AWS SOLUTIONS ARCHITECT ASSOCIATE (SAA-C03) HANDBOOK

INDEX

Amazon Athena

What It Is

Amazon Athena is an interactive query service that makes it easy to analyse data in Amazon S3 using standard SQL. It is serverless, so there is no infrastructure to manage, and you pay only for the queries you run.

Key Features

- Serverless, no need to provision or manage infrastructure
- SQL-based queries using standard ANSI SQL
- Supports querying data in S3 directly
- Pay-per-query pricing model, charged per TB of data scanned
- Integrated with AWS Glue Data Catalog for schema management
- Supports complex joins, window functions, and arrays
- Results can be stored in S3

How It Works

- Point Athena at data stored in S3
- Define table schema using Data Catalog or DDL in Athena
- Run SQL queries to analyse data directly in S3
- Query results are saved to S3 in specified location
- Can partition data to improve performance and reduce costs

Supported Data Formats

- CSV, TSV, JSON
- ORC, Parquet, Avro
- Text files with custom delimiters
- Apache Web Logs

Performance Optimization

- **Partitioning**: Breaks data into subsets for faster queries
- **Columnar formats:** ORC and Parquet reduce data scanned
- **Compression:** Reduces data transfer and storage costs
- **Bucketing:** Organizes data for efficient joins

AWS Glue Integration

- Uses AWS Glue Data Catalog to store metadata
- Can crawl S3 data to infer schema automatically
- Shared catalog with Amazon Redshift Spectrum and EMR

- Supports schema versioning

Security

- Encryption at rest using S3 server-side encryption (SSE-S3, SSE-KMS)
- Encryption in transit using HTTPS
- IAM policies to control access to queries and data
- Integration with AWS Lake Formation for fine-grained access control
- Query result encryption in S3 using SSE

Monitoring and Logging

- Integrated with AWS CloudWatch for query metrics
- CloudTrail logs for auditing Athena API calls
- Query history available in the Athena console
- Can monitor failed or long-running queries

Use Cases

- Ad hoc querying of log files in S3
- Analysing AWS service logs (CloudTrail, ELB, VPC Flow Logs)
- Data lake analytics without ETL
- Interactive data exploration and reporting
- Integrating with BI tools like Amazon QuickSight
- Cost-effective data warehousing for infrequent queries

Pricing

- Charged per TB of data scanned
- Compressed, columnar, and partitioned data reduces cost
- No charges for infrastructure provisioning or idle time
- AWS Glue Data Catalog pricing applies separately

Integration with Other AWS Services

- Amazon S3: Data source and result storage
- AWS Glue: Data Catalog for schema management
- Amazon QuickSight: For visualization and dashboards
- AWS Lake Formation: For data lake governance and security
- AWS CloudTrail and CloudWatch for logging and monitoring

Best Practices

- Use partitioning to limit data scanned

- Store data in columnar formats (Parquet or ORC)
- Compress data to reduce storage and scan costs
- Manage schemas in AWS Glue Data Catalog
- Secure data using IAM policies and encryption

Exam Tips

- Serverless SQL queries on data in S3
- Charged per TB scanned, so optimize with partitions and formats
- Integrated with AWS Glue Data Catalog
- Supports multiple data formats including Parquet and ORC
- Can be used with QuickSight for visualization
- Best for ad hoc queries on large datasets stored in S3
- Use IAM and Lake Formation for access control
- Encrypt data at rest and in transit

Quick Summary

Amazon Athena is a serverless query service that analyses S3 data using SQL without infrastructure management. It is cost-effective, integrates with AWS Glue for schema management, and supports multiple data formats and security features, making it ideal for ad hoc analytics and data lake querying.

AWS Data Exchange

What It Is

AWS Data Exchange is a fully managed service that makes it easy to find, subscribe to, and use third-party data in the cloud. It helps organizations securely exchange and consume data sets published by data providers, such as data from Reuters, Foursquare, and other providers, without complex licensing and delivery processes.

Key Features

- Catalog of thousands of third-party data products
- Supports public data products and private (entitlement-based) data sharing
- Automates data subscription, entitlement, and billing workflows
- Direct delivery of data sets to Amazon S3
- Data can be automatically updated as providers publish new revisions
- Supports REST API and AWS CLI for managing subscriptions and data sets
- Event notifications via Amazon EventBridge when new data is published

Data Sets and Revisions

- Data providers publish data sets in the AWS Data Exchange catalog
- Each data set contains one or more revisions (versions)
- Revisions can include files in formats like CSV, Parquet, or JSON
- Subscribers access data revisions through entitlement

Subscriptions

- Subscribers can browse and subscribe to products in AWS Marketplace
- Subscriptions can be free or paid (with integrated billing)
- Entitlement grants access to the data set revisions
- Automatic export of data revisions to an S3 bucket
- Option to automate workflows on data arrival using EventBridge

Data Delivery

- Data sets are delivered directly to subscribers' S3 buckets
- Supports incremental updates as new revisions are released
- Delivery logs available in CloudTrail and CloudWatch
- Data can be accessed via Athena, EMR, Redshift Spectrum, or other analytics tools

Private Data Products

- Enables data providers to share data products privately with specific AWS accounts
- Supports entitlement management for granular access control

- Helps monetize proprietary data while restricting access

Integration with Other AWS Services

- Amazon S3: storage destination for data sets
- AWS Glue: data cataloging and ETL
- Amazon Athena: query data directly from S3
- Amazon Redshift Spectrum: analyse data without loading into Redshift
- EventBridge: trigger workflows when new data arrives
- AWS CloudTrail: audit access and API calls

Security

- IAM policies control access to Data Exchange resources
- S3 bucket policies secure delivered data
- CloudTrail logging for governance and compliance
- Encryption at rest using S3 server-side encryption

Pricing

- Subscribers pay for data products according to provider pricing (subscription fees)
- AWS charges delivery fees based on data volume delivered to S3
- No upfront fees to browse or subscribe to free data sets

Use Cases

- Incorporating third-party data (market data, demographic data, geospatial data) into analytics workflows
- Monetizing proprietary data by creating data products for others to subscribe
- Simplifying data licensing, entitlement, and billing
- Automating ingestion and processing of external data sources

Best Practices

- Use S3 bucket versioning to track data revisions
- Leverage EventBridge to automate processing pipelines when new data arrives
- Apply granular IAM policies to limit access to sensitive data sets
- Review CloudTrail logs for compliance auditing
- Validate data schema and formats upon delivery to avoid downstream issues
- Use AWS Glue Data Catalog for metadata management and discovery

Exam Tips

- AWS Data Exchange simplifies finding, subscribing to, and using third-party data

- Data is delivered directly into your S3 buckets
- Supports automatic updates when providers publish new revisions
- Integrated with EventBridge for event-driven processing
- Subscription entitlements control access to data sets
- Private products allow sharing data with specific AWS accounts
- Use cases include analytics, ML training, and business intelligence with external data

Quick Summary

AWS Data Exchange enables secure and automated access to third-party and public data sets directly in AWS. It handles subscriptions, entitlement management, delivery to S3, and integrates with AWS analytics and storage services, making it easy to operationalize external data for analytics and machine learning workloads.

AWS Data Pipeline

What It Is

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources. It is used to automate the movement and transformation of data.

Key Features

- Orchestration service for data workflows
- Supports data movement and transformation across AWS services and on-premises
- Allows scheduling of recurring data processing activities
- Fault-tolerant with retry and failure notification options
- Supports dependencies between tasks for complex workflows
- Provides pre-built templates for common data processing scenarios

Components

- **Pipeline**: Defines the overall workflow and schedule
- **Activities**: Units of work like copying data or running EMR jobs
- **Data Nodes**: Define data sources and destinations (e.g., S3, RDS, DynamoDB, on-premises databases)
- **Preconditions**: Optional conditions that must be met before an activity runs
- **Schedules**: Define when and how often pipeline activities run
- **Resources**: Compute resources like EC2 instances or EMR clusters used to execute activities
- **Parameters**: Allow reusing pipeline definitions with different input values

Common Use Cases

- Move data from S3 to Redshift for analytics
- Run periodic ETL jobs on EMR clusters
- Copy data between RDS databases
- Archive data from DynamoDB to S3
- Trigger workflows based on time-based schedules

Supported Services

- Amazon S3
- Amazon RDS
- Amazon DynamoDB
- Amazon Redshift

- Amazon EMR
- On-premises data stores via Data Pipeline Task Runner

Execution Options

- Managed resources (AWS provisions and manages resources automatically)
- Customer-managed resources (users provide and manage the resources)
- On-premises Task Runner for hybrid data workflows

Scheduling

- Cron-like scheduling expressions supported
- Supports hourly, daily, weekly, or custom intervals
- Activities can be chained with dependencies

Fault Tolerance and Retries

- Automatic retry on failures
- Configurable retry policies
- Failure notifications via SNS
- Logging to Amazon S3 and CloudWatch Logs

Security

- IAM roles and policies to control access to pipeline resources
- Supports encryption of data in transit and at rest
- CloudTrail integration for auditing API calls
- VPC integration for controlling network access

Monitoring and Logging

- CloudWatch for monitoring pipeline activity and status
- Logs can be stored in S3 or CloudWatch Logs
- SNS notifications for success or failure events
- Detailed activity logs for debugging and auditing

Pricing

- Free for low-frequency pipelines (one activity per month)
- Charges based on the number of pipeline runs and the frequency of execution
- Costs for resources used (e.g., EC2, EMR) are billed separately

Best Practices

- Use parameters to make pipelines reusable and configurable
- Design pipelines with clear dependencies and failure handling

- Monitor pipeline execution using CloudWatch metrics and logs
- Use preconditions to ensure data quality before processing
- Secure pipeline roles and resources with IAM policies
- Automate alerts for failures using SNS

Exam Tips

- AWS Data Pipeline is used for orchestrating data movement and transformation workflows
- Supports copying data between AWS services and on-premises sources
- Activities define work, Data Nodes define sources and destinations
- Scheduling supports complex recurring workflows
- Integrates with S3, Redshift, RDS, EMR, DynamoDB, and on-premises systems
- Supports fault tolerance with retries and notifications
- Managed and customer-managed resource options available
- Logs and metrics available in CloudWatch and S3

Quick Summary

AWS Data Pipeline enables the reliable movement and transformation of data across AWS and on-premises systems. It provides scheduling, monitoring, fault tolerance, and automation for complex ETL workflows, making it easier to manage data pipelines at scale.

Amazon EMR

What It Is

Amazon EMR (Elastic MapReduce) is a fully managed big data platform that simplifies running large-scale distributed data processing jobs using open-source frameworks like Apache Hadoop, Spark, Hive, HBase, Presto, and others on AWS. It lets you process vast amounts of data quickly and cost-effectively.

Key Features

- Fully managed clusters for Hadoop, Spark, Hive, HBase, Presto, and more
- Quick and easy provisioning of resizable clusters
- Integration with Amazon S3 for data storage
- Decoupled storage and compute using EMRFS
- Cluster scaling with Auto Scaling and manual resizing
- EMR Managed Scaling to automatically adjust resources
- Cost-effective with per-second billing and Spot Instances
- Supports AWS Glue Data Catalog for schema management
- Integration with AWS Lake Formation for data lake permissions
- Integration with AWS Step Functions for orchestration
- Notebook support via EMR Notebooks for interactive analysis

Core Components

- **Cluster**: A collection of Amazon EC2 instances running EMR
- **Master Node**: Manages cluster and tracks job progress
- **Core Nodes**: Process data and store intermediate results
- **Task Nodes**: Process data only, no HDFS storage
- **EMRFS**: Connects EMR to S3 for scalable, durable data storage

Deployment and Management

- Launch clusters quickly via AWS Management Console, CLI, SDKs, or CloudFormation
- Choose EC2 instance types, purchase options (On-Demand, Spot)
- EMR Managed Scaling automatically adjusts nodes based on workload
- Cluster Auto Termination to save costs
- Bootstrap actions to install software and customize nodes at launch
- Supports multi-master for high availability in Hadoop YARN ResourceManager

Data Storage

- EMRFS for S3 integration with eventual consistency improvements
- HDFS for ephemeral storage on cluster nodes
- Supports AWS Glue Data Catalog for Hive Metastore compatibility
- Lake Formation integration for centralized data access control

Processing Frameworks

- Hadoop MapReduce for batch processing
- Apache Spark for fast, in-memory data processing
- Hive for SQL-like queries
- Presto for low-latency SQL queries
- HBase for NoSQL data storage
- Flink for real-time stream processing
- Zeppelin and Jupyter Notebooks for interactive analytics

Security

- IAM roles for fine-grained access control
- AWS Key Management Service (KMS) for encryption at rest
- TLS for encryption in transit
- Kerberos authentication for Hadoop clusters
- Security groups for controlling network access
- Integration with AWS Lake Formation for data permissions
- EC2 key pairs for SSH access to nodes
- AWS CloudTrail for auditing API calls

Monitoring and Logging

- Amazon CloudWatch for metrics and alarms
- CloudWatch Logs for application logs
- EMR console for real-time cluster monitoring
- Ganglia and Spark UI for detailed performance monitoring
- S3 for storing log files

Scaling and Flexibility

- Resize clusters manually or with Auto Scaling
- EMR Managed Scaling automatically scales clusters based on workload
- Support for Spot Instances to reduce cost

- Instance Fleets and Instance Groups for flexible provisioning strategies
- Decoupled compute and storage allows processing directly from S3

Cost Optimization

- Per-second billing for EC2 instances
- Support for Spot Instances for significant cost savings
- Cluster Auto Termination to avoid charges for idle clusters
- Use of Reserved Instances for predictable workloads
- EMR pricing based on instance hours plus EC2 costs

Integrations

- AWS S3 for data lake storage
- AWS Glue for schema management
- AWS Lake Formation for permissions management
- AWS Step Functions for orchestration
- AWS CloudTrail for auditing
- Amazon RDS and Redshift for data sources and sinks
- Kinesis Data Streams and Firehose for streaming ingestion

Use Cases

- Big data processing and ETL workloads
- Data lake analytics with S3
- Machine learning workflows with Spark MLlib
- Log and clickstream analysis
- Real-time stream processing with Flink
- Ad hoc analytics using notebooks
- Batch processing of large datasets

Pricing

- Charges based on EC2 instance hours used
- Additional EMR charge per instance hour
- Per-second billing for EC2 instances
- Costs for S3 storage and data transfer
- Additional costs for AWS Glue, Lake Formation, and other integrated services

Best Practices

- Use Spot Instances for core and task nodes to reduce cost
- Enable EMR Managed Scaling to optimize cluster size automatically
- Store data in S3 with EMRFS for separation of compute and storage
- Use AWS Glue Data Catalog for centralized schema management
- Secure clusters with IAM roles, KMS encryption, and Kerberos
- Monitor cluster health and costs with CloudWatch and CloudTrail
- Automate cluster creation and workflows with Step Functions and CloudFormation

Exam Tips

- Fully managed big data platform supporting Hadoop, Spark, Hive, Presto, HBase
- EMRFS enables direct read/write to S3
- Managed Scaling automatically resizes cluster based on workload
- Spot Instances reduce cost significantly
- Supports encryption in transit (TLS) and at rest (KMS)
- IAM roles control access to resources
- Glue Data Catalog integration for Hive compatibility
- Decoupled compute and storage architecture
- Use Case: Log processing, ETL, machine learning, data lake analytics

Quick Summary

Amazon EMR is AWS's managed big data platform for running open-source frameworks like Hadoop and Spark on scalable EC2 clusters, tightly integrated with S3 and other AWS services, with strong security, flexibility, and cost optimization features for processing massive datasets.

AWS Glue

What It Is

AWS Glue is a fully managed serverless data integration service that makes it easy to discover, prepare, move, and integrate data for analytics, machine learning, and application development. It provides a central platform for data cataloging, ETL (extract, transform, load), and data integration across various AWS services and data sources.

Key Features

- Fully managed, serverless architecture
- Automated ETL using Apache Spark
- AWS Glue Data Catalog as a persistent metadata store
- Visual and code-based job authoring
- Crawlers to automatically discover and catalog data
- Support for schema versioning and evolution
- Data preparation with AWS Glue DataBrew
- Integration with AWS Lake Formation for access control
- Job scheduling and orchestration
- Supports Python and Scala for custom ETL scripts
- Built-in transforms and connectors for popular data stores

AWS Glue Data Catalog

- Centralized metadata repository
- Stores table definitions, schemas, and job metadata
- Integrated with AWS services like Athena, Redshift Spectrum, EMR, and Lake Formation
- Supports schema versioning and evolution
- Can catalog data in S3, JDBC sources, DynamoDB, and more
- Enables discovery through AWS Glue Crawlers

AWS Glue Crawlers

- Automatically scan data sources to infer schema and create catalog tables
- Support multiple data stores (S3, JDBC, DynamoDB)
- Can be scheduled or triggered on demand
- Handle schema changes over time with versioning

ETL Jobs

- Serverless ETL using Apache Spark
- Can be created with visual job editor or by writing code in Python/Scala

- Built-in transforms like mapping, filtering, joining, partitioning
- Support for custom transforms and libraries
- Can read from and write to various data sources (S3, RDS, Redshift, DynamoDB, JDBC)
- Job bookmarks to track processed data and avoid reprocessing

AWS Glue DataBrew

- Visual data preparation tool
- No-code interface to clean and normalize data
- Over 250 built-in transforms for formatting, deduplication, validation
- Can process data stored in S3, Redshift, Athena, and other sources
- Automates creation of transformation recipes for reuse

AWS Glue Studio

- Visual interface for authoring, running, and monitoring ETL jobs
- Drag-and-drop nodes to define ETL flow
- Supports complex transformations without coding
- Allows code editing for advanced use cases
- Integrated monitoring and job runs management

Triggers and Workflows

- Schedule jobs or run them on-demand
- Event-driven triggers based on data arrival or other job completion
- Workflows to coordinate multiple ETL jobs and crawlers
- Visual interface to design and monitor workflows

Integration with Other AWS Services

- Amazon S3 for storage and data lakes
- Amazon Redshift for data warehousing
- Amazon RDS and Aurora for relational sources
- AWS Lake Formation for data lake permissions
- AWS Athena for querying cataloged data
- Amazon EMR for big data processing
- AWS Step Functions for orchestration of complex workflows
- AWS CloudWatch for monitoring and logging

Security

- IAM roles and policies for fine-grained access control

- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- Integration with AWS Lake Formation for table-level permissions
- VPC support for private network access
- CloudTrail logging for auditing API calls

Pricing

- Pay-per-use pricing model
- Charged based on Data Processing Units (DPUs) per job hour
- Separate pricing for crawlers, Data Catalog storage, and DataBrew usage
- No infrastructure to manage or provision

Use Cases

- Building and managing data lakes on AWS
- Data discovery and cataloging for analytics
- ETL pipelines for data warehouses like Redshift
- Preparing data for machine learning workflows
- Real-time event-driven ETL triggered by S3 uploads
- Data cleaning and preparation with DataBrew

Best Practices

- Use Glue Crawlers to keep Data Catalog updated with schema changes
- Take advantage of Job Bookmarks to avoid reprocessing duplicate data
- Use DataBrew for data preparation without coding
- Encrypt data in transit and at rest
- Integrate with AWS Lake Formation for fine-grained data lake security
- Monitor jobs and workflows with CloudWatch
- Optimize Spark jobs by tuning DPUs and partitioning strategies

Exam Tips

- Serverless ETL service with Apache Spark under the hood
- AWS Glue Data Catalog is a persistent, central metadata store
- Crawlers automatically discover schema and populate the catalog
- Jobs can be authoring visually (Glue Studio) or with code
- DataBrew is the no-code visual tool for preparing data
- Integration with Lake Formation for security and permissions

- Pay per DPU-hour, no infrastructure management
- Supports job triggers and workflows for orchestration
- Tight integration with AWS analytics and storage services

Quick Summary

AWS Glue is AWS's serverless data integration service providing ETL capabilities, data cataloging, schema discovery, and data preparation. It enables users to build scalable, cost-effective data pipelines and manage metadata centrally for a wide range of analytics and machine learning use cases on AWS.

Amazon Kinesis

What It Is

- Fully managed service for real-time processing of streaming data at massive scale.
- Enables you to collect, process, and analyse data streams in near real-time.
- Supports use cases like log analytics, event data processing, IoT telemetry, and video streams.

Kinesis Data Streams (KDS)

- Real-time streaming service for continuous capture of gigabytes of data per second.
- Producers send records to the stream.
- Data is stored in shards.
    - Each shard: 1 MB/s write, 2 MB/s read.
- Records retained for 24 hours by default, up to 7 days (extended retention).
- **Consumers**:
    - AWS Lambda for processing.
    - Kinesis Client Library (KCL) applications.
    - Supports enhanced fan-out (dedicated throughput per consumer).
- **Partition Key**:
    - Determines which shard the data lands in.
- **Encryption**:
    - Supports server-side encryption with AWS KMS.
- **Use Cases**:
    - Real-time dashboards
    - Log and event collection
    - Clickstream analysis

Kinesis Data Firehose

- Fully managed, no-shards-to-manage service to reliably load streaming data to destinations:
    - S3
    - Redshift
    - Elasticsearch Service (now OpenSearch)
    - Splunk
    - Custom HTTP endpoints

- Supports data transformation using AWS Lambda.
- Automatic scaling.
- Near real-time delivery (typical latency ~60 seconds).
- Built-in compression and encryption.
- No persistent storage of data in Firehose itself.
- Use Cases:
    - Loading logs to S3 for analytics
    - Near-real-time reporting
    - Streaming ETL pipelines

Kinesis Data Analytics

- Fully managed service to analyse streaming data using SQL.
- Ingests data from Kinesis Data Streams or Firehose.
- Runs SQL queries continuously against streaming data.
- Supports real-time aggregation, filtering, and anomaly detection.
- Can output to:
    - Kinesis Data Streams
    - Kinesis Data Firehose
    - AWS Lambda
- **Use Cases:**
    - Real-time metrics and alerts
    - Time-series analytics
    - Continuous data transformation

Kinesis Video Streams

- Capture, process, and store video streams for analytics and machine learning.
- Supports live and on-demand video ingestion.
- Automatically encrypts and indexes video data.
- **Integrates with:**
    - AWS Rekognition Video for analysis
    - Custom ML workflows
- Provides time-indexed video storage in S3.
- **Use Cases:**
    - Security camera feeds

    - Video-enabled IoT devices
    - Machine learning on video data

Data Retention and Replay

- **Kinesis Data Streams**:
    - Retains data for 24 hours by default, configurable up to 7 days.
    - Allows reprocessing of older data (replay capability).
- Firehose:
    - Does not store data, delivers directly to destinations.

Security

- Supports Server-Side Encryption (SSE) using AWS KMS.
- IAM policies control access to streams, Firehose delivery streams, analytics applications.
- VPC endpoints available for private access.

Scaling

- **Kinesis Data Streams**:
    - Scale by adding/removing shards.
    - Resharding supported (split/merge).
    - On-Demand mode for automatic scaling.
- **Firehose**:
    - Fully managed, auto-scales.
- **Data Analytics**:
    - Managed scaling, charges based on processing units (KPU).

Pricing

- Based on:
    - Number of shards (KDS)
    - PUT payload units
    - Data volume delivered (Firehose)
    - Processing units (Analytics)
    - Video streaming duration and storage

Kinesis Service Comparison

Service	Purpose	Data Storage	Scaling	Use Cases
Kinesis Data Streams	Real-time data ingestion and custom processing	Stored in shards (24h–7d retention)	Manual shard scaling or On-Demand	Real-time dashboards, log collection, clickstreams
Kinesis Data Firehose	Fully managed delivery to destinations	No storage (streams directly)	Auto-scales	Load data to S3, Redshift, OpenSearch, near real-time ETL
Kinesis Data Analytics	SQL-based analysis of streaming data	Processes input streams (no long-term storage)	Managed scaling via KPUs	Real-time metrics, continuous transformations
Kinesis Video Streams	Ingest and store video for analysis	Stores time-indexed video in S3	Managed ingestion, built for video	Security camera feeds, IoT video, ML analysis

Integration

- Lambda: Event source for processing.
- S3: Destination via Firehose.
- Redshift: Direct loading via Firehose.
- OpenSearch (Elasticsearch): Index streaming data.
- AWS Rekognition: Analyse video streams.

Exam Tips

- **Kinesis Data Streams**: Real-time, ordered, sharded. Best for real-time analytics with custom processing.
- **Firehose**: Simplest way to load data to S3, Redshift, OpenSearch. Fully managed, near real-time.
- **Data Analytics**: Run continuous SQL on streaming data.
- **Video Streams**: Ingest and analyse video feeds.
- Shards determine throughput in KDS.
- Enhanced fan-out provides dedicated throughput per consumer.

Quick Summary

Amazon Kinesis is a suite of services for real-time streaming data ingestion, processing, analysis, and delivery. It includes Kinesis Data Streams (sharded streams), Firehose (managed delivery), Data Analytics (SQL processing), and Video Streams (video ingestion and analysis), enabling you to build scalable, real-time data applications.

Amazon LakeFormation

What it is

A fully managed service that simplifies setting up, securing, and managing data lakes. Builds, catalogs, and secures data in Amazon S3 for analytics and ML. Integrates tightly with AWS Glue, Athena, Redshift Spectrum, and EMR.

Key Features

- **Centralized Data Lake Setup:** Registers S3 buckets as data lake locations. Automatically crawls and catalogs data. Grants fine-grained access controls.
- **Data Catalog:** Unified metadata catalog shared with AWS Glue. Defines tables, schemas, and partitions for analytics tools.
- **Access Control:** Fine-grained permissions at table, column, and row level. Permissions managed within Lake Formation, separate from IAM. Integrates with AWS IAM and AWS KMS for security.
- **BluePrints & Workflows:** Pre-built ETL blueprints automate data ingestion (e.g., from RDS or DynamoDB). Workflows orchestrate multiple jobs and data transformations.
- **Data Filtering:** Applies row- and column-level filters without duplicating data.
- **Transaction Support:** Supports ACID transactions for governed tables (insert/update/delete consistency).
- **Storage Optimization:** Uses Amazon S3 as the underlying data store. Manages metadata, versioning, and schema evolution.
- **Integration:** Works with Athena, Redshift Spectrum, QuickSight, Glue, and SageMaker. Supports federated queries from external sources.

How It Works

- You register data locations (S3 paths) and define permissions in Lake Formation.
- Data is crawled and cataloged into the Data Catalog (shared with AWS Glue).
- Permissions determine which principals (users, roles) can access what data.
- Authorized analytics services query the data directly from S3 using governed access.

Use Cases

- Centralize and govern enterprise data in one place.
- Simplify onboarding of new data sources for analytics.
- Apply fine-grained data access policies for different teams.
- Enable self-service analytics with consistent security controls.
- Manage ACID tables for data warehouses and ML pipelines.

Pricing

- Pay per transaction (read/write) on governed tables.

- Pay for metadata storage and access requests.
- Underlying S3, Glue, and query service charges still apply.

Security and Compliance

- Integrates with IAM, CloudTrail, and KMS for full governance.
- Data access logs through CloudTrail.
- Fine-grained permissions reduce data exposure.
- Regional service,data stays within its AWS region.

Exam Tips

- Lake Formation = simplified, secure data lake management on S3.
- Shares catalog with AWS Glue.
- Manages row/column-level permissions centrally.
- Integrates with Athena, Redshift Spectrum, and EMR.
- Governed tables = ACID compliance and transaction support.
- Permissions in Lake Formation are separate from IAM policies.
- Data filtering avoids data duplication.
- Use blueprints for automated ingestion from RDS or DynamoDB.
- CloudTrail logs all data access events for auditing.

Quick Summary

AWS Lake Formation helps you quickly build and secure a centralized data lake on S3, with fine-grained access control, governance, and integrated analytics access, making it easier to manage large-scale data securely across AWS services.

Amazon MSK

What It Is

Amazon Managed Streaming for Apache Kafka (Amazon MSK) is a fully managed service that makes it easy to build and run applications that use Apache Kafka to process streaming data. It removes the operational overhead of setting up, scaling, and managing Kafka clusters while offering high availability, security, and integration with AWS services.

Key Features

- Fully managed Apache Kafka service
- Supports open-source Kafka APIs
- Automated provisioning, configuration, and maintenance
- Broker patching, monitoring, and failure recovery built in
- High availability with replication across multiple Availability Zones
- Compatible with existing Kafka tools and client libraries
- AWS CLI, SDK, and Console support for management
- Serverless option with MSK Serverless for automatic scaling

MSK Cluster Components

- **Brokers**: Kafka server nodes handling ingestion and replication
- **ZooKeeper**: Coordinates brokers and maintains cluster metadata
- **Storage**: Elastic storage options, backed by EBS volumes
- **Networking**: VPC integration, private endpoints, security groups
- **Monitoring**: Integrated with CloudWatch metrics and logging

MSK Serverless

- No need to provision or manage brokers
- Automatically scales up and down based on data volume and throughput
- Pay only for data ingested, stored, and throughput used
- Ideal for variable or unpredictable workloads
- Supports same Kafka APIs, clients, and tooling

Integration with AWS Services

- AWS Lambda for real-time stream processing
- AWS Glue Schema Registry for enforcing schemas on data streams
- Amazon Kinesis Data Analytics for SQL-based stream processing
- Amazon S3 for storing processed data
- AWS Identity and Access Management (IAM) for managing permissions

- AWS CloudWatch for monitoring and alerting
- AWS Secrets Manager for storing credentials

Security

- Encryption at rest using AWS KMS
- Encryption in transit using TLS between clients and brokers
- IAM integration for authentication and access control
- TLS-based mutual authentication supported
- Private connectivity via AWS VPC
- AWS Secrets Manager integration for managing credentials

Monitoring and Logging

- Amazon CloudWatch metrics for broker performance, storage, throughput
- AWS CloudTrail for auditing API calls
- Kafka broker logs can be streamed to CloudWatch Logs or Amazon S3
- JMX and Node metrics for fine-grained Kafka monitoring
- AWS CloudFormation support for infrastructure as code

Performance and Scaling

- Horizontal scaling by adding brokers
- Storage scaling via EBS volume adjustments
- MSK Serverless scales automatically based on demand
- Multi-AZ replication for high availability and durability
- Custom configurations supported via MSK configurations

MSK Versions

- Supports multiple versions of open-source Apache Kafka
- Option to choose versions during cluster creation
- AWS manages upgrades and patching

Pricing

- MSK Standard clusters: Charged per broker-hour and storage used
- MSK Serverless: Charged per partition-hour, data volume, and throughput
- Data transfer charges may apply for cross-AZ traffic
- No upfront costs or long-term commitments required

Use Cases

- Real-time analytics and event processing

- Log aggregation pipelines
- Website activity tracking
- Messaging between microservices
- IoT data ingestion and processing
- Data lake ingestion workflows

Best Practices

- Use IAM policies for least-privilege access control
- Enable encryption at rest and in transit
- Monitor with CloudWatch and set up alarms
- Plan broker count and storage for expected throughput
- Use MSK Serverless for unpredictable workloads without managing capacity
- Store sensitive configuration details in AWS Secrets Manager
- Design topics and partitions for scalability and consumption patterns

Exam Tips

- Fully managed service for Apache Kafka on AWS
- MSK Serverless simplifies scaling and cost management
- Supports encryption at rest (AWS KMS) and in transit (TLS)
- Integrated with VPC for private networking
- Monitoring via CloudWatch, CloudTrail, and Kafka metrics
- IAM and TLS mutual authentication for access control
- Common use cases include real-time analytics, microservice communication, and log ingestion
- Compatible with open-source Kafka tooling and client libraries

Quick Summary

Amazon MSK is AWS's fully managed service for running Apache Kafka without the operational overhead. It offers secure, highly available, and scalable streaming data processing with integrations across AWS services, supporting both provisioned clusters and serverless deployments for flexible, cost-effective streaming workloads.

Amazon OpenSearch

What It Is

Amazon OpenSearch Service is a fully managed service that makes it easy to deploy, operate, and scale OpenSearch and legacy Elasticsearch clusters in the AWS Cloud. It is used for search, log analytics, and real-time application monitoring.

Key Features

- Fully managed service for OpenSearch and Elasticsearch
- Supports versions of OpenSearch and legacy Elasticsearch (up to 7.10)
- In-place version upgrades
- Integrated Kibana (Elasticsearch) and OpenSearch Dashboards
- Built-in integrations with AWS services (CloudWatch, Kinesis, S3)
- Provides domain-level security with fine-grained access control
- Supports encryption at rest and node-to-node encryption
- VPC support for secure network access
- Auto-Tune for performance optimization
- Ultrawarm and cold storage for cost-effective storage of infrequently accessed data

Core Concepts

- **Domain**: An OpenSearch or Elasticsearch cluster in AWS
- **Index**: Collection of documents with similar characteristics
- **Document**: Basic unit of information in JSON format
- **Shard**: Horizontal partition of an index
- **Replica**: Copy of a shard for high availability

Deployment and Scaling

- Supports cluster scaling by adjusting instance count and types
- Data nodes, dedicated master nodes, Ultrawarm nodes, cold storage
- Automated snapshots to S3 for backups
- Manual and scheduled snapshots available
- Supports multi-AZ deployments for high availability

Ultrawarm and Cold Storage

- **Ultrawarm:** Cost-effective storage tier for read-only, infrequently accessed data
- **Cold Storage:** Even lower-cost storage for long-term retention
- Seamlessly search across hot, Ultrawarm, and cold data tiers

Security

- IAM policies for domain-level access control
- Fine-grained access control using OpenSearch security plugin
- Cognito integration for user authentication to dashboards
- Encryption at rest using AWS KMS
- Node-to-node encryption for secure cluster communication
- HTTPS for data in transit
- VPC support for private network connectivity

Monitoring and Logging

- Integrated with Amazon CloudWatch for metrics and alarms
- Detailed logs to CloudWatch Logs
- Slow logs for indexing and search performance analysis
- Audit logs for security compliance
- AWS CloudTrail for API call logging

Data Ingestion

- Supports ingesting data via:
    - Amazon Kinesis Data Firehose
    - Logstash
    - Fluentd
    - Beats
    - Custom ingestion pipelines
- Supports OpenSearch ingest nodes for processing data on ingestion

Querying and Analytics

- Powerful full-text search capabilities
- Aggregations for analytics queries
- SQL support for querying data
- Piped Processing Language (PPL) for easier queries
- OpenSearch Dashboards for visualizing data

Integrations

- Amazon CloudWatch for monitoring and alerts
- AWS Kinesis Data Firehose for streaming ingestion
- AWS Lambda for serverless ingestion pipelines

- Amazon S3 for snapshot storage and cold storage
- AWS Cognito for dashboard authentication

Use Cases

- Application search
- Log and event analytics
- Infrastructure monitoring
- Security information and event management (SIEM)
- Real-time application and system monitoring
- Business and operational dashboards

Pricing

- Charges based on instance hours, storage, and data transfer
- Ultrawarm and cold storage tiers reduce cost for infrequently accessed data
- Data transfer within the same region is free for ingestion
- Snapshot storage in S3 incurs S3 costs

Best Practices

- Choose appropriate instance types and storage tiers
- Use Ultrawarm and cold storage for older, less-frequently queried data
- Apply fine-grained access controls and encryption
- Monitor performance using CloudWatch and slow logs
- Scale clusters based on ingestion rates and query workloads

Exam Tips

- Fully managed service for OpenSearch and Elasticsearch
- Supports log analytics and real-time search use cases
- Integrates with Kinesis Data Firehose, CloudWatch, and Cognito
- Ultrawarm and cold storage for cost-effective long-term retention
- Security features include encryption at rest, node-to-node encryption, and IAM
- Automated snapshots stored in S3
- VPC support for private network connectivity

Quick Summary

Amazon OpenSearch Service is AWS's fully managed search and analytics service supporting OpenSearch and legacy Elasticsearch, designed for log analytics, monitoring, and real-time search, with strong security, scalability, and integration with AWS services.

Amazon QuickSight

What It Is

Amazon QuickSight is a fully managed business intelligence (BI) service that enables you to easily create and publish interactive dashboards, visualizations, and analyses from various data sources. It is designed to be scalable, serverless, and cost-effective, with pay-per-session pricing for readers.

Key Features

- Managed, serverless BI service with no infrastructure to manage
- Interactive dashboards with rich visualizations and drill-downs
- Supports ad hoc analysis and sharing of insights
- Pay-per-session pricing model for readers
- SPICE (Super-fast, Parallel, In-memory Calculation Engine) for fast performance
- Supports scheduled and email reports
- ML Insights for anomaly detection and forecasting
- Integration with AWS data sources and on-premises databases

Data Sources

- AWS services: S3, Athena, Redshift, RDS, Aurora, EMR, Glue Data Catalog
- On-premises databases via QuickSight data sources
- Third-party databases: MySQL, PostgreSQL, SQL Server, Oracle, Teradata
- Files: CSV, Excel, JSON
- SaaS connectors: Salesforce, ServiceNow, Twitter, and others

SPICE Engine

- In-memory storage and calculation engine
- Enables fast, interactive querying even on large datasets
- Automatic scaling to support thousands of users
- Data can be refreshed on schedule or on demand
- Cost-effective compared to direct querying of data sources

Dashboard and Visualization

- Rich set of visual types: bar, line, pie charts, maps, tables, KPIs
- Supports drill-down and drill-through for exploration
- Parameterized dashboards for customization
- Filters and controls for interactive exploration
- Embedding dashboards into apps and portals

ML Insights

- Built-in machine learning capabilities without needing ML expertise
- Anomaly detection to identify unexpected changes
- Forecasting to predict future trends using ML models
- Narrative Insights to automatically generate textual summaries of data

User Management and Access Control

- Supports AWS IAM for authentication and access management
- Role-based access control within QuickSight
- Integration with AWS SSO and Active Directory for enterprise sign-in
- Row-level security to restrict data access within datasets

Deployment Options

- QuickSight Standard Edition for individuals and small teams
- QuickSight Enterprise Edition for advanced features, security, and enterprise integrations
- Pay-per-session pricing for readers to reduce costs for infrequent users
- SPICE capacity purchased separately for high-speed querying

Security

- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- VPC connectivity for secure access to private data sources
- Compliance with AWS security and compliance standards

Integration with AWS Services

- Direct integration with AWS data sources like Redshift, Athena, S3, RDS
- AWS Glue Data Catalog for schema and metadata management
- Integration with AWS IoT Analytics and other analytics services
- Ability to embed dashboards into custom applications using AWS SDKs

Monitoring and Management

- AWS CloudWatch for usage metrics and alarms
- Usage tracking for pay-per-session billing
- Audit logging via AWS CloudTrail
- Centralized management of users and permissions

Pricing

- Author pricing: fixed monthly fee per author user
- Reader pricing: pay-per-session or monthly pricing model
- SPICE storage pricing based on GB stored per month
- No upfront costs or infrastructure to manage

Use Cases

- Business intelligence dashboards and reporting
- Operational analytics with near real-time data
- Embedded analytics for SaaS applications
- Executive dashboards and KPI monitoring
- Self-service data exploration for business users

Exam Tips

- QuickSight is AWS's managed BI service with no servers to manage
- SPICE provides fast, in-memory query performance
- Supports pay-per-session pricing for cost-effective sharing
- Integrates deeply with AWS data sources like Athena, Redshift, and S3
- ML Insights add built-in anomaly detection and forecasting
- Supports embedding dashboards into apps for custom experiences
- Role-based access and row-level security control data access
- Can connect to on-premises and third-party databases via connectors

Quick Summary

Amazon QuickSight is a serverless, fully managed BI service that delivers fast, interactive dashboards and visualizations using the SPICE engine. It integrates with a wide range of AWS and external data sources, supports pay-per-session pricing, and offers ML-powered insights to help organizations make data-driven decisions at scale.

Amazon Redshift

What It Is

Amazon Redshift is a fully managed, petabyte-scale cloud data warehouse service designed for analysing large datasets using standard SQL and existing Business Intelligence (BI) tools. It offers high performance, scalability, and cost-effectiveness for data analytics workloads.

Key Features

- Columnar storage format optimized for analytics
- Massively Parallel Processing (MPP) architecture for performance
- Supports standard SQL and integrates with common BI tools
- Automated backups and snapshots
- Redshift Spectrum enables querying data in S3 without loading it into Redshift
- AQUA (Advanced Query Accelerator) improves query performance using hardware acceleration
- Built-in security features including encryption, VPC, IAM, and logging
- Compatible with PostgreSQL drivers and clients

Data Warehouse Architecture

- Redshift cluster consists of a leader node and one or more compute nodes
- Leader node receives queries and distributes them to compute nodes
- Compute nodes process the queries in parallel and return results to the leader node
- Supports provisioned and serverless deployment modes

Redshift Serverless

- Allows running analytics without managing infrastructure
- Automatically provisions and scales compute capacity
- Charges based on Redshift Processing Units (RPUs) used
- Suitable for variable or unpredictable workloads
- Easy to set up, integrates with data in Redshift-managed storage and S3

Data Loading

- Load data using COPY command from S3, DynamoDB, EMR, or other sources
- Supports parallel loading for high performance
- Can use AWS Glue Data Catalog for schema discovery
- Supports SQL-based INSERT statements for small-scale operations

Redshift Spectrum

- Query data in S3 directly using Redshift SQL

- Supports open data formats like CSV, Parquet, ORC, JSON, and Avro
- Integrates with AWS Glue Data Catalog for schema definitions
- Ideal for extending queries beyond data stored in Redshift clusters

Performance Optimization

- Use of distribution styles (KEY, EVEN, ALL) to manage data distribution across nodes
- Sort keys optimize query performance on large tables
- Compression (encoding) reduces storage usage and speeds up processing
- Materialized views store precomputed query results for faster performance
- Workload management (WLM) to prioritize and allocate query resources

Scalability and Elasticity

- Elastic resize for quick cluster scaling
- Concurrency scaling to handle spikes in query load
- RA3 instances with managed storage to scale compute and storage independently
- Cross-region data sharing supported using datashares

Security

- Data encryption at rest using AWS KMS or hardware security modules (HSM)
- SSL encryption for data in transit
- VPC support for network isolation
- IAM for access control and resource policies
- Audit logging with integration to Amazon CloudWatch and AWS CloudTrail

Monitoring and Management

- Performance Insights for query analysis and tuning
- CloudWatch integration for metrics, logs, and alarms
- Automatic backup to S3 with user-defined retention
- Snapshot-based restore options including cross-region snapshots
- Maintenance windows for patching and upgrades

Pricing

- Based on instance type and node count for provisioned clusters
- Redshift Serverless pricing based on RPU-seconds used
- Separate charges for backup storage, concurrency scaling, and data transfer
- No cost for querying S3 data via Redshift Spectrum (only standard S3 and query charges apply)

Use Cases

- Enterprise data warehousing and analytics
- Centralized data platform for BI reporting
- Large-scale log analytics and operational dashboards
- Predictive analytics and machine learning data prep
- Federated queries across data lakes and warehouses

Exam Tips

- Redshift uses columnar storage and MPP for performance
- COPY command is used to bulk-load data from S3
- Spectrum allows querying S3 data directly using Redshift
- RA3 nodes separate compute from storage for flexibility
- Concurrency scaling adds transient clusters for high query loads
- Redshift Serverless offers hands-free provisioning and scaling
- Use sort and distribution keys to optimize query performance
- Supports VPC, encryption, IAM, and logging for security compliance

Quick Summary

Amazon Redshift is AWS's fully managed data warehouse that supports petabyte-scale analytics workloads using standard SQL. With features like columnar storage, MPP architecture, Redshift Spectrum, serverless deployment, and deep AWS integration, it enables fast, scalable, and cost-effective data analysis for modern business needs.

Amazon AppFlow

What It Is

Amazon AppFlow is a fully managed integration service that automates data flows between SaaS applications and AWS services.

- **No code required to set up secure data transfers.**

Key Features

- Supports popular SaaS apps: Salesforce, ServiceNow, Slack, Google Analytics, Marketo, Zendesk, etc.
- Connects to AWS services like S3, Redshift, Salesforce, EventBridge.
- Bidirectional flows: Ingest data to AWS or send data from AWS to SaaS.
- On-demand or scheduled transfers.
- Event-triggered flows supported.
- Data transformations (masking, mapping, validation) built in.
- PrivateLink support for secure private connections.
- Handles encryption at rest and in transit.

Use Cases

- Sync customer data from Salesforce to S3 for analytics.
- Load marketing data to Redshift.
- Automate ticket data from Zendesk to S3.
- Move ServiceNow records to an AWS data lake.

Benefits

- No servers to manage.
- Reduces need for custom integration code.
- Secure, private data transfer via AWS PrivateLink.
- Scales automatically.
- Ensures data consistency and integrity.

Pricing

- Pay per flow run.
- Additional costs for data processed.
- No upfront fees.

Integration

- Works inside your AWS account.

- Integrates with IAM for access control.
- Supports CloudWatch for monitoring.

Exam Tips

- Managed integration service for SaaS ↔ AWS.
- Supports scheduled, on-demand, event-triggered flows.
- Transforms data during transfer.
- Uses PrivateLink for secure connectivity.
- Alternative to building custom ETL jobs or APIs for data movement.

Quick Summary

Amazon AppFlow simplifies secure data movement between SaaS apps and AWS without any custom coding. It's scalable, automated, and integrates seamlessly with core AWS services, making it a go-to solution for building reliable, serverless data pipelines for analytics and business workflows.

AWS AppSync

What It Is

AWS AppSync is a fully managed GraphQL API service to easily build, manage, and secure APIs that access, combine, and serve data from multiple sources.

Key Features

- Automatically generates a GraphQL API endpoint.
- Connects to multiple data sources: DynamoDB, Aurora, Lambda, HTTP APIs, Elasticsearch/OpenSearch, RDS.
- Supports real-time data with subscriptions and WebSocket.
- Handles queries, mutations, and subscriptions.
- Built-in caching for performance.
- Fine-grained access control with AWS IAM, Cognito, API key, OpenID Connect.
- Data merging and resolvers for combining sources.
- Offline support for mobile and web apps.

How It Works

- Client sends GraphQL queries to AppSync.
- AppSync uses resolvers to fetch and transform data from backend sources.
- Returns a single, optimized response.
- Supports real-time updates to clients via subscriptions.

Use Cases

- Unified API for multiple data sources.
- Mobile and web apps needing real-time data.
- Aggregating data from DynamoDB, Lambda, RDS in one API.
- Chat applications, dashboards, collaborative tools.

Benefits

- Reduces backend complexity single GraphQL endpoint.
- Built-in real-time and offline support.
- Scalable, secure, and fully managed.
- Integrates with AWS services for easy development.

Pricing

- Pay for queries, mutations, subscriptions, and data transfer.
- Optional caching incurs extra cost.

- No upfront fees.

Integration

- Works with AWS IAM, Cognito for auth.
- Monitored via CloudWatch.
- Can trigger Lambda for custom logic.

Exam Tips

- Managed GraphQL API service.
- Supports real-time subscriptions.
- Integrates with DynamoDB, Lambda, Aurora, HTTP APIs.
- Great for apps needing real-time, offline, and unified data.
- Alternative to API Gateway for GraphQL workloads.

Quick Summary

AWS AppSync streamlines API development by unifying data from multiple sources through a single GraphQL endpoint. With built-in real-time, offline, and security features, it's the ideal choice for modern, scalable apps that demand dynamic and responsive data interactions without heavy backend management.

