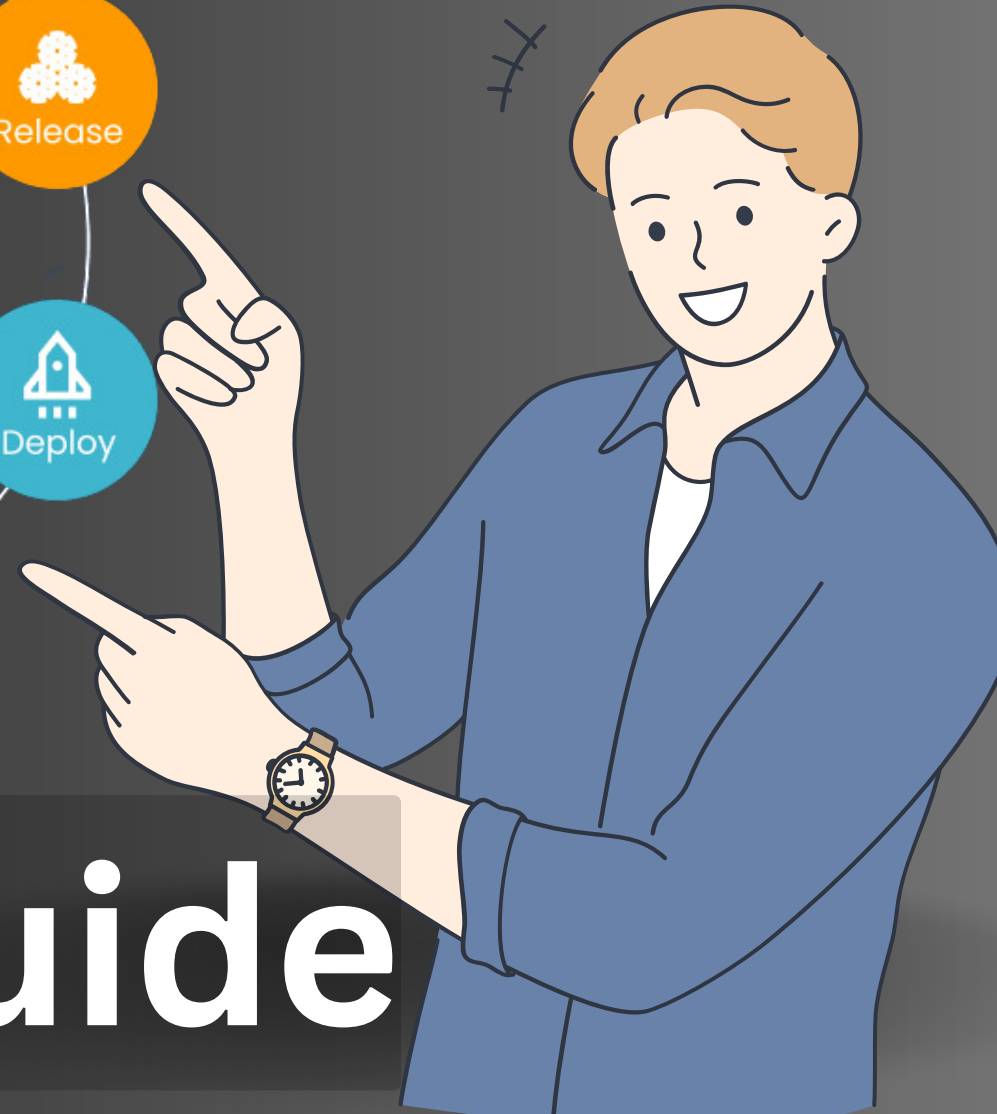




# Quick Exam Guide





# Analytics Services

Service Name	What is it?	When to Use?	How to Use in Real World?
<b>Amazon Athena</b>	Query service allowing analysis of data in Amazon S3 using SQL.	Ideal for querying vast datasets stored in S3 without the need for complex ETL processes.	Analyzing log files stored in S3 using SQL-like queries
<b>Amazon EMR</b>	Big data processing framework for processing vast amounts of data across dynamically resizable Amazon EC2 instances.	Best suited for processing large datasets and performing distributed data analysis using frameworks like Apache Spark, Hadoop, etc.	Processing and analyzing vast amounts of clickstream data
<b>Amazon Kinesis Data Firehose</b>	Service for loading streaming data into data stores and analytics tools.	Use for streaming data ingestion into various destinations such as S3, Redshift, Elasticsearch, etc., with simplicity and scalability.	Collecting and loading streaming data into Amazon S3 for analytics
<b>Amazon Kinesis Data Streams</b>	Platform for building custom applications that process or analyze streaming data in real-time.	Optimal for real-time data processing and analysis where low latency is crucial, like IoT data, clickstreams, and log monitoring.	Capturing and analyzing IoT device data in real-time
<b>Amazon OpenSearch Service</b>	Fully managed Elasticsearch service for deploying, securing, and scaling Elasticsearch clusters.	Useful for full-text search, log analytics, real-time application monitoring, and more.	Creating a search solution for a web application
<b>Amazon QuickSight</b>	Cloud-powered business intelligence service for visualizing data and generating insights.	Great for creating interactive dashboards and generating business insights from various data sources.	Creating dashboards to monitor sales performance or operational metrics



# Compute Services



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS App Runner</b>	Fully managed service for deploying containerized applications without managing infrastructure.	Perfect for quickly deploying web applications or APIs without worrying about the underlying infrastructure.	Deploying a web application without managing infrastructure
<b>Amazon EC2</b>	Scalable virtual servers on AWS, allowing users to run applications.	Ideal for various workloads requiring control over the underlying infrastructure, scalability, and customization.	Running a website on a virtual server
<b>Amazon EC2 Auto Scaling</b>	Automates the scaling of EC2 instances based on demand to maintain performance and reduce costs.	Best for applications with varying traffic patterns, ensuring optimal performance and cost-efficiency by adjusting the number of instances automatically.	Handling traffic spikes by automatically adding more EC2 instances
<b>EC2 Image Builder</b>	Service to create and maintain secure OS images for EC2 instances.	Useful for automating the creation, management, and distribution of customized EC2 machine images.	Creating golden images with pre-installed software configurations for consistent EC2 launches
