	Amazon Web Services	Microsoft Azure	Google
Introduction	Amazon offers various serverless services to their customers. They are divided into 3 following categories 1. Compute : AWS Lambda, AWS fargate 2. App Integerations : AWS step function, Amazon event bridge, Amazon Simple Queue Service etc. 3. Data Stores : Amazon S3, Dynamo DB etc.	Microsoft Azure offers various different options for serverless services following are the categories- 1. Compute : The compute services in azure offers to write code and call out specific functions without worrying about the infrastructure. It also provides with serverless microservices and serverless environments. 2. Dev op and Dev tools : It provides the means to build serverless apps, using the various serverless environments. It also provides various CI/CD operations to run serverless. 3. Databases : It provides serevrless options for relational as well as non relational data bases. Azure SQL Databases and cosmo Db comes under it 4. It provides storage for unstructured data and carry out various operations on it in a serverless environment.	Google Cloud Tech offers various different options for serverless services following are the categories- 1. Compute options to write code and call functions. google cloud function, app engine and cloud run are a part of it. 2. Developer tools : Firbase development are offerred by google its kind of a sister concern to GCP but offers various development tools. 3. Storage : GCP offers no sql databses as well real time nosql data bases
Performance and Operational Limits	1. Compute : It takes cold start up time of few miliseconds. The time will be different for different languages. languages like pyhon and node js take few miliseconds, whereas java may take upto hundreds of miliseconds. Adding VPI may increase the startup time by 10s of ms. Where as larger files of size more than 50 mb may take 100s of seconds to initialize 2. Databases : It takes single digit mili seconds to perform data operations whereas files greater than 400 kbs may take higher time. This is specific to dynamo DB. 3. It takes 300ms to run APIs and Web applications but the time could be reduced by caching.	1.Compute - Azure Functions : Runtime: Various runtime versions depending on the language. For example, .NET Core 3.1, Node.js 12, Python 3.8, etc. Maximum Execution Time: 5 minutes by default for the Consumption plan, but it can be extended to 10 minutes. The Premium plan has no execution time limit. Concurrent Executions: Depends on the plan. The Consumption plan has a dynamic scale, while the Premium plan offers enhanced performance and VNET integration. 2. Database - Azure Cosmos DB: Latency: Single-digit millisecond read and write latencies. Throughput: Can scale to millions of transactions per second. 3. Azure API Management : Consumption Tier: Requests Per Second (RPS): 2,500 RPS Scaling: Automatic, event-driven Cache: Not supported Premium Tier: RPS: Starts at 6,000 RPS for the first unit; each additional unit adds 4,000 RPS (up to 32,000 RPS for 10 units) Capacity: Up to 10 units Cache: 10 GB per unit	1. Compute - Google Cloud Functions: Concurrency: Each function instance handles only one request at a time. Google Cloud Functions automatically scales by deploying multiple instances of your function. Memory Allocation: Ranges from 128MB to 4GB. Timeout: Maximum execution time is 540 seconds (9 minutes) per request. Deployment Size: Deployment packages must be less than 540 MB (zipped) and 2 GB (unzipped). 2. Databases - Firestore (a serverless NoSQL database): Write Rate: Up to 10,000 writes per second per database. Read Rate: Up to 10,000 reads per second per database. Indexed Queries: Performance is proportional to the result set size, not data set size. This means a search for 10 items in a collection of 1 million is as fast as a search for 10 items in a collection of 100. Document Size: Maximum size for a document is 1 MiB. 3. API - Google Cloud Endpoints: Requests Per Second (RPS): By default, Endpoints services are limited to 2 million requests per day, and 3,000 requests per minute (RPM). You can request an increase for these limits. Backend Timeout: Default is 20 seconds, but you can configure it. Payload Size: Maximum request and response body size is 32 MB