AWS EventBridge

What It Is

- Serverless event bus service for building event-driven applications.
- Ingests, filters, routes, and delivers events from:
    - AWS services
    - Your own applications
    - SaaS partners

Core Concepts

1. Event Bus

- Logical pipeline that receives events.
- **Types of event buses:**
    - **Default Event Bus:**
        - Receives events from AWS services.
    - **Custom Event Bus:**
        - For your own applications.
    - **Partner Event Bus:**
        - For SaaS integrations.

2. Events

- JSON-formatted records describing changes in AWS resources or app state.
- **Example:**
    - S3 object created
    - EC2 instance state change
    - Custom application events

3. Rules

- Define event patterns to match incoming events.
- Determine targets that should receive matched events.
- Can transform event before sending to target.

4. Targets

- Services that process events.
- **Examples:**
    - Lambda
    - SNS

- SQS
- Step Functions
- Kinesis Data Streams
- EC2 instances via Systems Manager Run Command
- EventBridge API Destinations (HTTP endpoints)

5. Event Patterns

- JSON structure that filters events based on their contents.
- Rules use patterns to decide which events to route.

6. Event Archive and Replay

- **Archive:**
  - Store events for later analysis or compliance.
- **Replay:**
  - Re-send stored events to the event bus.
  - Useful for testing new rules or recovering from failures.

7. Schema Registry

- Automatically discovers and stores event schemas.
- Developers can view schemas for consistent parsing.
- Supports downloading code bindings for events in popular languages.

8. API Destinations

- Send events to HTTP endpoints outside AWS.
- Supports setting authorization with API keys or OAuth.
- Enables integration with external SaaS or custom APIs.

9. Cross-Account Event Delivery

- Send events between AWS accounts.
- Useful for centralizing events in a single account.
- Supports resource policies to allow cross-account access.

10. Partner Integrations

- SaaS providers can publish events directly to your Partner Event Bus.
- **Examples:**
  - Zendesk
  - Datadog
  - PagerDuty

- Enables seamless integration with third-party systems.

Key Features

- **Fully managed:** No servers to manage or scale.
- Near real-time event delivery.
- Highly scalable and reliable.
- Integrates with over 200 AWS services.
- Supports event filtering, transformation, and routing.

Use Cases

- Application decoupling with event-driven architecture.
- Real-time notifications and automation.
- Integrating AWS services with SaaS providers.
- Auditing and compliance through event archiving.
- Triggering workflows in response to resource changes.

Pricing

- Charged per event published and event matched.
- Separate charges for archived events and replays.
- Free tier available with millions of free events per month.

EventBridge vs SNS

- **SNS:**
    - Simple pub/sub with push delivery.
    - Broad, unfiltered delivery to subscribers.
- **EventBridge:**
    - Advanced filtering and routing rules.
    - Supports SaaS integration and schema registry.
    - Event archive and replay.
    - Designed for event-driven application architecture.

Exam Tips

- Default Event Bus = AWS service events.
- Custom Event Bus = your apps' events.
- Partner Event Bus = SaaS provider events.
- Rules filter events and route to targets.
- Supports multiple targets per rule.

- Archive and replay = compliance, testing, recovery.
- Schema Registry = discover, manage, and generate code for event schemas.
- API Destinations = integrate with external HTTP endpoints.
- Cross-account delivery = send events securely between accounts.
- Choose EventBridge over SNS when you need advanced filtering, routing, SaaS integration, or event archiving.

Quick Summary

Amazon EventBridge is a serverless event bus that connects AWS services, your applications, and SaaS providers using events. It supports advanced filtering, routing, archiving, and replay for building scalable, event-driven systems.

AWS MQ

What It Is

Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ. It makes it easy to set up, operate, and scale message brokers in the cloud. It supports industry-standard APIs and protocols so applications can easily migrate without rewriting messaging code.

Key Features

- Fully managed message broker service
- Supports Apache ActiveMQ and RabbitMQ engines
- Compatible with industry-standard APIs and protocols such as JMS, NMS, AMQP, STOMP, MQTT, and WebSocket
- Simplifies migration of existing messaging applications to AWS without code changes
- Provides message durability and high availability
- Supports failover and replication for fault tolerance
- Integrates with AWS monitoring and security services

Broker Engines Supported

- Apache ActiveMQ
- RabbitMQ

Protocols and APIs

- JMS (Java Message Service)
- NMS (.NET Messaging Service)
- AMQP (Advanced Message Queuing Protocol)
- STOMP (Simple Text Oriented Messaging Protocol)
- MQTT (Message Queuing Telemetry Transport)
- WebSocket

High Availability and Durability

- Supports replication for fault tolerance
- Active/standby brokers in different Availability Zones for failover
- Data replicated across multiple Availability Zones
- Automatic failover for high availability deployments
- Durable queues to persist messages

Deployment Options

- Single-instance brokers for development or low-cost use cases
- Active/standby brokers with multi-AZ replication for production workloads

- RabbitMQ clusters with multiple nodes for increased availability and throughput

Scaling

- Vertical scaling by selecting larger instance types
- Supports clustering with RabbitMQ for horizontal scaling
- No need to manually manage broker software or infrastructure

Security

- Encryption at rest using AWS KMS
- Encryption in transit using TLS
- VPC support for network isolation
- IAM for controlling access to Amazon MQ resources
- AWS Secrets Manager integration for storing credentials
- Access Control Lists (ACLs) for fine-grained user permissions

Monitoring and Management

- Integrated with Amazon CloudWatch for metrics and alarms
- Detailed broker logs available in Amazon CloudWatch Logs
- AWS Management Console, CLI, and SDKs for broker management
- Supports broker maintenance and patching without downtime in HA deployments

Integration with AWS Services

- Amazon EC2 for hosting applications that use the broker
- AWS Lambda for event-driven messaging workflows
- Amazon SQS and SNS for additional messaging patterns
- AWS Secrets Manager for managing credentials
- AWS Identity and Access Management (IAM) for permissions

Pricing

- Charged per broker instance hour based on instance type
- Storage for message persistence is billed per GB per month
- Data transfer charges may apply for cross-AZ replication
- Additional costs for CloudWatch metrics and logs

Use Cases

- Migrating existing on-premises ActiveMQ or RabbitMQ workloads to AWS
- Decoupling microservices with message-based communication
- Building reliable, fault-tolerant messaging workflows

- Supporting hybrid architectures with consistent messaging protocols
- Integrating legacy applications with modern cloud services

Best Practices

- Use Active/standby deployments for high availability in production
- Monitor broker metrics with CloudWatch and set alarms
- Encrypt data at rest and in transit
- Use IAM policies and ACLs to control access
- Plan for instance sizing and storage needs based on workload
- Regularly review and update security settings and credentials

Exam Tips

- Fully managed message broker service for ActiveMQ and RabbitMQ
- Supports standard messaging protocols like JMS, AMQP, STOMP, MQTT
- Simplifies migration of existing messaging apps without code changes
- Supports high availability with multi-AZ replication and failover
- Integrates with IAM, KMS, Secrets Manager, CloudWatch
- Encryption at rest (KMS) and in transit (TLS)
- Use for decoupling microservices and supporting hybrid messaging scenarios

Quick Summary

Amazon MQ is a fully managed message broker service for Apache ActiveMQ and RabbitMQ, designed to simplify the setup and management of message brokers in the cloud. It provides high availability, supports industry-standard protocols, integrates with AWS security and monitoring services, and enables seamless migration of existing messaging applications to AWS.

Amazon SNS

What It Is

- Fully managed pub/sub messaging service.
- Decouples application components through asynchronous message delivery.
- Enables message fan-out to multiple subscribers.

Core Concepts

1. Topics

- Logical access point for publishing messages.
- Publishers send messages to a topic.
- Subscribers receive messages from a topic.

2. Subscriptions

- Determine where the messages are delivered.
- **Supported protocols:**
    - Amazon SQS
    - AWS Lambda
    - HTTP/HTTPS
    - Email/Email-JSON
    - SMS
    - Application (mobile push)
    - AWS Kinesis Data Firehose

3. Publishers and Subscribers

- **Publisher (Producer):** Sends messages to the SNS topic.
- **Subscriber (Consumer):** Receives messages using one of the supported protocols.

Message Delivery

- Push-based delivery to multiple endpoints.
- Supports message filtering by attributes to control what each subscriber receives.
- Retries automatically on delivery failure.
- Dead-letter queues (DLQs) supported for message delivery failures.

Use Cases

- Application decoupling via pub/sub.
- Fan-out from one publisher to multiple consumers.
- Event-driven architecture.

- Mobile push notifications.
- Real-time alerts and monitoring.

SNS vs SQS

Feature	SNS	SQS
Model	Pub/Sub (Push)	Queue (Pull)
Delivery	Push to multiple subscribers	Pull by one consumer
Use Case	Fan-out to services	Decoupling with buffering
Target Options	Lambda, SQS, SMS, Email, HTTP/S	Typically, Lambda or polling

Message Filtering

- Subscribers receive only messages that match specified message attributes.
- Reduces need for additional logic in subscribers.
- Attribute-based filtering rules defined during subscription creation.

Message Format

- JSON by default.
- Raw message delivery can be enabled for SQS or Lambda.

Delivery Retries and DLQ

- Automatically retries failed deliveries.
- Configurable retry policy:
    - Backoff strategy
    - Retry attempts
- DLQs supported for unprocessed messages (SQS as target).

Access Control

- Uses IAM policies and topic access policies.
- Controls who can publish or subscribe.
- Can restrict access to certain accounts, services, or conditions.

Encryption

- Supports Server-Side Encryption (SSE) using AWS KMS.
- Protects messages at rest within SNS topics.

Monitoring and Logging

- **CloudWatch metrics:**
    - Number of messages published/delivered

    - Failed deliveries
    - Delivery latency
- CloudTrail logs API calls for auditing.

Mobile Notifications

- **Supports mobile push to:**
    - Amazon Device Messaging (ADM)
    - Apple Push Notification Service (APNS)
    - Firebase Cloud Messaging (FCM)
    - Baidu Cloud Push

Cross-Region and Cross-Account

- Topics can be accessed cross-account with appropriate topic policies.
- No native cross-region topic replication (can use Lambda or other mechanisms for replication).

Limits

- SNS is highly scalable.
- Default limits apply (e.g., number of topics, message size up to 256 KB).
- Limits can be increased via support.

Pricing

- **Charged per:**
    - Requests (publishes)
    - Notification deliveries (by protocol)
- SMS pricing varies by destination country.

Exam Tips

- SNS = push model, SQS = pull model.
- Use SNS + SQS for fan-out architecture.
- Lambda, SQS, and HTTP/S are commonly used subscribers.
- Encrypt messages at rest using KMS SSE.
- Use IAM policies and topic access policies to control publishing/subscribing.
- For mobile apps, use SNS for push notifications.

Quick Summary

Amazon SNS is a fully managed pub/sub service that allows you to send messages to multiple subscriber endpoints, enabling application decoupling, real-time notifications, and fan-out messaging patterns across AWS and mobile applications.

Amazon SQS

What It Is

- Fully managed message queuing service.
- Decouples and scales distributed systems, microservices, serverless apps.
- Stores, transmits, and retrieves messages between components without them needing to know each other's status or location.

Key Features

- Reliable, scalable message queuing.
- Supports standard queues (high throughput, at-least-once delivery) and FIFO queues (exactly-once processing, ordered delivery).
- Fully managed, no servers to provision or manage.
- Integrates with many AWS services (Lambda, SNS, ECS, Step Functions).

1. Types of Queues

a. Standard Queue

- Default queue type.
- Nearly unlimited transactions per second.
- At-least-once delivery (may deliver duplicates).
- Best-effort ordering (no strict ordering guaranteed).
- **Use case:** High-throughput applications, where occasional duplicates are acceptable.

b. FIFO Queue (First-In-First-Out)

- Guarantees exactly-once processing.
- Preserves strict message order.
- Supports up to 300 transactions per second (can increase with batching).
- **Use case:** Banking transactions, inventory updates.

2. Message Lifecycle

- Producer sends message to queue.
- SQS stores message redundantly across multiple AZs.
- Consumer polls queue and processes message.
- Once processed, message is deleted.

3. Message Visibility Timeout

- Period after a message is retrieved when it remains invisible to other consumers.
- Prevents multiple consumers processing the same message simultaneously.

- If the consumer fails to delete the message before timeout ends, it becomes visible again.

4. Dead-Letter Queues (DLQ)

- Stores unprocessed or unsuccessfully processed messages.
- Helps isolate problematic messages for debugging.
- Used to prevent endless processing retries.

5. Delay Queues

- Postpone delivery of new messages for a configurable time (0–15 minutes).
- Useful for delaying processing without managing timing in the producer app.

6. Long Polling

- Reduces cost and empty responses by waiting until a message arrives (up to 20 seconds).
- More efficient than short polling, which immediately returns even if no message is available.

7. Message Retention Period

- Defines how long SQS keeps a message if not deleted.
- Configurable between 1 minute and 14 days (default is 4 days).

8. Message Size

- Single message size limit is up to 256 KB.
- For larger payloads, can use S3 for message payloads with Amazon SQS Extended Client Library.

9. Encryption

- SSE (Server-Side Encryption) using AWS KMS.
- Ensures messages are encrypted at rest.
- Encryption keys can be managed via KMS.

10. Access Control

- IAM policies control who can send, receive, delete, or manage queues.
- Supports resource-based policies for cross-account access.

11. Integrations

- **AWS Lambda:** Trigger functions directly from SQS.
- **SNS:** Fan-out architecture by sending messages from SNS to multiple SQS queues.
- **Step Functions:** Orchestrate workflows with SQS as a step.
- **EventBridge:** Route events to SQS for decoupled processing.

12. Cost Model

- **Pay for:**
    - Number of requests.
    - Payload data transfer.
- FIFO queues typically cost more than standard queues due to ordering guarantees.

13. FIFO Queue Features

- Message groups for ordered processing in parallel.
- Exactly-once delivery with deduplication.
- Up to 20,000 in-flight messages (per message group).

14. DLQ and Redrive Policy

- DLQs help isolate and investigate failing messages.
- Redrive policy defines:
    - Maximum receive count before moving to DLQ.
    - DLQ destination.

15. Limits

- Default limits on number of queues, throughput per account (can be raised via support).
- FIFO throughput limits can be increased with batching.

Use Cases

- Decouple microservices.
- Buffer and smooth out workloads.
- Offload long-running tasks.
- Failure isolation via DLQs.

Exam Tips

- **Standard queues**: high throughput, at-least-once delivery, possible duplicates.
- **FIFO queues**: exactly-once processing, strict ordering.
- Visibility timeout avoids double-processing during work.
- Long polling reduces cost and latency by waiting for messages.
- Encryption with KMS secures data at rest.

Quick Summary

Amazon SQS is a fully managed message queuing service that enables decoupled, scalable, and reliable communication between distributed application components. It supports both standard and FIFO queues, offers encryption, DLQs, long polling, and seamless AWS integrations for building robust cloud architectures.

AWS Step Functions

What It Is

AWS Step Functions is a fully managed service for building serverless workflows that coordinate multiple AWS services into business-critical applications. It allows you to design workflows as state machines with visual modelling, error handling, and service integrations.

Key Features

- Visual workflow editor for defining state machines
- Supports Standard and Express Workflows
- Built-in error handling, retries, and catchers
- State input, output, and result manipulation with JSON Path
- Service Integrations for over 200 AWS services
- Wait states for delays or timeouts
- Choice states for branching logic
- Parallel execution of tasks
- Execution history and detailed logging

Workflow Types

- **Standard Workflows**
    - Long-running, durable, auditable
    - Supports executions up to 1 year
    - Exactly-once workflow execution
- **Express Workflows**
    - High-volume, short-duration workflows
    - Lower cost for high-frequency invocations
    - Executions up to 5 minutes

States and State Types

- **Task State:** Runs a single unit of work (e.g., Lambda function, AWS service integration)
- **Choice State:** Implements branching logic based on input
- **Parallel State:** Runs multiple branches in parallel
- **Map State:** Processes array items in parallel or sequentially
- **Wait State:** Introduces delay or waits for a specific time
- **Pass State:** Passes input to output, adds static data
- **Succeed State:** Marks successful completion
- **Fail State:** Marks failure of execution

Service Integrations

- AWS Lambda for serverless tasks
- AWS Batch for container-based jobs
- AWS Glue for ETL workflows
- Amazon ECS and Fargate for container execution
- Amazon SageMaker for machine learning workflows
- SNS, SQS for messaging and queuing
- DynamoDB, S3 for data manipulation
- AWS API Gateway for service calls
- AWS EventBridge for event-driven workflows
- Supports SDK integrations for direct AWS service API calls

Error Handling and Retries

- Retry Policies: Define retry intervals and maximum attempts
- Catchers: Handle errors and redirect workflow paths
- Enables resilient and fault-tolerant workflows

State Input and Output

- **InputPath:** Selects part of the input to pass to the state
- **Parameters:** Defines the parameters sent to the task
- **ResultSelector:** Transforms task results before output
- **ResultPath:** Specifies where to insert the result in the state output
- **OutputPath:** Filters the final output from the state

Execution Management

- View execution history in AWS Console
- CloudWatch Logs for detailed execution traces
- Step Functions API for programmatic management of executions
- Supports synchronous and asynchronous executions

Express Workflow Details

- Designed for high-volume, event-driven workloads
- Suitable for data processing pipelines, streaming analytics
- Supports thousands of executions per second
- Lower cost per execution compared to Standard Workflows
- Limited to 5-minute execution duration

Standard Workflow Details

- Durable with execution history stored for 90 days
- Supports human approvals and long-running processes
- Enables auditability with detailed step-by-step tracking
- Higher cost per state transition

Monitoring and Logging

- Integrated with Amazon CloudWatch for metrics and logs
- View state machine executions and results
- Enables alarms and automated responses to failures

Security

- IAM policies for controlling access to Step Functions
- Fine-grained permissions per state machine and execution
- Supports VPC Endpoints for private access
- Encryption of data in transit and at rest with AWS KMS

Pricing

- Charged per state transition in Standard Workflows
- Charged per execution and duration in Express Workflows
- Additional costs for AWS service calls and data transfer

Use Cases

- Microservice orchestration
- Serverless application workflows
- Data processing pipelines
- Machine learning model training and deployment
- ETL workflows with AWS Glue
- Human approval workflows with Wait and Choice states
- Error handling and retries for service integrations

Best Practices

- Choose workflow type based on execution duration and volume
- Use Choice and Parallel states to simplify complex logic
- Implement retries and error catchers to improve resilience
- Use Wait states for delays and human approval processes
- Secure workflows with IAM roles and policies

- Monitor executions with CloudWatch for performance and errors

Exam Tips

- Standard Workflows are durable and long-running
- Express Workflows are cost-effective for high-volume, short-lived tasks
- Supports multiple AWS service integrations directly without custom code
- Enables visual design of workflows as state machines
- Built-in error handling with retries and catchers
- Integrated with CloudWatch for monitoring and logging
- IAM roles control permissions for workflow execution and service calls
- JSON Path used to manipulate state input and output

Quick Summary

AWS Step Functions is a serverless orchestration service that coordinates AWS services and custom logic into resilient workflows. It supports Standard and Express Workflows, enabling visual design, robust error handling, service integrations, and flexible execution models for a wide range of use cases.

AWS Cost Management

What It Is

AWS offers a suite of tools that help you track, estimate, forecast, optimize, and alert on your cloud spending. These tools are essential for cost visibility, control, budgeting, and savings.

1. AWS Pricing Calculator

- Helps estimate monthly AWS costs before deploying.
- Supports EC2, S3, RDS, and many other services.
- Useful for planning and cost comparisons.
- Results can be exported and shared.

2. AWS Free Tier

- Offers free usage of AWS services:
    - **12-month free tier**: EC2, S3, RDS, etc.
    - **Always free tier**: IAM, DynamoDB, Lambda (limited usage).
- Helps users explore AWS at no cost.
- Usage tracked in Billing Dashboard to avoid unexpected charges.

3. AWS Billing and Cost Management Dashboard

- **Central place to:**
    - View and download billing and usage reports.
    - Set budgets and alerts.
    - Manage payment methods and invoices.
    - Monitor free tier usage.
    - Access Cost Explorer, CUR, and Budgets.

4. AWS Budgets

- Set spending and usage limits, with alerts triggered when thresholds are breached.
- **Types of budgets:**
    - **Cost Budgets** – Monitor total spend.
    - **Usage Budgets** – Track units (e.g., hours, GB).
    - **RI/Savings Plans Budgets** – Monitor coverage and utilization.
- Supports notifications via SNS or AWS Chatbot.
- Can trigger automated actions like shutting down resources.

5. AWS Cost Explorer

- **Interactive tool to:**
    - Analyse past and current spend.
    - Group/filter costs by service, region, account, tag, etc.
    - Forecast future costs.
    - Visualize RI and Savings Plan utilization.
- Ideal for identifying cost trends and inefficiencies.

6. AWS Cost and Usage Report (CUR)

- Most detailed billing data, delivered in CSV format to S3.
- Granular data (hourly, daily) on usage and cost.
- Can be queried with Athena, visualized in QuickSight, or loaded into Redshift.

7. Consolidated Billing (via AWS Organizations)

- Combines charges across multiple accounts into one payer account.
- **Benefits:**
    - Volume discounts shared across accounts.
    - Simplifies centralized cost control.
    - Still tracks per-account usage and cost.

8. Cost Allocation Tags

- Helps categorize and attribute costs by:
    - Project, department, team, etc.
- **Types:**
    - AWS-generated tags (e.g., createdBy).
    - User-defined tags (custom).
- Must be activated in Billing Console for them to appear in Cost Explorer/CUR.

9. Reserved Instances and Savings Plans

- Discount pricing for long-term, predictable workloads.
- **Reserved Instances:**
    - Commit to specific instance types and regions.
    - Apply to EC2, RDS, ElastiCache, Redshift.
- **Savings Plans:**
    - More flexible pricing model.
    - Commit to a dollar amount per hour.

- Apply to EC2, Lambda, Fargate.

10. AWS Trusted Advisor – Cost Optimization

- Recommends savings by identifying:
  - Idle/underutilized resources.
  - Unassociated Elastic IPs.
  - Low-traffic Load Balancers.
- Part of **five categories of checks**.
- Some checks require Business or Enterprise support plans.

11. AWS Compute Optimizer

- Uses machine learning to recommend optimal EC2 instance types.
- Based on historical usage patterns.
- Helps reduce cost and improve performance by right-sizing.

12. AWS Cost Anomaly Detection

- Monitors spend and usage patterns using machine learning.
- Detects unexpected cost spikes.
- Automatically notifies via:
  - Email
  - SNS topic

Exam Tips

- **Cost Explorer** = visual breakdown, trend analysis, and forecasting.
- **AWS Budgets** = alerts on thresholds (cost, usage, RI/SP).
- **CUR** = detailed data, ideal for advanced analysis.
- **Tags must be activated** for use in allocation reports.
- **Savings Plans** are more flexible than Reserved Instances.
- **Cost Anomaly Detection** = machine learning to detect abnormal spikes.
- **Consolidated Billing** = volume discounts + centralized billing.
- **Compute Optimizer and Trusted Advisor** = optimization insights.

Quick Summary

AWS Cost Management tools help architects and organizations estimate, control, monitor, and optimize their cloud expenses. From planning costs (Pricing Calculator) to tracking anomalies (Cost Anomaly Detection) and analysing trends (Cost Explorer), these tools enable financial accountability in the cloud.

AWS Batch

What It Is

AWS Batch is a fully managed batch processing service that efficiently runs hundreds to thousands of batch computing jobs on AWS. It dynamically provisions the optimal compute resources (e.g., EC2, Spot, Fargate) based on the volume and requirements of your jobs — no need to manage clusters manually.

Key Features

- Fully managed batch computing, no cluster setup or job schedulers needed
- Automatically provisions compute resources based on job queue demand
- Supports EC2 and Spot Instances for cost optimization
- Integrates with Amazon ECS for containerized workloads
- Scales up and down automatically depending on queued jobs
- Supports single, array, and multi-node parallel jobs
- Fine-grained job dependencies and priorities
- Integrated with CloudWatch for logging and monitoring
- No additional cost, pay only for compute and storage used

Core Components

- **Job** – A single unit of work submitted to AWS Batch (e.g., a script, container, or command).
- **Job Definition** – Template specifying job parameters such as Docker image, vCPUs, memory, IAM role, and retry strategy.
- **Job Queue** – Holds submitted jobs and determines their priority/order of execution.
- **Compute Environment** – Defines the compute resources (e.g., EC2, Spot, Fargate) where jobs are run.

Job Types

- **Single Job** – Standard one-off task.
- **Array Job** – Run multiple copies of the same job with different input parameters (up to 10,000).
- **Multi-Node Parallel Job** – Run tightly coupled, distributed workloads (e.g., MPI, HPC workloads) across multiple EC2 instances.

Job States

- **SUBMITTED** – Job accepted into the queue.
- **PENDING** – Waiting for resources or dependencies.
- **RUNNABLE** – Ready to run but waiting for compute capacity.
- **STARTING** – Container instance being provisioned.

- **RUNNING** – Job currently executing.
- **SUCCEEDED** – Job completed successfully.
- **FAILED** – Job terminated with error.

Job Scheduling and Dependencies

- Priority is determined by job queue order and compute environment allocation.
- Supports job dependencies, jobs can start only after other jobs succeed.
- Array jobs and multi-node jobs can be combined for complex workflows.
- Retry strategies and timeout policies handle transient failures gracefully.

Compute Environments

- **Managed Compute Environment** – **AWS Batch automatically creates and scales compute resources** (EC2 On-Demand or Spot).
- **Unmanaged Compute Environment** – You manage the compute resources yourself (custom ECS clusters).
- Supports multiple instance types, Spot Fleet, or On-Demand instances.
- **Can specify allocation strategy**: BEST_FIT, BEST_FIT_PROGRESSIVE, or SPOT_CAPACITY_OPTIMIZED.

Integration with Other AWS Services

- **Amazon ECS** – Runs containerized jobs in Batch.
- **Amazon ECR** – Stores and pulls Docker images.
- **Amazon CloudWatch Logs** – Collects and monitors job logs.
- **Amazon SNS/SQS** – Triggers or notifies job completion/failure.
- **AWS Lambda / Step Functions** – Orchestrates Batch job submissions and workflows.
- **AWS Identity and Access Management (IAM)** – Controls access and permissions for jobs and resources.
- **Amazon S3** – Commonly used for input/output data storage.

Monitoring and Logging

- Integrated with Amazon CloudWatch for metrics and alarms.
- Track job execution history, retries, and failures in the console.
- Custom metrics can monitor job duration, queue depth, and cost.

Security

- IAM Roles control job and service access.
- AWS KMS encrypts sensitive data (e.g., environment variables).
- Supports VPC configuration to run jobs in private subnets.

Pricing

- No extra charge for AWS Batch itself.
- Pay only for EC2 / Fargate resources and storage used.
- Save costs by using Spot Instances for non-urgent or flexible workloads.

Use Cases

- Data processing pipelines and analytics
- Machine learning model training or inference
- Media rendering or transcoding
- Financial risk analysis or report generation
- Nightly ETL jobs or large-scale file processing

Best Practices

- Use Managed Compute Environments for automatic scaling and optimization.
- Leverage Spot Instances for cost savings — but plan for interruptions.
- Use Array Jobs to parallelize similar workloads.
- Define IAM roles with least-privilege access for each job.
- Set retry strategies and timeout policies to handle transient errors.
- Use CloudWatch for monitoring and job-level metrics.
- Store large datasets in S3 and pass paths as environment variables.
- Combine Step Functions or EventBridge for orchestration and automation.

Exam Tips

- You pay only for the underlying compute and storage used.
- Integrates with ECS to run containerized workloads.
- Supports EC2 On-Demand, Spot, and Fargate compute options.
- Job definitions store container image, vCPU/memory, IAM role, and retry config.
- Managed compute environments are the easiest to use, AWS handles provisioning.
- Array jobs can run thousands of similar jobs concurrently.
- Commonly integrated with Step Functions for orchestration and CloudWatch for logging.
- Ideal for long-running or parallel workloads that don't require real-time responses.

Quick Summary

AWS Batch is a serverless batch computing orchestration service that eliminates the need to manage clusters or job schedulers. It integrates tightly with ECS, CloudWatch, and IAM, automatically scales compute resources, supports job dependencies and retries, and is designed for running large-scale, cost-efficient, containerized workloads in the cloud.

Amazon EC2

What It Is

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the AWS cloud. It allows you to launch virtual servers on demand, scale them easily, and pay only for what you use.

Key Features

- Resizable compute capacity with multiple instance types.
- Supports Linux and Windows operating systems.
- Choice of instance types optimized for compute, memory, storage, or GPU workloads.
- Flexible pricing models.
- Supports auto scaling, load balancing, and high availability architectures.
- Integrated with many AWS services (EBS, S3, IAM, VPC, CloudWatch).

EC2 Instance Types

- **General Purpose**: Balanced resources (e.g., t3, m5).
- **Compute Optimized**: High-performance processors (e.g., c5).
- **Memory Optimized**: Large memory needs (e.g., r5, x1).
- **Storage Optimized**: High sequential I/O, local NVMe (e.g., i3).
- **Accelerated Computing**: GPU or FPGA support (e.g., p3, g4).

EC2 Pricing Models

1. **On-Demand Instances**
    - Pay by the second (or hour).
    - Flexible, no commitment.
    - Use for short-term, unpredictable workloads.
2. **Reserved Instances (RI)**
    - 1-year or 3-year terms.
    - Significant discounts.
    - Standard and Convertible RIs.
    - Best for steady-state workloads.
3. **Savings Plans**
    - Flexible pricing model with hourly spend commitment.
    - Apply to EC2, Fargate, Lambda.

4. **Spot Instances**
    - Spare capacity at up to 90% discount.
    - Can be terminated by AWS with 2-minute warning.
    - Good for batch processing, stateless workloads.
5. **Dedicated Hosts**
    - Physical servers dedicated to your use.
    - Helps meet compliance and licensing requirements.
6. **Dedicated Instances**
    - Run in VPC on hardware dedicated to you.
    - Not tied to specific physical server.
7. **Capacity Reservations**
    - Reserve capacity in a specific AZ.

Storage Options

- **EBS (Elastic Block Store)**
    - Persistent block storage.
    - Types: gp3 (general), io1/io2 (provisioned IOPS), st1/sc1 (HDD).
    - Snapshots stored in S3.
    - Supports encryption with KMS.
- **Instance Store**
    - Ephemeral, physically attached disks.
    - Data lost when instance stops/terminates.
    - High IOPS.
- **EFS (Elastic File System)**
    - Network file system for Linux instances.
    - Scales automatically.
- **FSx**
    - Managed Windows File Server or Lustre.

Networking Features

- Launched in VPCs.
- Assign public and private IP addresses.
- **Elastic IP**: Static, routable public IPv4 address.
- **Security Groups**: Virtual firewall for inbound/outbound rules.

- **Network ACLs**: Subnet-level traffic control.
- **ENI (Elastic Network Interface)**: Network adapter with private/public IPs.
- **Placement Groups**:
    - **Cluster**: Low-latency, high throughput.
    - **Spread**: Critical instances across hardware.
    - **Partition**: Large distributed workloads.

Launch Options

- **Amazon Machine Images (AMIs)**
    - Preconfigured templates with OS, software, configuration.
    - AWS, Marketplace, or custom AMIs.
- **User Data**
    - Scripts that run at launch.
    - Used for bootstrapping.
- **Launch Templates**
    - Store launch parameters for reuse.
- **Launch Configurations**
    - Older alternative for Auto Scaling.

Auto Scaling

- Automatically adjusts capacity to meet demand.
- **Auto Scaling Groups (ASGs)**:
    - Define min, max, and desired capacity.
    - Scaling policies based on CloudWatch alarms.
    - Health checks and instance replacement.

Load Balancing

- **Elastic Load Balancing (ELB)** supports:
    - **Application Load Balancer (ALB):** Layer 7, path-based routing.
    - **Network Load Balancer (NLB):** Layer 4, ultra-low latency.
    - **Gateway Load Balancer:** Transparent appliances.
    - Classic Load Balancer (legacy).

Monitoring and Management

- **CloudWatch**:
    - Metrics and alarms.

    - Log collection with CloudWatch Agent.
- **AWS Systems Manager (SSM)**:
    - Run Command, Patch Manager.
    - Session Manager for SSH-free access.
- **CloudTrail**:
    - Records API calls for audit.

Security

- **IAM Roles** for EC2:
    - Grant temporary credentials to apps.
- **Security Groups**:
    - Stateful, instance-level firewall.
- **Key Pairs**:
    - SSH access for Linux, RDP for Windows.
- **Encryption**:
    - EBS volume encryption using KMS.
    - Encrypted snapshots.

Purchasing Options and Cost Savings

- Use Savings Plans for flexibility.
- Purchase Reserved Instances for predictable workloads.
- Use Spot Instances for interruptible workloads at low cost.
- Combine Auto Scaling with Spot and On-Demand using EC2 Fleet or Spot Fleet.

Placement Strategies

- **Cluster Placement Group**:
    - High performance within single AZ.
- **Spread Placement Group**:
    - Instances spread across distinct hardware.
- **Partition Placement Group**:
    - Distribute across partitions with isolation.

High Availability and Fault Tolerance

- Deploy across multiple AZs.
- Use Auto Scaling Groups with ELB.
- Use Elastic IPs for static addressing.

- Take EBS Snapshots for backups.

Best Practices

- Use IAM Roles for permissions, not static keys.
- Encrypt EBS volumes.
- Design for stateless workloads where possible.
- Enable CloudWatch monitoring.
- Store logs centrally with CloudWatch Logs.
- Use Auto Scaling for resilience.
- Regularly patch instances with AWS Systems Manager.

Exam Tips

- On-Demand for flexible, unpredictable workloads.
- Reserved Instances/Savings Plans for steady state.
- Spot for batch, fault-tolerant tasks.
- EBS is persistent, Instance Store is ephemeral.
- Security Groups = stateful, NACLs = stateless.
- Use IAM roles instead of hard-coded credentials.
- User Data runs at boot for configuration.
- Placement Groups optimize network layout.
- Auto Scaling works with ELB for high availability.
- Know pricing options and best practices for cost optimization.

Quick Summary

Amazon EC2 provides resizable virtual servers in the cloud with flexible instance types, pricing models, and integrated storage and networking. It supports secure, scalable, and highly available architectures using Auto Scaling, Load Balancing, and AWS best practices for cloud-native design.

AWS EC2 Auto Scaling

What It Is

AWS Auto Scaling is a fully managed service that automatically adjusts capacity to maintain performance and minimize costs. It can be used to scale EC2 instances, ECS tasks, DynamoDB throughput, Aurora Replicas, and Spot Fleets. It supports both vertical and horizontal scaling strategies.

Key Features

- Automatically scales resources based on demand
- Helps improve application availability and reduce costs
- Predictive scaling (machine learning-based forecasting)

Components of EC2 Auto Scaling

1. **Launch Template or Launch Configuration**
    - Specifies instance type, AMI, key pair, security groups, and other settings
    - Launch Template is recommended over Launch Configuration
2. **Auto Scaling Group (ASG)**
    - A logical group of EC2 instances managed together
    - Defines min, max, and desired capacity
    - Can span multiple Availability Zones within a region
3. **Scaling Policies**
    - **Dynamic Scaling:** Adjusts capacity based on real-time metrics
        - **Target Tracking:** Maintain metric (e.g., CPU) at target value
        - **Step Scaling:** Increase/decrease capacity in steps based on thresholds
        - **Simple Scaling:** Basic rule that adds/removes instances based on CloudWatch alarm
    - **Predictive Scaling:** Uses ML to forecast traffic and scale proactively
    - **Scheduled Scaling:** Scales at specific times based on schedule

Health Checks

- Performed via EC2 and optionally ELB
- Unhealthy instances are automatically terminated and replaced

Elastic Load Balancer Integration

- ASG automatically registers and deregisters instances with ELB
- Ensures traffic is distributed only to healthy instances
- Supports Classic, Application, and Network Load Balancers

Lifecycle Hooks

- Pause instance launching or terminating processes to run custom scripts
- Use cases include bootstrapping, log collection, or cleanup before termination

Monitoring and Alarms

- Uses Amazon CloudWatch for metrics and alarms
- **Key metrics:** CPUUtilization, NetworkIn, NetworkOut, etc.
- Alarms trigger scaling actions based on thresholds

Instance Refresh

- Replace instances in ASG with new configuration
- Helps apply updates or patches without downtime

Warm Pools

- Pre-initialize EC2 instances and keep them in a stopped state
- Reduces scale-out time by maintaining warm capacity

Pricing

- No additional cost for using Auto Scaling
- Pay only for the underlying AWS resources (EC2, ELB, etc.)

Use Cases

- Handling unpredictable traffic
- Ensuring fault tolerance and high availability
- Reducing operational overhead through automation
- Lowering cost by scaling in when demand is low

Exam Tips

- Use Target Tracking Scaling Policy for maintaining consistent performance
- Predictive Scaling is useful for known traffic patterns
- Health checks help maintain application availability
- Use Warm Pools to improve response time for scale-out events
- ASG can span multiple AZs but not across regions
- Use ELB to distribute traffic across Auto Scaling instances
- Instance Refresh is useful for patching or configuration updates

Quick Summary

AWS Auto Scaling helps maintain optimal performance and cost efficiency by dynamically adjusting capacity based on real-time metrics or forecasts. It integrates with EC2, ELB, CloudWatch, and other AWS services to provide scalable, fault-tolerant.

AWS Elastic Beanstalk

What It Is

AWS Elastic Beanstalk is a fully managed service that makes it easy to deploy, run, and scale web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker. It handles the provisioning of infrastructure such as EC2, Auto Scaling, and ELB.

Key Features

- Simplifies application deployment without worrying about the underlying infrastructure
- Automatically handles capacity provisioning, load balancing, scaling, and health monitoring
- Supports multiple programming languages and platforms
- Integrated with developer tools like Git, Jenkins, and IDEs
- Allows customization of environments using configuration files and extensions
- Provides full control over resources if needed
- Supports rolling deployments, immutable deployments, blue/green deployments

Deployment Options

- **All at once:** Deploys to all instances simultaneously
- **Rolling:** Deploys to a batch of instances at a time
- **Rolling with additional batch:** Adds a new batch before terminating the old one
- **Immutable:** Deploys to a new set of instances before switching traffic
- **Blue/Green:** Creates a new environment (green), tests it, and swaps with the old one (blue)

Supported Platforms

- Java with Tomcat, .NET on Windows Server with IIS, PHP, Python, Ruby, Go
- Node.js, Docker (single and multi-container)

Architecture and Components

- **Application**: Logical collection of components including environments, versions, and configurations
- **Application version:** Specific, deployable version of the application code
- **Environment:** Collection of AWS resources running the application version
- **Environment tier:** Determines if the environment is a web server or worker
- **Configuration template:** Predefined settings for environment creation
- Environment configurations are stored in .ebextensions configuration files

Environment Tiers

- **Web Server Tier:** Handles HTTP requests, uses a web server like Apache or Nginx

- **Worker Tier:** Handles background tasks using SQS

Customizations and Extensions

- Use .ebextensions YAML config files to customize EC2 instances and install packages
- Add cron jobs, configure logging, change instance types, and more
- Can use AWS CloudFormation templates embedded in extensions

Monitoring and Logging

- Provides real-time metrics and health status of environments
- Integrated with Amazon CloudWatch and Elastic Load Balancing health checks
- Logs available through the Elastic Beanstalk console or stored in S3
- Notifications via Amazon SNS

Scaling

- Integrated with Auto Scaling for horizontal scaling
- Allows setting min, max, and desired instance counts
- Load balancing with ELB for high availability

Security

- Supports IAM roles for environment instances
- Integrates with VPC for private networking
- SSL termination via ELB
- Environment variables can be encrypted with KMS

Application Lifecycle

- Upload application version
- Deploy to an environment
- Monitor and update as needed
- Terminate environment when no longer needed

Pricing

- No additional charge for Elastic Beanstalk
- You pay for the underlying resources (EC2, ELB, S3, etc.)

Use Cases

- Web application hosting with simplified infrastructure management
- Rapid deployment and iteration of applications
- Scalable and managed environments for microservices
- CI/CD pipelines for DevOps workflows

Best Practices

- Use immutable or blue/green deployments for safer updates
- Store configuration in version control using .ebextensions
- Monitor environment health and logs regularly
- Use environment cloning for testing changes before production
- Use managed updates to automate minor platform updates

Exam Tips

- Elastic Beanstalk automatically provisions and manages the infrastructure
- Supports multiple deployment strategies including rolling and immutable
- Uses EC2, ELB, Auto Scaling, RDS, and other AWS services under the hood
- You still have full access to underlying resources
- Configuration is managed via environment settings and .ebextensions
- Blue/green deployments provide zero-downtime updates

Quick Summary

AWS Elastic Beanstalk simplifies application deployment by managing the infrastructure and platform for you. It supports a variety of programming languages and frameworks, provides several deployment options, integrates with monitoring and scaling services, and allows deep customization when needed, all while giving you full control of the underlying AWS resources.

AWS Outposts

What It Is

AWS Outposts is a fully-managed hybrid-cloud service that brings AWS infrastructure, services, APIs and tools to your on-premises or edge location. It essentially extends an AWS Availability Zone into your data centre or colocation site.

Key Features

- Run AWS services locally (EC2, EBS, ECS/EKS, etc) in your premises.
- Managed hardware from AWS (racks/servers/switches) so you don't handle the low-level stuff.
- Same APIs, tooling, and experience as the cloud region (so your DevOps patterns stay consistent).
- Low latency access and on-premises data processing/residency advantages (for compliance, edge use-cases).

Form Factors / Configurations

- **Outposts Racks** – Full 42U rack installations with compute + storage + network, installed on-site by AWS.
- **Outposts Servers** – Smaller footprint (1U or 2U servers) for branch locations, retail, edge zones.

How It Works

- Your Outpost is "homed" to an AWS Region and Availability Zone; it extends that AZ into your premises.
- Outpost Subnet – You create a subnet in your VPC that's associated with the Outpost.
- Local Gateway (LGW) – For rack form factor: route tables point to LGW for on-prem network integration (direct routing or CoIP).
- Local Network Interface (LNI) – For server form factor: network interface to your on-prem network.
- Service Link – Encrypted connectivity between Outpost and AWS Region for management/intra-VPC traffic.

Supported Services & Capabilities

- **Compute:** EC2, ECS, EKS locally.
- **Storage:** EBS (and in some cases S3/other services depending on generation) locally.
- **Networking/VPC:** VPC extension, route tables, local routing to on-prem, Internet/Direct Connect integration.
- **Data residency, edge computing:** useful when you can't send data off-site because of regulation or latency constraints.

Pricing

- **Term-based pricing:** Typically, a 3-year term for Outpost rack capacity.
- **You pay for:**
    - The Outpost infrastructure (rack or server) term contract. –
    - AWS services running on the Outpost (the compute/storage you consume) and shared resources.
    - Data transfer associated with service link VPN traffic from the AWS Region.
- **Not charged for:** Data transfer from the Outpost to the parent AWS Region (some cases).

Use Cases

- Ultra-low latency workloads (on-site manufacturing systems, real-time analytics) where cloud latency isn't acceptable.
- Data residency / compliance scenarios: you must keep data within a country/region/your own site.
- Edge/branch offices or retail environments where you want cloud consistency but need local compute/storage.

Best Practices

- Evaluate if you genuinely need on-premises: hybrid comes with cost and complexity.
- Right-size your Outpost capacity (compute, storage) for your workload and growth.
- Ensure your site requirements (power, cooling, network, physical space) are met before ordering.
- Extend your VPC carefully: manage routing, subnets, utilization of LGW / LNI properly.
- Use consistent IAM and monitoring patterns as you use in the cloud region so your DevOps tooling remains sane.

Exam Tips

- Remember: Outposts = AWS on-premises. It's not just a "local VM" but AWS infrastructure extended into your site.
- It uses same AWS APIs, tool-chain, and management model — so you treat it like an AZ extension.
- Distinguish between rack vs server form-factors.
- The VPC routing to/from your on-premises network matters: Outpost subnets act like any other AZ subnet, with extended routing to local network.

Quick Summary

AWS Outposts brings the AWS cloud hardware, services, and APIs into your own data-centre or edge site, giving a truly hybrid experience. You get local compute/storage with consistent tooling, while still accessing cloud region services.

AWS Serverless Application Repository

What It Is

AWS Serverless Application Repository (SAR) is a managed repository that enables developers, teams, and organizations to find, deploy, and share serverless applications. It simplifies the discovery and reuse of serverless components and entire applications built using AWS services.

Key Features

- Fully managed repository for serverless apps
- Supports public and private sharing within AWS accounts or organizations
- Applications packaged as AWS SAM templates
- Deploy directly through the AWS Management Console, AWS CLI, AWS SDKs, or AWS SAM CLI
- Supports versioning of applications for tracking changes over time
- Integrated with AWS IAM for permission management

Publishing Applications

- Developers can publish serverless applications built using AWS SAM (Serverless Application Model)
- Must include an AWS SAM template defining resources and configurations
- Supports adding metadata such as application name, description, license, and usage instructions
- Can publish public applications available to all AWS customers or restrict sharing to specific accounts/organizations
- Enables versioning for iterative improvements and easy rollback

Deploying Applications

- Applications can be deployed directly to your AWS account from the console, CLI, or SAM CLI
- AWS CloudFormation handles provisioning and deployment using the included SAM template
- Supports customization through parameters defined in the template
- Ensures best practices by enforcing proper packaging and permissions

Security and Access Control

- Integrated with AWS IAM for fine-grained access control
- Publishers control who can view and deploy applications
- Permissions defined in the SAM template are reviewed during deployment
- AWS performs automated scanning for known security issues in published apps

Integration with AWS Services

- Works with AWS Lambda, API Gateway, DynamoDB, Step Functions, SNS, SQS, and other serverless components
- Deployments use AWS CloudFormation for consistent and repeatable provisioning
- AWS SAM CLI can be used for local testing and publishing to the repository

Benefits

- Accelerates development by reusing validated, production-ready serverless apps
- Reduces duplication of effort within organizations
- Simplifies sharing of best practices and vetted solutions
- Improves consistency across teams by standardizing serverless deployments
- Supports continuous integration and deployment workflows

Use Cases

- Sharing serverless microservices within a company
- Open-source distribution of serverless solutions
- Centralizing vetted serverless patterns and architectures
- Building internal catalogs of reusable serverless components
- Quickly deploying common integrations and utilities

Pricing

- No additional charge to use the AWS Serverless Application Repository itself
- Pay for underlying AWS resources provisioned by the application (e.g., Lambda invocations, API Gateway requests, DynamoDB storage)

Exam Tips

- SAR is a managed repository for packaging, sharing, and deploying serverless apps
- Apps are defined using AWS SAM templates
- Supports public sharing and private sharing with accounts or organizations
- Deployments are powered by AWS CloudFormation
- Ideal for promoting reuse and consistency in serverless development across teams and organizations

Quick Summary

AWS Serverless Application Repository is a fully managed service for discovering, sharing, and deploying serverless applications defined with AWS SAM. It helps developers and organizations reuse proven serverless architectures, accelerates delivery, and promotes best practices in building serverless solutions on AWS.

AWS Wavelength

What It Is

AWS Wavelength extends AWS compute and storage services into 5G networks by placing AWS infrastructure inside telecommunications data centres. This allows developers to run applications closer to mobile users, achieving ultra-low latency while still using the same AWS services, APIs, and tools as in a normal AWS Region.

