High-Level Design (HLD) — SageMaker, Bedrock, S3 (ML)

# 1) Purpose

This document provides a high-level design for an ML platform that leverages Amazon SageMaker, AWS Bedrock, and Amazon S3 as core services.   
It describes architecture patterns, components, operational controls, observability, backup & restore strategies, and common supporting services.   
It is intended to guide architecture decisions, implementation planning, and operations for building secure, scalable, and maintainable ML workloads on AWS.

# 2) Audience

- Cloud Architects and Solution Architects  
- ML Engineers and Data Scientists  
- DevOps / Platform Engineers  
- Security & Compliance Engineers  
- Engineering Managers and Project Owners

# 3) Introduction

Modern ML platforms require end-to-end capabilities: data storage and lifecycle management, training and model management, inference hosting,   
integration with foundation models, governance, monitoring, and secure operations. This HLD outlines a reference architecture that uses S3 as the primary data   
& model artifact store, SageMaker for training, experiment tracking and hosted inference, and Bedrock for access to managed foundation models where applicable.  
  
Goals:  
- Reproducible training and CI for models  
- Secure storage and access controls for datasets and models  
- Scalable, cost-effective training and inference  
- Observability, auditability and disaster recovery  
  
Constraints / Assumptions:  
- AWS is the cloud provider.  
- Workloads will run within customer VPCs with appropriate network controls.  
- Sensitive data will be encrypted at-rest and in-transit and subject to IAM and KMS controls.

# 4) Architecture and Design

Logical architecture includes:  
1. Data Layer (S3): raw/processed/feature buckets, model artifact buckets, audit logs.  
2. Compute & ML Layer: SageMaker Studio, Processing Jobs, Training Jobs, Model Registry, Endpoints, Bedrock APIs.  
3. Serving Layer: SageMaker real-time/serverless endpoints, VPC endpoints.  
4. Orchestration & CI/CD: Azure DevOps.  
5. Security & Governance: IAM, KMS, S3 policies, CloudTrail, Config, Guardrails.  
6. Observability & Operations: CloudWatch, Model Monitor, Debugger, centralized logging.  
  
Network design: VPC with private subnets, SageMaker, Bedrock, no public exposure.  
  
Data flow: ingest -> preprocess -> train -> register -> deploy -> monitor.  
  
Security patterns: IAM roles, encryption with KMS, S3 Object Lock, network isolation.

# 5) List of Components and Services:

|  |  |  |
| --- | --- | --- |
| Component | AWS Service | Purpose |
| Object store | Amazon S3 | Storage for raw, processed data and model artifacts |
| Notebook & Dev | SageMaker Studio | Interactive notebooks & dev experience |
| Training & Jobs | SageMaker Training / Processing | Distributed training and processing jobs |
| Model Registry | SageMaker Model Registry | Model versioning and approvals |
| Model Serving | SageMaker Endpoints | Real-time/serverless inference |
| Foundation Models | AWS Bedrock | Managed foundation models |
| Orchestration | SageMaker Pipelines / Step Functions | ML workflow orchestration |
| CI/CD | Azure DevOps | Training and deployment automation |
| Secrets & Keys | KMS / Secrets Manager | Key management and secrets storage |
| Monitoring | CloudWatch, Model Monitor | Metrics, logs, drift detection |
| Audit | CloudTrail, AWS Config | Audit trails |
| Networking | VPC, Endpoints | Network isolation & security |
| Identity | IAM | Access control and least privilege |

# 6) Screenshots

Suggested screenshots:  
- SageMaker Studio workspace  
- SageMaker Model Registry entries  
- S3 bucket configuration  
- CloudWatch dashboards  
- VPC & endpoint architecture diagrams

# 7) Observability (Backup and Restore)

Backup strategy:  
- Enable S3 versioning, lifecycle policies, replication  
- Persist SageMaker model artifacts in S3  
- Store notebooks in Git and backup to S3  
- Capture CloudTrail logs  
  
Restore strategy:  
- Recover from S3 versioning/replication  
- Redeploy models from artifacts  
- Recreate infrastructure via IaC  
- Restore archived datasets from Glacier  
  
Observability:  
- CloudWatch metrics & logs  
- SageMaker Model Monitor for drift  
- CloudTrail for auditability  
- Tracing with X-Ray or structured request IDs

# 8) Common Services

- Identity & Access: IAM, AWS Organizations, Control Tower  
- Secrets: AWS Secrets Manager, Parameter Store  
- Encryption: AWS KMS (CMKs)  
- Logging & Audit: CloudTrail, CloudWatch, Config  
- Monitoring & Alerts: CloudWatch Alarms, SNS, PagerDuty integration  
- CI/CD: Azure DevOps / CodePipeline  
- Cost Management: AWS Cost Explorer, Budgets  
- Governance: SCPs, Config Rules, Security Hub  
- Network: Transit Gateway, PrivateLink, VPC Endpoints