sending results).
    - **Snapshots & DR**: Stored incrementally in S3; can restore to a new cluster; can automatically copy snapshots to another region.
    - **Redshift Spectrum**: Queries data in S3 without loading it, requiring a Redshift cluster to start queries.
- **OpenSearch**: Successor of Amazon ElasticSearch; searches any field (even partial matches); managed or serverless cluster modes; ingests from Kinesis Data Firehose, Amazon IoT, CloudWatch Logs; security via Cognito, IAM, KMS, TLS; includes OpenSearch Dashboard for visualization.

- **Amazon EMR (Elastic Map Reduce)**: Creates Hadoop clusters for big data analysis and processing (hundreds of EC2 instances); handles provisioning/configuration; autoscaling and Spot Instances integration; use cases: data processing, ML, web indexing, big data.
    - **Node Type & Purchasing**: Master Node (manage cluster, long running), Core Node (run tasks, store data, long running), Task Node (optional, run tasks, usually Spot); purchasing options: On-demand, Reserved, Spot Instances.
- **Amazon QuickSight**: Serverless machine learning-powered business intelligence service for interactive dashboards; fast, auto-scalable, embeddable, per-session pricing; integrates with RDS, Aurora, Athena, Redshift, S3.
- **AWS Glue**: Managed Extract, Transform, and Load (ETL) service; serverless; prepares and transforms data for analytics.
    - **Glue Data Catalog**: Catalog of datasets.
    - **Glue Job Bookmarks**: Prevents re-processing old data.
    - **Glue Elastic Views**: Combines and replicates data across multiple data stores using SQL.
    - **Glue DataBrew**: Cleans and normalizes data using pre-built transformations.
    - **Glue Studio**: GUI to create, run, and monitor ETL jobs.
    - **Glue Streaming ETL**: Built on Apache Spark Structured Streaming; compatible with Kinesis Data Streaming, Kafka, MSK.
- **AWS Lake Formation**: Fully managed service to set up a data lake; automates complex manual steps (collecting, cleansing, moving, cataloging, de-duplicating data); built on AWS Glue.
- **Amazon Managed Streaming for Apache Kafka (Amazon MSK)**: Alternative to Kinesis; fully managed Amazon Kafka on AWS; creates and manages Kafka/Zookeeper nodes; deploys in VPC, Multi-AZ; automatic recovery; data stored on EBS.
    - **MSK Serverless**: Runs Apache Kafka without managing capacity; automatically provisions resources and scales compute/storage.
- **Big Data Ingestion Pipeline**: Example serverless, real-time data pipeline using IoT Core, Kinesis, Firehose, Lambda, S3, SQS, Athena, and QuickSight/Redshift for reporting.

XII. Machine Learning

- **Amazon Rekognition**: Finds objects, text, scenes, people in images and videos using ML; facial analysis, facial search, content moderation, celebrity recognition.
- **Amazon Transcribe**: Automatically converts speech to text using ASR (Automatic Speech Recognition); redacts PII; supports Automatic Language Identification for multi-lingual audio; use cases: transcribing calls, closed captioning, metadata for media assets.
- **Amazon Polly**: Turns text into lifelike speech using deep learning; customizes pronunciation with Lexicons and SSML.

- **Amazon Translate**: Natural and accurate language translation for localizing content (websites, apps) and translating large texts efficiently.
- **Amazon Lex & Connect**:
    - **Lex**: Powers Alexa; uses ASR (speech to text) and Natural Language Understanding (recognize intent); builds chatbots, call center bots.
    - **Connect**: Cloud-based virtual contact center; receives calls, creates contact flows; integrates with CRM/AWS; no upfront payments, 80% cheaper than traditional solutions.
- **Amazon Comprehend**: Fully managed, serverless NLP service using ML to find insights and relationships in text (e.g., language, key phrases, sentiment, topic organization).
- **Amazon Comprehend Medical**: Detects useful information in unstructured clinical text (e.g., physicians' notes, discharge summaries, test results); uses NLP to detect Protected Health Information (PHI).
- **Amazon SageMaker**: Fully managed service for developers/data scientists to build ML models.
- **Amazon Forecast**: Fully managed service using ML for highly accurate forecasts (e.g., sales prediction); 50% more accurate than traditional methods; reduces forecasting time from months to hours; use cases: product demand, financial planning, resource planning.
- **Amazon Kendra**: Fully managed document search service powered by ML; provides exact answers from various document types; natural language search; learns from user interactions; allows manual fine-tuning.
- **Amazon Personalize**: Fully managed ML service to build apps with real-time personalized recommendations; integrates with existing websites, applications, marketing systems; implements in days.
- **Amazon Textract**: Automatically extracts text, handwriting, and data from scanned documents using AI/ML; extracts data from forms and tables; reads PDFs, images; use cases: financial services, healthcare, public sector.

