AWS Whitepaper

Architectural Patterns to Build End-to-End Data Driven Applications on AWS

Architectural Patterns to Build End-to-End Data Driven Applications on AWS: AWS Whitepaper

Table of Contents

Architectural Patterns to Build End-to-End Data Driven Applications on AWS

Publication date: **August 3, 2022** (*Document revisions*)

Abstract

According to Forbes, 85% of businesses want to be data driven, but only 37% have been successful. Most of the organizations that try to use data to modernize and innovate struggle with finding the proven architectural patterns that customers have implemented to build data-driven applications. Successful customers use multiple Amazon Web Services (AWS) services between event-driven Internet of Things (IoT) to collect and manage billions of devices and purpose-built databases. This allows them to save costs, grow and innovate faster, and use data analytics to get the fastest insights on all their data and artificial intelligence/machine learning (AI/ML) to help them innovate for the future.

In this whitepaper, we present some commonly used data-driven applications and proven architectural patterns based on successful customer implementation. This enables customers who are looking to build those data driven applications to accelerate time to solution.

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 Data Analytics Lens, we describe a collection of customer-proven best practices for designing well-architected analytics workloads.

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

Introduction

There are more than 200 services provided by AWS. Customers can use this diverse set of choices to select the correct tool for the right job, but sometimes it can be confusing to map to different use cases. There are some services with overlapping capabilities, and customers often struggle to find a proven pattern when they're developing their data-driven applications. Some customers take the approach of doing multiple proof of concepts (POCs), which is time consuming, and might still fail to instill confidence in whether a set of services is a proven pattern based on other AWS customer implementations.

This whitepaper brings together the most commonly used data-driven applications and architectural patterns that other AWS customers have proven to be successful in their implementations. These architecture reference patterns provide guidance to quickly select a proven architectural pattern and further modify it to meet your application needs. In some cases, these architecture patterns can be used without modification, thereby minimizing the need to POC extensively, and accelerate time to solution.

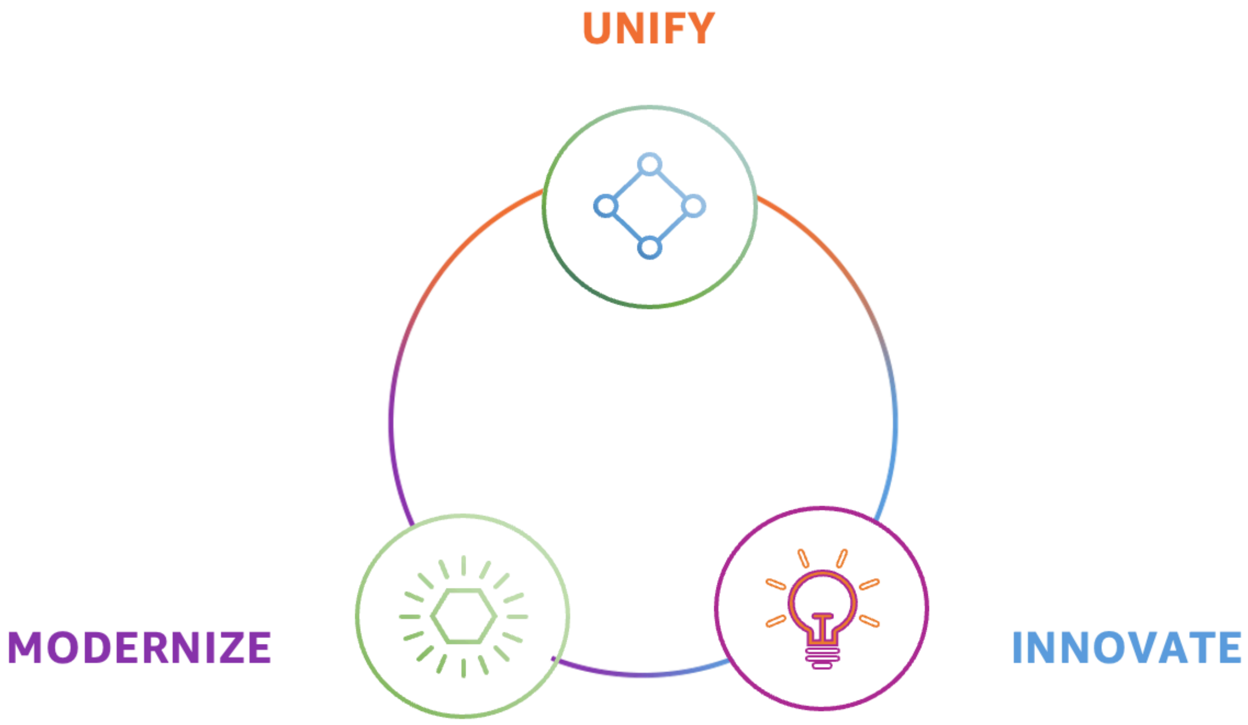
The architecture reference patterns covered in this whitepaper also provide thought leadership for you to create a future state strategy to modernize your data-driven applications. It helps you look through the lens of how various AWS data services, such as AWS IOT for event-driven IOT data collection, manage billions of devices and purpose-built databases to save costs, modernize databases for the cloud, and innovate faster. Use analytics to get fastest insights on all your data, from big data processing, data warehousing to visualization, and AI/ML, to build and operationalize ML and AI applications to innovate for the future.

AWS for data

There are three different stages to building a modern data strategy. These stages are not sequential, and you can start at any stage, depending on your data journey.

- **Modernize** — You can modernize your data infrastructure with the most scalable, trusted, and secure cloud provider.
- **Unify** — You can unify your data by putting it to work in the best data lakes and purpose-built data stores.
- **Innovate** — You can create new experiences and reimagine old processes with AI and ML.

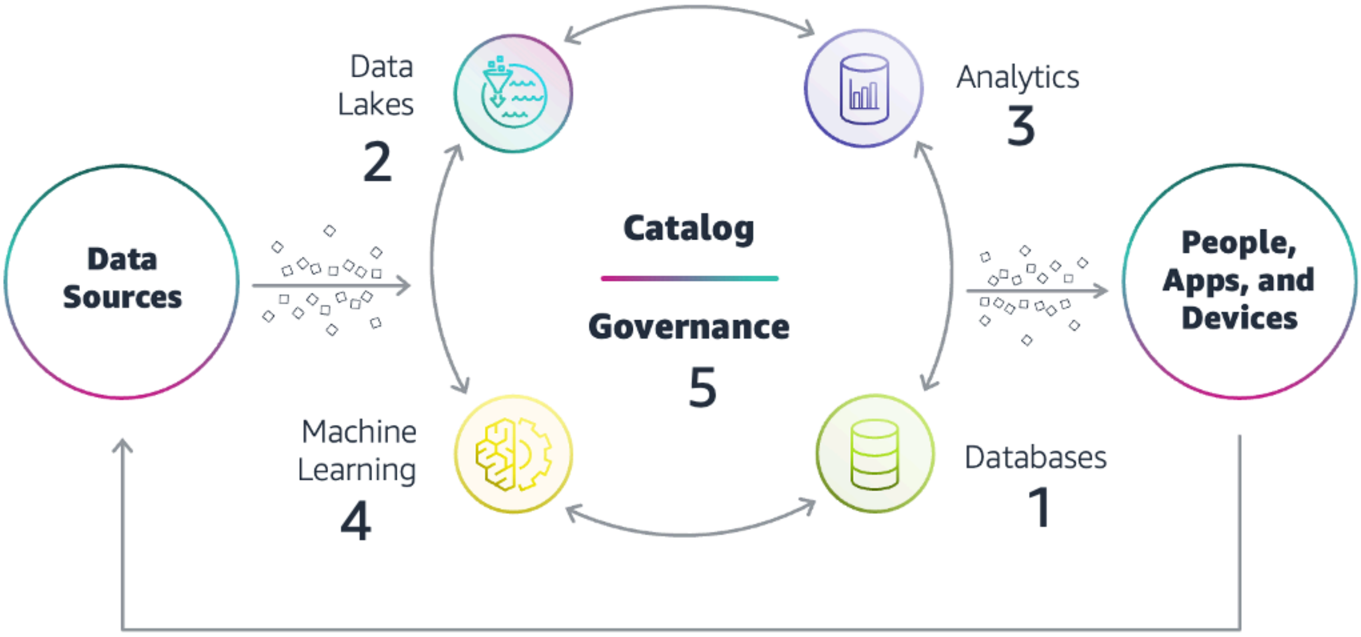
These three stages in turn lead to the development of a modern data strategy on AWS, as illustrated in the following figure. This strategy gives you the best of both data lakes and purpose-built stores. It enables you to store any amount of data at a low cost, and in open-standards data formats. This helps you avoid the risks of getting locked into a proprietary format. It helps you break down data silos and empower your teams to run analytics or ML at scale, using your preferred tools and techniques.



