AWS Services Study Guide

Analytics, Cloud Financial Management, Compute, Containers & Database Services

Analytics Services

AWS Data Exchange

Purpose: Marketplace for finding, subscribing to, and using third-party data in the cloud.

Key Features:

- **Data Marketplace**: Browse and subscribe to data products from third-party providers
- **Data Delivery**: Automatic delivery of data updates to your S3 buckets
- **API Integration**: RESTful APIs for programmatic access
- **Billing Integration**: Usage-based billing through AWS

Data Types Available:

- Financial market data
- Weather and climate data
- Demographics and census data
- Healthcare and life sciences data
- Geospatial and mapping data

AI/ML Use Cases:

- Enriching training datasets with external data sources
- Real-time market data for financial ML models
- Weather data for demand forecasting models
- Demographics data for customer segmentation

Pricing Model: Pay-per-use or subscription-based, depending on data provider

Integration: Works seamlessly with S3, Lambda, SageMaker, and other AWS analytics services

---

Amazon EMR (Elastic MapReduce)

Purpose: Managed big data platform for processing large datasets using open-source tools.

Core Components:

- **Master Node**: Coordinates cluster activities and task distribution
- **Core Nodes**: Run tasks and store data in HDFS
- **Task Nodes**: Run tasks only (optional, for additional compute)

Supported Frameworks:

- **Apache Spark**: Fast, general-purpose distributed computing
- **Apache Hadoop**: Distributed storage and processing framework
- **Apache Hive**: Data warehouse software for querying large datasets
- **Apache HBase**: NoSQL database for real-time read/write access
- **Presto**: Distributed SQL query engine
- **Apache Zeppelin**: Web-based notebook for interactive analytics

Instance Types:

- **On-Demand**: Standard pricing, pay for what you use
- **Reserved**: 1-3 year commitments with significant savings
- **Spot**: Up to 90% savings using spare EC2 capacity

AI/ML Applications:

- **Data Preprocessing**: Clean and transform large datasets for ML
- **Feature Engineering**: Extract and create features from raw data
- **Model Training**: Distributed training of ML models using Spark MLlib
- **Batch Inference**: Process large datasets for predictions

Security Features:

- **Encryption**: At-rest and in-transit encryption
- **IAM Integration**: Fine-grained access control
- **VPC Support**: Deploy in private subnets
- **Kerberos**: Authentication for Hadoop ecosystem

Best Practices:

- Use appropriate instance types for workload characteristics
- Leverage spot instances for cost optimization
- Monitor cluster utilization and auto-scaling
- Use S3 for persistent data storage

AWS Glue

Purpose: Fully managed extract, transform, and load (ETL) service for preparing data for analytics.

Key Components:

Data Catalog

- **Metadata Repository**: Central repository for metadata about data sources
- **Schema Discovery**: Automatically discovers and catalogs data schemas
- **Table Definitions**: Stores table schemas, partition information, and data location
- **Integration**: Works with Athena, EMR, Redshift Spectrum, and SageMaker

ETL Jobs

- **Serverless**: No infrastructure to manage

- **Apache Spark**: Runs on managed Spark environment
- **Python/Scala**: Support for both programming languages
- **Visual ETL**: Drag-and-drop interface for creating ETL workflows

Crawlers

- **Auto-Discovery**: Automatically discover data sources and extract metadata
- **Schema Evolution**: Detect and handle schema changes over time
- **Scheduling**: Run on schedules or triggered by events
- **Supported Sources**: S3, RDS, Redshift, DynamoDB, and JDBC sources

DataBrew Integration

- **Visual Data Preparation**: GUI-based data cleaning and transformation
- **Recipe Management**: Reusable transformation recipes
- **Data Quality**: Built-in data quality checks and profiling

AI/ML Use Cases:

- **Data Pipeline Creation**: Build ETL pipelines to prepare training data
- **Feature Store**: Create and manage feature datasets for ML models
- **Data Quality**: Ensure data quality before feeding into ML workflows
- **Schema Management**: Handle evolving data schemas in ML pipelines

Pricing: Pay for resources used during ETL job execution and Data Catalog requests

Integration: Native integration with S3, Athena, QuickSight, SageMaker, and other AWS services

---

AWS Glue DataBrew

Purpose: Visual data preparation tool for cleaning and normalizing data without writing code.

Key Capabilities:

Visual Interface

- **Point-and-Click**: No coding required for data transformations
- **Recipe Builder**: Create reusable transformation recipes
- **Data Preview**: See transformation results in real-time
- **Profile Insights**: Automated data quality and statistical insights

Data Transformation Functions:**

- **Data Cleaning**: Remove duplicates, handle missing values, standardize formats
- **Data Normalization**: Standardize data formats and values
- **Data Enrichment**: Add calculated columns and derived metrics
- **Data Filtering**: Filter rows and columns based on conditions

Data Sources

- **S3**: Primary data source for files
- **Data Catalog**: Use Glue Data Catalog tables
- **Redshift**: Connect to Redshift tables
- **RDS**: Connect to relational databases

Profile Jobs:

- **Data Quality Assessment**: Identify data quality issues
- **Statistical Analysis**: Generate descriptive statistics
- **Data Distribution**: Understand data patterns and distributions
- **Anomaly Detection**: Identify outliers and unusual patterns

Recipe Jobs:

- **Batch Processing**: Apply transformations to entire datasets
- **Incremental Processing**: Process only new or changed data
- **Scheduling**: Automated job execution
- **Monitoring**: Track job performance and success rates

AI/ML Applications:

- **Data Preparation**: Prepare raw data for ML model training
- **Feature Engineering**: Create new features from existing data
- **Data Quality**: Ensure high-quality data for better model performance
- **Exploratory Data Analysis**: Understand data characteristics before modeling

Pricing: Pay for profiling and recipe job execution time

AWS Lake Formation

Purpose: Service to build, secure, and manage data lakes on AWS.

Core Capabilities:

Data Lake Creation

- **Centralized Setup**: Single service to set up data lakes
- **Data Ingestion**: Import data from various sources into S3
- **Metadata Management**: Automatically catalog and organize data
- **Schema Evolution**: Handle changing data schemas over time

Security and Governance

- **Fine-Grained Access Control**: Column and row-level security
- **Centralized Permissions**: Manage permissions across all data lake services
- **Audit Logging**: Track all data access and modifications
- **Data Classification**: Automatically classify sensitive data

Data Discovery and Access

- **Search Capabilities**: Find relevant datasets across the data lake
- **Self-Service Access**: Allow users to discover and access data independently
- **Query Integration**: Works with Athena, Redshift Spectrum, EMR, and QuickSight

Key Features:

- **Blueprints**: Pre-built workflows for common data ingestion patterns
- **Transactions**: ACID transactions for data consistency
- **Time Travel**: Query historical versions of data
- **Compaction**: Optimize storage and query performance

Integration with Analytics Services:

- **Amazon Athena**: Query data using SQL without moving it
- **Amazon Redshift**: Load and analyze data in data warehouse
- **Amazon EMR**: Process data using big data frameworks
- **Amazon QuickSight**: Create visualizations and dashboards

AI/ML Benefits:

- **Centralized Data Access**: Single point of access for all ML training data
- **Data Governance**: Ensure data quality and compliance for ML workflows
- **Feature Discovery**: Help data scientists find relevant features
- **Secure Data Sharing**: Share data across teams while maintaining security

Pricing: Pay for underlying AWS services used (S3, Glue, etc.) plus Lake Formation management costs

---

Amazon OpenSearch Service

Purpose: Managed search and analytics engine for log analytics, real-time application monitoring, and clickstream analytics.

