**💡 Stop Writing Lambda Functions Just to Move Data Around**

Here's a costly mistake I see engineers making: using Lambda as a data courier between services.

If you're simply moving data from A to B, you're paying for compute you don't need—and maintaining code that AWS can handle natively.

**Enter: Amazon EventBridge Pipes** 🚀

Think of it as your intelligent data pipeline that connects sources to targets without custom integration code.

**Real-World Financial Example:**

Imagine you're building a real-time stock trading alert system:

📊 **The Flow:**

* Stock price updates stream into Amazon Kinesis (source)
* EventBridge Pipes filters for significant price movements (>5% change)
* Lambda enriches the data with historical volatility from your API
* Processed alerts land in SQS for your notification service (target)

**The Magic:** ✅ No custom polling code to maintain ✅ Built-in filtering—only pay for alerts that matter ✅ Lambda transforms data, doesn't transport it ✅ At $0.40 per million requests, it's incredibly cost-efficient

**Traditional Approach:** Multiple Lambda functions polling, processing, and forwarding **EventBridge Pipes:** One declarative configuration

The service supports SQS, Kinesis, DynamoDB Streams, and even third-party APIs as sources. Your Lambda function? It focuses on what matters—transforming and enriching your data, not babysitting data transfers.

If you haven't explored EventBridge Pipes yet, check out the AWS documentation. It's straightforward enough to master quickly, but powerful enough to become a go-to tool in your serverless architecture.

**Have you used EventBridge Pipes in production? Drop your use case below!** 👇

#AWS #Serverless #CloudArchitecture #EventDrivenArchitecture #FinTech #CostOptimization

<https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-pipes.html>