Scalability	1. AWS allows 1000 concurrent executions per region for accounts that have no active EC2 instances. This limit can be increased upon request. For accounts that have one or more active EC2 instances, the Lambda service limit is set to 4500 concurrent executions per region. 2. Dynamo DB : Provisioned Mode: Up to 40000 read capacity units and 40,000 write capacity units for tables. On-Demand Mode: Flexible scaling to handle up to double the previous peak traffic. 3. Storage: Unlimited storage (Dynamo DB) 4. 10000 requests per second (RPS) for REST APIs. 5000 RPS for HTTP APIs.	1. Azure Functions: Dynamic Scaling: Azure Functions on the Consumption plan scale automatically based on the number of incoming events 2. Azure Cosmos DB: Global Distribution: Turnkey global distribution to any number of Azure regions with a click. Provides active-active and active-passive configurations. Partitioning: Uses partition keys to distribute data and achieve high availability and low latency. Request Units (RU): Throughput can be provisioned in terms of RUs. Can be scaled up or down based on demand.. 3. Units: Depending on the tier, you can add more units to handle more RPS (Requests Per Second). For instance, the Premium tier supports up to 10 units. Multi-Region Deployment: In the Premium tier, you can deploy the API Management service across multiple Azure regions. Isolated Tier: Provides a dedicated environment that can be tailored specifically for extensive and unique workloads.	1. Cloud Functions: Scales automatically in response to the rate of incoming requests. There's no pre-set limit to the number of function instances that can run simultaneously. 2. Firestore: Designed to scale automatically to millions of concurrent connections and billions of writes per day. Distributed architecture allows for global scalability.
Cost	1. Compute : Invocation Cost: $0.20 per 1 million requests, and then $0.0000002 per request after that. Duration Cost: From $0.0000002083 for every GB-second 2. Dynamo DB : Provisioned Capacity: Write Capacity Unit (WCU): $1.25 per WCU per month. Read Capacity Unit (RCU): $0.25 per RCU per month. On-Demand Capacity: Write request: $1.25 per million write request units. Read request: $0.25 per million read request units. 3. HTTP APIs: $1.00 per million API calls received, up to the first 300 million, and then $0.90 per million thereafter.	1. Azure Functions: Consumption Plan: Execution cost: Approx. $0.20 per million executions Resource cost: Approx. $0.000016/GB-s (Gigabyte seconds) Premium Plan: Starts at around $0.013/hour for the smallest instance (EP1) 2. zure Cosmos DB: Provisioned Throughput: Approximately $0.008/hour per RU/s (Request Units per second), with a minimum of 400 RU/s per container or 400 RU/s per database shared among all containers. Database Storage: Approximately $0.25/GB-month 3. Azure API Management: Consumption Tier: Approx. $3.50 per million calls Additional cost for extra compute (~$0.0125/hour per additional unit) Basic Tier: Approx. $0.02/hour Standard Tier: Approx. $0.10/hour	1. Cloud Functions: Pricing is based on invocations, compute time, and network resources: Invocations: Around $0.40 per million invocations. Compute Time: The price depends on the amount of memory and time a function runs, ranging from $0.0000025 to $0.0000160 per GHz-seconds. 2. Firestore: Document Reads: Around $0.06 per 100,000 reads. Document Writes: Around $0.18 per 100,000 writes. Storage: Around $0.18 per GB/month. 3. Endpoints Requests: First 2 million requests per month: Free. 2-5 million requests per month: $3.00 per million requests. 5+ million requests per month: $1.50 per million requests.

Reliability	1. AWS Lambda :Multi-AZ Deployments, Built-in Fault Tolerance, Automatic Scaling 2. Dynamo DB : Backups and Point-in-Time Recovery, Global Tables	Azure Functions has a 99.95% uptime SLA (Service Level Agreement) when in the Premium plan. In the Premium plan, you can deploy your function app to multiple regions to ensure higher availability. You can integrate with Azure Blob Storage for logging and backup.	1. Google Cloud Functions: Redundancy: Deployed across multiple locations and zones to ensure high availability. Integration with Cloud Monitoring and Cloud Logging allows for real-time insights into function performance and potential issues. 2. Firestore: Multi-Region Replication: Data is automatically replicated in multiple regions, ensuring high availability and disaster recovery. Strong Consistency: Even though it's a NoSQL database, Firestore offers strong consistency, which ensures reliable data reads after writes.
Availability	1. AWS Lambda: Available in most AWS regions, including all major ones like US East (N. Virginia), US West (Oregon), EU (Ireland), and Asia Pacific (Tokyo). 2. Amazon DynamoDB: Present in all standard AWS regions.	Data Centers: Azure has data centers worldwide, allowing for geo-redundancy and failover capabilities. Monitoring & Diagnostics: Azure provides monitoring services like Azure Monitor and Application Insights to detect and diagnose issues, improving reliability.	Included in reliability
Security	Identity access management and encryptions are embeded in the serverless environments	1. Azure Functions: Identity and Access Management: Integration with Azure Active Directory (Azure AD) allows for role-based access control (RBAC) to the Azure Functions. Encryption: Data at rest is encrypted using Azure Storage Service Encryption. Data in transit is encrypted using SSL/TLS. Networking: With the Premium plan, Azure Functions can be integrated into Virtual Networks (VNET) for enhanced network security. 2. Azure Cosmos DB (Serverless Tier): Identity and Access: Uses Azure AD-based authentication and RBAC. Encryption: All data is encrypted at rest using service-managed or customer-managed keys and in transit using SSL/TLS. Firewall: Supports IP firewall rules to restrict access.	Identity access management and encryptions are embeded in the serverless environments

AWS serverless options

I have taken a deeper look into aws's serverless options. There are 2 broad categories through which AWS offers its services. I went to research upon all the broad categories and how they evolved in the last 5 years. I have put the specific numbers in my comparison and tried to keep a general overview about my research here. There are different services/functions provided by aws in those services. Following are the 4 broad categories :

- **Compute** : It basically provides a platform to write code and call individual functions, without setting up or worrying about the infrastructure or environment needed to run the code. It also provides a structure to manage the flow of data.