Core Capabilities:

Search and Analytics

- **Full-Text Search**: Advanced search capabilities with relevance scoring
- **Real-Time Analytics**: Analyze data as it's ingested
- **Aggregations**: Complex data aggregations and grouping
- **Geospatial Search**: Location-based search and analytics

Data Ingestion

- **Multiple Sources**: Ingest from CloudWatch, Kinesis, S3, and other sources
- **Real-Time Streaming**: Process data streams in real-time
- **Batch Processing**: Load large datasets for analysis
- **Data Transformation**: Transform data during ingestion

Visualization Tools

- **OpenSearch Dashboards**: Built-in visualization and dashboard creation
- **Kibana Compatibility**: Works with existing Kibana dashboards
- **Custom Visualizations**: Create custom charts and graphs
- **Real-Time Monitoring**: Monitor metrics and logs in real-time

Cluster Architecture:

- **Master Nodes**: Manage cluster state and metadata
- **Data Nodes**: Store data and execute queries
- **Ingest Nodes**: Process incoming data before indexing
- **Coordinating Nodes**: Route requests and merge results

AI/ML Integration:

- **Anomaly Detection**: Built-in ML for detecting anomalies in time-series data
- **k-NN Search**: k-nearest neighbor search for similarity matching
- **Learning to Rank**: ML-powered search result ranking
- **Performance Analyzer**: ML-driven performance insights

Security Features:

- **Fine-Grained Access Control**: Control access at index and field level
- **Encryption**: At-rest and in-transit encryption
- **VPC Support**: Deploy in private networks
- **SAML and Active Directory**: Enterprise authentication integration

Use Cases:

- **Log Analytics**: Centralized logging and analysis
- **Application Monitoring**: Real-time application performance monitoring
- **Security Analytics**: Security event analysis and threat detection
- **Business Intelligence**: Search-driven analytics and reporting

Pricing: Pay for compute instances, storage, and data transfer

---

Amazon QuickSight

Purpose: Fast, cloud-powered business intelligence service for creating visualizations and dashboards.

Key Features:

Data Sources

- **AWS Services**: S3, Redshift, RDS, Athena, Aurora, DynamoDB
- **SaaS Applications**: Salesforce, ServiceNow, Jira, GitHub
- **Databases**: MySQL, PostgreSQL, SQL Server, Oracle, Teradata
- **Files**: Excel, CSV, JSON, Apache Parquet, Apache ORC

Visualization Types

- **Charts**: Bar, line, pie, scatter, heat maps, treemaps
- **Tables**: Pivot tables, data tables with conditional formatting
- **Maps**: Geospatial visualizations with location data
- **Custom Visuals**: Extend with custom visualization types

Advanced Analytics

- **Machine Learning Insights**: Automated insights using ML
- **Forecasting**: Time-series forecasting with seasonal patterns
- **Anomaly Detection**: Identify unusual patterns in data
- **Natural Language Queries**: Ask questions in plain English

SPICE Engine

- **In-Memory Processing**: Super-fast, Parallel, In-memory Calculation Engine
- **Columnar Storage**: Optimized for analytical queries
- **Automatic Scaling**: Scales to handle large datasets
- **Compression**: Efficient data compression for cost optimization

Editions:

- **Standard**: Basic BI capabilities with pay-per-session pricing
- **Enterprise**: Advanced features, hourly refresh, row-level security

Security and Governance:

- **Row-Level Security**: Control data access at the row level
- **Column-Level Security**: Hide sensitive columns from users
- **Active Directory Integration**: Enterprise authentication
- **Data Encryption**: Encrypt data at rest and in transit

AI/ML Dashboard Integration:

- **SageMaker Integration**: Visualize ML model results and metrics
- **Model Performance Monitoring**: Track model accuracy and drift
- **Feature Importance**: Visualize which features impact predictions
- **A/B Testing Results**: Compare model performance across experiments

Pricing Models:

- **Pay-per-Session**: Pay only when users access dashboards
- **Annual Subscription**: Fixed monthly cost per user
- **Embedded Analytics**: Custom pricing for embedding in applications

---

Amazon Redshift

Purpose: Fast, fully managed data warehouse for analyzing large datasets using SQL.

Architecture:

Cluster Components

- **Leader Node**: Coordinates query execution and communicates with client applications
- **Compute Nodes**: Store data and execute queries in parallel
- **Node Slices**: Each compute node divided into slices for parallel processing

Storage Types

- **Dense Storage (DS2)**: HDD-based for large data warehouses
- **Dense Compute (DC2)**: SSD-based for high-performance workloads
- **Redshift Spectrum**: Query data directly in S3 without loading

Performance Optimization:

Columnar Storage

- **Column-Oriented**: Store data by columns rather than rows
- **Compression**: Automatic compression reduces storage and I/O
- **Zone Maps**: Automatically created to skip irrelevant blocks

Parallel Processing

- **Massively Parallel**: Distribute queries across all compute nodes
- **Node-to-Node Communication**: Efficient data movement between nodes
- **Automatic Query Optimization**: Cost-based query optimizer

Advanced Features

- **Materialized Views**: Pre-computed query results for faster access
- **Result Caching**: Cache frequently accessed query results
- **Workload Management**: Prioritize and allocate resources to queries

Data Loading Methods:

- **COPY Command**: Bulk load data from S3, DynamoDB, or remote hosts
- **INSERT Statements**: Load small amounts of data
- **AWS Data Pipeline**: Orchestrate complex data loading workflows
- **AWS Glue**: ETL service integration for data preparation

Security Features:

- **Encryption**: At-rest encryption using AWS KMS or HSM
- **Network Isolation**: VPC deployment for network security
- **IAM Integration**: Fine-grained access control
- **Audit Logging**: Track user activity and query execution

AI/ML Integration:

- **Amazon SageMaker**: Train and deploy ML models using Redshift data
- **Redshift ML**: Create, train, and deploy ML models using SQL

- **Feature Engineering**: Use SQL for feature creation and transformation
- **Model Inference**: Apply trained models to new data in Redshift

Pricing Components:

- **Compute**: Hourly rates based on node type and quantity
- **Storage**: Additional storage beyond included amount
- **Data Transfer**: Cross-region and internet data transfer
- **Backup Storage**: Automated and manual snapshot storage

Cloud Financial Management

AWS Budgets

Purpose: Set custom cost and usage budgets to track AWS spending and usage.