XIII. AWS Monitoring

- **AWS CloudWatch Metrics**: Provides metrics for every AWS service (e.g., EC2 CPU Utilization, Bucket Size); metrics have dimensions and timestamps; can create custom metrics (e.g., Memory usage in EC2) and dashboards.
    - **CloudWatch Metric Streams**: Continuously streams CloudWatch metrics to a destination (e.g., Kinesis Data Firehose) with near real-time delivery and low latency.
- **CloudWatch Logs**: Monitors, stores, and accesses log files from EC2, CloudTrail, Route 53, and other sources; centralizes logs; allows viewing, searching, filtering, and archiving; supports querying with a powerful language, auditing, masking sensitive data, and generating metrics from logs.
    - **CloudWatch Insight**: Search and analyze log data stored in CloudWatch Logs; queries multiple log groups.

    - **CloudWatch Logs Subscription**: Real-time log events from CloudWatch Logs for processing/analysis; sends to Kinesis Data Streams/Firehose or Lambda; supports filter and cross-account subscriptions.
    - **CloudWatch on EC2**: Requires a CloudWatch agent to push log files to CloudWatch.
    - **CloudWatch Logs Agent & Unified Agent**: Older Logs Agent sends only to CloudWatch logs; Unified Agent provides more granular metrics and monitoring details.
- **CloudWatch Alarms**: Triggers notifications for any metric; alarm states: OK, INSUFFICIENT DATA, ALARM; targets include stopping/terminating/rebooting/recovering EC2 instances, triggering Auto-Scaling actions, sending SNS notifications.
    - **Composite Alarms**: Monitor states of multiple alarms using AND/OR conditions.
    - **EC2 Recovery**: Alarms can be created based on CloudWatch Logs Metrics Filter for EC2 recovery.
- **Amazon EventBridge (CloudWatch Events)**: Schedules CRON jobs; defines event rules to react to service actions; triggers Lambda functions, sends SQS/SNS messages.
    - **EventBridge + CloudTrail**: Intercepts API calls and detects unusual activity.
- **CloudWatch Container Insights**: Collects, aggregates, summarizes metrics and logs from containers on ECS, EKS, Kubernetes, and Fargate.
- **CloudWatch Lambda Insights**: Monitoring and troubleshooting for serverless applications on AWS Lambda; collects system-level metrics (CPU, memory, disk, network) and diagnostic info (cold starts, worker shutdown).
- **CloudWatch Application Insights**: Provides automated dashboards for potential application problems, isolating issues, and enhancing visibility to reduce troubleshooting time.
- **AWS CloudTrail (CCTV for AWS)**: Provides governance, compliance, and audit for AWS accounts; enabled by default; identifies who called AWS, from where, and when (console, SDK, CLI, AWS services); logs to CloudWatch Logs or S3.
    - **CloudTrail Events**:
        - **Management Events**: Operations performed on resources; logged by default.
        - **Data Events**: S3 object-level activity (GetObject, DeleteObject), Lambda function execution activity; not logged by default.
        - **Insight Events**: Detects unusual activity (e.g., inaccurate resource provisioning, hitting service limits, maintenance gaps).
    - **Event Retention**: Events stored for 90 days; for longer retention, log to S3 and use Athena.
- **AWS Config**: Helps with auditing and recording compliance of AWS resources; records configurations and changes over time; sends SNS alerts for changes; per-region service; stores config data in S3 for Athena analysis.

    - **Config Rules**: AWS managed or custom rules (e.g., checking EBS disk type); notifications via EventBridge for non-compliant resources.

