AWS Well-Architected Framework

Modern Industrial Data Technology Lens

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Modern Industrial Data Technology Lens: AWS Well-Architected Framework

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Abstract and introduction ..... **i**
Custom lens availability 2
Definitions **3**
Design principles **5**
Scenarios **7**
MFGSCE1: Modern Industrial Data Architecture (MIDA) 7
Core architectural characteristics 8
MFGSCE2: Industrial data catalog 12
Industrial data cataloging on AWS 12
Integrating industrial data sources into the catalog 13
Metadata management and governance 16
Best practices 17
MFGSCE3: Industrial data mesh 19
Data mesh design goals 20
Data mesh end to end workflow 20
Access patterns 22
Industrial service bus 23
MFGSCE4: Digital thread: contextualization and knowledge graph 25
Digital thread solution framework 26
Digital thread reference architecture 27
MFGSCE5: Computer vision for automated quality inspection 29
Data acquisition and management challenges 29
Data preparation pipeline 30
MLOps for manufacturing vision systems 31
AWS architecture for vision-based quality inspection 32
Automated quality inspection architecture 34
Resources 35
Operational excellence **36**
Design principles 36
Organization 38
Prepare 38
MIDAOPS01-BP01 Implement a process to periodically review compliance requirements relevant to your industrial data infrastructure 38
MIDAOPS02-BP01 Use enhanced monitoring and alerting in the cloud and at the edge 40

Operate 42
MIDAOPS03-BP01 Define meaningful KPIs and metrics 42
MIDAOPS03-BP02 Implement real-time monitoring and alerting capabilities 43
MIDAOPS03-BP03 Implement predictive analytics and proactive anomaly detection 45
Evolve 46
Security 47
Design principles 47
Security foundations 50
MIDASEC01-BP01 Create a security Shared Responsibility Model 50
MIDASEC01-BP02 Standardize security baseline and implement role-based access controls 51
MIDASEC01-BP03 Secure machine-to-machine and human-to-machine access 52
MIDASEC01-BP04 Automate monitoring and reporting with cloud-ready compliance tools 53
MIDASEC01-BP05 Implement incident response playbooks 54
MIDASEC01-BP06 Establish a communication protocol between IT and OT systems 55
Identity and access management 56
MIDASEC02-BP01 Enforce least privilege and security policies to control system access 57
MIDASEC02-BP02 Enable multi-factor authentication (MFA) and token authorization (TA) 58
MIDASEC02-BP03 Use centralized access management tools 58
MIDASEC02-BP04 Develop a mechanism for regular review of IAM roles and policies 59
Detection 60
MIDASEC03-BP01 Integrate and enforce privacy by design principles 61
MIDASEC03-BP02 Implement industrial data classifications and protection policies 61
MIDASEC03-BP03 Identify source data and set data classifications 62
MIDASEC03-BP04 Implement industrial encryption policies 63
MIDASEC03-BP05 Implement data retention policies for each data class 64
Infrastructure protection 65
MIDASEC04-BP01 Segment OT networks from IT systems 65
MIDASEC04-BP02 Use firewalls, perimeter networks, and dedicated network zones 66
Data protection 67
MIDASEC05-BP01 Define access permissions 68
MIDASEC05-BP02 Build user identity solutions 69
MIDASEC05-BP03 Implement data authorization models 69
MIDASEC06-BP01 Use secure data exchange protocols 70

MIDASEC06-BP02 Establish clear data ownership and sharing agreements 71
Incident response 72
MIDASEC07-BP01 Deploy continuous monitoring tools 72
MIDASEC07-BP02 Implement SIEM systems 73
Application security 74
MIDASEC08-BP01 Align security controls with industry standards 75
MIDASEC08-BP02 Conduct regular security awareness programs 75
MIDASEC09-BP01 Apply secure coding practices for applications and data integrations 76
MIDASEC09-BP02 Perform regular vulnerability scans and penetration tests 77
MIDASEC09-BP03 Automate patch management for ICS and connected data infrastructure 78
Reliability 80
Design principles 80
Foundations 82
MIDAREL01-BP01 Monitor manufacturing-specific service quotas 82
MIDAREL01-BP02 Scale network architecture for manufacturing operations 84
Workload architecture 86
MIDAREL02-BP01 Design resilient industrial integration patterns 87
MIDAREL02-BP02 Verify data consistency and availability across OT/IT systems through redundancy and failover mechanisms 89
MIDAREL02-BP03 Enable automated recovery mechanisms 91
Change management 94
MIDAREL03-BP01 Implement dynamic scaling for shop floor data ingestion pipelines 94
Failure management 96
MIDAREL04-BP01 Implement a multi-layered backup strategy 97
MIDAREL04-BP02 Use cloud replication for manufacturing data resilience 99
MIDAREL04-BP03 Use versioning and automated backup procedures 102
MIDAREL05-BP01 Design a multi-Region disaster recovery strategy 104
MIDAREL05-BP02 Implement and regularly test DR procedures 106
Performance efficiency 110
Design principles 110
Real-time data access 112
MIDAPERF01-BP01 Use time series database for real-time analytics and data lake for long-term storage 112
MIDAPERF01-BP02 Compress, sample and summarize data at edge, before sending to the cloud environment. 115

Event-driven architecture 118
MIDAPERF02-BP01 Implement event-driven architectures for manufacturing systems 118
MIDAPERF02-BP02 Use historical cloud usage data aligned with production schedules and business forecasts 121
Data processing infrastructure 123
MIDAPERF03-BP01 Use cloud observability tools for manufacturing systems 124
MIDAPERF03-BP02 Implement comprehensive performance measurement for manufacturing data infrastructure 128
Troubleshoot and optimize pipelines 131
MIDAPERF04-BP01 Implement end-to-end observability for manufacturing data pipelines 131
MIDAPERF04-BP02 Decouple data ingestion from processing in manufacturing systems .. 134
Pre-process data at edge 137
MIDAPERF05-BP01 Implement edge data pre-processing 138
MIDAPERF05-BP02 Optimize storage and access for current manufacturing data 140
Data storage organization 143
MIDAPERF06-BP01 Implement efficient storage and access for historical manufacturing data 143
Cost optimization 147
Design principles 147
Forecast and optimize 150
MIDACOST01-BP01 Implement data-driven cost management using AWS cost tools and manufacturing data 150
MIDACOST01-BP02 Configure automated cost monitoring and alerts for manufacturing workloads 153
Data lifecycle 155
MIDACOST02-BP01 Track resources over their lifetime 156
MIDACOST02-BP02 Implement manufacturing-aware resource decommissioning process 159
MIDACOST02-BP03 Automate production-aware resource decommissioning 161
MIDACOST02-BP04 Implement manufacturing-specific data retention policies 163
MIDACOST02-BP05 Develop cloud resource policies aligned with manufacturing operations 165
MIDACOST02-BP06 Implement manufacturing-aware cost controls 167
Migration optimization 169

MIDACOST03-BP01 Analyze all components of your industry workloads for migration and modernization 169
Forecast resource provisioning 172
MIDACOST04-BP01 Perform an analysis on the historical manufacturing workloads 172
Balancing resources 175
MIDACOST05-BP01 Implement a buffering or throttling approach 175
MIDACOST05-BP02 Implement dynamic resource provisioning 178
Sustainability 181
Design principles 181
Region selection 183
MIDASUS01-BP01 Select Regions that offer services required by Manufacturing organizations that maximizes the reduction of your carbon footprint 183
Alignment to demand 186
MIDASUS02-BP01 Actively manage workloads and resource allocation based on production demands. 186
Data management 189
MIDASUS03-BP01 Implement edge data and cross-region movement strategies. 189
MIDASUS04-BP01 Design backup strategies that focus on critical data. 191
Hardware and services 193
MIDASUS05-BP01 Implement optimized compute, storage and networking hardware. 194
Process and culture 196
MIDASUS06-BP01 Implement sustainability driven manufacturing processes. 197
Conclusion 200
Contributors 201
References 202
Document revisions 203
AWS Glossary 204

Modern Industrial Data Technology Lens

Publication date: **October 30, 2025** (*Document revisions*)

This whitepaper describes the AWS Well-Architected Framework Lens on modern industrial data technology. It provides guidance to help customers apply the well-architected best practices in the design, delivery, and maintenance of AWS environments. We address general design principles as well as specific best practices and implementation guidance in six conceptual areas that we define as the pillars of the Well-Architected Framework.

The AWS Well-Architected Framework helps you understand and assess the pros and cons of decisions you make while building systems on AWS. By using the Framework, you can learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. The process for reviewing an architecture is a constructive conversation about architectural decisions and is not an audit mechanism. We believe that having well-architected systems greatly increases the likelihood of business success.

While the Framework provides foundational guidance, certain workloads, such as those in modern industrial data (MID), require specialized focus due to their unique challenges and requirements. MID workloads face distinct challenges including:

- OT and IT convergence and integration requirements
- Real-time data processing needs for production environments
- Industrial IoT device management and edge computing demands
- Complex regulatory compliance across manufacturing operations
- Mission-critical system reliability requirements

The AWS Well-Architected Framework documents a set of foundational questions that help you understand if a specific architecture aligns well with cloud best practices. The Framework provides a consistent approach to evaluating systems against the qualities you expect from modern cloud-based systems, as well as the remediation that would be required to achieve those qualities. As AWS continues to evolve and we continue to learn more from working with our customers, we will continue to refine the definition of well-architected.

AWS Solutions Architects (SAs) have extensive experience helping customers architect and optimize their MID solutions. Through thousands of customer engagements, we've aggregated best practices

specifically focused on manufacturing and industrial workloads. This lens builds upon the core Well-Architected Framework to provide manufacturing-specific guidance.

This framework is intended for technology leaders in manufacturing, including Chief Technology Officers (CTOs), industrial automation engineers and architects, manufacturing IT teams, system integrators, operations technology professionals, and anyone involved in designing and operating manufacturing systems on AWS. It describes AWS best practices and strategies to use when designing and operating a cloud workload, and provides links to further implementation details and architectural patterns. For more information, see the AWS Well-Architected homepage.

AWS also provides a service for reviewing your workloads. The AWS Well-Architected Tool (AWS WA Tool) is a service in the cloud user console that provides a consistent process for you to review and measure your architecture using the AWS Well-Architected Framework. The AWS WA Tool provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

To help you apply best practices, we have created AWS Well-Architected Labs, which provide you with a repository of code and documentation to give you hands-on experience implementing best practices. We also have teamed up with select AWS Partner Network (APN) Partners, who are members of the AWS Well-Architected Partner program. These APN Partners have deep AWS knowledge and experience and can help you review and improve your workloads.

How to use this lens

- Review the general design principles for Modern Industrial Data Architecture (MIDA)
- Evaluate your workloads against the manufacturing-specific best practices in each pillar
- Use the provided questions and guidance to identify areas for improvement
- Leverage the implementation patterns and examples to optimize your architecture.

Custom lens availability

Custom lenses extend the best practice guidance provided by AWS Well-Architected Tool. AWS WA Tool allows you to create your own custom lenses, or to use lenses created by others that have been shared with you.

To determine if a custom lens is available for the lens described in this whitepaper, reach out to your Technical Account Manager (TAM), Solutions Architect (SA), or Support.

Definitions

This section provides key terminology used throughout this lens, focusing specifically on modern industrial data concepts and manufacturing-specific definitions. These definitions complement the standard AWS Well-Architected Framework terminology.

- **Operational technology (OT)**: Systems and equipment used to monitor and control industrial processes on the shop floor, such as sensors, PLCs, and SCADA systems
- **Information technology (IT)**: Technologies used to manage business operations and data, including enterprise systems like ERP, MES, and cloud computing systems.
- **Programmable logic controller (PLC)**: An industrial computer used to control machinery and processes on the factory floor, running real-time logic based on sensor inputs.
- **Supervisory control and data acquisition (SCADA)**: A system that provides centralized monitoring and control of industrial equipment and processes, often across multiple sites.
- **Manufacturing execution system (MES)**: Software that manages and monitors production operations in real time, bridging the gap between the factory floor (OT) and business systems (IT).
- **Enterprise resource planning (ERP)**: A business management system that integrates core processes like finance, inventory, and procurement, often linking with MES for end-to-end visibility.
- **Message Queuing Telemetry Transport (MQTT)**: A lightweight messaging protocol used in manufacturing to transmit data from machines and sensors to monitoring systems in real time.
- **Advanced Message Queuing Protocol (AMQP)**: A protocol used to provide reliable communication between systems in manufacturing, often for integrating OT and IT layers.
- **Open Platform Communications Unified Architecture (OPC UA)**: A standard protocol for secure, system-independent communication between industrial equipment and software systems, enabling seamless data exchange across the manufacturing environment.
- **Reliability**: The ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle.
- **Resilience**: The ability of a workload to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions, such as misconfigurations or transient network issues.

- **Mean time to detection (MTTD)**: The average time required to detect a failure or anomaly in manufacturing systems after it occurs.
- **Mean time to resolution (MTTR)**: The average time taken to fully resolve an incident from the moment it is detected, including the time to restore normal manufacturing operations.
- **Recovery Time Objective (RTO)**: The maximum acceptable time to restore a manufacturing process or system after a disruption.
- **Recovery Point Objective (RPO)**: The maximum acceptable period of data loss measured in time. For manufacturing systems, this defines how much operational data loss can be tolerated in a recovery scenario.
- **Data mesh producer**: Any entity which offers a data product through the data mesh.
- **Data mesh consumer**: Any entity who subscribes to a data product in the data mesh.
- **Data product**: Today, a data product is scoped to be only an AWS Lake Formation table or database. In the future, this definition may expand.
- **Digital thread**: A framework that connects data flows and provides an integrated view of an asset throughout the manufacturing lifecycle, from design through production and in-service operation.
- **Manufacturing data lake**: A centralized repository that allows storing structured and unstructured manufacturing data at scale
- **Industrial data catalog**: A metadata management solution that helps manufacturing organizations find, organize and access their industrial data assets

Design principles

The modern industrial data architecture (MIDA) establishes a comprehensive framework for managing manufacturing data across the enterprise. These design principles address the unique challenges of industrial environments while enabling digital transformation initiatives.

- **Systematically contextualize data:** Manufacturing environments traditionally maintain separate IT and OT data infrastructures, creating operational silos that limit visibility and insights. MIDA addresses this through systematic data contextualization that bridges these domains. This involves mapping relationships between equipment telemetry, production metrics, quality data, and business KPIs. For example, sensor data from manufacturing equipment is automatically enriched with metadata about the production line, product being manufactured, and current process parameters. This contextual integration enables root cause analysis of quality issues, predictive maintenance optimization, and real-time production monitoring with full business context.
- **Enable data democratization:** While collecting data from shop floor equipment into an industrial data lake provides value, the true potential is realized through secure, organization-wide access. MIDA implements role-based access controls and data governance policies that enable different personas to use manufacturing data appropriately. Using governed, secure interfaces, data scientists can access raw sensor data for advanced analytics, process engineers can monitor real-time production metrics, and operations teams can track KPIs. This democratization drives innovation and continuous improvement while maintaining security and compliance.
- **Implement strategic data modeling:** MIDA employs sophisticated data modeling approaches optimized for manufacturing use cases. Hierarchical modeling supports equipment and process relationships, reflecting the natural organization of manufacturing operations, while graph modeling enables complex relationship analysis, such as tracing quality issues through production processes. Time-series optimization handles high-frequency sensor data, with intelligent data collection frequencies ranging from millisecond-level for critical processes to five-minute intervals for general monitoring. Unified data schemas standardize naming conventions and relationships across facilities, providing consistency in data interpretation and analysis.
- **Establish unified industrial communication:** The manufacturing industry employs various data standards that must be seamlessly integrated. MIDA implements a unified communication layer that supports industrial protocols like OPC UA for equipment connectivity, MQTT and AMQP for reliable messaging, RESTful APIs for enterprise system integration, and protocol translation

services for legacy systems. This standardization provides a reliable data flow while embracing modern integration patterns, enabling seamless communication across the manufacturing technology stack.

- **Implement comprehensive data lifecycle management:** MIDA implements comprehensive data lifecycle management through a multi-tier storage architecture using AWS IoT SiteWise for recent operational data. Automated archival policies based on data age and business value provides for efficient data management, while data retention rules align with compliance requirements. Cost-optimized storage selection based on access patterns maintains performance for active data while optimizing costs for historical information, creating a balanced approach to data management across the manufacturing environment.
- **Enable advanced analytics and insights:** The architecture enables multiple analytical approaches through real-time monitoring using streaming analytics, historical analysis through data warehousing capabilities, machine learning for predictive maintenance and quality optimization, and ETL pipelines for data preparation and transformation. These capabilities are supported by automated data quality checks, standardized metrics calculations, and visualization tools that provide actionable insights to different user groups. Through these integrated analytical capabilities, manufacturing organizations can derive maximum value from their operational data while maintaining system performance and reliability.

Through these principles, MIDA provides a scalable, secure foundation for manufacturing organizations to use their data assets effectively while maintaining operational excellence and driving continuous improvement.

Scenarios

Modern industrial data architectures must address diverse manufacturing challenges across the enterprise, from shop floor data collection to enterprise-wide analytics. This section presents five key scenarios that illustrate common manufacturing use cases and their implementation on AWS. These scenarios include establishing a modern industrial data architecture (MIDA) foundation, implementing industrial data catalogs for improved data discovery, creating data mesh architectures for scalable data sharing, building digital threads with knowledge graphs for end-to-end visibility, and deploying computer vision systems for automated quality inspection. Each scenario provides detailed architectural guidance, implementation considerations, and AWS service configurations to help manufacturing organizations successfully deploy these solutions. By understanding these reference architectures and implementation patterns, organizations can accelerate their digital transformation initiatives while maintaining operational excellence, security, and cost optimization.

Scenarios

- MFGSCE1: Modern Industrial Data Architecture (MIDA)
- MFGSCE2: Industrial data catalog
- MFGSCE3: Industrial data mesh
- MFGSCE4: Digital thread: contextualization and knowledge graph
- MFGSCE5: Computer vision for automated quality inspection

MFGSCE1: Modern Industrial Data Architecture (MIDA)

AWS industrial customers serving multiple industries have production facilities across the globe. They generate massive amounts of data from the various industrial machines, sensors, and equipment on their factory floors. This includes real-time telemetry data on equipment performance, quality control measurements, maintenance logs, and supply chain information.

In the past, this data was siloed across disparate systems and locations, making it extremely challenging for the manufacturer to gain a unified view of their operations. Engineers and analysts struggled to access the data they needed to drive process improvements, predict equipment failures, and optimize production. The lack of a cohesive data strategy hindered the manufacturer's ability to make informed, data-driven decisions and stay competitive.

MIDA by AWS services and ISV Industrial DataOps solutions address this problem. MIDA helps customers break down data silos while preserving their investments in purpose-built data stores. It creates an integrated environment of data lakes, data warehouses, and other specialized data stores, while enabling unified governance and seamless data movement across the architecture.

Core architectural characteristics

The well-architected MIDA supports the following characteristics.

Performance and scalability: The architecture should deliver high-performance data management capabilities that scale with manufacturing operations:

- Efficient ingestion and storage of petabyte-scale industrial data
- Support for billions of daily sensor readings and machine values
- Seamless scaling from single-plant pilots to multi-site deployments
- Cost-effective storage and processing optimization

Data services and integration: MIDA should implement purpose-built services to meet diverse manufacturing requirements:

- Real-time monitoring through streaming analytics and time-series databases
- Regulatory compliance through secure data warehousing
- Integration with standard industrial protocols (MQTT, OPC-UA)
- Support for open data formats like Parquet

Operational architecture: The framework must provide reliable operations across distributed manufacturing environments:

- Hybrid edge-cloud deployment for mission-critical operations
- High availability and local processing at the edge
- Flexible data movement patterns (inside-out, outside-in, perimeter)
- Decoupled storage and compute for optimal resource utilization

Security and governance: MIDA must have built-in controls for data integrity and compliance:

- Comprehensive data lifecycle management

- End-to-end security and access controls
- Data quality monitoring and observability
- Clear governance policies and procedures

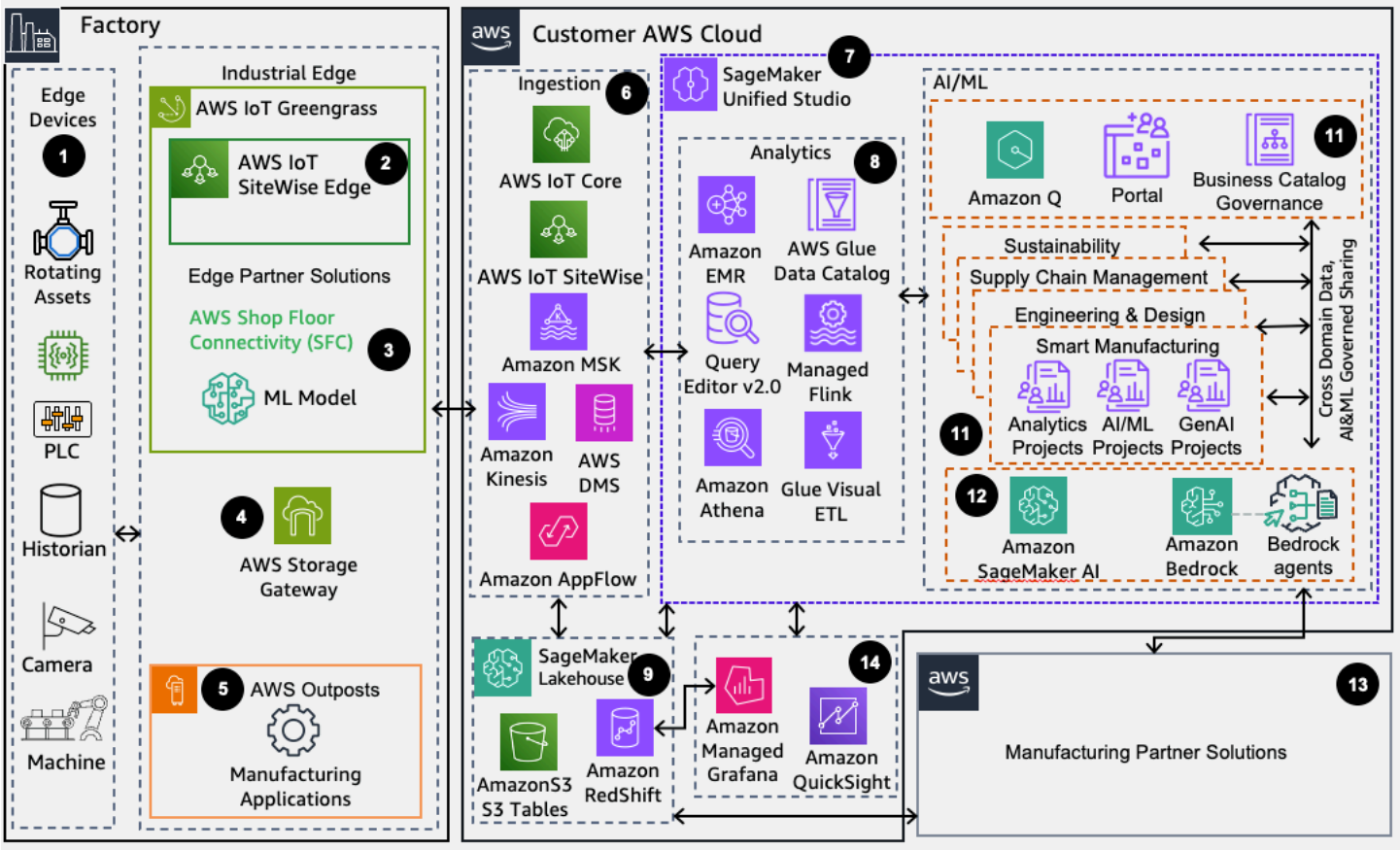
Industrial service bus: The MIDA also incorporates an industrial service bus to enable seamless integration and data exchange between OT and IT systems. The industrial service bus acts as a middleware layer that standardizes communication protocols like OPC UA, MQTT, and AMQP, providing a unified messaging backbone for data flow between sensors, control systems, enterprise applications, and cloud environments. This standardization is essential for real-time monitoring, control, and optimization of industrial processes.

AWS IoT Core serves as a cloud-based industrial service bus, supporting protocols like MQTT, HTTPS, and WebSockets to enable secure, low-latency, and reliable bi-directional communication between industrial equipment and cloud services. This architecture is further enhanced by AWS IoT SiteWise, which uses OPC UA to collect, structure, and analyze data from multiple sensors and industrial assets in near real-time, providing a unified data model for analytics and machine learning workloads.

This architecture enables manufacturers to:

- Accelerate digital transformation initiatives
- Scale analytics across global operations
- Provide operational reliability and compliance
- Drive continuous improvement through data-driven insights

The MIDA provides the foundation for manufacturers to build data capabilities while maintaining operational excellence and security. Its flexible, scalable design supports both current needs and future innovation in manufacturing analytics and automation.



ADD FIGURE CAPTION HERE

1. Identify information related to industrial activities from on-premises equipment.
2. Collect real-time data from edge devices using AWS IoT SiteWise Edge and transmit data streams securely to AWS IoT SiteWise in the cloud. Using partners like Litmus, Domatica's EasyEdge, Siemens, and Belden accelerates your integration with AWS IoT SiteWise Edge.
3. Connect to your edge devices through AWS Shop Floor Connectivity (SFC) Framework, an open-source solution from AWS, using multiple industrial protocols to securely stream data to AWS Cloud services. Deploy and run your cloud-developed machine learning models at the edge through AWS IoT Greengrass for defect detection and anomaly inference.
4. Connect your on-premises applications to Amazon S3 through AWS Storage Gateway using NFS and SMB file shares.
5. Extend AWS infrastructure and services to your premises with AWS Outposts, a fully managed service. Run your manufacturing-specialized applications on AWS services locally at your plant and integrate with AWS cloud infrastructure.
6. Process diverse data types through the data ingestion layer using AWS services. Stream real-time data through AWS IoT SiteWise, AWS IoT Core, Amazon Kinesis, and Amazon MSK. Transfer

structured data from legacy on-premises systems and data warehouses using AWS DMS and AWS Glue. Store and process unstructured and semi-structured data with Amazon S3 and Amazon AppFlow. You can use AppFlow to extract, create and update data with ERPs, like SAP.

7. Access functionality and tools from AWS Analytics and AI/ML services through Amazon SageMaker AI Unified Studio's single development environment. Find, access, and query data and AI assets across your organization. Collaborate on projects to build and share analytics and AI artifacts, including data, models, and generative AI applications, in a secure environment.
8. Process, model, and analyze your combined OT and IT data through AWS analytics services accessed using Amazon SageMaker AI Unified Studio's Portal. Catalog your data using AWS Glue Data Catalog, transform it with Glue Visual ETL, and run SQL analytics through Amazon Athena. Process streaming data with Managed Flink and perform distributed data processing using Amazon EMR to generate business insights.
9. Unify your data across Amazon S3 data lakes, including S3 Tables and Amazon Redshift data warehouses with Amazon SageMaker AI Lakehouse. Build powerful analytics and AI/ML applications on a single copy of data using the Apache Iceberg–compatible tools and engines.
10. Organize your assets, users, and projects within Amazon SageMaker AI Unified Studio domains. Create single or multiple domains to match your enterprise structure. Collaborate in projects to manage data assets, analyze data, develop ML models, and build generative AI applications for specific business needs.
11. Enrich metadata from your technical catalogs with business context using Amazon SageMaker AI Catalog. Discover and access approved data and models through semantic search powered by generative AI. Monitor data quality, track lineage, and enforce access policies centrally in Amazon SageMaker AI Unified Studio.
12. Build, train, and deploy machine learning models and generative AI capabilities using Amazon SageMaker AI AI and Amazon Bedrock. Use agentic AI to improve manufacturing, optimize supply chain, and acquire digital twin agents for engineering and design, which all impact your sustainability.
13. Integrate with cloud-hosted manufacturing partner solutions (ERP, supply chain, maintenance), including MCP servers for industrial knowledge sources.
14. Visualize your data with Amazon Managed Grafana natively from Amazon RedShift or from Amazon S3 using Amazon Athena. Build dashboards with Quick and Amazon Athena.

MFGSCE2: Industrial data catalog

In modern manufacturing (industry 4.0), organizations collect massive volumes of data from disparate sources, like IoT sensors and control systems on the shop floor, MES and ERP applications, and even vision systems. Without a unified catalog and governance, this data often ends up in silos, making it difficult for engineers and analysts to find, trust, and use the right data. An industrial data catalog addresses this challenge by cataloging the data assets (with metadata) in one place so that OT and IT professionals can discover what data exists, understand its context, and access it for analytics or machine learning. In an Industry 4.0 data environment, the data requires cataloging so that consumers can identify what is available.

A data catalog functions as a critical metadata management solution, centralizing data assets from disparate sources such as ERP, MES, and SCADA systems, as well as industrial internet of things (IIoT) devices. It enables comprehensive data governance by automating the ingestion, classification, and indexing of structured and unstructured data across the enterprise. This central repository allows for efficient metadata tagging, lineage tracking, and semantic search capabilities so that engineers, data scientists, and analysts can quickly locate, access, and utilize data for advanced analytics, digital twin modeling, and process optimization. The data catalog supports standardized data definitions and taxonomies, enhancing data interoperability and compliance with industry frameworks like ISO 8000 for data quality and ISO 10303 for product data representation.