Budget Types:

Cost Budgets

- **Track Spending**: Monitor actual and forecasted costs
- **Budget Periods**: Monthly, quarterly, or annual tracking
- **Granularity**: Account-level, service-level, or tag-based budgets
- **Currency Support**: Multiple currencies supported

Usage Budgets

- **Resource Utilization**: Track usage of specific AWS services
- **Unit Tracking**: Monitor hours, requests, or other service-specific units
- **Service Coverage**: EC2, RDS, S3, Lambda, and other services
- **Reserved Instance Utilization**: Track RI usage efficiency

Reservation Budgets

- **RI Coverage**: Monitor Reserved Instance coverage percentage
- **RI Utilization**: Track how well RIs are being used
- **Savings Plans**: Monitor Savings Plans utilization and coverage

Savings Plans Budgets

- **Coverage Tracking**: Monitor Savings Plans coverage
- **Utilization Monitoring**: Track Savings Plans utilization rates
- **Cost Savings**: Measure actual savings achieved

Alert Mechanisms:

- **Email Notifications**: Send alerts to specified email addresses
- **SNS Integration**: Trigger SNS topics for custom notifications
- **Threshold Types**: Actual spend, forecasted spend, or percentage thresholds

- **Multiple Thresholds**: Set up to 5 alert thresholds per budget

Advanced Features:

- **Filtering**: Filter by service, linked account, tag, or other dimensions
- **Time-Based Budgets**: Create budgets for specific time periods
- **Budget Reports**: Detailed spending reports and analysis
- **API Access**: Programmatic budget management and monitoring

AI/ML Cost Management:

- **SageMaker Training**: Budget for training job costs
- **Inference Endpoints**: Monitor real-time endpoint costs
- **Data Storage**: Track S3 costs for training datasets
- **Compute Resources**: Monitor EC2 costs for ML workloads

Best Practices:

- Start with broad budgets and refine over time
- Use tags for granular cost tracking
- Set up multiple alert thresholds
- Regular review and adjustment of budget amounts

Pricing: First 2 budgets free, then $0.02 per budget per day

AWS Cost Explorer

Purpose: Visualize, understand, and manage AWS costs and usage over time.

Core Capabilities:

Cost Visualization

- **Interactive Charts**: Customizable charts and graphs
- **Time Ranges**: View costs across different time periods
- **Granularity**: Daily, monthly, or yearly cost breakdowns
- **Service Breakdown**: Costs by AWS service
- **Account Breakdown**: Costs by linked accounts (for organizations)

Usage Analysis

- **Resource Utilization**: Analyze how resources are being used
- **Service Metrics**: Detailed usage metrics for each service
- **Trend Analysis**: Identify cost and usage trends over time
- **Comparative Analysis**: Compare costs across different periods

Filtering and Grouping

- **Dimension Filtering**: Filter by service, region, account, or tags
- **Group By Options**: Group costs by various dimensions

- **Tag-Based Analysis**: Analyze costs using resource tags
- **Custom Filters**: Create complex filtering criteria

Rightsizing Recommendations:

- **EC2 Rightsizing**: Identify over-provisioned EC2 instances
- **Cost Savings**: Potential savings from rightsizing
- **Performance Impact**: Assess impact of size changes
- **Implementation Guidance**: Step-by-step rightsizing instructions

Reserved Instance Recommendations:

- **Purchase Recommendations**: Suggest RI purchases for cost savings
- **Utilization Analysis**: Track existing RI utilization
- **Coverage Reports**: Monitor RI coverage across resources
- **ROI Calculations**: Calculate return on investment for RIs

Savings Plans Recommendations:

- **Commitment Recommendations**: Suggest optimal commitment levels
- **Savings Potential**: Calculate potential savings
- **Coverage Analysis**: Monitor Savings Plans coverage
- **Utilization Tracking**: Track Savings Plans utilization rates

Reporting Features:

- **Custom Reports**: Create and save custom cost reports
- **Scheduled Reports**: Automatically generated and delivered reports
- **CSV Export**: Export data for external analysis
- **API Access**: Programmatic access to cost and usage data

AI/ML Cost Analysis:

- **SageMaker Costs**: Analyze training, inference, and storage costs
- **Data Transfer Costs**: Monitor costs for data movement
- **Compute Optimization**: Identify opportunities for cost optimization
- **Resource Tagging**: Use tags to track ML project costs

Best Practices:

- Set up consistent tagging strategy for better cost attribution
- Regular review of recommendations and implementation
- Use grouping and filtering for detailed analysis
- Monitor cost trends and anomalies

Pricing: Basic Cost Explorer is free; advanced features have additional costs

---

Compute Services

Amazon EC2 (Elastic Compute Cloud)

Purpose: Resizable compute capacity in the cloud with complete control over computing resources.

Instance Categories:

General Purpose

- **A1**: ARM-based processors, cost-effective for scale-out workloads
- **M5/M5a/M5n**: Balanced compute, memory, and networking
- **M6i**: Latest generation with improved price-performance
- **T3/T4g**: Burstable performance with baseline CPU credits

Compute Optimized

- **C5/C5n**: High-performance processors for compute-intensive applications
- **C6i**: Latest generation with enhanced networking
- **C6g**: ARM-based Graviton2 processors for improved price-performance

Memory Optimized

- **R5/R5a/R5n**: High memory-to-vCPU ratio for memory-intensive applications
- **R6g**: ARM-based with large memory capacity
- **X1e**: Highest memory per vCPU ratio for in-memory databases
- **z1d**: High-frequency processors with NVMe SSD storage

Storage Optimized

- **I3/I3en**: NVMe SSD-backed instance storage for high I/O performance
- **D2**: Dense HDD storage for distributed file systems
- **H1**: High disk throughput with 16 TB of HDD storage per instance

Accelerated Computing

- **P3/P4**: GPU instances for machine learning and high-performance computing
- **G4**: GPU instances for graphics-intensive applications
- **F1**: FPGA instances for hardware acceleration
- **Inf1**: Machine learning inference with AWS Inferentia chips

Purchasing Options:

On-Demand Instances

- **Pay-as-you-go**: No upfront costs or long-term commitments
- **Flexibility**: Start and stop instances as needed
- **Use Cases**: Variable workloads, development/testing, short-term needs

Reserved Instances

- **Commitment**: 1 or 3-year terms with significant savings (up to 75%)
- **Types**: Standard (highest savings), Convertible (flexibility), Scheduled

- **Payment Options**: All upfront, partial upfront, or no upfront

Spot Instances

- **Cost Savings**: Up to 90% savings using spare EC2 capacity
- **Interruption**: Can be terminated with 2-minute notice
- **Use Cases**: Fault-tolerant, flexible applications, batch processing

Dedicated Hosts

- **Physical Servers**: Dedicated physical servers for compliance requirements
- **License Optimization**: Bring your own licenses (BYOL)
- **Compliance**: Meet regulatory requirements for dedicated hardware

Storage Options:

Elastic Block Store (EBS)

- **General Purpose SSD (gp3)**: Balanced price and performance
- **Provisioned IOPS SSD (io2)**: High-performance for critical workloads
- **Throughput Optimized HDD (st1)**: Low-cost HDD for throughput-intensive workloads
- **Cold HDD (sc1)**: Lowest cost HDD for infrequently accessed data

Instance Store

- **Temporary Storage**: Storage physically attached to host computer
- **High Performance**: Very high I/O performance
- **Ephemeral**: Data lost when instance stops or terminates

