AWS Whitepaper

Demand Forecasting

Demand Forecasting: AWS Whitepaper

Table of Contents

Demand Forecasting

Publication date: **September 23, 2022** (*Document revisions*)

Abstract

Amazon Web Services (AWS) customers look for easier, faster, more accurate, and more cost-effective ways to forecast the demand for their products, services, and materials. This whitepaper provides best practices, architectural patterns, technologies, and recommendations about demand forecasting on AWS. This paper addresses a wide array of readers, including technical professionals, non-technical professionals, and organizations with or without science teams.

We discuss general trends and AWS services, and architectures for wide variety of needs. You can use this paper to find the best solutions, next steps, and best practices specific to your business. The content is based on findings about industrials, manufacturing, Consumer Packaged Goods (CPG), retail, and utilities; but it is applicable to other industries.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

In the Machine Learning Lens, we focus on how to design, deploy, and architect your machine learning workloads in the AWS Cloud. This lens adds to the best practices described in the Well-Architected Framework.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

This document provides ways to build automations and data pipelines, and recommends AWS solutions and partner support to fit your business needs. It provides guidance on data sources and utilization of statistical and machine learning (ML)-based demand forecasting using managed AWS

technologies. It also addresses artificial intelligence/machine learning (AI/ML) technologies which do not require data scientists to be involved. The document also contains example architectures building custom solutions.

The first part of this document provides high-level information and background, such as common practices of demand forecasting in the industries. Next it introduces industry pain points to help you to identify your own business pain points.

In the second part, we provide solutions to address your business environments, skillsets, data residency, and business needs. The following section explores technical aspects of the solutions, including various reference architectures and patterns.

Demand forecasting: background and barriers to adopt AI/ML-based approaches

Background

Demand forecasting is a necessary capability for most industries. Demand forecasting touches everyone's lives on a daily basis. For example, demand forecasting ensures grocery shelves are stocked, packages are delivered on time, electricity generation meets the electricity load to keep our lights on, and wait times for our favorite restaurant delivery are short.

Demand forecasting is a field of predictive analytics, which attempts to forecast customer demand to optimize supply decisions. Demand forecasting methods are often split into two classes: *qualitative* and *quantitative* methods. Qualitative methods are based on subject matter expert (SME) opinion, while quantitative methods use data. The focus of this paper is on *quantitative* demand forecasting.

In quantitative demand forecasting, the basic assumption is that the demand of the factual past can be used to define future demand. Mathematically, historical demand is described by a time series, demonstrating a chronological sequence of logged observation points. By reflecting the specific features of this time series (pattern), this series is projected into the future using different forecasting methods and models.

While all businesses use some form of forecasting today, traditional methods such as rule-based forecasting, statistical methods such as extrapolation with regression analysis, or time series analyses may have limitations due to the increasing number of demand signals. The large size and variety of data available today was not factored into these older methods, and may be beyond their capacity. The use of ML, such as artificial neural networks (ANNs), has proven to be a preferable method of handling huge volumes of data. The main reason behind the accuracy of ML models is the complexity of the models and their ability to consider the amount of data, which may or may not be related to demand – a concept which is hard for statistical methods to grasp.

Customers often acquire large amounts of structured and non-structured data from internal and external resources, which makes ML forecasting a good solution for them. Market leaders make investments to bring in flexible, accurate ML-forecasting to all parts of their business. According to a recent study by McKinsey, organizations that have implemented ML forecasting techniques improved forecasting accuracy by 10 to 20 percent, which translated into a 5% reduction in

inventory costs and a 2-3% increase in revenue. AI/ML-based forecasting can increase your revenue as you make more informed decisions, prepare for upcoming changes, and invest better and more effectively based on the forecast.

However, there are some challenges to adopting AI/ML-based demand forecasting. There are various reasons and solutions for these pain points which we address in the following sections.

Challenges to adopting AI/ML-based demand forecasting

Even though there is growing interest by many organizations in investing in AI/ML capabilities for demand forecasting, quantifying and demonstration of the value of AI/ML remains a challenge. This has slowed widespread adoption of ML. Even if the organization decides to proceed with AI/ML-based demand forecasting, they can face challenges organizationally, technologically and skill-wise to realize the technology in their organization.

These challenges are grouped into five categories:

- **Adequate data to start with** — Demand is driven by multiple factors which may require a wide variety and a large set of data. In some cases, even if the customer has the expertise, they may not have collected enough data to accurately forecast the demand.
- **Data science expertise** — A company may not have necessary technical resources, such as data science expertise, available in their organization. Building in-house-grown models and the necessary data pipelines to perform recurring forecasting generally requires a dedicated team. The cost of building a data science team that has the expertise to build an in-house demand forecasting model can be high. For most organizations, this is not their business focus, so leadership may not provide the investment to proceed. A customer can always outsource this capability to data science vendors, but this comes with its own costs and operational dependencies.
- **Building competitive models** — Not all models are the same. A competitive model should be highly accurate to provide competitive advantage to the customer. Creating such a model requires heavy lifting efforts such as building the ANN model and tuning it, which can distract from the main work of the business without adding value.
- **Model lifecycle and ML operations (MLOps)** — A competitive model must be kept up to date. AI/ML models can drift over time, which can impact the model's accuracy. Having such a model is not a one-time effort, but a constant process of maintenance to keep it accurate and reliable. However, the data to train and keep the model updates are expected to be on a data pipeline which requires ingestion and storage layers. The pipeline is expected to work seamlessly with the

AI/ML models hosted in the consumption layer. Multiple data scientists and ML engineers may be needed to maintain the model. Continuous integration and continuous delivery (CI/CD) pipelines must be agile and high quality to keep AI/ML models and their endpoints up to date.

- **Incorporating and interpreting AI/ML-based demand insights into the decision cycles** — Eventually, the business data as well as the AI/ML inferred forecasted data should be ready to be *consumed* by various business stakeholders. A challenge is presented if there is a gap between the business and ML teams on how to best use and interpret the produced ML insights in the business decision-making processes, in a timely manner.

Addressing the challenges with demand forecasting solutions on AWS

This section dives deeper into the previously mentioned challenges, and provides solutions.