- **Amazon GuardDuty**: Threat detection service using ML to monitor for malicious behavior (e.g., unusual API calls, compromised instances); alerts in console or CloudWatch Events; monitors CloudTrail Logs, VPC flow logs, DNS logs.
- **Amazon Macie**: Fully managed data security and privacy service using ML/AI to discover and protect sensitive data (e.g., PII) in AWS.
- **Amazon Inspector**: Automated security assessment service to improve security/compliance; assesses applications for vulnerabilities or deviations from best practices.
    - **Types of Assessments**: Network Assessments (network config analysis, no agent required), Host Assessments (vulnerable software, host hardening, security best practices, agent required).
- **AWS Security Hub**: A centralized security service that aggregates, organizes, and prioritizes security alerts (findings) from multiple AWS services (e.g., GuardDuty, Inspector, Macie) and partner tools—providing a single pane of glass for your security posture and compliance status.
- **AWS Audit Manager**: A service that automates evidence collection to help you continuously audit your AWS usage against frameworks and standards (e.g., PCI-DSS, HIPAA, GDPR). It maps AWS resource configurations to controls, generates ready-to-use audit reports, and reduces manual effort.
- **AWS Artifact:** An on-demand, self-service portal for accessing AWS's own compliance reports (e.g., SOC 2, ISO 27001) and security and privacy documents. Use Artifact to download AWS audit artifacts, review certifications, and share evidence with your auditors.
- **AWS Detective**: A security investigation service that automatically collects log data (VPC Flow Logs, CloudTrail events, GuardDuty findings), builds a "graph" of resource interactions, and uses machine learning to help you quickly identify the root cause of security issues or suspicious activities.

XIV. AWS Security & Encryption

- **Encryption in flight (TLS/SSL)**: Data encrypted before sending and decrypted after receiving (HTTPS); prevents MITM attacks.
- **Server-Side Encryption at rest**: Data encrypted after being received by server; stored encrypted, decrypted before sending.
- **Client-Side Encryption**: Data encrypted by client, never decrypted by server, decrypted by receiving client; server cannot decrypt data.
- **AWS KMS (Key Management Service)**: Most used encryption service; AWS manages encryption keys; integrated with IAM for authorization; audits key usage via CloudTrail; secrets should never be stored in plain text.
    - **KMS Key Types**:

        - **Symmetric (AES-256 keys)**: Single key for encryption/decryption; used by integrated AWS services; never get unencrypted keys.
        - **Asymmetric (RSA & ECC Key pairs)**: Public (encrypt) and Private (decrypt) pair; public key downloadable, private key inaccessible unencrypted; for encryption outside AWS.
    - **CloudHSM**: Cloud-based HSM to generate and use encryption keys.
    - **Copying Snapshots Across regions/accounts**: Requires KMS key encryption and policy authorization for cross-account access.
    - **KMS Multi-region Keys**: Identical KMS keys in different regions for interchangeability (encrypt in one, decrypt in another); not global (primary + replicas); use cases: global client-side encryption, Global DynamoDB/Aurora.
- **S3 Replication with Encryption**: Unencrypted and SSE-S3 objects replicated by default; SSE-C objects can be replicated; SSE-KMS requires enabling option.
- **AMI Sharing Process Encrypted via KMS**: Requires modifying image attributes, sharing KMS keys, and IAM permissions for target account.
- **SSM (Simple Systems Manager) Parameter Store**: Secure storage for configuration/secrets; optional seamless KMS encryption; serverless, scalable, durable; version tracking; IAM security; EventBridge notifications; CloudFormation integration.
- **AWS Secret Manager**: Newer service for storing secrets; forces rotation every X days; automates secret generation (Lambda); integrates with RDS (MySQL, PostgreSQL, Aurora); secrets encrypted via KMS.
    - **Multi-Region Secrets**: Replicates secrets across regions; keeps read replicas in sync; use cases: multi-region apps, disaster recovery, multi-region DB.
- **AWS Certificate Manager (ACM)**: Easily provisions, manages, deploys TLS certificates for in-flight encryption (HTTPS); supports public/private certs; integrates with ELB, CloudFront, API Gateway; cannot be used with EC2.
    - **Requesting Public Certificates**: List domain names, select DNS/Email Validation; automatic renewal.
    - **Importing Public Certificates**: Generate outside ACM and import; no automatic renewal.
    - **Integration with ALB/API Gateway**.
- **AWS WAF (Web Application Firewall)**: Protects web applications from common web exploits (Layer 7 attacks - HTTP); deploys on ALB, API Gateway, CloudFront, AppSync GraphQL API, Cognito User Pool.
    - **Web ACL (Web Access Control Lists) Rules**: IP sets, HTTP header/body/URI strings (SQL injection, XSS protection), size constraints, geo-match, rate-based rules (DDoS protection); regional except for CloudFront.
    - **Fixed IP with WAF/Load Balancer**: WAF doesn't support NLB; Global Accelerator can provide fixed IP with WAF on ALB.