Modern data strategy on AWS

Modern data architecture

AWS modern data architecture connects your data lake, your data warehouse, and all other purpose-built stores into a coherent whole. The following figure depicts a modern data architecture on AWS.



Modern data architecture on AWS

Key:

1. Store data in a purpose-built database that can support a modern application and its different features. It no longer needs to be a single type of relational database, but could be a NoSQL database, cache store, or anything else which works for the application. The focus is on application empowerment, not analytics.
2. Use the data lake, which uses a storage service such as Amazon Simple Storage Service (Amazon S3), to store data from all those purpose-built databases and native open-file formats. Open formats give more power to organizations to use the data however they need to.
3. Once the data lake is hydrated with data, you can build analytics of any kind easily, and use any technology customers want to use for their use cases. This can range from traditional data

warehousing and batch reporting to more near real-time alerting and reporting. This can even be ad-hoc querying of data, or more advanced ML-based analytics use cases. Data is now stored in a layer that's more open and provides more freedom of analytics choices.

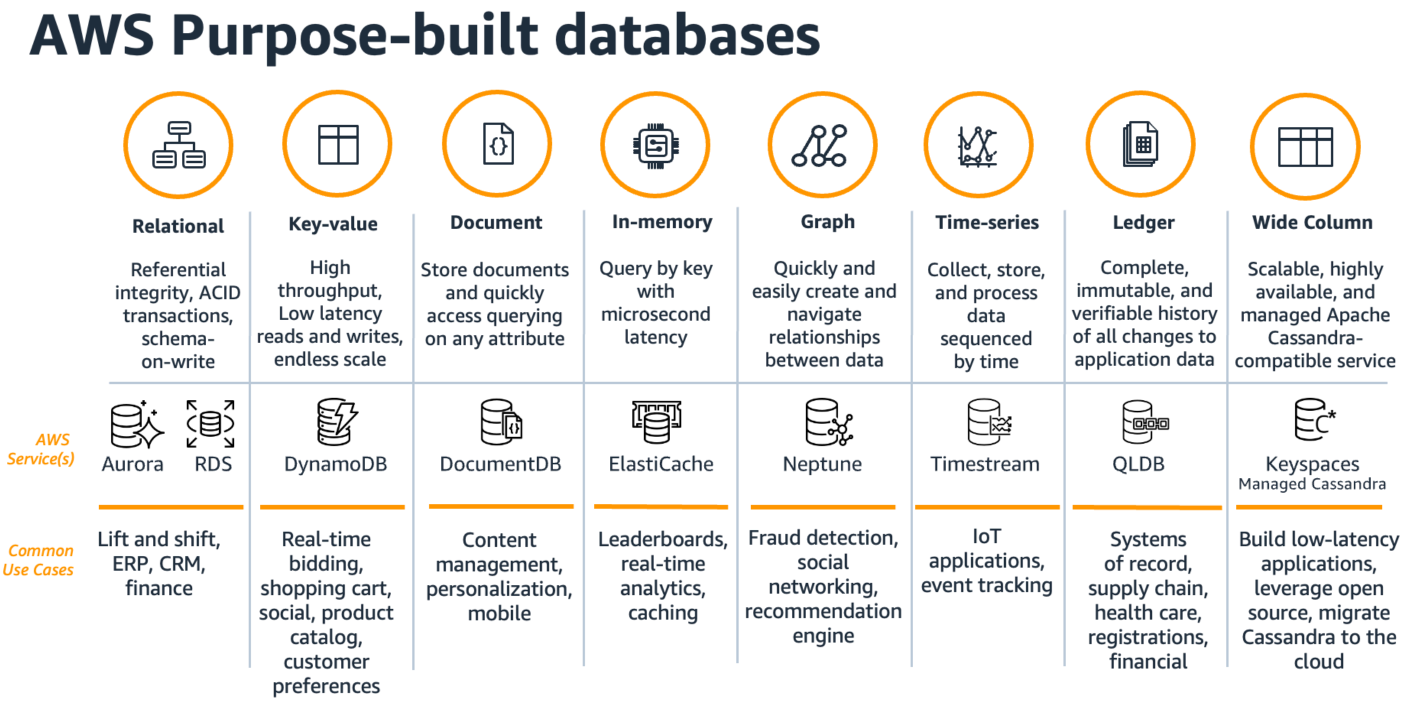
4. ML and AI are also critical for a modern data strategy, to help you predict what will happen in the future, and to build intelligence into your systems and applications. Customers have varied needs when it comes to AI/ML, and require a broad set of AI/ML services. This is true for expert ML practitioners, for everyday developers, and for customers who don't want to build and train models.
5. Finally, data governance is a critical element to combining and sharing data from different sources so you can put data in the hands of people at all levels of your organization. There are data residency compliance needs which come with globalization. There is need for granular control over who has permission to access data, and standard security needs such as encryption, auditability, monitoring, alerting, and so on.

Purpose-built databases

Organizations are breaking free from legacy databases. They are moving to fully managed databases and analytics, modernizing data warehouse databases, and building modern applications with purpose-built databases to get more out of their data. Organizations want to move away from old-guard databases because they are expensive, proprietary, lock you in, offer punitive licensing terms, and come with frequent audits from their vendors. Organizations that are moving to open-source databases are looking for the performance of commercial-grade databases with the pricing, freedom, and flexibility of open-source databases.

AWS provides purpose-built database services to overcome these challenges with on-premises database solutions. The most straightforward and simple solution for many organizations that are struggling to maintain their own relational databases at scale is a move to a managed database service, such as Amazon Relational Database Service (Amazon RDS). In most cases, these customers can migrate workloads and applications to a managed service without needing to re-architect their applications, and their teams can continue to use the same database skill sets. Organizations can lift and shift their self-managed databases such as Oracle, SQL Server, MySQL, PostgreSQL, and MariaDB to Amazon RDS. For organizations that are looking for better performance and availability, they can move their relational databases to MySQL and PostgreSQL to Amazon Aurora and get three to five times better throughput.

Many organizations also use non-relational databases such as MongoDB and Redis as document and in-memory databases for use cases such as content management, personalization, mobile apps, catalogs, and near real-time use cases such as caching, gaming leaderboards, and session stores. The most straightforward and simple solution for many organizations who are struggling to maintain their own non-relational databases at scale is a move to a managed database service (for example, moving self-managed MongoDB databases to Amazon DocumentDB or moving self-managed in-memory databases such as Redis and ElastiCache to Amazon ElastiCache). In most cases, these organizations can migrate workloads and applications to a managed service without needing to re-architect their applications, and their teams can continue to use the same database skill sets.