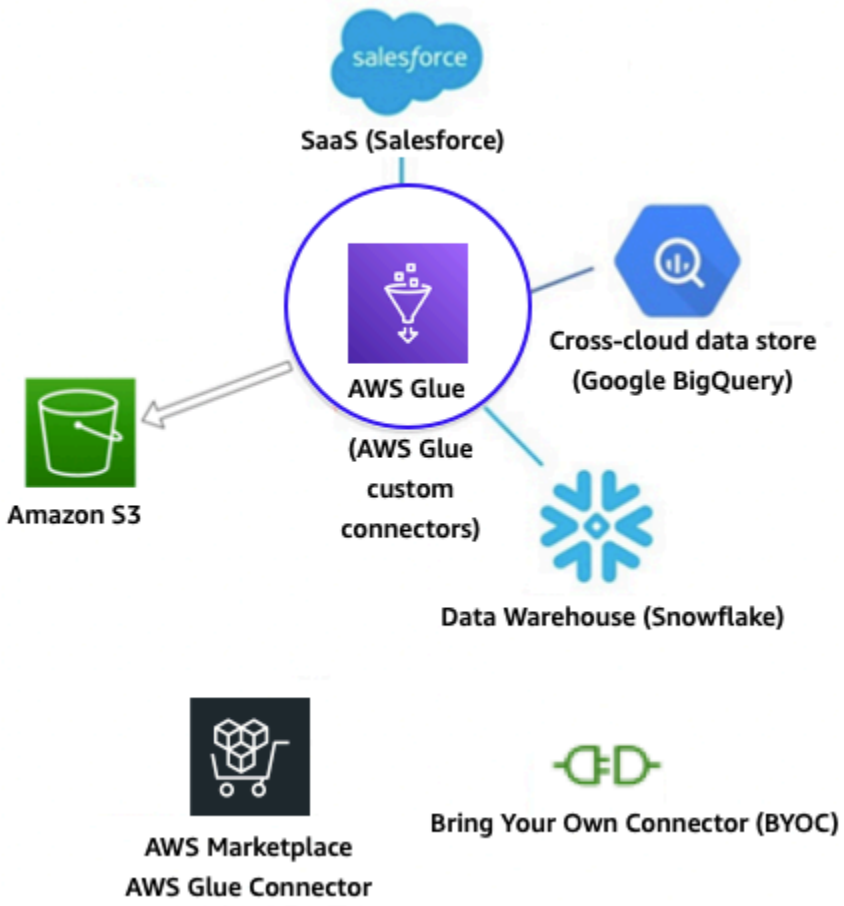
Adequate data to start: sources of data and data handling

In a well-architected solution, the data follows through a well-defined pattern of creation, ingestion, storage, and consumption layers where the consumption bears the forecasting stage. Whether the data is consumed by business partners or by data scientists, you may want to centralize the data to get more insights. In some cases, the data is created at the edge or by third parties. In these cases, customers can subscribe to new data to enhance their existing data. We provide third-party subscriptions to utilize the data without any hassle, including billing and data source management solutions.

Typically, the data required for demand forecasting is in the form of time series and metadata. You can use Amazon Simple Storage Service (Amazon S3) to store time series data and metadata typically in CSV format. You can directly use this data to employ Amazon SageMaker solutions which will be explained in the following section. If you are using Amazon Forecast for the demand forecasting, you can upload your time series and metadata data to Amazon Forecast. Follow the instructions in Prepare and clean your data for Amazon Forecast to prepare and upload your data to Amazon Forecast.

Solutions

Data is created at the edge, on your ecommerce site, or from social media by customers. You can use external connectors to direct data to your data storage on AWS. Use the AWS Marketplace to find other readily available connectors. This solution can be unified as a data lake on AWS, where you can not only use ML inference for demand forecasting, you can also perform data lifecycle management and analytics. If you have data on-premises, you can keep your data on-premises if you want, and add a connection to your data through AWS Storage Gateway. You can use data lakes on AWS to build a flexible and hybrid architecture.



External data ingestion

The other specific application for industrial, in-demand forecasting is the Internet of Things (IoT). Some of our customers have IoT devices on their production or manufacturing sites, with sensors or telemetry data coming from the devices. These customers may want to use their IoT data in their demand forecasting. You can utilize AWS IoT Analytics solutions to incorporate an IoT workload into the solutions provided in this document.

Customers may also look for external data resources to gather a wide-variety of data to forecast. These resources can include industry trends, society information, social media posts, and so on. You can subscribe to AWS Data Exchange to get third-party data to enhance your models. The advantage of this method is the ease of data utilization with APIs. Payment is easy, because your subscription billing will be paid to the third party as part of your AWS Billing and Cost Management.

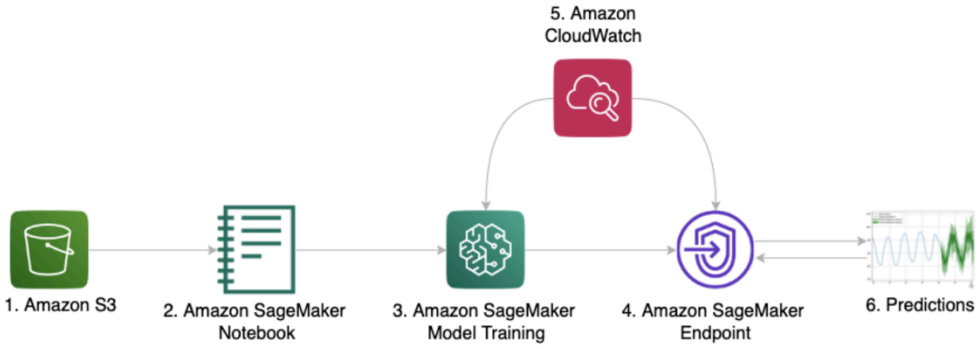
Data science expertise: empower your development teams with Amazon ML services

In this section, we address the data science expertise pain point, and the solutions for teams with and without data science expertise and pinpoint the services to cover a wide-variety of organizations.

Teams with data science expertise

Amazon SageMaker can empower customers with or without dedicated data science teams or developers. However, it is the recommended service for teams with data-science expertise. Amazon SageMaker brings agility, innovation, and automation to existing data science teams. SageMaker acts as the single source of all data-science related projects covering complete ML application lifecycle from data ingestion to model monitoring and lifecycle management. Amazon SageMaker comes with a fully-developed, user-friendly Integrated Development Environment (IDE) called SageMaker Studio. You can start reinventing your business by empowering your data-science teams with Amazon SageMaker through Studio IDE.