Data catalogs in manufacturing integrate with advanced analytics tools and machine learning systems, enabling automated anomaly detection, predictive maintenance, and real-time process monitoring. Data catalogs also facilitate data lineage and impact analysis, which are essential for root cause analysis and continuous improvement initiatives. They support data security through role-based access controls (RBAC), improving compliance with data privacy regulations such as GDPR and CCPA while maintaining operational agility.

Industrial data cataloging on AWS

An AWS-based industrial data catalog solution centers on the AWSAWS Glue Data Catalog, which is a fully-managed metadata repository. The AWS Glue Data Catalog stores technical metadata for datasets (for example, schemas, locations, and data formats) and acts as an index for data in your data lake and warehouses. It is serverless and integrated with many AWS analytics services (like Amazon Athena, Amazon Redshift, and Amazon EMR), enabling those tools to discover and query data using a common catalog. Key AWS components include:

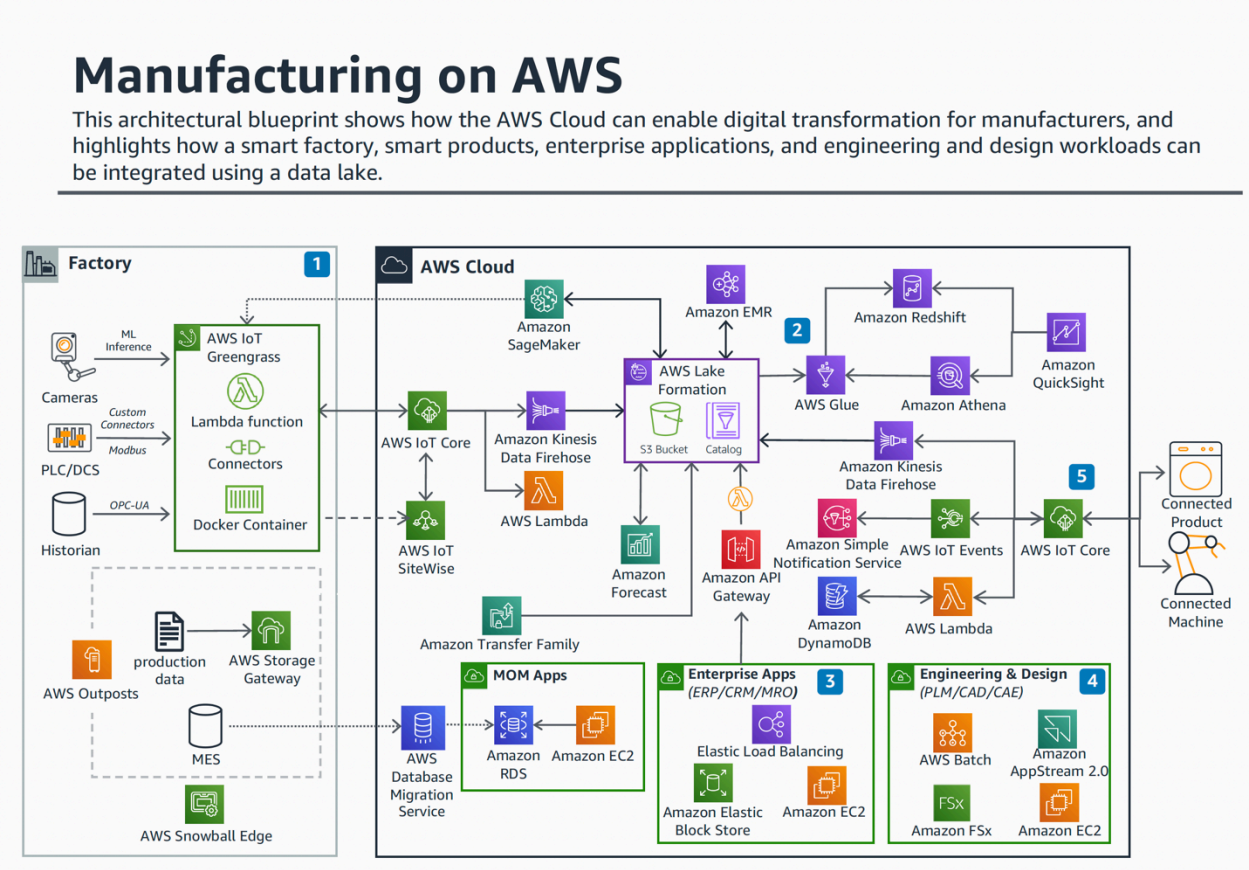
- **AWS AWS Glue Data Catalog and crawlers:** Core service to store dataset metadata and automatically discover data schemas. Glue crawlers can scan data stored in Amazon S3 (or other sources) and update the catalog with new tables and partitions. This is crucial for industrial data lakes, where new sensor files or log batches land frequently. The cataloged metadata allows tools like Amazon Athena to query raw IoT data in Amazon S3 using standard SQL.
- **AWS Lake Formation:** Builds on the Glue catalog to add data lake governance and fine-grained security. Lake Formation helps establish a secure data lake on Amazon S3 with controlled access policies. It allows defining who (which roles or IAM principals) can access specific databases, tables, or even columns and rows. This is important on the shop floor where certain sensitive process data or quality information may be restricted. On the backend, Lake Formation uses the AWS Glue Data Catalog to store metadata and security tags. With Lake Formation, the industrial data catalog can enforce data governance (for example, only authorized analytics users can see quality test results, or PII data from production logs can be masked).
- **Amazon DataZone (business data catalog):** Amazon DataZone provides a higher-level data catalog and portal for data discovery across the enterprise. You can use DataZone to publish data assets (from AWS or on-premises sources) to a business data catalog accessible with a web portal. This adds a business-friendly layer on top of the technical Glue catalog: data producers can enrich assets with business context (like descriptions, owners, and quality scores), and consumers can search and request access through DataZone. In manufacturing, DataZone can help catalog datasets like assembly line sensor readings or KPI by shift. Notably, DataZone integrates natively with the AWS AWS Glue Data Catalog and Lake Formation for metadata and access control.

Other relevant AWS services include Amazon S3 (the storage layer for data lakes), AWS Identity and Access Management (IAM) for security, and tools like Amazon SageMaker AI for data classification (for example, detecting sensitive data) which can complement your catalog governance. AWS Glue also provides a DataBrew tool for data profiling and Glue Data Quality to measure data quality, which can feed information into your catalog (for example, DataZone can display Glue Data Quality metrics for each data asset).

Integrating industrial data sources into the catalog

To implement an industrial data catalog, we must ingest and catalog data from diverse OT and IT sources. AWS offers scalable ingestion services to collect factory data into a central data lake, where it can be cataloged. The following diagram illustrates a reference architecture for an

Industry 4.0 data environment on AWS, highlighting how data flows from edge devices to the cloud data lake and into the AWS AWS Glue Data Catalog:



ADD FIGURE CAPTION HERE

Data ingestion from the factory floor can be achieved using multiple services, depending on data type and latency requirements:

- **Real-time IoT streams**: Use AWS IoT services at the edge and cloud to collect IIoT telemetry. For example, deploy AWS IoT Greengrass on plant floor gateways to preprocess data, then send sensor streams through AWS IoT Core. From IoT Core, you can route data to Amazon Data Firehose, which reliably buffers and delivers the streaming data into Amazon S3 (your data lake) in near real-time. This path is ideal for high-volume telemetry (like vibration or temperature) from PLCs or CNC machines that needs to be stored for analysis.
- **Industrial protocols and asset data**: AWS IoT SiteWise can be used to model industrial assets and collect time-series data from historians or OPC UA servers. SiteWise computes metrics and can stream data to the cloud. While SiteWise itself stores asset data in a managed time-series database, you can configure SiteWise to export data to your S3 data lake for long-term retention or advanced analytics. In such cases, the AWS Glue Data Catalog can register the exported

SiteWise data (using a crawler or ETL job) so that it's queryable alongside other datasets. SiteWise's asset metadata (like equipment hierarchy or attributes) can also be extracted through scripts and stored as JSON or CSV in an S3 bucket. A Glue crawler then creates tables for this metadata, making it queryable through Amazon Athena. This enriches your catalog by linking each asset's sensor data with its contextual info (like location or type).

- **Batch and file data**: Many manufacturing systems output files (like CSV reports or log dumps) on a schedule. AWS Transfer Family (SFTP) or AWS Storage Gateway can ingest these files into Amazon S3. For example, a historian might dump a daily CSV of batch results to an S3 bucket through Transfer Family. A Glue crawler can be scheduled to run after file arrival, updating the catalog with a new table partition (for example, DailyBatchResults for that date). This process tracks file-based datasets in the catalog even when they are created on the fly.
- **Enterprise application data**: Data from MES, ERP, or quality systems (often stored in relational databases) can be ingested using AWS Database Migration Service (DMS) to continually replicate tables into Amazon S3 or Amazon RDS in the cloud. Another method is to expose APIs and use Amazon API Gateway with AWS Lambda to pull data from on-premises systems into the cloud. Once these datasets are in Amazon S3 or a cloud database, they can be cataloged. For example, Glue can crawl the Amazon S3 data or directly catalog the Amazon RDS schema through a Java database connectivity (JDBC) connection.

All ingested data lands in the Amazon S3 data lake, which is the central storage tier. At this point, the AWS AWS Glue Data Catalog performs metadata ingestion. It's a best practice to automate schema discovery for new data. For instance, configure AWS Glue Crawlers to run on the raw-data S3 buckets. As new IIoT data files arrive (for example, hourly sensor JSON from Firehose or daily CSVs from SCADA dumps), the crawler will infer their schema (data types and columns), register the data, and update corresponding tables in the AWS Glue Data Catalog. Each table in the catalog represents a dataset (for example, a table for PressMachine_Telemetry linked to the JSON files in s3://data-lake/press1/ path). The catalog captures details like the S3 location, file format (like JSON or Parquet), and schema and abstracts these implementation details for users.

Now consumers can discover and query this data uniformly. For example, an engineer can use Amazon Athena to run SQL queries on raw sensor data in S3 through the Glue Catalog (Athena automatically uses the Glue Catalog's metadata to parse the data). Data scientists can find a curated historical sensor readings dataset in the catalog and load it into a Jupyter notebook for ML modeling. If using Amazon Redshift, the Redshift Spectrum feature can directly query the S3 data using the Glue Catalog as well. In effect, the AWS Glue Data Catalog becomes the system of record for data schemas across your industrial data estate.

Organize the AWS Glue Data Catalog with a clear database and table naming convention for your industrial datasets. For instance, you might have an AWS Glue database called factory_production with tables like press1_telemetry, press1_maintenance_logs, and assembly_line_yield. This logical grouping makes it straightforward for users to navigate the catalog. Consider tagging catalog entries with business-relevant tags (AWS Glue supports tagging datasets) such as Line=Assembly4 or Criticality=High, as these tags can be used later for search or governance.

Metadata management and governance

Implementing a data catalog is a key part of capturing technical metadata, managing business metadata and enforcing governance rules. AWS's catalog and governance tools help verify that the industrial data is understandable and accessible without compromising security or compliance.

From a metadata perspective, the AWS Glue Data Catalog primarily contains technical metadata (like database and table names, schemas, data location, size, and last update). You can augment this with business metadata by using AWS Glue Schema Registry or Data Catalog tags or by publishing the data to Amazon DataZone with rich descriptions. According to AWS, a data governance catalog revolves around metadata categorized as technical and business, the technical details (for example, file format and schema), and the business context (for example, data classification, owner, and retention policy).

Make sure to capture both technical and business details. For example, document that the press1_telemetry table contains vibration sensor readings from press machine A sampled every minute, and link it to an asset ID or digital twin ID. This context helps end users trust and correctly use the data.

Industrial data often benefits from standardized taxonomies. For example, use asset hierarchy names in your catalog or apply ISO 14224 or ISA-95 naming conventions for equipment and process data so that OT experts can align catalog entries with real-world devices.

For governance, AWS Lake Formation is the primary service to configure. When setting up your data lake, you'll register your S3 buckets with Lake Formation and define data permissions at a fine grain.

For instance, you could allow the DataScienceTeam to query the factory_production. press1_telemetry table but deny access to press1_maintenance_logs if it contains sensitive maintenance records. Lake Formation will enforce these rules for access through Amazon Athena, Amazon Redshift, or AWS Glue. It even supports column-level security and row-level filters, so you might mask a column operator_name or restrict rows where defect_flag = true to certain users.

This granular security is crucial in manufacturing environments that have trade secrets or privacy concerns in the data. For example, an OEM collecting operational data from customer sites can verify that each customer's analysts only see their own machines' data. AWS's data lake approach allows data retrieved from the Data Lake to be restricted at both the column-level and row-level, providing a granular approach to security.

Another aspect of governance is data quality and lifecycle. It's wise to store data quality metrics as part of your metadata. AWS Glue Data Quality can compute scores (for example, the percentage of missing values or anomalies in a dataset), and these can be stored or surfaced in Amazon DataZone. You could also maintain a golden dataset designation in the catalog. For example, if multiple sources provide production counts, mark one table as the authoritative source. In terms of lifecycle, S3 Object Lifecycle policies can tier older data to cheaper storage (Glacier), and the Glue Catalog will still retain the metadata (you might mark such tables with a tag like Archived=true). You can help users determine if data is current or historical by verifying that the catalog reflects data freshness and retention (through attributes or naming).

Finally, consider auditing and lineage. AWS Glue itself doesn't automatically store detailed data lineage, but you can use AWS Glue jobs or AWS Step Functions workflows to track processing pipelines. Storing information about data provenance (for example, that table daily_yield is derived using the ETL from press1_telemetry and press1_quality) can be done by adding descriptions in the catalog or using third-party tools. Amazon DataZone features for tracking data usage and lineage across the catalog, like Purview's lineage functionality. AWS CloudTrail logs and Lake Formation audit logs will record access events, which is important for compliance (you can see who accessed what data when).

Best practices

Building an industrial data catalog on AWS involves stitching together data ingestion, storage, cataloging, and governance in a seamless way. By using AWS AWS Glue Data Catalog with Lake Formation, you create a central source of truth for the manufacturing data assets, from raw sensor feeds to processed analytics datasets. This catalog underpins the larger Industry 4.0 architecture: it feeds AI/ML models, drives live digital twin visualizations, and powers analytical dashboards, all while enforcing security and data quality.

To successfully implement this in practice, follow these best practices:

1. **Start with a strong data lake foundation:** Establish an Amazon S3 data lake and enable AWS Lake Formation. Ingest data from OT sources (like AWS IoT Core and SiteWise) and IT

sources (like databases and files) into S3 in as raw a form as reasonable. Catalog from a single repository.

2. **Automate data cataloging:** Configure AWS Glue Crawlers or scheduled ETL jobs to continuously update the AWS Glue Data Catalog as new data arrives. Maintain logical databases (by plant or domain) and clear naming for tables. Verify that each critical data source has an entry in the catalog (no hidden datasets).
3. **Enrich metadata:** Don't treat the catalog as a passive registry, and actively add documentation. Use the data catalog's description field for tables and columns to document meaning (you can do this in the AWS Glue Console). Add tags to capture business metadata. If you use Amazon DataZone, curate the business catalog so that each dataset has an owner, description, and optional links to business glossary terms or data quality scores.
4. **Implement fine-grained security early:** Use Lake Formation to set up data access rules for various roles (like engineering, operations, data science, and third-party partners). Test these permissions with Athena to block unauthorized access. Also consider integrating your corporate identity and access management and single sign-on with Lake Formation (for example, through AWS IAM Identity Center) so that access can map to real-world teams. This helps prevent data mishaps and builds trust in the system.
5. **Use your catalog in your analytics:** Train your teams and build your tools to use the AWS Glue Data Catalog when accessing data. Whether it's an SQL query, a Python ETL script, or an ML pipeline, use the unified catalog references instead of one-off S3 paths. This not only simplifies code (since the schema is stored centrally) but also improves compliance with governance. If a dataset is removed or access revoked, it's handled through the catalog.
6. **Monitor and evolve:** Treat the data catalog as a living asset. Use Amazon CloudWatch or AWS CloudTrail logs to monitor Glue crawler runs and catalog changes. Implement alerts if a crawler fails or if a critical schema changes. Periodically review the catalog with domain experts to deprecate stale datasets and identify gaps (for example, if there is a new sensor that isn't yet cataloged). Also, as AWS releases new features (like automated lineage in DataZone or improved crawlers), integrate those to continuously improve your industrial data catalog.

By adhering to these practices, an industrial enterprise can create a robust data catalog that aligns IT and OT data management with Industry 4.0 principles. The result is greater visibility into data, improved governance, and faster insights, all of which compel data-driven manufacturing excellence.

References:

- Forrester, 'The Forrester Wave™: Machine Learning Data Catalogs, Q4 2022.
- ISO, 'ISO 8000: Data Quality' and 'ISO 10303: Industrial Automation Systems and Integration—Product Data Representation and Exchange.
- Getting Started with the Industrial Data Platform on AWS

For more information, see Manufacturing Reference Architecture.

MFGSCE3: Industrial data mesh

Industrial data mesh is an architectural concept that provides decentralized ownership and governed sharing of data across the disparate data sources. Data meshes are natively supported by multiple AWS Services. For large-scale organizations, using multiple AWS accounts and sharing data across them is a common approach.

AWS natively supports the data mesh pattern today:

- AWS Lake Formation supports cross account permissions and resource shares for data registered in the AWS Glue and Lake Formation catalog.
- Amazon Redshift supports cross account data sharing.
- AWS analytics services, including Amazon Athena, Glue, and Amazon EMR, natively support Lake Formation permissions that implement data mesh concepts of producers, consumers, shares, and central catalog.
- Amazon DataZone is a data management service that helps you catalog, discover, share, and govern data stored across AWS, on-premise, and third-party sources. With Amazon DataZone, administrators and data stewards who oversee an organization's data assets can manage and govern access to data using fine-grained controls.
- AWS account services like AWS Control Tower and AWS Organizations have service control policies (SCPs), which allow for decentralized governance of data and analytics accounts.
- The data lakehouse architecture is ideally suited to help teams build data domains, and you can use the data mesh approach to bring domains together to enable data sharing and federation across business units. This approach can improve autonomy and speed up innovation while building on top of a proven and well-understood architecture and technology stack, maintaining high standards for data security and governance.

The following are key points when considering a data mesh design:

- Data mesh is a pattern for defining how organizations can organize around data domains with a focus on delivering data as a product. However, it may not be the right pattern for every customer.
- A lakehouse approach and the data lake architecture provide technical guidance and solutions for building a modern data system on AWS.
- The lakehouse approach with a foundational data lake serves as a repeatable blueprint for implementing data domains and products in a scalable way.
- The way you use AWS analytics services in a data mesh pattern may change over time but remains consistent with the technological recommendations and best practices for each service.

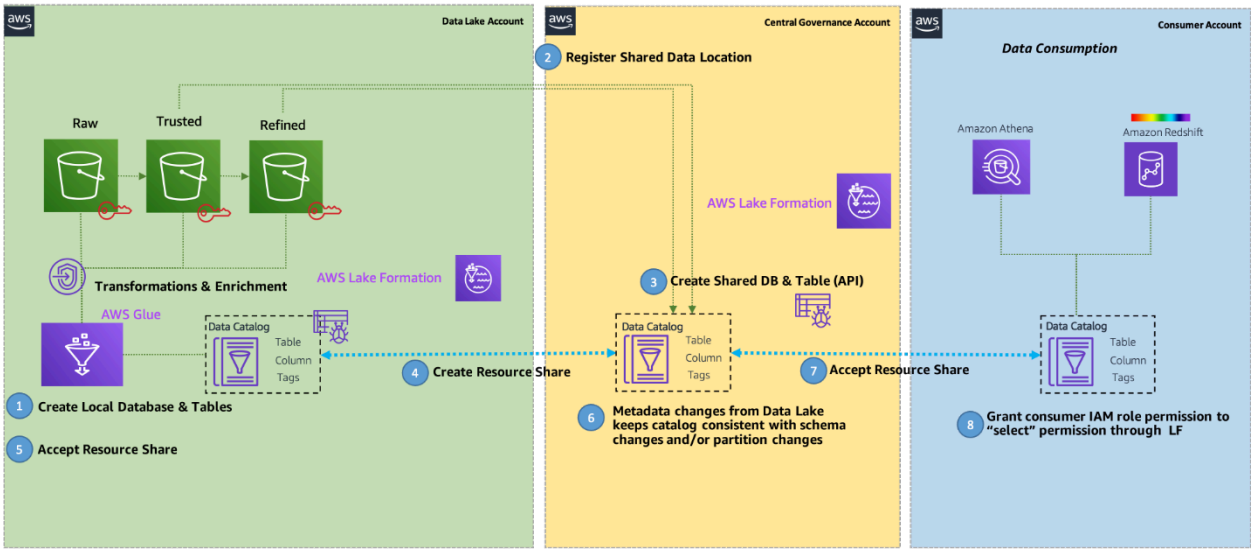
Data mesh design goals

- **Data as a product**: Each organizational domain owns their data entirely. They're responsible for building, operating, serving, and resolving issues arising from the use of their data. Data accuracy and accountability lies with the data owner within the domain.
- **Federated data governance**: Data governance verifies that data is secure, accurate, and not misused. The technical implementation of data governance such as collecting lineage, validating data quality, encrypting data at rest and in transit, and enforcing appropriate access controls can be managed by each of the data domains. However, central data discovery, reporting, and auditing is required to make it simple for users to find data and for auditors to verify compliance.
- **Common access**: Data must be consumable by subject matter personas like data analysts and data scientists, as well as purpose-built analytics and machine learning (ML) services like Amazon Athena, Amazon Redshift, and Amazon SageMaker AI AI. To do that, data domains must expose a set of interfaces that make data consumable while enforcing appropriate access controls and audit tracking.

Data mesh end to end workflow

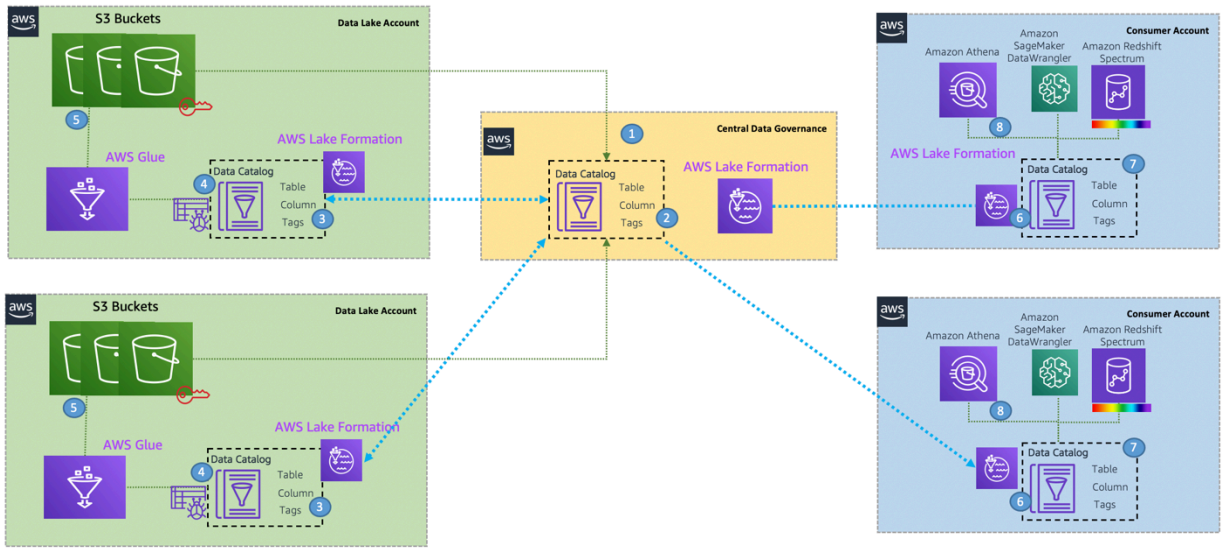
A data mesh design organizes around data domains. Each data domain owns and operates multiple data products with its own data and technology stack, which is independent from others. Data domains can be purely producers, such as a finance domain that only produces sales and revenue data for domains to consumers, or a consumer domain, such as a product recommendation service that consumes data from other domains to create the product recommendations displayed on an ecommerce website. In addition to sharing, a centralized data catalog can provide users with the

ability to more quickly find available datasets and allows data owners to assign access permissions and audit usage across business units.



ADD FIGURE CAPTION HERE

With this design, you can connect multiple data lakehouses to a centralized governance account that stores the metadata from each environment. The strength of this approach is that it integrates the metadata and stores it in one meta model schema that can be accessed through AWS services for various consumers. You can extend this architecture to register new data lake catalogs and share resources across consumer accounts. The following diagram illustrates a cross-account data mesh architecture.



ADD FIGURE CAPTION HERE

Access patterns

Each consumer obtains access to shared resources from the central governance account in the form of resource links. These are available in the consumer's local Lake Formation and AWS AWS Glue Data Catalog, allowing database and table access that can be managed by consumer admins. After access is granted, consumers can access the account and perform different actions with the following services:

- Amazon Athena acts as a consumer and runs queries on data registered using AWS Lake Formation. AWS Lake Formation verifies that the workgroup AWS Identity and Access Management (IAM) role principal has the appropriate Lake Formation permissions to the database, table, and Amazon S3 location as appropriate for the query. If the principal has access, AWS Lake Formation vends temporary credentials to Amazon Athena, and the query runs. Authentication is granted through IAM roles or users, or web federated identities using SAML or OIDC. For more information, see How Athena accesses data registered with Lake Formation.
- Amazon SageMaker AI Data Wrangler helps you quickly select data from multiple data sources, such as Amazon S3, Amazon Athena, Amazon Redshift, AWS Lake Formation, and Amazon SageMaker AI Feature Store. You can also write queries for data sources and import data directly into Amazon SageMaker AI AI from various file formats, such as CSV files, Parquet files, and database tables. Authentication is granted through IAM roles in the consumer account. For more information, see Prepare ML Data with Amazon SageMaker AI Data Wrangler.
- You can use Amazon Redshift Spectrum to register external schemas from AWS Lake Formation and provide a hierarchy of permissions to control access to Amazon Redshift databases and tables in a data catalog. If the consumer principal has access, AWS Lake Formation vends temporary credentials to Redshift Spectrum tables, and the query runs. Authentication is granted through IAM roles or users or web federated identities using SAML or OIDC. For more information, see Redshift Spectrum and AWS Lake Formation.
- Quick with Amazon Athena integrates with AWS Lake Formation permissions. If you're querying data with Amazon Athena, you can use AWS Lake Formation to simplify how you secure and connect to your data from Quick. AWS Lake Formation adds to the IAM permissions model by providing its own permissions model that is applied to AWS analytics and ML services. Authentication is granted through IAM roles that are mapped to Quick user permissions. For more information, see Authorizing connections through AWS Lake Formation.
- Amazon EMR Studio and EMR notebooks allow running Spark SQL against AWS Lake Formation's tables backed by a SAML authority. Beginning with Amazon EMR 5.31.0, you can launch a cluster that integrates with AWS Lake Formation. Authentication is granted through IAM roles or users

or web federated identities using SAML or OIDC. For more information, see Integrate Amazon EMR with AWS Lake Formation.

Customers can benefit from both data mesh and the lakehouse (modern data architecture), as they address different aspects of data management:

- All customers at a certain level of complexity already build multi-account data architectures.
- Building data lakes on S3 with purpose-built data and analytics services for processing is still the best architecture practice that customers should use to implement a data domain in a data mesh architecture.
- Data mesh is an advanced architecture which is appropriate for those customers who already have multiple data lakes or lakehouses. If you are just getting started with data lakes on AWS, the modern data architecture on AWS is the right way to build a data lake on S3 and integrate it with Redshift (data warehouse). AWS' purpose-built data store, AWS Lake Formation provides unified governance and seamless data access when you have multiple accounts. Data mesh and lakehouse architectures are complementary, and customers should start with a lakehouse and over time adopt a data mesh architecture if required.

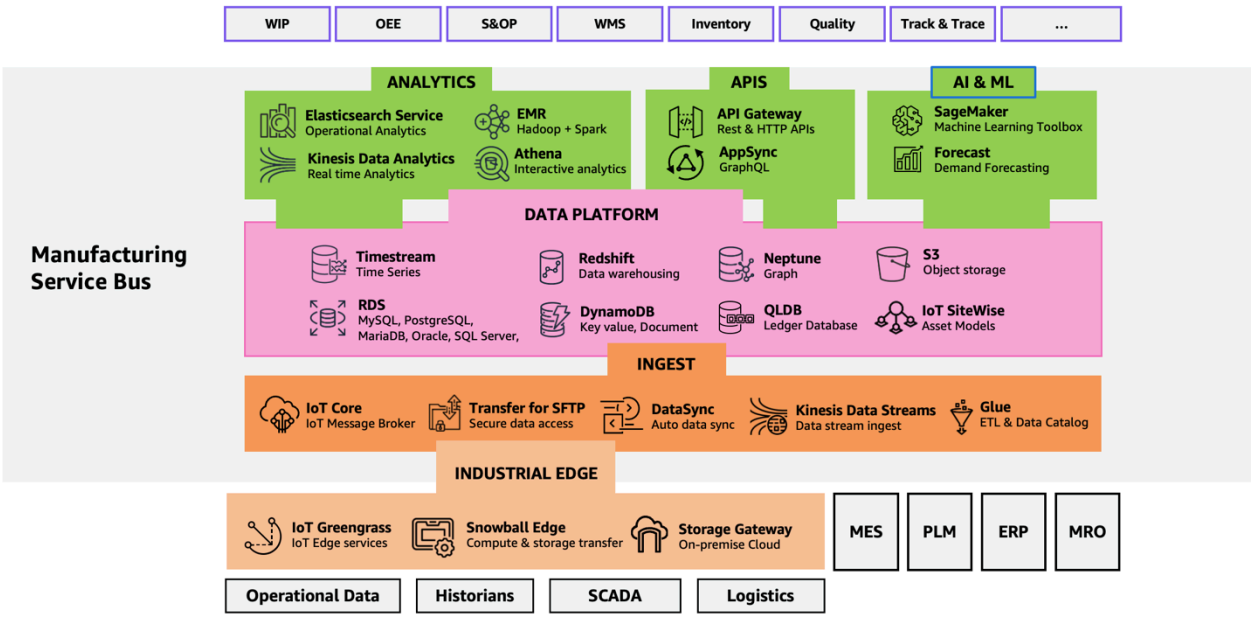
Industrial service bus

Manufacturing organizations are increasingly using industrial service buses (ISBs) to enable seamless integration, data exchange, and interoperability across diverse systems, applications, and devices in both OT and IT environments.

