**# Serverless Service Alternative report**

In the course, we learn a lot of different services in Azure cloud that uses for implementing event driven applications in our architecture. Now, I will explain the comparisons to the resources we use in Azure to resources to other cloud services such as AWS (Amazon Web Services) and GCP (Google Cloud Platform).

**## Azure Functions**

| Azure Service | AWS | Google Cloud |

| --- | --- | --- |

| Azure Functions | Lambda | Cloud Functions |

**### Overview**

-   **\*\*Azure Functions\*\***: It is a serverless compute services that allows cloud developers to run sample code that responds to events without managing the cloud infrastructure.

-   **\*\*AWS Lambda\*\***: Another serverless compute services that also allow cloud developers to run sample code in response to event without provisioning server. It is integrated with AWS resources like S3, DynamoDB, and API Gateway

-   **\*\*Google Cloud Functions\*\***: It is a serverless execution environment for event-driven applications that integrates with GCP services like Pub/Sub, Firestore, and Cloud Storage.

**### Core Features**

| Feature | Azure Functions | Lambda | Cloud Functions |

| --- | --- | --- | --- |

| Triggers | HTTP, Timer, Blob, Database, Event Grid, Service Bus, | API Gateway, S3, DynamoDB, SQS, EventBridge | HTTP, Pub/Sub, Cloud Storage, Firestore |

| Bindings | Input/Output bindings like Blob Storage and Cosmos DB) | uses SDKs to have limited binding | uses client libraries for limited binding |

| Languages | C#, JavaScript, Python, Java, PowerShell | Node.js, Python, Java, Go, Ruby, .NET | Node.js, Python, Go, Java, .NET |

**### Integration Options**

-   **\*\*Azure Functions\*\***: Integrates with Logic Apps, Service Bus, Event Grid.

-   **\*\*Lambda\*\***: Works with Step Functions, SNS, SQS, EventBridge.

-   **\*\*Cloud Functions\*\***: Connects to Pub/Sub, Cloud Scheduler, Firestore.

**### Monitoring & Observability**

-   **\*\*Azure Functions\*\***: Azure Monitor, Application Insights.

-   **\*\*Lambda\*\***: CloudWatch Logs, X-Ray.

-   **\*\*Cloud Functions\*\***: Cloud Logging, Cloud Trace.

**### Pricing Model**

-   **\*\*Azure Functions\*\***: Pay-per-execution with memory/time and free tier available.

-   **\*\*Lambda\*\***: Pay per request with GB-seconds and free tier included.

-   **\*\*Cloud Functions\*\***: Pay per invocation with compute time and free tier available.

**### Strengths & Weaknesses**

**- \*\*Azure Functions\*\*:** Strong bindings, Durable Functions for workflows.

-   **\*\*Lambda\*\***: Broadest event source integrations, mature ecosystem.

-   **\*\*Cloud Functions\*\***: Simpler for Google-native services, but fewer bindings.

**## Durable Functions**

| Azure Service | AWS | Google Cloud |

| --- | --- | --- |

| Durable Functions | AWS Step Functions | Google Cloud Workflows |

**### Overview**

-   **\*\*Durable Functions\*\***: Stateful serverless workflows in Azure.

-   **\*\*AWS Step Functions\*\***: Orchestration for Lambda and other AWS services.

-   **\*\*Google Cloud Workflows\*\***: Serverless workflow execution.

**### Core Features**

| **\*\*Feature\*\*** | **\*\*Durable Functions\*\*** | **\*\*AWS Step Functions\*\*** | **\*\*Google Cloud Workflows\*\*** |

| --- | --- | --- | --- |

| **\*\*Orchestration\*\*** | Chaining, Fan-out/Fan-in, Human Interaction | State machine-based, Retry policies | YAML/JSON-based workflow definitions |

| **\*\*State Management\*\*** | Built-in | Requires Amazon States Language | Limited native state handling |

**### Integration Options**

**#### Azure Durable Functions**

**\*\*Integrations\*\***: Works seamlessly with Service Bus, Event Grid, Blob Storage, Cosmos DB and supports HTTP APIs for external triggers.

**\*\*CI/CD\*\***: Deploy Azure DevOps, GitHub Actions, ARM templates.

**#### AWS Step Functions**

**\*\*Integrations\*\***: connect with Lambda, DynamoDB, ECS, SageMaker and supports API Gateway for HTTP triggers.

**\*\*CI/CD\*\***: Deploy AWS SAM, CloudFormation, Terraform.

**#### Google Cloud Workflows**

**\*\*GCP Service Integrations\*\***: Connects to Cloud Functions, Pub/Sub, BigQuery, Firestore and Supports HTTP endpoints (REST APIs).