<b>AWS Elastic Beanstalk</b>	Service for deploying and scaling web applications and services easily.	Ideal for quickly deploying web applications without dealing with the underlying infrastructure complexities.	Deploying a web app without worrying about underlying infrastructure
<b>AWS Serverless Application Repository</b>	Storage and discovery of serverless applications and components.	Great for sharing and discovering serverless applications, enabling reuse of application components across different projects.	Sharing a serverless application template across multiple teams or projects

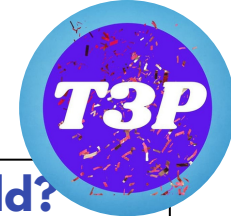


# Containers Services



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS App2Container</b>	Tool for containerizing applications and migrating them to AWS ECS or EKS.	Ideal for modernizing existing applications by containerizing them and moving to ECS or EKS.	Converting a monolithic application into microservices for scalability
<b>AWS Copilot</b>	CLI tool for building, deploying, and managing containerized applications on AWS.	Perfect for developers to streamline the development and deployment of containerized applications on AWS.	Managing containerized microservices applications with ease
<b>Amazon Elastic Container Registry</b>	Fully managed Docker container registry for storing, managing, and deploying container images.	Suitable for storing and managing container images securely, allowing seamless deployment to ECS, EKS, and on-premises environments.	Hosting Docker images for a microservices architecture
<b>Amazon Elastic Container Service</b>	Highly scalable, high-performance container orchestration service.	Ideal for deploying and managing containers at scale with deep integration to AWS services.	Running a distributed application on Docker containers across multiple EC2 instances
<b>Amazon Elastic Kubernetes Service</b>	Managed Kubernetes service for deploying, managing, and scaling containerized applications using Kubernetes.	Best for running Kubernetes-based applications, providing control and flexibility while managing infrastructure.	Managing containerized applications using Kubernetes clusters
<b>Amazon EKS Distro</b>	Kubernetes distribution derived from EKS and open-source Kubernetes.	Suitable for organizations wanting to run Kubernetes clusters in their data centers or on other clouds while staying consistent with EKS.	Setting up Kubernetes clusters on-premises or in other cloud environments
<b>AWS Fargate</b>	Serverless compute engine for running containers without managing infrastructure.	Perfect for running containers without provisioning or managing servers, suitable for microservices architectures.	Running a microservices architecture without worrying about the underlying infrastructure
<b>Red Hat OpenShift Service on AWS</b>	Fully managed Red Hat OpenShift service providing Kubernetes platform capabilities on AWS.	Ideal for organizations leveraging OpenShift to develop and deploy applications on AWS, ensuring compatibility and integration with AWS services.	Deploying and managing applications with OpenShift on AWS