Key Features

• Run applications within Wavelength Zones (extensions of an AWS Region)
• Ultra-low latency for mobile and edge devices
• Deep integration with 5G networks (Verizon, Vodafone, KDDI, SK Telecom, etc.)
• Use familiar AWS services like EC2, ECS, EKS, VPC, ALB, and CloudWatch
• Consistent AWS APIs, tools, AMIs, and IAM model
• Traffic stays local to 5G network → minimal hops, minimal jitter
• Supports containers, microservices, ML inference, AR/VR, gaming workloads

Wavelength Zones

• Specific, carrier-operated data centres that extend an AWS Region
• Appear inside your VPC as isolated subnets
• Connected to the parent Region via high-bandwidth, secure network
• Provide compute (EC2), storage (EBS), networking (VPC), and load balancing
• Used for low-latency applications targeted at mobile users in that geography

Supported Services

• Amazon EC2 (certain instance types optimized for edge)
• Amazon EBS for block storage
• Amazon VPC with Wavelength subnets
• Elastic Load Balancing (NLB only)
• Amazon ECS/EKS for container workloads
• Amazon CloudWatch for logs/metrics
• Amazon CloudFormation & CLI for automation

Networking

• Create a VPC in the parent Region, then add Wavelength subnets
• Carrier Gateways
-Enable traffic between Wavelength instances and 5G mobile users
-No direct internet access from Wavelength instances
• Service traffic usually flows: Mobile Device → 5G Network → Wavelength Zone
• Use NLB for distributing traffic to Wavelength EC2
• Supports Security Groups, NACLs, VPC routing

Storage

• Amazon EBS for persistent block storage
• Root volumes + data volumes supported
• No S3 inside Wavelength Zones (S3 lives in the parent Region)
• Use S3 for logs, backups, or static content (latency will be higher)

Compute

• EC2 instance families vary by carrier/location
• Supports running containers via ECS/EKS
• Ideal for workloads requiring high compute + low latency

Use Cases

• Real-time gaming and game streaming
• AR/VR applications
• Autonomous systems & robotics control
• Live video analytics
• ML inference at the edge
• Smart cities & IoT backends
• Telemedicine, remote monitoring
• Connected vehicles (V2X)

Security

• Uses AWS IAM, KMS, SGs, NACLs, same as standard Region
• VPC isolation for Wavelength subnets
• All communication to parent Region secured over AWS backbone
• No internet gateway inside Wavelength, use Carrier Gateway instead

Pricing

• **Pay for:**
- EC2 instance usage in Wavelength Zones
- EBS storage attached
Data transfer from Wavelength → parent Region
• No extra surcharge for using Wavelength Zones

Deployment Flow

1. Create VPC in the associated AWS Region
2. Add Wavelength Zone subnets
3. Deploy EC2 / ECS / EKS workloads into those subnets
4. Attach NLB for mobile user traffic
5. Route mobile traffic via Carrier Gateway
6. Monitor with CloudWatch

Exam Tips

• Wavelength = AWS at the edge of 5G networks
• Wavelength Zones use Carrier Gateways, not Internet Gateways
• Compute available: EC2, ECS, EKS

Quick Summary

AWS Wavelength extends AWS infrastructure directly into 5G networks, enabling developers to deploy compute and storage close to mobile users. It provides single-digit millisecond latency, consistent AWS tooling, and deep VPC integration through Wavelength Zones, perfect for real-time, mobile-heavy, edge-driven applications like AR/VR, gaming, analytics, and IoT.

Amazon ECS Anywhere

What It Is

Amazon ECS Anywhere allows you to run ECS tasks on your own hardware , whether on-premises, edge locations, or other environments outside AWS.
It provides a hybrid cloud model while still using the same ECS control plane that runs in AWS.

Key Features

- Run containers on-premises or in any non-AWS environment
- Uses the same ECS management and APIs
- Ideal for hybrid and edge workloads
- You manage the infrastructure (servers, networking, etc.)
- ECS manages the containers, scheduling, deployment, and monitoring

Benefits

- Extends AWS container management to on-prem environments
- Reduces operational complexity with a consistent ECS experience
- Single control plane for managing workloads across AWS and on-prem
- Enables hybrid application architectures

Use Cases

- Edge computing (IoT or low-latency workloads)
- Regulated industries where data must remain on-prem
- Gradual cloud migration while keeping control of on-prem workloads

Exam Tips

- ECS Anywhere = Run ECS tasks outside AWS
- Hybrid container orchestration, same ECS control plane
- User manages infrastructure, ECS manages containers

Amazon EKS Anywhere

What It Is

Amazon EKS Anywhere allows you to create and operate Kubernetes clusters on your own infrastructure while still using EKS tooling and APIs.
It extends the EKS experience to on-premises or edge data centres.

Key Features

- Run Kubernetes clusters locally using EKS tooling
- No dependency on AWS infrastructure
- Supports VMware vSphere and bare metal
- Integrated with EKS console and EKS CLI for management
- Supports GitOps-based workflows for consistent deployments

Benefits

- Bring Kubernetes to your data centre
- Maintain operational consistency with EKS in AWS
- Allows hybrid or multi-cloud architectures
- Improves flexibility for regulated or latency-sensitive workloads

Use Cases

- Hybrid Kubernetes deployments
- On-prem Kubernetes management with AWS consistency
- Organizations needing local control with AWS EKS tooling

Exam Tips

- EKS Anywhere = Run Kubernetes clusters on-prem using EKS tools
- No need to rely solely on AWS-managed EKS
- Ideal for hybrid and edge Kubernetes environments

Amazon EKS Distro (EKS-D)

What It Is

Amazon EKS Distro (EKS-D) is the open-source Kubernetes distribution that powers Amazon EKS.
It includes the same tested Kubernetes components AWS uses in EKS, available to run anywhere.

Key Features

- Open-source Kubernetes distribution
- Same versioned, security-hardened components used by EKS
- Run on any infrastructure AWS, on-prem, or other clouds
- AWS-maintained patches for security and stability
- Ensures compatibility with Amazon EKS clusters

Benefits

- Enables consistent Kubernetes operations across environments
- Provides transparency and control over Kubernetes upgrades
- Security updates aligned with EKS releases
- Suitable for custom Kubernetes environments needing EKS compatibility

Use Cases

- Self-managed Kubernetes clusters with AWS consistency
- Air-gapped or disconnected environments
- Multi-cloud and on-prem Kubernetes setups

Exam Tips

- EKS Distro (EKS-D) = Open-source Kubernetes distribution behind EKS
- Same tested components as AWS-managed EKS
- Helps maintain compatibility, reliability, and security
- Can be run anywhere, not limited to AWS

Quick Summary

ECS Anywhere, EKS Anywhere, and EKS Distro (EKS-D) extend AWS container services beyond the cloud. ECS Anywhere runs ECS tasks on your own hardware, EKS Anywhere brings EKS-managed Kubernetes to on-prem environments, and EKS Distro offers the same open-source Kubernetes used by AWS. Together, they enable consistent, secure, and flexible hybrid container management across on-premises and cloud environments.

Amazon Elastic Container Service (Amazon ECS)

What It Is

Amazon ECS is a fully managed container orchestration service that makes it easy to deploy, manage, and scale containerized applications using Docker. It eliminates the need to install and operate your own container orchestration software or manage clusters of virtual machines.

Key Features

- Fully managed and scalable container orchestration
- Supports Docker containers
- Deep integration with AWS services (IAM, CloudWatch, VPC, ALB, Service Discovery)
- Runs on EC2 instances or AWS Fargate for serverless containers
- Supports Windows and Linux workloads
- Integrated service discovery using AWS Cloud Map
- Rolling updates and deployment controls
- Task placement strategies and constraints

Launch Types

- EC2 Launch Type
    - Run container workloads on a cluster of EC2 instances you manage
    - Full control over instance types, AMIs, networking
    - Suitable for predictable, steady workloads
- Fargate Launch Type
    - Serverless, no EC2 management
    - Specify CPU and memory at the task level
    - AWS manages provisioning, scaling, patching
    - Ideal for variable or bursty workloads

Clusters

- Logical grouping of tasks or services
- Can include EC2 instances with ECS agent installed
- Managed using ECS Console, CLI, or API
- Integrated with AWS CloudFormation for infrastructure as code

Tasks and Task Definitions

- Task Definition
    - Blueprint for running containers

    - Specifies Docker images, CPU, memory, networking, IAM roles, environment variables
    - Supports multiple containers (sidecars) in a single task
- Task
    - Instantiation of a Task Definition
    - Runs on an EC2 instance or Fargate

Services

- Manage long-running tasks
- Maintain desired count of running tasks
- Supports rolling updates and blue/green deployments (with AWS CodeDeploy)
- Integrated with Elastic Load Balancing for distributing traffic
- Auto-scaling support based on CloudWatch metrics

Capacity Providers

- Define how ECS runs tasks across launch types (EC2, Fargate)
- Manage scaling policies and capacity weighting
- Supports EC2 Spot instances for cost optimization

Networking

- Integrated with Amazon VPC
- Supports awsvpc networking mode for task-level ENIs
- Security Groups and NACLs for controlling traffic
- Load balancing using ALB, NLB, or CLB

Storage

- Supports Amazon EFS integration for persistent, shared storage
- Data volumes defined in task definitions
- Ephemeral storage for scratch space

Monitoring and Logging

- Integrated with Amazon CloudWatch Logs
- Collect and view container logs
- CloudWatch metrics for CPU, memory usage, and custom metrics
- AWS X-Ray integration for tracing

Security

- IAM roles for tasks, services, and cluster management

- Task Role
    - Provides AWS API permissions to the containers in the task
- Task Execution Role
    - Permissions to pull images, write logs, manage secrets
- Supports encryption of sensitive data using AWS Secrets Manager and AWS KMS
- Private Docker registries supported via AWS Secrets Manager

Integration with Other AWS Services

- AWS Fargate for serverless container hosting
- Elastic Load Balancing for distributing traffic
- AWS App Mesh for microservices networking
- AWS CloudWatch for logs and metrics
- AWS X-Ray for tracing
- AWS CodePipeline and CodeDeploy for CI/CD
- Amazon ECR for storing Docker images
- AWS Secrets Manager and Parameter Store for managing secrets

Pricing

- No additional charge for ECS itself
- Pay for underlying compute resources
    - EC2 instances for EC2 Launch Type
    - Fargate task pricing based on vCPU and memory used per second
- Other AWS resources used (e.g., ALB, ECR, CloudWatch) incur standard charges

Use Cases

- Microservices applications
- Batch processing workloads
- Hybrid workloads (EC2 and Fargate)
- Migrating existing Docker workloads to AWS
- Event-driven container execution

Exam Tips

- ECS supports EC2 and Fargate launch types
- Task Definition = blueprint; Task = running instance
- Services manage scaling, load balancing, and deployment
- awsvpc networking mode assigns ENI per task for VPC integration

- IAM roles for Task and Task Execution enable secure access to AWS services
- ECS Capacity Providers support EC2 On-Demand, EC2 Spot, and Fargate
- Use Fargate for serverless, no-instance-management deployments
- Supports integration with ALB, EFS, Secrets Manager, CloudWatch, X-Ray

Quick Summary
Amazon ECS is AWS's fully managed container orchestration service supporting Docker workloads. It offers flexible deployment with EC2 and Fargate launch types, deep AWS integration for security and monitoring, and tools for scaling and managing container-based applications in production.

Amazon Elastic Container Registry (Amazon ECR)

What It Is

Amazon ECR is a fully managed container image registry that makes it easy to store, manage, share, and deploy container images and artifacts. It integrates with AWS services like ECS, EKS, and AWS Lambda, as well as open-source tools.

Key Features

- Fully managed, highly available registry
- Supports Docker and Open Container Initiative (OCI) images
- Integrated with ECS, EKS, Lambda, CodeBuild, CodeDeploy, CodePipeline
- Push and pull images using standard Docker CLI or AWS CLI
- Supports image scanning for vulnerabilities
- Repository-level permissions with AWS IAM
- Lifecycle policies for automated image cleanup
- Supports replication across regions for DR and latency optimization
- Encryption at rest with AWS KMS and in transit with TLS
- Cross-account access with resource policies

Repositories

- Stores container images in private or public repositories
- Private repositories for controlled access
- Public repositories for open sharing without authentication
- Each repository stores multiple image versions identified by tags

Authentication

- AWS CLI or SDK for authentication
- IAM policies control user and role access to repositories
- ECR Credential Helper for Docker CLI integration
- Supports temporary credentials via AWS IAM roles

Image Management

- Push images to ECR from build systems or local machines
- Pull images to ECS, EKS, Fargate, or other Docker environments
- Supports multi-architecture images
- Image tags for versioning and organizing images
- Immutable tags can be enforced to prevent overwrites

Image Scanning

- Built-in vulnerability scanning using Common Vulnerabilities and Exposures (CVE) database
- Can scan on image push or on-demand
- Provides detailed findings for remediation
- Integrated with AWS Security Hub for consolidated findings

Replication

- Supports cross-region replication of images
- Automatic replication to one or more AWS regions
- Simplifies DR, compliance, and global deployment needs
- Managed via repository settings and policies

Lifecycle Policies

- Automates cleanup of unused or old images
- Rules based on image count, age, or tags
- Reduces storage costs and clutter

Encryption

- Images encrypted at rest using AWS KMS
- Encryption in transit via HTTPS/TLS
- Supports customer-managed KMS keys

Logging and Monitoring

- Amazon CloudWatch integration for monitoring actions
- AWS CloudTrail records API calls for auditing
- EventBridge integration for image push notifications

Integration with AWS Services

- Amazon ECS and EKS for container orchestration
- AWS Fargate for serverless container deployment
- AWS CodePipeline, CodeBuild, CodeDeploy for CI/CD
- AWS Lambda for container-based functions

Pricing

- Charged based on data storage (per GB-month)
- Data transfer costs for image pulls outside the region
- Scanning fees per image scan

- Free tier includes 500MB storage per month for private repositories

Public Repositories

- Host public container images with no authentication required for pulls
- Ideal for sharing software, tools, or base images publicly
- Integrates with the AWS Container Registry website for browsing

Use Cases

- Storing private Docker images for ECS, EKS, Fargate
- Hosting public images for open-source projects
- Integrating secure, managed image storage in CI/CD pipelines
- Enabling multi-region deployments with cross-region replication
- Ensuring security with built-in vulnerability scanning

Exam Tips

- Supports both private and public repositories
- IAM policies and resource policies manage access
- Vulnerability scanning can be on-push or on-demand
- Cross-region replication for DR and global distribution
- Integrated with ECS, EKS, Fargate, CodePipeline, Lambda
- Lifecycle policies help automate image cleanup
- Encryption at rest with KMS and in transit with TLS
- Use ECR Credential Helper or AWS CLI for authentication

Quick Summary

Amazon ECR is a fully managed, secure, scalable container image registry that integrates with AWS container services and CI/CD workflows. It simplifies storing, managing, and sharing container images with built-in security, replication, and automation features for modern containerized application development.

Amazon Elastic Kubernetes Service (Amazon EKS)

What It Is

Amazon EKS is a fully managed Kubernetes service that simplifies deploying, managing, and scaling containerized applications using Kubernetes on AWS. It eliminates the need to install, operate, and maintain your own Kubernetes control plane.

Key Features

- Fully managed, highly available Kubernetes control plane
- Runs upstream, open-source Kubernetes for compatibility
- Automatic scaling and patching of control plane
- Integrated with AWS services like IAM, VPC, ALB, CloudWatch, ECR
- Supports EC2 and AWS Fargate as compute options
- Native Kubernetes tools (kubectl, Helm) work seamlessly
- Multi-AZ deployments for high availability

Control Plane

- Managed by AWS with automatic scaling and redundancy
- Runs across multiple Availability Zones for high availability
- Managed API server and etcd with automatic patching and backups
- AWS handles upgrades of the Kubernetes control plane

Worker Nodes

- EC2 Launch Type
    - Self-managed or managed node groups
    - Control over instance types and AMIs
- AWS Fargate Launch Type
    - Serverless, runs Kubernetes pods without managing EC2 instances
    - Specify CPU and memory at pod level
- Managed Node Groups
    - AWS provisions and manages EC2 nodes
    - Integrated with Auto Scaling

Networking

- Uses Amazon VPC for cluster networking
- Supports Kubernetes CNI (Container Network Interface) with VPC-native networking
- Each pod gets its own ENI and private IP
- Security Groups and NACLs for traffic control

- AWS Load Balancer Controller for managing ALB/NLB integration

Security

- IAM integration for Kubernetes RBAC
- IAM Roles for Service Accounts (IRSA) for granular permissions
- Supports Kubernetes RBAC for cluster-level access control
- Encryption at rest using AWS KMS
- Encryption in transit using TLS
- Integration with AWS Secrets Manager and AWS Parameter Store for managing secrets

Storage

- Supports Amazon EBS for persistent volumes
- Supports Amazon EFS for shared storage between pods
- Supports FSx for Lustre for high-performance workloads
- Dynamic provisioning of persistent volumes using Kubernetes Storage Classes

Logging and Monitoring

- Integrated with Amazon CloudWatch for logs and metrics
- Container Insights for detailed monitoring of cluster resources and applications
- Supports AWS X-Ray for tracing
- Fluent Bit and CloudWatch Logs for log collection

Cluster Autoscaler

- Automatically adjusts the number of nodes in your cluster based on pending pods
- Supports scaling EC2 instances in managed node groups

Kubernetes Versions

- AWS regularly updates supported Kubernetes versions
- Ability to choose versions for clusters
- Control plane and worker nodes can be upgraded independently

Integration with AWS Services

- AWS IAM for authentication and authorization
- AWS ALB/NLB for ingress
- Amazon ECR for container image storage
- AWS App Mesh for service mesh and observability
- AWS Fargate for serverless pods
- AWS CloudFormation and eksctl for infrastructure as code

Pricing

- Charged per EKS cluster per hour
- Pay for EC2 instances or Fargate pods used as worker nodes
- Standard AWS charges apply for other integrated services (EBS, ALB, CloudWatch)

Use Cases

- Deploying and managing microservices applications
- Hybrid workloads with EC2 and Fargate
- Batch processing and ML workflows on Kubernetes
- Migrating existing Kubernetes workloads to AWS
- Building multi-AZ, highly available applications

Exam Tips

- EKS control plane is managed by AWS and runs across multiple AZs
- Supports EC2 and Fargate launch types for worker nodes
- VPC CNI provides pod-level ENIs and IP addresses
- IAM Roles for Service Accounts allow fine-grained permissions for pods
- AWS Load Balancer Controller automates ALB/NLB integration for services
- Supports encryption at rest with KMS and in transit with TLS
- Managed Node Groups simplify worker node lifecycle management
- EFS integration enables persistent, shared storage between pods
- eksctl simplifies cluster provisioning and management
- Best for running standard Kubernetes workloads with AWS integrations

Quick Summary

Amazon EKS is a fully managed Kubernetes service providing a production-ready control plane, integrated AWS security and networking, and support for both EC2 and Fargate worker nodes. It allows organizations to run standard Kubernetes applications on AWS with high availability, scalability, and ease of management.

Amazon Aurora

What It Is

Amazon Aurora is a fully managed, MySQL- and PostgreSQL-compatible relational database engine built for the cloud. It delivers the performance and availability of high-end commercial databases at a lower cost, while being fully integrated with AWS services.

Key Features

- MySQL- and PostgreSQL-compatible
- Up to 5x faster than standard MySQL and 3x faster than standard PostgreSQL
- Storage automatically scales in 10GB increments up to 128TB
- Fault-tolerant and self-healing storage system with six-way replication across three Availability Zones
- Continuous backups to Amazon S3
- Automatic, incremental backups with point-in-time recovery
- Database cloning for fast, cost-effective copies
- Supports Global Databases for low-latency global reads and disaster recovery
- Integrated with AWS monitoring and security services

Aurora Cluster Architecture

- Consists of a cluster volume that spans multiple AZs
- One primary instance for read/write operations
- Up to 15 Aurora Replicas for read scaling with low replication lag
- Reader endpoint automatically load-balances read traffic
- Failover to a replica typically takes less than 30 seconds

Aurora Storage

- Distributed, fault-tolerant, self-healing storage
- Automatically replicated six ways across three AZs
- Continuous backup to Amazon S3 without affecting performance
- Supports storage auto-scaling up to 128TB

Aurora Replicas

- Provide read scalability
- Up to 15 Aurora Replicas per cluster
- Replication lag typically under 100 milliseconds
- Supports cross-Region replication with Global Databases
- Failover targets for high availability

Backups and Snapshots

- Automated backups with configurable retention (up to 35 days)
- Point-in-time recovery to any second during retention period
- Manual snapshots retained until deleted
- Snapshots can be shared with other AWS accounts or copied across regions

Global Databases

- Designed for globally distributed applications
- Single primary Region with read-only secondary Regions
- Cross-region replication with typical lag under 1 second
- Supports fast recovery in case of regional outages
- Enables low-latency global reads

Aurora Serverless v1

- On-demand, auto-scaling configuration
- Automatically starts, stops, and scales based on workload
- Ideal for variable or unpredictable workloads
- Charges based on actual usage (Aurora Capacity Units)
- Supports MySQL-compatible edition

Aurora Serverless v2

- Fine-grained, instant scaling with minimal latency
- Supports even higher workloads and connections
- Works with both MySQL- and PostgreSQL-compatible editions
- More production-ready and scalable than v1

Performance and Scaling

- Up to 64 vCPUs and hundreds of GBs of RAM per instance
- Storage and compute scale independently
- Supports read scaling with Aurora Replicas
- Write scaling can be improved with partitioning at the application layer
- Integrated with Amazon RDS Proxy for connection pooling

Security

- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- VPC isolation and subnet-level security

- IAM integration for management
- Database authentication with IAM for MySQL and PostgreSQL
- Audit logging for tracking database activity
- Integration with AWS Secrets Manager for managing credentials

Monitoring and Management

- Amazon CloudWatch for metrics and alarms
- Enhanced Monitoring for OS-level metrics
- Performance Insights for analysing query performance
- Automated backups, patching, and maintenance
- Database cloning for rapid creation of test and development environments

Integration with Other AWS Services

- AWS Lambda for triggers and integrations
- AWS DMS for migrations to Aurora
- AWS Secrets Manager for credential management
- AWS Backup for centralized backup management
- RDS Proxy for improving application scalability and resiliency

Pricing

- Charged per instance hour based on instance type
- Aurora Serverless charged based on Aurora Capacity Units (ACUs)
- Storage billed per GB per month
- I/O requests billed per million requests
- Backup storage beyond the allocated database size incurs additional charges
- Cross-region replication charges for Global Databases

Use Cases

- Enterprise-grade applications requiring high availability and durability
- SaaS applications needing high performance at scale
- Global applications with low-latency read requirements
- Variable workloads with unpredictable usage patterns (Aurora Serverless)
- Online transaction processing (OLTP) systems

Exam Tips

- Aurora is MySQL- and PostgreSQL-compatible but AWS-built for cloud performance
- Supports automatic storage scaling up to 128TB

- Six-way replication across three AZs ensures durability
- Multi-AZ by design with fast failover
- Aurora Replicas provide read scalability with minimal lag
- Global Databases enable low-latency global reads and DR
- Aurora Serverless is ideal for infrequent, unpredictable workloads
- Supports encryption at rest (KMS) and in transit (SSL/TLS)
- Point-in-time recovery and continuous backups to S3
- Database cloning for development and testing without affecting production workloads

Quick Summary

Amazon Aurora is AWS's high-performance, highly available relational database engine compatible with MySQL and PostgreSQL. It offers advanced features like storage auto-scaling, fault-tolerant replication, read replicas, global databases, and serverless deployments, making it an ideal choice for modern, cloud-native applications requiring scalability, durability, and high availability.

Amazon DocumentDB

What It Is

Amazon DocumentDB (with MongoDB compatibility) is a fully managed document database service designed to store, query, and index JSON-like documents. It is optimized for scalability, availability, and performance, and is used for semi-structured data applications that require flexible schemas, such as content management systems, catalogs, and user profiles.

Key Features

- Fully managed document database service
- Compatible with MongoDB APIs (version 3.6, 4.0, 5.0)
- Designed for high performance, scalability, and availability
- Stores semi-structured JSON-like documents
- Automatically scales storage up to 64 TB
- Built-in security, backup, and monitoring features
- Offers automatic failover, backups, and patching
- Integrates with AWS services like CloudWatch, IAM, and Secrets Manager

Architecture and Performance

- Separates compute and storage layers
- Each cluster has one primary instance (read/write) and up to 15 replicas (read-only)
- Six copies of data replicated across three Availability Zones
- Compute instances are based on Amazon EC2
- Automatically scales storage in 10 GB increments as needed
- Supports millions of reads per second with low latency
- Uses a purpose-built storage engine optimized for document workloads

Data Model

- Uses a flexible, schema-less JSON-like format
- Collections contain documents, similar to tables in a relational database
- Supports embedded documents and arrays for nested data
- Ideal for applications with changing or dynamic data structures

MongoDB Compatibility

- Supports MongoDB drivers and tools
- Compatible with popular MongoDB APIs, including find, insert, update, and delete
- Some features such as capped collections and change streams may not be fully supported

- Use AWS DMS to migrate data from existing MongoDB databases

Scaling and Availability

- Horizontal read scaling with up to 15 read replicas
- Automatic failover to replicas during primary failure
- Storage auto-scales without downtime
- Multi-AZ deployments ensure high availability
- Supports replica lag monitoring and alerts

Backups and Restore

- Continuous backups to Amazon S3
- Point-in-time recovery (PITR) up to 35 days
- Manual snapshots can be taken and retained as needed
- Snapshots can be shared across AWS accounts or copied to other regions

Security

- Encryption at rest using AWS KMS
- Encryption in transit using TLS
- VPC deployment for network-level isolation
- IAM integration for resource-level access control
- AWS Secrets Manager integration for secure credentials management
- Audit logging supported using CloudWatch Logs

Monitoring and Management

- Amazon CloudWatch for performance metrics and alarms
- Enhanced Monitoring for detailed OS-level metrics
- Event notifications via Amazon SNS
- Amazon CloudTrail integration for API activity tracking
- Fully managed backups, patching, and failover by AWS

Integration with AWS Services

- AWS DMS for data migration
- CloudWatch for monitoring and logging
- Secrets Manager for managing database credentials
- IAM for access control
- VPC for secure networking
- AWS Glue for data transformation and analytics

Use Cases

- Content management systems
- Product catalogs and inventory
- User profiles and personalization
- Mobile and web applications with flexible schemas
- Semi-structured data ingestion and processing

Exam Tips

- Amazon DocumentDB is MongoDB-compatible but not a native MongoDB engine
- Ideal for applications needing flexible, schema-less document storage
- Not a drop-in replacement for every MongoDB feature
- Multi-AZ support with six-way data replication provides high availability
- Automatically scales storage without downtime
- Supports up to 15 read replicas for horizontal scaling
- Fully managed by AWS including patching, backups, and failover
- Use IAM and VPC for security and access control
- Encryption at rest and in transit is enabled by default
- Integration with CloudWatch and CloudTrail for monitoring and auditing

Quick Summary

Amazon DocumentDB is a scalable, fully managed document database service designed for modern applications that need flexible data models. With MongoDB compatibility, high availability, automated scaling, and integrated AWS security and monitoring, it is ideal for content-rich, semi-structured workloads that require low-latency and resilience.

Amazon DynamoDB

What Is It

- Fully managed, serverless NoSQL database (key-value and document model)
- Single-digit millisecond latency at any scale
- Designed for high availability, durability, and scalability
- Automatically replicates data across multiple AZs
- Integrated with other AWS services (Lambda, AppSync, Glue, etc.)

Core Concepts

- **Table**: Collection of items (like a relational DB table)
- **Item**: Individual record (like a row)
- **Attribute**: Field within an item (like a column)
- **Primary Key**:
    - Partition Key: Determines data partition
    - Partition + Sort Key: Enables sorting and querying within the partition

Indexes

- **Global Secondary Index (GSI)**:
    - Query on different attributes
    - Supports different partition/sort keys
- **Local Secondary Index (LSI)**:
    - Same partition key, different sort key
    - Must be defined at table creation

Capacity Modes

- **On-Demand**:
    - No capacity planning
    - Scales automatically based on traffic
- **Provisioned**:
    - Manually define RCUs/WCUs
    - Can enable Auto Scaling

Performance Optimization

- **DynamoDB Accelerator (DAX)**:
    - In-memory cache
    - Reduces read latency to microseconds

- **Adaptive Capacity**:
    - Automatically adjusts throughput for imbalanced workloads

Data Consistency

- **Eventually Consistent Reads** (default, more scalable)
- **Strongly Consistent Reads** (read after write consistency)
- **Transactions**:
    - ACID-compliant operations across multiple items/tables

Streams

- Capture item-level changes (insert, update, delete)
- Can trigger AWS Lambda for real-time processing
- Provides exactly-once event delivery

Backup and Restore

- **On-Demand Backup**: Full backups, manual
- **Point-in-Time Recovery (PITR)**:
    - Continuous backup up to 35 days
    - Restore to any second within the window

Security

- **Encryption at Rest**: AWS KMS integration
- **IAM Policies**: Fine-grained access control
- **VPC Endpoints**: Private connectivity without Internet exposure

Global Tables

- Multi-Region, multi-active replication
- Low-latency global access
- Automatic conflict resolution
- Supports disaster recovery across regions

Integration

- **Lambda**: Trigger from Streams
- **AppSync**: GraphQL API layer for DynamoDB
- **Glue/Data Pipeline**: ETL and data movement
- **CloudWatch**: Monitoring and alerting
- **Backup**: Centralized backup through AWS Backup
- **Kinesis Firehose**: Data streaming to analytics tools

Pricing

- Charges based on:
    - Read/Write capacity (or On-Demand usage)
    - Storage (per GB/month)
    - Streams, backups, PITR, DAX, Global Tables billed separately

Use Cases

- Real-time bidding platforms
- Gaming leaderboards
- Shopping carts
- IoT telemetry data
- Mobile app backends
- User preference/profile storage

Best Practices

- Use On-Demand mode for unpredictable traffic
- Partition keys should distribute traffic evenly
- Enable Auto Scaling for provisioned workloads
- Use DAX for frequently accessed read-heavy data
- Secure with IAM, KMS, and VPC Endpoints
- Use Streams + Lambda for event-driven architecture

Exam Tips

- Choose On-Demand to avoid provisioning headaches
- LSI = must be defined at creation, GSI = can be added anytime
- Streams trigger Lambda, great for real-time applications
- Global Tables enable low-latency writes and reads across regions
- Transactions enable coordinated writes with ACID compliance
- PITR enables second-level restore up to 35 days
- Know difference between Eventually and Strongly Consistent Reads

Quick Summary

Amazon DynamoDB is a fully managed, serverless NoSQL database delivering high performance, scalability, and resilience. It offers on-demand and provisioned capacity, automatic multi-AZ replication, DAX for caching, PITR for recovery, and global tables for low-latency, multi-region access. Ideal for real-time, serverless, and globally distributed applications.

Amazon ElastiCache

Overview

- Fully managed in-memory data store and cache service
- Supports Redis and Memcached engines
- Provides microsecond latency for real-time applications
- Scales horizontally and vertically
- Used for caching, session storage, real-time analytics

Key Benefits

- Reduces load on databases by caching frequent queries
- Lowers latency and improves application performance
- Fully managed: patching, setup, failure recovery handled by AWS
- Supports multi-AZ with automatic failover (Redis)

Engines

- **Redis**
    - In-memory key-value store supporting data structures
    - Supports persistence (RDB and AOF)
    - Supports replication and automatic failover
    - Pub/Sub messaging capability
    - Advanced features like backups, cluster mode, multi-AZ
- **Memcached**
    - Simple, in-memory key-value store
    - No persistence or replication
    - Supports sharding for horizontal scaling
    - Ideal for straightforward caching use cases

Architecture and Deployment

- Deployed into Amazon VPC
- Nodes hosted on EC2 instances managed by AWS
- Redis supports single-node, replica sets, and cluster mode
- Multi-AZ deployments for high availability (Redis only)
- Automatic backups (Redis)
- Enhanced Monitoring and CloudWatch metrics

High Availability

- Redis: Supports replication groups with primary and replica nodes
- Automatic failover to replica in case of primary node failure
- Multi-AZ failover with minimal downtime
- Backup and restore support for disaster recovery

Scaling

- Vertical scaling by changing node types
- Horizontal scaling via sharding (Redis cluster mode, Memcached)
- Add or remove nodes to meet demand
- Supports online resharding in Redis cluster mode

Data Security

- Encryption in-transit using TLS
- Encryption at-rest with AWS KMS (Redis)
- IAM policies to control API access
- Redis AUTH for password protection
- VPC for network isolation and security groups for access control

Performance Features

- Microsecond latency reads and writes
- Supports large datasets due to memory optimization
- Connection multiplexing to reduce client connections
- Optimized for high-throughput workloads

Backup and Restore (Redis Only)

- Supports automatic daily snapshots
- Manual backups to Amazon S3
- Restore clusters from snapshots
- Useful for disaster recovery and migration

Monitoring and Management

- Amazon CloudWatch integration for metrics
- Enhanced Monitoring for detailed metrics
- AWS CLI, SDKs, Console for management
- Events for operational changes and alerts

Integration with AWS Services

- Works with EC2, Lambda, RDS, DynamoDB, and more
- Used to cache database query results
- Can serve as a session store for stateless applications
- Integrates with AWS CloudFormation for IaC

Use Cases

- Database query caching
- Real-time analytics and leaderboards
- Message brokering (Pub/Sub with Redis)
- Caching API responses

Pricing

- Pay for node hours based on instance type
- Backup storage billed separately (Redis)
- Data transfer within VPC free
- Additional cost for multi-AZ, cluster mode

Best Practices

- Choose Redis for advanced features and persistence
- Use Memcached for simple, ephemeral caching
- Enable multi-AZ for production workloads needing HA
- Monitor usage with CloudWatch and scale nodes appropriately
- Secure access with VPC, IAM, and Redis AUTH

Exam Tips

- Redis supports replication, backups, multi-AZ, cluster mode
- Memcached is simpler, no replication or persistence
- Multi-AZ is Redis-only with automatic failover
- Encryption options available for Redis
- Use ElastiCache to reduce database load and improve latency

Quick Summary

Amazon ElastiCache is a fully managed, in-memory caching service that supports Redis and Memcached to deliver microsecond latency and high throughput for real-time applications. It provides automatic scaling, multi-AZ failover (Redis), encryption, and monitoring, making it ideal for caching, session management, real-time analytics, and reducing database load in scalable cloud architectures.

Amazon Keyspaces (for Apache Cassandra)

What It Is

- Fully managed, serverless, and scalable Apache Cassandra–compatible database service
- Designed for high availability, durability, and low-latency performance
- Serverless, automatically scales up/down based on traffic
- No need to provision, patch, or manage Cassandra clusters
- Compatible with Cassandra Query Language (CQL) and drivers

Key Benefits

- Eliminates operational overhead of managing Cassandra clusters
- Serverless scaling, pay only for what you use
- Automatic replication across multiple AZs for fault tolerance
- Fully managed backup, patching, and maintenance
- Strong consistency and flexible read/write options
- Seamless integration with AWS ecosystem

Architecture and Deployment

- Fully managed, no EC2 or cluster provisioning needed
- Data automatically replicated across 3 Availability Zones
- Supports single-region deployments with multi-AZ redundancy
- CQL-compatible endpoint, no driver code changes required
- Integrates with AWS Identity and Access Management (IAM)
- Serverless: scales partitions and throughput automatically

High Availability and Durability

- Multi-AZ replication for fault tolerance
- Durable writes, all data written to 3 replicas across AZs
- No downtime during scaling or patching
- Highly available reads and writes even during node failures
- Point-in-time recovery (PITR) up to 35 days

Performance and Scaling

- **On-demand capacity mode:** scales automatically with workload
- **Provisioned capacity mode:** manually define read/write throughput
- Latency in single-digit milliseconds
- Automatic load balancing and partition management

- No manual resharding required like traditional Cassandra

Data Security

- Encryption at rest with AWS KMS
- Encryption in transit using TLS
- IAM for fine-grained access control
- VPC endpoints via AWS PrivateLink for private connectivity
- Audit logging using AWS CloudTrail
- No direct SSH or node-level access

Backup and Restore

- Continuous backups with Point-in-Time Recovery (PITR)
- Manual on-demand backups available
- Data stored redundantly across AZs
- Backups can be restored to new tables
- No impact on performance during backup operations

Monitoring and Management

- Integrated with Amazon CloudWatch for metrics
- AWS CloudTrail for auditing API activity
- No need to monitor nodes or repair clusters
- Manage via AWS Console, CLI, or SDKs
- CQL-compatible tools like cqlsh can connect

Integration with AWS Services

- Integrates with Lambda, Kinesis, Glue, Data Pipeline, and S3
- Use with Amazon MSK for streaming writes
- Connect via Amazon VPC endpoints
- Integrates with AWS CloudFormation, Terraform, and CDK for IaC
- Ideal for hybrid or multi-region architectures with data replication

Use Cases

- IOT data ingestion and time-series data
- Session and user profile storage
- High-velocity log data and clickstreams
- Real-time recommendation systems

- Catalog and product metadata storage
- Event tracking and telemetry systems

Pricing

- Pay per request for reads/writes (on-demand mode)
- Provisioned mode: pay for configured capacity
- PITR storage billed separately
- Data transfer within the same AWS Region is free
- Costs for backup storage and restores apply

Best Practices

- Use on-demand mode for unpredictable workloads
- Use provisioned mode for steady traffic to control costs
- Design tables with a well-planned partition key for even load distribution
- Enable PITR for production data protection
- Restrict access via IAM and VPC PrivateLink
- Monitor performance with CloudWatch metrics (throttling, latency)
- Avoid wide partitions distribute writes evenly

Exam Tips

- Serverless Cassandra service, no cluster management required
- Supports CQL and Cassandra drivers
- Automatically replicates data across 3 AZs
- Encryption at-rest and in-transit supported by default
- Two capacity modes: On-Demand & Provisioned
- PITR available for up to 35 days
- No VPC deployment access via PrivateLink if needed
- Great for scalable, low-latency, NoSQL workloads

Quick Summary

Amazon Keyspaces is a fully managed, serverless, and highly available Cassandra-compatible database designed for massive scalability, low-latency workloads, and seamless AWS integration, ideal for use cases like IoT, session management, and real-time analytics without the hassle of cluster administration.

Amazon Neptune

What It Is

Amazon Neptune is a fully managed graph database service designed for applications that need to work with complex, interconnected datasets. It supports graph models like Property Graph (TinkerPop Gremlin) and RDF (SPARQL), making it ideal for use cases like social networks, recommendation engines, fraud detection, and knowledge graphs.

Key Features

- Purpose-built for graph applications
- Supports two major graph models: Property Graph and RDF
- Uses open query languages: Gremlin for Property Graph and SPARQL for RDF
- Optimized for high performance and low-latency graph queries
- Fully managed with built-in backup, patching, and maintenance
- ACID-compliant transactions for data integrity
- Supports encryption at rest and in transit
- High availability with multi-AZ deployments
- Integrated with AWS Identity and Access Management (IAM)
- Built-in integration with CloudWatch, VPC, and other AWS services

Graph Models Supported

- **Property Graph**
    - **Query language:** Apache TinkerPop Gremlin
    - **Use case:** Social networks, fraud detection, real-time recommendations
- **RDF (Resource Description Framework)**
    - Query language: SPARQL
    - Use case: Knowledge graphs, linked data applications, semantic search

Architecture

- Deployed within a VPC for network isolation
- High availability with up to 15 read replicas
- Writer and reader endpoints for separating workloads
- Replicates six copies of data across three Availability Zones
- Automated failover to replicas during node failures
- Backups stored in Amazon S3 with point-in-time recovery

Performance and Scaling

- Designed for high-throughput, low-latency graph queries

- Supports up to 15 read replicas for read scaling
- Automatically detects and recovers from database failures
- Storage scales automatically in 10 GB increments
- Supports IOPS-optimized storage for consistent performance

Security

- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- VPC isolation with subnet-level control
- IAM policies for fine-grained access control
- Database authentication using IAM
- Integration with AWS Secrets Manager for managing credentials

Monitoring and Management

- Integrated with Amazon CloudWatch for monitoring metrics and logs
- Event notifications through Amazon SNS
- Amazon CloudTrail integration for auditing and compliance
- Supports Enhanced Monitoring for resource-level insights
- Maintenance and patching handled by AWS

Data Loading and Migration

- Supports bulk data loading via Amazon S3 using Neptune Bulk Loader
- Supports CSV, RDF, and JSON data formats for import
- AWS DMS can migrate relational data into Neptune
- Compatible with open-source graph tools and libraries

Integration with AWS Services

- Amazon S3 for backup and bulk data loading
- IAM for authentication and authorization
- CloudWatch for monitoring and alerting
- Secrets Manager for secure credential storage
- AWS Glue and DMS for ETL and migration
- VPC for network-level security and access control

Use Cases

- Social networking applications
- Recommendation engines

- Fraud detection systems
- Knowledge graphs and semantic search
- Identity and access management systems
- Network and IT operations graphs

Exam Tips

- Neptune is optimized for graph workloads, not relational or key-value data
- Supports Gremlin and SPARQL query languages for different graph models
- Replication across three AZs with 6 copies of data ensures durability
- Up to 15 read replicas for horizontal scaling of read workloads
- Encryption at rest and in transit is enabled by default
- IAM is used for access control and authentication
- Neptune is not used for OLTP or OLAP workloads
- Use Neptune when relationships between data elements are first-class citizens
- Bulk loading supported through Amazon S3 with the Neptune Loader
- Multi-AZ deployment provides automatic failover and high availability

Quick Summary

Amazon Neptune is a purpose-built, fully managed graph database service ideal for applications that require navigating and analysing complex relationships. It supports both Property Graph (Gremlin) and RDF (SPARQL) models, offers high availability, durability, and seamless AWS integration, making it a strong choice for graph-based use cases like social networks, fraud detection, and knowledge graphs.

Amazon Quantum Ledger Database (QLDB)

What It Is

Amazon QLDB is a fully managed ledger database designed to provide a transparent, immutable, and cryptographically verifiable transaction log. It tracks all changes to application data over time, making it ideal for use cases requiring auditability and data integrity without the overhead of managing a blockchain network.

Key Features

- Fully managed ledger database service
- Immutable transaction log that maintains a complete and verifiable history of changes
- Cryptographic verification with SHA-256 hashes to prove data integrity
- Transparent and append-only journal—no records can be deleted or modified
- Serverless with on-demand pricing and auto scaling
- Supports PartiQL for querying data with familiar SQL-like syntax
- High availability with replication across multiple Availability Zones
- No need to build or manage complex blockchain networks or consensus mechanisms

Ledger Structure

- **Ledger**: Primary container for all data and history
- **Journal**: Immutable transaction log that records every change in order
- **Tables**: Store current state of data with flexible document model (similar to JSON)
- **Document revisions**: Every change creates a new immutable revision
- **Digest**: Cryptographic summary of journal for integrity verification

Immutability and Verification

- Append-only journal guarantees that records cannot be altered or deleted
- Cryptographic hash chaining of entries provides proof that data has not been tampered with
- Periodic digest generation enables audit and verification
- Clients can compare digests to detect unauthorized changes

PartiQL Query Language

- SQL-compatible query language for querying data in tables
- Supports SELECT, INSERT, UPDATE, and DELETE operations
- Allows querying historical revisions using system-generated metadata
- Enables time-travel queries to see the state of a table at any point in time

Performance and Scalability

- Serverless design with no provisioning required
- Auto-scales to support variable workloads
- Supports high availability by replicating data across multiple AZs
- ACID transactions ensure consistency and reliability

Integration with AWS Services

- AWS Identity and Access Management (IAM) for user and permission management
- AWS Key Management Service (KMS) for encryption at rest
- Amazon CloudWatch for monitoring metrics and setting alarms
- AWS CloudTrail for logging API calls and tracking changes
- AWS Lambda for serverless triggers and event processing

Security

- Encryption at rest using AWS KMS-managed keys
- Encryption in transit using TLS
- Fine-grained access control using IAM policies
- Data is replicated for durability and availability across multiple AZs
- Audit logging and API monitoring with CloudTrail

Backups and Availability

- Automatic replication across Availability Zones for high availability
- Journal data is always durable and cannot be deleted
- Snapshots can be exported to Amazon S3 for backup or analysis

Use Cases

- Financial transactions and audit trails
- Supply chain management and tracking
- Insurance claim processing with full history
- Registries of assets such as vehicle or property ownership
- Systems of record requiring immutability and traceability without decentralization

Advantages over Blockchain

- Centralized ledger with no need for consensus algorithms or peer nodes
- No complex network or distributed governance required
- Same cryptographic integrity guarantees as blockchain
- Easier to manage and operate for single-party or regulated environments

Pricing

- Pay only for journal storage, indexed storage, read and write I/O requests
- No upfront costs or server provisioning
- Costs scale with usage

Exam Tips

- QLDB is not a blockchain, it is a centralized ledger database with blockchain-like immutability
- Journal is append-only and cryptographically verifiable
- Ideal for applications needing an authoritative system of record with full auditability
- Supports PartiQL for querying both current and historical data
- ACID transactions ensure data consistency
- Serverless design with automatic scaling and high availability
- Encryption at rest with KMS and in transit with TLS
- Integrates with CloudWatch, IAM, and CloudTrail for monitoring and security

Quick Summary

Amazon QLDB is a fully managed ledger database that delivers an immutable, transparent, and cryptographically verifiable transaction log. It simplifies building audit-ready systems by providing a central, trusted ledger with high availability, flexible querying, and strong security, making it ideal for financial systems, supply chains, and regulatory compliance use cases.

Amazon RDS

What It Is

Amazon RDS is a managed relational database service that makes it easier to set up, operate, and scale databases in the cloud. It automates tasks such as provisioning, patching, backup, recovery, and scaling.