The following figure summarizes a common pattern of Amazon SageMaker experience for data scientists or developers who perform demand forecasting. It starts with training data located in an S3 bucket (step 1). Amazon SageMaker utilizes Jupyter Notebook (step 2). Here, you can use Amazon SageMaker SDK for Python. R practitioners can also use SageMaker. Later, the model is trained (step 3) and the endpoint is deployed (step 4). Here you can monitor and log your SageMaker deployment and endpoint using Amazon CloudWatch (step 5) and finally using input data for inference located in S3, the prediction requests can be sent to the SageMaker endpoint and make predictions (step 6).



A common pattern of using Amazon SageMaker – training data in Amazon S3 and an inference endpoint deployed for production.

Starting with a well-documented, organized, and well-architected solution can be the fastest and the productive step to take. Some data scientists may want to start with Amazon SageMaker deployment without an *empty-page*, meaning they may want to utilize a readily available, best practiced, and fully developed solution to start with.

Amazon SageMaker JumpStart provides developers and data science teams ready-to-start AI/ML models and pipelines. SageMaker JumpStart can be used as-is, because it is ready to be deployed. Also, you can consider incrementally training your data or modifying the code based on your needs.

For demand forecasting, SageMaker JumpStart comes with a pre-trained, deep learning-based forecasting model, using Long- and Short-Term Temporal Patterns with Deep Neural Network (LSTNet). LSTNet is a state-of-the-art multi-variate (made of multiple correlated time series data) forecasting model, as explained in Cornell University's paper, Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks.

Deep learning-based demand forecasting with LSTNet uses the short-term and long-term effects of the historical data, which may not be accurately captured by conventional time series analyses such as autoregressive integrated moving average (ARIMA) or exponential smoothing (ETS) methods. This is especially important in forecasting complex behaviors in real-life businesses such as energy industry or retail.

To start using the deep learning model (LSTNet-based) with JumpStart, refer to this documentation. The output of the model is a trained model which can be deployed on your inference selection. Recently, Amazon SageMaker has added serverless inference to have a cost model of *per inference* for unpredictable traffics. Alternatively, you can use Amazon SageMaker Inference Recommender to get a data-based recommendation for your endpoint.

Amazon SageMaker comes with state-of-the-art time series algorithms using deep learning. Along with LSTNet, SageMaker provides DeepAR. Similar to LSTNet, DeepAR is a supervised learning algorithm for forecasting one-dimensional time series recurrent neural networks (RNN). Generally, demand forecasting in CPG, manufacturing, and energy industries comes with hundreds of similar time series across a set of business metrics.

> **Note**
>
> Deep learning models require a large amount of data, whereas conventional approaches such as ARIMA can start with data less than 100 data points.

For example, you may have demand data for wide variety of products, webpages, or households' electricity and gas usage. For this type of application, you can benefit from training a single model jointly over all of the time series to a one-dimensional output, which is *demand.* DeepAR takes this approach. Another deep learning model for demand forecasting is Prophet due to its excellence in considering seasonality effects. Refer to Deep demand forecasting with Amazon SageMaker, which compares the performance, cost, and accuracy of these models in demand-forecasting. The decision for the best model may not mean the highest accuracy for all users.

For some, an adequate accuracy with lower latency and faster training is a better option than a high accuracy but more expensive model. To bring automation and innovation, your data science or ML teams may want to experiment with other models to see which model works the best. For more information, refer to the Amazon SageMaker Experiments – Organize, Track And Compare Your Machine Learning Trainings blog post.

Data scientists may want to perform advanced data preparation for their time series data built for ML purposes using Amazon SageMaker Data Wrangler, as explained in the Prepare time series data with Amazon SageMaker Data Wrangler blog post. Data Wrangler enables your data scientist to quickly analyze, visualize, transform, and clean data with ease.

For a constantly changing business environment, such as in the retail sector, the model can easily drift to show lower accuracy in time. Use SageMaker Model Monitor and then an auto-training with an MLOps through Amazon SageMaker Pipelines to speed up your model deployments and create a full pipeline to integrate CI/CD pipelines to your model development and deployments.

Teams without data science expertise

Use of SageMaker is possible for teams without data science expertise by using Amazon SageMaker Canvas. With Canvas, you can start utilizing predictive analyses for demand forecasting without any coding skills. Refer to Reinventing retail with no-code machine learning: Sales forecasting using Amazon SageMaker Canvas.

SageMaker Canvas brings drag-and-drop style user-friendly user-interface. Canvas gives a complete lifecycle of AI/ML deployment: connect, access, join data, and create datasets for model training with automatic data cleansing and model monitoring. Canvas can automatically engage ML models for your use case, which still allows you to configure your model based on your use case.

Your developers can share the model and dataset with other teams (such as a data science team in future or a business intelligence team) to review and provide feedback. For customers who want

to start AI/ML without any expertise, Canvas is a good option to consider. In this case, you don't have SageMaker, but rather a fully-managed service where you do not worry about provisioning, managing, administrating, or any related activity to start using AI/ML technologies.

Amazon Forecast

For demand forecasting, you can use Amazon Forecast. Amazon Forecast can fully enable businesses, such as utilities and manufacturing, with modern forecasting vision by providing high quality forecasting using state-of-the-art forecasting algorithms. Amazon Forecast comes with six algorithms; ARIMA, CNN-QR, DeepAR+, ETS, Non-Parametric Time Series (NPTS), and Prophet.

Amazon Forecast puts the power of Amazon's extensive forecasting experience into the hands of all developers, without requiring ML expertise. Amazon Forecast comes with an AutoML option (refer to Training Predictors in the Amazon Forecast Developer Guide), including common patterns such as holidays, weather, and many more features perfecting a forecast. It is a fully managed service that delivers highly accurate forecasts, up to 50% more accurate than traditional methods.

As shown in the following figure, you can input your historical demand data in Amazon Forecast (only target data or some optional related data). The service then automatically sets up a data pipeline, ingests the input data, and trains a model (out of many algorithms, it can automatically choose the best performing model). Forecast then generates forecasts. It also identifies features that apply the most to the algorithm, and automatically tunes hyperparameters. Forecast then hosts your models so you can easily query them when needed. In the background, Forecast automatically cleans up resources you no longer use.