Amazon web services provides 2 significant computing options which are widely used :

1. **AWS Lambda** : Run programs without creating or controlling infrastructure. Automatically respond to code execution requests at any scale, from a dozen events per day to hundreds of thousands per second. It focuses on event driven execution at the time of function being called. First and foremost the lambda function is being designed to make changes to the programs. Once that's done, the lambda function is being triggered to run the specific function or the piece of code required to run. You specify the amount of memory allocated to your Lambda function. AWS Lambda allocates CPU power linearly in proportion to the amount of memory configured. Following are the evolutions that has happened over the last 5 years :
    - Initially, the maximum execution time for Lambda functions was 5 minutes. AWS later increased this limit to 15 minutes, allowing for longer-running serverless workloads.
    - Introduced in 2019, this feature allows developers to pre-warm a specified number of Lambda function instances to help with consistent start-up times, thus addressing the cold start issue.
    - In 2020 AWS Lambda got the capability to mount Amazon Elastic File System (EFS) volumes, making it suitable for workloads that require shared or persistent file storage.
    - AWS Lambda has added support for newer versions of programming languages, like Node.js, Python, Ruby, and Java.

- Lambda Extensions allows users to more easily integrate Lambda with their preferred monitoring, security, and governance tools.
- AWS changed the billing granularity for Lambda from 100 milliseconds to 1 millisecond
- In 2020, AWS added support for packaging and deploying Lambda functions as container images

2. **AWS Fargate :** AWS Fargate is a serverless compute engine for containers. It allows you to run containers without having to manage the underlying infrastructure, such as the servers. When defining your container, you simply specify the CPU and memory requirements. Fargate allocates the appropriate amount of underlying resources to meet these requirements. Fargate is deeply integrated with many AWS services like Application Load Balancer, Amazon VPC, AWS IAM, and Amazon CloudWatch. Following are the evolutions that has happened over the last 5 years :
   - In 2019, AWS announced Fargate support for Amazon EKS, allowing users to run Kubernetes pods
   - In 2019, Fargate Spot instances allow users to take advantage of unused Fargate capacity at up to a 70% discount, suitable for fault-tolerant and flexible applications.
   - AWS improved the networking capabilities of Fargate, introducing features like the ability to assign Elastic IP addresses to Fargate tasks and support for AWS PrivateLink
   - In 2020, AWS introduced the capability for Fargate tasks to use Amazon Elastic File System (EFS), allowing for persistent storage solutions with Fargate.
   - Fargate introduced support for larger container images and increased the size limits

- **Databases** : AWS offers a broad suite of database services to fit different needs. These services cover relational databases, non-relational databases, in-memory data stores, and more.

  1. Dynamo DB : Unlike traditional relational databases, DynamoDB is a NoSQL database, which means it provides a flexible schema and is