An ISB functions as a middleware layer that standardizes communication protocols such as OPC UA, MQTT and AMQP, providing a unified messaging backbone for data flow between sensors, control systems, enterprise applications, and cloud environments. This standardization is essential for real-time monitoring, control, and optimization of industrial processes, as it allows diverse equipment and software from multiple vendors to interoperate efficiently.

General industry solutions for ISBs also involve edge computing frameworks that extend processing capabilities closer to the source of data, reducing latency and minimizing bandwidth requirements for time-sensitive applications. For example, deploying lightweight edge gateways that support ISB protocols enables local data aggregation, filtering, and preprocessing before sending selected data to the cloud for further analysis.

This approach allows manufacturers to perform predictive maintenance, enhance operational efficiency, and improve decision-making by using real-time data analytics and machine learning. By integrating ISBs with both on-premises and cloud environments, manufacturers can achieve scalable, flexible, and secure data integration, paving the way for digital transformation initiatives such as smart factories and the Industry 4.0.



ADD FIGURE CAPTION HERE

As an example, AWS IoT Core serves as a cloud-based ISB, supporting protocols like MQTT, HTTPS, and WebSockets to enable secure, low-latency, and reliable bi-directional communication between industrial equipment and cloud services.

This architecture is further enhanced by AWS IoT SiteWise, which uses OPC UA to collect, structure, and analyze data from multiple sensors and industrial assets in near real-time, providing a unified data model for analytics and machine learning workloads.

This edge computing capability reduces latency, optimizes bandwidth usage, and verifies that critical operations can continue uninterrupted in case of network disruptions.

Resources

- Design a data mesh architecture using AWS Lake Formation and AWS Glue
- AWS Data Mesh Helper Library

MFGSCE4: Digital thread: contextualization and knowledge graph

In most manufacturing organizations, the product lifecycle data is typically stored in enterprise systems such as Product Lifecycle Management (PLM), Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES) and Customer Relationship Management (CRM). Due to the disconnected nature of enterprise infrastructure, the generated data is often fragmented and unused.

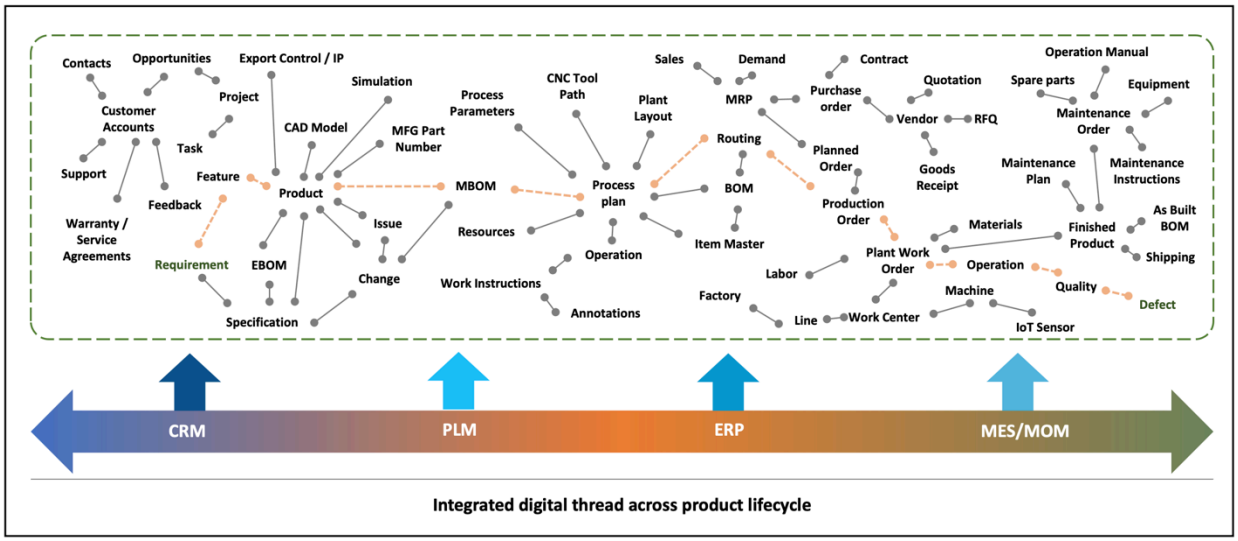
Managing and contextualizing data across the product lifecycle is challenging due to variations in data usage depending on the specific stakeholder interacting with it. Manufacturing organizations continuously aim to address the connected enterprise outcomes through various strategic initiatives. One notable initiative is the digital thread for connected enterprise.

Digital thread refers to the seamless flow of data across the entire product lifecycle, from requirements, design, production, and eventually to service and disposal. This data-driven approach enables manufacturers to contextualize information and gain insights for better decision-making.

A comprehensive digital thread should incorporate the requirements of various stakeholders, offering enhanced contextualization and knowledge integration capabilities, both structured and unstructured, to unlock the potential of manufacturing data.

Knowledge graphs provide a great way to represent and connect data from various sources, contextualizing and improving the understanding of complex relationships between entities. They provide a structured way to organize and link data entities, making it simple to find and traverse relationships between information from various sources.

Contextualization is the process of adding relevant context to data, which can help derive meaningful insights. By contextualizing data, manufacturers can better understand the context and conditions surrounding specific events or data points, which better informs their decision-making.



ADD FIGURE CAPTION HERE

A well-architected manufacturing digital thread with contextualization and knowledge graph capabilities should support the following characteristics:

1. **Data integration and ingestion:** The ability to ingest and integrate data from various sources, such as requirements, design, planning, supply chain, manufacturing execution data along with the unstructured data, is crucial for building a comprehensive digital thread.
2. **Contextualization:** Incorporating contextualization enables manufacturers to enrich their data with meaningful context and uncover hidden relationships and patterns.
3. **Scalability and availability:** The digital thread should be built on a scalable and highly available infrastructure, capable of handling variable data volumes and workloads without compromising performance or reliability.

By contextualizing in the manufacturing digital thread, manufacturers can better understand their data, gain deeper insights, and drive continuous improvement across the product lifecycle. This data-driven approach improves decision making, increases efficiency, enhances product quality, and helps you innovate within your organization.

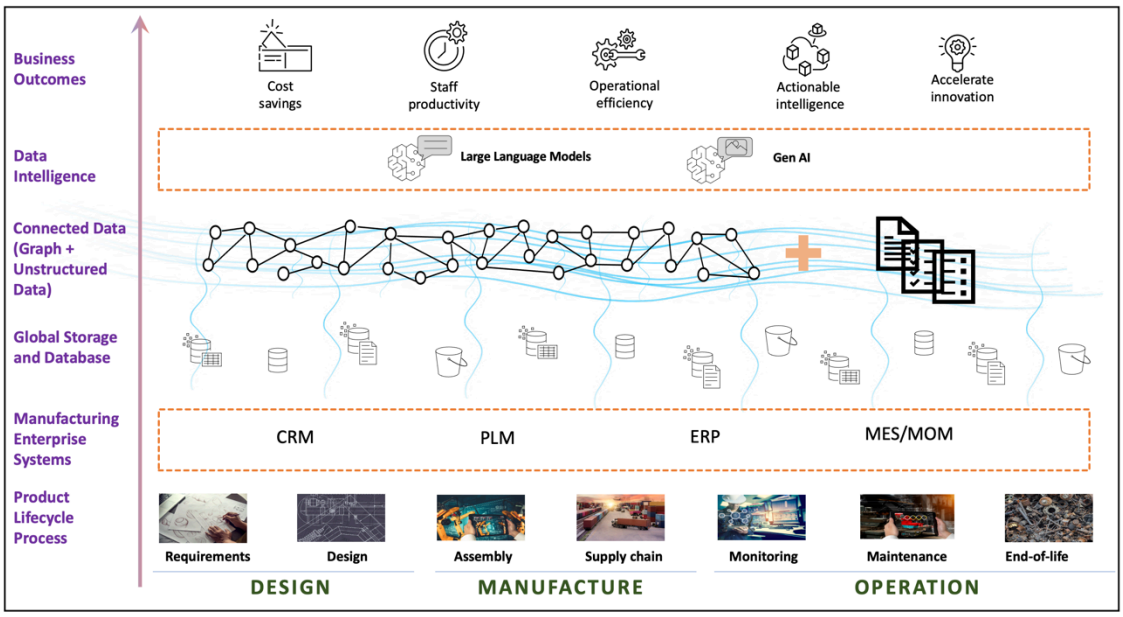
Digital thread solution framework

Data from various product lifecycle processes forms the foundation of this digital thread solution framework. The subsequent layer encompasses core enterprise systems, including PLM, ERP, and

MES, which manage specific aspects such as people, processes, engineering, and manufacturing data within the enterprise.

The next layer is the connected data, which involves both knowledge graph and unstructured data. Together, they uncover insights and provide a comprehensive understanding of the interconnected manufacturing enterprise data.

Finally, large language models are integrated with the knowledge graph and unstructured data, creating advanced queries and accessing natural language capabilities.

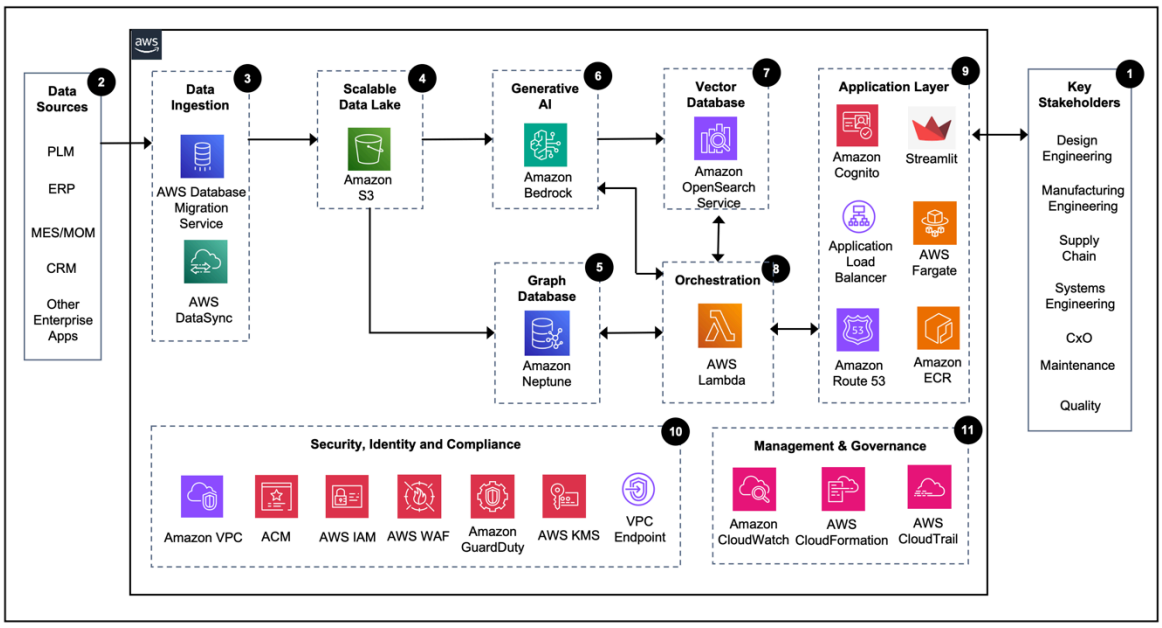


ADD FIGURE CAPTION HERE

The solution aims to accelerate innovation by seamlessly connecting data from various systems and generate insights through the manufacturing digital thread framework. This framework establishes an intelligent structure where data is interconnected, empowering manufacturing stakeholders in their decision-making processes.

Digital thread reference architecture

The manufacturing digital thread solution architecture is implemented through the strategic integration of Amazon Neptune graph database, Amazon OpenSearch Service, and Amazon Bedrock, a fully managed generative AI service. The components are further enhanced by various AWS services, creating a comprehensive solution for the digital thread.



ADD FIGURE CAPTION HERE

1. Identify key stakeholders in the manufacturing organization and understand the business needs.
2. Identify data sources to build a digital thread on AWS
3. Ingest data into AWS using AWS Data Migration Service for database migrations and AWS DataSync for large dataset movement.
4. Upload ingested data into Amazon S3 for secure storage for further processing and analysis.
5. Use Amazon Neptune bulk loader capability to ingest the data from Amazon S3 to Amazon Neptune graph database.
6. Create chunks from unstructured documents stored in Amazon S3, embed using Amazon Titan Text Embeddings V2, and store them in Amazon OpenSearch Service vector database.
7. Select foundation models in Amazon Bedrock.
8. Link Amazon Bedrock, Amazon Neptune, and Amazon OpenSearch Service, and integrate with AWS Lambda and Langchain. The orchestrator coordinates the process of generating the opencypher query using Amazon Bedrock Foundation models, runs the query against the Amazon Neptune graph, converts the graph query results into natural language using Amazon Bedrock , extracts the unstructured information from Amazon OpenSearch Service, and returns the relevant message to the user.
9. Create an application layer using AWS Fargate for container orchestration, Amazon Elastic Container Registry for managing container images, Elastic Load Balancing for efficient traffic distribution, Amazon Route 53 for DNS, and Amazon Cognito for authentication.

10. Use Amazon VPC to operate the application in a secure and isolated network. AWS Identity and Access Management enhances access control, while AWS Certificate Manager manages certificates and AWS WAF provides web application security. Malicious activity is constantly monitored by Amazon GuardDuty. The data at rest is encrypted with AWS Key Management Service and can be integrated with other third-party KMS solutions.
11. Use AWS CloudTrail to enhance transparency by tracking activities, Amazon CloudWatch to monitor resources, and AWS CloudFormation for automated resource deployment of digital thread application.

MFGSCE5: Computer vision for automated quality inspection

In manufacturing quality control, the transition from manual inspection to automated computer vision systems represents a significant advancement. However, the journey from raw image capture to reliable defect detection is complex, particularly from a data management and ML operations perspective.

This section explores the data challenges and AI/ML pipeline requirements for implementing effective computer vision-based quality inspection solutions.

Data acquisition and management challenges

Image capture and local storage

Manufacturing environments present unique challenges for image data acquisition:

- **Environmental variability**: Shop floor lighting conditions fluctuate throughout the day, affecting image consistency.
- **Camera configuration**: Selecting appropriate sensors (color or monochrome), lenses, and positioning to capture defects consistently requires careful optimization.
- **Real-time storage requirements**: High-resolution images from production lines create significant local storage demands, requiring efficient buffering and compression strategies.
- **Metadata integration**: Correlating images with production metadata (like timestamps, batch numbers, and machine parameters) is essential for contextual analysis.

Secure cloud transfer

Moving image data from operational technology (OT) environments to cloud systems involves:

- **Bandwidth optimization**: Manufacturing facilities often face network constraints, requiring intelligent transfer strategies like edge preprocessing, compression, or selective sampling.
- **Security protocols**: Implementing encryption and secure transfer mechanisms while maintaining OT and IT separation.
- **Data integrity**: Keeping images and associated metadata consistent during transfer processes.

Data preparation pipeline

Image segmentation

Breaking down complex manufacturing images requires:

- **Region-of-interest extraction**: Isolating specific product areas for focused analysis.
- **Background elimination**: Removing irrelevant visual elements that could confuse ML models.
- **Multi-object handling**: Managing images containing multiple instances of products or components.

Defect labeling

Creating quality training datasets involves:

- **Annotation complexity**: Defining consistent protocols for marking subtle manufacturing defects.
- **Expert knowledge transfer**: Converting tacit quality inspector knowledge into explicit annotation guidelines.
- **Labeling efficiency**: Implementing semi-supervised approaches to reduce manual labeling burden.
- **Data augmentation**: Using generative AI to create synthetic examples of rare defect types.

Training data management

Organizing image assets for optimal ML development:

- **Version control**: Maintaining trackable iterations of image datasets as production evolves.
- **Class balancing**: Managing the inherent imbalance between good products and defective ones.
- **Representativeness**: Verifying that the dataset covers seasonal variations and all possible defect types.

- **Cross-functional access**: Enabling secure data sharing between OT personnel and data science teams.

MLOps for manufacturing vision systems

Model development

Transforming manufacturing image data into reliable quality models:

- **Transfer learning optimization**: Adapting pre-trained computer vision architectures to specific manufacturing contexts.
- **Custom architecture development**: Creating specialized models for unique inspection requirements.
- **Explainability integration**: Building interpretable models that help operators understand defect classifications.
- **Lightweight model design**: Optimizing performance for edge deployment constraints.

Model packaging and deployment

Preparing models for production environments:

- **Edge compilation**: Converting trained models to formats optimized for edge hardware.
- **Component packaging**: Bundling models with necessary preprocessing and postprocessing code.
- **Deployment automation**: Creating CI/CD pipelines that can safely update models in production.
- **Version management**: Maintaining trackability of model versions across the manufacturing fleet.

Edge inference and monitoring

Running models in production environments:

- **Latency optimization**: Verifying that real-time performance meets production line speeds.
- **Resource utilization**: Balancing model complexity against available edge computing resources.
- **Quality metrics tracking**: Monitoring inference performance against established standards.
- **Integration with control systems**: Connecting inference outputs to downstream manufacturing systems.

Model drift detection

Maintaining model reliability over time:

- **Data distribution monitoring**: Detecting when production images diverge from training distributions.
- **Performance degradation alerts**: Identifying when model accuracy falls below acceptable thresholds.
- **Root cause analysis**: Determining whether drift stems from process changes, lighting variations, or other factors.
- **Targeted data collection**: Efficiently gathering new images of problematic cases for retraining.

Continuous learning cycles

Keeping models current with manufacturing realities:

- **Automated retraining events**: Initiating new training cycles based on drift metrics.
- **Knowledge retention**: Preserving performance on historical defect types while adapting to new ones.
- **A/B testing**: Safely evaluating model improvements before full deployment.
- **Model governance**: Maintaining audit trails of model changes for quality compliance.

AWS architecture for vision-based quality inspection

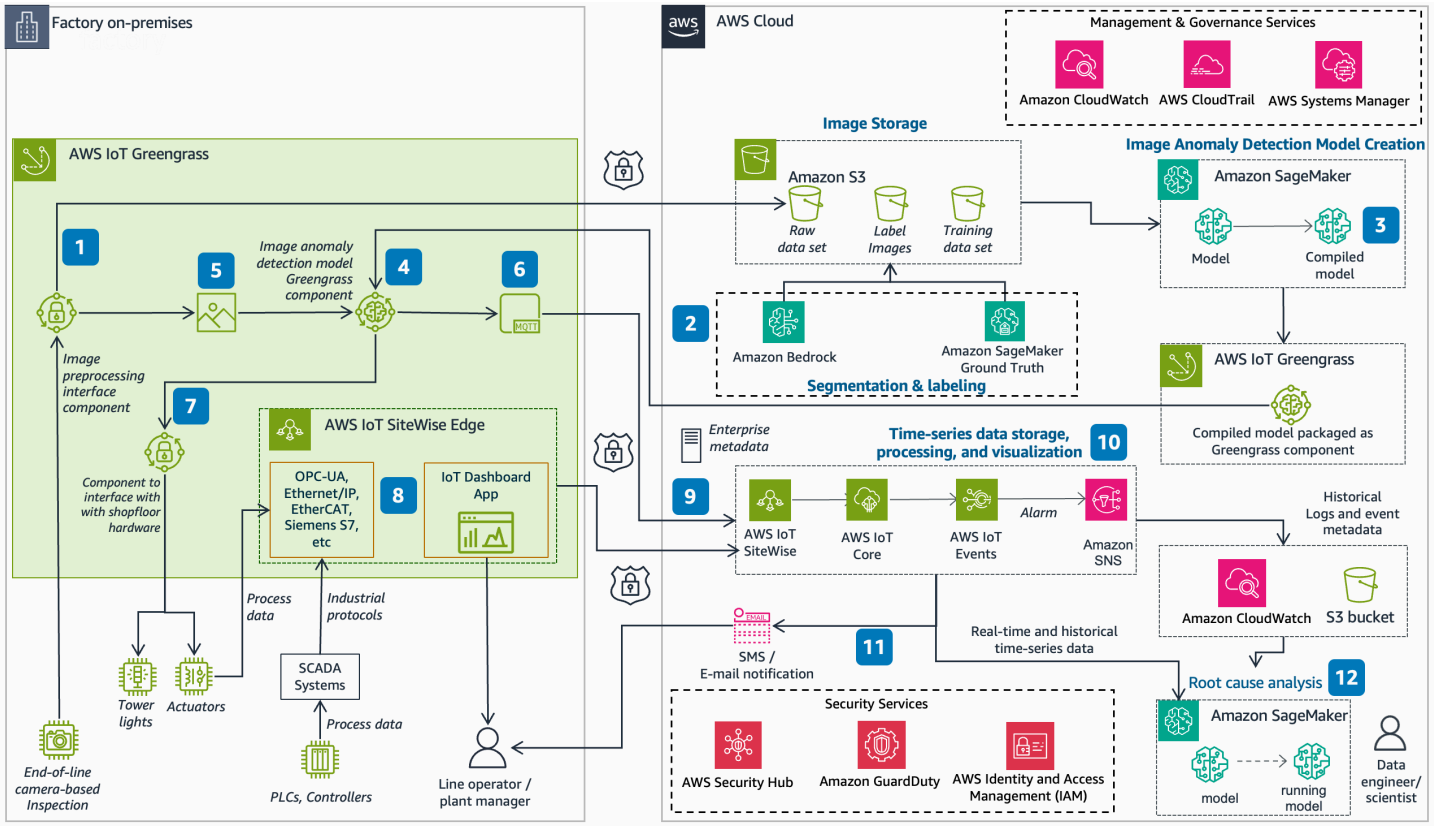
A comprehensive data architecture for computer vision quality inspection on AWS includes:

1. **Image acquisition and edge processing**
   - Use AWS IoT Greengrass to capture and preprocess images at the edge.
   - Implement edge-based inference for immediate quality decisions.
   - Buffer images locally with selective cloud uploading to manage bandwidth.
2. **Secure data transfer and storage**
   - Encrypt and transfer images securely to Amazon S3.
   - Organize raw images in data lake architecture with appropriate partitioning.
   - Implement lifecycle policies for cost-effective long-term storage.

3. **Image processing and labeling**
    - Use Amazon SageMaker Ground Truth for human labeling workflows.
    - Use Amazon Bedrock foundation models for automatic segmentation and labeling assistance.
    - Generate synthetic defect images for rare conditions using generative AI capabilities.
4. **Model development**
    - Use Amazon SageMaker AI AI for training and hyperparameter optimization.
    - Implement transfer learning from AWS computer vision model zoo.
    - Establish experiment tracking for model versioning and comparison.
5. **MLOps pipeline**
    - Create automated CI/CD pipelines for model deployment using AWS CodePipeline.
    - Compile models for edge deployment with SageMaker AI Neo.
    - Package models as AWS IoT Greengrass components.
6. **Production monitoring and feedback**
    - Track inference metrics and model performance using Amazon CloudWatch.
    - Detect data drift with SageMaker AI Model Monitor.
    - Implement automated retraining workflows initiated by performance degradation.
7. **Integration with manufacturing systems**
    - Connect inference results to AWS IoT Core MQTT topics.
    - Initiate control systems and quality workflows based on detection results.
    - Integrate with AWS IoT SiteWise for holistic process monitoring.
    - Calculate quality KPIs and OEE metrics based on vision system outputs.
8. **Analytics and continuous improvement**
    - Store historical inspection data in Amazon S3 for long-term analysis.
    - Use Quick for quality trend visualization.
    - Employ Amazon SageMaker AI for root cause analysis of quality issues.
    - Close the loop between quality detection and upstream process control.

By focusing on the data lifecycle and MLOps for computer vision applications, manufacturers can build sustainable, self-improving quality inspection systems that overcome the traditional challenges of computer vision deployment in industrial environments. This data-centric approach helps keep your models accurate and relevant even as manufacturing processes evolve over time.

Automated quality inspection architecture



ADD FIGURE CAPTION HERE

1. Automatically upload training images from the manufacturing line camera to Amazon Simple Storage Service (Amazon S3).
2. Label the training images and identify defects. You can use Amazon SageMaker Ground Truth for creating and managing labeling jobs. Also, you can use state-of-the-art large language models in Amazon Bedrock to create synthetic image datasets for scenarios that are too expensive, dangerous, or impossible to capture real-world image datasets and for automated segmentation and labeling images.
3. Begin model training using Amazon SageMaker AI, using the training images stored in Amazon S3. You can use AWS-provided image anomaly detection pre-trained models in Amazon SageMaker AI AI to create your own models through transfer learning.
4. Next, the model needs to be compiled and packaged as an AWS IoT Greengrass component.
5. Deploy the trained model from previous step to an Edge device for running production-level inferences using AWS IoT Greengrass.

6. Present camera images to the image anomaly detection model AWS IoT Greengrass component for anomaly detection.
7. Publish the inference results to an AWS IoT Core MQTT topic.
8. Perform automated actions on the machine of concern and notify plant personnel of detected anomalies using the AWS IoT Greengrass component. For example, this component can run a product rejection workflow by activating an industrial robot arm or switching the colors in the tower lights to indicate product quality issues.
9. Ingest process data into the AWS IoT SiteWise Edge gateway running on AWS IoT Greengrass from the machine and equipment using standard industrial protocols like OPC-UA, Modbus TCP/RTU, EtherNet/IP, or BACnet.
10. Push the ingested process data to AWS IoT SiteWise.
11. Compute key performance indicator (KPI) metrics, such as overall equipment effectiveness (OEE), from the process data in AWS IoT SiteWise. Create monitoring and KPI dashboards in IoT Dashboard application or Amazon Managed Grafana for operations users.
12. Create events from plant data and enterprise metadata by routing data to AWS IoT Events through AWS IoT Core, and send email or text notifications to operations users using the Amazon Simple Notification Service (Amazon SNS).
13. Store the historical logs, event metadata and vision inference data streams in Amazon S3. Using data stored in Amazon S3, and real-time and historical time-series process data from AWS IoT SiteWise, train root cause analysis models in Amazon SageMaker AI. Run model inference on Amazon SageMaker AI to identify the root cause of issues.

Resources

- Defect Detection App User Guide
- Using AWS generative AI to improve defect detection in Manufacturing
- Zero-Training Visual Defect Detection

Operational excellence

The operational excellence pillar focuses on the ability to run and evolve reliable and high-performing manufacturing data infrastructure that support critical operational functions while continuously improving over time. It includes practices that help organizations monitor system health, respond to incidents, and optimize processes, all while aligning with business objectives and production demands.

Manufacturing organizations are under constant pressure to maintain operational continuity, minimize downtime, and quickly adapt to shifting customer needs. In addition to common performance and reliability challenges faced by all industries, manufacturing environments must deal with industry-specific constraints such as real-time process monitoring, equipment integration, and highly specialized workflows.

Global operations add further complexity, requiring consistent operational standards across diverse plants and regions. The systems must be designed to detect and respond to issues proactively, support continuous improvement cycles, and provide visibility into key operational metrics. Through strong observability, automation, and feedback loops, manufacturers can drive sustainable improvements and operational excellence across the business.

The operational excellence pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Operational Excellence Pillar whitepaper.

Topics

- Design principles
- Organization
- Prepare
- Operate
- Evolve

Design principles

The operational excellence pillar in the MIDA Lens is concerned with enabling manufacturing organizations to achieve reliable, high-performing, and continuously improving data infrastructure

to support their operational needs. By emphasizing proactive monitoring, automated incident response, continuous optimization, and thorough documentation, MIDA empowers manufacturing teams to rapidly detect, diagnose, and resolve issues that could impact production, quality, or maintenance.

This operations-centric approach verifies that MIDA can adapt to changing business requirements, drive continuous improvement, and provide the responsive, observable data system that modern factories require to maintain a competitive edge. Through the principles of the operational excellence pillar, MIDA lays the foundation for manufacturing organizations to extract maximum value from their data and transform their operations.

