Serverless Cloud Services

CSYE 6225: Network Structure & Cloud Computing
Northeastern University

Instructor: Raja Alomari,PhD

M8: Overview

Topics include:

- Introduction to Serverless Computing.
- AWS Serverless Overview.
- Overview of various AWS serverless and fully managed services.

Serverless Cloud Services

Objectives:

- Understand Serverless services.
- Articulate pros and cons of serverless services.
- Learn several additional AWS services and their use cases.
- Gain hands-on experience using several additional AWS services.

What is Serverless Computing?

Definition: Building and running applications without managing servers. Serverless computing allows developers to focus solely on building and running applications without the need to provision, manage, or maintain servers.

Key Features:

- **Event-Driven Architecture:** Executes code in response to events such as HTTP requests, database updates, or messaging queues.
- **Automatic Scaling:** Adjusts resources dynamically based on workload demands.
- **Pay-Per-Use Pricing:** Charges are based on the actual compute time and resources used, not on pre-allocated server capacity.

Serverless Computing: Pros and Cons

Advantages

- **No Infrastructure Management:** Offloads server setup, maintenance, and scaling to the cloud provider.
- **Faster Time-to-Market:** Simplifies development and allows developers to focus on innovation.
- **Built-In High Availability:** Ensures reliability and fault tolerance without additional effort.

Challenges

- **Cold Starts:** Initial latency when spinning up functions that aren't in use.
- **Vendor Lock-In:** Dependency on a specific cloud provider's ecosystem and services.
- **Debugging Complexity:** Distributed nature can make tracing and resolving issues more difficult.

Key AWS Serverless Compute Services

AWS Lambda Overview

Event-driven compute service for running code.

Key Features:

- Supports multiple runtimes (Node.js, Python, Go, etc.)
- Integrates with other AWS services (S3, DynamoDB, API Gateway)

Use Cases:

- Data processing workflows
- Back-end for mobile/web apps
- Cron jobs for scheduled tasks

Diagram: Lambda function handling file uploads from S3.

AWS App Runner

Fully managed service designed to make it easy to **deploy containerized web applications** and **APIs** directly from source code or a container registry, without needing to manage infrastructure.

Key Features

- **Simplified Deployment:** Deploy directly from source repositories like GitHub or container registries such as Amazon Elastic Container Registry (ECR).
- **Fully Managed Environment:** Automatically handles load balancing, auto-scaling, and application updates.
- **HTTPS Support:** Provides end-to-end HTTPS connections with auto-renewed TLS certificates.
- **Scaling:** Dynamically adjusts the number of instances based on traffic, ensuring cost efficiency.
- **Built-in Observability:** Offers integration with AWS CloudWatch, AWS X-Ray, and Amazon EventBridge for monitoring and debugging.
- **Pay-as-You-Go:** You are billed for the resources used, including compute and memory.

AWS Fargate

A serverless compute engine for Amazon Elastic Container Service (**ECS**) and Amazon Elastic Kubernetes Service (**EKS**). It allows you to run containers without managing the underlying infrastructure.

Key Features

- **Serverless Containers:** Eliminates the need to provision and manage servers, VMs, or clusters.
- **Integration with ECS and EKS:** Works seamlessly with both ECS (native AWS container orchestration) and EKS (Kubernetes-based orchestration).
- **Customizable Configuration:** Offers fine-grained control over CPU and memory resources for your containers.
- **Automatic Scaling:** Containers scale independently based on the workload, optimizing costs.
- **Security by Isolation:** Each task runs in its own kernel-isolated environment, improving security.
- **Pricing:** Pay only for the resources (CPU and memory) used while your containers are running.

Amazon SNS – Pub/Sub Messaging

A fully managed service designed for the **publish/subscribe** (pub/sub) messaging pattern. It enables the decoupling of producers (publishers) and consumers (subscribers).

Key Features:

- **Publish/Subscribe Model:** Publishers send messages to an SNS topic, and multiple subscribers (e.g., SQS queues, Lambda functions, HTTP endpoints, email, SMS) receive them.
- **Message Fan-Out:** A single message can be distributed to multiple endpoints or services simultaneously.
- **Delivery Options:**
  - Push-based delivery to multiple subscribers.
  - Supports a variety of protocols (HTTP/S, email, SMS, Lambda, SQS, or custom endpoints).
- **Durability and Reliability:** Stores messages across multiple availability zones for high durability.
- **Filtering:** Message filtering allows subscribers to receive only relevant messages based on attributes.
- **Event-Driven:** Often used in event-driven architectures.

Amazon SQS – Message Queuing

A fully managed service designed for message queuing, enabling communication between distributed application components.