Time series forecasting with Amazon Forecast

With all of this work done behind the scene, you can save by not building your own ML expert team or resources to maintain your own in-house models.

In summary, Amazon Forecast is designed with these three main benefits in minds:

- **Accuracy** — Amazon Forecast uses deep neural networks and traditional statistical methods for forecasting. It can learn from historical data automatically, and pick the best algorithms to train a model designed for the data. When there are many related time series forecasts models (such as historical sales data of different items or historical electric load of different circuits) generated using the Amazon Forecast deep learning algorithms provides more accurate and efficient forecasts.
- **End-to-end management** — With Amazon Forecast, the entire forecasting workflow, from data upload to data processing, model training, dataset updates, and forecasting, can be automated. Then business planning tools or systems can directly consume Amazon forecasts as an API.
- **Usability** — With Amazon Forecast, you can look up and visualize forecasts for any time series at different granularities. Developers with no ML expertise can use the APIs, AWS Command Line Interface (AWS CLI), or the AWS Management Console to import training data into one or more Amazon Forecast datasets, train models, and deploy the models to generate forecasts.

Following are two case studies for customers who used Amazon Forecast for their demand forecasting.

Case study 1— Foxconn

Foxconn built an end-to-end demand forecasting solution in two months with Amazon Forecast. Foxconn manufactures some of the most widely used electronics worldwide such as laptops, phones. Customers are large electronics brands like Dell, HP, Lenovo, Apple.

Assembling these products is a highly manual process. Foxconn's primary use case was to manage staffing by better estimating demand and production needs. Overstaffing means unused worker hours, while understaffing means overtime pay. Overtime pay is less costly.

Two main questions needed to answer were:

1. How many workers to call into the factory to meet short-term production needs for1 week ahead?
2. How many workers need to hire in the future in order to meet long-term production needs for 13 weeks ahead?

Foxconn had just over a dozen SKUs with 3+ years of daily historical demand data. Previous forecasting approach was to use forecast provided by Foxconn's customers. With COVID-19 these forecasts became less reliable and labor costs increased as a result.

Foxconn decided to use Amazon Forecast for their demand forecasting. The whole process (importing data, model training and evaluation) took 6 weeks from start to finish. Initial solution provided 8% forecast accuracy improvement and an estimated $553K in annual savings.

Case study 2 — More Retail Ltd. (MRL)

More Retail Ltd. (MRL) is one of India's top four grocery retailers, with a revenue in the order of several billion dollars. It has a store network of 22 hypermarkets and 624 supermarkets across India, supported by a supply chain of 13 distribution centers, 7 fruits and vegetables collection centers, and 6 staples processing centers.

Forecasting demand for the fresh produce category is challenging because fresh products have a short shelf life. With over-forecasting, stores end up selling stale or over-ripe products, or throw away most of their inventory (termed as shrinkage). If under-forecasted, products may be out of stock, which affects customer experience.

Customers may abandon their cart if they can't find key items in their shopping list, because they don't want to wait in checkout lines for just a handful of products. To add to this complexity, MRL has many SKUs across its over 600 supermarkets, leading to more than 6,000 store-SKU combinations.

With such a large network, it's critical for MRL to deliver the right product quality at the right economic value, while meeting customer demand and keeping operational costs to a minimum. MRL collaborated with Ganit as its AI analytics partner to forecast demand with greater accuracy and build an automated ordering system to overcome the bottlenecks and deficiencies of manual judgment by store managers.

MRL used Amazon Forecast to increase their forecasting accuracy from 24% to 76%, leading to a reduction in wastage by up to 30% in the fresh produce category, improving in-stock rates from 80% to 90%, and increasing gross profit by 25%.

Building competitive demand forecasting models

The spectrum of model complexity changes from simple auto-regression extrapolation (such as ARIMA) to deep learning (such as DeepAR) models. In today's complex and competitive business

environment, most customers go with state-of-the-art deep learning models, as presented in the previous chapter.

Even if you decide to start with a deep learning model, the heavy lifting of training your selected model and optimizing model hyperparameters remain a challenge. The Amazon SageMaker training technologies help you to achieve your goals.

SageMaker hyperparameter tuning can also perform the optimization of finding the best hyperparameters based on your objective. During the process, SageMaker Debugger can debug, monitor, and profile training jobs in near real-time, detect conditions, optimize resource utilization by reducing bottlenecks, improve training time, and reduce costs of machine learning models.

You can still use traditional time series analyses such as auto-regression models on SageMaker. Due to the simplicity of those models, you can use traditional methods to code the model and run analysis on SageMaker notebooks. You can also create your own models from scratch on SageMaker and keep them as container images to perform model/instance lifecycles as if they are SageMaker models, so you can use all of the applicable SageMaker features.

You can develop your own advanced demand forecasting ML solutions. For example, one of the advanced applications is to employ hierarchical time series forecasting using Amazon SageMaker. This approach considers the hierarchical relationship between the data groups, of which overall combined effects form the demand (for example, a store is in a city, a city is in a state, and so on). Once your model is developed by your data scientists or engineers, you can package models in a container and start using the container image as a reference.

Here you can use model versioning, using SageMaker Model Registry so your engineers work on these *reusable* assets (models in containers) just like they can with built-in SageMaker models. Refer to Building your own algorithm container.

The other option is to use Amazon Forecast and SageMaker Canvas to completely overhaul the optimization, giving the heavy lifting to AWS. Amazon Forecast has many time series models, including autoregressive and advanced models, so you can start performing demand forecasting through the Amazon Forecast APIs. SageMaker Canvas gives you a short-no-code distance to SageMaker capabilities.

Model lifecycle and ML operations (MLOps) for robust demand forecasting

Any forecasting model will *drift* over time, creating inaccurate predictions. Along with drift, new models and technologies emerge, requiring a model lifecycle. Regular revisits to business goals, model build, and deployment need to be a part of your organization's operation to maintain a robust demand forecasting capability.