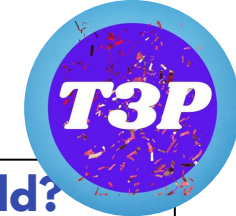
# Database Services



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>Amazon Aurora</b>	MySQL and PostgreSQL-compatible relational database	Suitable for applications demanding high-performance databases with compatibility to MySQL or PostgreSQL.	Running a scalable eCommerce application with high availability
<b>Amazon Aurora Serverless v2</b>	On-demand, auto-scaling Aurora database	Best for intermittent or unpredictable workloads, providing cost savings through auto-scaling and paying only for consumed resources.	Handling varying workloads in a serverless environment with Aurora
<b>AWS Database Migration Service</b>	Migrate databases to AWS easily	Ideal for migrating databases to AWS with minimal downtime and ensuring continuous data replication during migration.	Migrating an on-premises database to AWS
<b>Amazon DocumentDB</b>	Fully managed MongoDB-compatible document database	Suitable for applications requiring scalability, performance, and compatibility with MongoDB.	Building applications using MongoDB without managing the database infrastructure
<b>Amazon DynamoDB</b>	Fully managed NoSQL database service	Ideal for applications with predictable and low-latency requirements, such as gaming, IoT, and mobile apps.	Storing and querying high-traffic user profile data for a gaming application
<b>Amazon ElastiCache</b>	In-memory caching service	Suitable for improving the performance of applications by caching frequently accessed data.	Speeding up data retrieval by caching frequently accessed information
<b>Amazon MemoryDB for Redis</b>	Fully managed Redis-compatible in-memory database	Ideal for applications requiring sub-millisecond latency, durability, and scalability with Redis compatibility.	Implementing a leaderboard system for a multiplayer online game using Redis
<b>Amazon RDS</b>	Managed relational database service	Suitable for deploying, managing, and scaling relational databases without the administrative overhead.	Hosting an eCommerce application's MySQL database on AWS
<b>Amazon Redshift</b>	Fully managed data warehouse service	Ideal for analyzing large-scale datasets and generating business insights using SQL queries.	Analyzing historical sales data to make business decisions



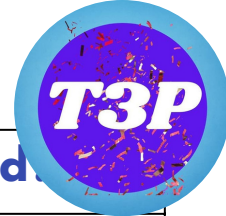
# Developer Tools (Part 1)



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS CLI</b>	Command-line tool for interacting with AWS services, enabling script automation, and managing resources.	Useful for scripting, automation, and managing AWS resources.	Automate deployments or manage AWS resources through scripts for streamlined operations.
<b>AWS Cloud Development Kit (AWS CDK)</b>	Infrastructure as Code (IaC) framework allowing developers to define AWS resources using familiar programming languages.	Ideal for creating and managing AWS resources using programming languages like TypeScript, Python, or Java.	Define AWS infrastructure using code, enabling version control and reuse across projects.
<b>AWS CloudShell</b>	Browser-based shell to access AWS management console and AWS services through a command-line interface.	Useful for quick AWS resource management without local setup.	Access and manage AWS resources directly from the browser without configuring local environments.
<b>AWS CodeArtifact</b>	Fully managed artifact repository enabling organization, storage, and sharing of software packages.	Suitable for securely storing and sharing software artifacts across development teams.	Store and manage software packages securely, allowing teams to share dependencies efficiently.
<b>AWS CodeBuild</b>	Fully managed build service for compiling source code, running tests, and producing software packages.	Use for automating build processes, ensuring consistent and scalable builds.	Automate build processes, such as compiling and testing, to streamline software delivery.
<b>AWS CodeCommit</b>	Git-based source control service to securely store and manage code repositories.	Suitable for securely storing and version-controlling code with collaboration features.	Host private Git repositories, manage code, and collaborate with team members securely.
<b>AWS CodeDeploy</b>	Automated deployment service facilitating code deployments to various AWS services.	Ideal for automating application deployments, ensuring reliable and rapid release cycles.	Automate deployments of applications to different environments, improving release management.
<b>Amazon CodeGuru</b>	Machine learning-powered service for automated code reviews and application performance recommendations.	Use for improving code quality, identifying issues, and optimizing application performance.	Automatically review code for best practices, performance bottlenecks, and potential improvements.