Key Features:

- **Queue-Based Model:** Messages are sent to a queue where they are stored until the receiving application processes them.
- **Message Persistence:**
  - Standard Queue: Offers at-least-once delivery and high throughput.
  - FIFO Queue: Ensures exactly-once processing and maintains message order.
- **Polling:**
  - Short polling: Returns messages immediately (even if none are available).
  - Long polling: Waits for messages to arrive, reducing empty responses.
- **Scalable:** Automatically scales to handle any number of messages.
- **Decoupling:** Provides a reliable way to decouple components in a distributed system.
- **Dead-Letter Queue (DLQ):** Captures messages that cannot be processed successfully.

Amazon EventBridge

A serverless event bus service used for building **event-driven applications**. It allows you to route, filter, and deliver events from AWS services, SaaS applications, or custom applications to various targets.

Key Features:

- **Event Routing:** Connects sources (AWS services, SaaS, or custom applications) to targets (e.g., Lambda, SQS, SNS, Kinesis). Uses rules to filter and route events based on specific conditions.
- **Schema Registry:** Automatically detects event structures and provides a schema registry for discovery and reuse.
- **Serverless:** Automatically scales with demand, ensuring reliable event delivery.
- **Integration with SaaS:** Supports integrations with popular SaaS applications like Zendesk, Datadog, and others.
- **High Availability and Durability:** Ensures events are processed reliably across multiple availability zones.
- **Flexible Targets:** Supports over 15 AWS services as event targets, including Lambda, Step Functions, and ECS.

Amazon API Gateway

A fully managed service for creating, publishing, and managing REST, HTTP, and WebSocket APIs. It acts as a bridge between clients and backend services, enabling you to expose your application's functionality securely and scalably.

Key Features:

- **Protocol Support:** Supports RESTful APIs, HTTP APIs (for simpler use cases), and WebSocket APIs for real-time two-way communication.
- **Customizable Routes:** Define endpoints with methods (GET, POST, PUT, DELETE) and connect them to backend services such as Lambda, ECS, or DynamoDB.
- **Authentication and Authorization:** Supports AWS IAM, Amazon Cognito, and custom authorizers for API security.
- **Throttling and Rate Limiting:** Manage request traffic to prevent abuse and overloading.
- **Caching:** Built-in caching reduces backend load and improves response times.
- **Monitoring and Metrics:** Integrated with AWS CloudWatch for API usage and performance monitoring.

Amazon VPC Lattice

A fully managed application networking service designed to simplify the c**onnectivity, security, and management of communication between services** across **multiple** Amazon Virtual Private Clouds (**VPCs**) and accounts.

Key Features:

- **Service-to-Service Connectivity:** Seamless communication between services deployed in different VPCs or accounts.
- **Simplified Networking:** Eliminates the need for complex networking setups such as VPC peering, Transit Gateway configurations, or custom networking solutions.
- **Application-Aware Networking:** Allows defining services as logical abstractions.
- **Secure Communication:** Automatically encrypts traffic between services with TLS.
- **Observability and Monitoring:** Integrated with Amazon CloudWatch, AWS X-Ray, and others.
- **Policy Management:** Define policies for traffic routing, load balancing, and service authorization.
- **Integration with AWS Services:** Works seamlessly with other AWS services like AWS Lambda, ECS, EKS, and on-premises systems connected via AWS Direct Connect or VPN.

AWS CloudFront – Content Delivery Network

Highly scalable and secure Content Delivery Network (CDN) service provided by AWS.

It enables businesses to deliver content (web pages, images, videos, APIs, and more) to users worldwide with low latency and high transfer speeds by caching content at strategically located edge locations.

AWS CloudFront – Content Delivery Network

Key Features:

- **Global Edge Network:** Operates a vast network of edge locations and regional edge caches globally to bring content closer to end users. Reduces latency by serving cached content from the edge.
- **Caching and Acceleration:** Serves cached versions of **static** assets such as images, CSS, and JavaScript. Optimizes **dynamic**, non-cacheable content delivery by leveraging AWS edge locations. Accelerates **API** traffic for applications, reducing round-trip latency.
- **Security:** Integrated with AWS Shield Standard for DDoS protection. Supports TLS/SSL encryption for secure data transmission. AWS Web Application Firewall (WAF) integration for protection against common threats such as SQL injection or cross-site scripting (XSS)

AWS CloudFront – Content Delivery Network

Key Features:

- **Customizable Behavior:** Define caching policies, origin settings, and request/response headers. Use **Lambda@Edge** or **CloudFront Functions** for running custom code at the edge to personalize content or modify HTTP requests/responses.
- **Integration with AWS Services:** Works seamlessly with services such as Amazon S3 (as an origin for static files), Elastic Load Balancing (ELB), and EC2. Supports custom origins, including on-premises servers.

AWS CloudFront – Content Delivery Network

Key Features:

- **Real-Time Metrics and Monitoring:** Integrated with Amazon CloudWatch for logging and monitoring. Provides detailed metrics such as cache hit/miss rates, latency, and data transfer volumes.
- **Geo-Targeting:** Delivers region-specific content based on users' geographic locations. Restricts access to content using geo-restriction.
- **Cost Optimization:** Uses regional edge caches to reduce requests to the origin, lowering data transfer and origin workload costs. Offers pay-as-you-go pricing based on data transfer and HTTP/HTTPS requests.