The key design principles for operational excellence in MIDA include:

- **Designing for failure modes in industrial environments:** MIDA accounts for common failure scenarios in manufacturing, such as edge device failures, network outages, and equipment malfunctions. For reliable operation despite these disruptions, it incorporates features like redundancy, failover mechanisms, and self-healing capabilities.
- **Operational simplicity:** MIDA prioritizes simplicity in its operations by using self-healing mechanisms, infrastructure as code (IaC), and centralized configuration management. This approach reduces complexity and manual effort in maintaining the data infrastructure, allowing the manufacturing team to focus on higher-value activities.
- **Proactive monitoring and observability:** Comprehensive monitoring and observability, as emphasized in the AWS Well-Architected Framework, are crucial for MIDA. Observability enables the manufacturing team to quickly detect, diagnose, and resolve issues that could affect production, quality, or maintenance.
- **Automated incident response and remediation:** MIDA incorporates automated mechanisms to respond to and remediate incidents, minimizing downtime and improving the reliability of manufacturing processes. Tools such as automated runbooks, playbooks, and self-healing capabilities help maintain the operational health of the data infrastructure.
- **Continuous improvement and optimization:** MIDA is designed with a culture of continuous improvement and optimization in mind. This allows the manufacturing team to regularly review performance metrics, identify bottlenecks, and implement enhancements that improve the overall reliability and efficiency of the data infrastructure.

By incorporating these design principles, the modern industrial data architecture can deliver the operational excellence required to support the ever-evolving needs of modern manufacturing organizations.

Organization

There are no operational excellence best practices for Organization specific to the Manufacturing Industry Lens.

Prepare

MIDAOPS01: How do you assess your industrial data workload against compliance requirements?

Industrial data workloads must comply with relevant regulatory requirements and industry standards. Establishing a systematic approach to compliance assessment helps you continually adhere to relevant standards while maintaining data integrity and security in industrial data infrastructure.

MIDAOPS02: How do you observe the health of your industrial data infrastructure?

Industrial data infrastructure requires comprehensive monitoring across edge devices, networks, and cloud resources for reliable manufacturing operations. Effective observation of infrastructure health is crucial for maintaining data quality, system availability, and operational continuity from the shop floor to the cloud.

Best practices

- MIDAOPS01-BP01 Implement a process to periodically review compliance requirements relevant to your industrial data infrastructure
- MIDAOPS02-BP01 Use enhanced monitoring and alerting in the cloud and at the edge

MIDAOPS01-BP01 Implement a process to periodically review compliance requirements relevant to your industrial data infrastructure

Implement a process to periodically review compliance requirements relevant to your industrial data infrastructure. Regular assessment of compliance requirements is crucial for manufacturing

environments where industrial data is subject to various regulations such as GDPR, HIPAA, ISO 27001, or industry-specific standards. Establish a systematic review process to verify ongoing compliance and adapt to changing requirements.

Desired outcome: Maintain a clear understanding of applicable compliance requirements and verify that your industrial data infrastructure remains compliant through regular assessments and updates. Identify gaps in compliance and implement necessary changes promptly.

Benefits of establishing this best practice: Adopting the best practice and prescriptive guidance helps you align with industry standards while reducing the risk of non-compliance penalties and regulatory issue. This enables proactive rather than reactive compliance management and simplifies auditing processes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Create a comprehensive compliance review framework for regular assessments, documentation, and updates to your industrial data infrastructure.

Implementation steps

- **AWS Config Rule implementation:** Create custom compliance rules in AWS Config to automatically evaluate your resources against industry standards. AWS Config maintains a detailed inventory of your AWS resource configurations and continuously checks for compliance violations.
- **Security Hub CSPM integration:** Enable AWS Security Hub CSPM with industry-specific security standards (CIS, PCI DSS, HIPAA). Security Hub CSPM automatically assesses your security status and aggregates alerts across integrated services.
- **Audit management:** Use AWS Audit Manager to automate evidence collection and continuously audit your AWS usage. Create assessment templates mapped to your compliance frameworks and schedule regular assessments.
- **Automated monitoring:** Configure Amazon EventBridge rules to run Lambda functions for compliance checks. Set up notifications through Amazon SNS when compliance issues are detected.

Key AWS services

- AWS Config

- AWS Security Hub CSPM
- AWS Audit Manager
- Amazon CloudWatch
- Amazon EventBridge
- Amazon Lambda

Resources

Related documents:

- Security Best Practices for Manufacturing OT
- Best practices to respond to security risks across AWS Organizations

MIDAOPS02-BP01 Use enhanced monitoring and alerting in the cloud and at the edge

Comprehensive observability is crucial for manufacturing teams to proactively monitor the health and performance of the data infrastructure.

Desired outcome: Achieve real-time visibility into the entire industrial data infrastructure, enabling proactive issue detection, rapid troubleshooting, and maintenance of optimal system performance across both cloud and edge environments.

Benefits of establishing this best practice: By implementing this robust observability framework, a manufacturing organization can gain visibility and insight needed to proactively manage the reliability, performance, and security of their data-driven operations. The observability tools and practices enable the team to quickly detect, investigate, and resolve issues before they impact production.

- Enables proactive identification of potential issues before they impact operations
- Reduces mean time to detection (MTTD) and resolution (MTTR)
- Provides comprehensive visibility across the entire data infrastructure
- Facilitates data-driven decision making for system optimization
- Enhances overall system reliability and performance

Level of risk exposed if this best practice is not established: High

Implementation guidance

Establish a comprehensive monitoring and alerting framework that covers both cloud and edge components of your industrial data infrastructure.

Implementation steps

1. Use AWS AWS IoT Greengrass to collect and stream operational telemetry data, such as device health metrics and diagnostic logs, back to the cloud. This allows the manufacturing team to maintain visibility into the state of edge components, even in disconnected scenarios.
2. Use Amazon CloudWatch to collect and analyze metrics, logs, and events from across the hybrid edge-cloud architecture. This includes metrics like data ingestion rates, database performance, and resource utilization. CloudWatch dashboards and alarms provide visibility into the overall system health, allowing the team to quickly identify and address issues. Set baseline performance metrics and configure automated alerting thresholds.
3. To enable end-to-end tracing and diagnostics, use AWS X-Ray. X-Ray provides detailed insights into the performance and dependencies of distributed components, helping the team rapidly pinpoint the root cause of problems.
4. Consider recording all API calls and configuration changes using AWS CloudTrail, enabling comprehensive auditing and security monitoring of the data infrastructure.

Key AWS services

- AWS CloudTrail
- Amazon CloudWatch
- AWS AWS IoT Greengrass
- AWS X-Ray

Resources

Related documents:

- Security Best Practices for Manufacturing OT
- Monitor edge application performance using AWS IoT Greengrass and AWS Distro for OpenTelemetry

Operate

MIDAOPS03: How do you understand the health of your industrial data workload?

Understanding the health of your industrial data workload is crucial for maintaining production reliability, optimizing processes, and making data-driven decisions.

Best practices

- MIDAOPS03-BP01 Define meaningful KPIs and metrics
- MIDAOPS03-BP02 Implement real-time monitoring and alerting capabilities
- MIDAOPS03-BP03 Implement predictive analytics and proactive anomaly detection

MIDAOPS03-BP01 Define meaningful KPIs and metrics

Manufacturing and industrial organizations require comprehensive visibility into the health and performance of their data-driven operations. Proper definition and monitoring of KPIs verify that manufacturing processes remain efficient, reliable, and aligned with business objective.

Desired outcome: Establish a comprehensive set of metrics and KPIs that provide insights into the performance, reliability, and efficiency of your operations, production, and industrial data workloads. These metrics and KPIs should enable proactive identification of issues and optimization opportunities.

Benefits of establishing this best practice: By adopting the best practice and prescriptive guidance, your organization gains improved visibility into manufacturing operations, enabling timely detection of potential production issues, enhanced operational efficiency, better resource utilization and reduced downtime and maintenance costs.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Implement industrial workload-specific KPIs monitoring and alerting that aligns with industry standards like ISA-95 and focuses on critical production metrics.

Implementation steps

Define industry-specific KPIs applicable for your workload. Examples of common metrics industrial data workloads:

1. System availability and uptime
2. Data ingestion rates and latency
3. Processing system performance
4. Security events and access patterns
5. Network connectivity status
6. Edge device health status
7. Gateway operational status
8. Resource utilization (for example, CPU, memory, or storage) for edge gateways and cloud services

MIDAOPS03-BP02 Implement real-time monitoring and alerting capabilities

Establish comprehensive monitoring of your industrial data infrastructure with automated alerts to quickly identify and respond to operational issues before they impact manufacturing processes. Configure appropriate thresholds and notification channels to quickly alert the right teams about system health, data quality, and performance anomalies.

Desired outcome: Your organization has established a clear, quantifiable understanding of industrial data workload performance through well-defined metrics and KPIs that align with business objectives and operational requirements.

Benefits of establishing this best practice: Monitoring and alerting capabilities allow the manufacturing team to quickly identify issues or impending problems that could impact production, quality, or maintenance.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Develop and implement a comprehensive metrics monitoring framework that captures both technical and business aspects of your industrial data workload.

Implementation steps

- Use AWS IoT Greengrass for device software to collect, process, and export data streams, including when devices are offline. It can act as an industrial data gateway between the shop floor and your AWS environment.
- For pre-built industrial connectors, implement the AWS Shop Floor Connectivity (SFC) open-source solution either as a stand-alone application or as an AWS IoT Greengrass component. SFC is a data ingestion enabler that delivers customizable greenfield and brownfield connectivity solutions. It addresses limitations and unifies data collection from existing IoT data collection services, allowing customers to consistently collect data from equipment across different vendors for use with various AWS services.
- Consider AWS IoT SiteWise to collect data from disparate data sources using OPCUA and MQTT connectors. AWS IoT SiteWise Monitor provides near real-time dashboard visualization of your key metrics and can be configured for alerting workflows.
- Use Amazon Managed Grafana to deliver integrated live dashboards for monitoring your production KPIs, operations, machine status, and alerts. The dashboards can use Grafana's native data source connectors to visualize data from multiple AWS services, including AWS IoT SiteWise, Amazon Timestream, Amazon RDS, and Amazon Aurora. This provides a unified visualization layer for your industrial data, enabling real-time monitoring and analysis of your manufacturing operations.
- Configure custom Amazon CloudWatch dashboards and alarms to track critical metrics, such as data ingestion rates and latency resource utilization (like CPU, memory, and storage) for edge gateways and cloud services, anomalies or deviations in key operational KPIs, security events and access patterns.

Key AWS services

- Amazon CloudWatch
- Amazon Managed Grafana
- AWS IoT Greengrass
- AWS IoT SiteWise

Resources

Related documents:

- Collecting data from industrial devices to AWS Services

MIDAOPS03-BP03 Implement predictive analytics and proactive anomaly detection

Use machine learning capabilities to analyse historical data patterns and identify potential issues before they impact your industrial operations.

Desired outcome: Identifying anomalies early enabling organizations to take corrective action before problems escalate and disrupt operations.

Benefits of establishing this best practice: By implementing these real-time monitoring, anomaly detection, and centralized observability capabilities, industrial organizations can understand the health of their data-driven operations and maintain the reliability, security, and performance required to support their business objectives.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Develop real-time monitoring and alerting across your industrial data systems to proactively detect and resolve issues. Use machine learning and predictive analytics to identify anomalies before they impact operations.

Implementation steps

1. Implement a centralized cloud data store as an enterprise historian using fully managed AWS services such as AWS IoT SiteWise and Amazon Timestream.
2. Configure multi-tier, cloud-optimized services to store time-series data. Use AWS IoT SiteWise or Timestream for hot data and Amazon S3 for cold data.
3. Use Amazon SageMaker AI AI to perform advanced AI/ML analyses of cold data for preventive maintenance, anomaly detection, and predictive quality insights. Use Amazon SageMaker AI Unified Studio as a single data and AI development environment where you can access and analyse your organization's data using the best tools across many use cases.
4. Use integrated observability tools to provide a centralized view of the overall system health and performance, such as AWS X-Ray for distributed tracing. AWS X-Ray allows the team to quickly pinpoint the root cause of issues that span multiple components of the data infrastructure.

5. For centralized observability and diagnostics, organizations can also use AWS CloudTrail to track API calls and configuration changes, providing an audit trail to understand the provenance of data and the lineage of operational changes.
6. For proactive anomaly detection, integrate Amazon GuardDuty to identify suspicious activity or security threats, and use machine learning models (for example, through Amazon SageMaker AI AI) to detect unusual patterns in equipment telemetry data.

Key AWS services

- AWS IoT SiteWise
- Amazon Timestream
- Amazon Simple Storage Service (Amazon S3)
- Amazon SageMaker AI AI
- Amazon GuardDuty
- AWS CloudTrail
- AWS X-Ray

Resources

Related documents:

- Strategies for modernizing data historians for the manufacturing industry in the AWS Cloud

Evolve

There are no operational excellence best practices for Evolve specific to the Manufacturing Industry Lens.

Security

The security pillar focuses on the ability to protect information, systems, and assets through risk assessments and mitigation strategies, while also delivering business value to the organization. In addition to the regulations that apply to most businesses, the manufacturing industry is challenged with industry-specific requirements such as frequently changing regulations and their variation by Region. Institutions that operate in more than one country or Region must also meet different requirements in different places. Manufacturing data systems are historically a frequent target of security incidents, both because of the assets they maintain and their fundamental role in the daily functioning of a modern society. The data mechanisms must comply with rules and laws around the protection of personal and regulatory information. The continuity of their operations depends directly on the security resilience they put in place.

Topics

- Design principles
- Security foundations
- Identity and access management
- Detection
- Infrastructure protection
- Data protection
- Incident response
- Application security

Design principles

In addition to the design principles found in the security pillar of the AWS Well-Architected Framework, the following security design principles can help you improve the security posture of your manufacturing data workloads:

- **Security by design**: Manufacturers must consider a security by design (SbD) approach to implement architectures that are pre-tested from a security perspective. SbD helps implement the control objectives, security baselines, security configurations, and audit capabilities for applications running on AWS. Standardized, automated, prescriptive, and repeatable design templates help accelerate the deployment of common use cases as well as help align with

security standards across multiple workloads. For example, to protect customer data and mitigate the risk of data disclosure or alteration of sensitive information by unauthorized parties, financial institutions need to employ encryption and carefully manage access to encryption keys. You can use SbD to turn on encryption for data at rest, in transit, and if necessary, at the application level by default.

- **Identify regulatory requirements to be implemented**: Regulators expect manufacturers to define security objectives for workloads and implement policies that help achieve those objectives. Regulators may also impose their own external requirements on specific workloads and expect institutions to monitor and report on their compliance with these requirements, with penalties for breaching them. Those requirements must be translated into security control objectives that are sustainable over time but flexible to adapt as regulations evolve.
- **Automated infrastructure and application deployment:** Automation helps companies perform and innovate quickly and scale security, compliance, and governance activities across their cloud environments. Manufacturers that invest in automated infrastructure and application deployment can accelerate the rate of deployments and embed security and governance best practices into their software development lifecycle.
- **Governance at scale:** Manual governance processes that rely on runbooks and checklists can lead to delays and inaccurate results. Automated governance provides a fast, definitive governance check for application deployments at scale. Governance at scale typically encompasses several key components:
  - Automating account provisioning to maintain strong security across numerous users and business units requesting cloud resources.
  - Enforcing and monitoring budgets across multiple accounts, workloads, and users for effective budget and cost management.
  - Managing security, risk, and compliance through automation to help your organization improve compliance while supporting business objectives.

- **Security concepts to handle customer information:** Manufacturing data protection requires fine-grained access controls at both data source and asset levels within AWS. All data must be encrypted at rest and in transit while following the principle of least privilege for role-based access. A multi-account strategy isolates different manufacturing environments, implementing both preventive and detective controls. Secure connectivity between manufacturing facilities and AWS is maintained through AWS Direct Connect with a Transit Account architecture for centralized security management.

Security in the manufacturing industry is composed of the following best practice areas.

- Security foundations
- Identity and access management
- Detection
- Infrastructure protection
- Data protection
- Incident response
- Application security

Before you architect a workload, you need to put in place practices that influence security. You should control who can do what. In addition, you want to be able to identify security incidents, protect your systems and services, and maintain the confidentiality and integrity of data through data protection. You should have a well-defined and practiced process for responding to security incidents. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations. Refer to AWS Well Architected Framework for the cloud security framework before adopting the manufacturing security practices.

Security foundations

MIDASEC01: How do you verify governance and compliance with security standards?

Manufacturing customers face unique challenges in securing workloads due to the convergence of IT and OT. Security in this environment must account for legacy protocols, isolated control systems, and real-time operational constraints. The following best practices provide guidance to implement robust governance, maintain compliance, and operate securely on AWS.

Best practices

- MIDASEC01-BP01 Create a security Shared Responsibility Model
- MIDASEC01-BP02 Standardize security baseline and implement role-based access controls
- MIDASEC01-BP03 Secure machine-to-machine and human-to-machine access
- MIDASEC01-BP04 Automate monitoring and reporting with cloud-ready compliance tools
- MIDASEC01-BP05 Implement incident response playbooks
- MIDASEC01-BP06 Establish a communication protocol between IT and OT systems

MIDASEC01-BP01 Create a security Shared Responsibility Model

Define and document a customized Shared Responsibility Model (SRM) that explicitly separates the security responsibilities between cloud environment, IT, and OT stakeholders. This provides clarity when managing security across cloud, edge, and on-premises industrial assets.

Desired outcome: Manufacturing stakeholders understand their distinct responsibilities in securing the infrastructure, data, and workloads across the IT/OT boundary.

Benefits of establishing this best practice: Reduces ambiguity during audits and incidents, strengthens collaboration across IT/OT, and supports secure innovation by clarifying boundaries of responsibility.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use the AWS Shared Responsibility Model as a foundation, then extend it to cover the specific roles and responsibilities across your IT and OT teams.

Implementation steps

- Review the AWS Shared Responsibility Model and corresponding manufacturing standards (for example, IEC 62443).
- Identify and document key security ownership across cloud, edge, and on-premises systems.
- Collaborate with IT, OT, and third-party vendors to define shared controls and data ownership boundaries.
- Incorporate the SRM into onboarding and compliance training for manufacturing teams.

Resources

Related documents:

Shared Responsibility Model

AWS Security Foundations

Amazon IoT

MIDASEC01-BP02 Standardize security baseline and implement role-based access controls

Establish a standardized security configuration across accounts and workloads using AWS Organizations, AWS IAM, and control policies. Implement role-based access controls (RBAC) to limit access based on job function and responsibility, especially across IT and OT environments.

Desired outcome: Establish consistent security controls across industrial cloud workloads that help protect both IT systems (MES, ERP) and OT systems (PLCs, SCADA), reducing exposure to unauthorized access and privilege escalation that could impact manufacturing operations.

Benefits of establishing this best practice: Enables centralized governance across IT and OT environments, improves auditability of manufacturing system access, and minimizes risk of misconfigurations through role-based controls aligned with production requirements.

Level of risk exposed if this best practice is not established: High

Implementation steps

- Define roles and responsibilities across IT and OT personas (for example, automation engineers, data scientists, or vendors).
- Create and assign IAM roles and policies aligned to job responsibilities.
- Use AWS IAM Identity Center for central identity management and federated access.
- Apply SCPs to enforce service-level restrictions at the organizational unit (OU) level.

Resources

Related documents:

- Service control policies (SCPs)
- Policies and permissions in IAM
- AWS Identity and Access Management

MIDASEC01-BP03 Secure machine-to-machine and human-to-machine access

Help protect communication between devices and systems in industrial settings by securing machine-to-machine (M2M) and human-to-machine (H2M) access using authentication, authorization, and encryption methods tailored to industrial protocols and devices.

Desired outcome: Avoid unauthorized access to industrial devices and trace actions across automation systems.

Benefits of establishing this best practice: Reduces the scope of impact, improves system integrity, and provides operational visibility for audit and compliance.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use X.509 certificates for M2M identity, IoT policies for access control, and managed bastion solutions like AWS Systems Manager Session Manager for H2M.

Implementation steps

- Provision devices with unique X.509 certificates using AWS IoT Core.
- Define and enforce AWS IoT policies for device communication.
- Use AWS Systems Manager Session Manager for secure remote access to edge and OT assets.
- Monitor access using AWS CloudTrail and AWS IoT Device Defender.

Resources

Related documents:

Iot Security Best Practices

Session Manager Userguide

MIDASEC01-BP04 Automate monitoring and reporting with cloud-ready compliance tools

Automate the collection, evaluation, and reporting of compliance evidence using AWS Cloud tools. Tailor configurations to meet industry-specific regulatory requirements such as NIST, CMMC, or ISO and IEC standards for manufacturing.

Desired outcome: Ongoing compliance posture monitoring and reduced manual effort in security and audit processes.

Benefits of establishing this best practice: Improves audit readiness, reduces cost and error in manual compliance efforts, and verifies continuous governance.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Establish manufacturing compliance baselines by documenting the required controls for industrial systems and mapping them to technical implementations.

Then, implement automated monitoring that evaluates industrial system configurations, tracks security control changes, and validates compliance with manufacturing standards.

Use AWS Config, Security Hub CSPM, and Audit Manager, configured specifically for manufacturing environments, to continuously monitor both IT and OT systems while maintaining required compliance evidence.

Implementation steps

- Enable AWS Config across all Regions and accounts.
- Use AWS Security Hub CSPM to aggregate security findings.
- Map controls in AWS Audit Manager to your industry framework.
- Schedule automated compliance report generation and alerting.

Resources

Related documents:

- What Is AWS Config?
- What is AWS Security Hub?
- What is AWS Audit Manager?

MIDASEC01-BP05 Implement incident response playbooks

Develop and test incident response playbooks for common OT/IT scenarios such as device compromise, unauthorized access, and data exfiltration. Verify your cross-functional coordination and readiness to minimize downtime and safety risks.

Desired outcome: Manufacturing organizations respond to incidents swiftly with predefined procedures, minimizing production disruption.

Benefits of establishing this best practice: Improves MTTD and MTTR, reduces risk of safety events, and improves regulatory and audit outcomes.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use AWS Systems Manager, AWS Lambda, and Amazon EventBridge to automate containment and response. Simulate scenarios to validate readiness.

Implementation steps

- Identify critical incident types and build corresponding playbooks.
- Use AWS Systems Manager Automation to orchestrate predefined remediation.
- Run chaos experiments using AWS Fault Injection Service to test response efficacy.
- Train IT and OT teams on roles and escalation procedures.

Resources

Related documents:

- AWS Systems Manager Automation
- AWS Fault Injection Service
- AWS Resilience Hub

MIDASEC01-BP06 Establish a communication protocol between IT and OT systems

Define secure communication and data exchange methods between IT and OT environments. Use edge services to control and monitor flows across the boundary and verify that only authorized systems interact.

Desired outcome: Reduced risk of unintended system access and data leakage between IT and OT, while enabling secure data-driven operations.

Benefits of establishing this best practice: Promotes security-by-design, improves clarity in responsibility demarcation, and supports long-term scalability and digital transformation.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Establish secure communication patterns between IT and OT networks by defining allowed protocols, data flows, and security controls.

Implement edge processing capabilities to manage data exchange, using protocol gateways for format translation and security enforcement.

Monitor all cross-boundary communications and implement automated alerts for unauthorized access attempts. This can be achieved using AWS IoT Greengrass and AWS IoT SiteWise for edge management, but the key is maintaining clear security boundaries while enabling necessary operational data flows.

Implementation steps

- Use AWS IoT Greengrass to manage and secure data ingestion at the edge.
- Define and monitor edge-to-cloud traffic patterns.
- Use VPC endpoints and private connectivity where needed for isolation.
- Document IT and OT interfaces, protocols, and access policies.

Resources

Related documents:

- What is AWS IoT Greengrass?
- What is AWS IoT SiteWise?

Identity and access management

MIDASEC02: How do you establish identity controls and access management patterns between industrial systems and humans?

Establish secure identity controls and access management patterns between industrial systems and human operators in manufacturing environments. This emphasizes proper authentication, authorization, and security policies to help protect both IT and OT systems while providing appropriate access levels for different roles in the manufacturing environment.

Best practices

- MIDASEC02-BP01 Enforce least privilege and security policies to control system access
- MIDASEC02-BP02 Enable multi-factor authentication (MFA) and token authorization (TA)

- MIDASEC02-BP03 Use centralized access management tools
- MIDASEC02-BP04 Develop a mechanism for regular review of IAM roles and policies

MIDASEC02-BP01 Enforce least privilege and security policies to control system access

Minimize access rights for users and systems by enforcing least privilege principles. This reduces the risk of lateral movement and privilege escalation in manufacturing environments.

Desired outcome: Users and systems can only access resources necessary for their roles, reducing exposure to unauthorized activities.

Benefits of establishing this best practice: Limits the scope of security incidents and enhances overall control over sensitive manufacturing resources.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Map access requirements for different production roles and critical system interactions. Implement IAM policies and SCPs aligned with manufacturing functions, enforcing clear boundaries between IT/OT systems. Monitor access patterns using AWS IAM Access Analyzer to continually align with operational needs.

Implementation steps

- Conduct role analysis across IT/OT to define minimum required privileges.
- Apply IAM permission boundaries and SCPs at account and organizational unit levels.
- Use IAM Access Analyzer to detect overly permissive roles.
- Implement regular audits to verify adherence to least privilege policies.

Resources

Related documents:

- Security best practices in IAM
- Using IAM Access Analyzer

MIDASEC02-BP02 Enable multi-factor authentication (MFA) and token authorization (TA)

Strengthen identity verification by enforcing MFA for human users and implementing token-based authorization for machines and services.

Desired outcome: Stronger authentication for both human users and industrial systems accessing AWS resources.

Benefits of establishing this best practice: Reduces risks associated with credential theft and replay attacks across IT/OT boundaries.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Enable MFA across all accounts, and integrate token services for secure, time-bound access.

Implementation steps

- Require MFA for all AWS accounts and IAM users using virtual or hardware devices.
- Implement SSO with MFA enforcement using AWS IAM Identity Center.
- Use temporary credentials and tokens through AWS Security Token Service for federated and service access.
- Enable and monitor MFA usage compliance with AWS Config rules.

Resources

- Using multi-factor authentication (MFA) in AWS
- What is AWS Identity and Access Management Access Analyzer?
- Welcome to the AWS Security Token Service API Reference

MIDASEC02-BP03 Use centralized access management tools

Consolidate identity and access management using centralized tools to streamline permission handling and improve visibility across multi-site operations.

Desired outcome: Reduced access complexity and improved governance across distributed industrial environments.

Benefits of establishing this best practice: Simplifies identity lifecycle management, supports consistent policy application, and enables centralized logging.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Adopt AWS IAM Identity Center or integrate with external identity providers to unify access controls.

Implementation steps

- Set up IAM Identity Center or integrate AWS accounts with your identity provider (for example, Active Directory).
- Configure fine-grained access permissions mapped to business roles.
- Enable centralized access logging and reporting.
- Regularly update identity mappings to reflect org structure changes.

Resources

- What is AWS IAM Identity Center?
- AWS IAM Identity Center

MIDASEC02-BP04 Develop a mechanism for regular review of IAM roles and policies

Establish processes to regularly review IAM roles and permissions to help prevent privilege creep and maintain access integrity over time.

Desired outcome: Stale or over-permissive access is detected and remediated proactively.

Benefits of establishing this best practice: Improves compliance posture, reduces operational risk, and enforces clean access policies aligned to least privilege.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Use tools like IAM Access Analyzer, AWS Config, and custom automation to audit and report access configuration regularly.

Implementation steps

- Establish a schedule for IAM access reviews.
- Use AWS IAM Access Analyzer to identify unused or overly broad permissions.
- Log and track review outcomes for auditing purposes.
- Automate revocation or modification of unneeded permissions using AWS Lambda or AWS Systems Manager.

Resources

- Using IAM Access Analyzer
- `iam-user-policy-check`

Detection

MIDASEC03: How do you design your manufacturing data environments for governance and compliance?

Design your manufacturing data environments with built-in governance and compliance capabilities, including data privacy, classification, and protection policies. This process involves implementing proper data handling procedures from the start, classifying, encrypting, and managing sensitive manufacturing data throughout its lifecycle, and maintaining regulatory compliance across different manufacturing operations.

Best practices

- MIDASEC03-BP01 Integrate and enforce privacy by design principles
- MIDASEC03-BP02 Implement industrial data classifications and protection policies
- MIDASEC03-BP03 Identify source data and set data classifications
- MIDASEC03-BP04 Implement industrial encryption policies

- MIDASEC03-BP05 Implement data retention policies for each data class

MIDASEC03-BP01 Integrate and enforce privacy by design principles

Apply privacy-by-design principles in the architecture of manufacturing data environments to help you comply with privacy regulations from the outset. This includes minimizing data collection, embedding access controls, and anonymizing personal data.

Desired outcome: Data privacy is protected from the start of the system design, reducing regulatory risks and improving trust with stakeholders.

Benefits of establishing this best practice: Aligns with data protection regulations like GDPR or CCPA, reduces risk of breaches, and supports ethical data use.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Design architectures with data minimization, separation, encryption, and auditability as core principles.

Implementation steps

- Identify personal and sensitive data processed within the manufacturing system.
- Limit collection and retention to only required fields and durations.
- Implement anonymization or pseudonymization where applicable.
- Embed access controls and audit logs from the start.

Resources

- General Data Protection Regulation (GDPR) Center
- Privacy by Design on AWS

MIDASEC03-BP02 Implement industrial data classifications and protection policies

Define classification tiers for industrial data (for example, public, internal, confidential, and restricted), and apply policies to control access, visibility, and protection levels accordingly.

Desired outcome: Manufacturing data is systematically classified and protected based on its criticality and sensitivity.