- **AWS Shield (DDoS protection)**: Protects against Distributed Denial-of-Service (DDoS) attacks.
    - **AWS Shield Standard**: Free; activated for every AWS customer; protects against Layer 3/4 attacks (SYN/UDP Floods, Reflection attacks).
    - **AWS Shield Advanced**: Optional ($3000/month); protects against more sophisticated attacks on EC2, ELB, CloudFront, Global Accelerator, Route 53; automatic application layer DDoS mitigation (creates WAF rules).
- **AWS Firewall Manager**: Manages rules across all accounts in an AWS Organization; defines common security policies (WAF rules, Shield Advanced, Security Groups, Network Firewall, Route 53 Resolver DNS Firewall); rules apply to new resources; policies created at region level.
- **AWS Network Firewall**.
- **WAF vs Firewall Manager vs Shield**: WAF defines Web ACL rules for granular protection; Firewall Manager accelerates WAF configuration across accounts and automates protection; Shield Advanced adds features like dedicated support and advanced reporting for frequent DDoS attacks.

XV. <u>Serverless</u>

- **AWS Lambda**: Virtual functions, no servers to manage; short executions; run on-demand; automated scaling; pay per request and compute time; integrates with many languages, container images; easy monitoring with CloudWatch.
    - **Lambda Limits**: Memory (128MB-10GB), Max Execution time (900 sec/15 min), Environment variables (4KB), Disk Capacity (/tmp: 512MB-10GB), Concurrency (1000, can be increased); Deployment size (compressed: 50MB, uncompressed: 250MB).
    - **AWS Lambda SnapStart**: Improves Java 11+ Lambda performance up to 10x at no extra cost by invoking from a pre-initialized snapshot of memory/disk state.
    - **Customization at the Edge**: Executes logic close to users to minimize latency.
        - **CloudFront Functions**: Lightweight JavaScript functions for high-scale, latency-sensitive CDN customization; sub-ms startup times, millions of req/sec; changes Viewer Request/Response; native CloudFront feature.
        - **Lambda@Edge**: Lambda functions (NodeJS/Python) for CloudFront Request/Response changes; scales up to 1000 req/sec; functions authored in one region, replicated by CloudFront.
    - **Lambda in VPC**: By default, Lambda runs outside your VPC; to access resources within your VPC (RDS, ElastiCache, internal ELB), you must define VPC ID, subnets, and security groups (Lambda creates ENI).
    - **Lambda with RDS Proxy**: Improves scalability by pooling/sharing DB connections for Lambda; improves availability/security; Lambda must be in your VPC as RDS Proxy is not publicly accessible.
    - **Invoking Lambda from RDS & Aurora**.

    - **DynamoDB Accelerator (DAX)**: Fully managed, highly available, seamless in-memory cache for DynamoDB; microsecond latency for cached data; 5 min TTL.
    - **DynamoDB – Stream Processing**: Ordered stream of item-level modifications; use cases: real-time changes, usage analytics, cross-region replication, invoking Lambda on changes.
- **API Gateway**: No infrastructure to manage when used with Lambda; supports WebSocket protocol; handles API versioning and environments (dev, test, prod); manages security (AuthN/AuthZ), API Keys, request throttling; Swagger/Open API import; transforms/validates requests/responses; generates SDK/API spec; caches API responses.
    - **High Level Integrations**: Lambda Function (expose REST API), HTTP (expose internal/external HTTP endpoints, ALB), AWS Services (expose any AWS API).
    - **Endpoint Types**:
        - **Edge Optimized (default)**: For global clients; requests routed through CloudFront Edge locations to improve latency; API Gateway still in one region.
        - **Regional**: For clients within the same region; can manually combine with CloudFront for more control.
        - **Private**: Accessible only from your VPC using an interface VPC endpoint (ENI).
    - **Security**: User Authentication via IAM Roles (internal apps), Cognito (external/mobile users), Custom Authorizer (custom logic); HTTPS security via ACM for custom domain names.
- **AWS Step Function**: Builds serverless visual workflows to orchestrate Lambda functions; features: sequence, parallel, conditions, timeout, error handling; integrates with EC2, ECS, on-premises servers, API Gateway, SQS; supports human approval; use cases: order fulfillment, data processing, web apps, any workflow.
- **AWS Cognito**: Provides user identity for web/mobile applications.
    - **Cognito User Pools**: Sign-in functionality for app users; serverless database of users; supports username/email/password login, password reset, email/phone verification, MFA, federation with Google/Facebook/SAML; integrates with API Gateway/ALB.
    - **Cognito Identity Pools (Federated Identity)**: Provides AWS credentials to users to access AWS resources directly; integrates with Cognito User Pools as identity provider.
    - **Cognito vs IAM**: Cognito for hundreds of external/mobile users, especially with SAML; IAM for internal AWS users/services.