AWS Purpose-built databases

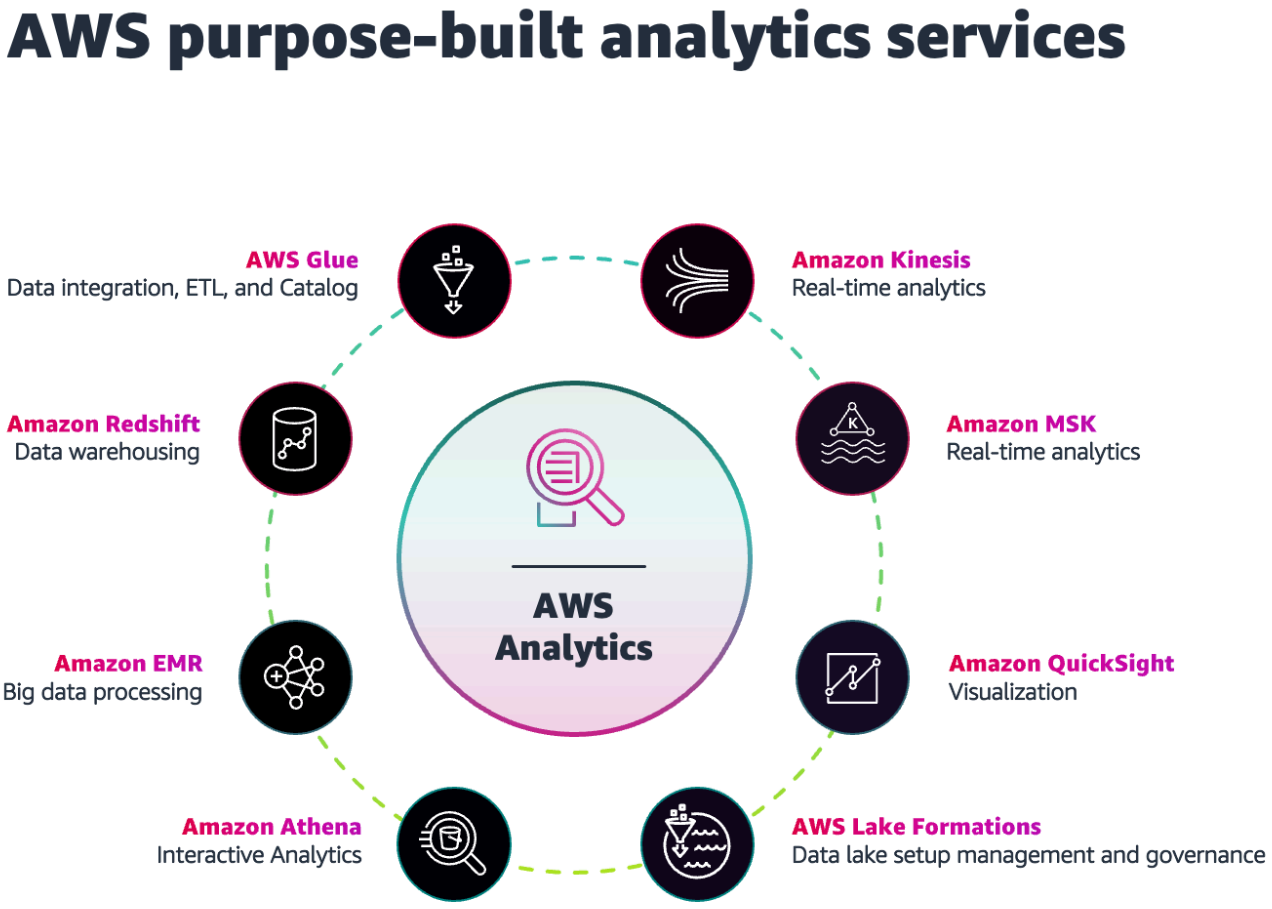
	Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger	Wide Column
	Referential integrity, ACID transactions, schema-on-write	High throughput, Low latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data	Scalable, highly available, and managed Apache Cassandra-compatible service
AWS Service(s)	Aurora RDS	DynamoDB	DocumentDB	ElastiCache	Neptune	Timestream	QLDB	Keyspaces Managed Cassandra
Common Use Cases	Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial	Build low-latency applications, leverage open source, migrate Cassandra to the cloud

Purpose-built database services on AWS

Purpose-built data analytics services

Purpose-built data analytics services enable you to use the right tool for the right job, and provides the agility to iteratively and incrementally build out the modern data architecture. You gain the flexibility to evolve the data lake house to meet current and future needs as you add new data sources, discover new use cases and their requirements, and develop newer analytics methods.

AWS provides a portfolio of purpose-built data analytics services, which include AWS Glue, Amazon EMR, Amazon Athena, Amazon Kinesis, Amazon Managed Streaming for Apache Kafka (Amazon MSK), Amazon OpenSearch Service, Amazon Redshift, and Amazon QuickSight for various unique analytics use cases. These are managed services, so organizations don’t need to worry about administration tasks such as software provisioning, configuration, backups, patching, and recovery. Instead, organizations can focus on getting insights from their data and creating a business value, not managing the infrastructure. These purpose-built data analytics services are optimized for specific use cases, because one size solution doesn’t fit for all use cases and leads to compromises in analytics. The purpose-built-analytics capabilities are illustrated in the following figure.



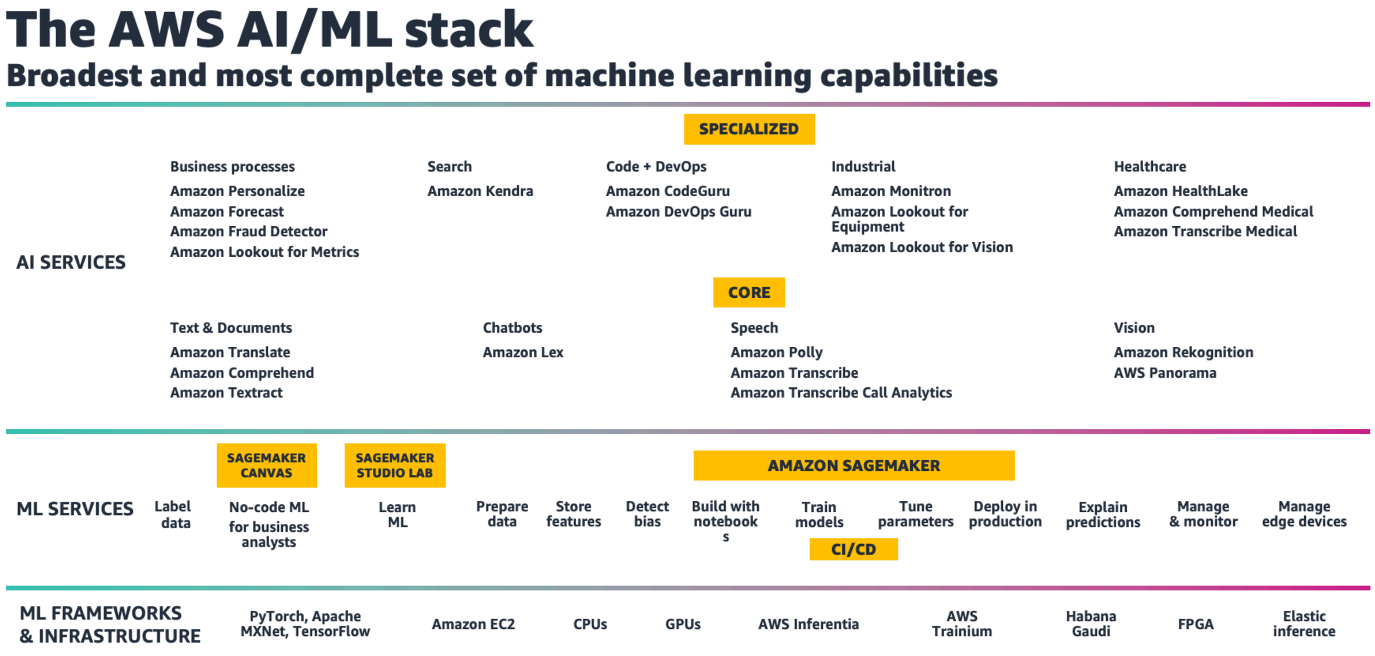
Purpose-built data analytics services on AWS

ML and AI services

In the past, ML technology was limited to a few major tech companies and hardcore academic researchers, but things began to change when cloud computing entered into the mainstream. Compute power and data became more available, and ML is now making an impact across almost every industry, including finance, retail, fashion, real estate, healthcare, and more. ML is moving from the peripheral to a core part of most every business and industry. In time, virtually every application will be infused with ML and AI.