Supported Database Engines

- Amazon Aurora (MySQL and PostgreSQL compatible)
- PostgreSQL
- MySQL
- MariaDB
- Oracle
- Microsoft SQL Server

Key Features

- Managed database service that handles maintenance tasks
- Automated backups with point-in-time recovery
- Automated software patching
- Automatic failure detection and recovery
- Multi-AZ deployments for high availability
- Read replicas for scaling read traffic
- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- Easy scaling of compute and storage

Deployment Options

- **Single-AZ:** Single database instance in one Availability Zone
- **Multi-AZ:** Synchronous standby replica in another AZ for high availability
    - Automated failover to standby on failure
- **Read Replicas:** Asynchronous replication for read-heavy workloads
    - Supports up to 5 read replicas for MySQL, MariaDB, PostgreSQL
    - Aurora supports up to 15 read replicas with low-latency replication

Storage Options

- **General Purpose (SSD)**
    - Cost-effective, balanced performance
    - Suitable for most workloads

- **Provisioned IOPS (SSD)**
    - High-performance storage for critical workloads
    - Consistent low latency and high throughput
- **Magnetic (legacy)**
    - Previous-generation storage, limited use cases

Backups and Snapshots

- **Automated Backups**
    - Enabled by default
    - Retention period configurable from 0 to 35 days
    - Supports point-in-time recovery
- **Manual Snapshots**
    - User-initiated
    - Retained until explicitly deleted
    - Can be shared with other AWS accounts or copied to other regions

Maintenance and Patching

- AWS manages patching of the database engine
- Can specify maintenance window
- Minor version upgrades can be applied automatically or manually

Monitoring and Metrics

- Amazon CloudWatch metrics for CPU, memory, storage, IOPS, connections
- Enhanced Monitoring for OS-level metrics
- Performance Insights for query performance analysis and optimization

Security

- Encryption at rest using AWS KMS
- Encryption in transit using SSL/TLS
- IAM integration for management access
- VPC support for network isolation
- Security groups to control inbound/outbound traffic
- Supports AWS Secrets Manager for credential management
- Option to enforce IAM Database Authentication (MySQL and PostgreSQL)

Pricing

- Pay for instance hours (on-demand or reserved instances)

- Storage and I/O requests billed separately
- Backups stored in S3 are free up to the allocated storage size
- Data transfer costs may apply

Aurora Highlights

- AWS-built, MySQL and PostgreSQL-compatible engine
- Up to 5x faster than standard MySQL, 3x faster than PostgreSQL
- Storage automatically grows up to 128 TB
- Six-way replication across three AZs
- Automated failover, continuous backups to S3
- Aurora Serverless: Auto-scales capacity based on load
- Global Databases for low-latency global reads and disaster recovery

Use Cases

- Web and mobile applications
- Enterprise applications
- E-commerce platforms
- Business analytics
- SaaS applications requiring relational data storage

Exam Tips

- Use multi-AZ for high availability and automatic failover
- Read Replicas are for scaling reads and disaster recovery
- Aurora offers better performance and high availability with low replication lag
- IAM Database Authentication adds IAM-based access control
- Backups support point-in-time recovery
- Encryption at rest uses KMS, in transit uses SSL/TLS
- Performance Insights helps identify slow queries
- Storage Auto Scaling automatically increases storage as needed
- Aurora Global Databases support globally distributed applications
- AWS handles maintenance tasks such as backups and patching

Quick Summary

Amazon RDS is a managed service that simplifies deploying, operating, and scaling relational databases in AWS. It supports multiple database engines, offers automated backups, multi-AZ deployments for high availability, read replicas for scaling, and encryption for data security. Aurora, AWS's proprietary database engine, offers additional performance and scaling benefits.

Amazon Redshift

What It Is

Amazon Redshift is a fully managed, petabyte-scale cloud data warehouse service designed for analyzing large datasets using standard SQL and existing Business Intelligence (BI) tools. It offers high performance, scalability, and cost-effectiveness for data analytics workloads.

Key Features

- Columnar storage format optimized for analytics
- Massively Parallel Processing (MPP) architecture for performance
- Supports standard SQL and integrates with common BI tools
- Automated backups and snapshots
- Redshift Spectrum enables querying data in S3 without loading it into Redshift
- AQUA (Advanced Query Accelerator) improves query performance using hardware acceleration
- Built-in security features including encryption, VPC, IAM, and logging
- Compatible with PostgreSQL drivers and clients

Data Warehouse Architecture

- Redshift cluster consists of a leader node and one or more compute nodes
- Leader node receives queries and distributes them to compute nodes
- Compute nodes process the queries in parallel and return results to the leader node
- Supports provisioned and serverless deployment modes

Redshift Serverless

- Allows running analytics without managing infrastructure
- Automatically provisions and scales compute capacity
- Charges based on Redshift Processing Units (RPUs) used
- Suitable for variable or unpredictable workloads
- Easy to set up, integrates with data in Redshift-managed storage and S3

Data Loading

- Load data using COPY command from S3, DynamoDB, EMR, or other sources
- Supports parallel loading for high performance
- Can use AWS Glue Data Catalog for schema discovery
- Supports SQL-based INSERT statements for small-scale operations

Redshift Spectrum

- Query data in S3 directly using Redshift SQL

- Supports open data formats like CSV, Parquet, ORC, JSON, and Avro
- Integrates with AWS Glue Data Catalog for schema definitions
- Ideal for extending queries beyond data stored in Redshift clusters

Performance Optimization

- Use of distribution styles (KEY, EVEN, ALL) to manage data distribution across nodes
- Sort keys optimize query performance on large tables
- Compression (encoding) reduces storage usage and speeds up processing
- Materialized views store precomputed query results for faster performance
- Workload management (WLM) to prioritize and allocate query resources

Scalability and Elasticity

- Elastic resize for quick cluster scaling
- Concurrency scaling to handle spikes in query load
- RA3 instances with managed storage to scale compute and storage independently
- Cross-region data sharing supported using datashares

Security

- Data encryption at rest using AWS KMS or hardware security modules (HSM)
- SSL encryption for data in transit
- VPC support for network isolation
- IAM for access control and resource policies
- Audit logging with integration to Amazon CloudWatch and AWS CloudTrail

Monitoring and Management

- Performance Insights for query analysis and tuning
- CloudWatch integration for metrics, logs, and alarms
- Automatic backup to S3 with user-defined retention
- Snapshot-based restore options including cross-region snapshots
- Maintenance windows for patching and upgrades

Pricing

- Based on instance type and node count for provisioned clusters
- Redshift Serverless pricing based on RPU-seconds used
- Separate charges for backup storage, concurrency scaling, and data transfer
- No cost for querying S3 data via Redshift Spectrum (only standard S3 and query charges apply)

Use Cases

- Enterprise data warehousing and analytics
- Centralized data platform for BI reporting
- Large-scale log analytics and operational dashboards
- Predictive analytics and machine learning data prep
- Federated queries across data lakes and warehouses

Exam Tips

- Redshift uses columnar storage and MPP for performance
- COPY command is used to bulk-load data from S3
- Spectrum allows querying S3 data directly using Redshift
- RA3 nodes separate compute from storage for flexibility
- Concurrency scaling adds transient clusters for high query loads
- Redshift Serverless offers hands-free provisioning and scaling
- Use sort and distribution keys to optimize query performance
- Supports VPC, encryption, IAM, and logging for security compliance

Quick Summary

Amazon Redshift is AWS's fully managed data warehouse that supports petabyte-scale analytics workloads using standard SQL. With features like columnar storage, MPP architecture, Redshift Spectrum, serverless deployment, and deep AWS integration, it enables fast, scalable, and cost-effective data analysis for modern business needs.

AWS X-Ray

What It Is

AWS X-Ray is a distributed tracing service that helps developers analyse and debug production, distributed applications. It provides insights into application behaviour and performance by collecting data about requests as they travel through an application.

Key Features

- End-to-end request tracing across microservices and components
- Visual service map to see how components interact
- Trace data includes latency, errors, and exceptions
- Helps identify bottlenecks and performance issues
- Supports sampling to control tracing costs
- Works with AWS services like EC2, ECS, Lambda, API Gateway, and more

Core Concepts

- **Segments**: Data collected for individual services handling a request
- **Subsegments**: Fine-grained data within a segment, such as downstream calls or annotations
- **Traces**: Complete end-to-end path of a single request through the system
- **Sampling**: Controls which requests are traced to balance cost and insight
- **Annotations**: Indexed key-value pairs for searching and filtering traces
- **Metadata**: Non-indexed data attached to segments for deeper analysis

How It Works

- X-Ray SDKs instrument applications to record trace data
- Trace data is sent to the X-Ray service
- Service map visualizes connections and latencies between components
- Developers analyse traces to identify errors, slow components, and service dependencies
- Sampling rules define which requests are traced to manage cost

Integrations

- **AWS Lambda**: Automatic integration with tracing enabled
- **API Gateway**: Can pass trace headers to downstream services
- **Elastic Load Balancer**: Passes trace headers to targets
- **ECS and EC2**: SDK-based integration for applications
- **AWS App Runner**: Supports X-Ray integration

- **AWS Step Functions**: Can propagate trace context between steps

Sampling

- Reduces overhead by only tracing a subset of requests
- Default rate of 1 request per second with 5% of additional requests
- Configurable sampling rules per service or endpoint
- Ensures consistent visibility without excessive cost

Security and Permissions

- IAM policies control access to X-Ray APIs and data
- Data is encrypted in transit using TLS
- Data is encrypted at rest in the X-Ray service

Storage and Retention

- Trace data retained by AWS X-Ray for 30 days
- No need to manage storage infrastructure
- Pay only for traces recorded and analysed

Analysis and Troubleshooting

- **Service Map**: Visual representation of request flow between services
- **Trace Timeline**: Breaks down request duration by segment and subsegment
- **Error and Fault Analysis**: Highlights errors, exceptions, and timeouts
- **Filter Expressions**: Search traces by annotation, service, or error type
- **Insights**: Automated root cause analysis of anomalies

Pricing

- **Free tier**: 100,000 traces recorded and 1,000,000 traces retrieved or scanned per month for the first three months
- Pricing based on traces recorded and retrieved beyond free tier
- No cost for integrating X-Ray SDK or agent

Use Cases

- Debugging production issues in microservices architectures
- Analysing and improving application performance
- Identifying latency bottlenecks across services
- Tracing user requests end-to-end across distributed systems
- Compliance and audit of service interactions

Best Practices

- Enable sampling to control costs without losing visibility
- Add meaningful annotations for better filtering and searching
- Monitor service maps regularly to detect unexpected dependencies
- Use X-Ray Insights for proactive detection of anomalies
- Integrate with CI/CD pipelines for continuous observability

Exam Tips

- X-Ray is used for distributed tracing and debugging
- Visualizes service maps and trace timelines
- Supports sampling to manage cost and performance impact
- Integrated with many AWS services, including Lambda and API Gateway
- Helps identify errors, bottlenecks, and dependencies in microservices
- IAM policies control access to X-Ray data and APIs

Quick Summary

AWS X-Ray is a managed service for distributed tracing, helping developers analyse, visualize, and debug applications in production. It enables end-to-end request tracing, identifies performance bottlenecks, and improves observability for microservices and serverless architectures.

AWS Amplify

What It Is

AWS Amplify is a development platform for building secure, scalable full-stack web and mobile applications. It provides a set of tools and services that work together to simplify frontend and backend development using AWS.

Key Features

- Enables rapid development and deployment of web and mobile applications
- Supports frameworks such as React, Angular, Vue, Next.js, iOS, Android, Flutter
- Provides libraries, CLI, UI components, and fully managed hosting
- Supports GraphQL and REST APIs, authentication, storage, analytics, and more
- Integrated with other AWS services like AppSync, Cognito, S3, and Lambda

Amplify CLI

- Command line toolchain for configuring and deploying backend services
- Supports categories such as auth, storage, API, hosting, functions, and more
- Generates infrastructure using AWS CloudFormation templates
- Easily integrates with source control (Git) and CI/CD workflows

Amplify Hosting

- Fully managed CI/CD and hosting service for static web apps
- Supports modern web frameworks like React, Angular, Vue, and Next.js
- Automatically builds, deploys, and hosts apps from Git repositories
- Custom domain setup and SSL support included
- Provides previews for pull requests and branch deployments

Authentication and Authorization

- Uses Amazon Cognito for user sign-up, sign-in, and access control
- Provides built-in UI components for login flows
- Supports multi-factor authentication (MFA), OAuth, and SAML
- Easily integrates with social identity providers like Google, Facebook, and Amazon
- Fine-grained access control via Cognito User Pools and Identity Pools

API Integration

- Supports building and managing GraphQL APIs via AWS AppSync
- Supports building REST APIs using Amazon API Gateway and AWS Lambda
- CLI can generate and manage API schemas and resolvers
- Automatically configures API authorization and access control

Storage

- Integrates with Amazon S3 for file and object storage
- Manages user-specific storage (private/public/protected)
- Upload, download, and manage files using Amplify libraries
- Works with Cognito for secure access to files

DataStore

- Programming model for leveraging shared and distributed data
- Provides conflict resolution and offline data access
- Automatically syncs with backend when reconnected

Functions

- Enables adding backend logic using AWS Lambda
- Write and deploy serverless functions from CLI
- Can be triggered by events (e.g., S3, DynamoDB), APIs, or manually
- Useful for custom authentication, notifications, data processing

Analytics

- Collect user session data and application metrics using Amazon Pinpoint
- Track user events, screen views, and custom analytics
- Integrate with marketing automation and user engagement campaigns

Predictions (AI/ML Integration)

- Simplifies the use of AWS AI/ML services in frontend applications
- Includes services like Rekognition (image analysis), Polly (text-to-speech), Translate, Comprehend
- Requires minimal configuration via Amplify CLI

Push Notifications

- Uses Amazon Pinpoint to configure and send push notifications
- Supports targeted campaigns and transactional messages
- Works with mobile platforms (iOS, Android) through Amplify libraries

CI/CD

- Supports Git-based deployment pipelines (GitHub, GitLab, Bitbucket, AWS CodeCommit)
- Automatically builds and deploys upon code commits
- Provides environment management for dev, test, and production
- Preview URLs generated for every feature branch

Security

- Uses AWS IAM roles and policies for secure access to backend resources
- Amplify CLI supports environment isolation
- Role-based access control for developers and apps

Monitoring and Logging

- Integration with Amazon CloudWatch for logs and metrics
- Monitor Lambda function execution, API errors, and performance
- View deployment status and error messages in Amplify console

Pricing

- Hosting: Based on build and hosting time (per-minute and per-GB transfer rates)
- Backend services: Pay-as-you-go for Cognito, Lambda, S3, etc.
- Free tier available for hosting and backend resources

Use Cases

- Full-stack web and mobile applications
- Serverless applications with real-time data sync and offline access
- MVP and rapid prototyping
- Applications requiring authentication, storage, APIs, and analytics
- React, Vue, and Angular apps with integrated backend

Best Practices

- Use Amplify environments (dev, prod) for safe deployments
- Use CLI and GitHub integration for CI/CD automation
- Secure your storage and APIs using Cognito and IAM

Exam Tips

- Amplify is a full-stack development platform for building cloud-based apps
- Uses Cognito for auth, AppSync/API Gateway for APIs, S3 for storage, and Lambda for backend logic
- Supports real-time, offline, and cross-platform mobile apps via DataStore
- CLI allows infrastructure provisioning using CloudFormation

Quick Summary

AWS Amplify is a full-stack development service that helps frontend and mobile developers build, deploy, and host scalable and secure applications. It offers deep integration with AWS services, real-time data sync, authentication, storage, APIs, and CI/CD, all from a unified CLI and web console.

Amazon API Gateway

What It Is

Amazon API Gateway is a fully managed service that makes it easy to create, publish, maintain, monitor, and secure APIs at any scale. It supports RESTful APIs, WebSocket APIs, and HTTP APIs, enabling developers to build serverless backends and expose AWS services or custom business logic to clients.

Key Features

- Supports REST, WebSocket, and HTTP APIs
- Fully managed, scalable, and highly available
- Request/response transformation with mapping templates
- Integrated authorization and authentication
- Throttling, caching, and quota management
- Monitoring and logging with CloudWatch
- Supports custom domain names and SSL certificates
- Private APIs accessible via VPC Endpoints

API Types

- **REST APIs**: Feature-rich, customizable, supports request validation and mapping
- **HTTP APIs:** Simpler, lower-cost alternative for most REST use cases
- **WebSocket APIs:** Real-time two-way communication between client and server

Integration Types

- **AWS Lambda:** Serverless backend for APIs
- **AWS Service Integrations:** Direct calls to AWS services using IAM roles
- **HTTP/HTTPS Endpoints:** Proxy to external services
- **Mock Integration:** Return mocked responses for testing
- **VPC Links:** Access private resources in a VPC via Network Load Balancer

Authorization and Authentication

- **IAM Authorization:** Control API access using AWS IAM policies and roles
- **Cognito User Pools:** User authentication and JWT token validation
- **Lambda Authorizers (Custom Authorizers):** Run custom logic to authorize requests
- **Resource Policies:** Control access based on IP addresses, VPCs, and AWS accounts

API Keys and Usage Plans

- **API Keys:** Identify and track API consumers
- **Usage Plans:** Define throttling and quota limits for API keys

- Enforce request limits per consumer to manage costs and resources

Throttling and Quotas

- Global and per-method request limits
- Prevents overuse and abuse
- Helps control costs and protect backends

Caching

- Enable response caching at the stage level
- Reduce backend load and improve latency
- Specify cache TTL and encryption options

Deployment and Stages

- Deploy APIs to stages (e.g., dev, test, prod)
- Stage variables for configuration management
- Supports canary deployments for gradual rollout

Custom Domain Names

- Map APIs to custom domains
- Supports ACM-managed SSL/TLS certificates
- Base path mapping for routing multiple APIs under one domain

Monitoring and Logging

- **CloudWatch Metrics:** Track requests, latency, errors, cache hits
- **CloudWatch Logs:** Full request/response logging for debugging
- AWS X-Ray integration for tracing API calls

Private APIs

- Accessible only within specified VPCs using Interface VPC Endpoints
- Secures internal microservices without exposure to the public internet

Security Features

- TLS encryption for data in transit
- IAM-based resource policies
- API keys for managing access to consumers
- AWS WAF integration for protection against common web exploits
- Cognito integration for user authentication and JWT validation
- Resource policies to restrict access to specific IP ranges or AWS accounts

Pricing

- Based on number of API calls and data transfer out
- Separate pricing for REST APIs, HTTP APIs, and WebSocket APIs
- Caching and custom domain name usage incur additional charges

Integration with Other AWS Services

- **AWS Lambda:** Create serverless backends
- **AWS Cognito:** User authentication and authorization
- **AWS Step Functions:** Orchestrate workflows
- **AWS DynamoDB:** Direct service integrations
- **AWS S3:** Serve static content or store API results
- **AWS Kinesis:** Streaming data ingestion
- **AWS SNS/SQS:** Messaging and queuing

Use Cases

- Build RESTful and WebSocket APIs for mobile and web applications
- Create serverless backends with AWS Lambda
- Securely expose AWS services to external applications
- Enable real-time communication with WebSocket APIs
- Build microservice architectures with managed APIs
- Support third-party developer access with API keys and usage plans

Best Practices

- Use IAM, Cognito, or Lambda Authorizers for securing APIs
- Enable request throttling and usage plans to prevent abuse
- Use caching to improve performance and reduce backend load
- Monitor usage with CloudWatch and enable logging for troubleshooting
- Apply resource policies to restrict access to trusted networks or accounts
- Use canary deployments to test new versions safely
- Design APIs for idempotency and error handling

Exam Tips

- REST APIs offer advanced customization and integration features
- HTTP APIs are simpler and cheaper, ideal for most REST use cases
- WebSocket APIs support two-way real-time communication
- Supports multiple integration types including Lambda, AWS services, HTTP endpoints

- Authorization options include IAM, Cognito User Pools, and Lambda Authorizers
- API Keys and Usage Plans enforce rate limiting and quotas
- VPC Links allow private integrations with VPC resources
- Supports custom domain names with ACM-managed certificates
- Can be integrated with AWS WAF for security
- Detailed monitoring with CloudWatch and X-Ray

Quick Summary

Amazon API Gateway is a fully managed service for building, deploying, and securing APIs at scale. It supports REST, HTTP, and WebSocket APIs with flexible integrations, authentication options, throttling, caching, monitoring, and advanced security features for both public and private APIs.

AWS Device Farm

What It Is

AWS Device Farm is a fully managed service for testing mobile (iOS/Android) and web applications on real physical devices hosted in the cloud. It allows developers and QA teams to automate tests, perform remote access, and validate app behaviour across a wide variety of devices without managing their own device lab.

Key Features

- **Real Devices in the Cloud:** Test apps on hundreds of physical devices.
- **Automated Testing:** Supports Appium, Calabash, Espresso, XCTest, and more.
- **Remote Access:** Manually interact with devices through a web browser.
- **Cross-Platform:** Supports Android, iOS, and web applications.
- **Test Reports:** Screenshots, logs, performance data, and videos for analysis.
- **Integration:** Works with CI/CD pipelines (Jenkins, GitHub Actions, GitLab CI, etc.).
- **Device Pools:** Organize devices by type, OS version, and other criteria.
- **Network Simulation:** Test apps under different network conditions.

Testing Options

- **Automated Testing:** Run scripts across multiple devices in parallel.
- **Manual Testing:** Interactively control a device via web browser.
- **Explorer Feature:** Automatic UI exploration to catch crashes or UI issues.

Data & Reports

- **Logs:** Capture device logs, crash reports, and console output.
- **Screenshots/Videos:** Visual evidence of test results.
- **Performance Metrics:** CPU, memory, and network usage.

Security

- **Device Isolation:** Each test runs in a clean device environment.
- **Secure Uploads:** Apps and test scripts are encrypted in transit.
- **Access Control:** Integrates with AWS IAM for permissions management.

CI/CD Integration

- **Jenkins Plugin:** Trigger tests automatically after build.
- **AWS CLI / SDK:** Programmatic control for automated workflows.
- **Pipeline Support:** Easily integrate in DevOps pipelines for continuous testing.

Pricing

- **Pay-as-you-go:** Charged by device minutes.

- No upfront cost for provisioning devices.
- **Cost Optimization Tips:** Use automated parallel tests, clean up unused test runs, schedule idle devices carefully.

Exam Tips

- **Focus on use cases:** SAA may ask when to use Device Farm vs emulators. Correct answer: real device testing without managing hardware.
- **Integration with CI/CD:** Key DevOps angle, "automatic, repeatable testing."
- **Security & Compliance:** Tests run in isolated environments, consider AWS shared responsibility.
- **Device Pools:** Can group devices for targeted testing scenarios.
- **Network Simulation:** Useful for performance/resilience testing questions.

Quick Summary

AWS Device Farm lets you test apps on real devices at scale. It supports automated and manual testing, integrates with CI/CD pipelines, provides detailed logs and metrics, and ensures security and isolation, perfect for SAA exam scenarios where you need to pick a managed service for scalable, reliable, and low-overhead mobile testing.

Amazon Pinpoint

What It Is

Amazon Pinpoint is a fully-managed AWS service for engaging customers across multiple messaging channels: push, in-app, email, SMS, voice, custom. Defined audiences, send campaigns/journeys, analyse behaviour.

Key Features

- **Multi-channel messaging:** Email, SMS/text, push notifications (mobile), voice, in-app, custom channels.
- **Audience segmentation:** Define segments dynamically (based on user behaviour/attributes) or import static segments.
- **Campaigns & journeys:** Create scheduled campaigns; or design multi-step journeys (event-triggered flows) to engage users.
- **Message templates:** Reusable message content/settings for different channels (email templates, push templates, etc).
- **Analytics & metrics:** Track endpoint usage, campaign responses, users, revenue, demographics.
- **Personalisation:** Use attributes and variables in message templates for dynamic, custom content.
- **Integration & extensibility:** Can integrate with mobile/web apps via SDKs; export/import data; tie in with other AWS services.

Core Components & Concepts

- **Project (App)**: The top-level container for your Pinpoint usage.
- **Endpoint**: A destination (e.g., a device token for push, an email address, a phone number) registered in Pinpoint for messaging.
- **Segment**: A group of endpoints/users defined either statically or dynamically.
- **Campaign**: A messaging job sent to a segment, on schedule or immediately.
- **Journey**: A multi-step, event-driven messaging workflow (e.g., user installs app → send welcome push → after 3 days send email if no action)
- **Template**: Pre-defined content and settings for messages that you send via campaigns/journeys.
- **Channel**: A medium of communication (email, SMS, push, voice, custom)
- **Analytics Data**: Behaviour of endpoints/users, message responses, revenue, demographics.

Security / Governance

- Integrates with IAM for access control: you manage which users/roles can create/edit projects, campaigns, segments etc.
- Data encryption at rest & in transit (e.g., uses TLS for API calls) in resilient architectures.

- **Opt-out management:** Especially for email/SMS; campaigns must respect user preferences (relevant for exam catch-question: opt-out means you can't keep sending).

Use Cases

- You have a mobile app and want to send a "Welcome" push notification + in-app message to new users. Use Pinpoint.
- Marketing team wants to send promotional emails and SMS messages to different user segments (e.g., high spenders, inactive users). Use Pinpoint.
- You want to design a multi-channel journey: SMS after app install, if user doesn't engage after 2 days -> email reminder. Pinpoint's "Journeys" feature.
- Track campaign performance (open rates, click rates, revenue per campaign) and feed into metrics dashboards. Use Pinpoint analytics.

Cost & Scalability

- **Fully managed:** You don't manage infrastructure for message delivery. Good for "reduce management overhead" answer.
- Scales with endpoints and messaging volume (channels may have quotas/regulatory requirements especially SMS/voice in regions).
- Cost drivers: number of messages, channels, endpoints, uses of dedicated resources (e.g., dedicated IP for email) & region-specific compliance/reg approvals.
- Regulatory compliance & global presence: When using multi-region architecture, ensure templates, opt-outs, campaigns, etc are aligned across regions.

Exam Tips

- For "journey" or "multi-step user lifecycle messages" → Pinpoint Journeys.
- If the requirement is "no infrastructure management" + "scalable messaging service" → lean toward Pinpoint.
- Be aware of regional/regulatory aspects: SMS/voice may need origination identity registration in countries.
- Know the difference: Pinpoint (customer engagement) vs other services (e.g., SES is email only), etc.

Quick Summary

Amazon Pinpoint is your AWS all-in-one for customer engagement: segments, campaigns, journeys, multi-channel (email/SMS/push/voice), templates, analytics. Use it when you want to engage & track users, not when you only need basic email sending.

AWS Machine Learning

Rekognition

- Face detection and analysis, Labelling of objects
- Celebrity recognition in photos and videos

Transcribe

- Converts speech to text
- Common use case: creating subtitles for audio or video content

Polly

- Converts text to lifelike speech
- Supports multiple languages and voices

Translate

- Language translation service
- Real-time and batch translations across many languages

Lex

- Build conversational interfaces and chatbots
- Natural language understanding and speech recognition

Connect

- Cloud-based contact centre service
- Provides inbound and outbound communication features

Comprehend

- Natural Language Processing (NLP) service, detects sentiment, entities, key phrases

SageMaker

- Managed service to build, train, and deploy machine learning models
- Supports all levels of ML developers and data scientists

Kendra

- Enterprise search service powered by machine learning
- Provides intelligent search results across data sources

Personalize

- Real-time personalization and recommendation service
- Uses ML to tailor recommendations for users

Textract

- Detects and extracts text and data from scanned documents

AWS CloudFormation

What It Is

AWS CloudFormation is an infrastructure-as-code (IaC) service that lets you model, provision, and manage AWS resources (and some third-party resources) via declarative templates (JSON or YAML).
You define everything (instances, networks, databases, permissions) as code; CloudFormation handles the creation, updates, deletions as "stacks".

Key Features

- **Templates:** YAML/JSON file with sections like Parameters, Mappings, Conditions, Resources, Outputs.
- **Stacks:** A deployed collection of AWS resources defined by a template. You manage all resources as one unit.
- **Change Sets:** Preview updates/changes before applying so you can see what will change or be replaced.
- **Nested Stacks / Modules / Registry:** Reuse templates/components for modularizing infrastructure.
- **StackSets:** Deploy stacks across multiple AWS accounts/regions with a single template.
- **Drift Detection:** Detect when stack resources diverge from the declared template.
- **Integration:** Works with CI/CD pipelines (since you're a DevOps intern, this is gold).
- Free to use for the service; you pay for the resources provisioned via CloudFormation.

Core Concepts & Components

- **Template**: The blueprint (YAML/JSON) defining resources.
- **Resources**: The only required section in a template; each resource maps to AWS service (EC2, VPC, S3…).
- **Parameters**: Inputs you can provide when creating/updating stacks for flexibility.
- **Mappings / Conditions**: Provide region/OS/AMI variations or conditionally deploy resources.
- **Outputs**: Values you want to export, or make available after stack creation (e.g., VPC ID).
- **Ref / GetAtt / Fn::ImportValue**: Intrinsic functions for referencing resources, attributes, cross-stack imports.
- **Stack**: The runtime instantiation of a template (i.e., the actual deployed set of resources).
- **Cross-Stack References**: Export outputs from one stack and import into another (modular design).
- **Module / Nested Stack**: Reusable, nested templates/components for better organization.
- **StackSet**: For multi-account/multi-region deployments.

Security, Governance & Cost

- You control access via AWS IAM to allow who can create/update/delete stacks.
- Don't hard code credentials in templates; use secure parameter types, Secrets Manager or SSM Parameter Store.
- Use Stack Policies, drift detection, CloudTrail logging to ensure governance/compliance.
- Cost: CloudFormation itself has no extra charge; you pay for the resources the stack creates.

Use Cases

- You need repeatable infrastructure provisioning across environments (dev/stage/prod). → Use CloudFormation.
- You want version-controlled infrastructure and ability to roll-back changes. → CloudFormation templates + Change Sets.
- You must deploy identical infrastructure in multiple regions/accounts. → Use StackSets.
- You want to modularize infrastructure (e.g., network stack, application stack referencing network outputs). → Use Cross-Stack references/modules.
- You're building a CI/CD pipeline for infrastructure changes (DevOps flow). → Use CloudFormation integration with pipelines, change sets.

Best Practices

- Keep templates modular: separate logical parts of architecture into individual stacks.
- Validate templates (linting, tools like cfn-lint) before deploying.
- Use version control for template files, treat as code (IaC).
- Use Stack Policies/Change Sets before applying to production stacks.

Exam Tips

- If question: "Repeatable, immutable infrastructure" → CloudFormation (or CDK under the hood) is likely.
- If they mention "deal with account/region automation" → StackSets might be the answer.
- If they ask "cost of the service", CloudFormation itself is free; you pay for underlying resources.
- Trick: CloudFormation vs other services (e.g., Terraform), CloudFormation is AWS-native, supported by exam.

Quick Summary

AWS CloudFormation is your go-to for automating and managing AWS infrastructure as code. Define resources in templates, deploy stacks, reuse modules, govern with policies, embed into DevOps pipelines. For the SAA exam you'll often pick it when the requirement emphasizes automation, repeatability, infrastructure lifecycle management, account/region scale, and version-controlled provisioning.

AWS CloudTrail

What It Is

AWS CloudTrail is a service that enables governance, compliance, and operational and risk auditing of your AWS account. It records account activity and API calls across AWS services, delivering log files to Amazon S3, CloudWatch Logs, or CloudWatch Events.

Key Features

- Records AWS Management Console actions, AWS SDKs, command-line tools, and other AWS services API calls
- Tracks who did what, when, and from where in your AWS account
- Delivers event history to S3 buckets for auditing and analysis
- Integrates with CloudWatch Logs for real-time monitoring and alarms
- Integrates with Amazon EventBridge (formerly CloudWatch Events) to automate responses to events

Events and Trails

- An event represents a single AWS API call, including request parameters and response elements
- Trails are configurations that specify how events are logged and delivered
- Single Region or Multi-Region Trails
- Supports logging for all AWS Regions in an account
- Management Events (control plane) record operations on AWS resources
- Data Events (data plane) record resource operations like S3 object-level actions and Lambda function invocations

Types of Events

- Management Events: By default, includes Read and Write operations for resource configuration
- Data Events: Must be explicitly enabled, e.g., S3 GetObject, PutObject, Lambda Invoke
- Insights Events: Identify unusual operational activity (e.g., high rate of API errors)

Trails Configuration

- Can have multiple trails per account
- Apply to single Region or all Regions
- Event logs delivered to S3 bucket specified in trail configuration
- Optional integration with CloudWatch Logs for near real-time monitoring
- Supports encryption of logs with AWS KMS

CloudTrail Event History

- Provides 90 days of management event history for free

- Can view, search, and download events in the AWS console
- Useful for viewing recent account activity without setting up a trail

CloudTrail Insights

- Analyses write management events to detect unusual activity
- Detects spikes in API call volume or error rates
- Provides Insight Events with details about the anomaly
- Helps identify security issues or operational problems

Integration with Other AWS Services

- Amazon S3 for storing logs
- AWS CloudWatch Logs for real-time analysis and alarms
- Amazon EventBridge for automating workflows and security responses
- AWS Lambda to automatically respond to specific events
- AWS Organizations for organization-level trails covering all member accounts

Organization Trails

- AWS Organizations can create a trail that applies to all accounts in the organization
- Centralized logging for governance and compliance
- Prevents member accounts from disabling logging

Security and Access Control

- Logs can be encrypted with AWS KMS for secure storage
- IAM policies control access to CloudTrail logs and settings
- Supports bucket policies and S3 ACLs for secure delivery of logs

Pricing

- Viewing CloudTrail Event History (last 90 days of management events) is free
- First copy of management events recorded in a trail is free
- Charges apply for additional copies, data events, and Insights events
- Storage costs for S3 and optionally CloudWatch Logs ingestion and storage

Best Practices

- Always enable CloudTrail in all Regions to ensure full coverage
- Use organization trails in AWS Organizations for centralized auditing
- Enable log file validation to ensure log integrity
- Enable encryption using AWS KMS for secure log storage
- Store logs in a dedicated S3 bucket with strict access controls

- Integrate with CloudWatch Logs and EventBridge for real-time alerting and automated responses
- Use Insights for detecting anomalous API activity

Exam Tips

- CloudTrail records all API calls made in your account, including from the console, SDKs, CLI, and other AWS services
- Delivers logs to S3, integrates with CloudWatch Logs for monitoring and alarms
- Management Events include control plane operations (e.g., creating resources)
- Data Events must be explicitly enabled (e.g., S3 object-level actions, Lambda invocations)
- Supports organization-wide trails in AWS Organizations
- Insights detect unusual API activity patterns
- Security best practices include encryption with KMS and log file validation

Quick Summary

AWS CloudTrail is the essential AWS service for auditing and compliance, recording API activity across your account and delivering logs to S3 for analysis. It supports real-time monitoring via CloudWatch, integrates with EventBridge for automation, and offers advanced features like Insights for detecting anomalies.

Amazon CloudWatch

What It Is

Amazon CloudWatch is a monitoring and observability service for AWS resources and applications. It provides data collection, visualization, alerting, and automated responses to changes in your environment.

Key Features

- Collects and tracks metrics for AWS resources and custom metrics
- Logs aggregation, storage, search, and analysis
- Alarms for monitoring thresholds and triggering actions
- Dashboards for visualization of metrics and logs
- Events for responding to state changes in AWS resources
- Anomaly detection for identifying unusual metric patterns
- Contributor Insights for analysing high-cardinality log data
- CloudWatch Synthetics for monitoring application endpoints
- ServiceLens for visualizing application performance and dependencies

Metrics

- Automatically collects metrics from AWS services like EC2, RDS, Lambda, and ELB
- Custom metrics can be published by applications
- Standard and detailed monitoring (e.g., 1-minute vs. 5-minute intervals)
- Supports percentile-based metrics and high-resolution (up to 1-second) metrics
- Metric math for creating derived metrics from existing ones

Logs

- Centralized storage and analysis of logs from AWS services and applications
- CloudWatch Logs Agents and AWS Lambda can send logs
- Supports log filtering and metric extraction
- Log groups and log streams for organization
- Subscription filters for streaming logs to other services (e.g., Lambda, Kinesis)
- Log Insights for querying and analysing logs with a SQL-like syntax

Alarms

- Watches metrics and triggers actions when thresholds are breached
- Supports static thresholds and anomaly detection
- Actions include SNS notifications, Auto Scaling policies, and EC2 actions (e.g., stop, terminate, reboot, recover)

- Composite alarms to combine multiple alarms with AND/OR logic

Dashboards

- Customizable views of metrics and logs in a single interface
- Supports text, metric graphs, and log queries
- Can include data from multiple regions and accounts
- JSON-based dashboard definitions for automation

Events (EventBridge Integration)

- CloudWatch Events is now Amazon EventBridge
- Delivers system events describing changes in AWS resources
- Allows setting up rules to match events and trigger targets like Lambda, SNS, or Step Functions
- Supports scheduled (cron) events for automated tasks

CloudWatch Synthetics

- Allows creation of canaries to monitor application endpoints and APIs
- Simulates user interactions to check availability and latency
- Captures screenshots, logs, and metrics for detailed analysis

CloudWatch Contributor Insights

- Analyses time-series log data to identify top contributors
- Helps troubleshoot high-cardinality issues like top talkers or top error sources
- Real-time and historic analysis supported

CloudWatch ServiceLens

- Integrates metrics, logs, and traces for end-to-end observability
- Visualizes service maps and dependencies
- Integrates with AWS X-Ray for distributed tracing
- Helps identify bottlenecks and performance issues across microservices

Retention and Storage

- Metrics retained for 15 months (granularity varies by age)
- Logs retention configurable from days to indefinite
- Log archives can be exported to S3

Security

- IAM policies to control access to CloudWatch resources
- Data encrypted at rest and in transit

- Supports KMS for log group encryption
- AWS CloudTrail integration for auditing API calls

Integration with AWS Services

- Works with EC2, Lambda, RDS, DynamoDB, ELB, ECS, EKS, and many other AWS services
- Supports custom applications publishing metrics via the PutMetricData API
- Integrates with AWS Auto Scaling for dynamic resource management
- Supports cross-account dashboards and metrics

Pricing

- Charged based on metrics, API requests, alarms, dashboards, logs ingested and stored, and canary runs
- Free tier includes basic monitoring metrics and 5GB of log ingestion per month

Best Practices

- Use detailed monitoring for production-critical resources
- Set up alarms for key metrics and use SNS for notifications
- Use log filters to extract custom metrics from logs
- Leverage Contributor Insights for analysing high-volume log data
- Combine metrics, logs, and traces in dashboards for full observability
- Use anomaly detection to avoid hard thresholds for dynamic workloads
- Archive old logs to S3 for cost optimization

Exam Tips

- **CloudWatch Metrics:** monitor AWS resources and custom data
- **CloudWatch Logs:** store, search, and analyse application and system logs
- **Alarms**: trigger actions based on metric thresholds
- **Dashboards**: visualize operational data in one place
- **Events/EventBridge:** respond to state changes and schedule tasks
- **Synthetics:** monitor API and web endpoint availability
- **Contributor Insights:** analyse top contributors in log data
- **ServiceLens:** correlate metrics, logs, and traces for application performance monitoring
- Integration with SNS, Lambda, EC2 Auto Scaling, and many other AWS services

Quick Summary

Amazon CloudWatch is AWS's integrated monitoring and observability service. It provides metrics, logs, alarms, dashboards, events, and advanced features like anomaly detection, Contributor Insights, and ServiceLens to help you understand and manage the performance, health, and availability of your AWS environment and applications.

AWS Command Line Interface (CLI)

What It Is

The AWS CLI is a unified command-line tool that enables you to interact with nearly all AWS services from your terminal. Instead of clicking around the console, you script, automate, and integrate in pipelines. It fits right into the "automate infrastructure/deployment" mindset.

Key Features

- Single tool to manage multiple AWS services via commands.
- Can be installed on Windows/macOS/Linux or used via AWS CloudShell.
- Supports configuration of multiple profiles and regions.
- Can run commands interactively or as part of scripts (CI/CD).
- Outputs in JSON, text, or table format for parsing and readability.
- Supports pagination, filters and powerful querying with --query.
- Integrates with automation frameworks (bash scripts, PowerShell, CI pipelines).

Core Concepts & Syntax

- **Installation & Configuration**:

```
aws configure        # set access key, secret, region, output format
aws configure --profile myprofile
```

- **Profiles**: Use --profile to specify alternative credentials.
- **Region override**: --region us-west-2 allows per-command region override.
- **Output formats**: --output json|text|table

Security / Governance & Cost Considerations

- **Credentials**: Should be stored securely. Do **not** embed access keys in code. Use IAM roles, environment variables, or AWS profiles.
- **Least privilege**: Commands executed via CLI are subject to IAM permissions — ensure users/roles have only necessary rights.
- **Auditing & logging**: Actions via CLI show up in AWS CloudTrail logs (just like console actions), so you're covered for governance.
- **Region/Account awareness**: CLI operations apply to the region/account context. Mistakes (wrong region/account) can lead to resource sprawl or cost surprises.
- **Cost impact**: CLI may automate resource creation/deletion; include cleanup logic in scripts to avoid orphaned resources that drive cost.

Use Cases

- Quick: "You need to script deployment of an S3 bucket + IAM role as part of CI/CD." → CLI is perfect.

- "Automate cleanup of EC2 instances across regions monthly" → Use CLI in a script or via AWS Systems Manager Run Command.
- "You want to query across multiple accounts/profiles for resource status" → CLI profiles + parameterization.
- "You need to integrate AWS commands into Jenkins or GitHub Actions pipeline" → CLI enables programmatic invocation.

Best Practices

- Use named profiles for different environments (dev/test/prod) rather than default profile only.
- Store configuration in ~/.aws/config and ~/.aws/credentials securely, avoid version controlling secrets.
- Use --dry-run or equivalent flags (when available) to check actions before executing.
- Combine CLI commands with infrastructure-as-code (IaC) tools (like AWS CloudFormation or AWS CDK) for repeatable builds.
- Script operations with error handling, logging output, and cleanup to avoid resource leakage.
- Use --output table for human readability in ad-hoc tasks; --output json for automation/parsing.
- Keep the CLI tool up-to-date (aws --version) because AWS adds new services/features constantly.

Exam Tips

- If a question describes "you need to automate repeated operations across AWS using scripts" → CLI is a key candidate.
- When they mention "use command-line to query resource details" → use CLI example.
- Don't confuse CLI with SDK/console — "command line" means CLI.
- Know that CLI actions are subject to IAM permissions and region/account context. If a command fails, misconfigured profile/region or insufficient IAM is probable.
- **Remember**: CLI is tool-agnostic of AWS services; it's the interface to them. So, in exam context pick answers that highlight "scriptable, automatable, less manual, supports multiple services".

Quick Summary

AWS CLI is your go-to command-line tool for controlling AWS services in an automated, scripted fashion. For the SAA exam: think "automation", "script", "multiple services via terminal", "region/profile context", "IAM controls". It's not a service like S3 or EC2, it's the interface you'll use to manage them.

AWS Config

What It Is

AWS Config is a fully managed service that enables you to assess, audit, and evaluate the configurations of your AWS resources. It continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired settings.

Key Features

- Records configuration changes of supported AWS resources
- Captures point-in-time snapshots and changes over time
- Evaluates resource configurations against compliance rules
- Provides a history of resource configuration changes
- Delivers configuration snapshots and change notifications to Amazon S3 and SNS
- Integrates with AWS Organizations for multi-account governance

Configuration Recorder

- Core component of AWS Config
- Records changes to supported AWS resources in your account
- Must be turned on to capture configuration changes
- Defines which resources are recorded (all supported resources or specific types)

Configuration Items (CIs)

- Represent the configuration of a resource at a point in time
- Include metadata, relationships, and current configuration details
- Delivered to S3 bucket in JSON format

Resource Inventory

- Provides a view of all recorded AWS resources and their configurations
- Searchable inventory of resources
- Tracks relationships between resources (e.g., which EC2 instances are in which VPC)

Configuration History

- Detailed history of resource configuration changes over time
- Can analyse historical configurations for troubleshooting and audits
- Stored in S3 for long-term retention and analysis

Configuration Snapshots

- Point-in-time capture of all resource configurations
- Delivered to S3 bucket

- Useful for audits and compliance reporting

AWS Config Rules

- Define desired configurations for resources
- Continuously evaluate resource configurations against these rules
- AWS Managed Rules: Pre-built rules covering common compliance scenarios
- Custom Rules: User-defined, often using AWS Lambda functions
- Rules can be triggered by configuration changes or on a periodic basis
- Noncompliant resources are flagged for remediation

Remediation

- Automate remediation of noncompliant resources
- Use AWS Systems Manager Automation documents to define remediation actions
- Can be triggered automatically or manually from the AWS Config console
- Ensures continuous compliance with policies

Aggregators

- Combine configuration data from multiple accounts and regions
- Centralized view of resource configurations and compliance across an organization
- Supports AWS Organizations to aggregate data from all member accounts

Integration with AWS Organizations

- Multi-account governance with AWS Config aggregators
- Organization-level AWS Config rules for consistent policy enforcement
- Can record and evaluate resources in all accounts of an organization

Delivery Channels

- Define where AWS Config delivers configuration snapshots and compliance notifications
- Typically, an S3 bucket for snapshots and an SNS topic for notifications
- Supports encryption of data at rest using AWS KMS

Security and Access Control

- Supports IAM policies for fine-grained access control
- Encryption of configuration data in S3 using KMS
- Logs delivery to CloudWatch Logs for monitoring

Pricing

- Charged based on the number of recorded configuration items
- Additional charges for active AWS Config rules evaluations

- Aggregators incur charges for cross-account and cross-region data aggregation
- Storage costs in S3 for snapshots and history files

Use Cases

- Compliance auditing and reporting
- Security analysis and configuration monitoring
- Operational troubleshooting and resource tracking
- Change management and governance in multi-account environments
- Continuous enforcement of infrastructure policies

Best Practices

- Enable AWS Config in all regions for complete coverage
- Use AWS Config Rules to enforce security and compliance standards
- Aggregate configuration data across accounts with AWS Organizations
- Automate remediation of noncompliant resources
- Store configuration snapshots and history in a secure, versioned S3 bucket
- Integrate with CloudWatch Logs and SNS for alerting and monitoring

Exam Tips

- AWS Config records and evaluates AWS resource configurations over time
- Supports AWS Managed Rules and Custom Rules for compliance enforcement
- Aggregators provide a centralized view across accounts and regions
- Delivers configuration history and snapshots to S3
- Can trigger notifications via SNS and logs via CloudWatch
- Supports automated remediation using Systems Manager Automation
- Integrates with AWS Organizations for organization-wide governance

Quick Summary

AWS Config is a powerful tool for tracking, auditing, and enforcing AWS resource configurations. It records changes, evaluates compliance against rules, and provides detailed histories to support governance, security, and operational excellence across your AWS environment.

AWS Control Tower

What It Is

AWS Control Tower is a fully managed service that helps you set up and govern a secure, multi-account AWS environment based on AWS best practices. It automates account provisioning, applies governance controls, and streamlines compliance for multi-account AWS setups using a landing zone.

Key Benefits

- Automated landing zone setup for multi-account environments.
- Applies guardrails for governance and compliance.
- Simplifies account provisioning with Account Factory.
- Integrates with AWS Organizations, AWS Config, Service Catalog, and CloudTrail.
- Enables centralized monitoring and policy enforcement.

Core Concepts

1. Landing Zone

A preconfigured, secure AWS environment that follows AWS best practices for setting up new accounts. Includes:

- Centralized logging.
- Account structure (management, audit, and log archive accounts).
- Networking configuration.
- Security baselines.

2. Account Factory

- A tool within Control Tower to automate account creation and provisioning.
- Uses AWS Service Catalog to provide account blueprints.
- Customizes VPC configurations, AWS Region selections, and other settings.

3. Guardrails

- Predefined policies that enforce rules and monitor compliance.
- **Two types:**
    - **Preventive**: Uses Service Control Policies (SCPs) to block actions that violate governance rules.
    - **Detective**: Uses AWS Config rules to monitor compliance and alert on violations.
- **Examples:**
    - Prevent use of unapproved regions.
    - Ensure CloudTrail is enabled.

    - Require MFA for root accounts.

Architecture Overview

- Control Tower builds on:
    - **AWS Organizations**: Manages account hierarchy.
    - **Service Catalog**: Used for Account Factory.
    - **CloudTrail**: Captures API activity logs.
    - **AWS Config**: Tracks resource compliance.
    - **CloudWatch & SNS**: For alerting and monitoring.
- Accounts Created:
    - **Management Account**: Central administration and billing.
    - **Audit Account**: Read-only access to all accounts for compliance.
    - **Log Archive Account**: Centralized storage for CloudTrail and Config logs.

