High-level Design Document — Amazon SageMaker, Amazon Bedrock, Amazon S3 (ML)

## Scope

High-level design covering the three AWS services focused on Machine Learning workflows: Amazon SageMaker, Amazon Bedrock, and Amazon S3. Each service is described independently with purpose, audience, introduction, architecture & design, components, screenshots, observability, and common services.

# Amazon SageMaker

## Purpose

Provide a managed platform for building, training, tuning, registering and deploying ML models at scale — from experiments to production endpoints (real-time, batch, and serverless inference).

## Audience

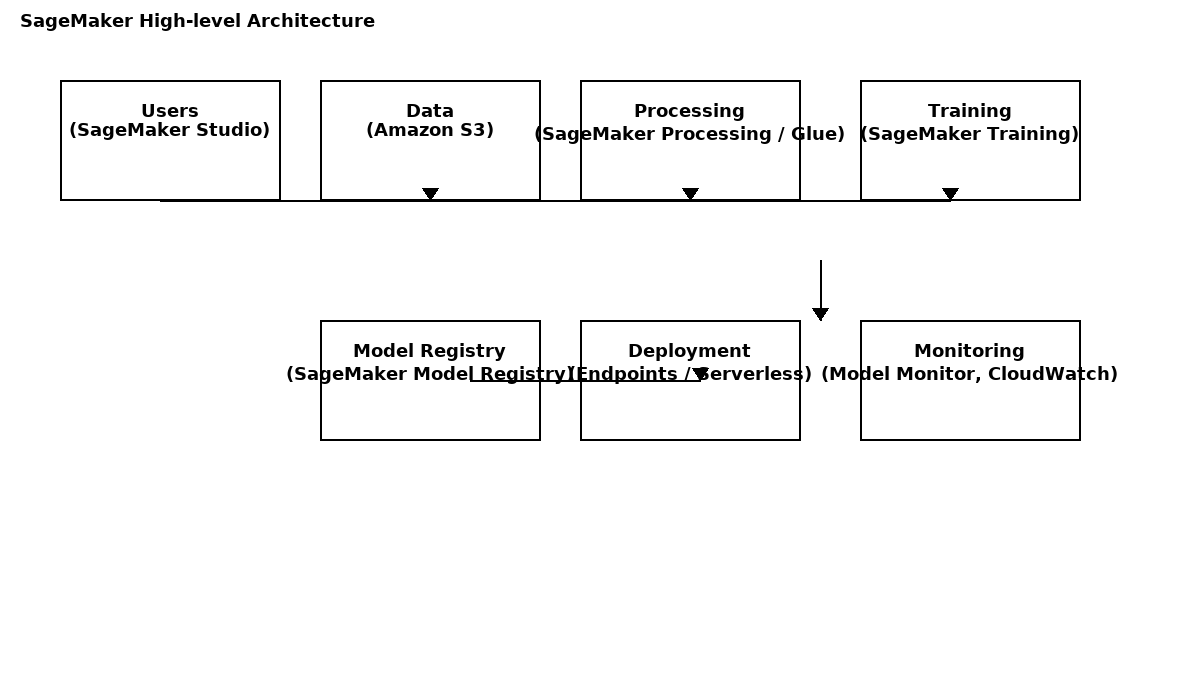
ML engineers, data scientists, MLOps engineers, DevOps/Cloud architects, and product managers.

## Introduction

SageMaker is a fully-managed ML service reducing heavy lifting for the ML lifecycle. It integrates tightly with other AWS services such as S3, ECR, IAM, CloudWatch, and KMS.

## Architecture & Design

Includes development layer (Studio), data layer (S3), training layer (distributed training), model registry, endpoints (real-time, serverless, batch), pipelines, and monitoring tools.



This diagram illustrates the ML workflow in SageMaker:

* **Users (Studio)** interact with Jupyter notebooks in SageMaker Studio to prepare data and build models.
* **Data (S3)** serves as the central storage for raw and processed datasets.
* **Processing (SageMaker Processing/Glue)** handles ETL, feature engineering, and preprocessing before training.
* **Training (SageMaker Training)** executes distributed or single-node training jobs using chosen ML/DL frameworks.
* **Model Registry** stores trained models with versioning and approval states.
* **Deployment (Endpoints)** provisions real-time or serverless inference endpoints for applications.
* **Monitoring (Model Monitor, CloudWatch)** provides metrics, drift detection, and alerting to ensure models remain reliable in production.

## Components & Services

Core: Studio, Notebooks, Processing, Training, Model Registry, Endpoints, Pipelines, Monitoring.  
Supporting: S3, ECR, IAM, KMS, VPC, CloudWatch, CloudTrail, Step Functions, SNS, SQS, CodePipeline.

## Observability

CloudWatch logs & metrics, S3 versioning & replication, AWS Backup, IaC templates for recovery, Model Monitor for drift detection.

## Common Services

IAM least-privilege, VPC endpoints, KMS encryption, CI/CD pipelines.

# Amazon Bedrock

## Purpose

Provide managed access to foundation models (FMs) from AWS and third-party providers via a unified API.