**\*\*CI/CD\*\***: Deploy via gcloud CLI, Terraform.

**### Monitoring and Observability**

-   **\*\*Azure Durable Functions\*\***: used Azure Monitor and Application Insights to tracks orchestration history, execution time and failures.

-   **\*\*AWS Step Functions\*\***: used CloudWatch Logs and X-Ray to Provides execution timelines, state transitions and errors.

-   **\*\*Google Cloud Workflows\*\***: used Cloud Logging and Cloud Monitoring to logs workflow execution steps, duration, errors.

**### Pricing Model**

-   **\*\*Azure Durable Functions\*\***: Pay per execution time and memory (Consumption Plan)

-   **\*\*AWS Step Functions\*\***: Pay per state transition (Standard: $0.025/1K transitions; Express: $0.10/1M transitions).

-   **\*\*Google Cloud Workflows\*\***: Pay per step execution ($0.01/1K steps) and external HTTP calls ($0.40/1M calls).

**## Azure Logic Apps**

**### Overview**

-   **\*\*Azure Logic Apps\*\***: Low-code workflow automation.

-   **\*\*AWS Step Functions\*\***: Fast, short-lived workflows

-   **\*\*Google Cloud Workflows\*\***: YAML/JSON-based orchestration for serverless services.

**### Core Features**

| **\*\*Azure\*\***            | **\*\*Azure Logic Apps\*\***                            | **\*\*AWS Step Functions (Express)\*\*** | **\*\*Google Cloud Workflows\*\***              |

| -------------------- | ----------------------------------------------- | -------------------------------- | --------------------------------------- |

| **\*\*Triggers\*\***         | HTTP, Timer, Connectors (Office 365, SQL, etc.) | EventBridge, API Gateway, Lambda | HTTP, Pub/Sub, Cloud Scheduler          |

| **\*\*Integration\*\***      | 300+ SaaS/enterprise connectors                 | Tight Lambda integration         | GCP-native services (Pub/Sub, BigQuery) |

| **\*\*State Management\*\*** | Stateless (Standard) / Stateful                 | Stateless (Express mode) | Limited state support                   |

### Integration Options

* \*\*Azure Logic Apps\*\*: The other cloud services that it can connect to are Azure Functions, Service Bus, Event Grid, Blob Storage, and Cosmos DB. It can also connect to Microsoft SaaS applications such as Office 365, Salesforce, SharePoint, Dynamics 365, Dropbox, and GitHub.
* \*\*Step Functions\*\*: It can connect to Lambda, ECS, Fargate, Batch, SQS, SNS, EventBridge, DynamoDB, S3, Glue, Athena, EMR.
* \*\***Google Cloud Workflows**\*\*: It can connect to cloud serverless applications such as Cloud Functions, Cloud Run, and GKE. It can also connect to cloud data resources such as BigQuery, Pub/Sub, Firestore, and Cloud Storage.

### Monitoring & Observability

* \*\*Azure Logic Apps\*\*: It has built-in monitoring tools such as Azure Monitor and Log analytics to track run history, trigger status and action execution times and use KQL to analyze logs. It also has a Built-in Run History Dashboard that visualize workflow and shows the action for each step.
* \*\*Step Functions\*\*: It’s integrated with CloudWatch and X-Ray that can track state transition metrics, keeps a history of execution logs, and maps distributed transactions across different AWS resources such as Lambda/SQS/SNS.
* \*\*Google Cloud Workflows\*\*: Its observes by Cloud Logging that keeps logs of workflow execution steps, Cloud Monitoring that provides a basic dashboard, and reports errors by auto-detects failures and links to detailed execution logs.

**### Pricing Model**

-   **\*\*Azure Logic Apps\*\***: Pay per action/trigger

-   **\*\*AWS Step Functions\*\***: Pay per state transition

-   **\*\*Google Cloud Workflows\*\***:  Pay per step execution + external call costs.

**### Strengths & Weaknesses**

-   **\*\*Azure Logic Apps\*\***: Best for enterprise integrations (e.g., SharePoint, SAP).

-   **\*\*AWS Step Functions\*\***: Faster execution (Express mode), but fewer pre-built connectors.

-   **\*\*Google Cloud Workflows\*\***:  Simple YAML definition but limited to GCP ecosystem.