A robust ML lifecycle starts with a business goal (such as increased revenue with better predictions in each quarter) that drives the ML problem framing (such as DL models running weekly). This, continued with data processing, including data acquisition, cleaning, visualizations, discovery, and feature engineering framing, are essential to maintaining an ML lifecycle.

Generally, time series data to be used for training needs to be cleaned, transformed and enhanced with feature engineering methods. If this process requires ML focused transformations (such as feature engineering) or will be performed on a single interface through SageMaker, you can use SageMaker Data Wrangler.

If the transformation is relatively straightforward and needs to be performed in a mass-scale, serverless manner and in an Extract, Transform, and Load (ETL) pipeline, you may prefer using AWS Glue DataBrew. These tools come with embedded user interfaces (UIs) to create a better user experience for data scientists and business intelligence teams.

Once your time series data is cleaned, transformed and enhanced, the process proceeds with training, model tuning, and optimization, which may include some experimentation in the early stages of a new model proposition. Based on the model development, the training data needs to be revisited through feature engineering, more related data acquisition (internal/external), or focus.

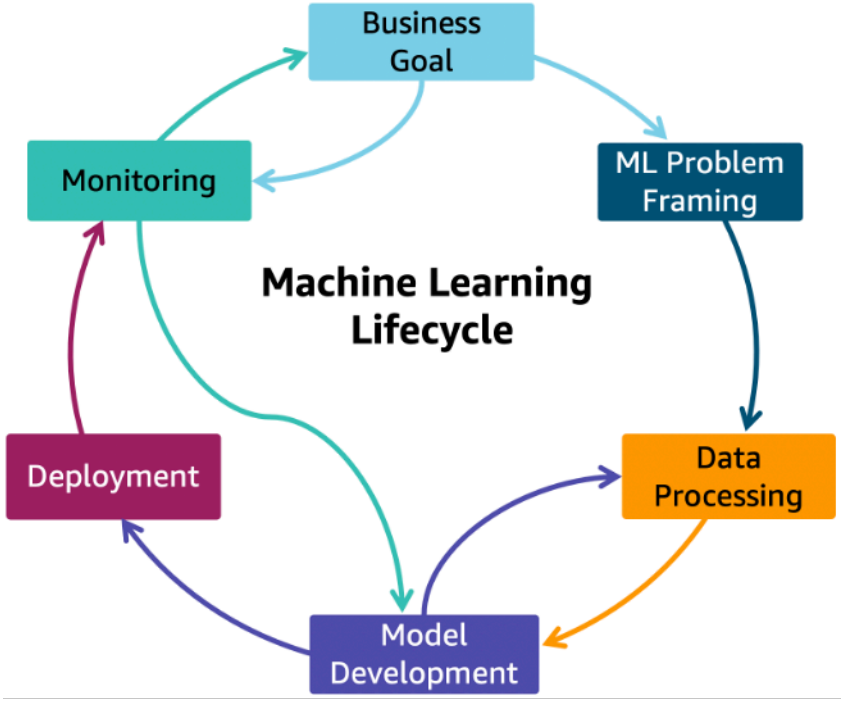
The training model needs to pass the test conditions. Typically, these are metrics for:

- Root Mean Square Error (RMSE)
- Weighted Quantile Loss (wQL)
- Mean Absolute Percentage Error (MAPE)
- Mean Absolute Scaled Error (MASE)
- Weighted Absolute Percentage Error (WAPE)

Next, deploy the ML models through SageMaker or through Amazon Forecast API when the predictor inferencing starts.

Next, the model should be monitored and continuously reviewed based on business goals. If needed, the whole cycle should be iterated to keep your demand forecasting capabilities competitive.

Sometimes, the model needs investigations. For example, if the new event creates a new type of behavior in the predicting system or a horizon of the data has changed, reducing accuracy. Therefore, the overall loop of ML operations should include other teams, not limited to data science teams.



The machine learning lifecycle

If you are using Amazon SageMaker, this generally means (except in some cases with SageMaker Canvas) there are data scientists collaborating on the project. Therefore, in addition to the loop in the preceding figure, the engineers need to efficiently work on the same models.

There may be some gates that are part of DevOps in your organization requiring a CI/CD pipeline. This is also called *ML infrastructure through code* using CI/CD. The idea is to make the MLOps visible, granular, and traceable, to make engineering/data science/development (DevOps) collaborations proceed faster and with higher quality (fewer defects).