AWS provides the broadest and deepest set ML and AI services, in three different levels, for builders of all levels of expertise in their ML journey:

- **For expert practitioners**, AWS supports all the major ML frameworks ,including TensorFlow, MXNet, PyTorch, Caffe 2, and so on. We offer the highest performance instances for ML training in the cloud with Amazon EC2 P4d instances, powered by the latest NVIDIA A100 Tensor Core GPUs, and coupled with first-in-the-cloud 400 Gbps instance networking. P4d instances are deployed in hyperscale clusters, called EC2 UltraClusters, offering supercomputer-class performance for the most complex ML training jobs. For inference, which represents 90% of the cost of ML, we provide the lowest cost for ML inference in the cloud with Amazon EC2 Inf1 instances, powered by AWS Inferentia chips.
- **For data scientists and ML developers**, AWS provides Amazon SageMaker. SageMaker was built from the ground up to simplify the process of ML with tools for every step of the ML development, including labeling, data preparation, feature engineering, statistical bias detection, auto-ML, training, tuning, hosting, explainability, monitoring, and workflows.
- **For developers and business users**, AWS provides pre-trained AI services that provide ready-made intelligence for applications and workflows, and end-to-end solutions that can solve business needs right out of the box using AutoML technology. These services address common use cases such as personalized recommendations, contact center intelligence, document processing, intelligent search, business metrics analysis, and more. AWS also provides industry-specific AI services for both industrial and healthcare industries.



The AWS AI/ML stack

Broadest and most complete set of machine learning capabilities

AI SERVICES

SPECIALIZED

Business processes
Amazon Personalize
Amazon Forecast
Amazon Fraud Detector
Amazon Lookout for Metrics

Search
Amazon Kendra

Code + DevOps
Amazon CodeGuru
Amazon DevOps Guru

Industrial
Amazon Monitron
Amazon Lookout for Equipment
Amazon Lookout for Vision

Healthcare
Amazon HealthLake
Amazon Comprehend Medical
Amazon Transcribe Medical

CORE

Text & Documents
Amazon Translate
Amazon Comprehend
Amazon Textract

Chatbots
Amazon Lex

Speech
Amazon Polly
Amazon Transcribe
Amazon Transcribe Call Analytics

Vision
Amazon Rekognition
AWS Panorama

ML SERVICES

SAGEMAKER CANVAS

SAGEMAKER STUDIO LAB

AMAZON SAGEMAKER

Label data

No-code ML for business analysts

Learn ML

Prepare data

Store features

Detect bias

Build with notebook s

Train models

Tune parameters

Deploy in production

CI/CD

Explain predictions

Manage & monitor

Manage edge devices

ML FRAMEWORKS & INFRASTRUCTURE

PyTorch, Apache MXNet, TensorFlow

Amazon EC2

CPUs

GPUs

AWS Inferentia

AWS Trainium

Habana Gaudi

FPGA

Elastic inference

AWS ML and AI services

Data driven architectural patterns

Many data-driven organizations seek the truth by treating data like an organizational asset, no longer the property of individual departments. They set up systems to collect, store, organize, and process valuable data and make it available in a secure way, to the people and applications that need it. They also use technologies such as ML to unlock new value from their data such as improving operational efficiency, optimizing processes, developing new products and revenue streams, and building better customer experiences.

We depict the five most commonly seen architecture patterns on AWS, that cover several use cases for various different industries and customer sizes:

- Customer 360 architecture
- Event-driven architecture with IOT data
- Personalized architecture recommendations
- Near real-time customer engagement
- Data anomaly and fraud detection

Build Customer 360 architecture on AWS

As customer expectations continue to rise, organizations face a gap between their need to use their data to meet customer expectations, and their current data management practices' ability to deliver. So how can companies close this gap? They must work toward taking back ownership of their data to harness the power of their customer and prospect information, to compete and win on customer experience.

Customer 360 describes the trend of building a complete and accurate picture of every customer's structured and unstructured data across an organization. Customer 360 is not a single solution, nor a single service. It is the aggregation of all customer data into a single unified location, that can be queried and used to generate insights for improving the overall customer experience.

Some of the common use cases and solutions that fit under these Customer 360 architecture patterns are as follows:

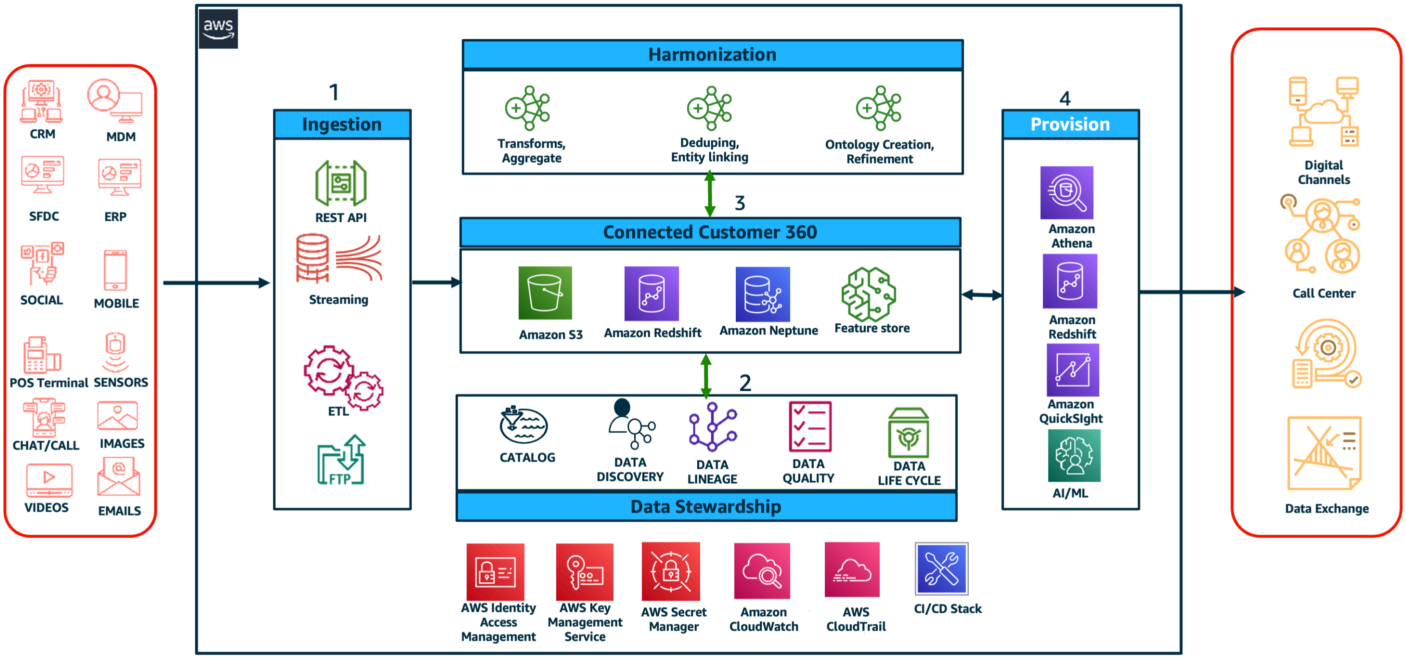
- **Marketing and demand generation**
  - Enrich customer profiles and identify the best prospects to drive customer acquisition.

  - Enable affinity marketing to achieve greater wallet share.
  - Personalize offers and website experiences based on historical customer data and micro-targeted segments to increase conversion rates.
  - Monitor campaign effectiveness and key performance indicators to improve the return on investment (ROI) of marketing efforts.
  - Predict consumer churn so you can take proactive actions to drive retention and repeat purchase.

- **Sales growth**
  - Gain insights to identify opportunities, predict customer intent, and foster stronger relationships.
  - Foster relationships with a complete view of a customer’s interactions to better understand the health of the relationship.
  - Increase customer lifetime value with data-driven, next best action suggestions, and intelligent cross-sell and upsell recommendations.
  - Deliver consistent cross-channel experiences. enabling an online-offline feedback loop.

- **Customer service**
  - Use holistic customer profiles to accelerate issue resolution and improve customer satisfaction.
  - Detect anomalies and strengthen trust.
  - Implement self-service tools and chatbots that allow customers to resolve issues themselves.
  - Deliver personalized loyalty offers to increase customer lifetime values and reduce churn.
  - Generate customer lifetime value and next best action recommendations.

The following diagram illustrates customer 360 architecture on AWS.