XVI. <u>Networking</u>

- **Region & Availability Zones**: A region is a geographical area with 2+ Availability Zones; AZs are separated for disaster isolation.
- **Edge Location**: Endpoints for AWS used for caching content (e.g., by CloudFront).

- **Route Table**: Set of rules (routes) to determine where network traffic is directed; associated with VPC subnets to define traffic flow.
    - **Key Concepts**:
        - **Route**: Single rule specifying destination (IP range) and target (where traffic goes).
        - **Destination CIDR Block**: Network range the route applies to.
        - **Target**: Where traffic is directed (e.g., Internet Gateway, Virtual Private Gateway, NAT Gateway/Instance, VPC Peering Connection, Local).
        - **Main Route Table**: Default route table for a VPC, applies to all subnets unless custom table is associated.
    - **Types of Route Tables**: Main Route Table (default), Custom Route Tables (user-defined for granular control).
    - **Common Scenarios**:
        - **Public Subnet**: Route table allows traffic to Internet Gateway.
        - **Private Subnet**: Isolated from internet, uses NAT Gateway/Instance for outbound access.
        - **VPN Connection**: Routes traffic to Virtual Private Gateway for on-premises connectivity.
        - **VPC Peering**: Routes traffic between two peered VPCs.
    - **Structure & Subnets**: Route Table ID, Routes (CIDR blocks/targets), Associations (subnets); each subnet associated with a route table.
    - **Best Practices**: Separate public/private subnets, least privilege routing.
- **CIDR (Classless Inter-Domain Routing)**: Method for allocating IP addresses; defines IP address ranges, base IPs, and subnet masks (e.g., /8, /16, /24, /32).
- **Public vs Private IP**: IANA established private IPv4 blocks (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16); all others are public.
- **VPC in AWS (Virtual Private Cloud)**: Private sub-section of AWS you control to place resources; full control over virtual networking environment (IP range, subnets, route tables, gateways); multiple VPCs allowed (max 5 per region, soft limit); allows private IPv4 ranges; min size /28 (16 IPs), max size /16 (65536 IPs).
    - **Subnets**: AWS reserves 5 IP addresses (first 4, last 1) in each subnet.
- **Internet Gateway (IGW)**: Allows AWS resources in a VPC to connect to the internet; horizontally scaled, highly available; created separately from VPC; one IGW per VPC; requires Route Table configuration.
- **Bastion Hosts**: Used to SSH into private EC2 instances; located in public subnet, connected to private subnets; security group must allow inbound on port 22 from internet.