Account Structure

- **Organizational Units (OUs)**: Group accounts logically (e.g., production, development).
- Guardrails are applied at the OU level.
- New accounts can be provisioned into specific OUs.

Customizations for Control Tower (CfCT)

- A framework that allows adding customizations, resources, and configurations to the landing zone.
- Can deploy CloudFormation templates across accounts and OUs.
- Supports lifecycle management for custom resources.

Integration and Extension

- **AWS Config Aggregator**: Central view of configuration compliance.
- **AWS CloudTrail Lake**: Central storage of activity logs.
- **IAM Identity Center (formerly AWS SSO)**: Enables single sign-on access.
- Supports custom SCPs in addition to managed guardrails.

Security and Compliance

- IAM policies, SCPs, and Config rules enforce governance.
- Automatically sets up:
    - CloudTrail logs stored in Log Archive account.
    - AWS Config enabled across accounts.
- Helps with audit readiness and aligns with frameworks like CIS and NIST.

Use Cases

- Establishing multi-account environments with best practices.
- Enforcing security baselines across multiple accounts.
- Centralized governance for enterprises and regulated environments.
- Streamlining account provisioning for dev/test/prod environments.

Pricing

- **No additional cost** for using AWS Control Tower.
- You pay for the underlying services it uses:
    - CloudTrail
    - AWS Config
    - S3
    - AWS Lambda
    - Other resources deployed by Control Tower

Limitations

- Limited customization of SCPs and guardrails within the Control Tower console (can be extended manually).
- Guardrails only apply to accounts **enrolled in Control Tower**.
- Certain AWS Regions may not be supported.
- Some services may not be **Control Tower-aware**.

Exam Tips

- Control Tower simplifies multi-account AWS governance by using landing zones, guardrails, and Account Factory.
- Guardrails are either preventive (SCPs) or detective (Config rules).
- Account Factory uses AWS Service Catalog for account provisioning.
- Logs and compliance data are centrally stored in designated accounts.
- Works with AWS Organizations to manage multiple accounts.
- Know the three key accounts: Management, Audit, Log Archive.
- Use Customizations for Control Tower for advanced control and resource deployment.

Quick Summary

AWS Control Tower helps you build, manage, and govern a secure, multi-account AWS environment at scale using preconfigured landing zones, automated account provisioning, and governance guardrails. It streamlines compliance, centralizes logging, and integrates tightly with other AWS services like IAM Identity Center, AWS Organizations, and AWS Config.

AWS Management Console

What It Is

The AWS Management Console is a web-based interface that lets you interact with and manage your AWS resources visually without needing the CLI or SDKs. Think of it as your control room for everything in AWS.

Key Features

- **Web-Based UI:** Manage and configure AWS services from your browser.
- **Global Search Bar:** Quickly access services, documentation, and recent resources.
- **Resource Dashboard:** Monitor active resources, costs, and billing data.
- **Service Favourites:** Pin your frequently used AWS services for faster access.
- **Account Management:** Manage users, billing, regions, and support plans.
- **Integrated Access:** Launch CloudShell, connect to Cloud9, or switch to CLI seamlessly.

Use Cases

- Spinning up EC2 instances, RDS databases, or S3 buckets manually.
- Managing IAM users, roles, and permissions.
- Monitoring application health and performance using CloudWatch dashboards.
- Viewing service quotas and billing summaries.

Exam Tips

- AWS Console is region-specific, always check your region in the top-right corner!
- Some services (like IAM or CloudFront) are global, meaning they aren't tied to regions.
- You can access AWS CloudShell directly from the console to execute CLI commands.
- For automation, prefer AWS CLI, SDKs, or CloudFormation, the console is mostly for setup, testing, or visualization.

Quick Summary

A web-based GUI that gives you full control of your AWS resources, ideal for visual management, quick deployments, and billing insights. It's user-friendly but not meant for automation.

AWS Health Dashboard

What It Is

The AWS Health Dashboard gives you real-time visibility into the status and health of AWS services globally and specific to your account.
It helps you stay informed about service disruptions, planned maintenance, and resource-specific issues that could affect your workloads.

There are two main views:

1. **Service Health Dashboard –** Public view of AWS service status across all regions.
2. **Your Account Health Dashboard –** Personalized view showing issues impacting *your* AWS resources.

Key Features

- **Service Status Updates:** See outages, degradation, or maintenance events in real-time.
- **Personalized Notifications:** Get alerts for your account-specific incidents.
- **Event History:** Review past issues and maintenance events.
- **Integration with CloudWatch & SNS:** Automate notifications for operational awareness.
- **APIs Available:** Access programmatically using AWS Health API (great for automation).

Use Cases

- Monitoring service interruptions that might impact production workloads.
- Tracking AWS infrastructure health during incident management.
- Setting up alerting systems via SNS or EventBridge for proactive responses.
- Reviewing post-incident events for compliance and RCA (root cause analysis).

Exam Tips

- Service Health Dashboard: Accessible publicly without login shows AWS-wide events.
- Account Health Dashboard: Requires AWS credentials shows issues related to your AWS resources.
- AWS Health API can be integrated with monitoring tools for proactive alerts.
- Often tested in exam questions related to troubleshooting, monitoring, and resiliency.

Quick Summary

Your window into AWS's operational health, track global and account-specific incidents, get proactive alerts, and ensure your architecture remains resilient against service disruptions.

AWS License Manager

What It Is

AWS License Manager is a service that helps you manage, track, and control software licenses in AWS and on-premises environments. It simplifies license administration, reduces the risk of non-compliance, and supports license optimization by enforcing usage rules.

Key Features

- Centralized license management for AWS and on-premises resources
- Supports BYOL (Bring Your Own License) models
- Define rules to track license usage and enforce limits
- Integrates with AWS Systems Manager for discovery of installed software
- Provides visibility into license usage across AWS accounts in an organization
- Supports integration with AWS Marketplace for license management of Marketplace products

License Configurations

- Define custom licensing rules including:
    - Number of vCPUs or cores
    - Number of instances or sockets
    - Licensing metrics (e.g., per VM, per core)
- Create rules that align with your vendor agreements
- Enforce compliance by preventing overuse of licenses

License Tracking

- Automatically track license consumption on AWS resources
- Supports tracking for:
    - EC2 instances
    - On-premises servers connected via AWS Systems Manager
- Visibility into resource usage against defined limits
- Generate reports for audit and compliance purposes

Enforcement

- Enforce licensing rules at launch time
- Prevents creation of resources that exceed defined license limits
- Integration with AWS Service Catalog to ensure compliant deployments
- Helps ensure usage stays within vendor agreements

Integration with AWS Systems Manager

- AWS Systems Manager Inventory discovers installed software on instances
- License Manager uses inventory data to track and manage licenses
- Supports hybrid environments (AWS and on-premises)

Cross-Account Management

- Integrated with AWS Organizations for centralized license tracking
- Share license configurations across multiple AWS accounts
- Enable governance in multi-account environments

AWS Marketplace Integration

- Supports license management for Marketplace products
- Track and manage entitlements purchased through AWS Marketplace
- Centralized view of Marketplace software usage

Reporting and Visibility

- Dashboard to view license configurations, usage, and compliance status
- Detailed reports to help with audits and vendor negotiations
- Alerts for nearing license limits or non-compliance risks

BYOL (Bring Your Own License)

- Supports bringing existing licenses to AWS
- Define and enforce rules to ensure BYOL usage complies with vendor terms
- Track usage to help maintain compliance

Security and Access Control

- Integration with AWS Identity and Access Management (IAM)
- Fine-grained permissions to control who can create, modify, or delete license configurations
- AWS CloudTrail logging for API calls to support auditing and compliance

Pricing

- No additional charge for using AWS License Manager
- Pay for the underlying AWS resources you use
- Helps reduce costs by optimizing license utilization and avoiding over-purchasing

Use Cases

- Enforcing vendor licensing rules in AWS environments
- Centralized license tracking for multi-account AWS setups

- Managing software license compliance for hybrid (AWS + on-premises) environments
- Optimizing costs through better visibility into license usage
- Supporting BYOL strategies for existing software investments

Best Practices

- Define clear license configurations that match your vendor agreements
- Integrate with AWS Systems Manager for discovery and tracking
- Enable cross-account sharing for Organizations to standardize governance
- Regularly review usage reports to identify optimization opportunities
- Use IAM policies to control access to License Manager configurations

Exam Tips

- AWS License Manager helps track and enforce licensing rules in AWS and on-premises
- Supports BYOL by defining custom licensing rules
- Integrated with Systems Manager for inventory and tracking
- Can enforce rules at launch to prevent non-compliance
- Supports cross-account management with AWS Organizations
- Helps manage AWS Marketplace licenses in addition to BYOL software

Quick Summary

AWS License Manager is a service that enables customers to centrally manage and track software licenses across AWS and on-premises environments. It supports BYOL, enforces license usage rules to ensure compliance, integrates with AWS Systems Manager for discovery, and provides visibility and reporting to help reduce costs and maintain audit readiness.

Amazon Managed Grafana

What It Is

Amazon Managed Grafana is a fully managed service for open-source Grafana that makes it easy to visualize and analyse operational data at scale. It helps you create, explore, and share dashboards to monitor AWS resources and other data sources without needing to manage Grafana infrastructure yourself.

Key Features

- Fully managed, highly available, and scalable Grafana workspace
- Integration with AWS data sources such as CloudWatch, X-Ray, Timestream, and IoT SiteWise
- Supports over 40 data sources including Prometheus, Elasticsearch, InfluxDB, and MySQL
- Built-in security features with AWS SSO and IAM Identity Center integration
- Supports Grafana Enterprise plugins for enhanced capabilities
- Automatic version updates and patching
- Supports creating, sharing, and embedding dashboards
- Alerting with multi-channel notifications including SNS and Slack

Workspaces

- Logical Grafana environments in your AWS account
- Multi-user support with AWS SSO for access control
- Each workspace is isolated and secure
- Supports provisioning multiple workspaces for different teams or projects

Data Sources

- Native integration with AWS services including:
    - Amazon CloudWatch
    - AWS X-Ray
    - AWS IoT SiteWise
    - Amazon Timestream
    - AWS Managed Prometheus
- Also supports popular third-party sources like:
    - Prometheus
    - Elasticsearch
    - InfluxDB
    - MySQL/PostgreSQL

- OpenSearch Service

Security and Access Control

- Integrated with AWS SSO for user authentication and authorization
- Supports IAM policies for fine-grained workspace access
- Encryption at rest using AWS KMS
- VPC connectivity for private data sources
- Audit logging with AWS CloudTrail

Alerting

- Define thresholds and conditions on metrics
- Supports multi-channel notification integrations such as SNS, Slack, and email
- Managed Grafana alerting supports deduplication and silencing
- Alert rule and notification policy management within Grafana

Grafana Enterprise Features

- Optional access to Enterprise plugins (available for additional cost)
- Examples: ServiceNow, Splunk, Datadog, Dynatrace
- Advanced reporting and data source integrations

Integration with AWS Services

- Amazon CloudWatch: Native dashboard creation for AWS metrics
- AWS X-Ray: Visualize application traces
- AWS IoT SiteWise: Industrial data visualization
- Amazon Timestream: Time-series data analysis
- AWS Managed Service for Prometheus: Scrape and visualize Prometheus metrics

Deployment and Management

- AWS fully manages setup, patching, scaling, and availability
- Simple provisioning through the AWS Console, CLI, or CloudFormation
- Automatic upgrades to latest Grafana versions
- Supports infrastructure-as-code for workspace creation and configuration

Pricing

- Billed per active user per workspace per month
- Free tier available for trial usage
- Additional charges for Grafana Enterprise plugin usage
- Data transfer costs may apply when querying external data sources

Use Cases

- Monitoring AWS infrastructure with CloudWatch metrics
- Centralized observability for microservices with Prometheus and AWS X-Ray
- Industrial IoT monitoring with AWS IoT SiteWise
- Operational dashboards for DevOps and SRE teams
- Correlating logs, metrics, and traces in a single view

Best Practices

- Use AWS SSO for secure user management and access control
- Leverage IAM policies to restrict workspace access
- Enable audit logging for compliance and visibility
- Organize workspaces by team or project to manage isolation
- Integrate with SNS for alerting to AWS services and other endpoints

Exam Tips

- Managed Grafana is AWS's fully managed, secure, scalable Grafana service
- Supports integration with AWS-native and third-party data sources
- AWS SSO and IAM help manage user access securely
- Use for building dashboards to monitor AWS services, application traces, and custom metrics
- Fully managed deployment means AWS handles patching, scaling, and availability
- Alerts can notify via SNS, email, Slack, and other channels

Quick Summary

Amazon Managed Grafana simplifies creating and managing Grafana dashboards for monitoring AWS and third-party data sources. It is fully managed by AWS, supports secure access with SSO and IAM, integrates natively with AWS services, offers enterprise plugin options, and provides advanced alerting to help teams monitor and respond to operational events effectively.

Amazon Managed Service for Prometheus

What It Is

Amazon Managed Service for Prometheus (AMP) is a fully managed, highly available, and secure monitoring service that is compatible with open-source Prometheus. It lets you collect, store, and query operational metrics at scale without managing the underlying infrastructure.

Key Features

- Fully managed, serverless, and automatically scalable
- Prometheus-compatible querying using PromQL
- High availability across multiple Availability Zones
- Integrated with AWS security services like IAM and AWS PrivateLink
- Secure data encryption at rest and in transit using AWS KMS
- Supports AWS native services for metric collection (e.g., Amazon EKS, ECS, EC2)
- Pay-as-you-go pricing with no upfront costs

Prometheus Compatibility

- Works with open-source Prometheus clients and tools
- Supports remote write for sending metrics from existing Prometheus servers
- Allows using PromQL for querying metrics
- Compatible with existing Prometheus exporters and instrumentation libraries

Data Ingestion and Storage

- Supports remote write API for pushing metrics
- Automatically scales ingestion based on workload
- Durable, long-term storage of metrics
- Data replication across multiple Availability Zones for high availability

Querying and Analysis

- Supports PromQL for flexible metric querying
- Integrated with Amazon Managed Grafana for visualization
- Allows building dashboards and alerts based on Prometheus metrics
- Low-latency query performance even at scale

Security and Access Control

- IAM policies for fine-grained access management
- AWS PrivateLink for secure, private connectivity to AMP endpoints
- Encryption of data in transit and at rest using AWS KMS
- Integration with AWS CloudTrail for auditing API calls

- Resource tagging for cost allocation and management

Integration with AWS Services

- Amazon EKS: Native integration for scraping metrics from Kubernetes clusters
- AWS Distro for OpenTelemetry: Supports metrics collection from various sources
- Amazon Managed Grafana: For building dashboards with Prometheus metrics
- AWS CloudWatch: Can be used alongside for logs and other metrics

Pricing

- Based on ingestion (data written), storage (GB-month), and query requests
- No upfront fees or long-term commitments
- Pay only for what you use, making it cost-efficient for workloads with variable monitoring needs

Use Cases

- Monitoring Kubernetes workloads on Amazon EKS
- Collecting and analysing application metrics using Prometheus exporters
- Creating custom alerts and dashboards using PromQL and Grafana
- Centralizing metrics from multiple AWS and on-premises environments
- Supporting SRE and DevOps practices with scalable, highly available metric storage

Best Practices

- Use IAM policies to control access to AMP workspaces
- Leverage AWS PrivateLink for secure, private connectivity
- Tag resources for easier cost allocation and management
- Design retention policies to balance compliance needs and cost

Exam Tips

- AMP is fully managed and Prometheus-compatible
- Supports PromQL for querying metrics
- Works with remote write for collecting data from existing Prometheus servers
- Integrates with AWS IAM for access control and AWS PrivateLink for secure connections
- Pay-as-you-go pricing model without upfront costs

Quick Summary

Amazon Managed Service for Prometheus enables customers to monitor and analyse metrics from AWS and on-premises environments using familiar Prometheus tools and PromQL, without worrying about the operational complexity of scaling and managing Prometheus infrastructure. It integrates with AWS security, networking, and visualization services to provide a secure, cost-effective, and easy-to-use monitoring solution.

AWS Organizations

What It Is

AWS Organizations is a free service that helps you centrally manage and govern multiple AWS accounts at scale. It provides a framework for consolidating billing, managing policies, and enforcing governance across accounts in a single organization.

Key Benefits

- Centralized billing and cost management.
- Simplified account creation and lifecycle management.
- Policy-based governance with Service Control Policies (SCPs).
- Organize accounts in Organizational Units (OUs).
- Enforce security and compliance consistently.
- Integration with AWS Control Tower, AWS RAM, and other services.

Core Concepts

1. Organization

- A collection of AWS accounts under a single management account.
- Can include hundreds or thousands of member accounts.

2. Management Account

- Formerly called Master Account.
- Has full administrative control over the organization.
- Pays the consolidated bill for all member accounts.

3. Member Accounts

- Accounts within the organization.
- Managed via the management account.
- Can have restrictions enforced via policies.

4. Organizational Units (OUs)

- Logical groupings of accounts within the organization.
- Can be nested hierarchically.
- Policies (like SCPs) can be applied at the OU level to govern multiple accounts together.

Features and Capabilities

Consolidated Billing

- Combines charges for all accounts into one bill.
- Benefits:

    - Share volume discounts across accounts.
    - Single payment method.
    - Simplified billing and reporting.
- No extra cost to use.
- Billing and usage tracked at both organization and individual account levels.

Service Control Policies (SCPs)

- Policies to define maximum available permissions for accounts or OUs.
- Work with IAM policies: SCPs set boundaries; IAM policies grant permissions within those boundaries.
- Can allow or deny specific AWS actions.
- SCP Types:
    - FullAWSAccess: Default SCP attached to root OU and new accounts.
    - Custom SCPs for fine-grained control.
- Examples:
    - Deny usage of certain regions.
    - Block deletion of specific resources.

AWS Single Sign-On (IAM Identity Center) Integration

- Enables centralized access management across accounts.
- Assign user permissions across AWS accounts.
- Supports SAML integration with external identity providers.

Account Creation and Invitation

- Create new accounts programmatically or via AWS Management Console.
- Invite existing standalone AWS accounts to join the organization.
- Central governance is enforced upon joining.

Tagging Support

- Tag AWS accounts for
- Helps with tracking and reporting. cost allocation and organization.

Policy Types

- **Service Control Policies (SCPs)**:
    - Control what services and actions accounts can use.
- **Tag Policies**:
    - Enforce consistent tagging across resources in accounts.

    - Define allowed tag keys and values.
- **Backup Policies**:
    - Define and enforce backup plans across accounts.
    - Automate AWS Backup settings.
- **AI Services Opt-Out Policies**:
    - Control whether AWS AI services can store and use customer content for service improvements.

Integration with AWS Services

- **AWS Control Tower**:
    - Builds landing zones on top of Organizations.
- **AWS Resource Access Manager (RAM)**:
    - Share resources across accounts.
- **AWS Budgets**:
    - Track and enforce cost controls across the organization.
- **AWS Config**:
    - Aggregate configuration data across accounts.
- **CloudTrail**:
    - Centralize logging across all accounts.

Security and Access Control

- SCPs enforce organization-wide permission guardrails.
- IAM policies work within the boundaries defined by SCPs.
- Management account can create and manage all OUs and member accounts.
- AWS CloudTrail can log all account activities centrally.

Best Practices

- Use OUs to reflect organizational structure (e.g., production, development, testing).
- Apply SCPs to enforce least privilege.
- Enable consolidated billing to maximize savings.
- Use tag policies to maintain resource consistency.
- Monitor and audit with AWS Config and CloudTrail.
- Enable AWS Budgets for cost tracking and alerts.

Pricing

- AWS Organizations itself is free.
- You pay only for the AWS services used by the accounts.
- No extra charges for consolidated billing or policy enforcement.

Limitations

- SCPs do not grant permissions; they only limit what is allowed.
- Not all AWS services fully support AWS Organizations features.
- Changes to SCPs can impact account operations immediately.
- Accounts can only belong to one organization at a time.

Common Use Cases

- Centralized billing for multiple AWS accounts.
- Enforcing security policies across a multi-account environment.
- Isolating workloads by account (e.g., dev, test, prod).
- Simplifying compliance and governance reporting.
- Sharing resources securely across accounts with AWS RAM.

Exam Tips

- SCPs are used to enforce permission boundaries, not grant permissions.
- Management account has full admin control over the organization.
- Consolidated billing allows sharing volume discounts and centralized payment.
- Organizational Units (OUs) enable applying policies to groups of accounts.
- AWS Organizations integrates tightly with AWS Control Tower, RAM, and IAM Identity Center.
- Tag Policies ensure consistent tagging across accounts.
- Backup Policies automate and enforce backup plans organization-wide.

Quick Summary

AWS Organizations helps you centrally manage, secure, and govern multiple AWS accounts by grouping them into Organizational Units, enforcing Service Control Policies, enabling consolidated billing, and integrating with IAM Identity Center, AWS Control Tower, and AWS RAM for streamlined multi-account management.

AWS Proton

What It Is

AWS Proton is a managed delivery service for provisioning and deploying container-based and serverless applications. It helps platform teams define standard stacks (infrastructure + CI/CD pipelines) as templates, and developers deploy services from those templates, so you get consistency, speed and governance.

Key Features

- Template-based architecture: supports Environment Templates (shared infrastructure) and Service Templates (application stacks)
- Infrastructure as Code support: Works with AWS CloudFormation and Terraform for provisioning
- Versioned templates: Define major/minor versions of templates; enforce standardization and upgrade paths
- Template sync & Git integration: Changes to template bundles in repositories can automatically create new versions
- Automated CI/CD pipeline provisioning along with infrastructure for each service instance
- Environment provisioning options: AWS-managed, self-managed, and cross-account environments

Core Concepts & Components

- **Environment Template** – Defines shared infrastructure used by many services (VPCs, clusters, load balancers)
- **Service Template** – Blueprint for a specific application or micro-service stack, including infrastructure and deployment pipeline
- **Service Instance** – A deployed instance of a Service Template within an Environment
- **Component** – Optional additional infrastructure that attaches to a service instance
- **Template Bundle** – ZIP/YAML bundle stored in S3 or Git repo containing IaC files, manifest, and schema for templates
- **Template Versions** – Minor versions are backward-compatible, while major versions may introduce breaking changes

Security, Governance & Cost

- Centralized control ensures developers use only approved architecture
- IAM roles and policies manage access to template registration, deployments, and environment management
- No additional Proton service cost, you pay only for the AWS resources provisioned by your templates
- Great for compliance and infrastructure standardization, preventing configuration drift

Use Cases

- Platform teams want to define standardized micro-service stacks (containers/serverless) and allow self-service deployments → Use Proton
- Enforcing architecture standards and rolling out updates across many micro-services → Proton's version control fits perfectly
- Need automated provisioning of both infrastructure and CI/CD pipelines for each new service → Proton handles both seamlessly

Best Practices

- Keep environment and service templates modular and easy to maintain
- Use versioning to handle infrastructure evolution and migration between template versions
- Integrate Proton with CloudFormation or Terraform for seamless DevOps workflows
- Continuously monitor deployments, automate updates, and track template drift

Exam Tips

- Keyword clues: "self-service deployment", "standardized templates", "automated CI/CD", "versioned templates"
- Proton is delivery-oriented, not just IaC, adds governance and self-service to infrastructure
- Don't confuse Proton with Service Catalog (which manages approved product stacks) or plain CloudFormation (which provisions resources)
- For questions about "governed, standardized app deployments at scale" → Proton is the answer

Quick Summary

AWS Proton helps teams standardize, automate, and manage deployment of containerized and serverless applications. It bridges the gap between DevOps automation and platform governance, empowering developers to deploy while keeping control centralized.

AWS Service Catalog

What It Is

AWS Service Catalog lets you create, manage and distribute a catalog of approved AWS-resource "products" (stacks, services, templates) to end-users. Administrators define what resources are approved, bundled, and how they can be used; end-users deploy from the catalog within guardrails.

Key Features

- **Standardization of assets:** Administrators define products via CloudFormation templates, version them, bundle them into portfolios.
- **Self-service discovery & launch:** End-users browse available products, launch with constrained parameters.
- **Fine-grained access control:** IAM + launch constraints, template constraints, tag update constraints.
- **Versioning & sharing:** Products can have active/inactive versions; portfolios can be shared across accounts.

Core Concepts

- **Product**: A CloudFormation template (or packaged AWS Marketplace offering) that defines one or more AWS resources.
- **Portfolio**: A collection of products, with configuration for who can access/deploy them, constraints applied, tags inherited.
- **Versions**: Each product can have multiple versions; you manage active/inactive/deleted states.
- **Constraints**: Template constraints (limit parameters), launch constraints (specify IAM role used when product is deployed), notification constraints (SNS notifications on stack events), tag update constraints (whether end-user can change tags).
- **Service Actions**: Additional actions end-users can perform on provisioned products (for example, run a System's Manager document, update resources).
- **AppRegistry (integration)**: Manage application metadata, attribute groups, associating stacks/provisioned products with applications for visibility.

Security & Governance & Cost

- Works with IAM for who can create/edit catalogs, portfolios, access products.
- Data in S3/DynamoDB used under the hood is encrypted at rest; communication via TLS.
- Integration with CloudTrail for auditing provisioning and changes.
- Pricing: Free tier includes first 1,000 API calls per month; charge starts on API usage beyond free tier.
- Because products often create AWS resources via CloudFormation, underlying resource cost still applies.

Use Cases

- Large organization wants to *standardize* infrastructure for developers; teams can only deploy approved stacks with preset configurations → Use Service Catalog.
- You want self-service provisioning by developers but still enforce governance (tagging policies, resource types, network configurations) → Use Service Catalog.
- Multi-account architecture: Central team shares portfolios into multiple accounts; each account uses same approved products → Service Catalog supports cross-account sharing.
- Track and manage application metadata across multiple stacks and accounts with AppRegistry integration → Service Catalog + AppRegistry.

Best Practices

- Define **clear portfolios** aligned with environments (dev/test/prod) and user roles.
- Use **launch constraints and template constraints** to restrict parameters (e.g., instance type, storage size) so you maintain cost control and compliance.
- Use **versioning** for product templates: when an update is needed, create new version, mark old inactive, migrate existing instances carefully.
- Integrate with tagging strategy: when launching a product, tags from portfolio should propagate into all underlying resources for tracking and cost allocation.
- Combine with automation: Use CloudFormation templates for products, integrate Service Catalog into CI/CD so updates go through code and review.
- Monitor usage of products, ensure resources spun via Service Catalog follow policies (cost, security).

Exam Tips

- Differentiate from simple CloudFormation: While CloudFormation defines infrastructure, Service Catalog manages distribution and governance of those defined infrastructures.
- Recognize key terms: Product, Portfolio, Version, Constraints, Provisioned Product.
- If question involves *multi-account distribution of standard stacks*, Service Catalog is a strong candidate.

Quick Summary

AWS Service Catalog allows you to create, manage and distribute a curated set of infrastructure products (via CloudFormation templates) so that developers or end-users can deploy what they need, but only what you've approved and configured. Think of it as a *governed store* of architecture blueprints: you get consistency, control, compliance; developers get self-service. On the SAA exam: when you see *self-service provisioning + standardization + guardrails*, the answer often points to Service Catalog.

AWS Systems Manager (SSM)

What It Is:

- A centralized operations hub to manage infrastructure and systems on AWS and on-premises.
- Helps with automation, patching, inventory, monitoring, configuration, and remote command execution.

Key Capabilities:

1. SSM Agent

- Installed on EC2 or on-prem machines.
- Enables Systems Manager to communicate with the instance.

2. Session Manager

- Secure shell access (SSH-like) to EC2 instances without needing a bastion host or open ports.
- Uses IAM permissions.
- Fully auditable with CloudTrail and CloudWatch Logs.

3. Run Command

- Run shell scripts or PowerShell commands across multiple EC2 or on-prem instances without logging in.
- Requires SSM Agent and appropriate IAM permissions.

4. Automation

- Automate repetitive tasks like instance start/stop, patching, backups.
- Use Automation Documents (runbooks) can be prebuilt or custom.

5. Patch Manager

- Automates OS patching for EC2 and on-prem.
- You can set patch baselines and maintenance windows.

6. Parameter Store

- Store configuration data and secrets securely.
- Supports plain text and encrypted (via KMS) values.
- Used by Lambda, EC2, and other services for dynamic configuration.

7. Inventory

- Collect metadata from EC2/on-prem instances (e.g., installed apps, network config).
- Helps with compliance and visibility.

8. State Manager

- Maintain desired state configuration (like enforcing antivirus installation or NTP settings).
- Applies configs continuously or at a schedule.

9. OpsCenter

- Central dashboard for operational issues (OpsItems).
- Integrated with CloudWatch and Config for diagnostics.

10. Change Manager

- Manage and approve infrastructure changes with built-in change control workflows.

Use Cases:

- Patch and maintain EC2 fleets at scale.
- Secure, auditable remote instance management.
- Store secrets and config outside of code.
- Automate AMI creation, backups, and compliance tasks.
- Replace bastion hosts and SSH with Session Manager.

Security & IAM:

- IAM policies control access to SSM documents, session access, and parameter values.
- All actions can be logged via CloudTrail.
- Parameter Store encryption via KMS.

Pricing:

- Core features are free (Run Command, Session Manager, Parameter Store - Standard tier).
- Advanced features like Parameter Store (Advanced), OpsCenter integrations, and Automation steps may incur cost.

Exam Tips:

- Use Session Manager instead of SSH for secure, auditable access.
- Use Run Command for quick multi-instance operations.
- Parameter Store stores secrets/configs; encrypt with KMS.
- Patch Manager automates patch compliance.
- Automation uses runbooks to automate tasks.
- SSM Agent must be installed and running on instances for SSM to work.

Quick Summary

AWS Systems Manager provides a single interface for managing infrastructure securely and at scale automating tasks, maintaining compliance, and avoiding the need for manual SSH or custom scripts.

AWS Trusted Advisor

What It Is

AWS Trusted Advisor is an online resource that provides real-time guidance to help you provision your AWS resources following best practices. It offers recommendations to optimize your AWS environment for cost, performance, security, fault tolerance, and service limits.

Key Features

- Scans your AWS account and provides actionable recommendations
- Covers five categories: Cost Optimization, Performance, Security, Fault Tolerance, and Service Limits
- Continuously evaluates AWS infrastructure against AWS best practices
- Recommendations are prioritized and actionable
- Dashboard provides a central view of checks and recommendations

Five Categories of Checks

- **Cost Optimization:** Identify idle and underutilized resources to reduce costs. Examples include underutilized EC2 instances, unattached EBS volumes, and idle load balancers.
- **Performance:** Improve the performance of AWS services. Includes checks for instance type optimization and usage of provisioned IOPS.
- **Security:** Strengthen security by identifying vulnerabilities. Examples include checks for MFA on root accounts, S3 bucket permissions, IAM password policies, and security group configurations.
- **Fault Tolerance**: Increase availability and redundancy. Checks include enabling Auto Scaling, monitoring EBS snapshots, and using multiple AZ deployments.
- **Service Limits**: Tracks usage against AWS service limits to avoid disruptions. Includes checks for EC2 instance limits, VPC limits, and others.

Available Checks

- Over 50 checks covering various AWS services
- Checks are regularly updated and improved by AWS
- Some checks are available to all customers, while full set requires Business or Enterprise Support Plan

Access Tiers

- Basic and Developer Support Plans: Access to seven core checks
- Business and Enterprise Support Plans: Access to all Trusted Advisor checks
- Core checks available to all customers include: Service Limits, Security Groups - Specific Ports Unrestricted, IAM Use, MFA on Root Account, EBS Public Snapshots, RDS Public Snapshots, and S3 Bucket Permissions

Dashboard

- Displays overall check status with color codes (green, yellow, red)
- Summarizes findings across all categories
- Links directly to AWS Console to act on recommendations
- Can export reports of recommendations

Notifications and Alerts

- Weekly email notifications with updated recommendations
- Alerts about changes in check status
- Helps maintain awareness of evolving best practices and environment changes

Programmatic Access

- Trusted Advisor API available for Business and Enterprise Support plans
- Enables automation and integration of recommendations into custom workflows
- Can retrieve results for specific checks or all checks programmatically

Integration with AWS Organizations

- AWS Organizations support allows consolidated view of Trusted Advisor recommendations across multiple accounts
- Helps central teams manage best practices compliance for entire organization

Security and Privacy

- Only analyses metadata and usage patterns, not customer data contents
- Recommendations are specific to your account and are only accessible by authorized IAM users

Pricing

- No additional cost for Basic access tier (includes seven core checks)
- Full access to all checks requires Business or Enterprise Support Plan
- Cost included in AWS Support plan pricing

Use Cases

- Cost savings by identifying idle or underused resources
- Improving security posture by enforcing best practices
- Avoiding service disruptions by monitoring service limits
- Ensuring highly available and fault-tolerant architectures
- Maintaining performance optimization across workloads

Best Practices

- Review Trusted Advisor dashboard regularly
- Act on high-priority recommendations promptly
- Automate remediation for common issues where possible
- Integrate Trusted Advisor with AWS Organizations for central visibility
- Use Trusted Advisor API to build custom monitoring and reporting solutions

Exam Tips

- Trusted Advisor provides real-time recommendations across five categories
- Full set of checks is available only with Business and Enterprise Support plans
- Core checks are available to all AWS customers
- Supports API access for automation
- Integrates with AWS Organizations for multi-account management
- Key categories to remember: Cost Optimization, Performance, Security, Fault Tolerance, Service Limits

Quick Summary

AWS Trusted Advisor is a best-practice assessment tool that provides real-time guidance to help optimize cost, improve performance, enhance security, ensure fault tolerance, and monitor service limits across your AWS environment. It is a critical tool for keeping AWS deployments efficient, secure, and compliant.

AWS Well-Architected Tool

What It Is

AWS Well-Architected Tool is a free service that helps you review and improve your cloud workloads using AWS best practices. It guides you through AWS's Well-Architected Framework, making it easier to identify risks and track improvements over time.

Key Features

- Provides a structured approach to reviewing workloads
- Helps evaluate workloads against AWS Well-Architected Framework best practices
- Supports consistent, repeatable reviews across teams
- Stores review history for tracking improvement progress
- Offers improvement plans for mitigating identified risks
- Accessible through the AWS Management Console

AWS Well-Architected Framework

- Foundation for the Tool's assessments
- **Consists of six pillars:**
    - **Operational Excellence**: Running and monitoring systems to deliver business value
    - **Security**: Protecting data, systems, and assets
    - **Reliability**: Ensuring workloads perform as intended and can recover quickly
    - **Performance Efficiency**: Using resources efficiently to meet requirements
    - **Cost Optimization**: Avoiding unnecessary costs
    - **Sustainability**: Minimizing environmental impacts of workloads
- Each pillar has design principles and best practices

Workload Reviews

- Define and document workloads to review
- Answer a set of best-practice questions across all six pillars
- Questions help identify high-risk and medium-risk issues
- Add custom lenses to tailor reviews to specific industry or organizational needs

Improvement Plans

- Tool generates prioritized improvement plans based on risks found
- Provides recommendations with direct links to AWS documentation
- Tracks completed and pending improvements
- Supports collaborative team review and remediation

Lenses

- Extend the framework with additional guidance for specific workload types or industries
- AWS provides lenses for Serverless, SaaS, Machine Learning, IoT, and more
- Custom lenses can be created for organization-specific standards
- Lenses add additional questions and best practices beyond the core framework

Integration with AWS Services

- AWS Organizations integration for sharing workload reviews across accounts
- AWS Service Catalog integration to ensure workloads meet Well-Architected standards before deployment
- Can use AWS IAM for fine-grained permissions to control access to workload reviews

Reporting and Tracking

- Stores all reviews and improvements in the AWS account
- Offers comparison over time to track progress
- Can download reports in PDF format for audits and internal reviews
- Highlights trends in risk status and completed improvements

Security and Access Control

- Uses AWS IAM for managing user permissions
- Supports resource-level permissions for controlling who can view or modify reviews
- All data is encrypted in transit and at rest using AWS encryption standards

Pricing

- Free to use within the AWS Management Console
- Improvement plans may recommend using other AWS services that have separate costs

Use Cases

- Conducting design and architecture reviews for workloads
- Identifying risks before production deployment
- Auditing existing workloads for compliance with best practices
- Standardizing architecture review processes across teams
- Supporting migration readiness assessments

Best Practices

- Review workloads early and often, especially before production deployment
- Involve cross-functional teams (developers, architects, security, operations) in reviews

- Use improvement plans to prioritize and track remediation
- Store review history for compliance and continuous improvement
- Leverage lenses to tailor reviews to workload types and industry needs

Exam Tips

- AWS Well-Architected Tool helps evaluate workloads against the Well-Architected Framework
- Supports six pillars, including the newer Sustainability pillar
- Identifies high-risk and medium-risk issues and generates improvement plans
- Offers custom lenses for industry-specific best practices
- Free to use in the AWS Console but may recommend services with additional costs
- Can integrate with AWS Organizations for multi-account management
- Ideal for ensuring workloads are secure, reliable, efficient, and cost-optimized

Quick Summary

AWS Well-Architected Tool enables consistent, structured reviews of workloads against AWS's best practices, helping teams identify and prioritize risks, implement improvements, and track architectural progress over time. It supports custom lenses, integrates with AWS Organizations, and provides reporting features for ongoing compliance and optimization.

AWS Elastic Transcoder

What It Is

AWS Elastic Transcoder is a fully managed media transcoding service that converts (transcodes) media files from their source formats into versions optimized for playback on various devices like smartphones, tablets, and web browsers. It handles all the heavy lifting of scaling, provisioning, and managing transcoding pipelines in the cloud.

Key Features

- Fully managed and scalable transcoding service
- Converts media files into formats suitable for multiple devices and platforms
- Supports popular formats like MP4, H.264, AAC, WebM, and MPEG-2
- Automatically scales based on the number of jobs and workloads
- Integrated with Amazon S3 for input and output file storage
- Delivers cost-effective transcoding through pay-as-you-go pricing

How It Works

1. You upload your source media file to an S3 bucket.
2. Elastic Transcoder fetches the file and processes it through a transcoding pipeline.
3. You specify presets or custom settings for output formats, resolutions, and bitrates.
4. The transcoded files are output to an S3 bucket, ready for delivery via CloudFront or direct download.

Pipelines

- A pipeline defines the workflow for transcoding: input, output, and notifications.
- Connects an input bucket (source media) and output bucket (transcoded files).
- Can send job status updates through Amazon SNS notifications.
- Each pipeline can process multiple transcoding jobs simultaneously.

Jobs

- A job represents a single transcoding request within a pipeline.
- Includes input file, output format(s), and optional thumbnails or captions.
- Multiple outputs (e.g., different resolutions or devices) can be defined per job.
- You can define encryption for both input and output content.

Presets

- Presets define the settings for transcoding (e.g., codec, resolution, bitrate, aspect ratio).
- AWS provides system presets for common devices (e.g., iPhone, Android, web).
- Custom presets can be created for fine-tuned control.
- Presets help standardize outputs and simplify job creation.

Thumbnails and Captions

- Elastic Transcoder can automatically generate thumbnail images during transcoding.
- Supports embedded or sidecar captions (e.g., WebVTT, SRT).
- Useful for video players that require previews or accessibility features.

Notifications

• Integrates with Amazon SNS to send updates on job status (e.g., progress, success, failure).
• Notifications can trigger Lambda functions, workflows, or external alert systems.

Security

• Uses IAM roles and policies to control access to pipelines and media files.
• Supports encryption for content in-transit and at-rest (using AWS KMS).
• Input and output buckets can have fine-grained access control.
• All requests to Elastic Transcoder use HTTPS for secure communication.

Integration with Other AWS Services

• **Amazon S3** – Stores input and output media files.
• **Amazon CloudFront** – Delivers transcoded content globally with low latency.
• **Amazon SNS** – Sends job status notifications.
• **AWS Lambda** – Automates post-processing workflows (e.g., metadata tagging, file cleanup).

Pricing

• Input and output S3 storage costs are billed separately.
• No upfront fees or minimum commitments.
• Different rates apply for standard-definition (SD), high-definition (HD), and audio-only outputs.

Use Cases

• Media streaming platforms needing device-optimized playback formats
• User-generated content applications that require automated video processing
• E-learning and marketing platforms delivering video content
• Transcoding pipelines for mobile and web distribution

Best Practices

• Use system presets for quick, reliable transcoding of common formats.
• Leverage CloudFront for content delivery to improve playback speed and reliability.
• Use IAM roles with least privilege access for pipelines and S3 buckets.
• Organize media files in S3 using structured folder hierarchies (input, output, logs).

Exam Tips

• Elastic Transcoder is used to convert (transcode) media files for playback on various devices.
• It's **fully managed** and works directly with **Amazon S3** for input/output.
• Transcoding is configured via **pipelines**, **jobs**, and **presets**.
• Notifications can be sent via **Amazon SNS**.
• It **automatically scales** and handles multiple simultaneous jobs.
• Elastic Transcoder is ideal for simpler, consumer-grade media transcoding tasks.

Quick Summary

AWS Elastic Transcoder is a scalable, managed service for converting media files into device-compatible formats. It integrates seamlessly with S3 and CloudFront, uses pipelines and presets for automation, and provides simple, cost-effective transcoding for streaming and mobile applications. Perfect for developers who want to deliver high-quality media without managing encoding infrastructure.

AWS Application Discovery Service

What It Is

AWS Application Discovery Service helps enterprises plan migration projects to AWS by automatically identifying on-premises servers, applications, and their dependencies. It collects detailed system configuration, performance, and usage data to provide insights that help with migration planning and cost estimation.

Key Features

- Automatically discovers on-premises servers and applications
- Collects system configuration, performance, and usage data
- Identifies application dependencies and network connections
- Provides detailed migration planning data to AWS Migration Hub
- Supports both agent-based and agentless discovery methods
- Enables accurate Total Cost of Ownership (TCO) and migration readiness analysis

Discovery Methods

1. Agent-Based Discovery

- Lightweight agent installed on each on-premises server
- Collects detailed data on CPU, memory, disk, and network usage
- Captures process-level and dependency information
- Ideal for complex workloads requiring deep visibility
- Data is encrypted and sent securely to AWS

2. Agentless Discovery (via AWS Agentless Collector)

- Deploys a virtual appliance (OVA) in VMware vCenter environments
- Discovers VM configurations, usage metrics, and network connections
- No need to install software on individual VMs
- Best suited for large-scale VMware environments

Data Collected

- Server details – hostnames, IPs, MAC addresses, OS, CPU, RAM, storage
- Performance metrics – CPU utilization, memory, disk I/O, network throughput
- Application and process information – running services and dependencies
- Network connections – inbound and outbound communication mapping
- Usage patterns and workload trends over time

Integration with AWS Migration Hub

- All collected data is stored and visualized in **AWS Migration Hub**
- Helps group servers into applications for migration planning
- Provides migration recommendations and status tracking
- Supports export of discovered data for further analysis

Security

- All communication between agents, collectors, and AWS is encrypted (HTTPS/TLS)
- IAM roles and policies control data access
- No data is shared with AWS regions unless explicitly configured
- Collected data can be deleted at any time via the console or API

Integration with Other AWS Services

- **AWS Migration Hub** – Central dashboard for discovery, migration tracking, and planning.
- **AWS Migration Evaluator** – Provides TCO analysis based on discovery data.
- **AWS Application Migration Service (MGN)** – For lift-and-shift server migrations.
- **AWS Database Migration Service (DMS)** – For database migration and modernization.

Pricing

- AWS Application Discovery Service is **free to use**.
- Standard AWS data transfer and storage charges may apply.
- Agents and collectors incur no additional software licensing costs.

Use Cases

- Assessing on-premises environments before migration
- Mapping application dependencies for multi-tier workloads
- Identifying underutilized resources for cost optimization
- Supporting rehosting and replatforming strategies
- Generating inventory and performance baselines

Best Practices

- Run discovery for at least 2–4 weeks to capture accurate workload patterns.
- Use agent-based discovery for in-depth performance data.
- Combine agent-based and agentless discovery for comprehensive visibility.
- Group discovered servers into logical applications before migration.
- Regularly export and back up discovery data from Migration Hub.
- Restrict IAM access to discovery data and use encryption for security.

Exam Tips

- Application Discovery Service identifies on-premises servers, applications, and dependencies for migration planning.
- **Agent-based discovery** gives deep performance and dependency details.
- **Agentless discovery** works with VMware environments for fast inventory.
- Integrates tightly with **AWS Migration Hub** for migration visualization.
- Helps build an accurate **TCO** and migration roadmap.
- Used during the **Assessment and Discovery** phase of migration not during actual migration.
- No additional cost, it's a free AWS service.

Quick Summary

AWS Application Discovery Service automates the detection and analysis of on-premises workloads to streamline migration planning. It provides insights into server performance, dependencies, and configurations, integrates with AWS Migration Hub, and enables data-driven migration strategies all without manual inventory or guesswork.

AWS Application Migration Service (MGN)

What It Is

AWS Application Migration Service (MGN) is a fully managed service that simplifies, expedites, and reduces the cost of migrating applications to AWS. It allows you to lift and shift (rehost) physical, virtual, or cloud-based servers to AWS with minimal downtime.

Key Features

- **Continuous Block-Level Replication:** Automatically replicates source servers to AWS in near real-time.
- **Minimal Downtime Cutover:** Enables testing and final cutover with minimal business disruption.
- **Automated Launch Templates:** Automatically generates EC2 launch templates for test and cutover instances.
- **Simplified Migration:** Reduces manual effort compared to traditional rehosting tools.
- **Supports a Variety of Environments:** Works with physical, VMware, Hyper-V, and other cloud sources.
- **Rollback Capability:** You can revert to the original environment if needed.
- **Integrated Conversion:** Converts your source servers to run natively on AWS infrastructure.

How It Works

1. **Install Agent:** Deploy the AWS replication agent on your source servers.
2. **Continuous Replication:** Data replicates continuously to a staging area subnet in your AWS account.
3. **Test Launch:** You can test migrated servers to validate functionality before cutover.
4. **Cutover Launch:** Perform a final migration (cutover) when ready.
5. **Post-Migration:** Optimize instances using AWS tools like AWS Systems Manager or Cost Explorer.

Core Components

- **Replication Agent:** Installed on each source server to replicate data.
- **Staging Area:** Temporary environment in AWS to store replicated data.
- **Launch Template:** Defines configuration for EC2 instances (type, VPC, subnet, etc.).
- **Test and Cutover Instances:** Created automatically for validation and final migration.

Supported Use Cases

- Migrating on-premises workloads to AWS with minimal changes.
- Moving workloads from other clouds to AWS.
- Modernizing legacy infrastructure through rehosting.

- Large-scale datacentre migration and disaster recovery setups.

Best Practices

- **Perform Test Launches:** Always test before final cutover.
- **Ensure IAM Permissions:** Grant correct permissions for the MGN service role.
- **Monitor Using CloudWatch:** Track replication lag and health status.
- **Tag Resources:** Helps with tracking and cost management.
- **Clean Up Post-Migration:** Remove staging resources after cutover to avoid extra costs.

Integration with Other AWS Services

- **AWS Identity and Access Management (IAM):** Controls access to MGN resources.
- **Amazon CloudWatch:** For performance and replication monitoring.
- **AWS Systems Manager:** Post-migration instance management.
- **AWS Migration Hub:** Centralized migration tracking and reporting.
- **AWS Cost Explorer:** Analyse and optimize costs after migration.