Customer 360 architecture on AWS

The steps that data follows through the architecture are as follows:

1. **Data ingestion —** Establishing customer 360 data by consolidating your disparate data sources in a schemeless ingestion approach, with the guiding principle of:
   - Getting the right data ingested as close to the source system as possible from past data.
   - Predicting future data and persisting them in object storage such as Amazon S3 or capturing them as streams in near real-time using:
   - SAP connectors.
   - AppFlow services that intuitively connect you to your Salesforce account.
   - Google analytics services and data movement service to connect to JDBC/ODBC sources to capture the data in incremental fashion with zero coding.
2. **Building a unified customer profile** — You need to extract and link elements from each customer record, creating a customer 360° dataset that will serve as a single source of truth for your customer data hub, with a data model to build your customers context on their journey. Using Amazon Neptune, you can create a near real-time, persistent, and precise view of the customer and their journey in a graph-based storage system. Use Neptune to store entities that provide context through an associated and predefined set of rules, allowing relationship traversals viewed in a connected fashion.

This 360° knowledge graph serves the potential information to uncover hidden customer segments, and deepens customer hierarchy, provides cluster analysis resulting in the creation of customer personas, and evaluates the size of each segment. The unique customer journey is preserved by writing reusable Gremlin/SPARQL libraries that are automated through a serverless data lake framework.

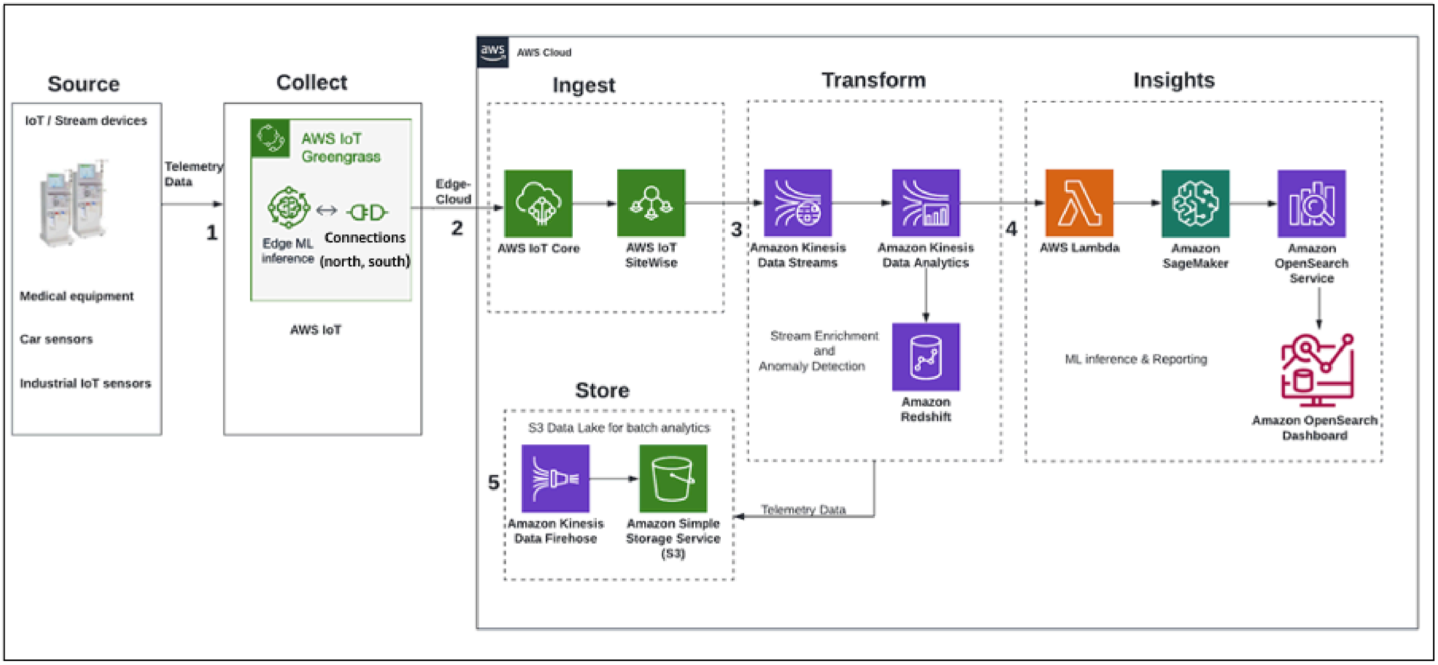
3. **Intelligence layer** — Offloading this journey into analytical store such as Amazon Redshift or S3 enables your data scientists and data analysts to refine the ontologies and shorten the graph path with access to raw information already preserved in S3 using AWS Glue DataBrew and Amazon Athena serverless.
4. **Activation layer** — This refinement to customer ontology represents connectedness between customer 360° data using AWS lake house architecture. It enriches customer profiles with recommendations, predictions using AI/ML to test the customer journey hypothesis, and creates the next best action APIs by sensing and responding to signals through the customer lifecycle using Amazon personalize and pinpoint. Actions offered via next-best APIs are now integrated and presented across distribution systems and channels for resilient and optimized personal experience.

Build event driven architectures with IOT sensor data

Organizations create value by making decisions from their data in near real-time. Some of the common use cases and solutions that fit under this event driven architecture pattern include:

- Industrial IOT use cases to monitor industrial equipment quality, to determine actions such as adjusting machine settings, using different sources of raw materials, or doing additional worker training that will improve the quality of the factory output.
- Medical device data collection for personalized patient health monitoring, adverse event prediction, and avoidance.
- Connected vehicle use cases such as voice interaction, navigation, location-based services, remote vehicle diagnostics and predictive maintenance, media streaming, and vehicle safety that includes computing within vehicles and in-the-cloud, near real-time predictive analytics.
- Sustainability and waste reduction solutions on AWS can provide access to dashboards, monitoring systems, data collection, and summarization tools that use ML algorithms that meet sustainability goals. Meeting sustainability goals is paramount to many use customers in travel and hospitality industries.

The following diagram illustrates event-driven IOT sensor data for near real-time predictive analytics.



Derive near real-time predictive analytics from IOT data

The steps that data follows through the architecture are as follows:

1. Data originates in IOT devices such as medical devices, car sensors, industrial IOT sensors, and so on. This telemetry data is collected close to the devices using AWS IoT Greengrass, which enables cloud capabilities to local devices.
2. Data is then ingested into the cloud using edge-to-cloud interface services such as AWS IOT Core, which is a managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices, or AWS IOT SiteWise, which is a managed service that lets you collect, model, analyze, and visualize data from industrial equipment at scale.
3. Stream data ingested into the cloud gets transformed in near-real time using Amazon Managed Service for Apache Flink, which offers an easy way to transform and analyze streaming data in near real-time with Apache Flink and Apache Beam frameworks. Stream data often needs to be enriched using lookup data which is hosted in a data warehouse. Amazon Redshift uses customer integration or stream aggregates (for example, one minute or five minutes) from Managed Service for Apache Flink. The Flink application gets written into Amazon Redshift for further business intelligence (BI) reporting downstream.

4. Amazon SageMaker is a fully ML learning service. Once the ML model is trained and deployed in SageMaker, inferences are invoked in micro-batch using AWS Lambda. Inferenced data is sent to Amazon OpenSearch Service to create personalized monitoring dashboards using Amazon OpenSearch Service Dashboards.
5. The data lake stores telemetry data for future batch analytics. The data is micro-batch streamed into the S3 data lake using Amazon Data Firehose, which is a fully managed service for delivering near real-time streaming data to destinations such as S3, Amazon Redshift, Amazon OpenSearch Service, Splunk, and any custom HTTP endpoints or owned by supported third-party service providers, including Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, and Sumo Logic.