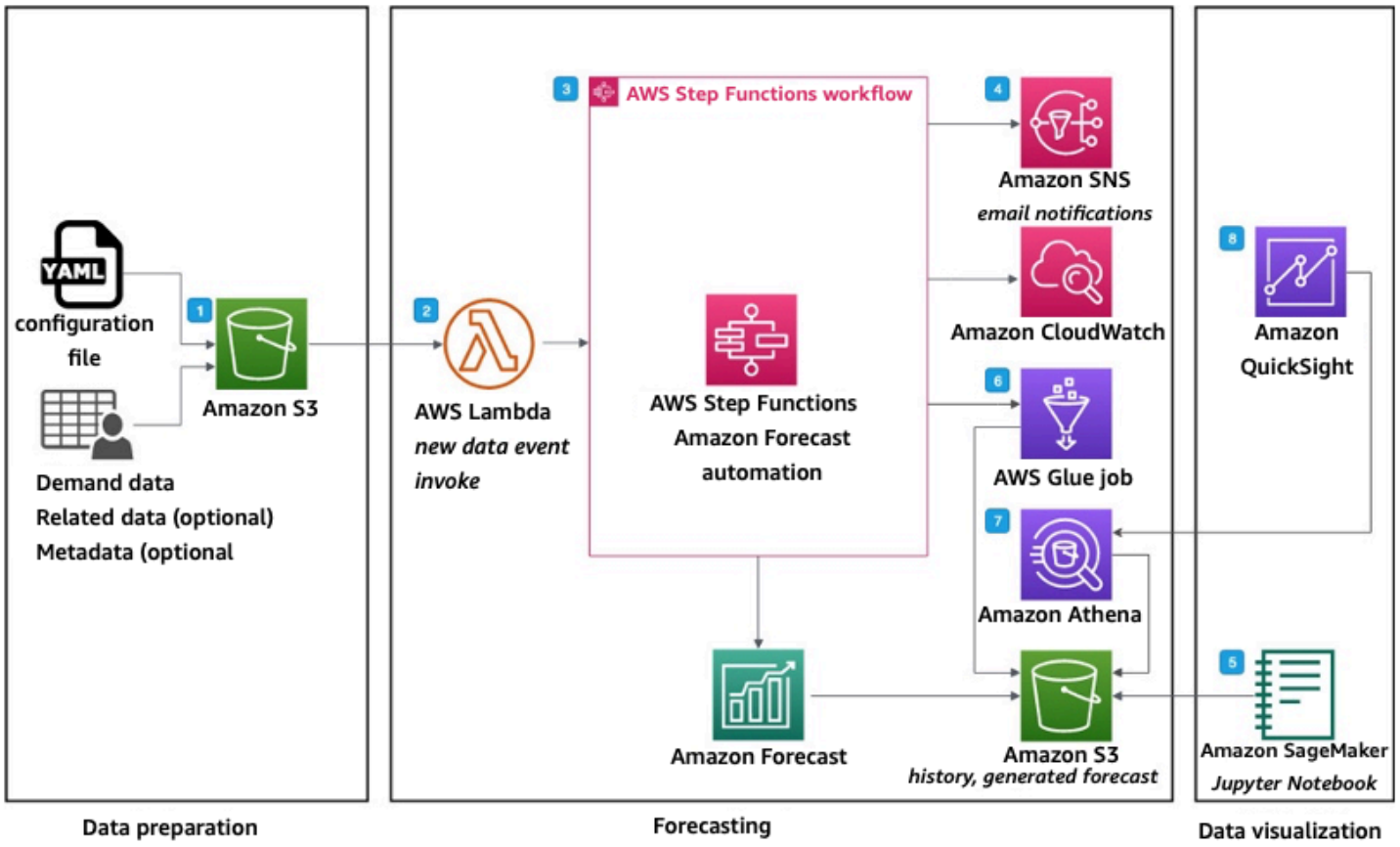
This means data scientists, or any teams running the operations, work together with minimal manual processes to increase speed and minimize human errors. You also need visibility and traceability to follow errors back, and put an approval mechanism for customers to create *gates* to deployments or share knowledge and commonality with other teams and projects.

MLOps is not a process, but an agile and flexible combination of human and machine code interactions which securely, reliably, and quickly get value from AI/ML capabilities. For SageMaker, you can use SageMaker Pipelines. We recommend reviewing the Amazon SageMaker for MLOps website for examples, descriptions, videos, and more details.

MLOps is also possible with Amazon Forecast, as seen in the following architecture. The critical component in this solution is AWS Step Functions, which allows you to build and tie process, including AWS services, deployment, and your workforce, in a serverless setting.

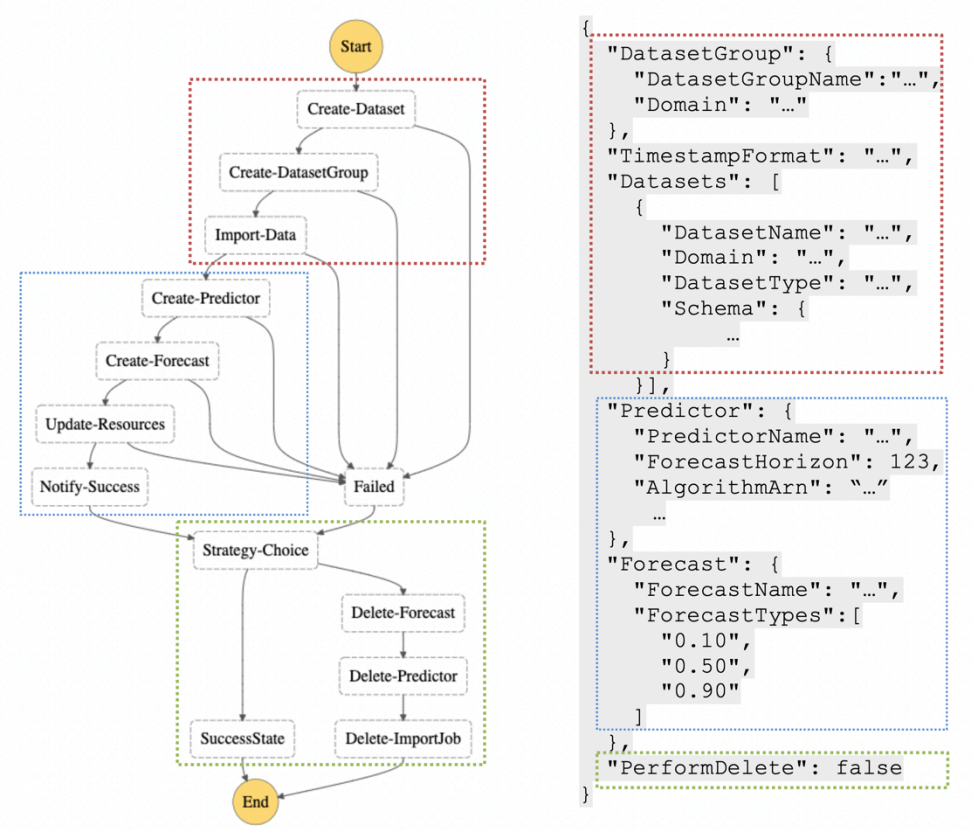
Complementary AWS services include Amazon CloudWatch and Amazon Simple Notification Service (Amazon SNS), which are used for monitoring processes and creating notifications.

AWS Glue automatically moves the data through an ETL pipeline to a S3 bucket, to be queried by and fed to Amazon QuickSight by Amazon Athena for the visualization of predictions and other data. You can deploy the following architecture through AWS CloudFormation templates. Refer to Improving Forecast Accuracy with Machine Learning and the Building AI-powered forecasting automation with Amazon Forecast by applying MLOps blog post for more information.



ML operations with AWS Step Functions

The following figure shows an example AWS Step Function definition that goes through MLOps steps with Amazon Forecast. To better understand this behavior, review Visualizing AWS Step Functions workflows from the AWS Batch console.



AWS Step Function steps example for an ML forecasting workflow

AWS has developed the Well-Architected Machine Learning Lens to help you review your operations and deployment, to determine whether or not you follow the best practices proposed by AWS. This approach utilizes security, operational efficiency, reliability, cost effectiveness, and performance. To keep your AI/ML operations robust, we highly recommend having these reviews internally and/or with your Solutions Architects or AWS Partners, regularly. Following Well-Architected best practices ensures that the MLOps process will reach its full potential for your organization.