Pricing

- **Pay-as-you-go:** Based on the number of source servers actively replicating to AWS.
- No separate charge for test or cutover instances beyond EC2 and EBS usage.

Exam Tips

- MGN is the recommended service for lift-and-shift migrations to AWS.
- Unlike AWS SMS (Server Migration Service), MGN provides continuous replication instead of snapshot-based replication.
- Staging area subnet is required for data replication.
- You can migrate physical, virtual, or cloud-based servers.
- MGN automatically creates EC2 launch templates for test and cutover.
- Always test migration before final cutover to avoid application downtime.

Quick Summary

AWS Application Migration Service (MGN) is the go-to tool for seamless lift-and-shift migrations to AWS. It continuously replicates your source servers physical, virtual, or cloud-based into AWS, minimizing downtime and manual effort. With automated EC2 launch templates, near real-time replication, and test launch capabilities, MGN ensures a smooth transition before final cutover. It integrates tightly with IAM, CloudWatch, and Migration Hub, making it ideal for large-scale, low-risk migrations and modernization projects.

AWS Database Migration Service (DMS)

What It Is

AWS Database Migration Service (AWS DMS) is a managed service that helps you migrate databases to AWS easily and securely. It supports homogeneous migrations (e.g., Oracle to Oracle) and heterogeneous migrations (e.g., Oracle to Aurora). It also supports ongoing replication for minimal downtime migrations.

Key Features

- Supports one-time migration and continuous data replication
- Handles homogeneous and heterogeneous database migrations
- Supports both schema and data migration (with AWS Schema Conversion Tool)
- Minimal downtime with change data capture (CDC)
- Supports on-premises, EC2-based, and AWS cloud database sources and targets
- High availability with multi-AZ deployments
- Supports data transformation during migration
- Integrated monitoring with Amazon CloudWatch

Supported Sources and Targets

- **Sources**:
    - On-premises databases
    - AWS databases
    - EC2-hosted databases
- **Targets**:
    - Amazon RDS (all engines)
    - Amazon Aurora
    - Amazon Redshift
    - Amazon S3
    - DynamoDB
    - Kinesis Data Streams
    - OpenSearch Service
    - Any database supported by JDBC driver

Homogeneous vs. Heterogeneous Migrations

- **Homogeneous**: Source and target engines are the same (e.g., Oracle to Oracle)
- **Heterogeneous**: Source and target engines differ (e.g., Oracle to Aurora PostgreSQL)

- AWS Schema Conversion Tool (AWS SCT) helps convert database schema and code for heterogeneous migrations

Replication Tasks

- Full load: Migrates all existing data
- CDC (Change Data Capture): Captures ongoing changes to keep source and target in sync
- Full load + CDC: Combines both for minimal downtime migrations

Change Data Capture (CDC)

- Enables near-real-time replication
- Captures insert, update, and delete operations from the source
- Ensures target stays up-to-date during and after initial migration

High Availability

- Multi-AZ deployment with failover support
- Ensures minimal disruption in case of infrastructure failure
- Replication tasks can be resumed after failover

Data Transformation

- Supports basic data transformation rules during migration
- Modify data types, column names, or content as it moves to the target
- Can map and filter data to control what is migrated

Performance and Scalability

- Scales with instance types to support large datasets
- Supports parallel table loading
- Optimized for high-throughput migration

Monitoring and Management

- Integrated with Amazon CloudWatch for metrics and logs
- AWS Management Console for task creation and monitoring
- Detailed task logs for troubleshooting
- Notifications via Amazon SNS

Security

- Supports encryption at rest and in transit
- VPC support for network isolation
- IAM for controlling access to AWS DMS resources
- AWS Key Management Service (KMS) for managing encryption keys

AWS Schema Conversion Tool (SCT)

- Complements AWS DMS for heterogeneous migrations
- Converts database schema, stored procedures, views, and other code objects
- Provides assessment reports to identify manual changes required
- Supports conversion to Amazon Aurora, PostgreSQL, MySQL, and others

Pricing

- Pay only for replication instance hours used
- Storage for migration logs and cached changes is billed separately
- No charge for data transfer between AWS DMS and AWS databases in the same region

Use Cases

- Migrating production databases to AWS with minimal downtime
- Continuous replication for disaster recovery or high availability setups
- Consolidating multiple databases into a single AWS target
- Moving analytical workloads to Amazon Redshift
- Archiving transactional data to Amazon S3

Best Practices

- Use AWS SCT for schema conversion in heterogeneous migrations
- Test migration tasks thoroughly before production cutover
- Monitor tasks using CloudWatch metrics and logs
- Enable multi-AZ deployments for high availability
- Plan for network connectivity and security (VPC, IAM policies)
- Validate data integrity post-migration

Exam Tips

- Supports homogeneous and heterogeneous migrations
- AWS SCT is required for schema conversion in heterogeneous migrations
- Supports sources and targets on-premises, in EC2, or AWS-managed databases
- Integrated monitoring with CloudWatch and notifications with SNS
- Encryption in transit and at rest supported via AWS KMS

Quick Summary

AWS DMS simplifies and automates database migrations to AWS with minimal downtime. It supports both same-engine and cross-engine migrations, continuous replication via CDC, and works with AWS SCT for schema conversion. With built-in monitoring, security, and multi-AZ support, it ensures reliable, secure, and low-disruption database migrations.

AWS DataSync

What It Is

AWS DataSync is a fully managed data transfer service that simplifies, automates, and accelerates moving large amounts of data between on-premises storage and AWS services.
It is designed for fast, secure, and automated data movement, supporting one-time migrations and ongoing transfers.

Key Features

- Transfers data up to 10x faster than open-source tools.
- Automates tasks like data validation, scheduling, monitoring, and encryption.
- Built-in integrity verification to ensure data consistency.
- Handles metadata and ACL preservation for files.
- No need to write custom transfer scripts or manage infrastructure.

Supported Transfer Locations

- **On-premises storage**:
    - NFS (Network File System)
    - SMB (Server Message Block)
- **AWS services**:
    - Amazon S3
    - Amazon EFS (Elastic File System)
    - Amazon FSx for Windows File Server
    - Amazon FSx for Lustre
    - AWS Snowcone (via DataSync agent)
    - Amazon S3-compatible storage

How It Works

- Deploy **AWS DataSync Agent** on-premises:
    - Virtual appliance for VMware, Hyper-V, or EC2.
    - Connects to on-premises NFS or SMB shares.
- Configure source and destination locations.
- Define and start tasks:
    - Transfer data from on-prem to AWS, AWS to on-prem, or between AWS services.
- Supports scheduled and incremental transfers.
- Transfers securely over AWS Direct Connect, VPN, or the Internet.

Performance

- Purpose-built protocol optimized for high-speed transfers.
- Can use parallel, multi-threaded operations.
- Transfers hundreds of terabytes or millions of files quickly.
- Supports throttling to avoid saturating network links.

Security

- Data encrypted in transit using TLS.
- Data at rest uses AWS-managed encryption:
    - S3 SSE (including SSE-KMS).
    - EFS and FSx encryption features.
- Integrates with AWS IAM for access control.
- Supports VPC endpoints for private connectivity.

Monitoring & Management

- AWS Management Console and CLI support for configuration and monitoring.
- AWS CloudWatch:
    - Monitor task metrics and performance.
    - Receive alarms on failures or thresholds.
- AWS CloudTrail:
    - Logs API calls for auditing.

Pricing

- Charged based on amount of data copied (per GB transferred).
- Additional AWS service costs apply for storing transferred data (e.g., S3 storage fees).

Typical Use Cases

- **Data migration**:
    - On-premises storage to AWS for cloud adoption.
- **Ongoing replication**:
    - Backup and DR strategies with continuous sync.
- **Hybrid cloud workflows**:
    - Sync data between on-premises and AWS.
- **Data movement between AWS services**:
    - E.g., S3 to EFS, FSx to S3.

Integration with Other AWS Services

- Transfer objects to/from S3 buckets.
- Migrate or replicate POSIX file systems.
- Supports Windows File Server and Lustre.
- Transfers data on/off Snowcone devices.
- AWS CloudWatch and CloudTrail for monitoring and auditing.

Data Validation and Integrity

- Automatic checksum verification during transfers.
- Ensures transferred data matches source data.
- Logs transfer results for audit and compliance.

Performance Tuning

- Supports configurable bandwidth limits.
- Can schedule transfers during off-peak hours.
- Enables efficient incremental transfers by only copying changed data.

Comparison with Other Services

Service	Use Case	Key Difference
AWS DataSync	Fast, automated bulk file transfer	Designed for NFS/SMB, optimized for speed
AWS Snowball	Large-scale, offline data migration	Physical device, TB–PB scale
AWS Transfer Family	Managed SFTP/FTP access to S3/EFS	Protocol-based user transfers
AWS Storage Gateway	Hybrid, on-prem to cloud storage integration	Mount points for files/volumes/tapes

Exam Tips

- AWS DataSync = automated, high-speed data transfer between on-premises and AWS.
- Supports NFS, SMB, S3, EFS, FSx, Snowcone.
- Requires DataSync Agent for on-premises transfers.
- Uses TLS encryption in transit.
- IAM controls for access permissions.
- Automatic integrity checks with checksums.

Quick Summary

AWS DataSync is a fully managed service for accelerated, secure, and automated data movement between on-premises storage systems and AWS services. It supports large-scale, repeatable transfers with monitoring, encryption, and integrity checks built in.

AWS Snowmobile

What It Is

AWS Snowmobile is an exabyte-scale data transfer service that helps move massive amounts of data (up to 100 PB) into AWS.
It uses a secure, physical shipping container (a 45-foot ruggedized truck) to transport data securely from your data centre to AWS.

Key Features

- **Capacity**
    - Each Snowmobile can hold up to 100 petabytes of data.
    - Multiple Snowmobiles can be used for exabyte-scale transfers.
- **Use Case**
    - Best suited for large-scale data migrations when moving data over the network is impractical due to bandwidth constraints or time limitations.
    - Typical scenarios:
        - Data centre decommissioning
        - Massive archive migrations
        - Disaster recovery data transfer
- **Physical Security**
    - Snowmobile is a tractor-trailer shipping container.
    - Equipped with:
        - GPS tracking
        - 24/7 video surveillance
        - Alarms and tamper-proof mechanisms
    - An accompanies it during AWS dedicated security vehicle escort transit.
- **Data Security**
    - All data is automatically encrypted using 256-bit encryption keys managed through AWS Key Management Service (KMS).
    - Encryption keys are never stored on the Snowmobile.
    - Digital signatures verify data integrity during transfer.
- **Data Transfer**
    - AWS personnel connect the Snowmobile to your local network via high-speed network connections.
    - Data transfer rates can reach up to 1 Tbps depending on your infrastructure.
    - Data is copied to the Snowmobile's on-board storage arrays.

- **Integration**
    - Once the Snowmobile arrives at the AWS data centre, AWS personnel unload the data into your specified S3 bucket(s).
- **Ordering**
    - The service must be requested through the AWS account team.
    - AWS conducts an assessment and planning process before scheduling delivery.

Comparison with Other Snow Services

Service	Capacity	Use Case	Delivery Time
Snowball Edge	Up to 80 TB per device	TB-scale migrations, edge computing workloads	Days to weeks
Snowmobile	Up to 100 PB	Exabyte-scale migrations	Weeks (depending on size)

Security Considerations

- Encryption in transit and at rest (AES-256).
- KMS integration for key management.
- Chain-of-custody tracking during entire operation.
- Tamper-evident enclosures and escort security.

Limitations

- Only available in specific AWS regions (must confirm availability).
- Requires significant logistical planning.
- Typically used for one-time, large-scale migrations.

Exam Tips

- Snowmobile = 100 PB truck used when moving data over network is infeasible.
- Data encrypted using KMS-managed 256-bit keys.
- Requires on-site setup by AWS and security escort.
- Often tested in exam scenarios describing massive data centre migrations.
- Compared to Snowball Edge, Snowmobile is much larger and designed for exabyte-scale transfer.

Quick Summary

AWS Snowmobile is a physically transported data transfer solution capable of moving up to 100 PB of data per unit securely and efficiently from customer sites to AWS, with strong encryption, chain-of-custody controls, and logistical support from AWS.

AWS Snowball Edge

What It Is

AWS Snowball Edge is a rugged, portable data transfer and edge computing device designed to move TB-scale data to and from AWS, and to run select AWS compute services locally.

Key Features

Storage and Capacity

- Two device types:
    - **Snowball Edge Storage Optimized**
        - 80 TB usable storage.
        - Best for large-scale data transfer, local storage.
    - **Snowball Edge Compute Optimized**
        - 42 TB usable storage.
        - Includes more vCPUs and GPU options for edge computing workloads.

Edge Computing Capabilities

- Can run AWS Lambda functions locally.
- Supports EC2 instances for local compute processing.
- Enables processing, filtering, and transforming data before transfer.
- Helps support workloads in disconnected or intermittent-network environments.

Use Cases

- Data migration to AWS (TB-scale).
- Edge computing in remote or mobile environments.
- IoT data processing at the edge.
- Disaster recovery and business continuity.
- Military or rugged field operations with limited connectivity.

Data Transfer

- Secure, offline transfer to AWS:
    - Customer loads data onto Snowball Edge on-premises.
    - Device is shipped back to AWS.
    - AWS uploads data into the customer's S3 bucket.
- Also supports online data transfer between devices in a local cluster.
- Supports data import and export.

Clustering

- Multiple Snowball Edge devices can be clustered for:
    - Increased local storage capacity.
    - High-availability deployments.
- Clusters can act as a local HDFS-compatible storage layer.

Security

- End-to-end 256-bit encryption.
- AWS KMS integration for managing encryption keys.
- Tamper-resistant enclosures with tamper-evident seals.
- Trusted Platform Module (TPM) for hardware-based key storage.
- Chain-of-custody tracking during transit.
- Data automatically wiped after successful transfer to AWS.

Ordering and Management

- Ordered via AWS Management Console.
- Managed with the Snow Family Management Console or CLI.
- Tracks shipping, jobs, and device status.
- Snowball Edge devices arrive pre-configured for the customer's job.

Networking

- Supports 10 Gb, 25 Gb, and 40 Gb network connections.
- Local interfaces allow for fast on-premises transfer.
- Can be used in edge locations with limited or no internet connectivity.

Snowball Edge Variants

Type	Storage Capacity	Use Case	Compute Capabilities
Storage Optimized	80 TB usable	Bulk data transfer, local storage	Basic EC2 and Lambda support
Compute Optimized	42 TB usable	Edge compute-heavy workloads, ML/AI	Extra vCPUs, optional GPU (NVIDIA Tesla V100)

Comparison with Snowmobile

Feature	Snowball Edge	Snowmobile
Capacity	42–80 TB per device	Up to 100 PB per truck
Use Case	TB-scale transfer, edge computing	Exabyte-scale data centre migration
Delivery	Shipped as a rugged appliance	Delivered as a secure 45-foot truck
Compute Capabilities	Runs EC2, Lambda locally	No compute capability, purely data transfer

Exam Tips

- Snowball Edge is for TB-scale transfer with edge computing.
- Choose Storage Optimized for bulk data movement.
- Choose Compute Optimized for local processing with EC2 instances and GPU workloads.
- Data is encrypted end-to-end with AWS KMS keys.
- Device automatically wiped after ingestion.
- Can be clustered for higher local storage and HA.
- Common exam scenario: Remote site or disconnected edge environment needing local processing and transfer to AWS.

Quick Summary

AWS Snowball Edge is a portable, rugged device for securely transferring TB-scale data and running edge computing workloads locally. It supports offline data movement to AWS with encryption, built-in compute resources, and clustering for local storage and processing in harsh or remote environments.

AWS Transfer Family

What It Is

AWS Transfer Family is a fully managed service that enables you to transfer files directly into and out of Amazon S3 or Amazon EFS using SFTP, FTPS, and FTP protocols.
It provides a secure, highly available, and scalable way to integrate traditional file transfer workflows with AWS storage services.

Supported Protocols

- SFTP (Secure File Transfer Protocol)
- FTPS (FTP over SSL/TLS)
- FTP (unencrypted; generally used only if required by legacy systems)

Key Features

- Fully managed service, no need to manage your own FTP servers.
- Supports direct integration with:
    - Amazon S3
    - Amazon Elastic File System (EFS)
- Scales automatically to handle thousands of concurrent connections.
- Supports custom domain names via AWS Certificate Manager (ACM).
- High availability with AWS-managed infrastructure.
- User authentication options:
    - Service-managed identities
    - AWS Directory Service (Active Directory)
    - Custom identity providers via API Gateway and Lambda.

Storage Integration

- **Amazon S3**:
    - Users' files land in specified S3 buckets.
    - Supports S3 access points for fine-grained permissions.
- **Amazon EFS**:
    - Mount EFS file systems as the backend for FTP servers.
    - Allows POSIX-compliant shared access.

Security

- Data in transit:
    - SFTP and FTPS ensure encryption over the wire.
    - FTP is available for compatibility but is unencrypted.

- Data at rest:
    - Protected via S3/EFS encryption.
    - Can leverage AWS Key Management Service (KMS) for managing encryption keys.
- IAM policies control user permissions and access to backend storage.
- Supports logging with AWS CloudTrail.
- Supports VPC Security Groups and VPC endpoints for private access.

User Management

- **Service-managed users:**
    - AWS stores user credentials and configuration.
- **Directory Service:**
    - Integrates with Microsoft AD for authentication.
- **Custom Identity Provider:**
    - Use API Gateway and Lambda to authenticate users against external systems.
- User permissions define which folders/buckets/files users can access.

Availability & Scalability

- High availability by design in multiple Availability Zones.
- Automatic scaling based on number of connections and workloads.
- No infrastructure to provision or maintain.

Monitoring & Logging

- **AWS CloudWatch:**
    - Monitor server activity.
    - Track metrics like connections and data transfer.
- **AWS CloudTrail:**
    - Logs API calls and configuration changes for auditing.

Typical Use Cases

- Replacing legacy FTP/SFTP servers with a managed service.
- Exchanging files securely with partners, vendors, or customers.
- Onboarding data feeds into data lakes stored in S3.
- Providing SFTP/FTPS access to EFS-based applications

Pricing

- **Based on:**
    - Endpoint hours.
    - Data uploaded/downloaded.
    - Additional costs for S3 or EFS storage used.

Comparison with AWS Storage Gateway

Feature	AWS Transfer Family	AWS Storage Gateway
Purpose	Protocol-based file transfer to/from AWS	Hybrid on-prem to cloud storage integration
Protocols	SFTP, FTPS, FTP	NFS, SMB, iSCSI
Storage Back-end	Amazon S3, Amazon EFS	S3, EBS Snapshots, Glacier
Use Cases	Partner data exchange, legacy FTP replacement	Backup, disaster recovery, hybrid cloud

Exam Tips

- AWS Transfer Family provides SFTP/FTPS/FTP access to S3 and EFS.
- Common exam scenario: Replacing a legacy FTP server with secure, scalable, managed AWS service.
- Supports custom authentication via API Gateway and Lambda.
- Integrates directly with S3 access points and EFS file systems.
- Supports VPC endpoints for private access.
- IAM policies and user management control permissions.
- Logs all API activity in CloudTrail.
- Encrypts data in transit (SFTP/FTPS) and at rest (via S3/EFS encryption).

Quick Summary

AWS Transfer Family provides managed, secure, scalable SFTP, FTPS, and FTP access to Amazon S3 and EFS, enabling easy modernization of legacy file transfer workflows without maintaining your own servers.

Amazon CloudFront

What It Is

Amazon CloudFront is a global Content Delivery Network (CDN) that uses a worldwide network of edge locations to cache and deliver content closer to users, dramatically reducing latency and improving performance for static and dynamic web content, videos, APIs, and applications.

Core Concepts

Edge Locations

- Global network of caching servers (hundreds of locations worldwide)
- Cache copies of content close to end users for faster delivery
- Serve content from the nearest location to minimize latency

Regional Edge Caches

- Intermediate caching layer between origin and edge locations
- Cache larger objects with lower request volumes
- Reduce load on origin servers by serving cached content to edge locations

Origins

- Source of content that CloudFront retrieves and caches
- Supported origin types:
    - Amazon S3 buckets (static site hosting, media files)
    - EC2 instances (application servers)
    - Elastic Load Balancers (ALB, NLB)
    - Custom HTTP/HTTPS endpoints (on-premises servers, third-party origins)

Distributions

- Configuration defining how CloudFront delivers content
- Types:
    - **Web Distribution** – Static and dynamic content (HTML, CSS, JavaScript, images, videos, APIs)
    - **RTMP Distribution** – Deprecated (Adobe Flash streaming media)

Caching and TTL

Cache-Control Headers

- Control **how long objects remain** in edge caches
- Origin servers specify caching behaviour via HTTP headers

Time-to-Live (TTL)

- **Minimum TTL** – Shortest time content stays cached

- **Default TTL** – Standard caching duration (typically 24 hours)
- **Maximum TTL** – Longest time content can remain cached
- Configurable per distribution or per object

Cache Invalidation

- Manually remove objects from cache before TTL expires
- Force immediate updates for changed content
- Supports wildcard patterns for batch invalidations
- Can be automated via API or console

Security Features

SSL/TLS Encryption

- AWS Certificate Manager (ACM) integration for custom SSL/TLS certificates at no extra cost
- HTTPS enforcement between viewers and CloudFront
- HTTPS support between CloudFront and origin servers
- Supports SNI (Server Name Indication) and dedicated IP options

Origin Access Control (OAC)

- Recommended method for securing S3 origins
- Prevents direct S3 bucket access, forcing traffic through CloudFront
- Successor to Origin Access Identity (OAI) with enhanced security
- Uses AWS Signature Version 4 for authentication

Signed URLs and Signed Cookies

- Restrict access to premium or private content
- Control who can access content and for how long
- **Signed URLs –** Single file access (video files, downloads)
- **Signed Cookies –** Multiple file access (entire sections of a website)

Field-Level Encryption

- Encrypt sensitive data (credit card numbers, personal information) at the edge before forwarding to origin
- Data remains encrypted throughout processing
- Only origin application can decrypt with private keys

Geo Restriction

- Restrict access based on viewer's geographic location

- Allowlist (whitelist) specific countries
- Blocklist (blacklist) specific countries
- Useful for licensing requirements and compliance

AWS WAF Integration

- Protect applications from common web exploits (SQL injection, XSS)
- Create custom rules to filter malicious traffic
- Rate limiting to prevent abuse

Shield Standard

- Included DDoS protection at no extra cost
- Protects against network and transport layer attacks
- Automatic detection and mitigation

Content Customization

Lambda@Edge

- Run custom code closer to users at edge locations
- Modify requests and responses in real-time
- Use cases:
    - URL rewrites and redirects
    - Header manipulation
    - A/B testing
    - User authentication
    - Content personalization
- Supports Node.js and Python
- More powerful but higher cost and longer latency than CloudFront Functions

CloudFront Functions

- Lightweight JavaScript execution at the edge
- Designed for high-volume, simple manipulations
- Faster (sub-millisecond) and cheaper than Lambda@Edge
- Use cases:
    - Header manipulation
    - URL rewrites
    - Request validation
    - Cache key normalization

Origin Types

- Amazon S3 – Static content storage → Ideal for static websites, images, videos, and downloads
- EC2 Instances – Application servers → Used for dynamic content and APIs
- Elastic Load Balancer (ELB) – Distributes traffic across instances → Ensures high availability and scalable applications
- Custom HTTP/HTTPS – Any web server → Suitable for on-premises servers or third-party origins

Logging and Monitoring

Standard Logs

- Write detailed logs to S3 bucket
- Includes request details (IP address, URI, user agent, referrer)
- Useful for analytics and troubleshooting

Real-Time Logs

- Stream logs to Kinesis Data Streams in near real-time
- Enable immediate analysis and alerting

CloudWatch Metrics

- Built-in metrics for monitoring performance
- Includes:
    - Total requests
    - Cache hit/miss ratios
    - Error rates (4xx, 5xx)
    - Bytes downloaded/uploaded

AWS CloudTrail

- Logs API calls for governance, compliance, and auditing
- Track configuration changes and access patterns

Price Classes

- **Price Class 100** – Covers US, Europe, and Canada → Cheapest option with fewer edge locations
- **Price Class 200** – Covers most regions (excludes the most expensive ones) → Balanced cost and coverage
- **Price Class All** – Covers all edge locations worldwide → Best performance, but highest cost

Compression

- Automatic compression of certain file types (JavaScript, CSS, HTML)
- Reduces object size delivered to viewers
- Improves transfer speeds and reduces bandwidth costs
- Must be enabled in distribution settings

Origin Groups

- Support failover scenarios for high availability
- Configure primary and secondary origins
- If primary origin fails (returns 5xx errors), CloudFront automatically switches to secondary
- Useful for disaster recovery and redundancy

Integration with Other AWS Services

- **Amazon S3** – Static site hosting, media storage
- **AWS WAF** – Application protection, custom rules
- **AWS Shield** – DDoS mitigation
- **Lambda@Edge** and **CloudFront Functions** – Request customization
- **AWS Certificate Manager (ACM)** – Free SSL/TLS certificates
- **Route 53** – DNS routing to CloudFront distributions
- **AWS Global Accelerator** – Optimized routing for dynamic content
- **Amazon CloudWatch** – Monitoring and alerting
- **AWS CloudTrail** – Audit logging

Exam Tips

- CloudFront = Global CDN caching content at edge locations for low latency
- Origin Access Control (OAC) secures S3 origins by preventing direct access
- Signed URLs/Cookies restrict access to premium or private content
- Field-Level Encryption protects sensitive data at the edge
- Lambda@Edge for complex customization; CloudFront Functions for simple, high-volume tasks
- Geo Restriction controls access by geographic location

Quick Summary

Amazon CloudFront is AWS's global Content Delivery Network (CDN) that improves performance and security by caching static and dynamic content at edge locations worldwide. It integrates deeply with AWS security services (WAF, Shield, ACM), offers advanced customization with Lambda@Edge and CloudFront Functions.

AWS Direct Connect

What It Is

AWS Direct Connect is a cloud service solution that establishes a dedicated, private network connection between your on-premises data centre or office and AWS. It reduces network costs, increases bandwidth throughput, and provides a more consistent network experience compared to internet-based connections.

Key Features

- Provides dedicated network connectivity from on-premises to AWS
- Reduces network variability and improves performance with low latency
- Supports high-bandwidth workloads by offering port speeds from 50 Mbps up to 100 Gbps
- Traffic does not traverse the public internet, enhancing security
- Integrated with AWS Virtual Private Cloud (VPC) via Virtual Interfaces (VIFs)

Connection Types

- **Dedicated Connections**
    - Provisioned at 1, 10, or 100 Gbps
    - Ordered via AWS Management Console
    - Delivered at AWS Direct Connect locations
- **Hosted Connections**
    - Provisioned by AWS Direct Connect Partners
    - Available from 50 Mbps to 10 Gbps
    - Useful for faster provisioning without needing to manage physical hardware

Virtual Interfaces (VIFs)

- **Private VIF**
    - Connects to VPC via Virtual Private Gateway or Transit Gateway
    - Used for accessing private AWS resources
- **Public VIF**
    - Provides access to all AWS public services using AWS public IP addresses
    - Supports access to services like S3, DynamoDB, or AWS public endpoints
- **Transit VIF**
    - Connects to AWS Transit Gateway for managing multiple VPC connections
    - Supports scalable, hub-and-spoke architectures

Link Aggregation Groups (LAG)

- Combine multiple connections (same speed) into a single managed connection
- Supports increased bandwidth and redundancy
- Up to 4 connections in a single LAG
- Managed as a single interface

Redundancy and Resiliency

- AWS recommends establishing redundant connections for high availability
- Multiple Direct Connect locations can be used for failover
- Integrated with AWS VPN for hybrid architectures that need failover over the public internet
- AWS SLA applies when using recommended redundant configurations

Direct Connect Gateway

- Enables connections to multiple VPCs across AWS Regions (except China)
- Supports global access using a single Direct Connect connection
- Simplifies network management in multi-Region, multi-VPC environments

Billing and Pricing

- Port-hour charges based on speed of connection
- Data transfer out charges, typically lower than internet-based data transfer
- No charge for inbound data transfer
- LAG billed as aggregated port hours

Use Cases

- High-volume data transfers such as backups, migrations, and analytics workloads
- Hybrid cloud deployments requiring consistent and secure connectivity
- Low-latency applications such as financial trading, gaming, or media
- Private access to AWS resources without traversing the internet
- Multi-Region VPC connectivity using Direct Connect Gateway

Security

- Traffic does not traverse the public internet, reducing exposure to attacks
- Supports MACsec (Media Access Control Security) for encryption on supported ports
- Can integrate with AWS VPN for encrypted connections over public networks
- Works with IAM for permission management and auditing

Integration with Other AWS Services

- AWS VPC for private connectivity
- AWS Transit Gateway for large-scale network hub-and-spoke architectures
- AWS Direct Connect Gateway for cross-region access
- AWS VPN for hybrid and backup connections
- AWS CloudWatch for monitoring metrics such as connection state and data transfer

Exam Tips

- Direct Connect provides private, dedicated connectivity and avoids the internet
- Private VIF connects to VPC for private resources; Public VIF connects to AWS public endpoints
- Direct Connect Gateway allows cross-region VPC connections
- Recommended to use redundant connections for high availability and SLA compliance
- Can combine Direct Connect with VPN for secure, resilient hybrid architectures
- LAG enables combining multiple connections for higher bandwidth and redundancy
- Cheaper outbound data transfer costs compared to internet routes

Quick Summary

AWS Direct Connect provides dedicated, secure, and consistent network connections between your on-premises infrastructure and AWS. It supports high-bandwidth workloads, reduces latency, avoids the public internet, and integrates with AWS services like VPC, Transit Gateway, and Direct Connect Gateway for flexible, scalable hybrid cloud architectures.

Amazon Elastic Load Balancer (ELB)

What It Is

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets such as EC2 instances, containers, IP addresses, and Lambda functions. It improves availability, fault tolerance, and scalability. ELB is fully managed and supports automatic scaling and high availability across Availability Zones.

Types of Load Balancers

1. **Classic Load Balancer (CLB)**
    - Legacy option (EC2-Classic only, but still supported in VPC)
    - Operates at Layer 4 (TCP) and Layer 7 (HTTP/HTTPS)
    - Basic routing and health checks
    - Limited features compared to modern alternatives
    - Not recommended for new architectures
2. **Application Load Balancer (ALB)**
    - Operates at Layer 7 (HTTP/HTTPS)
    - Advanced request routing (host-based, path-based routing)
    - Supports containerized applications with ECS
    - Can route to multiple target groups
    - Supports WebSocket and HTTP/2
    - Integrated with AWS WAF for protection
    - Supports user authentication (OIDC, Cognito)
3. **Network Load Balancer (NLB)**
    - Operates at Layer 4 (TCP, TLS, UDP)
    - Extremely high performance (millions of requests per second)
    - Static IP support and Elastic IPs
    - Preserves source IP for backend
    - Ideal for low latency and high throughput
    - TLS termination available
    - Supports IP addresses and AWS PrivateLink
4. **Gateway Load Balancer (GWLB)**
    - Operates at Layer 3
    - Used to deploy, scale, and manage virtual appliances (firewalls, IDS/IPS)

- Uses GENEVE protocol
- Simplifies insertion of security appliances into traffic path
- Integrates with third-party virtual appliances

Listeners and Rules

- Listeners check for connection requests on configured protocols and ports.
- ALB supports listener rules to forward traffic based on host or path conditions.
- NLB listeners forward TCP, TLS, or UDP traffic directly to targets.
- GWLB listeners forward traffic using the GENEVE protocol.

Target Groups

- Logical grouping of targets (EC2 instances, IP addresses, Lambda functions).
- Health checks performed at target group level.
- Supports weighted target groups for traffic distribution.
- ALB target types include Instance, IP, Lambda.
- NLB target types include Instance, IP.

Health Checks

- ELB performs health checks to determine target availability.
- Configurable intervals, thresholds, paths, and protocols.
- Unhealthy targets are automatically removed from routing.
- Health check types: HTTP, HTTPS, TCP, or gRPC (for ALB).

Cross-Zone Load Balancing

- Distributes traffic evenly across all healthy targets in all enabled Availability Zones.
- Enabled by default on ALB and CLB.
- Optional on NLB (can be enabled or disabled).
- Helps improve availability and fault tolerance.

SSL/TLS Termination

- Offload encryption/decryption to the load balancer.
- Certificates managed via AWS Certificate Manager (ACM).
- ALB and NLB support TLS termination.
- Helps simplify backend application configuration.

Sticky Sessions

- Also known as session affinity.
- ALB and CLB support sticky sessions via cookies.

- Enables user sessions to stay on the same backend for duration of session.
- Useful for stateful applications.

Security Features

- Integrated with AWS Certificate Manager for SSL/TLS.
- Supports Security Groups (for ALB and CLB).
- AWS WAF integration with ALB for Layer 7 protection.
- Access logs to S3 for audit and analysis.
- IAM policies control API-level access.

Logging and Monitoring

- Access logs can be sent to S3.
- CloudWatch metrics include request count, latency, healthy/unhealthy host count.
- CloudTrail records API calls.
- ALB supports detailed request tracing with X-Amzn-Trace-Id header.

Integration with Other AWS Services

- Works with Auto Scaling Groups for dynamic scaling of targets.
- Integrated with ECS for containerized applications.
- Supports Lambda as a target (for ALB).
- Integrated with AWS Global Accelerator for improved global performance.
- Can be fronted by Route 53 for DNS-based routing.

Pricing

- Charged based on hours the load balancer is running and the amount of data processed.
- Includes charges for LCU (Load Balancer Capacity Units) for ALB and NLB.
- GWLB pricing is based on processed data.

Use Cases

- **ALB**: Microservices architectures, advanced HTTP routing, containerized apps.
- **NLB**: Low-latency TCP/UDP workloads, real-time gaming, IoT, financial apps.
- **GWLB**: Deploying security appliances transparently.
- **CLB**: Legacy workloads needing simple Layer 4/7 routing.

Exam Tips

- ALB is best for Layer 7 advanced routing, path-based and host-based rules.
- NLB is optimized for high-throughput, low-latency Layer 4 connections.
- GWLB is for deploying and managing virtual appliances at Layer 3.

- ALB supports Lambda as a target.
- Sticky sessions are supported on ALB and CLB.
- Cross-Zone Load Balancing is on by default for ALB and CLB.
- Use ACM to manage SSL/TLS certificates.
- Health checks are critical for removing unhealthy targets automatically.
- Access logs help with auditing and analysis.
- Security Groups apply to ALB and CLB, not NLB.
- Use AWS WAF with ALB for application-layer protection.

Quick Summary

AWS Elastic Load Balancing offers fully managed load balancing across EC2, containers, IP addresses, and Lambda functions. It supports multiple types of load balancers tailored to different use cases: ALB for Layer 7 advanced routing, NLB for ultra-low latency Layer 4 traffic, GWLB for security appliance deployment, and CLB for legacy applications. It integrates with AWS services like Auto Scaling, ECS, ACM, and WAF to build scalable, secure, and highly available architectures.

AWS Global Accelerator

What It Is

AWS Global Accelerator is a networking service that improves the availability and performance of your global applications.
It uses the AWS global network to direct traffic to optimal endpoints, improving latency and resilience.

Key Features

- Provides static IP addresses that act as a fixed entry point to your applications.
- Uses the AWS global network to route traffic to optimal endpoints.
- Supports both TCP and UDP traffic.
- Health checks and automatic failover.
- Improves performance for global users by routing through AWS edge locations.

How It Works

1. Clients connect using static IP addresses provided by Global Accelerator.
2. Traffic enters AWS's global edge network.
3. Routed over AWS's private backbone to the best endpoint in the AWS region.
4. Health checks ensure only healthy endpoints are used.
5. Automatic rerouting during endpoint failures or health check failures.

Static IP Addresses

- Each accelerator provides two static IP addresses (for high availability).
- Can also bring your own IP addresses (BYOIP).

Components

- **Accelerator**:
    - The top-level resource.
    - Includes one or more listeners.
- **Listener**:
    - Defines the port and protocol (TCP/UDP) for incoming traffic.
    - Can have one or more endpoint groups.
- **Endpoint Group**:
    - Tied to a specific AWS Region.
    - Contains one or more endpoints.
    - Has a traffic dial to control traffic distribution to the region.
    - Supports health checks for endpoints.

- **Endpoint**:
    - Where traffic is ultimately delivered.
    - Can be:
        - Elastic IP addresses
        - Network Load Balancers (NLB)
        - Application Load Balancers (ALB)
        - EC2 instances
        - Elastic IP addresses
        - AWS Global Accelerator can also front AWS Application Load Balancer endpoints with regional support.

Health Checks

- Global Accelerator runs health checks on endpoints.
- Routes traffic only to **healthy** endpoints.
- Health checks can be customized (protocol, port, path, interval).

Traffic Distribution

- **Traffic Dial**:
    - Controls percentage of traffic sent to a region.
    - Allows easy shifting of traffic for testing, failover, blue/green deployments.
- **Client IP Preservation**:
    - Optionally preserves the original client IP.
    - Works with NLBs.

Routing and Performance

- Routes traffic over AWS's global network rather than the public internet.
- Reduces **latency**, **jitter**, and **packet loss**.
- Uses the optimal AWS edge location for user requests.
- Automatically finds the best healthy endpoint.

Failover and High Availability

- Monitors endpoint health with health checks.
- Automatic rerouting to healthy endpoints or regions if failures occur.
- Ensures high availability for global applications.

Pricing

- **Based on:**
    - Fixed monthly fee per accelerator.
    - Data transfer costs:
        - Accelerated data transfer (traffic routed over AWS backbone).
        - Non-accelerated traffic (standard internet path).

Use Cases

- Global web applications needing low latency and high availability.
- Disaster recovery setups needing quick failover across regions.
- Gaming applications with real-time networking requirements.
- IoT applications needing predictable latency.
- Moving to multi-region architecture with seamless failover.
- Blue/green deployments with controlled traffic shifting.

Security

- Integrates with AWS Shield Standard for DDoS protection (included).
- AWS WAF can be used with ALB/NLB behind Global Accelerator.
- Static IP addresses simplify firewall whitelisting and client configurations.
- Supports AWS Certificate Manager (ACM) for SSL/TLS certificates at the ALB/NLB layer.

Comparison with CloudFront

Feature	**AWS Global Accelerator**	**Amazon CloudFront**
Primary Use	Improving **performance and availability** of global applications with static IP and intelligent routing.	**Content Delivery Network (CDN)** for caching and distributing static/dynamic content.
Caching	No caching.	Edge caching for content.
Static IPs	Yes.	No.
Protocol Support	TCP and UDP.	HTTP and HTTPS only.
Routing	Routes at L4 (network layer) to AWS endpoints.	Routes at L7 (application layer) to content origins.
Latency Improvement	Via AWS private backbone and intelligent routing.	Via edge caching close to users.

Integration with Other AWS Services

- Works with ALB, NLB, EC2, and Elastic IP endpoints.
- Supports AWS Certificate Manager via ALB/NLB.
- Can be used alongside AWS WAF and Shield.
- Health checks integrate with CloudWatch for monitoring.

Exam Tips

- AWS Global Accelerator improves availability and performance by routing traffic over the AWS global network.
- Provides static IP addresses that stay the same even if endpoints change.
- Supports TCP and UDP.
- Traffic dial allows gradual traffic shifting.
- Uses health checks for failover and resilience.
- Best for global, latency-sensitive applications that need consistent performance.
- Not a CDN, no content caching like CloudFront.
- Pricing includes monthly fee plus accelerated data transfer costs.
- Compare carefully with CloudFront when selecting for exam scenarios.

Quick Summary

AWS Global Accelerator is a global traffic management service that uses AWS's private network to route traffic to optimal endpoints, improving availability, performance, and resilience for global applications. It offers static IP addresses, health checks, failover, and fine-grained traffic control without caching content.

Amazon Route 53

What It Is

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It connects user requests to AWS resources like EC2 instances, load balancers, and S3 buckets, as well as external endpoints. It also supports domain registration and health checking for DNS failover.

Key Features

- Authoritative DNS service providing reliable and cost-effective domain resolution
- Supports domain registration for a wide variety of TLDs
- Traffic management via routing policies including latency-based, geolocation, and weighted routing
- DNS health checks and DNS failover for high availability
- Integrated with AWS services such as ELB, S3, and CloudFront
- Fully compliant with IPv4 and IPv6

Routing Policies

- **Simple Routing:** Single record with one value; straightforward DNS resolution
- **Weighted Routing:** Distribute traffic across multiple resources with assigned weights
- **Latency-based Routing:** Route traffic to the lowest-latency endpoint
- **Failover Routing:** Primary/secondary configurations for high availability
- **Geolocation Routing:** Route traffic based on the geographic location of the requester
- **Geoproximity Routing (using Traffic Flow):** Bias traffic to endpoints based on geographic distance
- **Multivalue Answer Routing:** Return multiple healthy records to improve client-side load balancing

Domain Registration

- Register and manage domain names directly in Route 53
- Supports WHOIS privacy protection
- Automatic domain renewal options
- Easy integration with Route 53 hosted zones for DNS management

Health Checks and DNS Failover

- Monitor endpoint health via HTTP, HTTPS, and TCP checks
- Configure DNS failover to redirect traffic to healthy endpoints
- Health checks can monitor CloudWatch alarms for complex monitoring scenarios
- Support for alias records that integrate health check status

Alias Records

- Route 53–specific record type that points to AWS resources without additional DNS queries
- No charge for DNS queries to alias records
- Used with: ELB, CloudFront, API Gateway, S3 static websites, Global Accelerator, and VPC endpoint services
- Supports apex (root) domain records

Integration with AWS Services

- Direct integration with ELB for load balancing
- Supports CloudFront distributions for CDN integration
- Can route traffic to S3 static website endpoints
- Works with AWS Global Accelerator to route traffic through AWS edge network

High Availability and Scalability

- Globally distributed DNS infrastructure with edge locations worldwide
- Designed for 100% availability SLA
- Automatically scales to handle very large query volumes

Security

- Supports DNSSEC (Domain Name System Security Extensions) for domain registration and DNS hosting
- IAM policies control access to Route 53 resources
- Integrated with AWS CloudTrail for logging API calls
- Supports private hosted zones for VPC-internal DNS resolution

Traffic Flow

- Visual editor for creating complex routing policies
- Supports versioning of traffic policies
- Automates creation of policies for geoproximity, latency, and failover routing

Private Hosted Zones

- Internal DNS resolution for resources within one or more VPCs
- Namespaces are private to the selected VPCs
- Useful for service discovery and VPC-internal communication

Pricing

- Billed for hosted zones (per zone per month)
- Charged per DNS query

- Additional cost for health checks and domain registration
- No extra cost for alias record queries to AWS endpoints

Use Cases

- Global DNS resolution for applications
- High-availability websites using health checks and DNS failover
- Geolocation-based traffic distribution
- Load balancing with weighted routing
- Private DNS for internal VPC service discovery
- Domain registration and management

Exam Tips

- Alias records enable apex domain routing to AWS resources
- Weighted routing splits traffic proportionally across endpoints
- Latency-based routing improves user experience with lower latency
- Health checks can trigger DNS failover to backup endpoints
- Geolocation and geoproximity routing customize traffic targeting based on user location
- Private hosted zones provide internal-only DNS resolution within VPCs
- Route 53 integrates deeply with AWS services like ELB, CloudFront, and Global Accelerator
- DNSSEC ensures integrity of DNS responses and prevents spoofing

Quick Summary

Amazon Route 53 is AWS's scalable and highly available DNS and domain registration service. It supports flexible routing policies, DNS failover with health checks, seamless AWS integration via alias records, and private DNS zones for VPCs, making it a critical component for resilient, globally distributed architectures.

AWS Transit Gateway

What It Is

AWS Transit Gateway is a network transit hub that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and on-premises networks through a single gateway. It simplifies and scales network architectures by acting as a central router for traffic flowing between attached networks.

Key Features

- Central hub for connecting multiple VPCs, on-premises networks, and AWS services
- Simplifies many-to-many VPC peering connections
- Supports thousands of VPC attachments
- Scales elastically with your network
- Acts as a regional resource, but can support inter-Region peering

Attachments

- **VPC Attachments**
    - Connect a VPC to the Transit Gateway using a route table
    - Each VPC can have its own subnet associations
- **VPN Attachments**
    - Connect on-premises networks over AWS Site-to-Site VPN
    - Supports static and dynamic routing (BGP)
- **Direct Connect Gateway Attachments**
    - Connect Direct Connect Gateway to Transit Gateway for hybrid connectivity
    - Supports cross-Region VPC access
- **Peering Attachments**
    - Connect Transit Gateways in different AWS Regions
    - Supports global network architectures

Routing

- Uses Transit Gateway route tables to manage traffic flow
- Each attachment can associate with one or more route tables
- Controls which attachments can communicate with each other
- Supports segmentation for isolating traffic between networks

Inter-Region Peering

- Enables peering between Transit Gateways in different AWS Regions
- Provides private, low-latency connectivity across regions

- Data traffic does not traverse the public internet

Bandwidth and Performance

- Highly scalable throughput supporting tens of Gbps
- Designed for large-scale, high-performance network architectures
- AWS manages the scaling, availability, and fault tolerance

Multicast Support

- Native multicast support for workloads needing IP multicast
- Useful for real-time data distribution such as financial trading or media streaming

Security and Access Control

- Supports AWS Resource Access Manager (RAM) to share Transit Gateway across AWS accounts
- IAM policies control API access
- Traffic does not traverse the internet unless configured to do so
- Integrated with AWS CloudTrail for auditing API calls

Integration with AWS Services

- AWS Direct Connect for dedicated hybrid connectivity
- AWS Site-to-Site VPN for secure internet-based hybrid connections
- AWS Resource Access Manager for sharing with other accounts
- AWS Network Firewall can be deployed in VPCs connected via Transit Gateway

Pricing and Billing

- Charged based on attachments per hour
- Data processing charges per GB for traffic through Transit Gateway
- Inter-Region peering has separate pricing based on data transferred between regions

Use Cases

- Hub-and-spoke network architectures
- Simplifying many-to-many VPC peering
- Hybrid cloud architectures connecting on-premises to multiple VPCs
- Multi-Region applications requiring private connectivity
- Centralized egress filtering and inspection through shared security VPCs

Exam Tips

- Transit Gateway simplifies connecting multiple VPCs compared to full mesh peering
- Route tables control which attachments can talk to each other

- Supports VPN, Direct Connect, VPC, and peering attachments
- Inter-Region peering enables cross-Region private connectivity
- Supports multicast workloads
- Resource Access Manager (RAM) enables sharing across AWS accounts
- Recommended for large-scale, complex network topologies

Quick Summary

AWS Transit Gateway is a highly scalable, central networking hub that connects VPCs and on-premises networks through a single gateway. It simplifies routing, supports hybrid and multi-Region architectures, and integrates with AWS services to create secure, manageable, large-scale network designs.

Amazon VPC

What It Is

Amazon Virtual Private Cloud (VPC) lets you provision a logically isolated section of AWS where you can launch AWS resources in a virtual network you define. You have full control over networking configuration, including IP address ranges, subnets, route tables, and gateways.