Networking Features:

- **Enhanced Networking**: SR-IOV for high-bandwidth, low-latency networking
- **Placement Groups**: Control instance placement for optimal performance
- **Elastic Network Interfaces**: Multiple network interfaces per instance
- **IPv6 Support**: Dual-stack IPv4 and IPv6 addressing

AI/ML Specific Considerations:

GPU Instances (P3, P4, G4)

- **Deep Learning**: Optimized for training and inference workloads
- **CUDA Support**: NVIDIA GPU computing platform
- **Multi-GPU**: Multiple GPUs per instance for distributed training
- **Memory**: High memory capacity for large models and datasets

CPU Instances for ML

- **Compute Optimized**: C5 instances for CPU-intensive ML workloads
- **Memory Optimized**: R5 instances for large dataset processing
- **Inference Optimization**: M5 instances for cost-effective inference

Security Features:

- **Security Groups**: Virtual firewalls controlling inbound/outbound traffic
- **Key Pairs**: Secure SSH access using public-key cryptography
- **IAM Roles**: Secure access to AWS services without hardcoded credentials
- **Encryption**: EBS encryption and in-transit encryption options

Monitoring and Management:

- **CloudWatch**: Detailed monitoring and alerting
- **Systems Manager**: Patch management and remote access
- **Auto Scaling**: Automatically adjust capacity based on demand
- **Load Balancing**: Distribute traffic across multiple instances

Best Practices:

- Choose appropriate instance types based on workload characteristics
- Use Auto Scaling for variable workloads
- Implement proper security groups and IAM policies
- Monitor performance and costs regularly
- Use appropriate storage types for different data access patterns

Pricing Factors:

- Instance type and size
- Purchasing option (On-Demand, Reserved, Spot)
- Operating system
- Region and Availability Zone
- Data transfer
- Additional features (monitoring, storage, etc.)

Container Services

Amazon Elastic Container Service (Amazon ECS)

Purpose: Fully managed container orchestration service for running Docker containers at scale.

Core Components:

Clusters

- **Logical Grouping**: Group of EC2 instances or Fargate tasks
- **Resource Management**: Manages compute resources for containers
- **Scaling**: Automatic scaling based on resource needs
- **Multi-AZ**: Deploy across multiple Availability Zones for high availability

Task Definitions

- **Container Blueprint**: JSON document describing container configuration
- **Container Specifications**: CPU, memory, port mappings, environment variables

- **Networking Mode**: Bridge, host, awsvpc, or none
- **Storage**: Volume mounts and storage configuration

Services

- **Desired State**: Maintain specified number of running tasks
- **Load Balancing**: Integration with Application Load Balancer and Network Load Balancer
- **Service Discovery**: Automatic service registration and discovery
- **Rolling Updates**: Zero-downtime deployments with rolling updates

Tasks

- **Running Containers**: One or more containers running together
- **Placement**: Automatic placement across cluster resources
- **Health Monitoring**: Automatic restart of unhealthy tasks
- **Resource Allocation**: CPU and memory allocation per task

Launch Types:

EC2 Launch Type

- **EC2 Instances**: Run containers on managed EC2 instances
- **Instance Management**: You manage EC2 instances in the cluster
- **Cost Control**: More granular control over compute costs
- **Customization**: Custom AMIs and instance configurations
- **Use Cases**: Long-running applications, cost optimization, custom requirements

Fargate Launch Type

- **Serverless**: No EC2 instances to manage
- **Task-Level Isolation**: Each task runs in its own kernel runtime environment
- **Automatic Scaling**: Scales individual tasks based on CPU and memory requirements
- **Pay-per-Task**: Pay only for the resources your tasks consume
- **Use Cases**: Microservices, batch jobs, simplicity over cost optimization

Networking:

- **VPC Integration**: Deploy containers in VPC with security groups
- **Service Mesh**: Support for AWS App Mesh for microservices communication
- **Load Balancing**: Integration with Elastic Load Balancing
- **Service Discovery**: Cloud Map integration for service discovery

Security Features:

- **IAM Roles**: Task-level IAM roles for secure access to AWS services
- **Secrets Management**: Integration with AWS Secrets Manager and Systems Manager Parameter Store
- **Network Isolation**: VPC and security group isolation
- **Image Scanning**: ECR image vulnerability scanning

AI/ML Use Cases:

- **Model Serving**: Deploy ML models as containerized microservices
- **Batch Processing**: Run ML training jobs using containers
- **Data Pipeline**: Containerized data processing workflows
- **Multi-Model Serving**: Deploy multiple models with different resource requirements

Integration with AI/ML Services:

- **SageMaker**: Use ECS for custom model serving endpoints
- **Batch Processing**: Container-based ML training pipelines
- **Real-time Inference**: Scalable model serving with load balancing
- **Data Processing**: ETL jobs using containerized applications

Monitoring and Logging:

- **CloudWatch**: Container and service-level monitoring
- **X-Ray**: Distributed tracing for containerized applications
- **Container Insights**: Detailed container performance metrics
- **Log Drivers**: Multiple log drivers including CloudWatch Logs

Best Practices:

- Use Fargate for simplicity, EC2 for cost optimization
- Implement proper health checks for containers
- Use service discovery for microservices communication
- Implement proper security with IAM roles and secrets management
- Monitor container performance and resource utilization

Pricing:

- **EC2 Launch Type**: Pay for EC2 instances and EBS volumes
- **Fargate Launch Type**: Pay for vCPU and memory resources consumed by tasks

---

Amazon Elastic Kubernetes Service (Amazon EKS)

Purpose: Managed Kubernetes service for running Kubernetes applications without needing to install and operate Kubernetes clusters.

Core Components:

Control Plane

- **Managed Masters**: AWS manages Kubernetes master nodes
- **High Availability**: Multi-AZ deployment of master nodes
- **Patching**: Automatic security patching and updates
- **API Server**: Kubernetes API server with AWS authentication integration

Worker Nodes

- **Node Groups**: Managed groups of EC2 instances running kubelet
- **Fargate**: Serverless option for running pods

- **Spot Instances**: Cost optimization using Spot instances in node groups
- **Auto Scaling**: Automatic scaling of worker nodes based on demand

Networking

- **VPC Integration**: Native VPC networking for pods
- **CNI Plugin**: AWS VPC CNI plugin for pod networking
- **Load Balancing**: Integration with AWS Load Balancers
- **Ingress Controllers**: Support for various ingress controllers

Node Types:

Managed Node Groups

- **Simplified Management**: AWS manages EC2 instances lifecycle
- **Auto Scaling**: Built-in cluster autoscaler support
- **Rolling Updates**: Automatic rolling updates for worker nodes
- **Instance Types**: Support for various EC2 instance types including GPU instances

Self-Managed Nodes

- **Full Control**: Complete control over EC2 instances
- **Custom Configuration**: Custom AMIs and instance configurations
- **Cost Optimization**: Fine-grained cost control and optimization options
- **Advanced Networking**: Custom networking configurations

Fargate

- **Serverless Pods**: Run pods without managing EC2 instances
- **Pod-Level Isolation**: Each pod runs in its own compute environment
- **Automatic Scaling**: Scales pods based on resource requirements
- **Pay-per-Pod**: Pay only for the resources your pods consume