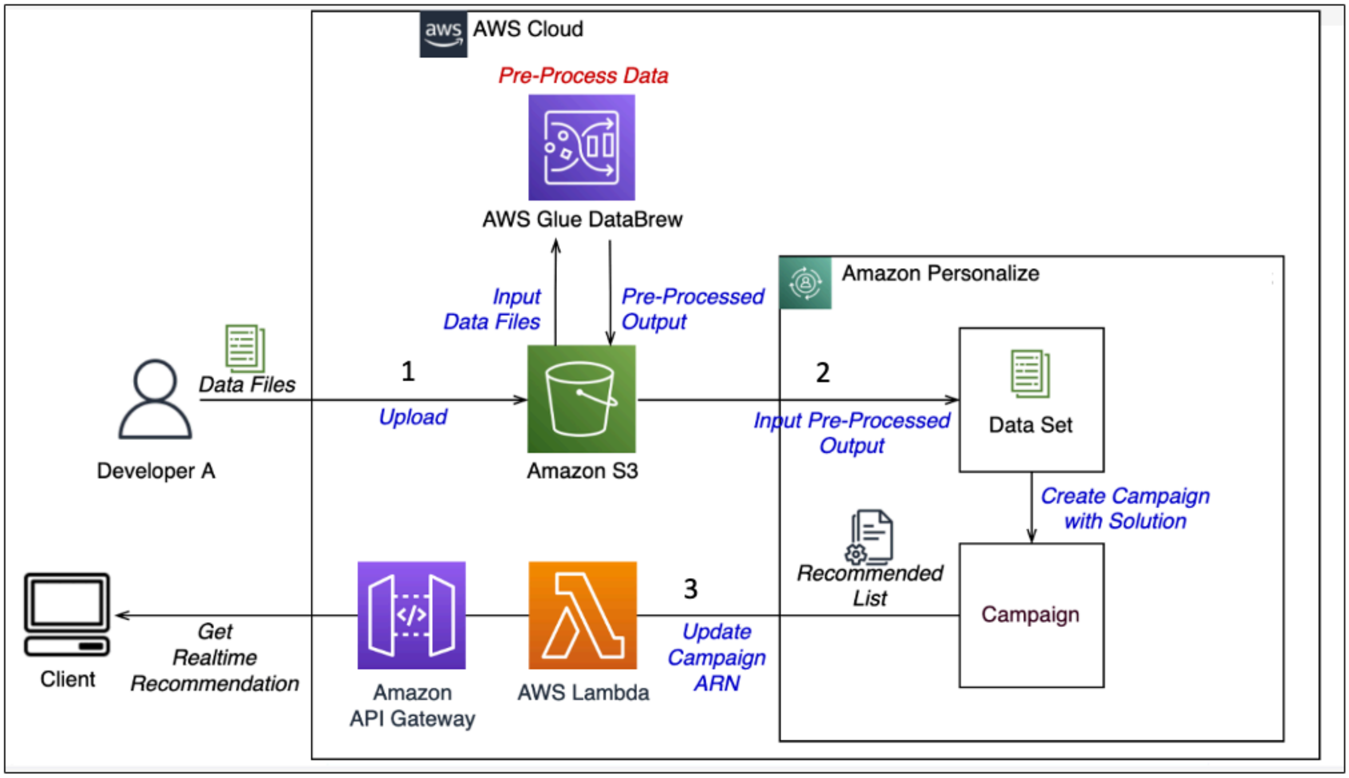
Build personalized recommendations architecture on AWS

Consumers are increasingly engaging with businesses through digital surfaces and touchpoints. Every touchpoint is an opportunity for businesses to give their consumers an experience tailored to their preferences and needs. In fact, consumers now expect personalized experiences from businesses. Market research tells us 63% of consumers see personalization as the standard level of service. Generic recommendations no longer meet customer expectations.

Some of the common use cases and solutions that fit under these personalized recommendations architecture patterns are as follows:

- **Personalization experience** — Personalize users' homepages with product or content recommendations based on their unique preferences and behavior. Personalize push notifications and marketing emails with individualized product, content, and promotional recommendations to help users find fresh, new products and content based on their unique tastes and preferences.
- **Personalization in retail** — Improving the customer experience by providing product recommendations based on their shopping history. Recommend similar items on product details pages to help users easily find what they are looking for.
- **Personalization in media and entertainment** — Deliver highly relevant, individualized content recommendations for videos, music, and e-books. Create personalized content carousels for every user based on their content consumption history.

The following diagram illustrates the system for building personalized recommendations on AWS.



Build real-time recommendations on AWS

The steps through the architecture are as follows:

1. **Data preparation —** Collect user interaction data such as item views, and item clicks. This plays a pivotal role in building a personalized recommendation. Once data is collected, upload your user interaction data into Amazon S3, then perform data cleaning using AWS Glue DataBrew to train the model in Amazon Personalize for real-time recommendations.
2. **Train the model with Amazon Personalize** — The data we use for modeling on Amazon Personalize consists of three types:
   - **The activity of your users, also known as events** — Examples include items your users click on, purchasing, and watching. The events you choose to send Amazon Personalize are dependent on your business domain. This dataset has the strongest signal for personalization, and is the only one required for personalization.
   - **Details about your items**, such as price point, category information, genre, and so on. Essentially, information in your catalog. This dataset is optional, but very useful for scenarios such as recommending new items. Personalize also enables customers to unlock the information trapped in their product descriptions, reviews, item synopses, or other

unstructured text using state-of-the-art natural language processing (NLP) techniques. Amazon Personalize automatically extracts key information about the items in your catalog to use when generating recommendations for your users.

- **Details about the users**, such as their location, age, and so on.

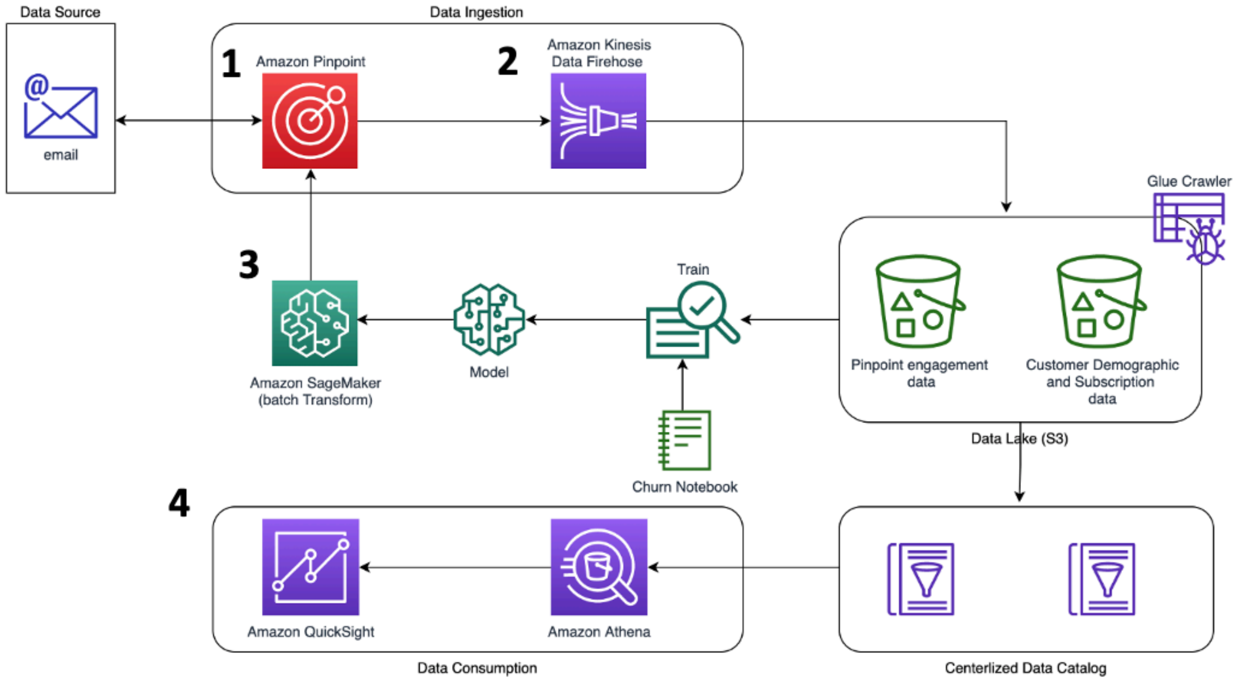
3. **Get real time recommendations —** Once you have the data, you can, in just a few clicks, get a custom, private, personalization model trained and hosted for you. You can then vend recommendations for your users through an API exposed through Amazon API Gateway.

Build near real-time customer engagement on AWS

This architecture is about maximizing customer engagement by delivering targeted messaging to your users in near real-time. For a business, engaging with customers is always an important day-to-day aspect, and those communications need to be clear, concise, and well-stated. There is even more of an impact if that messaging is also targeted and customized for the customer. Using this architecture, you can interact with that customer in near real-time and, utilizing advanced ML, access targeted messaging directly applicable to the customer. It enables the collection and analysis of campaign data, and the ability to use that data to create customized models within Amazon SageMaker. This allows you to be able to target a specific customer, or group of customers, and address their needs on a near real-time basis. This pattern can help you develop a production level SageMaker model, and a feedback loop that can unlock value from marketing campaign data.