# Developer Tools (Part 2)

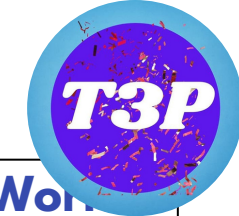


Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS CodePipeline</b>	Continuous integration and continuous delivery (CI/CD) service for automating build, test, and deployment pipelines.	Suitable for orchestrating automated software release workflows, from source code to deployment.	Create and manage CI/CD pipelines to automate software delivery processes, ensuring consistency and efficiency.
<b>AWS CodeStar</b>	Integrated development environment (IDE) facilitating project creation, management, and deployment on AWS.	Ideal for quickly setting up and managing AWS development projects with collaboration features.	Initiate and manage development projects, collaborate with team members, and deploy applications to AWS.
<b>AWS Fault Injection Simulator (AWS FIS)</b>	Service enabling controlled chaos engineering experiments to test application resilience and fault tolerance.	Use to simulate faults and test the resilience of applications running on AWS infrastructure.	Simulate various failure scenarios in AWS infrastructure to ensure application resilience and recovery mechanisms.
<b>AWS SDKs and Tools</b>	Collection of software development kits (SDKs) and tools for various programming languages, simplifying interaction with AWS services.	Essential for developers to easily integrate AWS services into their applications using preferred languages.	Use SDKs to interact with AWS services programmatically, enabling seamless integration with applications.
<b>AWS X-Ray</b>	Service for tracing, analyzing, and debugging distributed applications, providing insights into application performance and bottlenecks.	Useful for identifying performance issues and optimizing application performance in distributed systems.	Trace requests across distributed systems, identify performance bottlenecks, and troubleshoot application issues.

## Application Integration Services

Service Name	What is it?	When to Use?	How to Use in Real World?
<b>Amazon AppFlow</b>	Fully managed integration service for securely transferring data between SaaS applications and AWS services.	Ideal for automating data transfer between different applications without needing to write custom integration code.	Syncing data between Salesforce and Amazon S3 for analytics
<b>Amazon EventBridge</b>	Serverless event bus service for connecting applications using events.	Best for building event-driven architectures and simplifying application integrations by decoupling services via event-driven communication.	Triggering Lambda functions based on events from various sources

# Management & Governance (Part 1)



Service Name	What is it?	When to Use?	How to Use in Real World
<b>AWS Auto Scaling</b>	Service for automatically adjusting AWS resources to maintain application performance at a desired level and optimize costs.	Maintaining performance with varying workloads	Automatically adjusting EC2 instance count based on CPU utilization
<b>AWS CloudFormation</b>	IaC service allowing creation and management of AWS resources using templates, enabling consistent and reproducible infrastructure.	Automating resource provisioning and management	Defining a cloud infrastructure using a YAML or JSON template and deploying it using CloudFormation
<b>AWS CloudTrail</b>	Service providing visibility into user and resource activity by recording AWS API calls for auditing, compliance, and governance.	Tracking user activity and changes in AWS resources	Monitoring changes made to IAM policies and permissions using CloudTrail logs
<b>Amazon CloudWatch</b>	Monitoring and observability service for collecting and tracking metrics, logs, and events from AWS resources and applications.	Monitoring performance and health of AWS resources	Setting alarms to notify when CPU utilization exceeds a certain threshold on EC2 instances
<b>Amazon CloudWatch Logs</b>	Service for centralizing logs from AWS resources and applications, enabling easy monitoring, analysis, and troubleshooting.	Collecting, monitoring, and storing log data	Storing application logs for auditing and troubleshooting purposes
<b>AWS Compute Optimizer</b>	Service analyzing resource utilization to recommend optimal AWS resources and configurations, improving performance and reducing costs.	Optimizing application performance and costs	Analyzing historical usage data to rightsize EC2 instances for cost savings
<b>AWS Config</b>	Service for assessing, auditing, and evaluating the configurations of AWS resources, maintaining compliance and security.	Assessing, auditing, and evaluating resource configurations	Ensuring compliance by monitoring and evaluating AWS resource configurations
<b>AWS Control Tower</b>	Service for setting up and governing a secure, compliant multi-account AWS environment following best practices.	Managing multiple AWS accounts and resources	Establishing a baseline security and compliance configuration across multiple AWS accounts
<b>AWS Health</b>	Service providing personalized information and recommendations about AWS services' operational health.	Monitoring the health of AWS services and resources	Receiving notifications about AWS service disruptions or issues
<b>AWS License Manager</b>	Service managing software licenses and enforcing license configurations across AWS resources.	Tracking and controlling software usage and licenses	Ensuring compliance with software licensing agreements on AWS