Benefits of establishing this best practice: Reduces risk of data leakage, safeguards sensitive data, and supports scalable governance models.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use AWS tools like Amazon Macie and AWS IAM policies to tag, monitor, and restrict access to classified data.

Implementation steps

- Define classification categories based on business needs and risk.
- Tag data assets using AWS resource tags or AWS AWS Glue Data Catalog.
- Use Amazon Macie to identify and monitor sensitive data types.
- Enforce access controls and monitoring based on classification tags.

Resources

- What is Amazon Macie?
- Populating the AWS Glue Data Catalog

MIDASEC03-BP03 Identify source data and set data classifications

Discover and classify data at the source (for example, sensors, PLCs, MES, and ERP systems) to apply appropriate controls from ingestion onward.

Desired outcome: Data is classified and governed from the moment it enters the system, helping prevent downstream risk exposure.

Benefits of establishing this best practice: Improves control over high-risk data, simplifies pipeline security, and enhances lineage tracking and compliance auditing.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Build data ingestion pipelines with tagging and classification integrated into the ingestion and cataloging layers.

Implementation steps

- Identify all upstream data sources contributing to the system.
- Use edge and gateway services to apply initial metadata or tags.
- Ingest data into AWS using services like Amazon Kinesis, AWS IoT Core, or AWS IoT SiteWise with classification.
- Catalog and tag datasets in AWS Glue or AWS Lake Formation.

Resources

- AWS Lake Formation
- AWS Glue
- What is AWS IoT SiteWise?

MIDASEC03-BP04 Implement industrial encryption policies

Apply encryption policies across all layers of the manufacturing data system, including at rest, in transit, and optionally during processing, to help protect sensitive operational and proprietary information.

Desired outcome: Data remains encrypted end-to-end, providing confidentiality and integrity and helping with regulatory alignment.

Benefits of establishing this best practice: Reduces impact of data breaches, supports regulatory compliance, and builds trust with partners and customers.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use AWS KMS for key management and enforce encryption using S3 bucket policies, VPC security, and service-level configurations.

Implementation steps

- Enable default encryption on all data stores (for example, Amazon S3, Amazon RDS, Amazon Redshift, and Amazon DynamoDB).
- Use TLS 1.2+ for all data in transit.
- Create customer-managed keys (CMKs) using AWS KMS for sensitive workloads.
- Regularly rotate keys and audit access with AWS CloudTrail.

Resources

- AWS Key Management Service
- Setting default server-side encryption behavior for Amazon S3 buckets

MIDASEC03-BP05 Implement data retention policies for each data class

Define and enforce data retention rules based on classification and regulatory requirements to reduce storage costs and minimize compliance risks.

Desired outcome: Only required data is retained, improving cost efficiency and reducing exposure of legacy data.

Benefits of establishing this best practice: Helps prevent unnecessary storage costs, reduce audit complexity, and support lifecycle compliance.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Use Amazon S3 lifecycle policies, AWS Glue table TTLs, and tagging strategies to automate enforcement of retention policies.

Implementation steps

- Define data retention durations based on classification and compliance needs.
- Apply tags and metadata to datasets to identify lifecycle requirements.
- Use S3 lifecycle rules or Glue jobs to delete or archive data.
- Regularly review and update retention policies based on changing regulations.

Resources

- Managing your storage lifecycle
- AWS Glue Documentation

Infrastructure protection

MIDASEC04: How do you enable network segmentation and access controls for industrial data?

Implement clear network segmentation between OT and IT systems in manufacturing environments. Use firewalls, perimeter networks, and dedicated network zones to manage and secure traffic flows between cloud, IT, and OT systems, while verifying that you have proper isolation and controlled data exchange to help protect critical manufacturing operations from potential security threats.

Best practices

- MIDASEC04-BP01 Segment OT networks from IT systems
- MIDASEC04-BP02 Use firewalls, perimeter networks, and dedicated network zones

MIDASEC04-BP01 Segment OT networks from IT systems

Establish clear segmentation between OT networks and IT systems to limit the scope of impact and reduce risk of lateral movement between environments.

Desired outcome: OT systems are logically and physically isolated from IT networks while enabling secure data flows.

Benefits of establishing this best practice: Reduces propagation of malware, minimizes risk of compromise of safety-critical systems, and supports compliance with industrial cybersecurity frameworks.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Implement logical segmentation using VPCs, subnets, routing controls, and edge gateways. Verify that minimal and controlled data flows between domains.

Implementation steps

- Design separate network segments for IT and OT environments using VPCs and subnets.
- Deploy gateway services like AWS IoT Greengrass or AWS Transit Gateway to manage traffic between zones.
- Apply network ACLs and security groups to limit access paths.
- Log and monitor all cross-boundary traffic using VPC Flow Logs and AWS Network Firewall.

Resources

- Secure data ingestion from OT to IT using AWS IoT SiteWise and AWS IoT Greengrass
- Security in your VPC

MIDASEC04-BP02 Use firewalls, perimeter networks, and dedicated network zones

Implement perimeter defenses such as firewalls, perimeter networks and dedicated network zones to manage and secure traffic flows between cloud, IT, and OT systems.

Desired outcome: Traffic is filtered and restricted at each trust boundary to enforce zero trust and layered defense.

Benefits of establishing this best practice: Helps improve network security posture, prevent direct exposure of OT systems, and support layered defense-in-depth architecture.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Deploy AWS Network Firewall, private subnets, and multi-tier perimeter network architectures to filter and control inbound and outbound traffic.

Implementation steps

- Deploy AWS Network Firewall or third-party appliances in VPCs.
- Create perimeter networks between public internet and sensitive workloads using public and private subnet models.
- Enforce rule groups that restrict IPs, ports, and protocols.
- Continuously monitor and update firewall rules and network configurations.

Resources

- What is AWS Network Firewall?
- Building a Scalable and Secure Multi-VPC AWS Network Infrastructure

Data protection

MIDASEC05: How do you manage access to industrial data within your workloads?

Establish clear and granular access controls for industrial data based on job roles and operational responsibilities. Implement centralized identity management systems, proper authorization models (ABAC or RBAC), and secure data sharing protocols to verify that only authorized personnel can access specific manufacturing data resources. These best practices help you maintain security and meet compliance requirements.

MIDASEC06: How do you securely and efficiently share data in industrial environments?

Implement secure and standardized protocols for sharing industrial data both internally and externally across manufacturing environments. Establish clear data ownership responsibilities, implement secure data exchange mechanisms like MQTT over TLS or OPC-UA with encryption, and maintain proper access controls to help protect data confidentiality and integrity while enabling effective collaboration across the industrial environment.

Best practices

- MIDASEC05-BP01 Define access permissions
- MIDASEC05-BP02 Build user identity solutions
- MIDASEC05-BP03 Implement data authorization models
- MIDASEC06-BP01 Use secure data exchange protocols
- MIDASEC06-BP02 Establish clear data ownership and sharing agreements

MIDASEC05-BP01 Define access permissions

Establish clear and granular access permissions to control who can access industrial data, based on job roles and operational responsibilities.

Desired outcome: Only authorized personnel can access specific data resources, reducing the risk of data leakage or misuse.

Benefits of establishing this best practice: Supports principle of least privilege, improves accountability, and reduces insider threats.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use IAM policies and resource tagging strategies to enforce fine-grained permissions aligned to user roles.

Implementation steps

- Inventory data assets and define roles for access control.
- Apply IAM roles and permissions based on job functions.
- Use resource tags to apply conditional access policies.
- Review and refine permissions regularly using AWS IAM Access Analyzer.

Resources

- Policies and permissions in IAM
- Using IAM Access Analyzer

MIDASEC05-BP02 Build user identity solutions

Deploy centralized identity systems that integrate with existing directories and cloud resources to manage user authentication and authorization efficiently.

Desired outcome: Consistent and secure identity management across all industrial and cloud systems.

Benefits of establishing this best practice: Improves user lifecycle management, simplifies access governance, and enhances login security with MFA and federation.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Implement AWS IAM Identity Center or integrate third-party identity providers with AWS.

Implementation steps

- Deploy IAM Identity Center for central identity control.
- Enable federation with existing AD or SAML-based systems.
- Set up MFA for all privileged roles and access points.
- Log all authentication events using AWS CloudTrail.

Resources

- AWS Identity and Access Management Access Analyzer
- Using multi-factor authentication (MFA) in AWS

MIDASEC05-BP03 Implement data authorization models

Design and implement scalable data authorization frameworks using attribute-based or role-based access control (ABAC or RBAC) models.

Desired outcome: Users are granted access to data based on roles, attributes, or context, supporting scalable and secure access decisions.

Benefits of establishing this best practice: Improves flexibility in data sharing, reduces over-permissions, and enhances compliance controls.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Define authorization attributes based on manufacturing context (like job function, facility, equipment type, or product line). Implement access controls using AWS Lake Formation for data-level permissions, AWS IAM conditions for resource access, and custom logic for specialized manufacturing scenarios.

Regularly review access patterns to verify alignment with production needs while maintaining security boundaries between IT and OT systems.

Implementation steps

- Define business roles and relevant attributes (for example, plant location or department).
- Implement policies in Lake Formation or Amazon S3 bucket policies using conditions.
- Monitor access patterns and refine authorization models over time.
- Log and audit access for compliance reporting.

Resources

- What Is AWS Lake Formation?
- Policies and permissions in IAM

MIDASEC06-BP01 Use secure data exchange protocols

Use secure and standardized protocols for sharing industrial data internally and externally, helping protect integrity and confidentiality.

Desired outcome: Data is transmitted securely across different systems and organizations without unauthorized interception or tampering.

Benefits of establishing this best practice: Helps prevent data breaches, supports secure collaboration, and aligns with industry data exchange standards.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use MQTT over TLS, HTTPS, or OPC-UA with encryption and certificate-based authentication.

Implementation steps

- Configure IoT and gateway devices to communicate over secure protocols.
- Enforce TLS 1.2 or higher for all data-in-transit.
- Implement endpoint authentication using certificates or tokens.
- Monitor traffic for anomalies using AWS IoT Device Defender.

Resources

- AWS IoT Core protocols
- What is AWS IoT Device Defender?

MIDASEC06-BP02 Establish clear data ownership and sharing agreements

Define clear ownership responsibilities and data sharing rules to guide access and usage across internal teams and partner organizations.

Desired outcome: Data is shared responsibly, with clarity around who controls, accesses, and governs its lifecycle.

Benefits of establishing this best practice: Improves accountability, supports regulatory compliance, and fosters trusted partnerships in the industrial environment.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Document data ownership roles and access conditions as part of your data governance framework and reinforce with access controls.

Implementation steps

- Identify and document data owners across all domains.
- Define acceptable use policies and access controls for each dataset.
- Use AWS Lake Formation and IAM policies to enforce agreements.
- Review agreements regularly to align with evolving compliance needs.

Resources

- AWS Lake Formation
- Policies and permissions in IAM

Incident response

MIDASEC07: How do you implement monitoring and threat detection for industrial systems?

Implement continuous monitoring and threat detection across both IT and OT environments in manufacturing settings. Deploy comprehensive monitoring tools and SIEM systems to detect anomalies and threats in real-time, while enabling centralized visibility and automated response capabilities to help protect industrial systems from security incidents.

Best practices

- MIDASEC07-BP01 Deploy continuous monitoring tools
- MIDASEC07-BP02 Implement SIEM systems

MIDASEC07-BP01 Deploy continuous monitoring tools

Implement continuous monitoring of your industrial systems to detect and respond to threats in real time. Monitoring should span cloud, edge, and on-premises systems across both IT and OT environments.

Desired outcome: Security teams can detect anomalies and threats early, reducing the impact of potential breaches.

Benefits of establishing this best practice: Enables real-time visibility, faster incident response, and proactive threat mitigation.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use AWS monitoring tools such as Amazon CloudWatch, AWS IoT Device Defender, and AWS Config to gather and analyze telemetry.

Implementation steps

- Enable Amazon CloudWatch metrics and logs for all AWS workloads.
- Deploy AWS IoT Device Defender to monitor device behavior and audit configurations.
- Use AWS Config to track configuration changes across resources.
- Integrate alerts with AWS SNS or AWS Security Hub CSPM for real-time response.

Resources

- What is AWS IoT Device Defender?
- What Is AWS Config?

MIDASEC07-BP02 Implement SIEM systems

Aggregate and analyze logs from industrial and cloud systems using a security information and event management (SIEM) system to help detect and respond to threats efficiently.

Desired outcome: Centralized visibility across hybrid environments enables faster detection of coordinated threats or unusual activities.

Benefits of establishing this best practice: Improves threat correlation, reduces alert fatigue, and strengthens compliance with audit trails.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use Amazon Security Lake or integrate with third-party SIEM tools such as Splunk or IBM QRadar for advanced analytics and incident workflows.

Implementation steps

- Set up Amazon Security Lake to collect and normalize logs from AWS and industrial sources.
- Integrate with a SIEM system for event correlation and alerting.
- Define detection rules and dashboards tailored to OT and ICS environments.
- Automate incident response workflows with runbooks or SOAR integrations.

Resources

- Amazon Security Lake
- AWS SIEM Partners

Application security

MIDASEC08: How do you adhere to regulatory guidelines with application teams in industrial environments?

Application teams must align their security controls with recognized industry standards like NIST 800-82, ISO/IEC 27001, and IEC 62443 in industrial environments. Conduct regular security awareness training programs for industrial application teams and implement automated compliance monitoring tools to maintain adherence to regulatory requirements while supporting secure manufacturing operations.

MIDASEC09: What practices do you follow for securing OT application systems?

Implement secure coding practices, regular vulnerability scanning, and automated patch management for OT applications and systems. Apply secure coding guidelines for applications and data integrations, perform regular vulnerability assessments and penetration testing, and maintain systematic patch management processes to help protect industrial control systems while verifying operational continuity and compliance with security standards.

Best practices

- MIDASEC08-BP01 Align security controls with industry standards
- MIDASEC08-BP02 Conduct regular security awareness programs
- MIDASEC09-BP01 Apply secure coding practices for applications and data integrations
- MIDASEC09-BP02 Perform regular vulnerability scans and penetration tests
- MIDASEC09-BP03 Automate patch management for ICS and connected data infrastructure

MIDASEC08-BP01 Align security controls with industry standards

Implement security controls across industrial workloads, and map those controls to recognized industry standards such as NIST 800-82, ISO/IEC 27001, and IEC 62443.

Desired outcome: Security implementations are auditable, and your organization improves its compliance with applicable industrial regulations.

Benefits of establishing this best practice: Streamlines audit processes, standardizes your security practices, and demonstrates due diligence in regulated industries.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use AWS Audit Manager and AWS Control Tower to align cloud configurations with industry-standard security frameworks.

Implementation steps

- Identify regulatory and industry-specific frameworks applicable to your workloads.
- Map AWS services and controls to required standards using AWS Audit Manager frameworks.
- Continuously monitor control adherence using AWS Config and Security Hub CSPM.
- Integrate compliance requirements into the CI/CD process for industrial apps.

Resources

- What is AWS Audit Manager?
- Security standards and controls reference

MIDASEC08-BP02 Conduct regular security awareness programs

Develop training and awareness programs for industrial application teams to reinforce best practices, policy understanding, and threat awareness.

Desired outcome: Engineering and operations staff are informed about evolving security threats and policies, reducing the risk of human error.

Benefits of establishing this best practice: Improves organizational security culture, reduces social engineering risks, and increases policy adherence.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Deliver targeted training sessions and simulate threat scenarios relevant to OT and industrial environments.

Implementation steps

- Identify key roles that need training (for example, developers, operators, and integrators).
- Use AWS learning resources or third-party courses tailored to industrial security.
- Conduct quarterly refresher sessions and phishing simulations.
- Track and report training completion and outcomes to leadership.

Resources

- AWS Security Fundamentals (Digital)
- NICE Framework Resource Center

MIDASEC09-BP01 Apply secure coding practices for applications and data integrations

Implement secure coding guidelines across industrial application development and integration pipelines to help prevent common vulnerabilities.

Desired outcome: Software and data interfaces in industrial systems are resilient against common attack vectors.

Benefits of establishing this best practice: Reduces injection attacks and vulnerabilities in custom code and enables secure interoperability.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Adopt secure coding checklists (like OWASP), enforce static code analysis, and secure APIs during development.

Implementation steps

- Incorporate security requirements into software specifications.
- Use tools like Amazon CodeGuru Reviewer and SonarQube in pipelines.
- Secure APIs with authorization, throttling, and validation.
- Review and test all data transformations and payloads for tampering risks.

Resources

- OWASP Top Ten
- What is Amazon CodeGuru Reviewer?

MIDASEC09-BP02 Perform regular vulnerability scans and penetration tests

Identify and mitigate vulnerabilities in applications and environments by conducting recurring scans and authorized penetration testing.

Desired outcome: Exposed vulnerabilities are proactively identified and mitigated before exploitation.

Benefits of establishing this best practice: Enhances visibility into system weaknesses and builds resilience against external and internal threats.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Use Amazon Inspector and integrate third-party scanning tools for deep and layered assessments.

Implementation steps

- Schedule recurring scans using Amazon Inspector across EC2 and container workloads.
- Perform black-box and white-box pen tests with third-party experts.

- Integrate findings with AWS Security Hub CSPM for centralized visibility.
- Remediate critical vulnerabilities through prioritized CI/CD updates.

Resources

- Getting started with Amazon Inspector
- Penetration Testing

MIDASEC09-BP03 Automate patch management for ICS and connected data infrastructure

Patch known vulnerabilities in a timely manner across industrial control systems (ICS), gateways, and cloud services by automating patch management processes.

Desired outcome: Patches are deployed consistently and timely, minimizing exposure to known exploits.

Benefits of establishing this best practice: Reduces manual overhead, enhances system stability, and supports compliance with vulnerability remediation SLAs.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Use AWS Systems Manager Patch Manager for cloud-side automation and coordinate closely with OT vendors for ICS-specific patch cycles.

Implementation steps

- Inventory all patchable assets across OT and IT systems.
- Use AWS Systems Manager Patch Manager to automate patching for EC2 and managed nodes.
- Align maintenance windows with production downtime cycles.
- Monitor patch compliance using AWS Config and AWS Systems Manager reports.

Resources

- AWS Systems Manager Patch Manager

- Patch Orchestration with AWS Systems Manager

Reliability

Reliable and resilient data infrastructure is critical for manufacturing organizations to drive sustained innovation, operational excellence, and product quality. MIDA is designed with reliability as a core principle, verifying that data is consistently available, processed correctly, and recoverable even in the face of disruptions.

Manufacturing environments face unique reliability challenges due to their hybrid nature and critical operational requirements. The integration of OT with IT systems, real-time processing needs, and the necessity to maintain production continuity demand a sophisticated approach to reliability. MIDA addresses these challenges through a comprehensive architecture that spans from the shop floor to the cloud.

The architecture employs a hybrid edge-cloud model where edge components provide resilient local processing for mission-critical applications while cloud infrastructure offers scalable, fault-tolerant services for broader data management and analytics. This approach means that manufacturing operations can continue even during network disruptions, maintaining data consistency and enabling recovery across the entire system.

Through services like AWS IoT Greengrass, AWS IoT SiteWise, Amazon Kinesis, and purpose-built databases, MIDA provides the foundation for highly available manufacturing systems. The architecture incorporates automated failover, data replication, and comprehensive monitoring capabilities to provide reliability at every layer, from sensor data collection to enterprise-wide analytics.

Topics

- Design principles
- Foundations
- Workload architecture
- Change management
- Failure management

Design principles

The following design principles can help you implement reliable manufacturing data architectures in the cloud:

- **Design for edge resilience:** Manufacturing operations require continuous data collection and control capabilities even during network disruptions. Implement robust edge processing capabilities that can operate autonomously during cloud connectivity issues. Use local data buffering, edge analytics, and automated synchronization mechanisms to maintain operational continuity while upkeeping data consistency when connectivity is restored. This approach helps prevent production disruptions and maintains data integrity across the manufacturing environment.
- **Enable fault isolation:** Manufacturing systems involve complex interactions between shop floor equipment, control systems, and enterprise applications. Design your architecture with isolated fault domains to help prevent cascading failures across these systems. Implement circuit breakers, bulkheads, and proper error handling to contain failures within specific components while maintaining overall system stability. This way, issues in one area don't compromise the entire manufacturing operation.
- **Implement reliable data flows:** Manufacturing data must move reliably between OT and IT systems without loss or corruption. Design data pipelines with guaranteed delivery mechanisms, proper ordering, and deduplication capabilities. Use buffering and retry mechanisms to handle intermittent connectivity issues while maintaining data consistency so that critical manufacturing data is accurately captured and processed for operational decision-making.
- **Build for recovery:** Manufacturing systems must quickly recover from failures to minimize production impact. Design your architecture with automated recovery mechanisms, including backup systems, failover capabilities, and disaster recovery procedures. Implement regular testing of recovery procedures and maintain clear documentation of recovery processes. This approach minimizes downtime and improves business continuity across manufacturing operations.
- **Monitor system health:** Manufacturing environments require comprehensive visibility into system health to maintain reliability. Implement proactive monitoring across all components - from edge devices to cloud services. Use automated alerting, predictive maintenance indicators, and clear escalation procedures to identify and resolve issues before they impact production. These practices help your team rapidly respond to potential problems and maintain system reliability.
- **Scale with production demands:** Manufacturing workloads vary with production schedules and business cycles. Design systems that can automatically scale resources based on actual demand while maintaining reliability. Implement proper capacity planning and automated scaling mechanisms that align with manufacturing patterns. As a result, your system can have reliable performance during both peak production periods and routine operations.

Foundations

MIDAREL01: How do you manage service quotas and constraints with real-time data processing and critical operations?

Manufacturing environments rely heavily on real-time data processing for operations like production monitoring, quality control, and predictive maintenance. Service quotas and constraints can impact the reliability of these critical operations if not properly managed.

Best practices

- MIDAREL01-BP01 Monitor manufacturing-specific service quotas
- MIDAREL01-BP02 Scale network architecture for manufacturing operations

MIDAREL01-BP01 Monitor manufacturing-specific service quotas

Implement proactive monitoring and management of service quotas that are critical to manufacturing operations. Manufacturing workloads often have unique patterns of resource consumption based on production cycles, which require specialized monitoring approaches.

Desired outcome: Your manufacturing workloads remain fully operational without disruption from quota limits, especially during peak production periods or when processing high volumes of sensor data. Early identification of approaching quota limits allows for timely adjustments.

Benefits of establishing this best practice: Helps prevent production downtime caused by service disruptions, maintains consistent performance of real-time monitoring systems, and supports continuous operation of automated manufacturing processes. This approach also helps optimize costs by right-sizing resources according to actual production needs.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Before implementing service quota management in manufacturing environments, it's crucial to understand how quotas impact different aspects of your industrial operations. Manufacturing workloads often have unique patterns of resource consumption based on production cycles,

maintenance windows, and seasonal demands. A comprehensive quota management strategy should consider both steady-state operations and peak production periods to help prevent disruptions to critical manufacturing processes.

Start by establishing baseline quota requirements for different manufacturing scenarios:

- Normal production operations
- Peak production periods
- Maintenance and quality inspection windows
- End-of-month/quarter reporting cycles
- Emergency response situations

Consider implementing a multi-layered approach that includes proactive monitoring, automated alerting, and buffer capacity for unexpected spikes in demand. This strategy helps prevent quota-related disruptions while maintaining operational efficiency.

Implementation steps

1. **Use AWS Service Quotas and AWS Trusted Advisor:** Configure Service Quotas to track usage across critical manufacturing systems and set up Amazon CloudWatch alarms to notify when approaching thresholds. Use Trusted Advisor to receive recommendations about service limits and usage optimization.
2. **Implement request queuing:** Deploy Amazon SQS queues to buffer incoming requests during peak production periods, helping prevent quota exceedance while processing all data eventually.
3. **Configure dynamic resource allocation:** Use AWS Auto Scaling to automatically adjust resource capacity based on manufacturing production schedules and real-time demand from factory floor systems.
4. **Set up cross-Region redundancy:** Implement AWS Global Accelerator to distribute workloads across multiple Regions, reducing the risk of regional quota limitations affecting manufacturing operations.

Key AWS services

- AWS Service Quotas
- AWS Trusted Advisor

- Amazon CloudWatch
- Amazon SQS
- AWS Auto Scaling
- AWS Global Accelerator

Resources

- Viewing service quotas and setting CloudWatch alarms
- Using Amazon SQS to manage request rates
- Target tracking scaling policies for Amazon EC2 Auto Scaling
- Optimizing global network performance with AWS Global Accelerator

MIDAREL01-BP02 Scale network architecture for manufacturing operations

In manufacturing environments, network reliability is crucial as disruptions can directly impact production, quality, and safety. A reliable network architecture must provide continuous operations even during component failures, maintenance windows, or unexpected spikes in industrial data traffic. Implementing redundant connections, proper network segmentation, and automated failover mechanisms helps maintain operational continuity across the manufacturing floor.

Desired outcome: A resilient network infrastructure that supports manufacturing operations without disruption, creating a continuous data flow between manufacturing equipment and edge devices, with appropriate redundancy and failover capabilities.

Benefits of establishing this best practice:

- Minimizes production downtime due to network failures
- Provides consistent data collection from production equipment
- Maintains operational integrity during maintenance or failures
- Supports scaling production capacity without compromising reliability
- Enables real-time decision making through dependable data transmission

Level of risk exposed if this best practice is not established: High

Implementation guidance

Establish redundant connectivity with automatic failover capabilities to improve the continuity of your network operations. Implement multiple network paths to help prevent single points of failure, with automated routing to alternative connections during disruptions.

Consider implementing redundant connections using services like AWS Direct Connect and AWS Site-to-Site VPN with AWS Transit Gateway to enable automatic failover and routing capabilities across multiple network paths.

Design network segmentation that isolates critical manufacturing zones (production, quality control, and maintenance) while maintaining appropriate communication paths. Implement security boundaries that help protect sensitive manufacturing operations without impeding necessary data flows.

Consider using Amazon VPC with multiple subnets and AWS Transit Gateway to create isolated network zones while maintaining controlled communication paths. Network ACLs and security groups can provide additional security boundaries for manufacturing-specific requirements.

Deploy comprehensive network monitoring that tracks performance metrics, sets up alerts for latency or throughput issues, and creates dashboards specific to manufacturing network requirements. Include proactive monitoring of network health and automated notification systems for potential issues.

Consider implementing monitoring using services like Amazon CloudWatch and AWS Network Manager to track network health metrics and create automated alerts. Amazon EventBridge can help orchestrate automated responses to network events.

Configure automated recovery procedures when network anomalies are detected in manufacturing environments. Implement self-healing capabilities that can automatically reroute traffic or fail over to backup systems without manual intervention.

Consider using AWS Systems Manager Automation with Amazon Route 53 health checks to automatically detect and respond to network issues. Use AWS Lambda to implement custom recovery logic based on your manufacturing requirements.

Key AWS services

- AWS Direct Connect

- AWS Site-to-Site VPN
- AWS Transit Gateway
- Amazon VPC
- AWS Network Manager
- Amazon CloudWatch
- AWS Lambda
- AWS Auto Scaling

Resources

Building a Scalable and Secure Multi-VPC AWS Network Infrastructure

AWS Direct Connect Resiliency Recommendations

Monitoring Network Health with Network Manager

AWS Transit Gateway Network Manager for Industrial IoT

Workload architecture

MIDAREL02: How does your industrial data architecture maintain operational continuity when OT or IT systems fail during production?

Manufacturing facilities depend on continuous data flow between OT and IT systems to maintain production. When connectivity issues or system failures occur, production lines can experience costly downtime, defective products, and safety risks if data architecture isn't designed for resilience.

Best practices

- MIDAREL02-BP01 Design resilient industrial integration patterns
- MIDAREL02-BP02 Verify data consistency and availability across OT/IT systems through redundancy and failover mechanisms
- MIDAREL02-BP03 Enable automated recovery mechanisms

MIDAREL02-BP01 Design resilient industrial integration patterns

Implement redundant and fault-tolerant architectural patterns that maintain operational continuity during system failures. Your manufacturing data architecture should include local processing capabilities, data buffering, and automatic recovery mechanisms to verify that critical production processes continue even when cloud connectivity or key systems fail.

Desired outcome: Manufacturing operations continue with minimal disruption when systems fail, with production data preserved and synchronized once systems are restored. Critical operational parameters remain accessible to machine operators, and production workflows continue functioning through degraded modes.

Benefits of establishing the Best Practice: By implementing resilient integration patterns, manufacturers can minimize production downtime, maintain product quality during system failures, preserve critical operational data, and properly synchronize systems once they are restored, ultimately helping protect revenue and customer satisfaction.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Implement edge computing solutions

First, assess your manufacturing systems' criticality, required processing latency, and local autonomy requirements.

Document which operations must continue during cloud connectivity disruptions.

Design local processing capabilities that can operate independently when disconnected, with clear rules for autonomous decision-making and data buffering.

Consider implementing AWS IoT Greengrass on factory floor gateways to enable local processing and autonomous operations during connectivity issues, with rules that determine which operations can continue locally and which require cloud connectivity.

Create data buffering mechanisms

Begin by analyzing your data generation patterns, storage requirements, and acceptable data latency windows. Document recovery time objectives (RTOs) and recovery point objectives (RPOs) for different data types.

Implement local storage mechanisms that can retain manufacturing data during outages and automatically synchronize when connectivity is restored.

Consider using AWS IoT SiteWise Edge to store time-series production data locally during connectivity interruptions, with automatic synchronization to AWS IoT SiteWise in the cloud when connectivity returns.