- **NAT Instances**: Allows EC2 instances in private subnet to connect to internet; launched in public subnet with Elastic IP; requires route table configuration from private subnets.
- **NAT Gateway (NATGW)**: AWS managed NAT service; higher bandwidth, high availability, no administration; pay per hour for usage/bandwidth; created in specific AZ with Elastic IP; cannot be used by EC2 in same subnet; requires IGW; no security groups to manage.
- **Gateway Endpoints** – A free, highly available VPC endpoint for S3 or DynamoDB traffic that doesn't traverse the internet.
- **Interface Endpoints (AWS PrivateLink)** – Elastic Network Interfaces (ENIs) in your subnet that privately connect to supported AWS services and marketplace SaaS.
- **Site to Site VPN**: Connects on-premises networks to AWS VPC over the public internet.
    - **Virtual Private Gateway (VGW)**: VPN concentrator on AWS side, attached to VPC.
    - **Customer Gateway**: Software/physical device on customer side.
    - **AWS VPN CloudHub**: Secure communication between multiple sites; low-cost hub-and-spoke model for network connectivity.
- **NACL (Network Access Control List)**: Firewall controlling traffic to/from subnets; one NACL per subnet; new NACLs deny everything by default; stateless (inbound/outbound rules must be explicit); default NACL accepts all.
    - **Ephemeral Port**: Clients connect to defined port, expect response on an ephemeral port.
- **VPC Peering**: Privately connects two VPCs (even in different accounts/regions) using AWS network, making them behave as if in same network; requires non-overlapping CIDRs and route table updates.
- **VPC Endpoints**: Privately connect to supported AWS services and VPC endpoint services powered by AWS PrivateLink; virtual, horizontally scaled, redundant, highly available devices; remove need for IGW/NATGW.
    - **Types**:
        - **Interface Endpoints (PrivateLink)**: Provisions an ENI (private IP) as entry point; supports most AWS services; cost: $ per hour + $ per GB; preferred for access from on-premises/different VPC/region.
        - **Gateway Endpoint**: Provisions a gateway used as a target in a route table; supports S3 and DynamoDB; free; usually preferred for S3/DynamoDB.
- **VPC Flow logs**: Captures IP traffic info for VPC, Subnet, ENI; monitors/troubleshoots connectivity issues; data to S3, CloudWatch Logs, Kinesis Data Firehose.
- **Direct Connect (DX)**: Provides dedicated private connection from remote network to VPC; set up between data center and AWS DX Locations; requires VGW on VPC; accesses public/private resources on same connection; longer lead times (1+ month).
    - **Use Cases**: Increased bandwidth throughput (no public internet), consistent network experience (real-time data), hybrid environments.

    - **Direct Connect Gateway**: For DX to multiple VPCs in different regions (same account).
    - **Encryption**: Data in transit not encrypted but private; DX + VPN provides IPsec-encrypted private connection.
    - **Resiliency**: Can set up backup DX or Site-to-Site VPN.
- **Transit Gateway**: For transitive peering between thousands of VPCs and on-premises (hub-and-spoke); shares cross-account using Resource Access Manager (RAM); peers across regions; sets up route tables for communication; works with DX Gateways, VPN connections; supports IP Multicast.
    - **Site to Site VPN ECMP**: Equal cost multi-path routing; allows forwarding packets over multiple best paths to increase bandwidth.
    - **Share Direct Connect** : Can be shared between multiple accounts via Transit Gateway .

XVII. Cost Management & Billing Tools

- **AWS Cost Explorer** – Visualize and analyze your spending patterns over time.
- **AWS Budgets** – Create custom cost, usage, or reservation budgets and receive alert notifications.
- **AWS Cost and Usage Report (CUR)** – The most granular billing data, delivered to S3, for custom analysis.
- **AWS Pricing Calculator** – Estimate the monthly cost of a planned architecture before you build it.
- **Resource Tagging** – Key-value pairs attached to AWS resources to identify ownership, environment, cost center, etc.
- **Cost Allocation Tags** – A special subset of tags activated in the billing console so that you can break down your bill by tag.
- **AWS Organizations** – Centrally manage policies, SCPs (Service Control Policies), and consolidated billing across multiple AWS accounts.
- **Consolidated Billing** – Aggregate usage from all member accounts into one invoice—and share volume pricing discounts.

Flashcards

Here are all 100 AWS Certified Cloud Practitioner Flashcards, organized and formatted so you can easily copy them into Notion, Google Docs, or Quizlet.

✅ AWS Cloud Practitioner Flashcards (1–100)

🌩️ Cloud Concepts & Deployment

Q1: What are the three main types of cloud computing?
A1: IaaS, PaaS, SaaS

Q2: What are the three types of cloud deployment models?
A2: Public Cloud, Private Cloud, Hybrid Cloud

Q3: What is the shared responsibility model?
A3: AWS secures the cloud; customers secure what's in the cloud.

Q4: What is cloud computing?
A4: On-demand delivery of IT resources over the internet with pay-as-you-go pricing.

Q5: Name three benefits of cloud computing.
A5: Agility, Elasticity, Cost Savings

Q6: What is elasticity in cloud computing?
A6: Automatically adding or removing resources based on demand.

Q7: What is scalability in AWS?
A7: Ability to grow infrastructure to meet increased demand.

Q8: What are AWS Regions?
A8: Geographically isolated locations containing multiple AZs.

Q9: What are Availability Zones?
A9: Data centers within a region that are isolated from each other.

Q10: What are Edge Locations?
A10: Locations used by CloudFront to cache content closer to users.