# Management & Governance (Part 2)



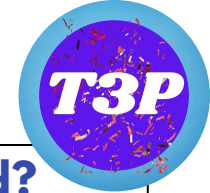
Service Name	What is it?	When to Use?	How to Use in Real World?
<b>Amazon Managed Grafana</b>	Service offering managed Grafana for visualizing and analyzing metrics from multiple data sources.	Data visualization and monitoring	Creating dashboards to monitor AWS resource performance using Managed Grafana
<b>Amazon Managed Service for Prometheus</b>	Managed Prometheus service for monitoring containerized applications at scale.	Monitoring and alerting using Prometheus	Collecting and visualizing application metrics using Managed Service for Prometheus
<b>AWS OpsWorks</b>	Configuration management service automating server provisioning, configuration, and deployment.	Managing applications and infrastructure	Automating deployment and scaling of applications on AWS using OpsWorks
<b>AWS Organizations</b>	Service for centrally managing and governing multiple AWS accounts, simplifying management and security at scale.	Centralized management of multiple AWS accounts	Managing permissions and policies across multiple AWS accounts
<b>AWS Proton</b>	Service for automating and managing infrastructure provisioning and code deployments for serverless and container-based applications.	Standardizing and automating application deployments	Simplifying deployment of microservices with predefined templates using Proton
<b>AWS Resilience Hub</b>	Service providing a unified view of multiple AWS accounts' readiness and resilience against disruptions.	Monitoring and managing AWS service resilience	Tracking and managing fault tolerance strategies across AWS services
<b>AWS Service Catalog</b>	Service for creating and managing catalogs of approved IT services, allowing control and standardization of deployments.	Standardizing and managing IT services across an organization	Offering a catalog of pre-approved AWS resources for easy deployment
<b>AWS Systems Manager</b>	Collection of services for managing AWS resources, automating tasks, and configuring operational parameters.	Automating operational tasks and configuration management	Patching and updating EC2 instances using Systems Manager Run Command
<b>AWS Trusted Advisor</b>	Service offering best practice recommendations for optimizing AWS environments in terms of cost, performance, security, and fault tolerance.	Recommending best practices for cost optimization	Identifying and reducing underutilized resources on AWS