Incorporating and interpreting AI/ML-based demand insights into the decision cycles

You can't fully utilize the value of AI/ML without democratizing your insights. Setting business goals is an important step to success. Business goal should start with identification of a clear business objective. We recommend a measurable objective, because AI/ML will eventually utilize metrics for your organization. Continuously measure business value against specific business

objectives and success criteria. Involve all stakeholders from the beginning to establish realistic but concrete value-delivering targets.

Once you determine your criteria for success, proceed with evaluating your organization's ability to move to the target. This includes the achievability of the objective, skill set, and tool set, as well as a clearly defined timeline for the objectives, which are regularly tracked and evaluated.

The business objective metric can easily be integrated into the Business Intelligence tool you use. In the previous chapter's example for Amazon Forecast MLOps, you can build a dashboard for your business metrics in Amazon QuickSight. With Amazon QuickSight Q, your teams can ask questions about the results using natural communication, or use insights to your metrics.

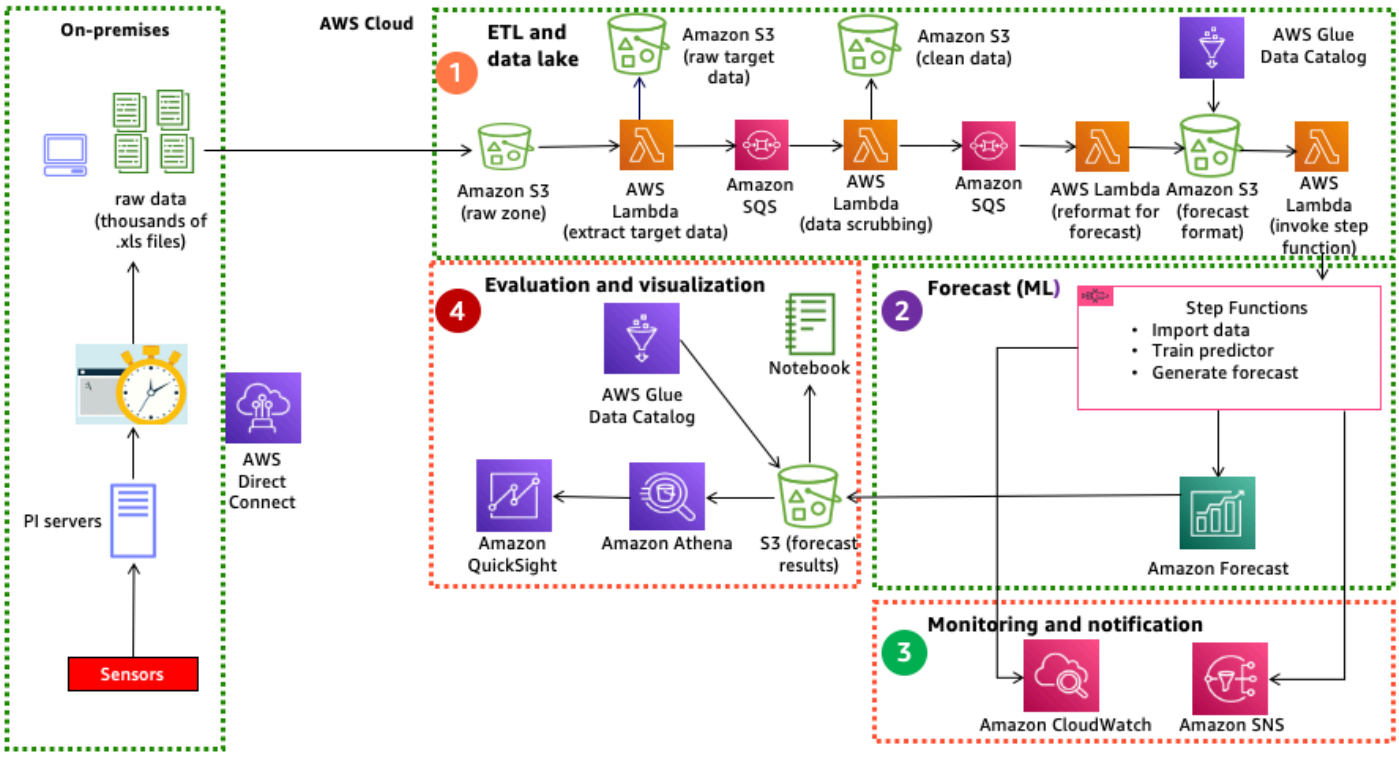
If you prefer to use your own BI tool, you can still calculate custom metrics or use other types of queries using Amazon Athena views. The same process can be used with the SageMaker approach. You can use QuickSight with SageMaker through the relevant S3 bucket. Refer to the Visualizing Amazon SageMaker machine learning predictions with Amazon QuickSight blog post.

Example architectures

The following section shows two practical examples of demand forecasting for energy companies and GCP.

Energy forecasting example

The following diagram is an architecture for short-term electric demand forecasting that can be used for other demand forecasting use cases, as the concept is similar to other use cases. The proposed solution is using advanced Amazon Forecast features to solve forecasting problems. It is fully automated and event driven, with reduced manual processes. Also, it can scale as the forecasting needs increase.



Energy forecating architecture

The solution includes these broad steps:

- **Module 1 -** Ingest and transform data from the on-premises system
- **Module 2** - Forecast: Calling series of APIs to Amazon Forecast
- **Module 3** - Monitoring and notification
- **Module 4** - Evaluation and visualization

The left box on the preceding diagram (labeled *On-premises)* is an example of field data ingestion that a typical utility has established on-premises. In this case, data ingestion from field sensors (a feeder head meter) is already in place through the PI System. Therefore, the starting point is raw circuit demand data extracted in Excel format, and stored on-premises. For Amazon Forecast to use this data, we first need to upload it to Amazon S3. This can be performed periodically by a scheduled script.