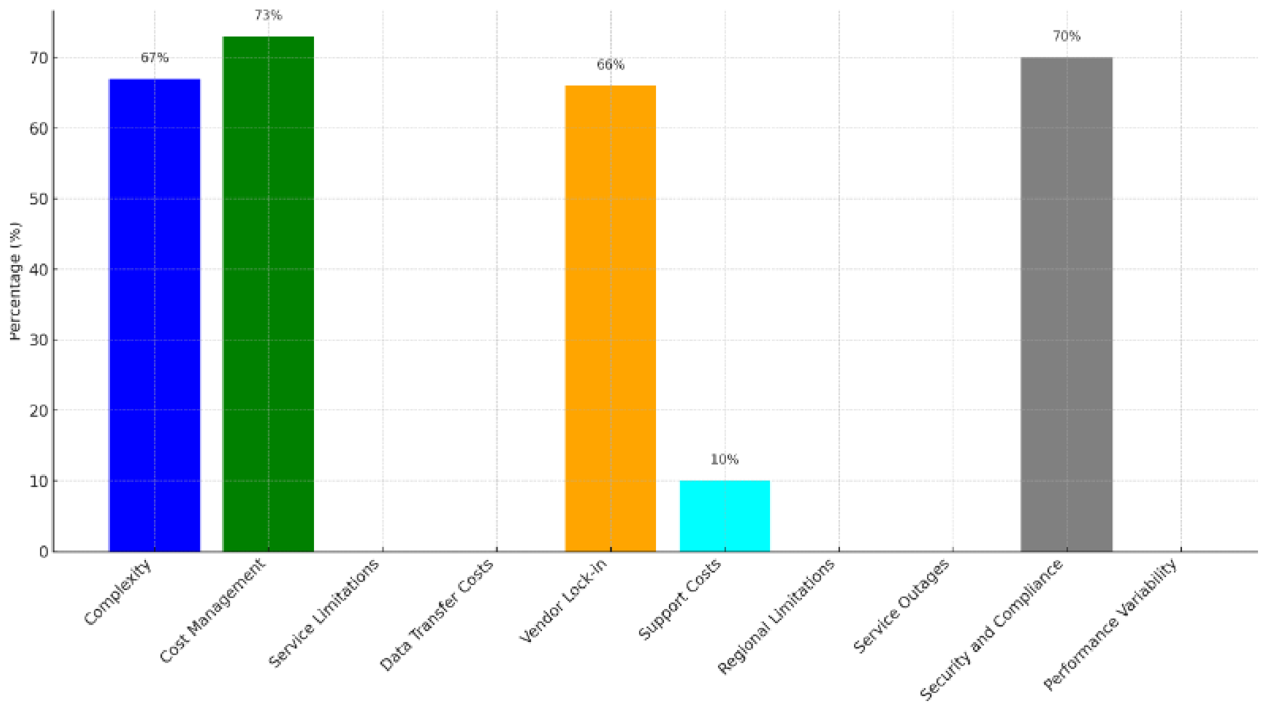
optimized for a wide variety of data models. DynamoDB is designed to handle large amounts of data and traffic. It can scale to handle more than 10 trillion requests per day DynamoDB is designed to handle large amounts of data and traffic. It can scale to handle more than 10 trillion requests per day Following are the evolutions that has happened over the last 5 years :

- AWS added on-demand backup and point-in-time recovery (PITR) capabilities in 2017, allowing users to create full table backups and restore data from a specific point in time.
- To provide more consistent performance, DynamoDB introduced adaptive capacity
- This feature, introduced in 2019, provides an in-depth view of a table's traffic trends and helps identify the most frequently accessed keys.
- AWS enhanced the integration between DynamoDB Streams and AWS Lambda, enabling more advanced, serverless data processing workflows.
- ver time, AWS increased the item size limit in DynamoDB from 400 KB to 1 MB
- Introduced in 2020, this feature allows users to export data from DynamoDB tables to Amazon S3 without needing to write custom code or manage the export process.

2. AWS RDS : Relational Database Service is a fully managed relational database service provided by AWS. It offers a distributed relational database engine that facilitates the setup, operation, and scaling of a relational database in the cloud. Following are the evolutions that has happened over the last 5 years :
    - Introduced in 2019, RDS Proxy is a fully managed, highly available database proxy that allows applications to pool and share database connections, enhancing scalability, resilience, and security.
    - Over the years, AWS has constantly improved the security of RDS by introducing features like encryption at rest and in transit, IAM authentication, VPC peering, and more.
    - AWS introduced read replicas for MySQL, PostgreSQL, MariaDB, and others to offload read traffic. The Multi-AZ deployments option was made available

- RDS has been integrated with many AWS services like Lambda, CloudWatch
- AWS introduced the ability to create an RDS or Aurora database clone. This feature is very useful in testing various use cases.

New Ideas/Features:



1. Complexity : According to a survey conducted by cloud guru, 67% of the users felt that the services are too complex. Similarly the gartner survey conducted in 2022 pours light on the fact that the complexity issues significantly arise due to misconfigurations by the user ends.
    - First and foremost there should be detailed documentation that takes the user through the aws services and how and what combinations arise complexity in the system.
    - Introduce a tool that allows users to design their own visual architecture and group the services they need. The deployment of the created architecture should happen directly through the tool.

2. Security : The Sophos State of Cloud Security report released in 2020 mentions that 70% of the security issues arised due to misconfigured cloud set up. Following are the solutions that could reduce the same
    - Create a visual tool with a graphical user interface that allows IAM relationships and permissions to be formed out.
    - Introduce a feature where it auto encrypts the data in the service , sort of a default encryption feature.
    - Use machine learning to quickly detect and block potential attacks with DDoS based on unusual activities by analyzing typical incoming traffic patterns.