💸 Billing, Pricing, and Support

Q11: What are the four main AWS pricing models?
A11: On-Demand, Reserved, Spot, Savings Plans

Q12: Which instance pricing is best for temporary workloads?
A12: Spot Instances

Q13: What is the benefit of Reserved Instances?
A13: Lower cost for long-term workloads with a 1–3 year commitment.

Q14: What are AWS Savings Plans?
A14: Flexible pricing model offering cost savings with a usage commitment.

Q15: What is the AWS Free Tier?
A15: Free usage of certain AWS services for 12 months after sign-up.

Q16: What is the AWS Pricing Calculator used for?
A16: Estimate the cost of using AWS services.

Q17: What is Consolidated Billing?
A17: Combine multiple accounts under one to get volume discounts.

Q18: What does the AWS Budgets service do?
A18: Set custom cost and usage budgets and get alerts.

Q19: What is AWS Cost Explorer?
A19: A tool to visualize and analyze AWS costs over time.

Q20: What is the TCO Calculator?
A20: Estimates cost savings by comparing AWS with on-premises.

📜 Support Plans

Q21: What are the four AWS Support Plans?
A21: Basic, Developer, Business, Enterprise

Q22: Which support plans offer 24/7 access to AWS Support?
A22: Business and Enterprise

Q23: Which plan includes a Technical Account Manager (TAM)?
A23: Enterprise

Q24: Which support plan is free?
A24: Basic

Q25: What is AWS Trusted Advisor?
A25: Gives recommendations on cost, security, performance, and more.

Q26: What kind of access does the Developer support plan provide?
A26: Email access to cloud support engineers during business hours.

Q27: What kind of guidance is included in Business support?
A27: General architectural and production guidance.

Q28: What is an AWS Support Concierge?
A28: Assists Enterprise customers with billing and account questions.

Q29: What is the SLA for AWS support response time for critical issues under Enterprise plan?
A29: 15 minutes or less

Q30: Does Basic support include access to Trusted Advisor checks?
A30: Yes, but only for 7 core checks.

🔐 Security & Identity

Q31: What is IAM?
A31: Identity and Access Management service to control access to AWS resources.

Q32: What is Multi-Factor Authentication (MFA)?
A32: Adds a second level of security for IAM users.

Q33: What is AWS Organizations?
A33: Service for centrally managing multiple AWS accounts.

Q34: What is AWS Shield?
A34: DDoS protection service.

Q35: What is AWS WAF?
A35: Web Application Firewall for filtering HTTP/S traffic.

Q36: What is encryption at rest?
A36: Data is encrypted while stored.

Q37: What is encryption in transit?
A37: Data is encrypted during transfer.

Q38: What is AWS Artifact?
A38: Provides access to compliance documentation and reports.

Q39: What is AWS Inspector?
A39: Scans EC2 for vulnerabilities and security issues.

Q40: What is the AWS Well-Architected Framework?
A40: A guide based on five pillars to help build secure, high-performing applications.

🧰 Core Services – Compute, Storage, Network

Q41: What is EC2?
A41: Elastic Compute Cloud — virtual servers in the cloud.

Q42: What is S3?
A42: Simple Storage Service for object storage.

Q43: What is EBS?
A43: Elastic Block Store — persistent block storage for EC2.

Q44: What is EFS?
A44: Elastic File System — shared file storage for EC2.

Q45: What is AWS Lambda?
A45: Serverless compute service to run code without provisioning servers.

Q46: What is Amazon RDS?
A46: Relational Database Service — managed DB (e.g., MySQL, PostgreSQL).

Q47: What is Amazon DynamoDB?
A47: Fully managed NoSQL database.

Q48: What is Amazon VPC?
A48: Virtual Private Cloud — isolated section of AWS network.

Q49: What is CloudFront?
A49: Content Delivery Network (CDN) that caches content at edge locations.

Q50: What is Route 53?
A50: Scalable DNS and domain name management service.

🛠️ DevOps, Monitoring, and Tools

Q51: What is CloudFormation?
A51: Infrastructure as Code (IaC) service to provision AWS resources.

Q52: What is AWS Elastic Beanstalk?
A52: PaaS for deploying applications without managing infrastructure.

Q53: What is AWS CodeDeploy?
A53: Automates deployment of apps to EC2, Lambda, or on-prem servers.

Q54: What is AWS CloudTrail?
A54: Logs all API activity in an AWS account.