Design circuit breaker patterns

Start by mapping dependencies between manufacturing systems and identifying potential failure points. Define graceful degradation modes for each critical service and establish fallback mechanisms.

Implement patterns that can detect failures and automatically switch to alternative processing paths.

Consider implementing circuit breakers using AWS Step Functions to handle failures in downstream dependencies and automatically switch to predefined fallback mechanisms that maintain critical manufacturing operations.

Configure automatic failover systems

First document your manufacturing system's availability requirements and acceptable downtime windows. Establish clear failover activations and recovery procedures.

Design redundant connectivity paths with automated switching capabilities to maintain continuous operations.

Consider deploying redundant AWS IoT Core endpoints with automatic failover capabilities to provide reliable connectivity between manufacturing systems and cloud services, with monitoring to verify successful failover operations.

Key AWS services

- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS Step Functions
- AWS IoT Core
- Amazon Kinesis Data Streams

Resources

- Implementing Edge Computing with AWS IoT Greengrass
- Configuring Data Buffering with AWS IoT SiteWise Edge
- Designing Resilient Industrial Applications with AWS Step Functions
- Building High Availability Industrial IoT Solutions with AWS

MIDAREL02-BP02 Verify data consistency and availability across OT/IT systems through redundancy and failover mechanisms

Manufacturing environments operate with complex interactions between OT and IT systems. When these systems fail, the impact can cascade through production lines, quality control processes, and enterprise planning systems. Successfully maintaining data consistency and system availability requires careful orchestration of data synchronization, communication paths, and recovery procedures across the manufacturing technology stack.

Desired outcome: Production operations continue with minimal disruption during system failures. Critical operational data remains available and consistent during disruptions, allowing for automated or manual fallback procedures to maintain production output while primary systems are restored.

Benefits of establishing the Best Practice:

- Minimize production downtime and associated revenue losses.
- Preserve data integrity across manufacturing systems.
- Enable smooth recovery without data reconciliation challenges.
- Maintain quality control and regulatory compliance during system failures.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Real-time manufacturing data management

First, map your critical manufacturing data flows and establish requirements for data synchronization across OT/IT systems. Document recovery time objectives (RTOs), recovery point objectives (RPOs), and regulatory compliance needs.

Define data models that support consistent representation of equipment states and process parameters across systems.

For implementation, establish standardized data models and real-time state management systems for production equipment and processes. Verify data consistency through distributed databases with conflict resolution capabilities.

Consider using AWS IoT SiteWise for consistent equipment and process data models, AWS IoT Core's Device Shadow service for reliable state management, and Amazon DynamoDB global tables for distributed manufacturing data management.

Data validation and quality controls

Document data quality requirements for different manufacturing processes and establish validation rules. Define acceptable tolerance ranges for process parameters and identify critical quality metrics that must be maintained during system transitions.

Implement automated validation mechanisms that verify data integrity across systems, with standardized data quality rules and monitoring.

Set up continuous validation of data synchronization between OT/IT systems and establish regular backup points for critical manufacturing datasets.

Consider implementing AWS Glue Data Quality rules for automated validation, AWS IoT SiteWise Asset Models for standardized validation, and AWS Backup for consistent recovery points.

Recovery mechanisms and failover systems

Document system dependencies and define acceptable failover scenarios. Establish clear procedures for both automated and manual recovery processes.

Determine sequence of operations for graceful system transitions that preserve data integrity. Design idempotent processing mechanisms to help prevent data duplication during recovery.

Implement point-in-time recovery capabilities for critical production systems and configure automated failover with health monitoring.

Consider using Amazon DynamoDB Point-in-Time Recovery for maintaining accurate operational states, Amazon Route 53 health checks for automated failover, and AWS Systems Manager for coordinated recovery automation.

Key AWS services

- AWS IoT SiteWise
- AWS IoT Core
- Amazon Timestream
- Amazon DynamoDB
- AWS Glue Data Quality
- Amazon CloudWatch
- AWS Backup
- Amazon Route 53
- AWS Systems Manager

Resources

- AWS IoT SiteWise for industrial data collection and monitoring
- Using Device Shadows for manufacturing equipment state management
- Implementing DynamoDB Global Tables for distributed manufacturing data
- AWS Glue Data Quality for manufacturing data validation

MIDAREL02-BP03 Enable automated recovery mechanisms

Design your manufacturing workloads with automated recovery processes that allow production to continue despite OT and IT system failures. Implement edge processing capabilities that can operate independently when disconnected from central IT systems.

Desired outcome: Production operations continue without interruption even when data collection systems, network connectivity, or cloud services experience failures. Critical manufacturing data is preserved and synchronized once systems are restored.

Benefits of establishing the best practice: Minimizing production downtime, preserving data integrity during system failures, reducing manual intervention during recovery processes, and maintaining product quality metrics during system disruptions.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Implement edge computing capabilities

Assess your manufacturing system's autonomy requirements and identify critical processes that must continue during connectivity loss. Document acceptable operational modes during disconnected states and establish clear thresholds for autonomous decision-making.

Design local processing systems that support offline operations, with clearly defined rules for degraded mode operations and boundaries for autonomous decisions based on safety requirements and operational parameters.

For implementation, deploy edge computing capabilities that can maintain essential operations during network disruptions. This requires integration with industrial control systems and PLCs for real-time operations with appropriate redundancy and failover mechanisms for critical manufacturing processes.

Consider using AWS IoT Greengrass to maintain local processing and decision-making when connectivity is lost. Configure AWS IoT Greengrass components to cache production data locally and execute critical workflows without cloud dependency.

Deploy local data buffering mechanisms

Analyzing your data generation patterns and retention requirements. Define priorities for different data types and establish minimum retention periods based on operational and compliance needs. Document synchronization requirements for when connectivity is restored.

Implement robust local storage systems with appropriate capacity planning for expected outage durations. Configure data retention policies that preserve critical production metrics during disconnection periods.

Consider using AWS IoT SiteWise Edge to continually collect data during network outages, with configured data retention policies for high-value production metrics.

Create automated data synchronization protocols

Map data dependencies and establishing synchronization priorities. Define conflict resolution rules for handling data updates during disconnected operations.

Document recovery procedures for different types of outages.

Design synchronization mechanisms that can handle data reconciliation when connectivity is restored. Implement prioritization rules for critical production data during reconnection events.

Consider using AWS IoT Core rules to prioritize critical production data during reconnection events, with clear handling of potential data conflicts.

Establish monitoring and alerting systems

Define normal operational parameters and alert thresholds. Document escalation procedures and response requirements for different types of failures.

Establish KPIs for measuring recovery effectiveness. Implement comprehensive monitoring of both edge and cloud components.

Configure automated alerts based on predefined thresholds and create recovery procedure activations.

Consider using Amazon CloudWatch to detect failures and automatically initiate recovery procedures based on predefined thresholds specific to manufacturing processes.

Key AWS services

- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS IoT Core
- Amazon CloudWatch
- AWS Lambda
- Amazon S3

Resources

- Implementing local processing with AWS IoT Greengrass
- Data buffering with AWS IoT SiteWise Edge
- Designing resilient manufacturing applications with AWS IoT Core

Change management

MIDAREL03: How do you design your workload to adapt to changes in demand?

Manufacturing environments experience variable workloads due to production shifts, seasonal demands, and unforeseen changes. Designing systems that can automatically adjust to these fluctuations improves performance and cost efficiency.

Best practices

- MIDAREL03-BP01 Implement dynamic scaling for shop floor data ingestion pipelines

MIDAREL03-BP01 Implement dynamic scaling for shop floor data ingestion pipelines

Manufacturing operations generate varying volumes of data from machines, sensors, and production lines. Dynamic scaling verifies that your data ingestion pipelines can handle peak production times without overprovisioning during slower periods.

Desired outcome:

A resilient data ingestion system that automatically scales to accommodate fluctuating data volumes from the shop floor, providing consistent performance during peak production while optimizing costs during lower activity periods.

Benefits of establishing this best practice:

- Cost optimization by avoiding overprovisioning of resources.
- Improved system reliability during production surges.
- Consistent data processing regardless of production volumes.
- Enhanced ability to capture critical manufacturing data during unexpected events.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Design scalable data ingestion architecture

Assess your manufacturing data sources, volumes, and velocity requirements. Document peak production periods, shift patterns, and seasonal variations that affect data generation rates.

Identify critical data streams that require guaranteed delivery and establish performance baselines for normal operations.

Define acceptable latency thresholds for different types of manufacturing data.

For implementation, create a flexible ingestion architecture that can automatically adapt to changing production volumes. Design ingestion paths with appropriate buffering and throttling mechanisms to handle sudden surges in manufacturing data.

Consider using AWS IoT Core and AWS Lambda for serverless ingestion points that automatically scale with the volume of incoming sensor data from manufacturing equipment.

Configure stream processing capabilities

Map your data processing requirements and identifying real-time analytics needs.

Document throughput requirements for different production scenarios and establish monitoring thresholds.

Define scaling triggers based on production metrics rather than just system metrics.

For implementation, establish stream processing mechanisms that can handle variable throughput from production lines while maintaining data consistency. Include automatic scaling capabilities based on actual production demands and data volumes.

Consider setting up Amazon Kinesis Data Streams with appropriate shard management to handle varying throughput from production lines.

Implement data flow controls

Analyze your data priority levels and establish handling rules for different data types.

Define quality of service requirements for critical manufacturing data and document acceptable processing delays for non-critical data.

Establish clear overflow handling procedures for peak production periods.

For implementation, deploy mechanisms to manage data flow rates and help prevent system overload during production peaks. Create prioritization schemes that process critical manufacturing data first during high-load periods.

Consider implementing Amazon SQS queues between data collection points and processing systems to handle sudden surges in manufacturing data.

Establish comprehensive monitoring

Define key performance indicators for your data ingestion system. Document alerting thresholds based on production requirements and establish escalation procedures.

Create baseline metrics for normal operations across different production scenarios.

For implementation, deploy monitoring systems that track both technical metrics and production-related indicators. Configure automated alerts for potential bottlenecks or capacity issues before they impact production.

Consider using Amazon CloudWatch to track performance metrics and adjust scaling parameters based on historical production patterns and seasonal manufacturing demands.

Key AWS services

- Amazon Kinesis Data Streams
- AWS Lambda
- Amazon SQS
- AWS IoT Core
- Amazon CloudWatch
- AWS Auto Scaling

Resources

- Auto Scaling for Amazon Kinesis Data Streams
- Serverless event-driven architecture for industrial data ingestion
- Implementing the IoT analytics pipeline for manufacturing
- Designing scalable IoT applications with AWS

Failure management

MIDAREL04: How do you back up data?

Manufacturing operations depend on consistent access to critical data such as production plans, machine configurations, and quality control metrics. Without proper backup strategies, manufacturing facilities risk significant downtime, loss of intellectual property, and inability to meet production schedules if data becomes corrupted or unavailable.

Best practices

- MIDAREL04-BP01 Implement a multi-layered backup strategy
- MIDAREL04-BP02 Use cloud replication for manufacturing data resilience
- MIDAREL04-BP03 Use versioning and automated backup procedures
- MIDAREL05-BP01 Design a multi-Region disaster recovery strategy
- MIDAREL05-BP02 Implement and regularly test DR procedures

MIDAREL04-BP01 Implement a multi-layered backup strategy

Manufacturing environments require comprehensive backup solutions that account for both on-premises equipment data and cloud-based systems. A multi-layered approach helps protect critical production data against various failure scenarios, from local hardware failures to Regional disruptions.

Desired outcome: Manufacturing data is consistently backed up, recoverable, and protected against various failure modes. Recovery Point Objectives (RPOs) and Recovery Time Objectives (RTOs) are met to minimize production impact during recovery operations.

Benefits of establishing this best practice: Implementing a multi-layered backup strategy helps reduce manufacturing downtime, protect intellectual property, improve compliance with industry regulations, and provide business continuity during disruptions. It also improves recovery from data loss incidents and provides confidence in data integrity.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Design comprehensive backup policies

Analyze your manufacturing data criticality levels and establish retention requirements for different data types. Document regulatory compliance needs, audit requirements, and data sovereignty rules.

Define Recovery Time Objectives (RTOs) and Recovery Point Objectives (RPOs) for different manufacturing systems.

Map dependencies between production systems to understand the impact of data loss scenarios.

For implementation, create tiered backup policies aligned with data criticality and compliance requirements. Design automated backup schedules that don't impact production operations and provide consistent backup states across interconnected systems.

Consider using AWS Backup to help protect manufacturing data across multiple services with schedules aligned to production cycles and maintenance windows.

Implement cross-Region redundancy

Identify geographic distribution requirements for your manufacturing operations. Document Regional compliance requirements and data residency restrictions. Establish performance requirements for cross-region data access and recovery procedures.

For implementation, create backup copies in geographically separate regions to help protect against regional disasters. Verify that your backup strategies take into account data sovereignty requirements while providing necessary redundancy.

Consider implementing AWS Backup cross-Region copy capabilities to provide manufacturing data resilience against Regional disruptions.

Establish hybrid backup solutions

Map your on-premises manufacturing systems and their backup requirements. Document integration points between cloud and on-premises systems. Define data transfer windows that align with production schedules and network capacity.

For implementation, create backup mechanisms that help protect both cloud and on-premises manufacturing data. Design efficient data transfer methods that minimize impact on production networks.

Consider deploying AWS Storage Gateway to create hybrid backup solutions that help protect on-premises manufacturing equipment data while enabling seamless recovery.

Configure version control and lifecycle management

Establish version retention requirements for critical manufacturing data. Document change control procedures and audit requirements. Define archival policies based on data access patterns and compliance needs.

For implementation, create versioning policies that maintain appropriate historical records while managing storage costs. Design lifecycle rules that automatically transition aging backups to cost-effective storage tiers.

Consider implementing S3 Versioning with lifecycle policies to enable point-in-time recovery of critical files like CAD designs and production recipes.

Key AWS services

- AWS Backup
- Amazon S3
- AWS Storage Gateway
- Amazon EBS
- AWS Backup cross-region copy

Resources

- Creating a backup plan with AWS Backup
- AWS Storage Gateway for manufacturing data protection
- Using cross-Region backup copies
- Retaining multiple versions of objects with S3 Versioning

MIDAREL04-BP02 Use cloud replication for manufacturing data resilience

Implement automated, multi-Region cloud replication to maintain manufacturing operational data integrity across production systems, quality control databases, and equipment monitoring systems. Establish consistent backup patterns that align with recovery time objectives (RTOs) critical to manufacturing operations.

Desired outcome: Manufacturing data is replicated according to defined recovery point objectives, helping you avoid data loss in failure scenarios. Critical systems can be restored rapidly, minimizing production downtime and maintaining continuity of operations.

Benefits of establishing this best practice:

- Minimizes production downtime during recovery operations.

- Preserves historical manufacturing data needed for quality control and compliance.
- Enables rapid recovery of operational systems supporting the production line.
- Provides resilience against Regional failures that could impact manufacturing operations.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Design automated replication strategies

Assess your manufacturing data types and their replication requirements. Document recovery point objectives (RPOs) for different systems and establish data consistency requirements across replicated environments. Map dependencies between manufacturing systems to understand the impact of replication delays. Define acceptable latency thresholds for data synchronization.

For implementation, create automated replication mechanisms that maintain data consistency without impacting production performance. Design replication schedules that align with manufacturing cycles and maintenance windows.

Consider using AWS Backup for automated and centralized replication management that aligns with your production schedules and operational requirements.

Implement multi-Region data distribution

Analyze your geographic manufacturing footprint and Regional compliance requirements. Document data sovereignty rules and cross-border data transfer restrictions. Define performance requirements for data access across Regions. Establish recovery procedures for Regional failures.

For implementation, create Region-specific replication policies that help you maintain compliance while providing higher data availability. Design efficient data transfer mechanisms that optimize costs and performance.

Consider implementing Amazon S3 Cross-Region Replication for critical production data, making manufacturing specifications and quality records available across required geographic locations.

Configure database replication

Map your manufacturing databases and their criticality levels. Document consistency requirements for replicated databases and establish acceptable lag times. Define failover procedures and recovery priorities.

For implementation, create database replication mechanisms that maintain data integrity during normal operations and system failures. Design automated failover procedures that minimize production disruption.

Consider using AWS Database Migration Service (DMS) continuous replication for manufacturing databases containing production recipes and equipment configurations.

Establish deployment automation

Document your infrastructure requirements and configuration standards. Define change management procedures and testing requirements for replicated environments. Establish validation protocols for replicated resources.

For implementation, create automated deployment procedures that provide consistent configuration across replicated environments. Design validation checks that verify replication health and data consistency.

Consider implementing AWS CloudFormation templates to automate the deployment of consistent backup infrastructure across manufacturing facilities.

Key AWS services

- AWS Backup
- Amazon S3
- AWS Database Migration Service (DMS)
- AWS CloudFormation

Resources

- AWS Backup
- S3 Cross-Region Replication for Business Continuity
- Continuous Database Replication with AWS DMS
- Automating Backup Deployment with CloudFormation

MIDAREL04-BP03 Use versioning and automated backup procedures

Implement comprehensive data backup strategies with automated procedures and versioning to help protect manufacturing data assets from corruption, accidental deletion, or system failures, helping improve business continuity and speeding up recovery.

Desired outcome: Manufacturing data is consistently backed up with comprehensive versioning that captures different types of changes and modifications over time. This encompasses full versions of production recipes and process parameters, iterative versions of equipment configurations and control logic, sequential versions of quality control specifications, and temporal versions showing historical changes to manufacturing master data. The versioning system maintains delta versions for incremental changes to production workflows and baseline versions that establish known good states for critical systems. The versions are properly tagged, cataloged, and retrievable to support various recovery scenarios, from point-in-time restoration to full system rollbacks. This comprehensive versioning strategy helps you consistently meet RTOs and RPOs, maintaining operational continuity while providing a clear audit trail of changes for compliance and troubleshooting purposes.

Benefits of establishing this best practice: Implementing automated backups with versioning reduces the risk of data loss, helps you improve compliance with industry regulations, minimizes downtime during recovery events, and provides audit trails for manufacturing processes and quality control.

Level of risk exposed if this best practice is not established: High

Implementation guidance

Establish versioning policies

Identify version control requirements for different types of manufacturing data - production recipes, equipment configurations, quality specifications, and operational parameters. Document retention requirements for each data type based on regulatory compliance, quality audits, and operational needs. Define clear versioning rules including baseline versions, major and minor changes, and delta tracking requirements. Map dependencies between versioned items to maintain system-wide consistency.

For implementation, create versioning mechanisms that maintain complete historical records while optimizing storage utilization. Design automated version management that integrates with existing manufacturing workflows.

Consider implementing S3 versioning and lifecycle policies that support manufacturing data versioning requirements while automatically managing version retention and archival.

Configure automated backup procedures

Analyze your backup requirements across different manufacturing systems. Document backup windows that align with production schedules and maintenance periods. Establish verification procedures to verify backup integrity and completeness. Define backup frequency based on data change rates and business risk tolerance.

For implementation, create automated backup procedures that maintain data consistency across interconnected manufacturing systems. Design backup validation checks that verify both content and version integrity.

Consider using AWS Backup to create automated, scheduled backups of manufacturing systems with appropriate retention policies and integrity checks.

Implement data transfer optimization

Map data transfer patterns and volumes across your manufacturing environment. Document network capacity constraints and identify potential bottlenecks. Define acceptable transfer windows that don't impact production operations.

For implementation, create efficient data movement mechanisms that minimize impact on production systems. Design transfer optimization strategies including compression and deduplication.

Consider using AWS DataSync to automate and optimize data transfer between on-premises manufacturing systems and AWS storage services.

Establish monitoring and validation

Define success criteria for backup and versioning operations. Document monitoring requirements including version completeness, backup status, and system health indicators. Establish alert thresholds and escalation procedures.

For implementation, create comprehensive monitoring systems that track both technical success and business relevance of versioning operations. Design automated validation checks that verify version consistency and backup integrity.

Consider implementing AWS Backup test restore capabilities to regularly verify backup integrity and recovery procedures.

Key AWS services

- Amazon S3
- AWS Backup
- Amazon RDS
- AWS DataSync
- Amazon EBS

Resources

- Using versioning in S3 buckets
- Creating manufacturing backup plans with AWS Backup
- Backup and recovery approaches on AWS
- Testing recovery capabilities with AWS Backup

MIDAREL05: How do you plan for disaster recovery in manufacturing environments?

In manufacturing environments, production continuity is critical as downtime directly impacts operational efficiency and revenue. Manufacturing systems often involve specialized equipment, industrial control systems, and complex supply chains that require careful consideration for disaster recovery planning. Implementing a robust disaster recovery (DR) strategy verifies that manufacturing operations can recover quickly from disruptions.

Best practices

- MIDAREL05-BP01 Design a multi-Region disaster recovery strategy
- MIDAREL05-BP02 Implement and regularly test DR procedures

MIDAREL05-BP01 Design a multi-Region disaster recovery strategy

Manufacturing operations require high availability of systems that control production lines, inventory management, and supply chain logistics. A multi-Region disaster recovery strategy

means that if one Region experiences an outage, manufacturing systems can continue operating from another region with minimal disruption.

Desired outcome: A resilient manufacturing environment that can quickly recover from regional failures, which improves continuous production capabilities, maintains access to critical manufacturing data, and minimizes operational downtime.

Benefits of establishing this best practice:

- Reduced production downtime during Regional outages.
- Protection of critical manufacturing data and systems.
- Maintained supply chain continuity.
- Improved compliance posture with industry regulations and customer SLAs.
- Enhanced business resilience against natural disasters or large-scale infrastructure failures.

Level of risk exposed if this best practice is not established: High

Implementation guidance

- **Implement data replication across Regions:** Set up continuous replication of manufacturing data, including ERP and MES systems and production databases using Amazon RDS Multi-AZ with cross-Region read replicas or Amazon S3 Cross-Region Replication.
- **Create automated recovery procedures:** Develop AWS CloudFormation templates or use AWS Elastic Disaster Recovery to automate the recovery of manufacturing applications and infrastructure components with predefined RTOs and RPOs.
- **Establish production monitoring and failover mechanisms:** Implement Route 53 health checks and failover routing policies to automatically redirect traffic to backup manufacturing systems in case of primary system failure.
- **Test DR procedures regularly:** Schedule periodic disaster recovery tests using AWS Fault Injection Service to validate that manufacturing operations can be recovered within defined time frames and that all production-critical systems function properly after recovery.

Key AWS services

- AWS Elastic Disaster Recovery
- Amazon S3 Cross-Region Replication

- Amazon RDS Multi-AZ and Cross-Region Read Replicas
- AWS CloudFormation
- Amazon Route 53
- AWS Fault Injection Service

Resources

- Getting Started with AWS Elastic Disaster Recovery
- Cross-Region Replication for Manufacturing Workloads
- Disaster Recovery Options in the Cloud
- AWS Manufacturing and Industrial Reference Architectures

MIDAREL05-BP02 Implement and regularly test DR procedures

Regularly test and review your disaster recovery procedures to verify they work as expected when needed. Manufacturing environments have unique requirements including machine connectivity, production data integrity, and maintaining operational technology systems, which makes systematic testing critical to validate recovery capabilities.

Desired outcome: Successfully validated and regularly updated disaster recovery procedures that can be executed with confidence during actual incidents, minimizing downtime and resuming manufacturing operations within defined RTOs and RPOs.

Benefits of establishing this best practice:

- Regular testing builds confidence in recovery procedures
- Identifies gaps before real incidents occur
- Reduces recovery time during actual disasters
- Helps meet compliance requirements for business continuity in manufacturing environments
- Maintains currency of procedures as manufacturing systems evolve
- Aligns with changing production requirements
- Validates recovery capabilities across seasonal manufacturing patterns

Level of risk exposed if this best practice is not established: High

Implementation guidance

Establish regular review cycles: First document all components of your manufacturing disaster recovery plan including systems, data, and processes. Map changes in production environments that could impact recovery procedures. Define review participants from operations, IT, and business teams. Identify regulatory requirements that necessitate periodic review and testing. For implementation, create a structured review process that evaluates recovery plans against current manufacturing operations. Design evaluation criteria that validate recovery procedures remain aligned with business needs. Consider implementing AWS Backup for manufacturing data and systems with review cycles that align with your quarterly assessment schedule.

Conduct systematic testing: Begin by defining test scenarios that reflect realistic disaster situations in your manufacturing environment. Document test scope and success criteria for different recovery scenarios. Establish test schedules that align with production maintenance windows. Map dependencies between systems to test them comprehensively. For implementation, create testing protocols that validate all aspects of recovery procedures. Design testing environments that accurately reflect production configurations. Consider deploying AWS Elastic Disaster Recovery to test replication of critical manufacturing systems with minimal RPO and enable rapid recovery validation.

Configure automated health checks: Start by identifying critical components requiring continuous monitoring. Document acceptable performance thresholds and recovery triggers. Define escalation procedures for different types of failures. Establish monitoring metrics that indicate recovery readiness. For implementation, create automated monitoring systems that continuously validate recovery capability. Design health checks that verify system and data availability. Consider using Route 53 health checks and Application Recovery Controller to automate failover testing of manufacturing applications.

Create automated recovery scripts: Document all critical manufacturing systems and their recovery requirements. Map dependencies between systems to establish proper recovery sequence. Define acceptance criteria for successful recovery and establish test scenarios that reflect realistic failure modes. Identify required permissions and access controls for recovery operations.

For implementation, create automated recovery procedures that can consistently restore manufacturing systems. Design recovery orchestration that maintains data integrity and system dependencies.

Consider developing AWS CloudFormation templates or AWS CDK scripts that can automatically recreate your manufacturing infrastructure, including device connections and monitoring systems.

Schedule regular recovery testing: Identify testing windows that minimize impact on production operations. Document test scope and success criteria for different scenarios. Establish clear roles and responsibilities during recovery tests. Define test frequency based on system criticality and compliance requirements.

For implementation, create a systematic testing program that validates recovery procedures across different failure scenarios. Design testing protocols that simulate various disruption types while maintaining safety and operational controls.

Consider using AWS Fault Injection Service to simulate failures in your manufacturing workloads and practice recovery procedures at least quarterly.

Implement cross-Region backup testing: Map your geographic redundancy requirements and identify critical systems requiring cross-Region recovery. Document regional compliance requirements and data sovereignty constraints. Define acceptable recovery time frames for cross-Region restoration.

For implementation, create backup verification procedures that provide data availability across regions. Design testing mechanisms that validate cross-Region recovery capabilities.

Consider configuring AWS Backup to regularly test restore operations of critical manufacturing data across multiple AWS Regions.

Establish recovery monitoring: Define key metrics for measuring recovery effectiveness. Document monitoring requirements during recovery operations. Establish thresholds for recovery success and failure conditions.

For implementation, create monitoring systems that track recovery progress and success rates. Design dashboards that provide clear visibility into recovery operations.

Consider using Amazon CloudWatch to track recovery time and success rates during recovery tests, creating dashboards that highlight areas needing improvement.

Implement improvement processes: Begin by establishing methods to capture lessons learned from each test cycle. Document required updates to recovery procedures based on test results. Define change management processes for updating recovery plans. Create training requirements for updated procedures. For implementation, create systematic processes to incorporate test findings into recovery plans. Design validation procedures for updated recovery methods. Consider conducting quarterly tabletop exercises using AWS Fault Injection Service to validate recovery procedures and team readiness

Key AWS services

- AWS Backup
- AWS CloudFormation
- Amazon CloudWatch
- AWS Fault Injection Service
- AWS Systems Manager
- AWS Elastic Diater Recovery

Resources

- Disaster Recovery of Workloads on AWS
- Managing AWS Fault Injection Service experiments
- AWS Backup Documentation

Performance efficiency

The performance efficiency pillar includes the ability to use cloud resources efficiently to meet performance requirements, and to maintain that efficiency as demand changes and technologies evolve. This component establishes quantifiable metrics and standardized methodologies to achieve optimal throughput-to-cost ratios across the entire data infrastructure to meet performance goals. By continuously benchmarking workload performance against established baselines, users can identify and reduce computational bottlenecks that directly impact cycle times in production data processing.

Resource optimization methodologies applied to the data architecture mirror lean manufacturing principles, enabling precise allocation of computing power, memory, and storage based on workload demands rather than peak capacity requirements. This data-driven approach facilitates maximum utilization of available resources while maintaining response time SLAs critical to production decision support. The performance monitoring framework serves as a digital dashboard for system efficiency, providing real-time visibility into resource consumption patterns and enabling predictive capacity planning to help prevent processing delays that could impact production schedules.

Topics

- Design principles
- Real-time data access
- Event-driven architecture
- Data processing infrastructure
- Troubleshoot and optimize pipelines
- Pre-process data at edge
- Data storage organization

Design principles

Take a data-driven approach to build a high-performing architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types.

In addition to the design principles described in the performance efficiency pillar of the AWS Well-Architected Framework whitepaper, the following design principles can help manufacturing

customers build and operate efficient and performant workloads that meet business performance requirements and allowing growth without sacrificing performance at great cost.

