Compare serverless options from AWS, Google & Azure

- **AWS Lambda**: As the most established platform, AWS Lambda provides extensive language support and integrates seamlessly with AWS services. Its 15-minute runtime limit supports more complex tasks.
- **Google Cloud Functions**: Google's offering is known for its developer-friendly interface and clear documentation, making it ideal for rapid development. However, its 9-minute runtime limit can be restrictive.
- **Azure Functions**: With a 60-minute runtime limit, Azure is best for long-running processes. It also supports PowerShell, catering to IT professionals and enterprise automation tasks.

Evolution of AWS Lambda's serverless offering over the last 5 year

AWS Lambda has continuously evolved, addressing developer challenges such as cold starts and resource limitations:

- **2019**: AWS introduced **Lambda Layers** to share libraries and dependencies across functions, reducing code duplication.
- **2020**: AWS added **Container Image Support**, allowing functions to be packaged as Docker containers for custom runtimes.
- **2021**: AWS increased resource limits to 10GB of memory and 6 vCPUs, supporting more demanding workloads.
- **2022**: AWS launched **Function URLs**, making it easier to call Lambda functions without needing API Gateway.
- **2023**: AWS introduced **ARM64 support**, improving performance and lowering costs for ARM-based workloads.
- **2024**: AWS rolled out **SnapStart for Java**, reducing cold start times by pre-warming Java functions.

AWS Lambda's evolution reflects a commitment to enhancing performance, expanding flexibility, and addressing cold starts—one of the biggest hurdles in serverless architecture.

New Feature Proposal:

I would like to name this feature as **AWS Lambda CodePath** and this would provide developers with real-time feedback on how their code runs in AWS Lambda. After you change and deploy your code, this tool will analyze how it performs and give you detailed reports. These reports would show things like:

- **Function performance** (like cold start times, memory usage, latency).
- **Cost insights** (like how much a specific function costs to run based on recent changes).

- **Suggest improvements**, such as reducing costs or improving response times.

Support Argument For Proposal

1. **I worked as a Full-Stack Developer and faced these issues**: Full-stack developers often work on both front-end and back-end components. In serverless development, understanding how a single code change affects the overall performance can be tough. Developers often don't get enough feedback on how their code impacts things like latency or memory usage in production.
2. **Real-Time Feedback**: Currently developers don't have tools that connect **code changes** directly to **performance metrics**. With **AWS Lambda Code Insights**, you would get feedback right after your code is deployed, showing you how the changes impacted your app's performance.
3. **Optimized for Full-Stack Teams**: For teams using Git (which you're familiar with), this feature would integrate directly into the Git workflow. When you push new code, it automatically generates a report with:
    - How the code change affected memory usage or API response times.
    - If the function is under- or over-provisioned (using too many or too few resources).
    - Possible errors or bugs caused by the new code.

Benefits for Developers Like US:

- **Better Performance Tracking**: You would know right away how your code is performing, helping you quickly fix issues or optimize functions.
- **Cost Savings**: If your Lambda functions are using too many resources (which increases costs), you'll get suggestions on how to reduce them.
- **Easier Debugging**: The tool would highlight where performance bottlenecks or errors are happening, tied directly to your recent code changes.

Why This is Unique:

This feature is specifically built for **full-stack developers** working on **serverless applications**. It focuses on giving you clear, actionable feedback on how your code affects AWS Lambda's performance and cost. It's simple, integrates with tools you already use (like Git), and would save developers time by automating performance analysis and recommendations after every deployment.

This idea directly aligns with your experience as a full-stack developer using Git, serverless tech, and performance benchmarking, making AWS Lambda even more powerful for developers like you.