Key Features

- Complete control over virtual networking environment
- Define custom IP address ranges using IPv4 and IPv6 CIDR blocks
- Create public, private, or VPN-only subnets
- Control routing with route tables
- Configure network gateways and NAT for internet access
- Integration with AWS Directory Service, Lambda, ECS, RDS, and many AWS services

Subnets

- **Subnet**: a range of IP addresses in your VPC
- **Public subnets**: have route to an internet gateway
- **Private subnets**: no direct route to the internet
- Multiple subnets can span multiple Availability Zones for HA
- Subnet CIDR blocks must not overlap

Route Tables

- Control routing for subnets
- Default route table provided, can create custom tables
- Routes define traffic destinations and targets (e.g., Internet Gateway, NAT Gateway, Virtual Private Gateway)

Internet Gateway (IGW)

- Horizontally scaled, redundant gateway for internet access
- Required for public subnets to communicate with the internet
- Must attach one IGW per VPC

NAT Gateway / NAT Instance

- Allows instances in private subnets to initiate outbound internet traffic while blocking inbound connections
- NAT Gateway is managed, highly available within an AZ
- NAT Instance is a user-managed EC2 instance acting as NAT

Elastic IP Addresses

- Static, public IPv4 addresses associated with AWS account
- Used for consistent addressing of EC2 instances, NAT gateways

VPC Peering

- Connects two VPCs to route traffic privately using AWS backbone
- Supports intra- and inter-region peering
- No transitive peering: traffic must go through direct peering connections
- Must update route tables and security groups

AWS Transit Gateway

- Central hub for connecting multiple VPCs and on-prem networks
- Simplifies many-to-many VPC connectivity
- Supports transitive routing
- Integrated with AWS Direct Connect and VPN

AWS PrivateLink

- Provides private connectivity to AWS services or customer-managed services
- Exposes services as endpoint services in your VPC
- Uses VPC endpoints with Elastic Network Interfaces (ENIs)

VPC Endpoints

- Private connections to supported AWS services without internet
- Types: Interface Endpoints (powered by PrivateLink) and Gateway Endpoints (for S3 and DynamoDB)
- Improve security by avoiding public internet exposure

VPN Connections

- Connect on-premises networks to AWS VPC over encrypted VPN tunnels
- Supports static or dynamic routing with BGP
- Can use AWS Site-to-Site VPN or Customer Gateway devices

Virtual Private Gateway

- AWS side of a VPN connection
- Enables connectivity between on-premises network and VPC over VPN

Customer Gateway

- On-premises side of a VPN connection
- Represents physical or software device in customer network

Carrier Gateway

- Used with AWS Outposts to provide connectivity to carrier networks
- Supports traffic between Outposts and the carrier's network

Security Groups

- Virtual firewall for controlling inbound/outbound traffic at instance level
- Stateful: return traffic is automatically allowed
- Rules defined by protocol, port, and CIDR

Network ACLs (NACLs)

- Optional stateless firewall for subnets
- Control inbound and outbound traffic at the subnet level
- Supports allow and deny rules
- Evaluated in order of rule number

Flow Logs

- Capture information about IP traffic going to and from network interfaces
- Delivered to CloudWatch Logs or S3
- Used for troubleshooting, compliance, security monitoring

IPv6 Support

- Assign IPv6 CIDR blocks to VPC and subnets
- Internet Gateway supports IPv6 traffic
- Security groups and NACLs can filter IPv6 traffic

VPC Sharing

- Share subnets with other AWS accounts using AWS Organizations
- Enables centralized VPC management in multi-account environments

Elastic Network Interfaces (ENIs)

- Virtual network cards attached to EC2 instances
- Support multiple IP addresses, security groups, and MAC addresses
- Used for high availability networking, failover

Pricing

- No charge for creating or using VPC itself
- Charges apply for NAT Gateway, VPN connections, Transit Gateway, PrivateLink endpoints, traffic across AZs and peering connections

Use Cases

- Hosting secure, scalable applications in isolated network environments
- Hybrid cloud networking with on-premises data centres
- Multi-VPC architectures with centralized routing via Transit Gateway
- Providing private access to AWS services with PrivateLink
- Securely exposing internal services to partners via VPC endpoints

Exam Tips

- Internet Gateway required for internet access in public subnets
- NAT Gateway enables private subnet instances to reach the internet
- Security Groups are stateful, NACLs are stateless
- VPC Peering does not support transitive routing
- Transit Gateway enables hub-and-spoke architectures with transitive routing
- PrivateLink provides private connectivity to AWS and partner services
- Flow Logs help monitor and troubleshoot traffic flows
- Interface Endpoints connect to services over PrivateLink; Gateway Endpoints for S3 and DynamoDB
- IPv6 support is built-in without NAT requirements for internet access

Quick Summary

Amazon VPC gives you full control over your virtual networking in AWS, including IP ranges, subnets, route tables, gateways, and security controls. It enables you to securely connect resources within AWS and to on-premises networks, while providing powerful tools for managing traffic, access, and compliance.

AWS Artifact

What It Is

- A self-service portal for accessing AWS compliance-related documents, like:
    - Audit reports
    - Security and compliance certifications
    - Agreements (e.g., HIPAA BAA, GDPR DPA)

Key Features

Two Main Types of Content

- **Artifact Reports**:
    - Downloadable compliance reports such as:
        - SOC 1, SOC 2, SOC 3
        - ISO certifications
        - PCI DSS, etc.
    - Verifies AWS's compliance with various industry standards.
- **Artifact Agreements**:
    - Let you review and accept legal agreements like:
        - Business Associate Addendum (BAA) for HIPAA
        - GDPR Data Processing Addendum (DPA)

Key Benefits

- Provides transparency into AWS's compliance posture.
- Helps customers with their own audit, risk, and compliance programs.
- No need to contact AWS Support for reports, self-service access.

Security & Access

- IAM policies control who can download or accept agreements.
- All activity is logged in AWS CloudTrail for auditing.

Use Cases

- Auditors reviewing AWS's compliance before migrating sensitive workloads.
- Businesses needing official proof of AWS compliance (e.g., SOC reports).
- Organizations requiring legal agreements like HIPAA BAA or GDPR DPA before storing regulated data.

Pricing

- AWS Artifact is free to use.

Exam Tips

- AWS Artifact = compliance documents and agreements portal.
- Use it to get SOC, ISO, PCI, and other audit reports.
- Accept HIPAA BAA and GDPR DPA agreements in Artifact.
- IAM can control access; CloudTrail logs activity.
- Available as a self-service tool, no AWS support ticket needed.

Quick Summary

AWS Artifact is a free self-service portal that provides access to AWS compliance reports and agreements (e.g., SOC, ISO, HIPAA BAA, GDPR DPA), helping customers meet regulatory and audit requirements.

AWS Audit Manager

What It Is

AWS Audit Manager is a service that helps you continuously audit AWS usage to simplify risk assessments and compliance with regulations and industry standards. It automates evidence collection to reduce manual effort and maintain audit readiness.

Key Features

- Automates evidence collection for audits
- Supports prebuilt frameworks for common compliance standards (e.g., CIS, GDPR, PCI DSS, HIPAA)
- Custom frameworks to meet internal or industry-specific requirements
- Mapping of AWS resources and configurations to control requirements
- Continuous monitoring to maintain audit readiness
- Evidence stored securely and organized for auditor review

Assessment Frameworks

- Prebuilt frameworks cover common compliance standards
- Custom frameworks can be built from scratch or by modifying existing frameworks
- Controls defined within frameworks specify what evidence to collect and how to evaluate it
- Mapping of controls to AWS resources ensures relevant evidence is gathered

Assessments

- Created from frameworks to evaluate specific environments or accounts
- Define scope by selecting AWS accounts and services
- Automates collection of evidence based on defined controls
- Ongoing assessments provide continuous visibility into compliance posture

Evidence Collection

- Automatic collection from AWS services and resources
- Manual evidence upload supported for non-AWS resources or custom requirements
- Evidence includes configuration snapshots, policies, logs, and API activity
- Evidence is time-stamped and organized by control for easy auditor access
- Supports export of evidence to AWS S3 for long-term storage or sharing

Continuous Auditing

- Ongoing monitoring ensures compliance posture is maintained
- Tracks changes over time, enabling identification of noncompliance as it occurs
- Reduces point-in-time audit preparation burden

- Facilitates proactive remediation of issues

Integrations

- AWS Organizations to manage multi-account environments
- AWS CloudTrail for capturing API activity
- AWS Config for resource configuration tracking
- AWS Security Hub for security findings and posture insights
- Amazon S3 for evidence storage and export

Security and Access Control

- Evidence encrypted in transit and at rest
- IAM policies control access to Audit Manager features and assessments
- Integration with AWS Key Management Service (KMS) for encryption key management
- Fine-grained access permissions for team members and auditors

Reporting and Exporting

- Downloadable evidence reports for auditors
- Export evidence to Amazon S3 buckets
- Organized evidence folders by control and assessment
- Facilitates external auditor access without exposing the entire AWS environment

Supported Compliance Frameworks

- PCI DSS
- CIS AWS Foundations Benchmark
- HIPAA
- GDPR
- SOC 2
- ISO 27001
- Custom frameworks to meet internal requirements

Pricing

- Charged per assessment per month
- Additional costs for storing evidence in S3
- Costs depend on volume of collected evidence and number of active assessments

Use Cases

- Preparing for regulatory audits (PCI DSS, HIPAA, SOC 2, ISO 27001)
- Automating compliance evidence collection to reduce manual effort

- Continuous compliance monitoring across AWS accounts
- Supporting internal security and risk assessments
- Streamlining third-party auditor reviews with organized evidence

Exam Tips

- AWS Audit Manager automates evidence collection for compliance audits
- Supports both prebuilt and custom frameworks
- Integrates with AWS services like CloudTrail, Config, and Security Hub
- Evidence is securely collected, time-stamped, and exportable
- Helps maintain continuous compliance readiness rather than one-time audits
- Supports multi-account environments through AWS Organizations
- IAM controls and KMS integration ensure secure access and encryption

Quick Summary

AWS Audit Manager helps AWS customers simplify compliance and audit processes by automating evidence collection, supporting standard and custom frameworks, and maintaining continuous audit readiness with secure, organized, and exportable evidence.

AWS Certificate Manager (ACM)

What It Is

- A fully managed AWS service that provisions, manages, and deploys SSL/TLS certificates.
- Used to secure websites, applications, APIs, and other endpoints.

Key Features

1. Free Public Certificates

- You can request SSL/TLS certificates for use with:
    - Elastic Load Balancers (ALB/ELB)
    - Amazon CloudFront
    - Amazon API Gateway
- These public certificates are free and auto-renewed.

2. Private Certificates (ACM PCA)

- Create and manage private CA (Certificate Authority).
- Use for internal applications, microservices, or VPN devices.
- Paid service with fine-grained control over private cert issuance.

3. Automatic Renewal

- ACM handles renewal and deployment of certs for supported AWS services automatically.
- Reduces manual overhead and downtime risk.

4. Integrated with AWS Services

- **Seamless integration with:**
    - CloudFront
    - Elastic Load Balancers (ALB/CLB/NLB)
    - API Gateway
    - Elastic Beanstalk
    - AWS CloudFormation

5. Secure Key Management

- AWS manages the private key for the certificate.
- Customers cannot download or export private keys for public certs.

Validation Methods

1. DNS Validation (Recommended)

- You add a CNAME record in your DNS to prove domain ownership.
- Preferred for automation and renewal.

2. Email Validation

- Verification email sent to admin/contact/tech email of the domain.
- Manual and prone to expiration if not confirmed.

Use Cases

- Securing web apps hosted on AWS with HTTPS.
- Protecting custom domains on CloudFront and API Gateway.
- Automating certificate lifecycle (renewals, deployments).
- Internal TLS usage via ACM Private CA.

Pricing

- Public certificates are free.
- Private CA (ACM PCA) incurs cost based on CA hours and certificate issuance.

Security & Compliance

- AWS manages private keys securely.
- Certificates use 2048-bit RSA keys or ECDSA.
- Supports FIPS-compliant algorithms.

Exam Tips

- Use ACM for SSL/TLS certificates on AWS services.
- Public certs = free + auto-renewed, but only usable with supported AWS resources.
- Use ACM PCA for issuing internal/private certs.
- Choose DNS validation for easier automation.
- For custom domain HTTPS on CloudFront, use ACM in the us-east-1 (N. Virginia) region.
- ACM does not support downloading the private key for public certs.

Quick Summary

AWS Certificate Manager provides free, automatically managed public certificates and scalable private certificate authority services for securing your applications and services using SSL/TLS.

AWS CloudHSM

What it is

- Managed Hardware Security Module (HSM) service in AWS Cloud.
- Let's you generate and use your own encryption keys using dedicated HSM appliances.
- Fully customer-controlled hardware for highest security and compliance requirements.

Key Features

- Dedicated, single-tenant HSMs in your VPC.
- FIPS 140-2 Level 3 validated.
- You have full administrative control over the HSM.
- Supports standard cryptographic APIs: PKCS#11, Java JCE, Microsoft CNG.
- Scales horizontally, adds more HSMs to cluster.
- Automated backups for durability.
- Integrated with AWS services (though not as seamless as KMS).

How It Works

- AWS provisions the HSM in your VPC.
- You manage users, keys, and policies directly.
- AWS manages maintenance, monitoring, backups.
- You connect securely to HSM using standard protocols.

Use Cases

- Compliance requirements needing FIPS 140-2 Level 3 HSMs.
- Regulatory controls demanding customer-exclusive hardware.
- Managing PKI (Public Key Infrastructure).
- Secure key generation and storage.
- Offloading SSL/TLS termination.
- Code signing.

Differences from AWS KMS

- KMS = AWS-managed service for encryption keys.
- CloudHSM = Customer-managed, dedicated hardware.
- KMS is simpler, integrated, easier for general use.
- CloudHSM is used when full control and strongest compliance are required.

Pricing

- Pay per HSM instance hourly.
- Additional data transfer costs may apply.
- No upfront fees.

Security and Compliance

- FIPS 140-2 Level 3 validation.
- Customer controls all keys and operations.
- Data encrypted with keys never leaves the HSM unencrypted.
- AWS can't access or recover customer keys.

Exam Tips

- CloudHSM = dedicated hardware, full customer control.
- Required for strict compliance and regulatory use cases.
- FIPS 140-2 Level 3 certified.
- Compare to KMS: KMS is simpler but AWS-managed.
- Used for PKI, code signing, SSL/TLS offload, secure key storage.
- Keys generated in CloudHSM stay under customer's exclusive control.

Quick Summary

AWS CloudHSM is a fully managed hardware security module service that gives customers dedicated, FIPS 140-2 Level 3 validated HSMs in their VPC for secure key management and compliance.

AWS Key Management Service (AWS KMS)

What it is

- Fully managed service to create, control, and manage encryption keys.
- Central service for encrypting data across AWS.
- Integrated with most AWS services (e.g., S3, EBS, RDS, Lambda, SQS).

Key Features

- **Customer Master Keys (CMKs):**
  - Main resources in KMS.
  - Can be AWS-managed, customer-managed, or imported.
  - Define permissions with IAM policies and key policies.
- **AWS-managed keys:**
  - Simple, AWS handles everything.
  - Limited customization.
- **Customer-managed keys (CMKs):**
  - Full control over lifecycle, rotation, and permissions.
  - Audit via AWS CloudTrail.
- **Envelope Encryption:**
  - Encrypts data keys with CMKs.
  - Improves performance for large-scale encryption.
- **Integrated with AWS CloudTrail:**
  - Logs all key usage for auditing and compliance.
- **Import Your Own Keys:**
  - For compliance/regulatory needs.
- **Multi-Region Keys:**
  - Replicate CMKs across regions for global apps.

How It Works

- Apps call KMS API to encrypt/decrypt data or data keys.
- KMS never exposes plaintext of CMKs.
- Uses envelope encryption, data encrypted with data key, data key encrypted with CMK.

Use Cases

- Encrypt S3 objects (SSE-KMS).
- Encrypt EBS volumes.

- Encrypt RDS databases.
- Lambda environment variables.
- Application-layer encryption with KMS APIs.
- Regulatory/compliance needs for key management.

Pricing

- Pay per CMK stored monthly.
- Pay per KMS API request (encrypt, decrypt, generate data key).

Security and Compliance

- FIPS 140-2 validated HSMs used under the hood.
- Customer-defined access policies.
- CloudTrail logs for full visibility.
- Regionally isolated keys by default (except multi-region keys).

Exam Tips

- AWS KMS = central, managed key management.
- CMKs = AWS-managed or customer-managed.
- SSE-KMS = S3 objects encrypted with KMS keys.
- Integrated with many AWS services.
- Use CloudTrail for logging key usage.
- For full hardware control, use AWS CloudHSM.
- Envelope encryption = encrypt data keys with CMKs for performance and security.
- Import keys for strict compliance requirements.

Quick Summary

AWS KMS is the central service for creating and managing encryption keys across AWS, enabling secure encryption of data at rest and in transit, with fine-grained control, auditing, and integration with AWS services.

Amazon Cognito

What It Is

Amazon Cognito is a fully managed service that provides authentication, authorization, and user management for web and mobile apps. It allows developers to add user sign-up, sign-in, and access control to applications securely and at scale.

Key Features

- User sign-up and sign-in with customizable UI
- User pools for user directory management
- Federated identities to allow sign-in with social and SAML providers
- Secure token generation for authentication and authorization
- Integration with AWS IAM for access control to AWS resources
- Supports MFA and advanced security features
- User profiles and attributes storage

User Pools

- Managed user directories to handle registration and authentication
- Supports email and phone number verification
- Built-in support for Multi-Factor Authentication (MFA)
- Password policies and account recovery settings
- Hosted UI for sign-up and sign-in flows
- Integration with social identity providers (Google, Facebook, Amazon, Apple)
- Integration with SAML 2.0 and OIDC providers
- Provides OAuth 2.0 endpoints for token-based authentication
- Tokens: ID Token, Access Token, Refresh Token
- Lambda triggers for customizing workflows (e.g., pre-sign-up, post-confirmation)

Identity Pools (Federated Identities)

- Provide temporary AWS credentials for accessing AWS services
- Support sign-in with User Pools, social providers, SAML, OIDC, and unauthenticated guest access
- Integrates with AWS IAM roles for fine-grained access control
- Role mapping to assign different IAM roles based on user attributes or provider

Authentication Flows

- User Pools manage authentication and generate JWT tokens
- Identity Pools exchange tokens for AWS credentials via STS

- Supports custom authentication challenges with AWS Lambda triggers
- OAuth 2.0 authorization code and implicit grant flows supported
- Secure token storage and management

Security Features

- Multi-Factor Authentication (MFA) with SMS or TOTP apps
- Adaptive authentication and risk-based security
- Account recovery with email or phone number
- Encryption of data at rest and in transit
- AWS KMS integration for encryption keys
- Device tracking and remembered devices
- Advanced security features to detect compromised credentials and unusual sign-in attempts

Integration with AWS Services

- **IAM**: Control access to AWS resources
- **API Gateway**: Use Cognito Authorizer to protect REST and WebSocket APIs
- **AWS AppSync**: Secure GraphQL APIs with Cognito authentication
- **Lambda**: Invoke functions with Cognito identities
- **S3**: Control object access based on user identities
- **Mobile SDKs**: iOS, Android, JavaScript, and more for integrating Cognito into apps

Tokens and Sessions

- **ID Token:** Contains user profile information (JWT)
- **Access Token:** Used to authorize API calls
- **Refresh Token:** Used to obtain new ID and Access tokens
- Token expiration settings configurable in User Pool settings

Deployment and Management

- Fully managed, scalable service
- Supports region-specific deployments
- User Pools and Identity Pools managed separately
- User data stored securely in AWS-managed directories
- Integrates with CloudWatch for monitoring and metrics

Pricing

- **User Pools:** Charged based on Monthly Active Users (MAUs)

- Free tier includes 50,000 MAUs per month
- **Identity Pools:** Charges for AWS STS calls and data transfer
- Advanced Security Features priced separately

Use Cases

- Add sign-up/sign-in to mobile and web applications
- Federate user access across social and enterprise identity providers
- Enable secure access to AWS resources for authenticated users
- Build serverless backends that authenticate and authorize users
- Implement Multi-Factor Authentication and advanced security controls

Best Practices

- Enable MFA for additional security
- Use custom domains for OAuth flows
- Store minimal sensitive data in user attributes
- Secure Lambda triggers with IAM permissions
- Use IAM roles with least privilege when granting AWS resource access
- Monitor metrics and logs with CloudWatch
- Rotate encryption keys regularly

Exam Tips

- User Pools manage user directories and authentication
- Identity Pools provide AWS credentials for resource access
- Supports social providers, SAML, OIDC for federated sign-in
- Integrates with API Gateway, AppSync, S3, Lambda
- Supports MFA and adaptive authentication
- Generates ID, Access, and Refresh tokens (JWT)
- Use IAM roles for access control based on identity
- Advanced security features help detect compromised credentials

Quick Summary

Amazon Cognito is a fully managed service for adding user sign-up, sign-in, and access control to applications. It includes User Pools for managing user directories and authentication, and Identity Pools for providing AWS credentials to authenticated users, with support for social, SAML, and OIDC identity providers, MFA, and advanced security features.

Amazon Detective

What It Is

Amazon Detective is a fully managed security service that simplifies the process of investigating potential security issues or suspicious activities. It automatically collects log data from AWS resources and uses machine learning, statistical analysis, and graph theory to build interactive visualizations for investigation.

Key Features

- Automatically collects and organizes data from AWS CloudTrail, Amazon VPC Flow Logs, Amazon GuardDuty findings, and AWS Security Hub
- Creates a behaviour graph that links disparate pieces of security data for faster analysis
- Provides visualizations for entities like AWS accounts, EC2 instances, IAM users, and IP addresses
- Helps security teams quickly identify the root cause of incidents
- No need for manual data collection or complex querying
- Automatically updates and retains up to 12 months of security data
- Data is processed and stored in a graph model that enables efficient querying and visualization
- Integrates with Amazon GuardDuty for streamlined threat detection and investigation
- Maintains context over time, helping to understand changes in behaviour

How It Works

- Ingests data from AWS security services (CloudTrail, VPC Flow Logs, GuardDuty, Security Hub)
- Organizes data into a linked set of entities in a graph model
- Provides dashboards and entity profiles to explore relationships and activity
- Investigators can navigate between entities, review timelines, and compare historical behaviour
- No agents or additional log forwarding is required

Supported Data Sources

- AWS CloudTrail for API activity logs
- VPC Flow Logs for network traffic metadata
- Amazon GuardDuty for threat detection findings
- AWS Security Hub for additional context from security alerts

Entity Types Analysed

- AWS Accounts
- EC2 Instances

- IAM Users and Roles
- IP Addresses

Security Use Cases

- Investigate suspicious login activity or unauthorized access attempts
- Trace lateral movement and network anomalies
- Identify misconfigured permissions or policy changes
- Review behaviour before and after GuardDuty findings
- Understand long-term patterns for persistent threats

Access and Permissions

- Integrated with AWS Organizations for centralized investigation
- Delegated administrator account can view behaviour across all member accounts
- IAM roles and policies control access to Amazon Detective

Pricing

- Charged based on the volume of data ingested from supported sources
- No additional charge for querying or visualizing the data
- Data is retained for up to 12 months automatically

Best Practices

- Enable GuardDuty and Detective together for a unified threat detection and investigation workflow
- Use in combination with AWS Security Hub to centralize alerts
- Grant security teams read-only access to Amazon Detective dashboards
- Monitor key entities like IAM users and EC2 instances for unusual activity

Exam Tips

- Amazon Detective is used for security investigations and root cause analysis
- Works by analysing CloudTrail, VPC Flow Logs, and GuardDuty findings
- Builds a behaviour graph to visualize entity relationships and changes over time
- No manual data collection is needed—data is ingested and correlated automatically
- Not used for real-time alerting, but for deep dive forensic analysis
- Complements, but does not replace, services like GuardDuty and Security Hub

Quick Summary

Amazon Detective helps security teams quickly investigate and understand the root cause of security issues using visualized and correlated data from AWS security services. It reduces the time and effort needed for analysis by automatically building a behaviour graph and providing interactive dashboards for security-relevant entities.

AWS Directory Service

What It Is

AWS Directory Service is a managed service that enables you to set up and run directories in the AWS Cloud or connect AWS resources with existing on-premises Microsoft Active Directory (AD).
It is essential for managing user access and enabling authentication for AWS resources and applications.

Why Use AWS Directory Service

- Manage user authentication and authorization.
- Enable single sign-on (SSO) for AWS applications and services.
- Support Microsoft Active Directory-aware applications in AWS.
- Simplify management of user accounts, groups, and policies.

Directory Types

1. AWS Managed Microsoft AD

- Actual Microsoft Active Directory deployed in AWS.
- AWS manages patching, replication, and maintenance.
- Supports trust relationships with on-premises AD.
- Ideal for full AD compatibility.
- **Features:**
    - Kerberos and NTLM authentication.
    - Group Policy support.
    - AD-integrated applications (e.g., SQL Server, SharePoint).
    - MFA integration with AD FS.
- **Two editions:**
    - **Standard Edition**: up to 5,000 objects.
    - **Enterprise Edition**: up to 500,000 objects.

2. AD Connector

- **Proxy** that connects AWS to your existing on-premises Active Directory.
- No directory data stored in AWS.
- Allows AWS services (e.g., EC2, WorkSpaces) to authenticate against on-premises AD.
- **Use Cases:**
    - Leverage existing credentials.
    - Enable SSO to AWS Management Console.

- **Advantages:**
    - Simple to set up.
    - Cost-effective, no AD infrastructure to manage in AWS.
- Limitations:
    - Requires reliable network connectivity to on-premises AD.

3. Simple AD

- Standalone, AWS-managed directory based on Samba 4 Active Directory Compatible Server.
- Inexpensive option for smaller workloads.
- **Supports basic AD features:**
    - User and group management.
    - Kerberos-based SSO.
- **Limitations:**
    - Not as feature-rich as AWS Managed Microsoft AD.
    - No support for trust relationships.
- **Two sizes:**
    - **Small**: up to 500 users.
    - **Large**: up to 5,000 users.

Integration with AWS Services

- **Amazon WorkSpaces**:
    - Integrates for user authentication.
- **Amazon WorkDocs**:
    - Use existing AD users and groups.
- **Amazon WorkMail**:
    - Native integration with AWS Managed Microsoft AD.
- **EC2 Windows Instances**:
    - Domain join support.
- **AWS Single Sign-On (SSO)**:
    - Use Directory Service as an identity source.

Security

- **AWS Managed Microsoft AD:**
    - Multi-AZ deployment for high availability.

    - Automated daily snapshots.
    - Supports AWS KMS encryption for data at rest.
    - Enforces strong password policies.
- **AD Connector:**
    - No AWS-stored credentials.
    - Relies on on-premises AD security.
- **Simple AD:**
    - Managed by AWS.
    - Basic security controls.

Pricing

- **AWS Managed Microsoft AD:**
    - Pay per domain controller hour.
    - Cost varies by edition (Standard vs. Enterprise).
- **AD Connector:**
    - Pay per connector hour.
    - Two sizes: Small and Large.
- **Simple AD:**
    - Pay per directory hour.
    - Cost depends on size (Small or Large).

Use Cases

- Migrate on-premises Windows workloads to AWS while keeping existing AD integration.
- Enable centralized user management and authentication for AWS-based applications.
- Provide AD support for AWS WorkSpaces and WorkDocs.
- Simplify Windows instance domain joins in the cloud.
- Allow on-premises users to authenticate to AWS resources with existing credentials.

Exam Tips

- AWS Managed Microsoft AD is best for full AD support in AWS, including trust relationships.
- AD Connector is ideal if you want to reuse on-premises AD without storing AD data in AWS.
- Simple AD is for small, cost-sensitive environments with basic AD needs.
- AWS Managed Microsoft AD supports Kerberos, NTLM, Group Policy, and integrates with MFA and AD FS.

- AD Connector requires reliable network connectivity to on-premises AD.
- Directory Service integrates with WorkSpaces, WorkDocs, WorkMail, EC2 Windows, and AWS SSO.
- Supports high availability with multi-AZ deployments.

Comparison Table

Feature	AWS Managed Microsoft AD	AD Connector	Simple AD
Type	Fully managed Microsoft AD	Proxy to on-premises AD	Standalone, Samba-based AD-compatible
AWS-Hosted Directory	Yes	No	Yes
Trust Relationships	Yes	Uses on-premises AD trust	No
Group Policy Support	Yes	Via on-premises AD	Limited
Use Cases	Full AD compatibility, hybrid environments	Leverage existing AD without migration	Small-scale AD needs, low cost
Editions/Sizes	Standard/Enterprise	Small/Large	Small/Large
Typical Customers	Enterprises with AD workloads	Customers with existing AD wanting integration	Small organizations with simple needs

Quick Summary

AWS Directory Service offers multiple ways to use or connect to Active Directory in AWS. AWS Managed Microsoft AD provides a fully managed, highly available Microsoft AD in the cloud. AD Connector links AWS resources to your on-premises AD without migrating it. Simple AD offers a low-cost, standalone AD-compatible directory for smaller deployments.

AWS Firewall Manager

What It Is

- A security management service that centrally configures and manages firewall rules across multiple AWS accounts and resources.
- Designed for AWS Organizations to enforce consistent security policies.

Key Features

1. Centralized Security Policy Management

- Create security policies once and apply them across accounts and resources in your organization.
- Supports AWS Organizations integration, manage all member accounts from a central admin account.

2. Supported Security Services

- **AWS WAF**: Manage and deploy Web ACLs across accounts.
- **AWS Shield Advanced**: Apply advanced DDoS protection consistently.
- **AWS Network Firewall**: Define VPC-level traffic filtering rules.
- **VPC Security Groups**: Audit and remediate security group rules.
- **Route 53 Resolver DNS Firewall**: Manage DNS filtering policies.

3. Automatic Policy Enforcement

- New resources (e.g., ALBs, CloudFront distributions) automatically get the appropriate policies.
- Ensures compliance even as new accounts or resources are created.

4. Compliance Monitoring

- Continuously checks for policy compliance across all accounts.
- Flags and optionally remediates noncompliant resources.

5. Integration with AWS Organizations

- Firewall Manager can only work if your accounts are in an AWS Organization.
- **Policies can target:**
    - Specific OUs (Organizational Units)
    - Entire Organization
    - Specific accounts

Use Cases

- Enforcing consistent AWS WAF rules across all company websites.
- Automatically protecting new resources with AWS Shield Advanced.

- Centrally managing Network Firewall rules for all VPCs.
- Auditing and fixing overly permissive security groups.
- Applying DNS Firewall rules to block malicious domains.

Security & Compliance

- Helps meet enterprise security requirements for policy consistency.
- Reduces misconfigurations and drift across accounts.
- Provides visibility into security posture across the organization.

Pricing

- AWS Firewall Manager charges based on:
    - Number of policies created.
    - AWS security services used (e.g., WAF, Shield Advanced, Network Firewall).
- Additional costs for underlying services (WAF rules, Shield Advanced, Network Firewall usage).

Exam Tips

- Firewall Manager = central policy management across multiple accounts.
- Requires AWS Organizations to function.
- Supports AWS WAF, Shield Advanced, Network Firewall, Security Groups, and DNS Firewall.
- Auto-applies security policies to new resources.
- Monitors compliance and can remediate automatically.
- Great for enterprises with multi-account strategies.

Quick Summary

AWS Firewall Manager lets you centrally configure, deploy, and enforce security policies (WAF, Shield Advanced, Network Firewall, Security Groups, DNS Firewall) across all AWS accounts in an organization for consistent, automated security.

Amazon GuardDuty

What It Is

- Managed threat detection service that continuously monitors AWS accounts, workloads, and data for malicious or unauthorized activity.
- Uses machine learning, anomaly detection, and threat intelligence feeds.

Key Features:

1. Continuous Threat Detection

- Monitors AWS resources for:
    - Unauthorized API calls.
    - Reconnaissance (port scanning, unusual login attempts).
    - Malware activity.
    - Data exfiltration attempts.
    - Cryptocurrency mining.

2. Data Sources Analysed

- **VPC Flow Logs**: Network traffic patterns.
- **AWS CloudTrail**: API activity across AWS accounts.
- **DNS Logs**: Domain name lookups.
- **Kubernetes Audit Logs** (optional): For Amazon EKS clusters.
- **Malware Scan**: (Optional feature) Scans EC2 workloads for malware.

3. Threat Intelligence Feeds

- Integrated AWS threat intelligence.
- Third-party threat intel feeds.
- Helps detect known malicious IPs, domains, and behaviours.

4. Findings

- Alerts called Findings describe suspicious activities.
- Classified by severity (Low, Medium, High).
- Includes recommended remediation steps.

5. Integration with AWS Security Hub

- Findings can be sent to AWS Security Hub for centralized security visibility.
- Integrates with AWS EventBridge for custom workflows and automation.

Key Benefits:

- Fully managed, no need to deploy or maintain infrastructure.

- Continuous monitoring without performance impact on workloads.
- No agent required, works from AWS log data.
- Regional service with cross-account support via AWS Organizations.

Pricing

- Pay-as-you-go pricing:
  - Based on volume of analysed logs.
  - Malware scans billed per GB scanned.
- No upfront costs.

Use Cases

- Detecting compromised EC2 instances running crypto-miners.
- Monitoring for unusual API calls that suggest account compromise.
- Flagging data exfiltration attempts via DNS or traffic anomalies.
- Centralized threat detection for multi-account environments.

Security & Compliance

- Supports AWS Organizations for multi-account monitoring.
- Findings logged to AWS CloudWatch Events / EventBridge for alerting and remediation.
- Helps meet compliance requirements for continuous monitoring.

Exam Tips

- GuardDuty = threat detection, not prevention.
- Analyses VPC Flow Logs, CloudTrail, DNS Logs, and Kubernetes Audit Logs.
- No agent installation required.
- Integrates with Security Hub, EventBridge, and AWS Organizations.
- Findings include severity levels and recommended remediation.
- Pay-as-you-go pricing with no upfront commitments.

Quick Summary

Amazon GuardDuty is a managed AWS service that continuously monitors AWS accounts and workloads for malicious activity using threat intelligence and machine learning, delivering actionable security findings with no infrastructure to manage.

AWS IAM

What It Is

- AWS Identity and Access Management (IAM) is a global AWS service that enables you to securely control access to AWS services and resources.
- Used to authenticate (who) and authorize (what they can do) in your AWS account.

Key IAM Concepts

1. IAM Users

- Represents individual people or applications.
- Has credentials (passwords, access keys) for AWS access.
- Can be assigned permissions directly or via groups.

2. IAM Groups

- Collection of users.
- Attach policies to a group to grant permissions to all members.
- Simplifies permission management for multiple users.

3. IAM Roles

- Temporary credentials for AWS resources or users.
- Used for:
    - EC2 instances accessing AWS services.
    - Cross-account access.
    - AWS services assuming roles.
    - Federated users via SAML, OIDC, or custom identity providers.
- No long-term credentials.

4. IAM Policies

- JSON documents defining **permissions**.
- Attached to **users, groups, or roles**.
- Types:
    - **Managed Policies**:
        - AWS-managed (created/maintained by AWS).
        - Customer-managed (created by you).
    - **Inline Policies**:
        - Embedded directly in a user, group, or role.

5. IAM Policy Elements

- **Effect**: Allow or deny.
- **Action**: API operations permitted/denied.
- **Resource**: Targeted AWS resources.
- **Condition**: Optional, for fine-grained control.

6. IAM Permissions Boundaries

- Advanced feature that limits maximum permissions a role or user can have.
- Useful in delegating permissions without over-provisioning.

7. IAM Identity-Based vs Resource-Based Policies

- **Identity-based policies**: Attached to IAM users, groups, roles.
- **Resource-based policies**: Attached to resources (e.g., S3 bucket policies, Lambda function policies).
- Resource policies enable cross-account access.

8. IAM Access Analyzer

- Analyses policies to identify resources shared outside your account.
- Helps maintain least privilege by highlighting unintended access.

9. IAM Password Policy

- Enforces account-wide password rules for IAM users.
- Configure complexity, rotation, reuse prevention.

10. MFA (Multi-Factor Authentication)

- Adds an extra layer of security beyond username and password.
- Supports:
    - Virtual MFA apps.
    - U2F security keys.
    - SMS MFA (less recommended).
- Recommended for the root user and privileged IAM users.

11. IAM Best Practices

- Enable MFA, especially on the root account.
- Use least privilege: Grant only the permissions needed.
- Rotate access keys regularly.
- Avoid using the root account for daily tasks.
- Use roles for EC2 instances instead of embedding access keys.

- Use AWS Organizations and Service Control Policies (SCPs) for multi-account setups.

12. IAM Roles for Service Accounts / Applications

- **EC2 Instance Profiles**:
  - Attach IAM roles to EC2 instances.
  - Automatically provide temporary credentials via Instance Metadata Service (IMDS).
- **Lambda Execution Role**:
  - Grants permissions for AWS Lambda functions to access other AWS services.
- **Cross-Account Roles**:
  - Allow users in one AWS account to access resources in another.

13. Temporary Security Credentials

- IAM roles issue **short-lived credentials** via:
  - AWS Security Token Service (STS).
  - Used for:
    - Cross-account access.
    - Federation with external IdPs.
    - AWS services assuming roles.

14. Federation and SSO

- IAM supports integration with external identity providers:
  - SAML 2.0 (e.g., Active Directory, Okta).
  - OIDC (e.g., Google).
  - AWS SSO (now AWS IAM Identity Center).
- Enables single sign-on for accessing AWS Console or CLI.

15. AWS Managed Policies vs Customer Managed Policies

- **AWS Managed Policies**:
  - Pre-built by AWS.
  - Easy to use, less customizable.
  - Automatically updated by AWS.
- **Customer Managed Policies**:
  - Fully customizable.
  - Created and maintained by your organization.

16. Service-Linked Roles

- Pre-defined roles linked directly to AWS services.
- Required for certain AWS service operations (e.g., ECS, Auto Scaling).
- Managed entirely by AWS.

17. IAM Credential Report

- CSV file listing all IAM users and their credentials.
- Includes details on passwords, access keys, and MFA status.
- Helps with **auditing and compliance**.

18. IAM Access Advisor

- Shows last used information for permissions.
- Helps identify unused permissions for least privilege enforcement.

Pricing

- IAM itself is free.
- You pay for the underlying AWS resources you use.

Exam Tips

- IAM is global (not tied to a region).
- Roles = temporary credentials, no passwords or access keys.
- Policies define who can do what on which resources.
- Use least privilege everywhere.
- MFA strongly recommended for root and privileged users.
- IAM Access Analyzer = identify unintended external access.
- Use IAM roles for EC2, Lambda, and cross-account access.
- Federation enables SSO with external IdPs.

Quick Summary

AWS IAM is a global service for managing user access and permissions securely. It uses users, groups, roles, and policies to control who can access what in AWS, with best practices like least privilege, MFA, and credential rotation to ensure security.

Amazon Inspector

What It Is

- Automated vulnerability management service that scans AWS workloads for software vulnerabilities and unintended network exposure.
- Helps maintain security and compliance by continuously assessing EC2, container images (ECR), and Lambda functions.

Key Features

Automated, Continuous Scanning

- Continuously scans for:
    - Operating system vulnerabilities (CVEs).
    - Application vulnerabilities.
    - Network reachability (exposure to the internet).
    - Container image vulnerabilities (Amazon ECR).
    - Lambda function code packages.

Integration with AWS Services

- **EC2:**
    - Scans installed packages, OS.
    - Checks network exposure.
- **Amazon ECR:**
    - Scans container images for known vulnerabilities.
    - Continuous scanning on push.
- **AWS Lambda:**
    - Scans function code for known vulnerabilities.

Findings and Scoring

- Generates security findings with:
    - CVE details.
    - Severity scores (CVSS-based).
    - Recommended remediation steps.
- Prioritizes by risk to help with patching.

AWS Organizations Integration

- Supports multi-account environments.
- Central management of scans and findings across accounts.

Automated Remediation Workflows

- **Findings can be sent to:**
    - AWS Security Hub.
    - Amazon EventBridge.
    - Custom remediation workflows.

Key Benefits

- No manual assessment setup, fully managed.
- Continuous, automated vulnerability management.
- Integrates with AWS developer workflows (ECR, Lambda).
- Helps maintain compliance requirements.

Pricing

- **Pay-as-you-go:**
    - Based on number of EC2 instances, container images scanned, and Lambda function scans.
- No upfront costs.

Use Cases

- Ensuring EC2 instances are patched against CVEs.
- Scanning container images before deployment.
- Identifying Lambda code vulnerabilities.
- Meeting security compliance frameworks (PCI DSS, CIS benchmarks).

Security & Compliance

- Helps enforce security best practices.
- Supports continuous vulnerability management.
- Findings integrate with Security Hub for centralized visibility.

Exam Tips

- Amazon Inspector = vulnerability scanning, not threat detection.
- Continuous, automated scans for EC2, ECR containers, Lambda.
- Findings include CVE details, severity, remediation.
- Integrates with Security Hub and EventBridge.

Quick Summary

Amazon Inspector is an automated AWS service that continuously scans EC2 instances, container images, and Lambda functions for software vulnerabilities and network exposure, helping ensure secure and compliant workloads.

Amazon Macie

What It Is

- Managed data security and privacy service that uses machine learning to discover, classify, and protect sensitive data in Amazon S3.
- Designed to help detect PII (Personally Identifiable Information) and other sensitive data.

Key Features

Automated Sensitive Data Discovery

- Continuously monitors and automatically discovers sensitive data in S3 buckets.
- Uses machine learning and pattern matching to identify:
    - PII (e.g., names, addresses, credit card numbers).
    - Financial data.
    - Credentials.

Data Classification

- Classifies objects in S3 buckets based on sensitive data types.
- Built-in managed data identifiers (common PII patterns).
- Supports custom data identifiers for specific patterns.

Security and Privacy Alerts

- Generates findings when sensitive data is detected.
- Findings include:
    - Bucket name.
    - Object details.
    - Type of sensitive data.
    - Severity levels.

S3 Bucket-Level Analysis

- Evaluates S3 bucket permissions.
- Identifies publicly accessible buckets or buckets shared with other accounts.
- Alerts on misconfigurations that could lead to data exposure.

Integration with AWS Services

- Findings sent to AWS Security Hub for central visibility.
- EventBridge for automation workflows (e.g., remediation).
- AWS CloudWatch for alerting.

Key Benefits

- Fully managed, no infrastructure to deploy.
- Helps meet compliance requirements (GDPR, HIPAA, PCI DSS).
- Reduces risk of unintentional data exposure.
- Supports multi-account management via AWS Organizations.

Pricing

- Pay-as-you-go pricing:
    - Based on S3 buckets evaluated.
    - Volume of data processed for sensitive data discovery.
- No upfront costs.

Use Cases

- Identifying sensitive data stored in S3 buckets.
- Monitoring S3 buckets for public access or sharing.
- Enforcing data security policies.
- Supporting compliance audits with automated reporting.

Security & Compliance

- Helps maintain data privacy standards.
- Integrates with AWS Organizations for multi-account governance.
- Supports CloudTrail logging for full auditing.

Exam Tips

- Amazon Macie = S3 data discovery and classification.
- Detects PII and sensitive data automatically using ML.
- Evaluates S3 bucket permissions for public or cross-account access.
- Integrates with Security Hub, EventBridge, CloudWatch.
- Supports custom data identifiers for specific patterns.

Quick Summary

Amazon Macie is a fully managed service that automatically discovers, classifies, and protects sensitive data in S3 using machine learning, helping ensure privacy, security, and compliance

AWS Network Firewall

What It Is

AWS Network Firewall is a managed service that provides customizable, stateful, and stateless network traffic filtering for your Amazon VPC. It helps protect VPC networks by inspecting and controlling inbound and outbound traffic at the subnet level.

Key Features

- Managed, highly available, and scalable firewall service
- Supports both stateful and stateless inspection rules
- Deep packet inspection (DPI) and domain list rule groups
- Integration with AWS Firewall Manager for centralized management
- Supports Suricata-compatible rule sets for advanced inspection
- Provides logging to Amazon S3, CloudWatch, and Kinesis Data Firehose
- Automatically scales to meet traffic demands
- Built-in high availability within an Availability Zone

Components

- **Firewall**: Defines the inspection policies and logging settings
- **Firewall Policy**: Contains rule groups and settings for traffic handling
- **Rule Groups**: Sets of stateless or stateful rules to inspect traffic
- **Stateless Rules**: Match criteria based on header fields without maintaining session state
- **Stateful Rules**: Track and inspect connection state for advanced filtering
- **Domain List Rules**: Allow or deny based on domain names in DNS traffic
- **Logging Configuration**: Define how flow logs and alert logs are sent to S3, CloudWatch Logs, or Firehose

Traffic Flow

- Deployed in a VPC subnet using AWS Gateway Load Balancer or as a VPC ingress/egress filter
- Traffic is routed through the firewall endpoint for inspection
- Supports routing rules in route tables to redirect traffic to firewall endpoints

Use Cases

- Protecting VPC subnets with inbound and outbound filtering
- Enforcing security policies across workloads in multiple accounts using AWS Firewall Manager
- Detecting and blocking known bad IP addresses and domains
- Preventing data exfiltration through domain-based filtering

Deployment

- Integrated with AWS Transit Gateway for centralized inspection of traffic between VPCs
- Can be deployed in individual VPCs for distributed architectures
- Automatically scales horizontally to meet throughput demands

Logging and Monitoring

- Send alert and flow logs to S3, CloudWatch Logs, or Kinesis Data Firehose
- Supports detailed analysis and forensic investigations
- Integration with CloudWatch Metrics for monitoring throughput and packet counts

Integration with Other AWS Services

- **AWS Firewall Manager:** Centrally configure and deploy firewall policies across accounts and VPCs in AWS Organizations
- **AWS Transit Gateway:** Integrate for centralized inspection of East-West and North-South traffic
- **Amazon VPC:** Direct integration for ingress and egress filtering
- **AWS Security Hub:** Aggregate findings and integrate with other security tooling

Pricing

- Charged based on firewall endpoint hours and the amount of traffic processed
- Separate pricing for stateful and stateless rule evaluations
- Logging costs based on delivery to S3, CloudWatch Logs, or Firehose

Best Practices

- Use AWS Firewall Manager for multi-account, multi-VPC management
- Define clear segmentation in VPC subnets to control traffic flow
- Maintain and update Suricata-compatible rule sets to address evolving threats
- Enable logging for audit and compliance purposes

Exam Tips

- AWS Network Firewall provides stateful and stateless filtering at the VPC level
- Supports integration with AWS Firewall Manager for centralized policy enforcement
- Can inspect traffic between VPCs via Transit Gateway integration
- Scales automatically with traffic without user-managed infrastructure

Quick Summary

AWS Network Firewall is a fully managed, scalable service that delivers customizable traffic filtering at the VPC level. It supports both stateful and stateless inspection, integrates with AWS services like Firewall Manager and Transit Gateway, and provides logging for compliance and auditing, helping secure AWS workloads against network threats.

AWS Resource Access Manager (AWS RAM)

What It Is

AWS Resource Access Manager (RAM) is a service that lets you securely share AWS resources with other AWS accounts, within your organization (AWS Organizations), or with Organizational Units (OUs).
It removes the need to duplicate resources across accounts, simplifying management and reducing costs.

Why Use AWS RAM

- Centralize management of shared resources.
- Avoid duplication of resources across accounts.
- Enforce access controls and security boundaries.
- Facilitate multi-account architectures recommended by AWS.
- Reduce operational overhead in large organizations.