- **Real-time performance visibility:** Implement comprehensive performance dashboards that capture sub-second manufacturing metrics. Deploy predictive performance analysis using machine learning to anticipate workload patterns and automatically optimize resource allocation based on specific processing requirements.
- **Automated performance tuning:** Use AI-driven optimization to continuously evaluate and adjust resource configurations against actual performance requirements. Implement self-tuning database configurations that automatically adapt to changing query patterns while maintaining consistent sub-10ms response times.
- **Continuous performance testing:** Conduct regular synthetic transaction testing to establish baseline latency metrics and evaluate production performance against these standards. Implement multi-dimensional performance monitoring across compute, network, storage, and database layers to identify specific optimization opportunities.
- **Parallel processing architecture**: Design high-concurrency processing architectures to increase data ingestion rates by up to 120% compared to synchronous approaches. Implement standardized data models that enable in-memory parallel processing, achieving up to 9x faster analytics than sequential processing. Verify that the architecture maintains sub-second latency even under 10x burst loads.
- **Event-driven performance architecture:** Adopt reactive processing patterns to reduce processing latency by up to 78% compared to polling-based approaches. Implement high-performance message handling with ordered processing queues for time-sensitive manufacturing data. This approach typically achieves 45-65% higher throughput with 30% lower latency compared to traditional request-response models. For more detail and metrics calculations, see Demystifying event-driven architecture in modern distributed systems.
- **Edge computing for performance:** Deploy critical workloads to edge locations to reduce round-trip latency by up to 98% for time-sensitive control systems. Process high-frequency sensor data at the edge to enable 10kHz+ sampling rates with local feedback loops while maintaining cloud analytics integration. This typically reduces end-to-end latency from 120-200ms to 5-12ms in manufacturing control scenarios. For more detail and metrics calculations, see Demystifying event-driven architecture in modern distributed systems.
- **Data reduction strategy:** Implement data condensing, summarizing, or compression before transmission to reduce network traffic and storage costs. Consider appropriate sampling rates based on actual business needs. For example, converting per-second temperature readings to

5, 10, or 30 minute averages when appropriate. Only maintain high-frequency data collection where specifically justified by business requirements.

- **Performance-optimized data storage:** Implement a multi-tier storage architecture that can handle 250,000+ reads per second for recent operational data with automatic archival of historical data. Use in-memory caching for frequently accessed production metrics to achieve microsecond-level response times with high availability. Select purpose-built databases for specific workloads to achieve up to 40x faster analytics compared to general-purpose solutions.

Real-time data access

MIDAPERF01: How does your architecture support high-performance access to real-time data?

Modern manufacturing architectures must prioritize high-performance access to real-time data to maintain competitive advantage in production environments. The strategic implementation of distributed processing systems and optimized database structures enables plant operators to make data-driven decisions with minimal latency, directly impacting overall equipment effectiveness and throughput.

Best practices

- MIDAPERF01-BP01 Use time series database for real-time analytics and data lake for long-term storage
- MIDAPERF01-BP02 Compress, sample and summarize data at edge, before sending to the cloud environment

MIDAPERF01-BP01 Use time series database for real-time analytics and data lake for long-term storage

In manufacturing environments, access patterns for operational data vary significantly based on data age. Current data requires high-performance, low-latency access for real-time decision making, while historical data typically serves longer-term analysis with less stringent performance requirements. Implementing a tiered storage architecture with time series databases for recent data and data lakes for historical information helps optimize both performance and cost.

Desired outcome: A multi-tiered data storage architecture that provides millisecond-level query performance for recent manufacturing data while cost-effectively storing and enabling analytics on historical data spanning months or years, with appropriate retention policies and data lifecycle management.

Common anti-patterns:

- Using only one type of database (relational or noSQL) for both real-time operational data and years of historical data
- Keeping years of manufacturing data in high-performance databases designed for real-time access
- Attempting to store millisecond-level sensor data in traditional RDBMs without proper optimization
- Storing data without logical separation by time, production line, or equipment type, leading to full table scans
- Relying on manual intervention to move aging data between storage tiers
- Allowing unlimited data accumulation in high-performance storage without lifecycle rules
- Moving data to archival storage too quickly before operational teams have adequate access for troubleshooting
- Moving data between tiers without optimizing format, compression, or structure for the target storage
- Running long-term trend analysis queries against time series databases optimized for recent data
- Forcing applications to know and manage which storage system contains the data they need
- Using generic database indexes instead of time-series optimized indexing for temporal queries
- Attempting to join data across time series databases and data lakes in real-time queries
- Requiring each application to integrate separately with time series databases and data lakes
- No unified query interface, forcing users to learn different query languages and APIs for current versus Historical data
- Synchronous data migration, blocking real-time operations while moving data between storage tiers
- Direct storage access from applications, allowing applications to directly query storage systems without abstraction layers
- Inadequate tagging and dimensions, storing time series data without proper metadata tags for equipment, location, or process context

- Row-based storage for analytics, using row-oriented formats in data lakes when columnar formats would provide better compression and query performance
- Normalized schemas for time series, applying traditional database normalization to high-frequency sensor data
- Using one-size-fits-all schemas instead of optimizing for specific manufacturing data patterns

Benefits of establishing this best practice:

- Delivers sub-second dashboard response times for real-time operational monitoring
- Reduces query costs for frequent access to current production metrics
- Enables cost-effective long-term storage of complete manufacturing history
- Optimizes storage costs by matching data access patterns to appropriate technologies
- Supports both real-time alerting and historical trend analysis from the same dataset

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Implement time series database layer**:** Deploy Amazon Timestream, a purpose-built time series database optimized for industrial IoT data from manufacturing equipment like CNC machines, conveyor systems, and temperature sensors. Configure retention policies (typically 30-90 days in memory store, longer periods in magnetic store) based on operational requirements such as real-time quality control monitoring. Design efficient data models with appropriate tags (equipment_id, production_line, and shift) and dimensions (temperature, pressure, and vibration) to support common manufacturing queries like equipment performance analysis and predictive maintenance alerts.

Establish data lake architecture: Create an Amazon S3-based data lake with appropriate partitioning strategies (by date=2024/01/15, production_line=assembly_1, product_type=automotive_parts) to optimize query performance on historical manufacturing data. Implement Apache Parquet columnar storage format to improve compression and query efficiency for manufacturing analytics such as Overall Equipment Effectiveness (OEE) calculations, quality trend analysis, and production optimization studies across multiple factories.

Configure data lifecycle management: Develop AWS Lambda functions triggered by Amazon EventBridge to automatically migrate data from Amazon Timestream to S3 as it ages beyond immediate operational relevance (for example, after 90 days). Use AWS Glue ETL jobs to

implement data transformation during migration, converting real-time sensor data into optimized Parquet format and aggregating metrics for long-term analytics like annual production trends and equipment lifecycle analysis.

Design unified query interface: Create a query abstraction layer using Amazon Athena for historical data analysis and Amazon Timestream Query for real-time operational data, with Amazon API Gateway providing a unified REST interface. Implement intelligent routing logic using AWS Lambda that directs queries to Timestream for recent data (last 30 days for live production monitoring) and to Athena for historical analysis (older data for quarterly performance reviews and compliance reporting), verifying that manufacturing engineers and analysts can access data consistently regardless of storage location.

Key AWS services

- Amazon Timestream for time series data storage
- Amazon S3 for data lake foundation
- AWS Glue for data transformation and cataloging
- Amazon Athena for querying historical data
- AWS Lambda for lifecycle management automation

Resources

- Amazon Timestream: Purpose-built time series database
- Guidance for Data Lakes on AWS
- Time Series Forecasting Principles with Amazon Forecast

MIDAPERF01-BP02 Compress, sample and summarize data at edge, before sending to the cloud environment

In manufacturing environments, IoT devices and sensors often generate massive volumes of high-frequency data that can overwhelm networking, processing, and storage resources.

Desired outcome: Reduce amount of data flowing from on-premises to cloud by summarizing time series machine data, for example average temperature over a time period instead of raw temperature values every second. This allows for quicker data processing long term for trending insights.

Common anti-patterns:

- Sending raw, unprocessed data streams directly to the cloud
- Ignoring data compression opportunities
- Not implementing edge-level data processing
- Overwhelming network bandwidth with high-frequency data

- Creating unnecessary network congestion
- Ignoring data transmission timing optimization
- Bypassing gateway-level processing capabilities
- Not using MQTT topic filtering
- Skipping data summarization strategies

Benefits of establishing this best practice:

- Less processing time processing and querying long term data
- Less time transmitting data to cloud
- Reduced storage costs
- Reduced network congestion to cloud

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

To reduce network traffic and overhead to allow faster processing:

1. Configure small data processing applications to summarize data on your gateway devices using AWS IoT Greengrass components
   a. **Manufacturing example:** Deploy AWS IoT Greengrass on factory floor gateways to run edge analytics components that process real-time data from CNC machines, conveyor belt sensors, and quality control cameras, summarizing production metrics like throughput rates, defect counts, and equipment utilization before sending to the cloud.
2. Subscribe to direct topics through MQTT of machine data, then use components to summarize and re-publish data on a new topic that is routed to AWS IoT Core or SiteWise.

   a. **Industrial example:** Use AWS IoT Greengrass components to subscribe to MQTT topics from industrial equipment like turbines, pumps, and generators, then aggregate temperature, vibration, and pressure readings into health score summaries that are republished to AWS IoT SiteWise for asset monitoring dashboards and AWS IoT Core for further processing and alerting.

3. Alternatively, locally compress summarized data into Apache Parquet format and transfer directly to Amazon S3.
   a. **Manufacturing example:** Configure edge devices in automotive plants to compress daily production data (part counts, cycle times, energy consumption) from assembly line robots and quality inspection systems into Parquet files, then batch upload to Amazon S3 for long-term storage and analysis with AWS analytics services like Amazon Athena and Quick for operational intelligence reporting.

Implementation Steps

1. Create data processing component in your language of choice, using the AWS IoT Greengrass Development Kit (GDK).
2. Have component subscribe to raw data topics on-premises.
3. Build components to do tasks such as summarize data and rolling averages for set time periods, and re-publish to new topic.
4. Relay only new topic from on premises to AWS IoT Core or SiteWise for storage and processing.

Key AWS services

- AWS IoT Core
- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS IoT SiteWise Edge

Resources

- Cost-effectively ingest IoT data directly into Amazon S3 using AWS IoT Greengrass
- Ingest and analyze equipment data in the cloud
- Getting Started with AWS IoT Greengrass Solution Accelerators for Edge Computing

Event-driven architecture

MIDAPERF02: How do integrate event-driven architecture into your data processing system?

Operations benefit from event-driven architectures that can immediately initiate responses to production anomalies without continuous polling of equipment status. Plant managers implementing these systems typically see significant improvements in reaction times to critical events, as data processing occurs precisely when needed rather than in scheduled batches that may miss time-sensitive production issues.

Best practices

- MIDAPERF02-BP01 Implement event-driven architectures for manufacturing systems
- MIDAPERF02-BP02 Use historical cloud usage data aligned with production schedules and business forecasts

MIDAPERF02-BP01 Implement event-driven architectures for manufacturing systems

In manufacturing environments, operational data is generated based on specific events such as equipment state changes, threshold violations, or production milestones. Implementing event-driven architectures allows systems to respond efficiently to these events rather than constantly polling for changes, significantly improving resource utilization and system responsiveness. This approach aligns perfectly with IoT communication patterns while enabling scalable, loosely-coupled manufacturing systems.

Desired outcome: A responsive, efficient manufacturing data architecture that processes information only when meaningful events occur, reducing unnecessary computation, minimizing latency for critical operations, and enabling dynamic scaling based on actual processing demand rather than peak capacity requirements.

Common anti-patterns:

- Transforming all incoming manufacturing data immediately instead of lazy evaluation when needed
- Making multiple small database calls per event instead of batching operations or using bulk APIs

- Processing all events and filtering in application code rather than using message-level filtering capabilities
- Routing all events from similar equipment to the same partition, creating processing bottlenecks
- Creating point-to-point integrations between manufacturing systems instead of using event mediators
- Making blocking calls between manufacturing subsystems instead of asynchronous event-driven communication
- Processing events without validating structure, leading to runtime failures and data corruption
- Building event consumers that depend on specific event producer implementations rather than standardized interfaces
- Allowing event processing failures to occur without proper logging, alerting, or dead letter handling
- Not implementing flow control when downstream systems cannot keep up with event volume
- Failing to implement end-to-end tracing for manufacturing processes spanning multiple event handlers
- Only monitoring for failures instead of proactively tracking performance metrics and trends

Benefits of establishing this best practice:

- Reduces processing overhead by 40-60% compared to polling-based systems
- Improves response time to critical manufacturing events by removing processing queues
- Enhances system scalability by allocating resources only when needed for event processing
- Simplifies integration between manufacturing subsystems through standardized event interfaces
- Enables more granular cost allocation by associating resource usage with specific event types

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

1. Implement a publish or subscribe messaging architecture where manufacturing devices and systems publish events to centralized topics. Configure consumers to process only relevant event types using message filtering capabilities to reduce unnecessary processing.
2. Deploy durable message queues between producers and consumers to handle throughput spikes and provide reliable event delivery even during processing backlogs or temporary downstream

system outages common in manufacturing environments. AWS SQS and Amazon EventBridge services are tools that can accomplish these goals.

3. Design event handlers with idempotency in mind to help prevent duplicate processing when events are retried. Implement deduplication mechanisms using event IDs or processing timestamps to maintain data integrity during retries. A
4. Establish dead-letter queues to capture events that cannot be processed successfully after multiple attempts. Implement automated monitoring and alerting for these queues to quickly identify and resolve processing issues that could impact manufacturing operations. AWS Step functions, Amazon EventBridge, and AWS IoT core are example services to help accomplish these tasks.
5. For multi-step manufacturing processes, implement state machines to coordinate event sequences and manage process state. Design workflows that can handle long-running operations while maintaining visibility into process status. AWS Step functions, Amazon EventBridge, and AWS IoT core are example services to help accomplish these tasks.

Key AWS services

- Amazon EventBridge for event routing and filtering
- Amazon SQS for reliable message queueing
- AWS Lambda for serverless event processing
- Amazon SNS for event notifications
- AWS Step Functions for manufacturing process orchestration
- AWS IoT Core for device-generated events

Resources

Related documents:

- Building Event-Driven Architectures on AWS
- Serverless Patterns for Event-Driven Architectures
- Implementing Idempotency Patterns with AWS Lambda
- Handling Failure Scenarios with Amazon SQS Dead-Letter Queues
- Build a serverless Amazon Bedrock batch job orchestration workflow using AWS Step Functions

MIDAPERF02-BP02 Use historical cloud usage data aligned with production schedules and business forecasts

Aligning cloud resource allocation with production schedules and business forecasts enables organizations to optimize system performance during critical periods while helping prevent resource constraints that could impact throughput and quality. By analyzing patterns in historical cloud usage alongside manufacturing cycles, plant managers can anticipate processing requirements for data-intensive operations like quality inspection systems, predictive maintenance algorithms, and real-time production monitoring, providing optimal performance when manufacturing demands are highest.

Desired outcome: A predictive resource management approach that provides manufacturing systems with precisely calibrated computing capacity, removing performance bottlenecks during peak production periods while maintaining processing responsiveness for time-sensitive manufacturing analytics and control systems.

Common anti-patterns:

- Using static capacity planning without considering manufacturing cycles, seasonal demands, or planned maintenance windows
- Analyzing cloud usage data in isolation without correlating with production schedules, quality metrics, or business forecasts
- Applying uniform auto-scaling rules across all manufacturing workloads regardless of their specific performance characteristics
- Triggering resource scaling exactly when demand increases without accounting for provisioning and initialization delays
- Using only basic CPU/memory metrics for scaling decisions without considering manufacturing-specific performance indicators
- Running large ETL jobs or analytics workloads during active production periods, competing for resources with real-time systems
- Processing all manufacturing data synchronously, even for non-time-sensitive analytics
- Retaining all historical data at the same performance tier regardless of access patterns
- Setting static performance thresholds that don't account for normal variations in manufacturing operations
- Monitoring individual components without understanding overall system performance impact on manufacturing processes

- Failing to establish and maintain performance baselines for different production scenarios
- Running development, testing, and production workloads on shared infrastructure during critical manufacturing periods
- Not considering latency between cloud resources and manufacturing equipment locations when designing system architecture
- Failing to properly tag resources to correlate performance investments with specific manufacturing outcomes and ROI

Benefits of establishing this best practice:

- Improves production system responsiveness by up to 40% during peak manufacturing periods
- Removes data processing bottlenecks that can cause manufacturing quality or throughput issues
- Enables higher-fidelity monitoring and analytics during critical production runs
- Accelerates time-to-insight for manufacturing intelligence during complex production sequences
- Facilitates seamless integration between production scheduling and infrastructure provisioning teams

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Analyze cloud resource utilization patterns during different manufacturing operations to identify performance-critical periods requiring enhanced computing capacity, especially for vision systems, complex analytics, or high-frequency data collection. Analysis and alerting can be implanted through using Amazon CloudWatch, AWS X-Ray, and Amazon CloudWatch insights.
- Establish relationships between specific manufacturing activities (high-precision runs, quality inspections, material changeovers) and corresponding infrastructure performance requirements to develop predictive capacity models.
- Develop automated scaling mechanisms that proactively adjust computing resources based on upcoming production schedules, which verifies that critical systems have sufficient processing power before high-demand manufacturing phases begin. Using services such as Amazon SageMaker AI for predictive modeling, auto scaling with AWS Auto Scaling, and Amazon CloudWatch for monitoring metrics can help with implementation.
- Refine ETL processes and analytics workflows based on historical performance data to maximize throughput during peak production periods when real-time insights are most valuable. AWS

services such as Amazon Kinesis Data Streams, Amazon MSK, and AWS IoT Core can help with implementing optimized data processing pipelines. Real time processing can be implemented through Lambda, and Amazon Kinesis Data Analytics. AWS X-Ray can help with end to end pipeline tracking and anomaly detection.

- Implement continuous performance monitoring that compares actual versus expected response times and processing capabilities, refining resource allocation models to improve manufacturing system responsiveness over time. AWS services that can help with implementation are Amazon CloudWatch, AWS X-Ray, and Application Load Balancer.

Key AWS services

- Amazon CloudWatch for performance monitoring and metrics collection
- AWS Auto Scaling for automatically adjusting capacity based on production needs
- AWS Forecast for predicting resource requirements based on historical patterns
- Amazon Kinesis for managing high-throughput data streams from manufacturing equipment
- AWS Lambda for dynamic processing of production event data
- Amazon RDS Performance Insights for database performance optimization

Resources

- Performance Efficiency Pillar - AWS Well-Architected Framework
- Implementing Predictive Scaling with AWS Auto Scaling
- Real-time Analytics with Amazon Kinesis
- Optimizing AWS Lambda Performance for Manufacturing Workloads

Data processing infrastructure

MIDAPERF03: How do you measure performance of infrastructure that processes your data?

Effective manufacturing infrastructure performance is measured through comprehensive metrics that track both throughput capacity and processing latency across the production data pipeline.

Plant managers typically establish baseline KPIs for data processing systems, monitoring factors such as equipment data capture rates, analytics processing times, and dashboard refresh rates to base operational decisions on timely information.

Best practices

- MIDAPERF03-BP01 Use cloud observability tools for manufacturing systems
- MIDAPERF03-BP02 Implement comprehensive performance measurement for manufacturing data infrastructure

MIDAPERF03-BP01 Use cloud observability tools for manufacturing systems

In manufacturing environments, comprehensive visibility into system performance, data processing pipelines, and infrastructure health is critical for maintaining operational excellence. Implementing cloud-native observability tools provides unified monitoring, actionable alerting, and diagnostic capabilities across the entire manufacturing technology stack. This integrated approach enables rapid identification and resolution of performance issues before they impact production operations.

Desired outcome: A comprehensive observability framework that provides real-time visibility into manufacturing data systems, enabling proactive performance optimization, rapid troubleshooting, and data-driven capacity planning while providing maximum uptime for critical manufacturing operations.

Common anti-patterns:

- Waiting for system failures before implementing monitoring instead of proactive performance tracking
- Collecting massive amounts of data without establishing manufacturing-specific KPIs or business relevance
- Only monitoring at infrastructure level while ignoring application and business process performance
- Setting too many low-priority alerts or poorly tuned thresholds that create noise instead of actionable insights
- Using disconnected monitoring solutions that prevent correlation across system components
- Operating manufacturing systems without comprehensive API activity logging and analysis

- Ignoring API call patterns and frequency that could indicate over-utilization or inefficient integrations
- Storing critical operational logs in systems where they can be modified, compromising audit trails
- Mixing production logs with development/testing data instead of maintaining dedicated audit accounts
- Manually configuring edge devices instead of using standardized fleet management approaches
- Pushing configuration changes directly to all devices without canary or blue/green deployment strategies
- Operating edge devices without hardware performance monitoring and health tracking
- Relying on human intervention for common device communication issues instead of automated remediation
- Collecting metrics without establishing appropriate performance thresholds for manufacturing-critical systems
- Using only critical alerts instead of progressive severity levels based on threshold proximity
- Creating alerts that don't run automated remediation or clear escalation procedures
- Learning about performance issues only after they affect manufacturing operations
- Operating manufacturing data flows without end-to-end visibility into processing steps and dependencies
- Either over-sampling (performance impact) or under-sampling (missing critical issues) distributed traces
- Running manufacturing systems without understanding request flow timing and bottleneck identification

Benefits of establishing this best practice:

1. Reduces MTTD for performance anomalies by 65-85%
2. Enables correlation of issues across different system components for faster root cause analysis
3. Provides quantifiable metrics to justify optimization investments and measure their impact
4. Enhances capacity planning through historical performance trend analysis
5. Minimizes production impact through early detection of emerging performance bottlenecks

Level of risk exposed if this best practice is not established: High

Implementation guidance

Deploy AWS CloudTrail across all manufacturing data services to capture API calls from SCADA systems, MES integrations, and data pipelines. Store logs in a dedicated Amazon S3 bucket with S3 Object Lock for immutability. Use AWS CloudWatch Insights to analyze patterns in equipment data ingestion rates and identify potential security issues in your OT/IT bridge connections. Configure AWS Organizations to centralize trail management across production, staging, and development environments.

Implement Amazon CloudWatch as your unified metrics system, capturing infrastructure metrics from EC2 instances running historian services, custom manufacturing metrics using CloudWatch Custom Metrics for OEE, throughput, and quality indicators, application performance from containerized services using Amazon ECS/EKS with Container Insights, and business process metrics through CloudWatch Embedded Metric Format in your Lambda functions processing production data. Use Amazon Managed Service for Prometheus for time-series data from edge devices and Amazon Managed Grafana for manufacturing dashboards.

Configure CloudWatch Alarms with manufacturing-specific thresholds (for example, data ingestion gaps indicating equipment downtime). Implement progressive alerting using Amazon SNS topics with different severity levels: Critical for production line stoppage detection, Warning for trending toward SLA violations, and Info for planned maintenance windows. Use AWS Lambda functions triggered by CloudWatch Events for automated remediation, such as restarting stuck data collection services or switching to backup data sources.

Deploy AWS X-Ray across your manufacturing data pipeline to trace requests from edge devices through AWS IoT Core, Kinesis Data Streams, Lambda processing functions, and final storage in Amazon Timestream or S3. Configure sampling rules to capture 100% of critical production data flows while sampling routine maintenance data at lower rates. Use X-Ray service maps to visualize dependencies between your MES, ERP, and analytics systems.

Use AWS IoT Device Management to manage your industrial edge devices and gateways. Deploy AWS IoT Greengrass for edge computing capabilities. Implement fleet-wide updates using IoT Jobs with controlled rollout strategies by using IoT Device Management Fleet Indexing to group devices by production line or equipment type, configuring progressive deployment patterns with canary releases to test configuration changes on non-critical equipment first, and monitoring deployment success rates with automatic rollback of failed updates.

Configure AWS IoT Events to detect offline devices, abnormal sensor readings, or communication pattern anomalies. Set up AWS IoT Device Defender for security monitoring of your industrial

devices. Create automated recovery procedures using AWS Step Functions to orchestrate device troubleshooting workflows, AWS Systems Manager to remotely diagnose and restart edge gateway services, and Amazon SNS notifications to operations teams when manual intervention is required.

Implement intelligent data retention using Amazon S3 Intelligent Tiering for historical manufacturing data, S3 Lifecycle policies to transition detailed sensor data from Standard to IA to Glacier based on access patterns, Amazon Timestream with automatic data tiering for time-series data (memory for recent data, magnetic storage for historical), and CloudWatch Logs retention policies configured by criticality (30 days for debug logs, one year for production events). Use AWS Cost Explorer and AWS Budgets to monitor storage costs and set alerts for unexpected data growth. Consider Amazon Redshift with automatic table optimization for long-term analytics on production trends while maintaining cost efficiency through Reserved Instance planning for predictable workloads.

Key AWS services

- AWS CloudTrail for API activity monitoring
- Amazon CloudWatch for metrics, logs, and alerting
- AWS X-Ray for distributed tracing
- AWS IoT Greengrass for edge device management
- AWS Systems Manager for configuration management
- AWS IoT Events for device state monitoring

Resources

- Monitoring AWS IoT Applications
- Analyzing API Calls with CloudTrail
- AWS X-Ray features
- AWS IoT Device Management features
- AWS Solutions Library

MIDAPERF03-BP02 Implement comprehensive performance measurement for manufacturing data infrastructure

Understanding the performance characteristics of data processing infrastructure is essential for maintaining performance efficiency and planning for growth. Implementing robust measurement frameworks with appropriate metrics, dashboards, and alerting enables organizations to proactively manage performance, optimize resource utilization, and justify infrastructure investments with quantifiable data.

Desired outcome: A comprehensive performance measurement framework that provides real-time visibility into all aspects of manufacturing data infrastructure, enabling data-driven optimization decisions, capacity planning, and early detection of performance degradation before it impacts production operations.

Common anti-patterns:

- Waiting for system failures or user complaints before investigating performance issues instead of implementing proactive monitoring and alerting
- Using separate, disconnected monitoring tools for different infrastructure components without centralized observability and correlation
- Creating technical dashboards with standard IT metrics that don't relate to manufacturing operations or business context
- Setting fixed performance thresholds that don't account for normal manufacturing workload variations and production cycle patterns
- Deploying monitoring without first conducting controlled testing to understand normal performance characteristics
- Failing to track API usage patterns, leading to undetected redundant calls, inefficient integrations, and quota exhaustion
- Monitoring only basic system metrics while ignoring manufacturing-specific measurements like message throughput by device type or processing latency for time-sensitive data
- Configuring too many low-priority alerts or alerts without clear escalation paths, leading to ignored notifications
- Not retaining sufficient performance history for trend analysis and capacity planning decisions
- Implementing only critical threshold alerts without predictive or warning-level notifications for emerging performance trends

- Requiring human intervention for common, predictable performance issues that could be automatically resolved
- Operating without granular resource utilization tracking, making it impossible to allocate costs by workload or justify optimization investments
- Not implementing distributed tracing to understand end-to-end data processing delays across manufacturing workflows
- Skipping controlled performance validation during infrastructure changes or capacity planning exercises

Benefits of establishing this best practice:

1. Enables proactive identification of performance bottlenecks before they impact production
2. Provides quantifiable metrics to justify optimization investments and infrastructure scaling
3. Facilitates accurate capacity planning based on historical performance trends
4. Improves cost allocation through precise measurement of resource utilization by workload
5. Reduces troubleshooting time by pinpointing specific performance constraints

Level of risk exposed if this best practice is not established: High

Implementation guidance

1. Deploy Amazon CloudWatch agents on EC2 instances and integrate with AWS IoT Core device metrics to collect comprehensive performance data across the manufacturing infrastructure components. Configure CloudWatch Custom Metrics specific to industrial data processing needs alongside standard system metrics through AWS Systems Manager and Amazon Kinesis Data Streams.
2. Key metrics to consider:
    a. Message throughput rates (messages per second) by device type and production area using AWS IoT Device Management groupings
    b. Data storage utilization trends with forecasted growth patterns via Amazon S3 Storage Lens and Amazon EBS monitoring
    c. Bandwidth consumption during different production phases through VPC Flow Logs and AWS Direct Connect monitoring

   d. Processing latency for time-sensitive manufacturing data flows using Amazon Kinesis Analytics and AWS Lambda duration metrics
   e. API call volumes and patterns with service quota utilization percentages through AWS CloudTrail and Service Quotas integration
3. Establish a unified monitoring environment using Amazon CloudWatch Dashboards and AWS Grafana that aggregates metrics from the infrastructure components. Create manufacturing-specific dashboards using Quick that visualize performance metrics in the context of production operations rather than just technical indicators, integrated with AWS IoT SiteWise for operational technology data correlation.
4. Enable comprehensive API activity logging through AWS CloudTrail and Amazon API Gateway access logging to track service usage patterns. Configure Amazon CloudWatch Insights and AWS X-Ray to identify redundant or inefficient API calls that could impact performance or exceed service quotas, with cost optimization insights from AWS Cost Explorer API usage analysis.
5. Conduct controlled performance testing using AWS Load Testing Solution and Amazon CloudWatch Synthetics to establish baseline metrics for normal operations. Configure dynamic thresholds using CloudWatch Anomaly Detection based on these baselines to account for expected variations in manufacturing workloads, leveraging Amazon Forecast for predictive baseline modeling.
6. Design a multi-tiered alerting strategy using Amazon CloudWatch Alarms with Amazon SNS notifications and predictive alerts through CloudWatch Anomaly Detection that identify concerning trends before they reach critical thresholds. Implement automated remediation using AWS Systems Manager Automation, AWS Lambda functions, and Amazon EventBridge rules for common performance issues to minimize human intervention in manufacturing operations.