## Audience

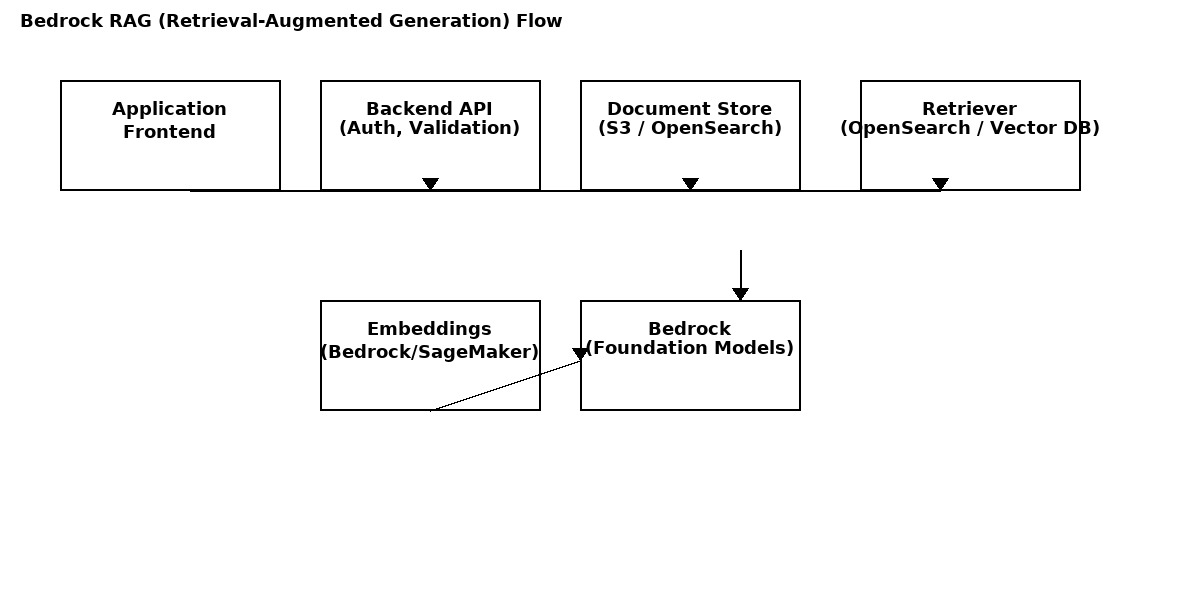
App developers, product teams, data scientists, security & compliance teams.

## Introduction

Bedrock provides API access to foundation models for building generative AI apps without managing infrastructure.

## Architecture & Design

Frontend -> Backend -> Bedrock API -> optional SageMaker/S3. Supports RAG using S3, OpenSearch, Kendra, or vector DBs.



This diagram explains how applications use **Amazon Bedrock** to integrate generative AI models:

* **Frontend Application** (web/mobile app) sends user queries.
* **Backend API** validates, enriches, and secures requests before forwarding.
* **Document Store (S3/OpenSearch)** stores domain knowledge or contextual documents.
* **Retriever (OpenSearch/Vector DB)** searches relevant documents based on embeddings.
* **Embeddings (Bedrock/SageMaker)** generate vector representations of text or documents for semantic search.
* **Bedrock (Foundation Models)** receives both user query + retrieved context and produces a grounded response.

## Components & Services

Core: Bedrock API, Model catalog, Fine-tuning (where available).  
Supporting: S3, OpenSearch, Kendra, IAM, KMS, CloudWatch, CloudTrail, API Gateway.

## Observability

CloudWatch metrics, logs, CloudTrail, S3 versioning for prompts, OpenSearch snapshots for backup.

## Common Services

Vector stores, embedding pipelines, API Gateway + WAF, data classification/PII detection.

# Amazon S3 (ML)

## Purpose

Durable object store for datasets, model artifacts, and ML pipeline outputs.

## Audience

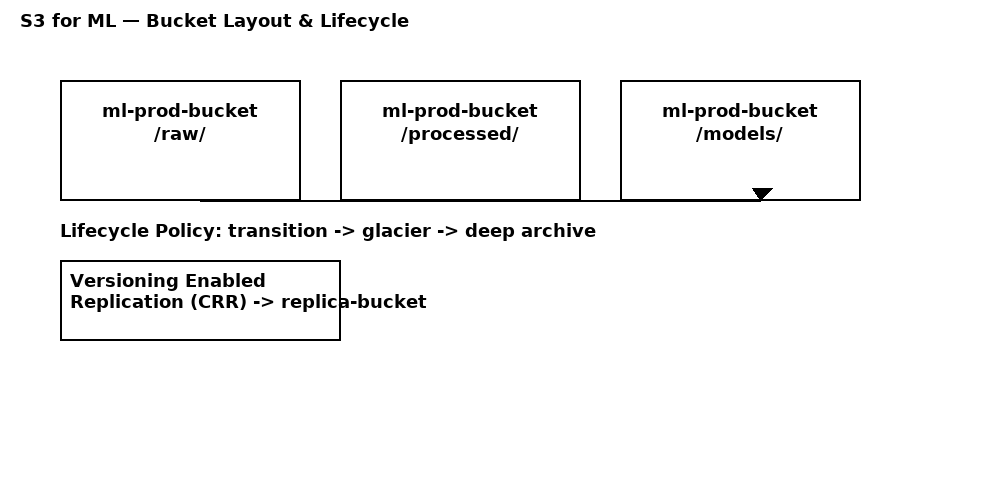
Data engineers, MLOps, data scientists, compliance teams.

## Introduction

S3 is the primary storage layer for ML workloads. Requires bucket design, lifecycle policies, encryption, and access controls.

## Architecture & Design

Buckets per environment, prefixes for raw/staging/processed/models/logs, versioning, lifecycle policies, replication, VPC endpoints.



## Components & Services

Core: Versioning, Lifecycle, Encryption (KMS), Object Lock, Replication, Access Points, Notifications.  
Supporting: Glue, Athena, EMR, Transfer/FileSync.

## Observability

S3 versioning, replication, lifecycle policies, CloudTrail data events, CloudWatch metrics.

## Common Services

IAM policies, KMS encryption, CloudTrail, EventBridge.