# Networking and Content Delivery



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>Amazon API Gateway</b>	Fully managed service for creating, publishing, maintaining, monitoring, and securing APIs at scale.	Use to create and manage APIs, enabling secure and efficient communication between applications and services.	Create, publish, and secure APIs at scale, facilitating communication between different applications and services.
<b>AWS Client VPN</b>	Managed VPN service allowing secure access to AWS resources and on-premises networks from any location.	Ideal for securely connecting users to AWS resources and on-premises networks from remote locations.	Provide secure access to AWS resources and on-premises networks for users working from remote locations.
<b>Amazon CloudFront</b>	Content delivery network (CDN) service delivering data, videos, applications, and APIs with low latency and high transfer speeds.	Suitable for accelerating content delivery globally, reducing latency, and improving user experience.	Distribute content with low latency and high transfer speeds, improving performance for end users globally.
<b>Elastic Load Balancing (ELB)</b>	Service for automatically distributing incoming application traffic across multiple targets, enhancing fault tolerance and scalability.	Use to ensure fault tolerance and scalability by distributing incoming traffic across multiple targets.	Distribute incoming application traffic across multiple targets, ensuring high availability and fault tolerance.
<b>AWS PrivateLink</b>	Service enabling private connectivity between VPCs, AWS services, and on-premises applications securely.	Suitable for securely accessing services hosted on AWS or on-premises through private connectivity.	Establish private connections between VPCs, AWS services, and on-premises applications, ensuring security.
<b>Amazon Route 53</b>	Scalable and highly available DNS service for routing traffic to internet resources and AWS services.	Ideal for managing DNS routing, domain registration, and health checking for internet-facing applications.	Manage DNS routing, domain registration, and health checks for applications, ensuring high availability and routing.
<b>AWS Site-to-Site VPN</b>	Managed VPN service enabling secure communication between on-premises networks and AWS through encrypted connections.	Useful for securely connecting on-premises networks to AWS infrastructure over encrypted connections.	Establish secure communication between on-premises networks and AWS infrastructure using encrypted connections.
<b>AWS Transit Gateway</b>	Service simplifying network connectivity between VPCs, VPNs, and on-premises networks, managing traffic at scale.	Suitable for managing network traffic at scale, connecting multiple VPCs and on-premises networks seamlessly.	Simplify network connectivity between VPCs, VPNs, and on-premises networks, managing traffic efficiently at scale.
<b>Amazon VPC</b>	Service enabling creation of isolated virtual networks within AWS, providing control over IP addressing, subnets, and security.	Ideal for designing and controlling network settings, ensuring secure and isolated environments within AWS.	Create isolated virtual networks, define IP addressing, subnets, and security settings for AWS resources.

# Storage Services



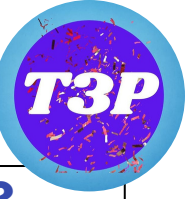
Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS Backup</b>	Centralized backup service for AWS resources.	Use for automating and managing backups across various AWS services.	Create and manage backup plans to protect data stored in different AWS resources.
<b>Amazon Elastic Block Store (Amazon EBS)</b>	Block-level storage volumes for EC2 instances.	Use for durable and low-latency block storage for EC2 instances.	Attach persistent block storage to EC2 instances for data storage.
<b>AWS Elastic Disaster Recovery</b>	Disaster recovery service for on-premises applications.	Use for replicating on-premises applications to AWS for disaster recovery.	Ensure business continuity by replicating on-premises applications to AWS for quick recovery in case of disasters.
<b>Amazon Elastic File System (Amazon EFS)</b>	Fully managed file storage for EC2 instances.	Use for scalable and highly available file storage for EC2 instances.	Share file data across multiple EC2 instances with scalable storage capacity.
<b>Amazon FSx for Lustre</b>	High-performance file system for compute-intensive workloads.	Use for high-performance file storage for compute-intensive applications.	Accelerate data processing by leveraging high-performance file storage optimized for compute-intensive workloads.
<b>Amazon FSx for NetApp ONTAP</b>	Fully managed ONTAP file storage service.	Use for enterprise-grade file storage with NetApp ONTAP features.	Utilize enterprise-grade file storage with advanced data management capabilities.
<b>Amazon FSx for OpenZFS</b>	OpenZFS-based file storage service.	Use for scalable and cost-effective file storage with OpenZFS features.	Store and manage data cost-effectively using OpenZFS file storage.
<b>Amazon FSx for Windows File Server</b>	Fully managed Windows-native file storage service.	Use for Windows-compatible file storage with high availability and performance.	Store and access Windows-based file data with native compatibility and high performance.
<b>Amazon S3</b>	Object storage service for scalable and durable data storage.	Use for storing and retrieving any amount of data from anywhere.	Store and retrieve objects (files, images, documents) securely in S3 buckets.
<b>Amazon S3 Glacier</b>	Low-cost archival storage with flexible retrieval options.	Use for long-term storage of data with infrequent access.	Archive data at a low cost and retrieve when needed with flexible retrieval options.
<b>AWS Storage Gateway</b>	Hybrid cloud storage integration service.	Use for integrating on-premises environments with cloud storage.	Seamlessly connect on-premises environments with cloud storage using storage gateway configurations.