**## Azure Service Bus**

**### Overview**

-   **\*\*Azure Service Bus\*\***: Enterprise messaging (Queues & Topics/Subscriptions).

-   **\*\*AWS SQS/SNS\*\***: SQS (queues), SNS (pub/sub with topics).

-   **\*\*Google Pub/Sub\*\***: Global, scalable messaging service.

**### Core Features**

| **\*\*Feature\*\*** | **\*\*Azure Service Bus\*\*** | **\*\*AWS SQS/SNS\*\*** | **\*\*Google Pub/Sub\*\*** |

| **\*\*Messaging Patterns\*\*** | Queues, Topics (Pub/Sub) | SQS (Queues), SNS (Pub/Sub) | Pub/Sub only |

| **\*\*Ordering\*\*** | FIFO (Premium tier) | FIFO queues (SQS) | No native FIFO (requires client-side handling) |

| **\*\*Dead-lettering\*\*** | Supported | Supported (SQS) | Supported |

### Integration Options

* \*\*Azure Service Bus\*\*: It integrate with Azure Functions, Logic Apps, Event Grid by AMQP 1.0, HTTP/REST.
* \*\*AWS SQS/SNS\*\*: It integrate with Lambda, Step Functions, EventBridge by HTTP/S for SQS and HTTPS (SNS).
* \*\*Google Pub/Sub\*\*: It integrate with Cloud Functions, Dataflow, and Workflows with HTTP/S, Grpc.

### Monitoring & Observability

* \*\*Azure Service Bus\*\*: To monitor the resource is to use Azure Metrics and Logs. To analyze logs, use Diagnostic Logs. To have alerting tool, use Azure Monitor Alerts.
* \*\*AWS SQS/SNS\*\*: To monitor the resource is to use metrics tool from CloudWatch. To analyze logs, use Logs tool from CloudWatch. To have alerting tool, use Alarms tool from CloudWatch.
* \*\*Google Pub/Sub\*\*: To monitor the resource is to use Cloud Monitoring. To analyze logs, use Cloud Logging. To have alerting tool, use Cloud Alerting.

**### Pricing Model**

-   **\*\*Azure Service Bus\*\***: Pay per operation + namespace fee.

-   **\*\*AWS SQS/SNS\*\***: Pay per message (SQS) or notification (SNS).

-   **\*\*Google Pub/Sub\*\***: Pay per message volume + throughput.

**### Strengths & Weaknesses**

-   **\*\*Azure Service Bus\*\***: Advanced features (sessions, dead-lettering), but complex pricing.

-   **\*\*AWS SQS/SNS\*\***: Simpler pricing, but no built-in topic filtering (requires SNS + Lambda).

-   **\*\*Google Pub/Sub\*\***: Fully managed, global, but lacks FIFO without workarounds.

**## Azure Event Grid**

**### Overview**

-   **\*\*Azure Event Grid\*\***: Event routing for serverless and Azure services.

-   **\*\*AWS EventBridge\*\***: Schema Registry, cross-account event bus.

-   **\*\*Google Eventarc\*\***: Event-driven architecture for GCP services.

**### Core Features**

| **\*\*Feature\*\*** | **\*\*Azure Event Grid\*\*** | **\*\*AWS EventBridge\*\*** | **\*\*Google Eventarc\*\*** |

| **\*\*Event Sources\*\*** | Azure services, custom topics | AWS services, SaaS partners | GCP services, Audit Logs |

| **\*\*Targets\*\*** | Functions, Logic Apps, Webhooks | Lambda, SQS, SNS, Step Functions | Cloud Run, Functions, Workflows |

| **\*\*Schema Support\*\*** | Custom schemas | Schema Registry (discovery) | Limited (CloudEvents) |

### Integration Options

* \*\*Azure Event Grid\*\*: It integrate inputs from different Azure resources and custom topics and integrate output to Functions, Logic Apps, Webhooks.
* \*\*AWS EventBridge\*\*: It integrate inputs from different AWS resources and different SaaS resources and integrate output to Lambda, SQS, SNS, Step Functions.
* \*\*Google Eventarc\*\*: It integrate inputs from different Google resources and Audit Logs and integrate output to Cloud Run, Functions, Workflows.

### Monitoring & Observability

* \*\*Azure Event Grid\*\*: To monitor the resource is to use Azure Metrics. To analyze logs, use Diagnostic Logs. To have alerting tool, use Azure Monitor Alerts.
* \*\*AWS EventBridge\*\*: To monitor the resource is to use metrics tool from CloudWatch. To analyze logs, use Logs tool from CloudWatch. To have alerting tool, use Alarms tool from CloudWatch.
* \*\*Google Eventarc\*\*: To monitor the resource is to use Cloud Monitoring. To analyze logs, use Cloud Logging. To have alerting tool, use Cloud Alerting.