Kubernetes Features:

- **Latest Versions**: Support for latest Kubernetes versions
- **Native APIs**: Full access to Kubernetes APIs and features
- **Helm**: Native support for Helm package manager
- **Operators**: Support for Kubernetes operators
- **Custom Resources**: Full support for custom resource definitions

Security Features:

- **IAM Integration**: Kubernetes RBAC with AWS IAM
- **Pod Security**: Pod security policies and security contexts
- **Network Policies**: Kubernetes network policies for micro-segmentation
- **Secrets Management**: Integration with AWS Secrets Manager
- **Image Scanning**: ECR integration with vulnerability scanning

AI/ML Capabilities:

Machine Learning Workloads

- **GPU Support**: P3, P4, and G4 instances for ML training and inference
- **Kubeflow**: Support for Kubeflow ML pipelines
- **Distributed Training**: Multi-node distributed training using MPI and Horovod
- **Jupyter Notebooks**: JupyterHub deployment for data science workflows

ML Operators

- **TensorFlow Operator**: Distributed TensorFlow training
- **PyTorch Operator**: Distributed PyTorch training
- **MPI Operator**: Message Passing Interface for distributed computing
- **Argo Workflows**: ML pipeline orchestration

Integration with AWS AI Services

- **SageMaker**: Use EKS for custom training and inference
- **Batch Processing**: GPU-accelerated batch ML jobs
- **Model Serving**: Scalable model serving with Kubernetes services
- **Data Processing**: Spark on Kubernetes for big data processing

Add-ons and Extensions:

- **AWS Load Balancer Controller**: Advanced load balancing features
- **EBS CSI Driver**: Persistent storage using EBS volumes
- **EFS CSI Driver**: Shared storage using EFS
- **Cluster Autoscaler**: Automatic node scaling based on pod demand
- **AWS App Mesh**: Service mesh for microservices

Monitoring and Observability:

- **CloudWatch Container Insights**: Detailed container and cluster metrics
- **Prometheus**: Native Prometheus support for monitoring
- **Grafana**: Integration with Grafana for visualization
- **X-Ray**: Distributed tracing for applications
- **Fluentd**: Log aggregation and forwarding

Best Practices:

- Use managed node groups for simplicity
- Implement proper RBAC and security policies
- Use appropriate instance types for workloads (GPU for ML)
- Monitor cluster and application performance
- Implement proper resource requests and limits
- Use namespaces for resource isolation

Pricing Components:

- **Control Plane**: Fixed hourly charge per cluster
- **Worker Nodes**: EC2 instance costs (On-Demand, Reserved, or Spot)

- **Fargate**: Pay for vCPU and memory consumed by pods
- **Data Transfer**: Standard AWS data transfer charges

Database Services

Amazon DocumentDB (with MongoDB compatibility)

Purpose: Fully managed document database service that supports MongoDB workloads with enterprise-grade features.

Key Features:

MongoDB Compatibility

- **API Compatibility**: Compatible with MongoDB 3.6, 4.0, and 5.0 APIs
- **Drivers**: Works with existing MongoDB drivers and tools
- **Query Language**: Support for MongoDB query language and operations
- **Indexing**: MongoDB-style indexing with compound and text indexes

Architecture

- **Cluster-Based**: Deployed as clusters with primary and replica instances
- **Storage**: Distributed, fault-tolerant storage that automatically scales
- **Compute**: Separate compute and storage for independent scaling
- **Multi-AZ**: Automatic replication across multiple Availability Zones

Performance Features

- **Read Replicas**: Up to 15 read replicas for read scaling
- **Connection Pooling**: Built-in connection pooling for better performance
- **Caching**: Query result caching for frequently accessed data
- **Parallel Query**: Parallel execution for complex aggregation queries

Security and Compliance:

- **Encryption**: At-rest encryption using AWS KMS
- **VPC Isolation**: Deploy in VPC for network isolation
- **IAM Authentication**: Integration with AWS IAM for authentication
- **Audit Logging**: Comprehensive audit logging for compliance

AI/ML Use Cases:

- **Feature Store**: Store and retrieve ML features as documents
- **Model Metadata**: Store model configurations, parameters, and metadata
- **Training Data**: Store unstructured training data and annotations
- **Real-time Personalization**: Store user profiles and preferences for recommendation engines
- **Content Management**: Store and index documents for NLP applications

Backup and Recovery:

- **Continuous Backup**: Point-in-time recovery up to 35 days
- **Automated Snapshots**: Daily automated backups
- **Manual Snapshots**: On-demand backup creation
- **Cross-Region Backup**: Copy backups to different regions

Monitoring:

- **CloudWatch Integration**: Built-in metrics and monitoring
- **Performance Insights**: Database performance monitoring
- **Slow Query Logs**: Identify and optimize slow queries
- **Connection Monitoring**: Track database connections and usage

Pricing: Pay for compute instances, storage consumed, I/O operations, and backup storage

---

Amazon DynamoDB

Purpose: Fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.

Core Characteristics:

Serverless Architecture

- **No Server Management**: Fully managed service with no servers to provision
- **Automatic Scaling**: Automatically scales up and down based on traffic
- **Pay-per-Use**: Pay only for the resources you consume
- **Global Distribution**: Multi-region, multi-active replication

Performance Features

- **Single-Digit Millisecond Latency**: Consistent fast performance
- **Unlimited Storage**: Virtually unlimited storage capacity
- **High Throughput**: Handle millions of requests per second
- **Burst Capacity**: Temporary burst above provisioned capacity

Data Model:

Tables and Items

- **Tables**: Top-level container for data (similar to tables in relational databases)
- **Items**: Individual records in tables (similar to rows)
- **Attributes**: Name-value pairs within items (similar to columns)
- **Schema-less**: No fixed schema, items can have different attributes

Primary Keys

- **Partition Key**: Single attribute that determines data distribution
- **Composite Key**: Partition key + sort key for more complex queries
- **Hash Distribution**: Uses partition key to distribute data across partitions

Secondary Indexes

- **Global Secondary Index (GSI)**: Different partition and sort key from base table
- **Local Secondary Index (LSI)**: Same partition key, different sort key
- **Query Flexibility**: Enable additional query patterns beyond primary key

Capacity Modes:

On-Demand Mode

- **Pay-per-Request**: Pay for actual read/write requests
- **Automatic Scaling**: Instantly scales to handle traffic spikes
- **No Capacity Planning**: No need to specify capacity requirements
- **Use Cases**: Unpredictable workloads, new applications

Provisioned Mode

- **Pre-specified Capacity**: Define read and write capacity units
- **Auto Scaling**: Optional auto scaling based on utilization
- **Reserved Capacity**: Purchase reserved capacity for cost savings
- **Use Cases**: Predictable workloads, cost optimization

Advanced Features:

DynamoDB Streams

- **Change Data Capture**: Capture changes to items in DynamoDB tables
- **Real-time Processing**: Process changes in near real-time
- **Lambda Integration**: Trigger Lambda functions on data changes
- **Use Cases**: Audit trails, replication, analytics