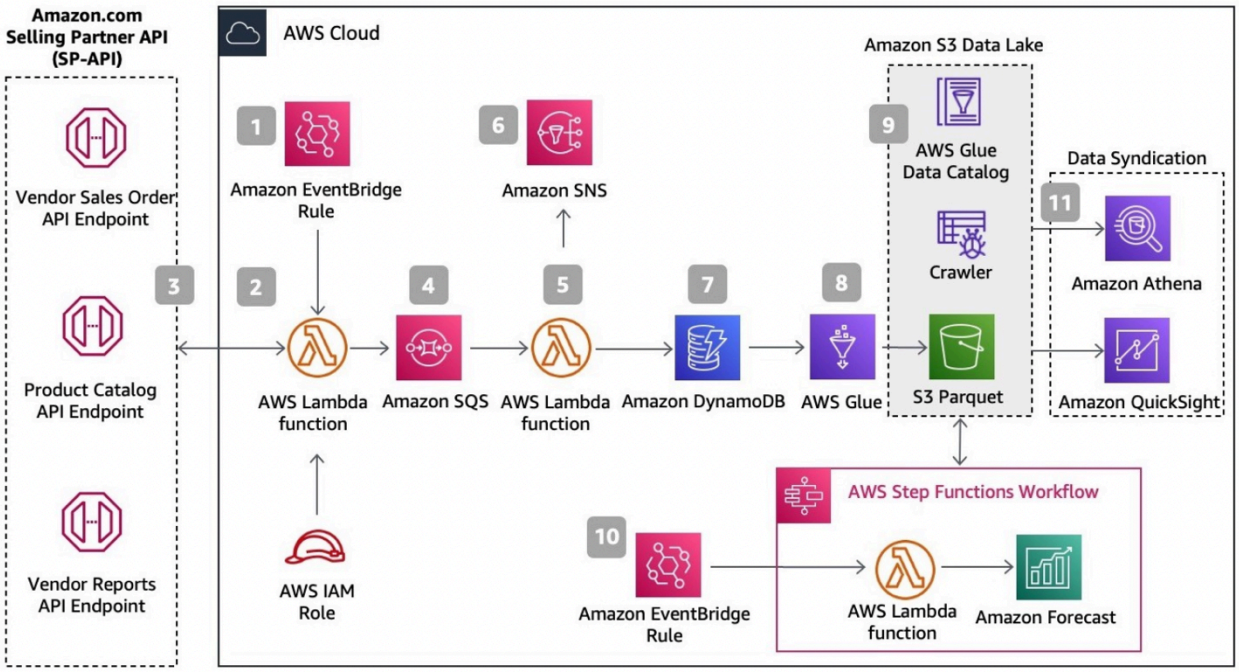
Next (in Module 1), as the raw data is available in Amazon S3 in utility custom-built formats, we need to massage the data. This involves extracting target data (consumption power in kilowatts),

cleaning the data, and reformatting it. To do this, we use the services outlined in the box (labeled *ETL and data lake*) as follows:

- Amazon S3 is used to store raw and formatted data. This highly durable and highly available object storage is where the short-term electric load forecast (ST-ELF) input datasets will be consumed by Amazon Forecast.
- AWS Lambda is used to massage and transform the raw data. This serverless compute service runs your code in response to events (in this case, new raw file uploads) and automatically manages the computing resources required by the code.
- The Amazon Simple Queue Service (Amazon SQS), a fully managed message queuing service, decouples these transformation AWS Lambda functions. The queue acts as a buffer and can help smooth out traffic for the systems when consuming a multitude of events from many field sensors.
- AWS Glue is used for data cataloging, this service can run a crawler to update tables in your AWS AWS Glue Data Catalog, and as configured here, runs on a daily schedule.
- With data wrangling complete, the ST-ELF model is now ready for training. To train the model, we use Amazon Forecast (in Module 2). Using Amazon Forecast requires the import of training data, creation of a predictor (the ST-ELF model in this case), and creation of a forecast using the model, and finally exporting of the forecast and model accuracy metrics to Amazon S3.
- To streamline the process of ingesting, modeling and forecasting multiple ST-ELF models, the Improving Forecast Accuracy with Machine Learning solution is leveraged. This best practice AWS solution streamlines the process of ingesting, modeling, and forecasting using Amazon Forecast by providing AWS CloudFormation templates and a workflow around the Amazon Forecast service.
- Because forecasting might take some time to complete, the solution uses the Amazon Simple Notification Service (Amazon SNS) to send an email when the forecast is ready. Also, all AWS Lambda function logs are captured in Amazon CloudWatch (Module 3).
- Once the forecast is ready, the solution ensures your forecast data is ready and stored in Amazon S3. This fulfills a 14-day-ahead forecasting need. The solution also provides a mechanism to query the forecast input and output data with SQL using Amazon Athena (Module 4). Additionally, the solution can automatically create a visualization dashboard in the AWS Business Intelligence (BI) service Amazon QuickSight, which gives you the ability to visualize the data interactively in a dashboard.

Consumer Packaged Goods (CPG) forecasting example

This solution works with Amazon Vendor Central (sign-in required), but it can be used for other in-house or commercial endpoints. This solution builds a serverless architecture which is an event-based or scheduled pattern, that consistently and constantly pulls data from the CPG info endpoint into a persistent datastore. The data is then cataloged and analyzed through Amazon Athena, AWS Glue crawlers, and Amazon QuickSight. Here, Amazon purchase order (PO) data is ingested into Amazon Forecast to generate predictive analysis to help CPG companies improve on-time, in-full (OTIF) performance. To learn more about this solution, refer to CPG Companies: Improve Demand Forecasting to Boost Sales on Amazon.



Architecture to automate data collection for selling partners

Conclusion

This whitepaper provided best practices and common architectural patterns, introducing managed AWS technologies and recommendations about demand forecasting. AWS has the expertise, breadth of services, and partner landscape to provide all industries, including industrials, manufacturing, CPG, retail, and utilities with the right tools to create demand forecasting workloads in scale. This paper helps you on find the best solutions, next steps, and best practices specifically for demand forecasting. It also shows ways to build automations and data pipelines using AWS solutions.

Contributors

Contributors to this document include:

- Dr. Pedram Jahangiri, Senior Solutions Architect, Amazon Web Services
- Dr. Burak Gozluklu, Principal ML Specialist Solutions Architect, Amazon Web Services

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Whitepaper published.	September 22, 2022

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.