**### Pricing Model**

-   **\*\*Azure Event Grid\*\***: Pay per million operations.

-   **\*\*AWS EventBridge\*\***: Pay per million events (free tier available).

-   **\*\*Google Eventarc\*\***: Pay per event delivery attempt.

**### Strengths & Weaknesses**

-   **\*\*Azure Event Grid\*\***: Deep Azure integration, low latency.

-   **\*\*AWS EventBridge\*\***: Best for multi-account/multi-region (Event Bus).

-   **\*\*Google Eventarc\*\***: Tightly coupled with GCP services (e.g., Cloud Run).

**## Azure Event Hubs**

**### Overview**

-   **\*\*Azure Event Hubs\*\***: High-throughput event ingestion (streaming).

-   **\*\*AWS Kinesis\*\***: Real-time data streams (Kinesis Data Streams/Firehose).

-   **\*\*Google Pub/Sub\*\***: High-throughput messaging (similar to Event Hubs).

**### Core Features**

| **\*\*Feature\*\*** | **\*\*Azure Event Hubs\*\*** | **\*\*AWS Kinesis\*\*** | **\*\*Google Pub/Sub\*\*** |

| **\*\*Throughput\*\*** | Millions/sec (scalable) | Shard-based scaling (manual) | Auto-scaling |

| **\*\*Retention\*\*** | 1-7 days (configurable) | Up to 365 days (Kinesis) | 7 days (default) |

| **\*\*Integrations\*\*** | Stream Analytics, Functions | Lambda, Firehose, Redshift | Dataflow, BigQuery |

### Integration Options

* \*\*Azure Event Hubs\*\*: It integrate with Stream Analytics and Azure Functions by AMQP 1.0, Kafka, HTTP.
* \*\*AWS Kinesis\*\*: It integrate with Lambda, Firehose and Redshift by HTTP/S (Kinesis Data Streams).
* \*\*Google Pub/Sub (Streaming)\*\*: It integrate with Dataflow and BigQuery by HTTP/S, and gRPC.

### Monitoring & Observability

* \*\*Azure Event Hubs\*\*: To monitor the resource is to use Azure Metrics. To analyze logs, use Diagnostic Logs. To have alerting tool, use Azure Monitor Alerts.
* \*\*AWS Kinesis\*\*: To monitor the resource is to use metrics tool from CloudWatch. To analyze logs, use Logs tool from CloudWatch. To have alerting tool, use Alarms tool from CloudWatch.
* \*\*Google Pub/Sub (Streaming)\*\*: To monitor the resource is to use Cloud Monitoring. To analyze logs, use Cloud Logging. To have alerting tool, use Cloud Alerting.

**### Pricing Model**

-   **\*\*Azure Event Hubs\*\***: Pay per throughput unit (TU) + ingestion volume.

-   **\*\*AWS Kinesis\*\***: Pay per shard hour + data volume (Firehose additional cost).

-   **\*\*Google Pub/Sub\*\***:  Pay per message volume + throughput.

**### Strengths & Weaknesses**

-   **\*\*Azure Event Hubs\*\***: Best for Azure-centric streaming pipelines.

-   **\*\*AWS Kinesis\*\***: Flexible retention (Kinesis), but manual shard management.

-   **\*\*Google Pub/Sub\*\***: Simpler auto-scaling, but shorter retention.

**## Final Comparison Summary (Expanded)**

| **\*\*Service Category\*\*** | **\*\*Azure Service\*\*** | **\*\*AWS Equivalent\*\*** | **\*\*GCP Equivalent\*\*** | **\*\*Key Differentiator\*\*** |

| **\*\*Workflow Automation\*\*** | Logic Apps | Step Functions | Cloud Workflows | Azure: Connectors; AWS: Speed; GCP: YAML |

| **\*\*Messaging\*\*** | Service Bus | SQS/SNS | Pub/Sub | Azure: Enterprise; AWS: Simple; GCP: Global |

| **\*\*Event Routing\*\*** | Event Grid | EventBridge | Eventarc | AWS: Schema Registry; GCP: GCP-native |

| **\*\*Streaming\*\*** | Event Hubs | Kinesis | Pub/Sub | AWS: Long retention; GCP: Auto-scale |