# Serverless Services



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS Lambda</b>	Runs code without provisioning or managing servers.	Use for executing code in response to events or triggers.	Utilize Lambda to execute code in response to events from various AWS services or custom triggers, allowing serverless, event-driven architectures.
<b>AWS Serverless Application Model (AWS SAM)</b>	Framework for building serverless applications.	Ideal for simplifying the deployment of serverless applications.	Use SAM to simplify the creation, deployment, and management of serverless applications using AWS CloudFormation templates and predefined configurations.
<b>Amazon Simple Notification Service (Amazon SNS)</b>	Pub/sub messaging service for microservices, distributed systems, etc.	Use for event-driven communication between services.	Implement SNS for pub/sub messaging between microservices, distributed systems, or applications, enabling event-driven communication and coordination.
<b>Amazon Simple Queue Service (Amazon SQS)</b>	Fully managed message queuing service.	Ideal for decoupling and scaling microservices.	Use SQS to decouple and scale microservices by enabling asynchronous message communication between components, ensuring reliability and scalability.
<b>AWS Step Functions</b>	Orchestrates serverless workflows using visual workflows.	Use for coordinating multiple AWS services in workflows.	Employ Step Functions to create visual workflows that coordinate multiple AWS services, enabling the orchestration of tasks and managing complex workflows in a serverless environment.

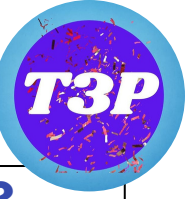
# Security, Identity, and Compliance (Part 1)



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS Certificate Manager (ACM)</b>	Manages SSL/TLS certificates for AWS resources.	Use to simplify certificate management for AWS services and deploy SSL/TLS certificates.	Securely deploy SSL/TLS certificates to web servers or load balancers.
<b>AWS CloudHSM</b>	Offers dedicated hardware security module (HSM)	Ideal for regulatory compliance and secure key storage.	Protect sensitive data and comply with strict security regulations by securely storing and using cryptographic keys.
<b>Amazon Cognito</b>	Provides authentication, authorization, and user management.	Use for user authentication and authorization in applications.	Implement user sign-up, sign-in, and access control in applications.
<b>Amazon Detective</b>	Analyzes, investigates, and identifies security issues.	Ideal for investigating security incidents and breaches.	Investigate and resolve security incidents through automated analysis.
<b>AWS Directory Service</b>	Manages and connects AWS resources to an existing on-premises directory.	Useful when integrating AWS resources with existing on-premises directories.	Seamlessly extend on-premises AD to AWS services for user authentication and access management.
<b>Amazon GuardDuty</b>	Intelligent threat detection service.	Use for continuous monitoring and threat detection.	Detect and respond to security threats by analyzing AWS logs and network activity.
<b>AWS IAM Identity Center (AWS Single Sign-On)</b>	Centrally manages access to multiple AWS accounts.	Ideal for managing access across multiple AWS accounts.	Centrally manage user access and permissions across various AWS accounts and services.
<b>AWS Identity and Access Management (IAM)</b>	Manages access to AWS services and resources.	Essential for controlling access within AWS environments.	Define fine-grained access control and permissions for users, groups, and roles.
<b>Amazon Inspector</b>	Automated security assessment service.	Useful for assessing application security and compliance.	Automatically assess security and compliance of EC2 instances and applications.



# Security, Identity, and Compliance (Part 2)



Service Name	What is it?	When to Use?	How to Use in Real World?
<b>AWS Key Management Service (AWS KMS)</b>	Manages encryption keys for other AWS services.	Use for secure encryption key generation and management.	Encrypt data and manage keys to maintain control over data access and security.
<b>Amazon Macie</b>	AI-powered data security and privacy service.	Ideal for discovering, classifying, and protecting sensitive data.	Automatically discover and protect sensitive data using machine learning.
<b>AWS Network Firewall</b>	Managed firewall service for filtering network traffic.	Use for protecting network infrastructure.	Set up and manage network security policies to filter traffic based on customizable rules.
<b>AWS Resource Access Manager (AWS RAM)</b>	Shares AWS resources across AWS accounts.	Useful for resource sharing across multiple accounts.	Easily share resources across multiple accounts for centralized management.
<b>AWS Secrets Manager</b>	Manages and rotates secrets and credentials.	Ideal for secure storage and automated rotation of secrets.	Store and manage credentials securely and automate their rotation.
<b>AWS Security Hub</b>	Provides a comprehensive view of security posture.	Use for centralized security monitoring and compliance checking.	Gain insights and take action on security and compliance issues across AWS environments.
<b>AWS Security Token Service (AWS STS)</b>	Issues temporary security credentials.	Useful for granting limited and temporary access.	Enable users or services to temporarily access AWS resources with limited permissions.
<b>AWS Shield</b>	Managed DDoS protection service.	Essential for safeguarding against DDoS attacks.	Automatically detect and mitigate DDoS attacks to maintain the availability of applications.
<b>AWS WAF</b>	Web Application Firewall for filtering HTTP traffic.	Use for protecting web applications from common web exploits.	Protect web applications from common attacks by setting up rules and filters.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 1: SDLC Automation (22%)