Key Concepts

- **Resource Share**:
    - A container for resources you want to share.
    - Specifies which AWS accounts, Organizations, or OUs have access.
- **Principal**:
    - The AWS accounts, OUs, or entire organization with which you share resources.
- **Managed Permissions**:
    - Define what actions principals can perform on shared resources.
    - AWS provides default managed permissions for supported resource types.

Supported Resource Types

AWS RAM supports sharing of specific AWS resources across accounts. Examples include:

- **VPC Subnets**
    - Enables multiple accounts to launch resources into a shared subnet.
- **Transit Gateways**
    - Allows centralized routing across multiple accounts.
- **Route 53 Resolver Rules**
    - Enables sharing DNS resolution configurations.
- **License Manager Configurations**
    - Share software licenses across accounts.
- **AWS Outposts**

- **AWS Network Firewall Policies**
- **Cloud WAN Core Network Policies**
- **Resource Groups**
- **Custom resources integrated with AWS Service Catalog**

How It Works

1. **Create a Resource Share**:
    - Choose the resources you want to share.
    - Select the principals (accounts, OUs, or the whole Organization).
    - Choose permissions.
2. **Invitation Process**:
    - If sharing **outside your organization**, the recipient must accept the invitation.
    - **Within the organization**, sharing is automatic (if sharing with AWS Organizations accounts).
3. **Access and Usage**:
    - Shared resources appear in recipient accounts.
    - Recipients can use shared resources as if they owned them, within permission boundaries.

Sharing Types

- **Within Organization**:
    - Simplest method.
    - Automatic access (no invitation/acceptance needed).
- **With Specific AWS Accounts**:
    - Cross-account sharing even outside the organization.
    - Requires invitation acceptance.
- **Organizational Units (OUs)**:
    - Fine-grained control using AWS Organizations hierarchy.

Permissions and Security

- AWS RAM uses AWS Identity and Access Management (IAM) for defining policies.
- Managed permissions help control actions on shared resources.
- Sharing is limited to supported resource types, preventing accidental oversharing.

Monitoring and Auditing

- **AWS CloudTrail integration:**

- Records all AWS RAM API calls for audit and compliance.
- Tracks sharing creation, modification, and deletion events.

Pricing

- AWS RAM itself has no additional charges.
- You pay for the resources you create and use, even when shared.
  - Example: If you share a VPC subnet and someone launches an EC2 instance in it, that account pays for the EC2 instance.

Common Use Cases

- **Centralized Network Management**:
  - Share VPC subnets, Transit Gateways for consistent networking.
- **Hybrid Environments**:
  - Centralize DNS resolution rules with Route 53 Resolver Rules.
- **Cost Optimization**:
  - Avoid duplicating resources in multiple accounts.
- **License Management**:
  - Share License Manager configurations to enforce license limits.
- **Cloud WAN and Network Firewall Policies**:
  - Enforce consistent network security policies across accounts.

Best Practices

- Use AWS Organizations for easier, scalable sharing.
- Apply least privilege principles when defining managed permissions.
- Tag shared resources for better tracking and cost allocation.

Exam Tips

- AWS RAM lets you share resources securely across AWS accounts and organizations.
- Resource Shares are the central mechanism: you pick resources and principals.
- Supports VPC subnets, Transit Gateways, Route 53 Resolver Rules, and other specific resources.
- Sharing within an AWS Organization does not require acceptance.
- Cross-account sharing outside the org requires invitation acceptance.

Quick Summary

AWS Resource Access Manager (AWS RAM) lets you securely share supported AWS resources across accounts and organizations. It simplifies multi-account architectures by enabling central resource management, reduces duplication, and enforces access control using IAM and AWS Organizations.

AWS Secrets Manager

What It Is:

- A managed service that helps you store, manage, and retrieve secrets securely.
- Commonly used for database credentials, API keys, OAuth tokens, etc.
- Provides automatic rotation, fine-grained access control, and audit logging.

Key Features:

1. Secure Storage

- Secrets are encrypted at rest using AWS KMS.
- Automatically encrypted when stored, you can bring your own KMS key (CMK).

2. Automatic Rotation

- Supports automatic rotation of secrets using AWS Lambda.
- Prebuilt templates for common integrations (e.g., RDS, Redshift, Aurora).
- No downtime required during rotation.

3. Fine-Grained Access Control

- Use IAM policies and resource-based policies to control who can access specific secrets.
- Can restrict access to parts of a secret using IAM condition keys.

4. Audit and Monitoring

- Integrated with AWS CloudTrail, log every access and change.
- Supports CloudWatch for monitoring usage.

5. Cross-Account Access

- Supports secure sharing of secrets across AWS accounts using resource policies.

6. SDK and CLI Integration

- Retrieve secrets via API, SDK, or AWS CLI.
- Secrets are fetched at runtime, no need to hard-code credentials.

Use Cases:

- Managing RDS credentials securely with automatic rotation.
- Replacing hardcoded secrets in code or environment files.
- Securely managing third-party API keys and tokens.
- Sharing secrets across environments or accounts.

Secrets Manager vs Parameter Store

Feature	Secrets Manager	Parameter Store (Standard Tier)
Rotation	Built-in automatic rotation	Manual only
Secret type	Designed for sensitive secrets	Config data and secrets
Pricing	Paid service	Free (Standard), Paid (Advanced)
API rate limit	Higher throughput	Lower in standard tier
Encryption	KMS	KMS

Security

- Secrets encrypted with KMS at rest.
- IAM + Resource policies define access.
- Rotate secrets using Lambda functions.
- Audit all access via CloudTrail.

Pricing

- Charged per secret stored per month.
- Additional cost per 10,000 API calls.
- Rotation incurs Lambda execution cost.

Exam Tips

- Use Secrets Manager when secrets need automatic rotation.
- Do not hardcode credentials, fetch them securely via Secrets Manager API.
- Use Parameter Store for less sensitive, infrequently updated configs.
- Always configure KMS encryption and IAM access control.
- Know how Secrets Manager integrates with RDS for auto-rotation.

Quick Summary

AWS Secrets Manager is a secure and scalable service for storing and automatically rotating sensitive secrets like database credentials, API keys, and tokens with full integration into AWS security and monitoring tools.

AWS Security Hub

What It Is

- Centralized security service that aggregates, organizes, and prioritizes security findings from multiple AWS services and partner tools.
- Provides a single pane of glass for your AWS security posture.

Key Features

1. **Aggregated Security Findings**

- Collects findings from:
  - AWS services (e.g., GuardDuty, Inspector, Macie, Firewall Manager).
  - AWS Partner solutions (third-party security tools).
- Normalizes findings into a standard format for easy analysis.

2. **Security Standards & Best Practices**

- Built-in security standards to assess AWS resources:
  - CIS AWS Foundations Benchmark.
  - AWS Foundational Security Best Practices.
  - PCI DSS v3.2.1 standard.
- Runs automated checks against these standards.
- Generates findings for non-compliant resources.

3. **Consolidated Dashboard**

- Central view of:
  - Active findings.
  - Compliance status.
  - Historical trends.
- Easy filtering and sorting.

4. **Automated Response and Remediation**

- Integrates with AWS EventBridge.
- Supports building custom remediation workflows.
- Example: Auto-remediate non-compliant security groups.

5. **Cross-Account & Multi-Region Aggregation**

- View findings from multiple AWS accounts and regions in one place.
- Central security operations across an AWS Organization.

Integrates With

- AWS GuardDuty
- AWS Inspector
- AWS Macie
- AWS Firewall Manager
- AWS IAM Access Analyzer
- AWS Systems Manager
- AWS Config
- Third-party security solutions (via AWS Security Finding Format - ASFF)

Security & Compliance

- Helps meet audit and compliance requirements.
- Supports continuous security posture management.

Pricing

- Charged based on:
    - Number of security checks per account.
    - Volume of ingested findings.
- No upfront fees.

Use Cases

- Centralize and simplify security management.
- Automate compliance checks against industry standards.
- Enable automated remediation for faster incident response.
- Provide executive-level visibility into AWS security posture.

Exam Tips

- Security Hub = central dashboard for security findings.
- Supports CIS Benchmarks, AWS Best Practices, PCI DSS checks.
- Integrates with EventBridge for automated responses.
- Cross-account and multi-region aggregation.
- Findings are standardized using ASFF.

Quick Summary

AWS Security Hub is a centralized security service that aggregates, prioritizes, and standardizes findings from AWS services and third-party tools to give a unified view of your security posture. It continuously evaluates your AWS environment against industry standards like CIS, AWS Best Practices, and PCI DSS, generating findings for non-compliance.

AWS Shield

What It Is

- Managed Distributed Denial of Service (DDoS) protection for AWS resources.
- Defends against network and transport layer (Layer 3/4) and application layer (Layer 7) attacks.

Key Features

1. AWS Shield Standard

- Included at no extra cost for all AWS customers.
- Automatic protection against common, most frequent DDoS attacks.
- Protects services like:
    - Amazon CloudFront
    - Route 53
    - Elastic Load Balancer (ALB/NLB)
    - AWS Global Accelerator
- Always-on detection and automatic inline mitigation.
- No need to enable, always active.

2. AWS Shield Advanced

- Paid service with enhanced DDoS protection.
- Protects against larger and more sophisticated attacks.
- Features include:
    - 24x7 access to the AWS DDoS Response Team (DRT).
    - Advanced attack detection and mitigation at application and network layers.
    - Protection against volumetric attacks, state exhaustion, and application-layer attacks.
    - Cost Protection: DDoS cost protection against scaling-related charges during an attack.
    - Real-time attack visibility and detailed reports.
    - Layer 7 protections when combined with AWS WAF.
    - Global threat environment dashboard.

Integrated AWS Services

- CloudFront
- Route 53
- Global Accelerator

- Elastic Load Balancers (ALB/NLB)
- EC2 instances (with Elastic IP)
- AWS Global Infrastructure

Pricing

- **Shield Standard:** Free, automatic for all customers.
- **Shield Advanced:** Monthly fee per protected resource + additional data transfer fees.

Security and Compliance

- Protects mission-critical apps from downtime and degraded performance due to DDoS.
- Helps meet compliance requirements for resilience and uptime.

AWS Shield vs AWS WAF

- **AWS Shield:** DDoS protection (Layer 3/4 + some Layer 7).
- **AWS WAF:** Custom rules for Layer 7 (HTTP/S) filtering.
- Often used together:
    - Shield for DDoS mitigation.
    - WAF for fine-grained application-layer security.

Use Cases

- Protect public-facing web apps from DDoS attacks.
- Maintain high availability during attack attempts.
- Reduce operational risk and cost from DDoS incidents.
- Meet compliance standards for uptime and security.

Exam Tips

- Shield Standard is always on and free for AWS customers.
- Shield Advanced provides:
    - 24/7 AWS DDoS Response Team access.
    - Enhanced, customizable protections.
    - DDoS cost protection.
    - Detailed attack diagnostics.
- Combine AWS WAF + Shield Advanced for best Layer 7 protection.

Quick Summary

AWS Shield is a managed DDoS protection service with two levels: Shield Standard (free, automatic protection) and Shield Advanced (paid, enhanced protection with expert support and cost safeguards).

AWS WAF

What It Is

- A Web Application Firewall to protect web applications from common exploits.
- Let's you control which HTTP/S requests reach your resources.
- Helps block, allow, or count web requests based on defined rules.

Key Features

1. Web ACL (Access Control List)

- The main container for WAF rules.
- Attach Web ACL to:
    - Amazon CloudFront distributions.
    - Application Load Balancers (ALB).
    - AWS App Runner services.
    - Amazon API Gateway.
    - AWS AppSync.

2. Rules and Rule Groups

- Define conditions for filtering web traffic.
- Types of rules:
    - IP match conditions (allow/block specific IPs).
    - String match (inspect headers, body, query string).
    - Geo match (block/allow by country).
    - Regex pattern sets.
    - Size constraints.
    - Rate-based rules (limit requests per IP).
- **Managed Rule Groups**
    - AWS provides pre-configured rules against common threats (SQL injection, XSS).
    - AWS Marketplace sellers also offer curated rule groups.

3. Rate-Based Rules

- Automatically block IPs that exceed a configurable request threshold.

4. Bot Control

- Identify and block unwanted bots and scrapers.
- AWS-managed detection with optional CAPTCHA challenges.

5. Custom Responses

- Return custom error pages or messages when blocking traffic.

6. Logging and Metrics

- Detailed request logs sent to Amazon Kinesis Data Firehose.
- Integration with CloudWatch Metrics and Alarms.

Pricing:

- Pay for:
    - Web ACLs.
    - Number of rules per ACL.
    - Number of requests processed.

Security and Compliance:

- Helps meet PCI DSS, GDPR, and other compliance requirements.
- Protects against OWASP Top 10 threats.

Use Cases:

- Blocking common web exploits like SQL injection and cross-site scripting.
- Limiting request rates to prevent DDoS or scraping.
- Allowing or blocking traffic from specific geographies.
- Integrating with CloudFront for global edge protection.
- Protecting APIs hosted on API Gateway or AppSync.

AWS WAF vs AWS Shield

- **AWS WAF**: Protects at the application layer (Layer 7) with custom rules.
- **AWS Shield**: Protects against DDoS attacks.
    - **Shield Standard** = automatic protection, free.
    - **Shield Advanced** = additional DDoS protection and response team access.

Exam Tips:

- AWS WAF protects web apps at Layer 7.
- Attach WAF Web ACLs to CloudFront, ALB, API Gateway, App Runner, AppSync.
- Use Managed Rule Groups for easy protection.
- Rate-based rules to throttle or block abusive IPs.
- Logging via Kinesis Firehose, metrics via CloudWatch.
- Often used with AWS Shield and AWS Firewall Manager for complete protection.

AWS Fargate

What It Is

AWS Fargate is a serverless compute engine for containers that lets you run containers without managing servers or clusters. It works with Amazon ECS and Amazon EKS to provide on-demand, right-sized compute capacity.

Key Features

- Serverless compute for containers
- Works with ECS and EKS
- No need to provision or manage EC2 instances
- Per-second billing based on vCPU and memory
- Automatic scaling based on task or pod requirements
- Integrated with AWS IAM, VPC, CloudWatch, and other services

How It Works

- Define task definitions in ECS or pod specs in EKS
- Specify CPU and memory requirements per task or pod
- AWS Fargate provisions and manages compute resources
- Supports networking with ENIs in your VPC
- Supports logging with CloudWatch Logs

Integration with ECS

- Runs ECS tasks without EC2 instances
- Tasks get ENIs for networking in your VPC
- Supports Application Load Balancer and Network Load Balancer integration
- Uses ECS Service Auto Scaling to scale tasks based on demand
- Supports ECS Capacity Providers for managing mixed EC2 and Fargate workloads

Integration with EKS

- Runs Kubernetes pods without managing EC2 nodes
- EKS Fargate profiles specify which pods run on Fargate
- Provides separate ENIs for pod networking in your VPC
- Fully integrated with Kubernetes API for deployment and scaling

Task Definitions

- Define container images, CPU, memory, networking mode, and IAM roles
- Fargate supports Linux containers

- Supports secrets and environment variables with AWS Secrets Manager and SSM Parameter Store
- Define logging configuration for CloudWatch Logs

Networking

- Uses AWS VPC networking (awsvpc mode)
- Each task or pod gets its own ENI with private IP
- Supports security groups and NACLs
- Enables fine-grained network isolation between tasks or pods

Security

- IAM roles for tasks and pods provide fine-grained permissions
- Encryption of data at rest and in transit
- Supports Secrets Manager and Parameter Store for managing secrets
- VPC integration ensures private networking
- Supports AWS PrivateLink for secure access to AWS services

Monitoring and Logging

- Integrated with Amazon CloudWatch for logs and metrics
- Container-level logs pushed to CloudWatch Logs
- ECS and EKS metrics available in CloudWatch
- Integration with AWS X-Ray for tracing

Scaling

- Automatically scales tasks and pods based on demand
- ECS Service Auto Scaling for task-based scaling policies
- EKS Horizontal Pod Autoscaler (HPA) for Kubernetes-based scaling
- Per-task or per-pod resource allocation ensures efficient use of resources

Pricing

- Pay for vCPU and memory resources requested per second
- Separate pricing for Fargate Spot capacity (discounted for interruption-tolerant workloads)
- Additional charges for data transfer, load balancing, and storage

Use Cases

- Microservices architectures
- Batch processing workloads

- Event-driven applications
- Continuous integration and deployment pipelines
- Migrating legacy container workloads to serverless
- Running Kubernetes workloads without managing nodes

Benefits

- No server management or provisioning
- Fine-grained billing based on actual resource usage
- Improved security with task/pod-level isolation
- Integration with AWS services for networking, monitoring, and security
- Seamless scaling to meet demand

Exam Tips

- Fargate removes need to manage EC2 instances for ECS and EKS
- Supports per-task/pod billing for vCPU and memory
- Uses awsvpc networking mode with dedicated ENIs
- Integrated with IAM roles for fine-grained permissions
- Supports ECS Service Auto Scaling and EKS HPA
- Secrets management via Secrets Manager and Parameter Store
- Best for variable, unpredictable workloads or simplifying operations

Quick Summary

AWS Fargate is a serverless compute engine that runs containers without managing servers. It integrates with ECS and EKS to provide scalable, secure, and cost-efficient container deployments with per-second billing and deep AWS integration.

AWS Lambda

What It Is

AWS Lambda is a serverless, event-driven compute service that lets you run code without provisioning or managing servers. It automatically scales and manages compute resources in response to events and charges only for compute time consumed.

Key Features

- Serverless compute with no infrastructure to manage
- Runs code in response to events from AWS services or custom apps
- Automatically scales based on the number of events
- Supports multiple languages including Python, Node.js, Java, C#, Go, Ruby, PowerShell
- Pay only for compute time (per millisecond) and number of requests
- Supports up to 15 minutes per invocation

Triggers and Event Sources

- Can be triggered by AWS services such as S3, DynamoDB, Kinesis, SNS, SQS, EventBridge
- Supports API Gateway and Application Load Balancer for HTTP(S) endpoints
- Can be invoked directly using AWS SDKs or CLI
- Event source mappings allow Lambda to poll services like SQS and DynamoDB Streams

Execution Environment

- Each function runs in its own isolated environment
- Includes specified runtime, memory, temporary storage (/tmp up to 10 GB)
- Supports environment variables for configuration
- Can use Lambda Layers to share code and libraries across functions

Concurrency and Scaling

- Scales automatically based on incoming events
- Default concurrency limit per account/region (can request increases)
- Reserved concurrency guarantees a set number of concurrent executions
- Provisioned Concurrency keeps functions pre-initialized to reduce cold starts

Function Configuration

- Memory allocation: 128 MB to 10,240 MB (10 GB)
- CPU power proportional to memory
- Timeout setting up to 15 minutes
- Permissions managed via AWS IAM roles assigned to the function

Deployment and Versioning

- Supports function versions (immutable snapshots)
- Aliases can point to specific versions for safe deployments
- Supports blue/green deployments using AWS CodeDeploy with traffic shifting
- Code packages can be uploaded as .zip files or via container images (up to 10 GB)

Container Image Support

- Build and deploy Lambda functions as container images
- Supports up to 10 GB image size
- Use AWS ECR to store and manage images

Security

- IAM roles define what resources a function can access
- VPC integration allows functions to access resources in a VPC
- AWS KMS for encrypting environment variables
- Supports AWS Secrets Manager and AWS Systems Manager Parameter Store for secure configuration

Monitoring and Logging

- Integrated with Amazon CloudWatch Logs for function logs
- CloudWatch Metrics track invocation count, duration, errors, throttles
- AWS X-Ray provides tracing for performance analysis and debugging
- CloudTrail logs API calls made to Lambda

Pricing

- Pay per request (first 1 million requests per month are free)
- Pay per compute time in GB-seconds, rounded to nearest millisecond
- Additional charges for Provisioned Concurrency and data transfer

Use Cases

- Real-time file processing (e.g., S3 uploads)
- Event-driven data processing (e.g., DynamoDB Streams, Kinesis)
- Serverless APIs using API Gateway + Lambda
- Automation and infrastructure management tasks
- Backend for IoT, mobile, and web applications

Integration with AWS Services

- API Gateway: Create REST or WebSocket APIs backed by Lambda

- S3: Process object events like uploads or deletes
- DynamoDB Streams: React to data changes
- Kinesis: Real-time stream processing
- SNS/SQS: Asynchronous messaging and queue processing
- EventBridge: Event routing and orchestration
- Step Functions: Orchestrate multiple Lambda functions in workflows

Best Practices

- Use environment variables for configuration
- Break down monolithic functions into smaller, single-purpose functions
- Use Provisioned Concurrency for latency-sensitive workloads
- Minimize package size for faster cold starts
- Monitor with CloudWatch and set alarms on errors or throttles
- Secure IAM roles with least privilege access
- Use Lambda Layers to share common dependencies

Exam Tips

- Lambda is fully managed and serverless with automatic scaling
- Charges are based on request count and execution time
- Supports multiple event sources and triggers
- Can run up to 15 minutes per invocation
- IAM role attached to Lambda defines its permissions
- Integrates with VPCs for private resource access
- Supports container images for deployment
- Use Provisioned Concurrency to avoid cold start latency
- CloudWatch for logs and metrics, X-Ray for tracing
- Works well with API Gateway, S3, DynamoDB, Kinesis, SNS, SQS, EventBridge, Step Functions

Quick Summary

AWS Lambda is a serverless compute service that runs code in response to events, scales automatically, and eliminates the need to manage servers. It integrates with many AWS services, supports various languages and deployment models, and is priced based on execution time and requests, making it ideal for building scalable, event-driven applications.

AWS Backup

What It Is

AWS Backup is a fully managed backup service that makes it easy to centrally automate and manage backups across AWS services and on-premises resources. It provides centralized backup management, compliance tracking, and backup activity monitoring.

Key Features

- Centralized backup management across AWS services
- Policy-based automation for scheduling backups
- Backup vaults to store and manage backups securely
- Cross-region and cross-account backup capability
- Backup activity monitoring and auditing
- Integration with AWS Organizations for managing multiple accounts
- Supports backup of on-premises resources via AWS Storage Gateway

Supported AWS Services

- Amazon EBS volumes
- Amazon RDS databases (including Aurora)
- DynamoDB tables
- Amazon EFS file systems
- Amazon FSx file systems
- AWS Storage Gateway volumes
- Amazon EC2 instances
- AWS Backup Vault Lock for immutable backups

Backup Plans

- Define policies for creating backups on a schedule
- Specify frequency, lifecycle rules, and retention
- Support for tags to identify resources for backup
- Lifecycle rules for transitioning backups to cold storage
- Central policy management for consistency across resources

Backup Vaults

- Logical containers to store backups
- Supports encryption with AWS KMS
- Access control with IAM policies
- Cross-account access can be granted

- Vault Lock feature for Write-Once-Read-Many (WORM) compliance

Backup Lifecycle Management

- Define retention rules for backups
- Automate transition of backups to low-cost cold storage
- Control backup expiration to manage costs

Cross-Region and Cross-Account Backups

- Enable replication of backups to other AWS Regions for disaster recovery
- Support for cross-account backup sharing for secure collaboration
- Enhance resilience and meet compliance requirements

Backup Monitoring and Reporting

- AWS Backup console provides job status and history
- Integration with AWS CloudTrail for auditing API activity
- AWS CloudWatch integration for alerts and metrics
- Compliance reporting for backup plan adherence

AWS Organizations Integration

- Centralized backup policy management across accounts
- Apply backup plans consistently in multi-account setups
- Use delegated administrator account to manage backup policies

Security and Encryption

- Encryption at rest using AWS KMS keys
- Encryption in transit using TLS
- IAM roles and policies control access to backup resources
- Vault Lock for immutable, tamper-proof backups

Pricing

- Charges based on backup storage used (warm and cold storage)
- Charges for backup transitions between warm and cold storage
- Additional costs for cross-region backup copies
- No upfront fees or commitments—pay-as-you-go model

Use Cases

- Centralized and automated backup for AWS workloads
- Disaster recovery with cross-region backups
- Compliance with data retention and audit requirements

- Backup of hybrid cloud workloads via Storage Gateway
- Securing critical business data with encrypted, immutable backups

Best Practices

- Define clear backup policies for all critical resources
- Use Vault Lock for compliance and immutability
- Implement cross-region replication for disaster recovery
- Monitor backup activity with CloudWatch and CloudTrail
- Regularly test backup and restore processes

Exam Tips

- AWS Backup centralizes and automates backup management
- Supports cross-region and cross-account backups
- Backup Vaults store encrypted backups with access controls
- Vault Lock enforces Write-Once-Read-Many compliance
- Integrates with AWS Organizations for policy management
- Monitors and audits backup activity via CloudWatch and CloudTrail

Quick Summary

AWS Backup provides a centralized, automated way to manage backups across AWS services and on-premises systems. It supports policy-based automation, secure storage in backup vaults, compliance reporting, cross-region and cross-account replication, and integrates with AWS Organizations for large-scale management.

Amazon EBS

What It Is

Amazon Elastic Block Store (EBS) is durable block storage for use with Amazon EC2 instances. Provides persistent storage volumes that can be attached to instances in the same Availability Zone.

Key Characteristics

- Block-level storage for EC2.
- Persistent, data remains even after instance stops or terminates.
- Designed for 99.999% availability.
- Supports snapshots for backup and recovery.
- Encryptable using AWS KMS.
- Can be dynamically resized without downtime.

EBS Volume Types

1. General Purpose SSD (gp3)

- Baseline 3,000 IOPS, can provision up to 16,000 IOPS.
- Throughput up to 1,000 MB/s.
- Suitable for most workloads (boot volumes, dev/test).

2. Provisioned IOPS SSD (io1/io2)

- Designed for critical applications needing high performance.
- io1: Up to 64,000 IOPS.
- io2: More durability (99.999% durability SLA).
- Supports multi-attach (attach to multiple instances in same AZ).

3. Throughput Optimized HDD (st1)

- Low-cost, high-throughput workloads.
- Up to 500 MB/s throughput.
- Good for big data, data warehouses, log processing.
- Cannot be a boot volume.

4. Cold HDD (sc1)

- Lowest cost.
- Infrequently accessed data.
- Up to 250 MB/s throughput.
- Cannot be a boot volume.

Volume Features

- **Durability**:
  - Replicated within an AZ.
  - Designed for high availability and durability.
- **Elastic Volumes**:
  - Modify size, performance, and type without downtime.
- **Snapshots**:
  - Point-in-time backups.
  - Stored in Amazon S3.
  - Incremental, only changes since last snapshot.
  - Can be copied across regions.
- **Encryption**:
  - Uses AWS KMS.
  - Encrypts data at rest, data in transit between instance and volume, snapshots, and volume copies.
  - Enabled at creation; cannot encrypt an existing unencrypted volume directly (must copy to new encrypted volume).

Performance

- IOPS depends on volume type and size.
- Consistent and predictable performance with Provisioned IOPS.
- Can burst IOPS on gp3.
- Throughput and IOPS can be adjusted for gp3.

Multi-Attach (io1/io2)

- Attach a single io1/io2 volume to multiple Nitro-based EC2 instances in the same AZ.
- Enables shared storage scenarios like clustered applications.

Snapshots

- Stored in S3, but invisible as S3 objects.
- Incremental by design, saving storage costs.
- Used to restore volumes or create new volumes.
- Can copy snapshots to other regions for disaster recovery.
- **Fast Snapshot Restore (FSR)**:
  - Enables low-latency, fully-initialized volumes from snapshots.

EBS Lifecycle Manager

- Automates snapshot creation, retention, and deletion.
- Helps enforce backup policies consistently.

Data Lifecycle

- Create EBS volume in an AZ.
- Attach to EC2 instance in same AZ.
- Detach and reattach to other instances in same AZ.
- Delete when no longer needed.
- Snapshots allow cross-region or cross-account restoration.

Security

- Supports encryption at rest with AWS KMS.
- Transparent encryption for data in transit between instance and volume.
- IAM policies control access to snapshots and volumes.
- Supports encrypted snapshots and sharing encrypted snapshots with other accounts (requires KMS permissions).

Pricing

- Charged per GB per month for provisioned storage.
- Charged for provisioned IOPS (io1/io2).
- Snapshot storage charged per GB-month.
- Fast Snapshot Restore incurs additional cost.
- Data transfer within same AZ free; cross-AZ/region transfers billed separately.

Best Practices

- Use gp3 for general workloads.
- Use io1/io2 for critical low-latency workloads needing high IOPS.
- Take regular snapshots for backup and DR.
- Use EBS Lifecycle Manager to automate backups.
- Encrypt sensitive data with AWS KMS.
- Monitor performance with CloudWatch metrics.
- Consider multi-attach for shared-disk applications.
- Use Fast Snapshot Restore for production recovery scenarios.

Exam Tips

- EBS volumes persist independently from EC2 instances.
- Snapshots are incremental and stored in S3.
- Encryption at rest uses AWS KMS.
- You can resize volumes without downtime (Elastic Volumes).
- gp3 is general purpose with customizable IOPS.
- io1/io2 support high IOPS and Multi-Attach.
- st1 and sc1 are HDD options for large, sequential workloads—cannot be used as boot volumes.
- Fast Snapshot Restore reduces initialization times for volumes created from snapshots.
- Always provision in same AZ as EC2 instance.

Quick Summary

Amazon EBS provides durable, high-performance block storage for EC2, with flexible volume types, snapshot-based backups, encryption, and advanced features like Multi-Attach and Fast Snapshot Restore. Essential for building reliable, scalable, and secure applications on AWS.

Amazon EFS

What It Is

Amazon Elastic File System (EFS) is a fully managed, scalable, elastic NFS file system for use with AWS services and on-premises resources. Provides shared, concurrent access to thousands of EC2 instances.

Key Characteristics

- Managed Network File System (NFSv4.1, NFSv4.0).
- Elastic: Automatically grows and shrinks as files are added or removed.
- Scalable: Supports petabytes of data, high levels of throughput and IOPS.
- Accessible concurrently by multiple instances across AZs.
- Supports POSIX permissions for access control.

Storage Classes

1. **Standard**
    - For frequently accessed files.
    - Higher cost.
    - Designed for low latency, high throughput.
2. **Infrequent Access (IA)**
    - Lower cost for infrequently accessed files.
    - Charged per GB stored and per access request.
    - Ideal for files not read often but must be available.
3. **Lifecycle Management**
    - Automatically moves files to IA based on age.
    - Configurable transition policy (e.g., 30 days of no access).

Performance Modes

1. **General Purpose**
    - Default mode.
    - Low latency for latency-sensitive use cases.
    - Suitable for web servers, CMS, home directories.
2. **Max I/O**
    - Supports higher levels of aggregate throughput.
    - Higher latencies.
    - Best for big data, media processing, analytics workloads.

Throughput Modes

1. **Bursting Throughput**
    - Default.
    - Throughput scales with file system size.
    - Suitable for most workloads.
2. **Provisioned Throughput**
    - Specify throughput independent of storage size.
    - Useful for workloads needing higher, consistent throughput.

Availability and Durability

- Data is stored redundantly across multiple AZs in a region.
- Designed for 99.999999999% (11 9s) durability.
- Regional service: accessible from all AZs in a region.

Security

- Supports AWS KMS for encryption at rest.
- TLS encryption for data in transit.
- IAM policies and POSIX permissions for access control.
- Supports VPC security groups and network ACLs.
- EFS Access Points:
    - Managed application access.
    - Enforce user and group identities, root access control.

Mounting Options

- Mount directly on EC2 instances using NFS protocol.
- Mount targets in each AZ for high availability.
- Supported on:
    - Linux EC2 instances.
    - On-premises servers via AWS Direct Connect or VPN.
    - AWS services like ECS and Lambda (via EFS Access Points).

Backup and Restore

- AWS Backup supports EFS file systems.
- Centralized, automated backup service.
- Define backup plans and retention policies.
- Point-in-time recovery of file systems.

Use Cases

- Content management.
- Web serving and hosting.
- Home directories.
- Media processing workflows.

Pricing

- Charged based on:
    - Storage used (Standard or IA), Requests to IA storage class.
    - Provisioned throughput if used, Data transfer within VPC is free; across regions incurs charges.

Integration with Other AWS Services

- **EC2**: Primary mounting target.
- **ECS**: Persistent shared storage for containers.
- **Lambda**: Native integration with EFS via Access Points.
- **AWS Backup:** Centralized backup management.

Best Practices

- Use Lifecycle Management to reduce costs.
- Choose General Purpose performance mode for low-latency workloads.
- Use Max I/O for highly parallel, throughput-heavy workloads.
- Encrypt data at rest and in transit.
- Design for regional redundancy by using mount targets in each AZ.

Exam Tips

- EFS is POSIX-compliant, network file system.
- Supports concurrent access from thousands of instances.
- Elastic, managed, scales automatically with usage.
- Two storage classes: Standard and IA with Lifecycle Management.
- Can be mounted on-premises using Direct Connect or VPN.
- Integrated with AWS Backup for managed backups.

Quick Summary

Amazon EFS is a regional, managed, elastic NFS file system that supports scalable, concurrent access from thousands of instances. It offers multiple performance and storage classes, POSIX-compliant access controls, strong security and encryption options, and seamless integration with AWS services for building highly available, shared file-based workloads.

Amazon FSx

What It Is

Amazon FSx is a fully managed service that makes it easy to launch, run, and scale feature-rich, high-performance file systems in the cloud.
It provides native Windows and Linux file systems, eliminating the need to manage file servers or storage hardware.

Key Features

- Fully managed by AWS: patching, backups, monitoring.
- Supports multiple file system types to match different workloads.
- Provides high performance, scalable storage.
- Integrates with AWS security and networking.
- Supports encryption at rest and in transit.

Supported File Systems

1. Amazon FSx for Windows File Server

- Built on Microsoft Windows Server.
- Provides SMB protocol access.
- Supports Windows NTFS features:
    - ACLs
    - User quotas
    - Shadow copies
- Integrates with Active Directory for user authentication.
- Supports DFS namespaces and replication.
- Use Cases:
    - Windows-based applications.
    - Home directories.
    - Content management systems.

2. Amazon FSx for Lustre

- High-performance file system for fast processing of workloads.
- POSIX-compliant.
- Seamlessly integrates with Amazon S3:
    - Can link S3 buckets for data processing.
    - Results can be written back to S3.
- Sub-millisecond latencies.

- Throughput up to hundreds of GB/s.
- Use Cases:
    - Machine learning.
    - High-performance computing (HPC).
    - Media processing.

3. Amazon FSx for NetApp ONTAP

- Managed NetApp ONTAP file systems.
- Supports NFS, SMB, and iSCSI protocols.
- Advanced NetApp features:
    - Multi-protocol access.
    - Snapshots.
    - Data deduplication and compression.
    - Cloning and replication.
- Supports FlexCache for caching volumes in different regions.
- Can tier cold data to Amazon S3.
- Use Cases:
    - Enterprise workloads needing NetApp features.
    - Data replication and disaster recovery.
    - Hybrid cloud storage.

4. Amazon FSx for OpenZFS

- Managed OpenZFS file systems.
- POSIX-compliant, Linux-native.
- Supports NFS v3 and v4.1.
- High throughput and low latency.
- Use Cases:
    - Linux-based applications.
    - Container storage.
    - Dev/test environments.

Integration and Access

- Deploys inside VPC, with control over networking.
- Supports AWS Direct Connect and VPN for hybrid workloads.
- Integrated with AWS IAM for managing access.

- AWS Backup integration for automated backups.
- Data replication within and across AWS regions (for some file systems).

Security

- Encryption at rest using AWS KMS.
- Encryption in transit using SMB encryption, NFS encryption, and TLS.
- VPC Security Groups and Network ACLs for access control.
- IAM policies for API-level access.
- Supports AWS Identity Store integrations (e.g., Active Directory).

Performance

- **FSx for Windows:**
  - Up to tens of GB/s throughput.
  - Supports SSD and HDD storage.
- **FSx for Lustre:**
  - Hundreds of GB/s throughput.
  - Sub-millisecond latencies.
- **FSx for ONTAP:**
  - Advanced caching.
  - Efficient deduplication/compression.
- **FSx for OpenZFS:**
  - High IOPS.
  - Consistent low latency.

Backup and Recovery

- Supports automated daily backups.
- User-initiated backups at any time.
- Snapshots for point-in-time recovery.

Monitoring and Management

- Amazon CloudWatch for performance metrics and alarms.
- AWS CloudTrail logs API calls for auditing.
- AWS Backup integration for centralized backup management.
- Management via AWS Console, CLI, and SDKs.

Pricing

- Charged based on:

    - Storage capacity provisioned.
    - Throughput capacity.
    - Backup storage used.
    - Data transfer (for replication).
- Separate pricing for SSD vs. HDD storage.

Typical Use Cases

- Lift-and-shift Windows applications needing shared storage.
- HPC workloads using Lustre.
- Hybrid enterprise workloads needing NetApp features.
- Linux-native apps using OpenZFS.
- Media rendering, genomic analysis, ML training.

Comparison Table

FSx Type	Protocols	Best For	Key Features
FSx for Windows File Server	SMB	Windows apps, user shares	NTFS, AD integration, DFS, Shadow Copies
FSx for Lustre	NFS, POSIX	HPC, ML, analytics	S3 integration, sub-ms latency, high throughput
FSx for NetApp ONTAP	NFS, SMB, iSCSI	Enterprise workloads, hybrid storage	Snapshots, deduplication, FlexCache, tiering
FSx for OpenZFS	NFS v3/v4.1	Linux workloads, container storage	Snapshots, clones, compression, encryption

Exam Tips

- FSx is fully managed, no need to maintain file servers.
- Choose FSx for Windows for SMB/Windows workloads needing AD.
- Use FSx for Lustre for HPC and S3-integrated data processing.
- FSx for ONTAP offers advanced enterprise storage features with multi-protocol access.
- FSx for OpenZFS is best for Linux workloads with ZFS features.
- Supports encryption at rest (AWS KMS) and in transit.
- Integrated with VPC, IAM, CloudWatch, CloudTrail, AWS Backup.

Quick Summary

Amazon FSx provides fully managed, scalable, high-performance file systems tailored for Windows, Linux, HPC, and enterprise workloads. Choose the right FSx variant based on protocols, performance, and features needed for your application.

Amazon S3

What It Is

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance.
Store and retrieve any amount of data from anywhere on the web.

Core Concepts

- **Buckets**:
    - Top-level container for objects.
    - Globally unique name.
    - Defined in a single AWS Region.
- **Objects**:
    - Data stored in buckets.
    - Consists of key (name), value (data), metadata, and version ID.
- **Keys**:
    - Unique identifier for an object within a bucket.
- **Regions**:
    - Buckets are created in a specific AWS Region.

Storage Classes

- **Standard**:
    - Frequent access.
    - High durability (99.999999999% - 11 9s).
    - 99.99% availability.
- **Intelligent-Tiering**:
    - Automatically moves objects between access tiers.
    - Suitable for unknown or changing access patterns.
- **Standard-IA (Infrequent Access)**:
    - Lower cost for infrequent access.
    - Retrieval fee applies.
- **One Zone-IA**:
    - Single AZ storage.
    - Cheaper but less resilient.
- **Glacier**:

    - Archival storage.
    - Retrieval times from minutes to hours.
- **Glacier Deep Archive**:
    - Lowest-cost archival.
    - Retrieval within 12 hours.
- **Reduced Redundancy Storage (Deprecated)**:
    - Legacy, not recommended.

Durability and Availability

- **Durability**:
    - 99.999999999% (11 9s) across storage classes.
- **Availability**:
    - Standard: 99.99%
    - Standard-IA: 99.9%
    - One Zone-IA: 99.5%

S3 Object Versioning

- Preserves, retrieves, and restores every version of every object.
- Can be **enabled** or **suspended**.
- Helps recover from unintended overwrites and deletions.

S3 Encryption Options

- **Encryption In-Transit**:
    - HTTPS/TLS.
- **Encryption At Rest**:
    - SSE-S3 (AWS-managed keys).
    - SSE-KMS (AWS KMS-managed keys with audit).
    - SSE-C (Customer-provided keys).
    - Client-side encryption.

Access Control

- **Bucket Policies**:
    - JSON-based policies defining access permissions.
    - Applied at bucket level.
- **IAM Policies**:
    - User/role-based permissions.

- **ACLs (Access Control Lists)**:
    - Legacy, finer-grained control.
- **Block Public Access**:
    - Central settings to prevent public access.
    - Strongly recommended for securing buckets.

S3 Access Points

- Simplified way to manage access for shared datasets.
- Each access point has its own policy.
- Supports VPC restrictions for private access.

S3 Storage Lens

- Provides organization-wide visibility into storage usage and activity.
- Helps identify cost-saving opportunities and improve security posture.

S3 Event Notifications

- Can trigger actions when objects are created, deleted, etc.
- Supports:
    - SNS
    - SQS
    - Lambda

S3 Lifecycle Policies

- Automate object transitions between storage classes.
- Define expiration rules to delete objects automatically.

S3 Replication

- **Cross-Region Replication (CRR)**:
    - Automatically replicates objects to a bucket in another Region.
    - Useful for disaster recovery and compliance.
- **Same-Region Replication (SRR)**:
    - Replicates objects within the same Region.
    - Useful for log aggregation and latency reduction.
- Supports replication of new objects and existing objects via batch operations.

S3 Transfer Acceleration

- Speeds up uploads and downloads by routing through Amazon CloudFront edge locations.

- Suitable for geographically distributed users.

S3 Multipart Upload

- Splits large objects into parts for parallel upload.
- Recommended for objects larger than 100 MB.
- Required for objects over 5 GB.

S3 Select

- Retrieve a subset of data from an object using SQL expressions.
- Reduces the amount of data transferred.

S3 Object Lock

- Enforces **WORM (Write Once Read Many)** protection.
- Can set **retention periods** or **legal holds**.
- Used for compliance with regulations.

S3 Access Analyzer

- Identifies buckets with public or cross-account access.
- Helps remediate unintended access.

S3 Inventory

- Provides CSV reports listing objects and their metadata.
- Useful for auditing and compliance.

S3 Batch Operations

- Perform actions on millions or billions of objects.
- Supports copying, tagging, ACL updates, and Lambda invocations.

Security

- IAM Policies, Bucket Policies, ACLs.
- Encryption at rest and in transit.
- Block Public Access settings.
- VPC endpoints for private S3 access.
- CloudTrail logs for auditing API calls.

Pricing

- Charges based on:
    - Storage used (per GB/month).
    - Requests and data retrieval.
    - Data transfer out.

- Replication costs.
- Features like S3 Select and Inventory reports.

Integration with AWS Services

- AWS Lambda for event-driven processing.
- CloudFront for content delivery.
- Athena for querying data in S3.
- AWS Glue for ETL workflows.
- Amazon Macie for sensitive data discovery.
- DataSync for large-scale transfers.
- Snowball and Snowmobile for offline migration.

Exam Tips

- S3 is object storage, not block or file.
- Highly durable (11 9s), region-scoped buckets.
- Versioning enables protection against deletes/overwrites.
- Encryption can use SSE-S3, SSE-KMS, SSE-C, or client-side.
- Lifecycle rules automate class transitions and deletions.
- Replication supports CRR and SRR for DR and compliance.
- Block Public Access is critical for securing data.
- Access via IAM, bucket policies, access points, ACLs.
- Transfer Acceleration speeds up global transfers.
- Object Lock enables compliance with WORM requirements.
- Integration with Lambda, SNS, SQS for event notifications.

Quick Summary

Amazon S3 is AWS's highly durable, scalable, secure object storage service, supporting advanced features like lifecycle management, replication, encryption, event notifications, and integrations with a wide range of AWS services. It is essential for storing unstructured data in the cloud with fine-grained access controls and cost-effective storage classes.

AWS Storage Gateway

What It Is

AWS Storage Gateway is a hybrid cloud storage service that connects on-premises environments to AWS storage infrastructure.
It enables you to seamlessly integrate on-prem applications with cloud storage, supporting backup, archiving, disaster recovery, and tiered storage use cases.

Gateway Types

1. File Gateway (NFS / SMB)

- Provides file-based access to objects in Amazon S3.
- Interfaces with on-prem apps via NFS or SMB.
- Files are stored as objects in S3.
- Frequently accessed data cached locally for low-latency access.
- Supports Amazon S3 Object Lock for WORM (Write Once Read Many) compliance.
- Use Cases:
    - File server replacement
    - Backup storage
    - Data lake ingestion

2. Volume Gateway (iSCSI)

- Presents cloud-backed block storage volumes to on-prem servers via iSCSI.
- Two modes:
    - **Stored Volumes**:
        - Primary data is stored locally.
        - Asynchronous backups to AWS (EBS Snapshots).
        - Low-latency access with local storage.
    - **Cached Volumes**:
        - Primary data stored in AWS.
        - Frequently accessed data cached locally.
        - Reduces on-prem storage requirements.
- Use Cases:
    - Local apps with cloud backup
    - Disaster recovery

3. Tape Gateway (VTL)

- Presents itself as a virtual tape library (VTL) to backup applications.

- Compatible with most enterprise backup solutions.
- Virtual tapes stored in S3.
- Archived tapes stored in S3 Glacier or S3 Glacier Deep Archive.
- Eliminates need for physical tape infrastructure.
- Use Cases:
    - Backup/archive replacement
    - Long-term compliance storage

Deployment Options

- VMware ESXi, Microsoft Hyper-V, or Amazon EC2 (for cloud-based deployment).
- Requires local disk space for cache and upload buffer.
- Connects securely to AWS using HTTPS.
- Managed via AWS Storage Gateway Console or AWS CLI.

Security

- All data transferred is encrypted using TLS.
- Data at rest is encrypted using AWS Key Management Service (KMS).
- Supports IAM policies for access control.
- File Gateway supports Active Directory integration for SMB access control.

Monitoring & Management

- Integrated with:
    - Amazon CloudWatch for monitoring.
    - AWS CloudTrail for auditing API calls.
    - AWS Backup for managing backup policies (for Volume and Tape Gateway).
- Storage Gateway software and configuration updates are managed automatically.

Storage Integration

Gateway Type	Access Protocol	AWS Backend Storage	Caching	Use Case
File Gateway	NFS / SMB	Amazon S3	Local disk	File share, ingest to S3
Volume Gateway	iSCSI	EBS Snapshots (Stored/Cached)	Local (opt)	Block storage with backup
Tape Gateway	iSCSI (VTL)	S3 + Glacier	Local disk	Virtual tape backup/archive

Data Durability and Availability

- Leverages S3's **11 nines durability**.
- Virtual tapes in Glacier provide **long-term, low-cost archival**.
- Automatic replication and failover using AWS backend services.

Exam Tips

- File Gateway = Files to S3 (via NFS/SMB), good for S3 integration.
- Volume Gateway = Block storage over iSCSI, good for on-prem block apps with cloud backup.
- Tape Gateway = VTL to S3/Glacier, good for backup/archive use cases.
- Cached Volumes = Store in AWS, cache locally.
- Stored Volumes = Store locally, backup to AWS.
- Often tested in hybrid cloud, DR, and backup scenarios.
- Security = TLS + KMS, and IAM access control.
- AWS Backup can manage Volume and Tape Gateway backups.

Quick Summary

AWS Storage Gateway provides three gateway types: File, Volume, and Tape, to bridge on-premises environments with AWS cloud storage. It supports hybrid storage, backup, and archival use cases with local caching, encryption, and seamless AWS integration.