Global Tables

- **Multi-Region Replication**: Automatically replicate data across regions
- **Multi-Active**: Read and write from any region
- **Eventual Consistency**: Eventual consistency across regions
- **Conflict Resolution**: Last-writer-wins conflict resolution

DynamoDB Accelerator (DAX)

- **In-Memory Caching**: Microsecond latency for cached data
- **Fully Managed**: No cache infrastructure to manage
- **Write-Through**: Automatic cache updates on writes
- **API Compatibility**: No application changes required

Security Features:

- **Encryption at Rest**: Automatic encryption using AWS KMS
- **Encryption in Transit**: All data encrypted in transit

- **IAM Integration**: Fine-grained access control using IAM
- **VPC Endpoints**: Private connectivity from VPC

AI/ML Applications:

- **Feature Store**: Real-time feature serving for ML models
- **Session Store**: Store user sessions for personalization
- **Recommendation Engines**: Store user preferences and item catalogs
- **Real-time Analytics**: Store and query real-time metrics
- **IoT Data**: Store sensor data and device states
- **Gaming**: Player profiles, leaderboards, and game state

Data Import/Export:

- **DynamoDB Import**: Import data from S3 into DynamoDB
- **DynamoDB Export**: Export table data to S3 for analytics
- **AWS Data Pipeline**: ETL workflows for data migration
- **Point-in-Time Recovery**: Restore to any point in the last 35 days

Best Practices:

- Design efficient partition keys for even data distribution
- Use sparse indexes to reduce costs
- Implement proper error handling and retry logic
- Monitor consumed capacity and throttling
- Use batch operations for bulk data operations

Pricing Components:

- **Read/Write Capacity**: Provisioned or on-demand capacity
- **Storage**: Per GB of data stored
- **Features**: Global Tables, Streams, Backups, and other features
- **Data Transfer**: Cross-region replication and internet data transfer

---

Amazon ElastiCache

Purpose: Fully managed in-memory caching service supporting Redis and Memcached engines.

Supported Engines:

Redis

- **Data Structures**: Strings, hashes, lists, sets, sorted sets, bitmaps, hyperloglogs
- **Persistence**: Optional data persistence to disk
- **Replication**: Master-replica replication for high availability
- **Clustering**: Redis Cluster for horizontal scaling
- **Pub/Sub**: Message publishing and subscription capabilities
- **Lua Scripting**: Server-side scripting support

Memcached

- **Simple Caching**: Key-value caching with simple data types
- **Multi-threading**: Multi-threaded architecture for high performance
- **Horizontal Scaling**: Easy horizontal scaling by adding nodes
- **No Persistence**: Pure in-memory caching without persistence
- **Protocol**: Simple text-based protocol

Architecture Options:

Redis Cluster Mode

- **Sharding**: Automatic data sharding across multiple nodes
- **Scaling**: Scale read and write capacity independently
- **High Availability**: Multi-AZ replication with automatic failover
- **Cross-Region**: Cross-region replication for disaster recovery

Redis Non-Cluster Mode

- **Single Shard**: One primary node with optional read replicas
- **Simpler Architecture**: Easier setup and management
- **Vertical Scaling**: Scale by changing node types
- **Use Cases**: Simple caching scenarios, session store

Memcached

- **Node-based**: Add or remove nodes to scale horizontally
- **No Replication**: No built-in replication or persistence
- **Auto Discovery**: Automatic discovery of cache nodes
- **Use Cases**: Simple caching, web application acceleration

Performance Features:

- **Sub-millisecond Latency**: Extremely fast data access
- **High Throughput**: Handle millions of operations per second
- **Memory Optimization**: Efficient memory usage and management
- **Connection Pooling**: Built-in connection pooling for better performance

Security Features:

- **VPC Deployment**: Deploy in VPC for network isolation
- **Security Groups**: Control network access using security groups
- **Encryption**: At-rest and in-transit encryption (Redis only)
- **AUTH**: Redis AUTH for authentication (Redis only)
- **IAM Authentication**: IAM-based authentication for Redis

AI/ML Use Cases:

Model Serving

- **Model Cache**: Cache trained models for fast inference
- **Feature Cache**: Cache computed features for real-time predictions

- **Result Cache**: Cache prediction results for frequently requested items
- **Session Storage**: Store user session data for personalization

Real-time Analytics

- **Metrics Storage**: Store real-time metrics and counters
- **Leaderboards**: Maintain sorted sets for ranking and scoring
- **Time-series Data**: Store time-based metrics using sorted sets
- **Event Tracking**: Track user events and behaviors

Data Pipeline Acceleration

- **ETL Caching**: Cache intermediate results in data processing pipelines
- **Database Offloading**: Reduce database load by caching frequent queries
- **API Response Caching**: Cache API responses for better performance
- **Content Delivery**: Cache static and dynamic content

Monitoring and Management:

- **CloudWatch Integration**: Built-in metrics and alarms
- **Parameter Groups**: Manage engine configuration parameters
- **Maintenance Windows**: Scheduled maintenance for updates
- **Notifications**: SNS notifications for cluster events

Backup and Recovery:

- **Automated Backups**: Daily automated backups (Redis only)
- **Manual Snapshots**: On-demand backup creation (Redis only)
- **Point-in-Time Recovery**: Restore to specific points in time (Redis only)
- **Cross-Region Backup**: Copy backups across regions (Redis only)

Best Practices:

- Choose appropriate engine based on use case requirements
- Implement proper key expiration policies
- Monitor memory usage and eviction policies
- Use appropriate node types for workload characteristics
- Implement connection pooling in applications

Pricing: Pay for cache node hours, data transfer, and backup storage

---

Amazon MemoryDB

Purpose: Redis-compatible, durable, in-memory database service for ultra-fast performance with data durability.

Key Differentiators from ElastiCache:

Durability

- **Multi-AZ Durability**: Data persisted across multiple Availability Zones
- **Transaction Log**: Distributed transaction log for data durability
- **ACID Compliance**: Strong consistency and ACID transaction support
- **Primary Database**: Can serve as primary database, not just cache

Performance

- **Microsecond Latency**: Sub-millisecond read latency
- **High Throughput**: Over 160 million requests per second per cluster
- **Fast Recovery**: Quick recovery from node failures
- **Consistent Performance**: Predictable performance under load

Architecture:

Cluster Architecture

- **Sharded Clusters**: Data automatically sharded across multiple nodes
- **Primary and Replica Nodes**: Each shard has one primary and up to 5 replicas
- **Automatic Failover**: Fast failover to replica nodes
- **Cross-AZ Replication**: Data replicated across Availability Zones

Storage Engine

- **Memory-First**: Data stored in memory for fast access
- **Durable Storage**: Data also written to distributed transaction log
- **Snapshot and Recovery**: Point-in-time snapshots for recovery
- **Data Tiering**: Automatic data tiering based on access patterns

Redis Compatibility:

- **Redis API**: Compatible with Redis 6.2 and later
- **Data Structures**: Full support for Redis data structures
- **Commands**: Support for most Redis commands
- **Existing Applications**: Drop-in replacement for Redis applications

Security Features:

- **Encryption**: Encryption at rest and in transit
- **VPC Integration**: Deploy in VPC with security groups
- **IAM Authentication**: Integration with AWS IAM
- **Access Control Lists**: Fine-grained access control
- **Audit Logging**: Comprehensive audit logging

AI/ML Applications:

Real-time Feature Store

- **Durable Features**: Store ML features with guaranteed durability
- **Fast Retrieval**: Sub-millisecond feature retrieval for inference
- **Feature Updates**: Real-time feature updates with consistency

- **Historical Features**: Maintain historical feature values

Session and State Management

- **User Sessions**: Durable user session storage
- **Application State**: Maintain application state across instances
- **Shopping Carts**: Persistent shopping cart data
- **Personalization State**: Store user preferences and recommendations

Real-time Analytics

- **Metrics and Counters**: Durable storage of real-time metrics
- **Leaderboards**: Persistent leaderboards and rankings
- **Event Processing**: Real-time event processing and aggregation
- **Fraud Detection**: Store and analyze transaction patterns

Monitoring and Operations:

- **CloudWatch Integration**: Built-in monitoring and alerting
- **Parameter Groups**: Manage database configuration
- **Maintenance Windows**: Automated maintenance and updates
- **Performance Insights**: Detailed performance monitoring

Backup and Recovery:

- **Continuous Backup**: Continuous backup with point-in-time recovery
- **Automated Snapshots**: Daily automated snapshots
- **Manual Snapshots**: On-demand snapshot creation
- **Cross-Region Backup**: Backup replication across regions

Use Cases:

- Primary database for applications requiring fast access with durability
- Real-time recommendation engines
- Gaming leaderboards and player data
- Financial services applications requiring low latency and durability
- IoT applications with high-frequency data updates

Pricing: Pay for compute, storage, and data transfer

Amazon Neptune

Purpose: Fully managed graph database service optimized for storing billions of relationships and querying the graph with milliseconds latency.

Graph Database Concepts:

Graph Elements

- **Vertices (Nodes)**: Represent entities in the graph

- **Edges (Relationships)**: Represent relationships between vertices
- **Properties**: Key-value pairs attached to vertices and edges
- **Labels**: Categorize vertices and edges

Graph Models

- **Property Graph**: Vertices and edges with properties and labels
- **RDF (Resource Description Framework)**: Subject-predicate-object triples
- **Mixed Workloads**: Support both models in same database cluster

Query Languages:

Apache TinkerPop Gremlin

- **Graph Traversal Language**: Query property graphs using traversals
- **Step-based Queries**: Chain steps to navigate graph structure
- **Complex Patterns**: Express complex graph patterns and algorithms
- **Real-time Queries**: Interactive graph exploration and analysis

SPARQL

- **RDF Query Language**: Query RDF graphs using SPARQL syntax
- **Federated Queries**: Query across multiple RDF datasets
- **Inference Support**: Built-in reasoning and inference capabilities
- **Standards Compliance**: W3C standard for semantic web applications

openCypher

- **Declarative Query Language**: SQL-like syntax for graph queries
- **Pattern Matching**: Express graph patterns using ASCII art syntax
- **Aggregations**: Group and aggregate graph data
- **Neo4j Compatibility**: Compatible with Neo4j Cypher queries

Architecture:

Cluster Configuration

- **Primary Instance**: Handles read and write operations
- **Read Replicas**: Up to 15 read replicas for read scaling
- **Multi-AZ**: Automatic failover across Availability Zones
- **Storage**: Distributed, fault-tolerant storage layer

Performance Features

- **Fast Queries**: Optimized for graph traversal operations
- **Parallel Processing**: Parallel execution of complex queries
- **Caching**: Multi-level caching for frequently accessed data
- **Index Optimization**: Automatic index creation and optimization

Security Features:

- **VPC Integration**: Deploy in VPC with security groups
- **IAM Authentication**: Integration with AWS IAM for authentication
- **Encryption**: At-rest and in-transit encryption
- **Audit Logging**: Comprehensive audit logging for compliance
- **Fine-grained Access**: Control access to specific graph elements

AI/ML Applications:

Knowledge Graphs

- **Entity Resolution**: Link and resolve entities across datasets
- **Semantic Search**: Enhanced search using graph relationships
- **Content Recommendation**: Graph-based recommendation algorithms
- **Question Answering**: Knowledge graph-powered Q&A systems

Fraud Detection

- **Transaction Networks**: Model financial transactions as graphs
- **Pattern Detection**: Identify suspicious transaction patterns
- **Entity Linking**: Connect related entities across different data sources
- **Risk Scoring**: Calculate risk scores based on graph structure

Social Network Analysis

- **Social Graphs**: Model social relationships and interactions
- **Influence Analysis**: Identify influential users and communities
- **Recommendation Systems**: Friend and content recommendations
- **Community Detection**: Discover communities and clusters

Machine Learning Graph Applications

- **Graph Neural Networks**: Store graphs for GNN training and inference
- **Feature Engineering**: Extract graph-based features for ML models
- **Network Analysis**: Analyze network topology and structure
- **Link Prediction**: Predict missing or future relationships

Data Loading:

- **Bulk Loading**: Efficient bulk loading from S3
- **Real-time Loading**: Stream data using Kinesis or Lambda
- **Format Support**: Support for various graph data formats
- **ETL Integration**: Integration with AWS Glue for data transformation

Analytics and Algorithms:

- **Graph Algorithms**: Built-in graph algorithms (PageRank, centrality, etc.)
- **OLAP Queries**: Online analytical processing for graph data
- **Machine Learning**: Integration with SageMaker for graph ML

- **Visualization**: Integration with graph visualization tools

Monitoring and Management:

- **CloudWatch Integration**: Built-in metrics and monitoring
- **Query Performance**: Query execution plan analysis
- **Resource Utilization**: Monitor CPU, memory, and storage usage
- **Slow Query Logs**: Identify and optimize slow queries

Backup and Recovery:

- **Continuous Backup**: Point-in-time recovery up to 35 days
- **Automated Snapshots**: Daily automated backups
- **Manual Snapshots**: On-demand backup creation
- **Cross-Region Backup**: Copy backups to different regions

Best Practices:

- Design graph schema for efficient traversals
- Use appropriate indexes for query patterns
- Implement proper data partitioning strategies
- Monitor query performance and optimize accordingly
- Use read replicas for read-heavy workloads

Pricing: Pay for compute instances, storage consumed, I/O operations, and backup storage

---

Amazon RDS (Relational Database Service)

Purpose: Managed relational database service supporting multiple database engines with automated management features.