Q55: What is Amazon CloudWatch?
A55: Monitoring service for logs, metrics, and alarms.

Q56: What is the AWS Management Console?
A56: Web-based GUI to access AWS services.

Q57: What is the AWS CLI?
A57: Command Line Interface for managing AWS services via commands.

Q58: What is the AWS SDK?
A58: Software tools to interact with AWS from programming languages.

Q59: What is AWS OpsWorks?
A59: Configuration management service using Chef and Puppet.

Q60: What is AWS Service Catalog?
A60: Allows organizations to create and manage approved service portfolios.

🌐 Networking & Global Infrastructure

Q61: What is an Internet Gateway in AWS?
A61: Allows VPC resources to access the internet.

Q62: What is a NAT Gateway?
A62: Allows instances in a private subnet to access the internet.

Q63: What is a subnet?
A63: A range of IP addresses in your VPC.

Q64: What is a security group?
A64: A virtual firewall for EC2 instances.

Q65: What is NACL (Network ACL)?
A65: Optional layer of security at the subnet level.

Q66: What is Direct Connect?
A66: Dedicated network connection between on-prem and AWS.

Q67: What is a VPC Peering Connection?
A67: Connects two VPCs to communicate privately.

Q68: How many AZs are in a typical region?
A68: Usually 3 or more.

Q69: Can you launch an EC2 in a specific AZ?
A69: Yes, you can choose the AZ during launch.

Q70: What AWS service lets you build a hybrid environment?
A70: AWS Outposts

🧠 Miscellaneous & Best Practices

Q71: What is the principle of least privilege?
A71: Users should have only the permissions needed to do their jobs.

Q72: What is CAPEX vs. OPEX?
A72: Capital expense (upfront) vs. Operational expense (ongoing pay-as-you-go)

Q73: What are tags in AWS?
A73: Key-value pairs for organizing resources.

Q74: What is a use case for using S3 Glacier?
A74: Long-term, infrequent access data archival.

Q75: What is AWS QuickSight?
A75: Business intelligence and data visualization tool.

Q76: What is Amazon Athena?
A76: Query data in S3 using SQL.

Q77: What is AWS Control Tower?
A77: Automates account setup and governance in AWS Organizations.

Q78: What is AWS Config?
A78: Monitors and records AWS resource configurations.

Q79: What is AWS Systems Manager?
A79: Operations hub for visibility and control of infrastructure.

Q80: What is a Spot Fleet?
A80: A collection of Spot Instances to meet target capacity.

🧪 Final 20: Scenarios, Real-World, and Governance

Q81: Which service allows serverless batch processing?
A81: AWS Lambda

Q82: How do you restrict S3 bucket access to specific users?
A82: Use bucket policies and IAM roles.

Q83: How can you ensure data integrity in S3?
A83: Enable versioning and use checksums.

Q84: How does CloudFront reduce latency?
A84: Caches content at edge locations closer to users.

Q85: How can you reduce EC2 cost for predictable workloads?
A85: Use Reserved Instances or Savings Plans.

Q86: What is a landing zone in AWS?
A86: Preconfigured environment with best practices.

Q87: How can you automate infrastructure deployment?
A87: Use CloudFormation templates.

Q88: What is a common use of Route 53 health checks?
A88: DNS-based failover for high availability.

Q89: What is a benefit of multi-AZ RDS deployments?
A89: High availability and automatic failover.

Q90: How do you prevent public access to an S3 bucket?
A90: Block public access settings at bucket and account level.

Q91: What is a resource group?
A91: Collection of AWS resources grouped by tags.

Q92: What is a use case for Amazon Kinesis?
A92: Real-time data streaming and analytics.

Q93: What is Amazon Lightsail?
A93: Simplified cloud platform for small applications.

Q94: What is AWS Snowball?
A94: Physical device to transfer large data volumes to/from AWS.

Q95: What is AWS Batch?
A95: Manages and runs batch computing jobs.

Q96: What is Amazon Macie?
A96: Uses ML to discover and protect sensitive data in S3.

Q97: What is AWS Personal Health Dashboard?
A97: Provides alerts and remediation guidance for AWS events.

Q98: What is the use of Elastic Load Balancer (ELB)?
A98: Automatically distributes incoming traffic across targets.

Q99: What is AWS Marketplace?
A99: Digital catalog of third-party software and services.

Q100: What is the AWS Service Health Dashboard?
A100: Shows current and past status of AWS services globally.
