



Serverless Computing – Revision Notes

☑ Serverless Computing – Quick Revision Notes

◊ Definition

- **Serverless Computing** is a cloud computing execution model where the **cloud provider dynamically manages the infrastructure**. You focus on **code**, not servers.

◊ Major Serverless Service Providers

1. **Amazon AWS** – *AWS Lambda*
2. **Google Cloud Platform** – *Google Cloud Functions*
3. **Microsoft Azure** – *Azure Functions*

◊ AWS Lambda

☑ Key Concepts

- **Event-driven** serverless platform by AWS.
- Automatically **manages infrastructure**; you only upload the **code**.
- Runs code **in response to AWS events** (e.g., S3, API Gateway, DynamoDB).
- **Pay only for execution time**.

☑ Use Cases

- **Background tasks** like image processing, notifications, etc.
- **Stateless function executions**.

☑ Process Steps

1. Upload code (Java, Python, C#, Go, etc.).
2. Define **triggering services** (e.g., S3, API Gateway).
3. Attach trigger to function.
4. Executes code **only when triggered**.
5. Charges only during execution.

☑ Components

- **Function**: Script/code that processes an event and returns a response.
- **Runtime**: Execution environment that supports different languages.
- **Event Source**: AWS service or custom event triggering Lambda.
- **Lambda Layers**: Shared code/dependencies/libraries.
- **Log Streams**: Used for custom logging for debugging and performance analysis.

◊ Google Cloud Functions

☑ Key Concepts

- Serverless execution for **event-based cloud services**.
- Automatically **scales** and **manages infrastructure**.
- You write **single-purpose functions** tied to cloud **events**.
- Works best in **stateless** architecture.

☑ Working

- Event occurs → Cloud Function is triggered.
- Event data is passed to the function.
- Function processes event → may call other APIs/services → completes.
- **Multiple event occurrences = multiple function invocations**.

☑ Trigger Types

- **HTTP**
- **Cloud Storage**
- **Cloud Pub/Sub**
- **Firebase**
- **Stackdriver Logging**
- **Firestore**
- **BigQuery**
- **Google Compute Engine**

☑ Terms

- **Event**: Something that happens (e.g., file upload).
- **Trigger**: Defines which function to run for which event.
- **Event Data**: Data passed into function upon trigger.

◊ Azure Functions

☑ Key Concepts

- **Serverless compute** from Microsoft Azure.
- Write **less code**, manage **less infra**, save costs.
- Primary building block: **Function** = code + config.

☑ Features

- Code written in **C#, Java, Python, JavaScript, PowerShell**.
- Use **function.json** for configuration.

- Auto-generated for compiled languages; manual for scripting.

☒ Benefits

- **Consumption Plan:** Pay-per-execution.
- **Premium/App Service Plans:** For high-demand workloads.

☒ Common Serverless Architectures

- REST APIs, mobile/web backends.
- Event/stream processing, IoT pipelines.
- Big data/ML integration.
- Pub/Sub event communication.
- SaaS integrations and automation pipelines.

☒ Key Comparisons

Feature	AWS Lambda	Google Cloud Functions	Azure Functions
Event-Driven	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Languages Supported	Java, Python, Go, C#	JS, Python, Go, Java	C#, Java, JS, Python
Pricing	Pay per execution	Pay per execution	Consumption/Premium Plan
Trigger Types	S3, API Gateway, etc.	Storage, Pub/Sub, HTTP	HTTP, Timer, Queue, etc.
Stateless Execution	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

☒ High Probability MCQ Points

- **Lambda** charges only when code executes → ☒
- **Google Cloud Functions** are best for **stateless** single-purpose code → ☒
- **Azure Functions config** for scripting requires manual function.json → ☒
- **AWS Lambda trigger sources:** S3, API Gateway, DynamoDB → ☒
- **Google Cloud Function triggers:** HTTP, Pub/Sub, Firebase → ☒
- **Lambda Layers** used for shared code/libraries → ☒
- **Event + data = Trigger** in Google Cloud → ☒
- **Azure Functions support HTTP, Timer, Event Grid** triggers → ☒
- **No need to provision infra** in any of the three → ☒
- **All providers follow event-driven model** → ☒