Some of the common use cases and solutions that would work with this architecture are as follows:

- **Churn modeling –** Using all the relevant data stored within your data lake, you have the ability to create a churn model. This is an AI/ML algorithm using a known set of historical data to learn if similar customers will be more likely to leave in the future. This information can then be used to engage with the customer and save their business.
- **Recommendation modeling via customer segmentation –** Not all customers are alike, and they won’t all respond to the same type of marketing campaign. But by creating multiple personas and assigning the customer base to those personas, a marketing team can create personalized messaging or promotions for specific groups.
- **Marketing efficiency –** Collecting streaming event data such as opens and clicks, and allowing analysis of an ongoing campaign. Stream this data directly into your data lake, and augment this data with other data located there
- **Customer engagement –** Use transactional messaging to communicate with customers based upon information that has just happened.



Near real-time customer engagement architecture on AWS

The steps in this architecture are as follows:

1. **Initialize Pinpoint and create a marketing project** — You first need to configure your project to add your users and their contact info, such as email addresses. At this time, you also need to configure your metrics collection to ensure that you can capture your customer interactions to Amazon S3.
2. **Near real-time data ingestion** — Grab data from Amazon Pinpoint in near real-time through Amazon Data Firehose. Optionally, you can change this to an Amazon Kinesis Data Stream for near real-time use cases. This data is collected into S3, and used to both train the ML model and to analyze activity in your Amazon QuickSight dashboard.
3. **SageMaker model implementation** — Using a combination of Amazon Pinpoint engagement data with other customer demographic data, you can train a model that predicts the likelihood of customer churn (or segmentation, or other customer modeling insight). This is done in an iterative manner until you validate that your model is effective. Once that's done, you can set up a SageMaker endpoint to host this model and run inference against. This is done in a batch manner. It exports the results to Amazon Pinpoint and S3 for consumption.

4. **Data consumption with Athena and QuickSight** — View and analyze all of the data being collected from the Amazon Pinpoint engagement, and combine it with other data facts from your data lake with Amazon Athena. You can explore this data and run ad-hoc SQL queries to gain direct insight directly from S3 storage. Combine this with Amazon QuickSight to visualize these insights and share them with others in the organization.

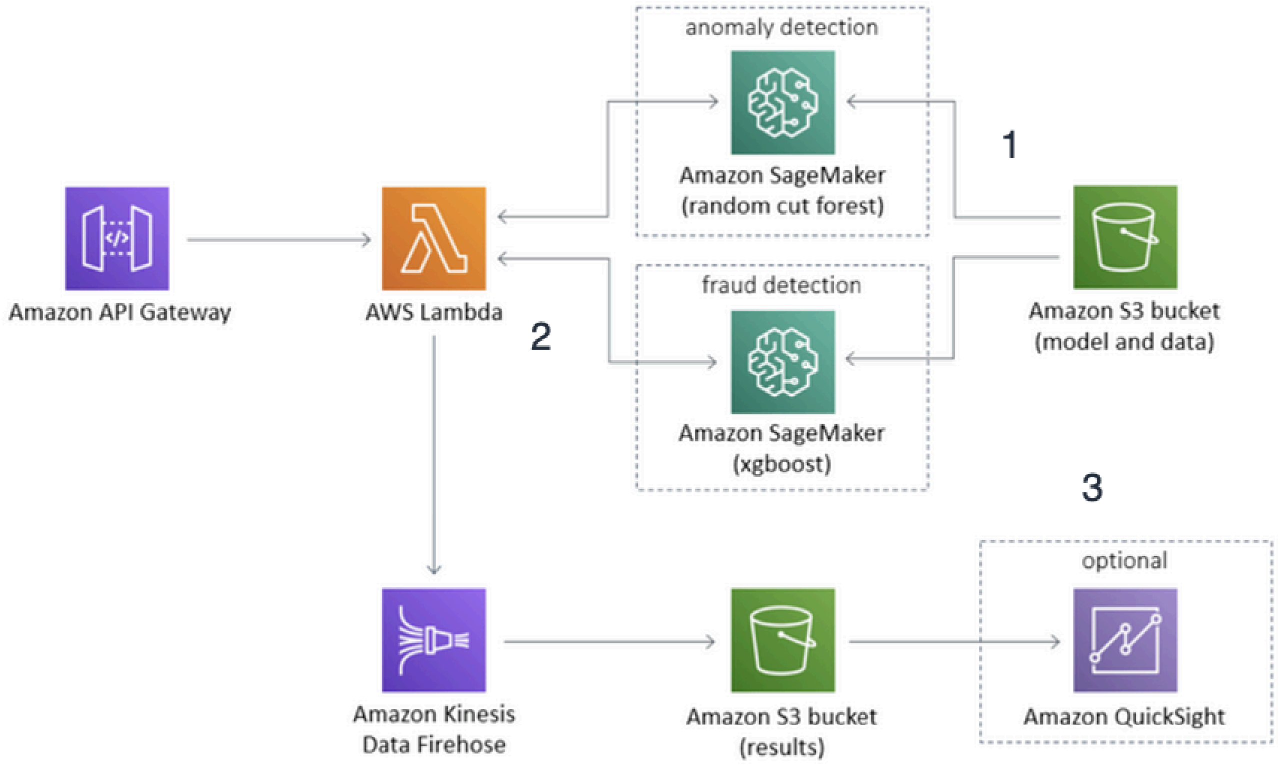
Build fraud detection architectures on AWS

The ability to move fast and serve customers in near real-time is becoming necessary for businesses to stop fraud in its tracks. Fraud represents a significant problem for businesses, and by addressing this challenge as soon as it occurs, the ability to recover from and reduce damages is greatly increased. Using this architecture, you can perform fraud detection in near real-time, and build fraud detection visualization dashboards for further analysis.

The AWS Solutions Implementation, Fraud Detection Using Machine Learning, enables you to run automated transaction processing. This can be on an example dataset or your own dataset. The included ML model detects potentially fraudulent activity, and flags that activity for review. The following diagram presents an architecture you can automatically deploy using the solution’s implementation guide, and accompanying AWS CloudFormation template.

Some of the common use cases and solutions that work with this architecture are as follows:

- **Fraud detection** – Organizations with online businesses have to be on guard constantly for fraudulent activity, such as fake accounts or payments made with stolen credit cards.
- **Near real-time analytics** – Use this architecture to understand the fraudulent activities data that you have streaming.


anomaly detection
Amazon SageMaker (random cut forest)
1
Amazon API Gateway
AWS Lambda
2
Amazon S3 bucket (model and data)
fraud detection
Amazon SageMaker (xgboost)
3
optional
Amazon Kinesis Data Firehose
Amazon S3 bucket (results)
Amazon QuickSight

Fraud detection architecture on AWS

The steps through the architecture are as follows:

1. **Develop a fraud prediction machine learning model —** The AWS CloudFormation Template deploys an example dataset of credit card transactions contained in an S3 bucket. An Amazon SageMaker notebook instance with different ML models is trained on the dataset.
2. **Perform fraud prediction** — The solution also deploys an AWS Lambda function that processes transactions from the example dataset. It invokes the two SageMaker endpoints that assign anomaly scores and classification scores to incoming data points. An Amazon API Gateway REST API initiates predictions using signed HTTP requests. An Amazon Data Firehose delivery stream loads the processed transactions into another Amazon S3 bucket for storage. The solution also provides an example of how to invoke the prediction REST API as part of the Amazon SageMaker notebook.
3. **Analyze fraud transactions —** Once the transactions have been loaded into S3, you can use analytics tools and services for visualization, reporting, ad-hoc queries, and more detailed analysis.

Considerations for building data driven applications on AWS

When you are building data driven applications on AWS, work backwards from what are important to your business application (for example, SLA's, cost, performance, consumer patterns) so you do not over-engineer nor select a service because it's the new shiny tool. Once you identify the key requirements, then leverage the reference patterns outlined to pick the right services that fits your use case. Some key considerations for building data driven applications are listed below.

