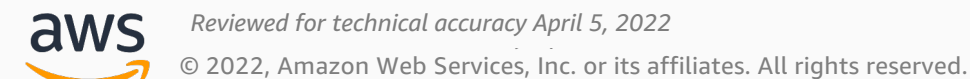


Companies large and small who sell their products on Amazon.com as first-party vendors want to not only understand prior demand trends but also anticipate future demand so they can improve On-Time-In-Full (OTIF) metrics and, as a result, maximize their revenue. This architecture programmatically sources vendor performance data from the Amazon.com platform through vended APIs, which can be helpful to consumer packaged goods (CPG) companies who are manually collecting data. The solution allows analysis to occur on what has happened, and also uses advanced machine learning (ML) to create future demand predictions that help to proactively plan operations in the weeks ahead.



## AWS Reference Architecture

- 1 **Amazon EventBridge** initiates scheduled tasks that help keep the data fresh with controlled latency.
- 2 **AWS Lambda** makes various API calls to the Amazon Selling Partner API, entitled by a named **AWS Identity and Access Management (AWS IAM)** role that is attached to the Amazon Vendor Central account.
- 3 The Amazon Selling Partner API first authenticates the calling API and then uses the IAM role to authorize access to a specific set of vendor codes.
- 4 Some responses from APIs are stored in **Amazon SQS** as a queue, and as part of a decoupled microservices architecture. Some response artifacts are landed directly on **Amazon Simple Storage Service (Amazon S3)** as Parquet objects (not shown here).
- 5 **AWS Lambda** is responsible for consuming newly received messages from the Amazon SQS queue. Lambda then transforms, persists, and creates notifications.
- 6 **Amazon SNS** receives notifications on certain data conditions such as product catalog changes. This could enable an email to interested parties, or trigger a manufacturer corrective response action.
- 7 **Amazon DynamoDB** acts as an intermediate database to store data that may see many mutations while in flight. This provides an effective way to accept create, read, update, delete (CRUD) operations from the publishing **Lambda** function, while maintaining data integrity.
- 8 A scheduled **AWS Glue** job can publish data to a serverless data lake for long-term persistence at scale.
- 9 **Amazon S3** is the persistence tier for a serverless data lake, able to serve time-series data at any scale, without capacity management.
- 10 **Amazon EventBridge** triggers a recurring **Amazon Forecast** routine managed by an **AWS Step Functions** workflow that sources demand history from the data lake to generate future demand predictions that become a new, reportable data lake asset.
- 11 **Amazon S3** data can be syndicated to a variety of consuming applications. **Amazon QuickSight** and **Amazon Athena**, respectively, provide examples of business intelligence and SQL capabilities.