AWS Glue

A fully managed data integration service that simplifies the process of **preparing and loading data** for analytics, machine learning, and application development.

It automates the tasks of discovering, cataloging, cleaning, transforming, and moving data between various data sources and destinations.

AWS Glue

Key Features:

Data Catalog: Centralized Metadata Repository. Automatic Schema Discovery (Glue Crawlers). Integration with other AWS services (S3, Athena).

ETL (Extract, Transform, Load): Serverless ETL. Supports Python and Scala. Visual interface for designing, running, and monitoring ETL workflows.

Data Preparation: Data Cleaning and Normalization. Transformations.

Job Orchestration: Workflows for job dependencies and DAGs. Triggers based on events, schedules, or job completions.

Integration with Other AWS Services: Compatible with S3, Redshift, Athena, RDS, DynamoDB, Lake Formation, and others.

AWS Glue

Key Features:

Glue DataBrew: A no-code visual interface for data profiling, cleaning, and transformation. Allows users without extensive coding expertise to prepare datasets interactively.

Built-in Machine Learning: FindMatches (ML-based capability to identify duplicates or similar records in datasets). Data Profiling (Provides insights into data quality and patterns).

Streaming Support:

- Process streaming data in near real-time using AWS Glue Streaming.
- Integrates with streaming services like Amazon Kinesis and Kafka.

Amazon Athena

A serverless query service that allows you to **analyze data** directly in Amazon S3 using **SQL**. It's designed to make data querying simple and cost-effective without requiring the setup of infrastructure or a traditional database.

Key Features:

- **Serverless Querying**
- **Standard SQL Support**
- Broad Data Format Support (CSV, JSON, Parquet, ORC, Avro, and others).
- Integration with AWS Services
- Security: Integrated with AWS IAM, KMS, and column level access.

Amazon Redshift Spectrum

A feature of **Amazon Redshift** that enables the user to query data directly in **Amazon S3** without having to load it into Amazon Redshift first.

It extends the power of Redshift by allowing the user to run **SQL queries** across both Amazon Redshift and external data stored in Amazon S3, enabling data lake analytics and **hybrid workloads**.

Amazon Kinesis

A suite of services for **real-time** data streaming, processing, and analytics at scale.

Enables the user to collect, process, and analyze streaming data such as video, audio, log files, clickstreams, and more, to gain insights and take actions in real-time.

Kinesis is designed to handle high-throughput, low-latency data streams, allowing applications to process and analyze the incoming data continuously.

Amazon SageMaker Serverless

A serverless machine learning (ML) compute environment offered by Amazon SageMaker, AWS's fully managed machine learning service.

It allows the user to run machine learning inference workloads without needing to manage the underlying infrastructure. With SageMaker Serverless, you don't have to worry about provisioning or scaling the underlying infrastructure for inference; it automatically adjusts based on demand.

This is ideal for variable or unpredictable workloads that require the flexibility to scale automatically and efficiently.

AWS Rekognition

A fully managed **image and video** analysis service offered by AWS that uses machine learning (ML) to identify objects, people, text, scenes, activities, and even detect inappropriate content in images and videos.

It provides pre-trained models that allow developers to easily integrate computer vision capabilities into their applications without needing to have deep knowledge of machine learning or computer vision.

AWS Transcribe and Polly

AWS Transcribe: Speech-to-Text Service

Fully managed automatic speech recognition (ASR) service that allows developers to convert **speech into text**.

It is used for applications that need to transcribe audio content from multiple languages into written form, enabling features such as voice search, real-time transcription, and subtitling.

AWS Polly: Text-to-Speech Service

A fully managed **text-to-speech** (TTS) service that uses deep learning to convert text into lifelike speech.

It enables developers to create applications with natural-sounding voices in **various languages and accents**, enhancing user experiences with voice interaction.

AWS CloudWatch

A comprehensive monitoring and observability service that helps users **collect, monitor, and analyze** data from the cloud resources and applications in real time.

CloudWatch allows users to track **performance**, **operational health**, and system **metrics** to ensure that your applications are running efficiently, with the ability to respond to any issues as they arise.

AWS X-Ray

A fully managed service that helps developers and DevOps teams analyze and debug distributed applications, particularly those built using microservices.

It provides **detailed insights** into the performance and health of your applications by **tracing requests** as they travel through various services and resources in your AWS environment.

AWS X-Ray helps pinpoint bottlenecks, detect errors, and gain a better understanding of the behavior and performance of the application.

Module 8 Conclusion

- Serverless services are becoming a main trend in cloud computing due to scalability, simplification, efficiency, flexible billing, and less technical expertise required.
- Heavily relying on serverless services is a major reason for vendor lock-in in the long term.
- Relying on serverless services empowers the organization with tools to focus on innovation, continuous learning, and fast adaptation.