- **User personas** — You must first identify user personas who are the correct stakeholders for your application, such as data engineers/developers, data analysts, BI teams, data scientists, external entities, and so on, and learn how they like to use the application.
- **Data sources** — Based on the application, there can be one or more diverse sets of data sources that may host the data required for your application. Treat data like an organizational asset, meaning it is no longer the property of individual departments. Break down data silos by thinking for the future, and make data available to the entire organization. Create a trusted data platform that can drive actionable insights with increased agility, improved customer experience, and increased operational efficiencies.
- **Data ingestion** — Bring your data into the storage layer using AWS purpose-built services. They are specially designed to ingest data from a wide variety of data sources at any velocity, to support unique data sources and data types. These purpose-built services can deliver data directly into data lakes and data warehouse storage layers, as illustrated.

- **Real-time data ingestion** — You can use either Amazon Kinesis Data Streams or Amazon Managed Streaming for Apache Kafka (Amazon MSK) to ingest near real-time data such as application logs, website clickstreams, and IoT telemetry data. If you have a streaming use case and you want to use an AWS native, fully managed service, consider Amazon Kinesis Data Streams. If open-source technologies are critical for your data processing strategy, you are familiar with Apache Kafka, you intend to have multiple consumers per stream, and you are looking for near real-time latency less than 200ms, we recommended you choose Amazon MSK rather than Amazon Kinesis.
- **Batch data ingestion** — You can ingest operational database data into a storage layer using AWS Database Migration Service (AWS DMS) and AWS Lake Formation blueprints. AWS DMS can be used to bring commercial databases such as Oracle and Microsoft SQL Server, open-source databases such as MySQL and PostgreSQL, Amazon databases such as Amazon RDS and

Amazon Aurora, data warehouses such as Teradata, and No-SQL databases such as MongoDB into the storage layer. AWS Lake Formation blueprints workflows make data ingestion simple by generating AWS Glue crawlers, jobs, and triggers that discover and ingest database data into a data lake. The blueprints are designed to support both one-time database data migration into data lakes, or incremental data updates into data lakes.

- **File shares** — One of the most commonly used methods to move data within and outside of the organization is by using files. A widely used pattern to move files is AWS DataSync, which makes it simple and fast to move large amounts of files from Network File System (NFS) shares, Server Message Block (SMB) shares, and Hadoop Distributed File Systems (HDFS) into Amazon S3 data lakes.
- **Software as a Service (SaaS) applications** — Using Amazon AppFlow makes it easy to ingest SaaS applications data into Lake House storage layer. It is a fully managed service, and it can connect to various SaaS applications such as Salesforce, ServiceNow, Datadog, Dynatrace, Zendesk, TrendMicro, Veeva, Snowflake, and Google Analytics. Amazon AppFlow then ingests SaaS applications data directly into the data lake, or staging tables in an Amazon Redshift data warehouse.
- **Partner data feeds** — AWS Transfer Family is a serverless service that provides secure file transfer protocol (FTP) endpoints and integrates with Amazon S3. It stores partner data feeds as S3 objects in the landing zone of the data lake.
- **Third-party data sources** — AWS Data Exchange has hundreds of commercial data products across various industries such as financial services, healthcare, retail, media, and entertainment. It includes hundreds of free data sets collected from popular public sources. You can ingest third-party data sets into an S3 data lake landing zone by subscribing to third-party data products. Amazon Data Exchange has native integration with Amazon Redshift, where you can share your Amazon Redshift database data sets as a data producer.
- **Custom data sources** — AWS Glue custom connectors can discover and integrate with a variety of data sources, such as SaaS applications and custom data sources. You can search and select connectors from the AWS Marketplace, and begin your data ingestion workflow in minutes to bring custom data sources’ data into an S3 data lake.
- **Query federation** — You can use Amazon Redshift Federated Query together with materialized views using scheduled refresh or Amazon Aurora query federation to minimize extract, transform, load (ETL) overhead wherever it's applicable, depending on your use case.

- **Data storage** — The storage layer consists of Amazon S3 and Amazon Redshift, besides the purpose-built databases outlined earlier in this whitepaper such as Amazon RDS and Amazon

DynamoDB. They provide an integrated storage layer for the data platform. You can ingest data from various sources into data lake raw zones in S3. Next, the processing layer transforms the data by applying schema, partition, and stores into cleaned or transformed zones in S3. Then the processing layer curates a transformed dataset by modeling it and joining it with other datasets, and stores it in a curated layer. The datasets from the curated layer in S3 are partially or fully loaded into Amazon Redshift data warehouse storage for the use cases that need very low latency, or need to run complex SQL queries.

- **Data catalog** — The data catalog layer is responsible for storing business and technical metadata from the storage layer. As you build out your modern data architectures on AWS, you will start ingesting hundreds to thousands of data sets from a variety of sources into data lakes and data warehouses. You will need a central data catalog for all datasets in the storage layer in a single place, making it easily available for retrieving information such as data location, format, and columns schema of data sets. AWS AWS Glue Data Catalog provides a central catalog to store metadata for all datasets hosted in the storage layer. Also, AWS Glue Data Catalog provides APIs to enable metadata management using custom scripts and third-party products.
- **Data processing** — Data processing pipelines can be multi-step, or scheduled on a regular interval. You can also invoke data processing pipelines based on event triggers. AWS Glue and AWS Step Functions provide all required components to build, orchestrate, and run pipelines that can scale to process huge data volumes for various use cases. You can build multi-step workflows using AWS Glue and Step Functions that can catalog, transform, and enrich the datasets, and load the data sets into raw, transformed, and curated zones in the Lake House storage layer. If you are processing for large data sets, use Amazon EMR for data processing. If your data processing is based on Apache Spark and you are looking for a serverless option, use AWS Glue instead of Amazon EMR for data sets less than 5TB. You can also use Amazon EventBridge for building event-driven data pipelines, and Amazon Athena for lightweight ETL.
- **Data consumption —** Data consumption layer is responsible for enabling user personas for purpose-built analytics. For example, you can use Amazon Athena for ad-hoc interactive queries, Amazon EMR for big data processing and ETL, Amazon Redshift for modern data warehouses, Amazon OpenSearch Service for operational analytics, Amazon Kinesis Analytics for near real-time analytics, and Amazon QuickSight for BI reports and dashboards.
- **Data governance** — You need to consider how to enforce data security and data governance in your data lake architecture. AWS Lake Formation integrates with AWS AWS Glue Data Catalog, and provides fine-grained access control so you can have a unified data governance across various AWS purpose-built analytics services.

- **Data prediction** — You can use multiple options for your data prediction solutions. You can use Amazon SageMaker-built algorithms for ML, or you can bring your algorithm into SageMaker for training and deployment of your ML model for data prediction. You can also use Amazon Web Services AI services to solve many business problems. Use Amazon Kendra for intelligent search, Amazon Fraud Detector for fraud detection, Amazon Personalize for personalized recommendations, Amazon Forecast for time-series forecasting, and so on. These services are developed, trained, and continuously optimized by Amazon data scientists, so there is no ML knowledge or experience required of the end user. These services can simply be accessed by the end user by using an API call. You can also consider ML built-in integrations for various data services such as Aurora ML, Neptune ML, Amazon Redshift ML, Athena ML, QuickSight ML, and Amazon OpenSearch Service ML for your data prediction solutions.

Conclusion

AWS provides various purpose-built data services to build data driven architecture for any use case. Each of these services has its own characteristics in terms of scalability, latency, and cost. Most importantly, they are deeply integrated with one another, making them easy to use. The reference architectures outlined in this whitepaper give you a great start, and accelerate time to solution. We encourage you to make your architecture extensible by using new data services as they become available.

Contributors

- Harsha Tadiparthi, Analytics Specialist Principal Solutions Architect, Amazon Web Services.
- Raghavarao Sodabathina, Enterprise Solutions Architect, Amazon Web Services

Further reading

For additional information, refer to:

- Analytics on AWS
- Machine Learning on AWS
- Databases on AWS
- Harness the power of your data with AWS Analytics (AWS blog post)
- *Derive Insights from AWS Modern Data* (AWS whitepaper)
- Design a data mesh architecture using AWS Lake Formation and AWS Glue (AWS blog post)

Document revisions

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

Change	Description	Date
Initial publication	Whitepaper published.	August 3, 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*.