Supported Database Engines:

Amazon Aurora

- **MySQL Compatible**: Up to 5x performance of standard MySQL
- **PostgreSQL Compatible**: Up to 3x performance of standard PostgreSQL
- **Serverless**: On-demand, auto-scaling configuration
- **Global Database**: Cross-region replication with low latency
- **Multi-Master**: Multiple write instances for high availability

MySQL

- **Community Edition**: Open-source MySQL with AWS management
- **Version Support**: Multiple MySQL versions supported
- **Storage Engine**: InnoDB storage engine optimized for AWS
- **Replication**: Master-slave replication for read scaling

PostgreSQL

- **Open Source**: Full-featured PostgreSQL with AWS management
- **Extensions**: Support for PostgreSQL extensions
- **JSON Support**: Native JSON data type and operations
- **Advanced Features**: Window functions, CTEs, and advanced SQL features

MariaDB

- **MySQL Fork**: Enhanced MySQL-compatible database
- **Performance**: Improved performance over MySQL
- **Storage Engines**: Multiple storage engine options
- **Compatibility**: High compatibility with MySQL applications

Oracle Database

- **Enterprise Features**: Full Oracle Database feature set
- **License Options**: Bring Your Own License (BYOL) or License Included
- **Version Support**: Multiple Oracle versions supported
- **Migration Tools**: AWS Database Migration Service integration

Microsoft SQL Server

- **Multiple Editions**: Express, Web, Standard, and Enterprise editions
- **Windows Authentication**: Integration with Active Directory
- **Always On**: High availability and disaster recovery features
- **Business Intelligence**: Integration with SQL Server BI tools

Deployment Options:

Single-AZ Deployment

- **Basic Configuration**: Single database instance in one AZ
- **Cost-Effective**: Lower cost for development and testing
- **No Automatic Failover**: Manual intervention required for failures
- **Use Cases**: Development, testing, non-critical applications

Multi-AZ Deployment

- **High Availability**: Synchronous replication to standby instance
- **Automatic Failover**: Automatic failover in case of primary failure
- **Zero Downtime**: Minimal downtime during maintenance
- **Use Cases**: Production applications requiring high availability

Read Replicas

- **Read Scaling**: Offload read traffic from primary instance
- **Cross-Region**: Create read replicas in different regions
- **Asynchronous Replication**: Eventually consistent read replicas
- **Use Cases**: Read-heavy workloads, reporting, analytics

Instance Classes:

General Purpose (db.t3, db.m5)

- **Balanced Performance**: Balanced CPU, memory, and network
- **Burstable Performance**: T3 instances with CPU credits
- **Use Cases**: Most database workloads, development/testing

Memory Optimized (db.r5, db.x1e)

- **High Memory**: High memory-to-vCPU ratio
- **Performance**: Optimized for memory-intensive applications
- **Use Cases**: In-memory databases, real-time analytics

Compute Optimized (db.c5)

- **High CPU**: High-performance processors
- **Compute Intensive**: Optimized for CPU-intensive workloads
- **Use Cases**: High-performance web servers, scientific computing

Storage Types:

General Purpose SSD (gp2)

- **Balanced Performance**: 3 IOPS per GB, burstable to 3,000 IOPS
- **Cost-Effective**: Good price-performance ratio
- **Use Cases**: Most database workloads

Provisioned IOPS SSD (io1)

- **High Performance**: Up to 64,000 IOPS per volume
- **Consistent Performance**: Predictable IOPS performance
- **Use Cases**: I/O-intensive applications, critical databases

Magnetic Storage

- **Low Cost**: Lowest cost storage option
- **Legacy Support**: For backward compatibility
- **Use Cases**: Light usage, infrequent access

AI/ML Integration:

Data Source for ML

- **Training Data**: Store structured training data
- **Feature Engineering**: Use SQL for feature creation and transformation
- **Data Preprocessing**: Clean and prepare data using database functions
- **Historical Data**: Store historical data for time-series analysis

Model Metadata Storage

- **Model Registry**: Store model metadata and versioning information
- **Experiment Tracking**: Track ML experiments and results
- **Performance Metrics**: Store model performance metrics over time
- **Hyperparameter Storage**: Store hyperparameter configurations

Real-time Applications

- **Feature Store**: Store pre-computed features for real-time inference
- **User Profiles**: Store user data for personalization
- **Transaction Data**: Store transaction data for fraud detection
- **Application Data**: Backend database for ML-powered applications

Security Features:

- **Encryption**: At-rest encryption using AWS KMS
- **Network Isolation**: VPC deployment with security groups
- **IAM Authentication**: Database authentication using IAM
- **SSL/TLS**: Encrypted connections to database
- **Audit Logging**: Database activity logging and monitoring

Backup and Recovery:

- **Automated Backups**: Daily automated backups with point-in-time recovery
- **Manual Snapshots**: On-demand database snapshots
- **Cross-Region Backup**: Copy snapshots to different regions
- **Retention Period**: Configurable backup retention period

Monitoring and Performance:

- **CloudWatch Integration**: Built-in metrics and monitoring
- **Performance Insights**: Database performance monitoring and tuning
- **Enhanced Monitoring**: OS-level metrics and monitoring
- **Slow Query Logs**: Identify and optimize slow queries

Maintenance and Updates:

- **Automatic Updates**: Automated minor version updates
- **Maintenance Windows**: Scheduled maintenance periods
- **Zero Downtime**: Multi-AZ deployments enable zero-downtime updates
- **Manual Control**: Option to control update timing

Best Practices:

- Choose appropriate instance class and storage type for workload
- Use Multi-AZ for production databases
- Implement read replicas for read-heavy workloads
- Regular backup testing and disaster recovery planning
- Monitor performance and optimize queries

- Use connection pooling to manage database connections

Pricing Components:

- **Instance Hours**: Hourly rate based on instance class
- **Storage**: Cost per GB of storage allocated
- **I/O Operations**: Additional charges for high I/O workloads
- **Backup Storage**: Storage costs for automated backups
- **Data Transfer**: Cross-region and internet data transfer costs

---

Summary and Key Takeaways

Service Categories Overview

Analytics Services: Focus on data processing, transformation, and visualization

- **Data Processing**: EMR for big data, Glue for ETL, DataBrew for visual data prep
- **Data Management**: Lake Formation for data lakes, Data Exchange for external data
- **Search and Analytics**: OpenSearch for search and log analytics
- **Visualization**: QuickSight for business intelligence and dashboards
- **Data Warehousing**: Redshift for analytical workloads

Financial Management: Cost monitoring and optimization

- **Budgets**: Proactive cost management and alerting
- **Cost Explorer**: Historical cost analysis and optimization recommendations

Compute: Scalable computing resources

- **EC2**: Flexible virtual machines for any workload including AI/ML
- **Container Services**: ECS and EKS for containerized applications and microservices

Database Services: Managed database solutions for different use cases

- **Relational**: RDS for traditional RDBMS workloads
- **NoSQL**: DynamoDB for key-value, DocumentDB for documents, Neptune for graphs
- **Caching**: ElastiCache for caching, MemoryDB for durable in-memory storage

AI/ML Integration Points

- **Data Pipeline**: Use analytics services for data preparation and feature engineering
- **Model Training**: Use compute services (EC2, containers) for training workloads
- **Model Serving**: Use containers and databases for scalable model deployment
- **Feature Storage**: Use databases for real-time and batch feature serving
- **Cost Management**: Monitor and optimize AI/ML workload costs

Study Tips

1. **Understand service purposes**: Know what each service is designed for
2. **Learn integration patterns**: Understand how services work together

3. **Focus on AI/ML use cases**: Pay special attention to AI/ML applications
4. **Know pricing models**: Understand how each service is billed
5. **Security considerations**: Understand security features across all services