- 1 Setting Up a Basic CI/CD Pipeline:** Use AWS CodePipeline, CodeBuild, and CodeDeploy to create a simple CI/CD pipeline for a sample application.
- 2 Advanced Pipeline Configuration:** Explore features like manual approval steps, integration with external tools, and custom actions within CodePipeline.
- 3 Deploying Microservices with AWS Tools:** Create a CI/CD pipeline for a microservices architecture using AWS CodePipeline and CodeDeploy.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 2: Configuration Management and IaC (17%)

- 4 Infrastructure as Code with AWS CloudFormation:** Deploy an infrastructure stack using CloudFormation templates to create resources like EC2 instances, VPC, and S3 buckets.
- 5 AWS Systems Manager (SSM) Automation:** Automate routine tasks using Systems Manager Automation documents for instance configuration, patching, and maintenance.
- 6 Configuration Management with Ansible or Chef:** Set up and configure a basic infrastructure using Ansible or Chef playbooks.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 3: Resilient Cloud Solutions (15%)

- 7 Auto Scaling and Load Balancing:** Create an Auto Scaling group and associate it with an Elastic Load Balancer (ELB) to manage traffic across instances.
- 8 Implementing Multi-AZ Deployments:** Deploy a highly available application across multiple Availability Zones (AZs) using CloudFormation.
- 9 Disaster Recovery Setup:** Design a disaster recovery plan using services like AWS Backup, AWS Import/Export, and cross-region replication.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 4: Monitoring and Logging (15%)

- 10** **Setting Up CloudWatch Metrics and Alarms:** Configure CloudWatch to monitor resource usage, set alarms, and trigger actions based on specific thresholds.
- 11** **Centralized Logging with CloudTrail and CloudWatch Logs:** Enable CloudTrail for auditing API calls and aggregate logs in CloudWatch Logs for analysis.
- 12** **Application Performance Monitoring with X-Ray:** Instrument an application with AWS X-Ray to trace requests and identify performance bottlenecks.



# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 5: Incident and Event Response (14%)

- 13** **Configuring AWS Config Rules:** Set up AWS Config rules to ensure compliance and trigger actions based on rule violations.
- 14** **Automating Incident Response with Lambda:** Create Lambda functions to automate responses to specific events detected in CloudWatch or Config.
- 15** **Practice Incident Response:** Simulate incidents using AWS services like Systems Manager Incident Manager and follow an incident response plan.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### Domain 6: Security and Compliance (17%)

- 16 IAM Roles and Policies:** Create IAM roles, policies, and manage permissions for users, groups, and roles following the principle of least privilege.
- 17 Encryption and Key Management:** Implement encryption using AWS KMS for data at rest and in transit. Rotate keys and manage key policies.
- 18 Compliance Checks with AWS Config and Inspector:** Use AWS Config and Inspector to perform compliance checks and security assessments.

# 20 Hands-on Labs

## AWS Certified DevOps Engineer - Professional exam

### General Labs Covering Multiple Domains

19

**End-to-End Application Deployment:** Build a complete application deployment using a combination of services from various domains.

20

**Advanced Troubleshooting Scenarios:** Simulate complex scenarios involving scaling issues, configuration errors, and security breaches to troubleshoot and resolve.

CGNNECT



<https://www.youtube.com/@T3Ptech>



<https://t.me/LearnDevOpsForFree>