Key AWS services

- Amazon CloudWatch for metrics collection and visualization
- AWS X-Ray for distributed tracing and latency analysis
- AWS CloudTrail for API activity monitoring
- AWS Compute Optimizer for resource optimization recommendations
- Amazon Managed Service for Prometheus and Amazon Managed Grafana for advanced monitoring scenarios

Resources

- Monitoring AWS IoT Applications with CloudWatch
- Analyzing API Calls with CloudTrail
- AWS X-Ray: Use a console
- Service Quotas and Amazon CloudWatch alarms

Troubleshoot and optimize pipelines

MIDAPERF04: How do you troubleshoot and optimize your machine performance data ingestion and processing pipelines?

Manufacturing data pipelines require systematic troubleshooting protocols that isolate bottlenecks from sensor collection points through to visualization systems, providing production teams reliable access to equipment performance metrics. Advanced plants implement automated monitoring systems that continuously validate data quality and pipeline throughput, triggering alerts when latency increases or data gaps emerge, thereby maintaining the integrity of performance analytics used for critical production decisions.

Best practices

- MIDAPERF04-BP01 Implement end-to-end observability for manufacturing data pipelines
- MIDAPERF04-BP02 Decouple data ingestion from processing in manufacturing systems

MIDAPERF04-BP01 Implement end-to-end observability for manufacturing data pipelines

In manufacturing environments, comprehensive visibility into data processing and ingestion infrastructures is critical for maintaining operational excellence and providing timely delivery of production insights. Implementing robust observability solutions enables rapid identification and resolution of issues that could impact data quality, processing efficiency, or analytical outcomes.

Desired outcome: A fully observable data processing and ingestion infrastructure that provides immediate visibility into performance metrics, error conditions, and processing bottlenecks,

enabling teams to quickly troubleshoot issues, minimize downtime, and maintain reliable data flows that support critical manufacturing operations.

Common anti-patterns:

- Waiting for production teams to report data issues instead of proactive monitoring and alerting
- Using separate, disconnected monitoring tools that prevent correlation of issues across the entire data pipeline
- Implementing basic logging without contextual information like correlation IDs, production batch identifiers, or equipment-specific metadata
- Failing to trace data flows end-to-end through multi-stage processing pipelines, making bottleneck identification difficult
- Not monitoring edge gateways, industrial PCs, and on-premises servers that are critical points of failure
- Failing to track API interactions, retries, throttling, and authentication failures that can silently degrade pipeline performance
- Either over-sampling (causing performance overhead) or under-sampling (missing critical performance insights) in trace collection
- Inability to correlate symptoms across distributed manufacturing systems during troubleshooting
- Relying on manual detection of data quality issues, latency increases, or data gaps instead of automated monitoring
- Not configuring appropriate logging verbosity or failing to enhance detail levels during active troubleshooting scenarios
- Tracking only technical metrics without correlating to manufacturing KPIs like production batch status or equipment performance
- Lacking systematic protocols that isolate bottlenecks from sensor collection through visualization systems

Benefits of establishing this best practice:

1. Reduces mean time to identification and resolution for data pipeline issues
2. Enables correlation of symptoms across distributed manufacturing systems
3. Provides transparency into third-party component interactions

4. Facilitates root cause analysis through comprehensive tracing capabilities
5. Supports continuous improvement of pipeline reliability and performance

Level of risk exposed if this best practice is not established: High

Implementation guidance

- Implement structured logging across the data pipeline components using Amazon CloudWatch Logs for centralized log aggregation, AWS IoT Device Management for edge device logging, and Amazon Data Firehose for high-volume log streaming. Use AWS X-Ray trace IDs as correlation identifiers and leverage CloudWatch Log Insights for querying logs with manufacturing context like batch numbers and equipment tags.
- Deploy AWS X-Ray across your data services including Lambda functions, ECS containers, and API Gateway endpoints to trace data flow end-to-end. Configure X-Ray sampling rules to reduce overhead while maintaining visibility for critical manufacturing processes and use AWS App Mesh for service mesh tracing in containerized environments.
- Install Amazon CloudWatch agent on edge gateways and industrial PCs to collect system metrics, logs, and custom manufacturing metrics. Use AWS Systems Manager for agent deployment and configuration management and leverage AWS IoT Greengrass for edge computing monitoring with local data processing capabilities.
- Enable AWS CloudTrail for API call logging, configure Amazon API Gateway access logging and throttling monitoring, and use Amazon CloudWatch alarms to detect retry patterns and authentication failures. Implement AWS WAF logging for additional API security monitoring and use Amazon EventBridge for real-time API event processing.
- Create unified dashboards using Amazon CloudWatch Dashboards combined with AWS X-Ray service maps for end-to-end pipeline visualization. Use Amazon OpenSearch Service for advanced log analytics, Amazon Managed Grafana for custom manufacturing dashboards, and AWS Systems Manager OpsCenter for centralized operational issue management and correlation.

Key AWS services

- Amazon CloudWatch for metrics, logs, and dashboards
- CloudWatch Agent for on-premises monitoring
- AWS X-Ray for distributed tracing and service maps
- AWS CloudTrail for API activity monitoring

- Amazon OpenSearch Service for advanced log analytics
- Amazon Managed Grafana for visualization (where applicable)

Resources

- Monitoring AWS IoT Applications with CloudWatch
- Installing and Configuring the CloudWatch Agent
- Getting Started with AWS X-Ray
- Analyzing API Calls with CloudTrail

MIDAPERF04-BP02 Decouple data ingestion from processing in manufacturing systems

In manufacturing environments, tightly coupled data pipelines create single points of failure that can lead to data loss and operational disruptions. Implementing a decoupled architecture that separates ingestion from processing enhances system resilience, enables independent scaling of components, and provides the foundation for reliable data processing even during partial system failures common in industrial settings.

Desired outcome: A resilient manufacturing data architecture where ingestion and processing components operate independently, enabling continuous data capture during processing failures, supporting reprocessing capabilities when ingestion recovers from outages, and maintaining overall system performance through appropriate component scaling.

Common anti-patterns:

- Tightly coupling data ingestion directly to processing components - Creates single points of failure that can cause complete system shutdowns and data loss during processing failures
- Implementing synchronous processing without buffering - Forces ingestion to wait for processing completion, creating bottlenecks and reducing overall system throughput
- Using shared scaling policies for ingestion and processing - Leads to over-provisioning or under-provisioning of resources since components have different load patterns and scaling requirements
- Designing non-idempotent processing operations - Causes data corruption and inconsistencies during replay scenarios, requiring expensive cleanup operations that impact performance

- Failing to implement dead-letter queues or error handling - Results in infinite retry loops that consume resources and degrade system performance during data quality issues
- Configuring insufficient buffer retention periods - Forces data loss during extended outages, requiring expensive data recovery operations and potential reprocessing from external sources
- Omitting queue depth monitoring for scaling decisions - Causes reactive rather than proactive scaling, leading to buffer overflows and performance degradation during traffic spikes
- Creating processing components without replay capabilities - Requires rebuilding entire datasets during recovery, consuming significant computational resources and extending downtime
- Using inadequate buffer storage capacity planning - Results in data loss during peak ingestion periods or extended processing outages, requiring expensive data reconstruction
- Implementing blocking operations in ingestion pipelines - Creates cascading failures where upstream data collection stops when downstream processing experiences issues
- Designing stateful processing without proper checkpointing - Forces complete reprocessing from the beginning during failures, wasting computational resources and extending recovery times
- Configuring overly aggressive retry policies without backoff - Overwhelms failing components and prevents recovery while consuming network and computational resources unnecessarily

Benefits of establishing this best practice:

1. Helps prevent critical production data loss during equipment shutdowns or unplanned downtime – Continually captures sensor readings, alarm states, and process variables even when historians, SCADA systems, or edge devices require maintenance or experience hardware failures.
2. Allows dynamic resource allocation to match plant operational cycles and data volumes - Enables scaling data ingestion during high-production periods or maintenance windows while independently adjusting processing power for complex analytics like predictive maintenance algorithms or real-time quality control calculations.
3. Provides data replay capabilities for root cause analysis and process optimization - Supports reprocessing historical operational data after system outages, calibration changes, or when new analytics models are deployed to backfill insights for compliance reporting or process improvement initiatives.
4. Maintains data pipeline integrity despite industrial network instabilities and equipment faults – Continually operates through common industrial challenges like network congestion, PLC communication errors, fieldbus disruptions, or temporary sensor malfunctions that frequently impact manufacturing environments.

5. Minimizes production impact during system upgrades and reduces maintenance windows - Enables rolling updates of data processing systems without disrupting critical real-time monitoring, trending, or automated control loops, allowing maintenance activities during normal production hours rather than costly scheduled downtime.

Level of risk exposed if this best practice is not established: High

Implementation guidance

1. Implement Durable Ingestion Buffers - Deploy Amazon Kinesis Data Streams or Amazon MSK (Kafka) for high-throughput streaming data with configurable retention periods up to 365 days. For batch workloads, use Amazon SQS with extended message retention and DLQ configuration, combined with Amazon S3 for long-term storage overflow when queue limits are approached.
2. Design Idempotent Processing - Leverage Amazon DynamoDB conditional writes or Amazon RDS with upsert operations for processing idempotency. Implement AWS Lambda with event source mapping deduplication features or use Amazon Managed Service for Apache Flink for exactly-once processing semantics. Store processing state in DynamoDB with composite keys to track message processing status.
3. Configure Dead-Letter Handling - Set up Amazon SQS Dead Letter Queues with Amazon CloudWatch alarms for message count thresholds. Use Amazon SNS to trigger notifications when DLQ thresholds are exceeded. Store failed messages in Amazon S3 with lifecycle policies for cost optimization and use AWS Step Functions for orchestrating retry logic and failure investigation workflows.
4. Implement Replay Capabilities - Utilize Kinesis Data Streams' time-based replay functionality or Amazon MSK's offset management for streaming data replay. For batch data, implement S3-based data lake architecture with AWS Glue ETL jobs that can reprocess partitioned data based on timestamps. Use AWS Batch for large-scale reprocessing jobs with automatic retry and scaling capabilities.
5. Establish Independent Scaling - Configure Amazon EC2 Auto Scaling Groups with custom CloudWatch metrics for queue depth monitoring. Use AWS Application Auto Scaling for Kinesis shard scaling based on incoming records and iterator age metrics. Implement AWS Lambda concurrent execution limits and reserved concurrency to help prevent downstream system overload while allowing independent scaling of processing components.

Key Services

- Amazon Kinesis Data Streams for scalable data ingestion
- Amazon SQS for durable message queuing
- Amazon MSK (Managed Streaming for Apache Kafka) for high-throughput streaming
- Amazon S3 for durable data landing and replay capabilities
- AWS Lambda for serverless processing of ingested data
- Amazon EventBridge for event-based processing orchestration

Resources

- Resilience in AWS Data Pipelines
- AWS Prescriptive Guidance Patterns
- Amazon EventBridge features

Pre-process data at edge

MIDAPERF05: How do you pre-process data at edge to avoid sending all data to the cloud?

Manufacturing facilities are implementing intelligent edge computing solutions that filter and analyze production data directly on the factory floor, significantly reducing bandwidth requirements and cloud processing costs. This strategic pre-processing approach allows plant operations to maintain local control over time-sensitive decisions while still benefiting from cloud-based analytics for longer-term process optimization and predictive maintenance modeling.

Best practices

- MIDAPERF05-BP01 Implement edge data pre-processing
- MIDAPERF05-BP02 Optimize storage and access for current manufacturing data

MIDAPERF05-BP01 Implement edge data pre-processing

In manufacturing settings, industrial devices and sensors often generate massive volumes of raw data that may not all be valuable for cloud-based analytics. Implementing edge pre-processing capabilities enables local summarization, filtering, and aggregation of data before transmission, significantly reducing bandwidth requirements, cloud processing needs, and storage costs while still preserving analytical value.

Desired outcome: A manufacturing data architecture that optimally distributes processing between edge and cloud, performing appropriate data reduction, summarization, and filtering at the edge while preserving essential information for cloud-based analytics and long-term storage.

Common anti-patterns:

- Sending all sensor data unfiltered to the cloud without edge processing, creating massive bandwidth waste and storage costs
- Using fixed sampling rates regardless of operational context or equipment state, missing critical events during high-activity periods while wasting resources during stable operations
- Designing systems that cannot function locally during connectivity disruptions, losing valuable operational data and halting local decision-making
- Deploying edge devices with insufficient CPU, memory, or storage capacity to handle required pre-processing, creating bottlenecks and system failures
- Sending data without contextual filtering or prioritization, treating routine operational data the same as critical alerts or anomalies
- Failing to implement temporary storage at the edge, resulting in permanent data loss during network outages or connectivity issues
- Performing all data analysis in the cloud instead of leveraging edge capabilities for real-time local decisions and immediate responses
- Excessive data summarization that loses critical analytical value or masks important operational insights needed for maintenance and optimization
- Applying identical pre-processing logic across all equipment types and operational contexts without considering specific requirements or characteristics
- Not implementing immediate local processing for time-sensitive data that requires instant action or real-time operational adjustments
- Ignoring the compound effect of raw data storage costs over time, failing to implement appropriate data lifecycle and retention policies

- Designing systems assuming unlimited or cheap network bandwidth without considering actual connectivity constraints in industrial environments

Benefits of establishing this best practice:

1. Reduces network bandwidth requirements by 60-90% in typical manufacturing deployments
2. Decreases cloud storage costs proportionally to reduction in data volume
3. Lowers data ingestion and processing costs in cloud environments
4. Minimizes latency for local decision-making through edge processing
5. Improves overall system resilience by enabling continued local operation during connectivity disruptions

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Evaluate existing data pipelines to identify compression and aggregation opportunities that preserve analytical insights. Use Amazon Managed Service for Apache Flink to perform real-time stream processing for averaging high-frequency sensor data, use AWS IoT Device SDK to implement delta-based transmission protocols that only send data when values exceed defined thresholds, and deploy AWS Lambda functions to calculate KPIs and derived metrics at the edge before transmission to reduce payload sizes.
- Architect distributed computing capabilities at plant locations using ruggedized hardware optimized for industrial environments. Deploy AWS IoT Greengrass on industrial-grade edge devices to enable local compute, messaging, and ML inference capabilities, utilize Amazon EC2 instances or AWS Outposts for locations requiring substantial remote processing power, and implement AWS Systems Manager for remote device management and software deployment across manufacturing sites.
- Configure intelligent data collection systems that adapt based on operational states and process conditions. Use AWS IoT Device Defender and AWS IoT Events to create rules-based filtering that correlates equipment status with data collection requirements, implement Amazon DynamoDB at the edge using AWS IoT Greengrass to store operational context and filtering rules, and leverage AWS IoT Core message routing to direct different data streams based on production state classifications.

- Develop dynamic data acquisition systems that automatically adjust collection frequencies based on real-time equipment health and process stability indicators. Implement AWS IoT Greengrass ML Inference to run anomaly detection models locally that trigger increased sampling rates, use Amazon CloudWatch metrics and alarms to define operational state thresholds, and deploy AWS Lambda functions that dynamically reconfigure sampling parameters based on equipment condition scores and process variables.
- Implement resilient data buffering and intelligent transmission systems that maintain data integrity during network disruptions. Configure AWS IoT Greengrass local storage capabilities with Amazon DynamoDB local tables for temporary data persistence, implement Amazon Data Firehose for reliable data delivery with automatic retry mechanisms, and use AWS IoT Core device shadows to maintain synchronization state and prioritize critical alarm data transmission over historical trend data during bandwidth-constrained conditions.

Key AWS services

- AWS IoT Greengrass for edge processing and analytics
- AWS IoT SiteWise for equipment modeling and edge processing
- AWS IoT Core for secure device connectivity
- Amazon Kinesis for data streaming from edge to cloud
- AWS Lambda for custom edge processing functions
- Amazon S3 for storing pre-processed data

Resources

- Processing Data at the Edge with AWS IoT Greengrass
- Edge Processing with AWS IoT SiteWise Edge

MIDAPERF05-BP02 Optimize storage and access for current manufacturing data

In manufacturing environments, rapid access to recent operational data is critical for real-time monitoring, anomaly detection, and immediate decision-making. Implementing specialized time series database solutions for current data while leveraging cost-effective storage for historical

information creates an optimal balance between performance and cost, verifying that dashboards and analytics remain responsive for operational needs.

Desired outcome: A tiered data storage architecture that provides millisecond-level query performance for recent manufacturing data while cost-effectively storing historical information, resulting in responsive operational dashboards, efficient anomaly detection, and appropriate performance-to-cost ratios across different data lifecycle stages.

Common anti-patterns:

- Storing all historical data in expensive, high-performance databases without implementing data lifecycle management or tiered storage strategies
- Using relational databases or general-purpose storage solutions instead of purpose-built time series databases for manufacturing sensor data and operational metrics
- Implementing overly complex data models with excessive normalization that require multiple joins for simple time series queries in operational dashboards
- Querying raw, unaggregated historical data spanning years directly from operational dashboards instead of using pre-computed aggregations or summaries
- Setting excessively long retention periods (6+ months) in high-performance time series databases without analyzing actual operational access patterns
- Creating inefficient tagging and indexing strategies that don't align with common manufacturing query patterns, causing slow dashboard performance
- Failing to implement query federation, forcing applications to maintain separate connection logic for different storage tiers and complicating data access
- Loading operational dashboards with unnecessary historical context that extends query windows beyond immediate operational needs (24-48 hours)
- Using synchronous data migration processes that block real-time ingestion during data lifecycle transitions between storage tiers
- Implementing generic caching strategies instead of manufacturing-specific data access patterns, missing opportunities for significant performance gains
- Storing all data at full resolution permanently instead of implementing intelligent down sampling for aging data based on operational value
- Creating monolithic storage architectures that force all queries through a single database tier regardless of data age or access frequency

Benefits of establishing this best practice:

1. 1. Substantially reduces dashboard refresh latency for current operational data
2. Provides sub-second query response for real-time operational decision support
3. Optimizes storage costs by matching data access patterns with appropriate technologies
4. Improves overall system scalability by distributing query load across appropriate storage tiers
5. Enables more sophisticated real-time analytics without performance penalties

Level of risk exposed if this best practice is not established:

Medium

Implementation guidance

- Conduct comprehensive analysis of your manufacturing systems' data consumption using Amazon CloudWatch to monitor current query patterns and AWS X-Ray to trace application performance. Use Quick usage analytics to understand dashboard access frequency and identify critical real-time metrics that drive production decisions. Use AWS Cost and Usage Reports to correlate data access costs with business value.
- Deploy Amazon Timestream as your primary industrial IoT data store, configured for high-throughput sensor data ingestion with magnetic storage tier for 30-90 day retention windows. Complement with Amazon MemoryDB for sub-millisecond query requirements on critical process variables. Use AWS IoT Core and AWS IoT SiteWise for seamless OT-to-cloud data pipeline integration.
- Structure your Timestream tables with equipment-based partitioning and implement hierarchical tagging using AWS Resource Groups naming conventions. Use AWS Glue Data Catalog to maintain metadata schemas and leverage Amazon OpenSearch Service for fast dimensional queries across manufacturing assets and process parameters.
- Establish automated data archival using AWS Lambda functions initiated by Amazon EventBridge schedules to move aged data from Timestream to Amazon S3 with S3 Intelligent-Tiering. Implement data aggregation pipelines using AWS Glue ETL jobs to create summarized views during the transition process, reducing storage costs while preserving analytical value.
- Deploy Amazon Athena with federated queries to create unified access across Timestream (hot data) and Amazon S3 (warm and cold data) using a single SQL interface. Use AWS AppSync GraphQL APIs to provide consistent data access patterns for manufacturing applications and implement Amazon API Gateway caching to optimize performance across storage tiers.

Key AWS services

- Amazon Timestream for time series data storage
- Amazon OpenSearch Service for operational data visualization
- Amazon S3 for cost-effective historical data storage
- AWS Glue for data lifecycle management
- Amazon Athena for queries across multiple storage tiers
- Quick for operational dashboards

Resources

- Getting Started with Amazon Timestream

Data storage organization

MIDAPERF06: How do you organize your data storage to balance performance to cost?

Manufacturing plants that implement efficient storage and access systems for historical production data gain crucial insights to help protect against costly compliance violations during quality audits and miss valuable opportunities to identify recurring production performance issues, potentially leading to extended downtime and significant material waste that could have been avoided through pattern recognition in historical data.

Best practices

- MIDAPERF06-BP01 Implement efficient storage and access for historical manufacturing data

MIDAPERF06-BP01 Implement efficient storage and access for historical manufacturing data

In manufacturing environments, historical data serves critical functions beyond immediate operational needs, supporting long-term trend analysis, root cause investigations, and business performance validation. Implementing properly structured data lakes or warehouses for historical

manufacturing data improves cost-effective storage at scale while maintaining analytical capabilities for deriving strategic insights from extended operational timelines.

Desired outcome: A scalable, cost-effective historical data architecture that efficiently stores years of manufacturing data while enabling performant analytics, supporting business intelligence requirements, and providing evidence-based validation of continuous improvement initiatives across extended time periods.

Common anti-patterns:

- Storing all historical data in a single, unpartitioned repository without considering access patterns or query performance requirements
- Using row-based storage formats without compression, leading to unnecessarily high storage costs and slower query performance
- Keeping all historical data in expensive, high-performance storage tiers regardless of access frequency or business value
- Failing to maintain data catalogs, schemas, or business context, making historical data difficult to discover and interpret over time
- Not implementing materialized views, aggregation tables, or indexing strategies for common analytical patterns
- Storing manufacturing data without logical partitioning by time, production line, or product, forcing full dataset scans for targeted queries
- Using expensive, high-performance databases for historical data that doesn't require sub-second access times
- Not implementing automated tiering policies to move older data to cost-effective storage classes while maintaining accessibility
- Storing historical data in multiple incompatible formats without standardization, complicating cross-temporal analysis
- Relying on manual processes for data organization, optimization, and lifecycle management instead of automated policies and procedures

Benefits of establishing this best practice:

1. Enables cost-effective storage of multi-year manufacturing data at petabyte scale
2. Supports sophisticated trend analysis and pattern detection across extended production history

3. Provides factual basis for validating return on technology and process investments
4. Facilitates root cause analysis of intermittent or slowly developing quality issues
5. Serves as a foundation for advanced analytics and machine learning initiatives

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Build a multi-tier data lake using Amazon S3 with intelligent partitioning by date, hour, line, and product hierarchies, use AWS Lake Formation for centralized data lake management, and implement Amazon S3 Transfer Acceleration for high-speed manufacturing data uploads from plant edge systems.
- Deploy Apache Parquet and ORC columnar formats through AWS Glue ETL jobs with automatic compression algorithms, use Amazon S3 Intelligent-Tiering for cost optimization, and schedule AWS Glue crawlers to continuously optimize data layout based on manufacturing query access patterns.
- Implement AWS AWS Glue Data Catalog as your central metadata repository for the manufacturing datasets, use Amazon DataZone for business glossary management and data governance, and integrate AWS Lake Formation permissions to maintain data lineage and regulatory compliance across industrial data assets.
- Deploy Amazon Redshift materialized views for common manufacturing KPI aggregations, use Amazon Athena with AWS Glue for historical analysis when needed, and implement Amazon ElastiCache for frequently accessed production metrics and real-time dashboard acceleration.
- Configure S3 Lifecycle Management policies to automatically transition manufacturing data through storage classes (Standard to IA to Glacier to Deep Archive), implement AWS DataSync for automated archival processes, and use Amazon Macie to classify sensitive manufacturing data for appropriate retention and compliance management.

Key AWS services

- Amazon S3 for scalable, durable object storage
- AWS Lake Formation for data lake management
- Amazon Athena for serverless SQL queries
- AWS Glue for data cataloging and ETL
- Amazon Redshift for data warehousing

- Quick for business intelligence

Resources

- Building a Data Lake on AWS
- Best Practices for Organizing S3 Objects
- Implementing Data Lifecycle Management in S3

Manufacturing analytic in regulated industries with MachineMetrics on AWS

Cost optimization

The cost optimization pillar provides guidance to help manufacturing customers apply best practices in the design, delivery, and maintenance of their AWS environments, with the most effective use of services and resources at a minimal cost. Cost optimization includes the continual process of refinement and improvement of a system over its entire lifecycle.

Businesses of different sizes in the manufacturing sector can benefit significantly from the flexibility and scalability of the AWS Cloud when it comes to cost optimization. Whether a small workshop, a mid-sized factory, or a large enterprise with complex IT environments, each can use AWS to continuously evolve and optimize their resources, usage, and processes. The cloud allows companies to efficiently manage their infrastructure and applications, scaling as needed and paying only for what they use. This approach to cost optimization is applicable and advantageous across the entire spectrum of manufacturing businesses, enabling them to stay efficient and competitive regardless of their size or complexity.

However, even small and medium-sized manufacturing companies can maximize their return on investment (ROI) by migrating their workloads to the AWS cloud and implementing cost optimization best practices. This includes understanding cost at both the application and workload level, using AWS services and features that optimize for cost, and continuously monitoring and adjusting their infrastructure to maintain efficiency.

Topics

- Design principles
- Forecast and optimize
- Data lifecycle
- Migration optimization
- Forecast resource provisioning
- Balancing resources

Design principles

In addition to the design principles described in the cost optimization pillar of the AWS Well-Architected Framework whitepaper, the following design principles can help manufacturing

customers build and operate cost-aware workloads that achieve business outcomes while minimizing costs and allowing their organizations to maximize the return on investment:

- **Use managed services:** Manufacturing companies can leverage managed services to quickly implement industrial data management, supply chain optimization, and other critical capabilities without the overhead of building and maintaining the underlying infrastructure. This helps them focus their engineering efforts on driving business value rather than managing complex IT systems, ultimately reducing costs.

  For example, Use Amazon SageMaker AI to develop and deploy machine learning models for predictive maintenance, reducing the need for in-house ML infrastructure.
- **Optimize for manufacturing processes:** By designing workloads with an understanding of unique manufacturing workflows like just-in-time production and predictive maintenance, manufacturers can identify specific opportunities to optimize costs. For example, using spot instances for non-critical tasks or reserved instances for predictable workloads can result in significant savings.

  For example, Implement AWS Auto Scaling for compute resources that align with production schedules, scaling up during peak hours and down during off-hours.
- **Align with production schedules:** Manufacturers can monitor and scale their resource usage in sync with production schedules, ramping up during peak periods and scaling down during off-peak or maintenance periods. This dynamic optimization helps them avoid over-provisioning and minimize wasted spend.

  For example, Use AWS Lambda to automatically adjust database capacity based on real-time production data from IoT sensors.
- **Use edge computing:** Bringing processing power closer to the manufacturing plant through edge devices can reduce data transfer costs and latency for time-sensitive workloads like quality control. The ability to pre-process data at the edge and only send necessary telemetry to the cloud further optimizes costs.

  For example, Deploy AWS IoT Greengrass on edge devices to process quality control data locally, sending only anomalies or aggregated data to the cloud.
- **Implement asset lifecycle management:** By using cloud services to monitor equipment performance, predict failures, and proactively schedule maintenance, manufacturers can extend the lifespan of their assets and avoid costly unplanned downtime. This holistic approach to asset management helps optimize CAPEX and OPEX.

For example, Use AWS IoT Core and Quick to create a real-time dashboard of equipment health, enabling proactive maintenance scheduling.

- **Optimize energy consumption:** Using cloud-based monitoring and optimization tools, manufacturers can identify opportunities to reduce energy usage across their operations, from production equipment to facilities and transportation. Lowering energy costs directly impacts the bottom line.

  For example, Implement AWS IoT SiteWise to collect and analyze energy consumption data across the factory, identifying inefficiencies and opportunities for optimization.

- **Use big data analytics and machine learning:** Building a scalable data lake and applying advanced analytics and machine learning enables manufacturers to uncover insights that drive process improvements, quality enhancements, and overall cost optimization across their business.

  For example, Use Amazon EMR to process large volumes of production data, identifying patterns that can lead to process optimizations and cost reductions.

- **Use generative AI capabilities:** Incorporating generative AI into manufacturing workflows can automate tasks like product design, process optimization, and customer interactions, ultimately increasing efficiency and reducing labor costs. These generative AI capabilities also empower the workforce to ask natural language questions about optimizing costs of their infrastructure and receive data-driven suggestions tailored to their specific factory and manufacturing environments.

  Expanded manufacturing-specific generative AI use cases:

  - **Product design optimization:** Use generative AI models to automatically generate and iterate on product designs, reducing time-to-market and design costs.
  - **Process parameter optimization:** Use AI to suggest optimal process parameters for different manufacturing scenarios, improving yield and reducing waste.
  - **Predictive quality control:** Use generative models to simulate various production scenarios and predict potential quality issues before they occur.
  - **Supply chain optimization:** Use AI to generate and evaluate multiple supply chain configurations, optimizing for cost, reliability, and resilience.
  - **Automated documentation:** Implement AI to generate and update technical documentation, work instructions, and training materials, reducing manual effort and improving consistency.

- **Natural language interfaces:** Develop AI-powered chatbots that allow shop floor workers to query systems, report issues, or request assistance using natural language, improving efficiency and reducing training costs.

By adopting these design principles, manufacturing companies can build cost-optimized workloads that use the full capabilities of the cloud, enabling them to be more agile, efficient, and sustainable in their operations.

Forecast and optimize

MIDACOST01: How do you forecast and optimize cloud costs for manufacturing data workloads?

Manufacturing organizations need to effectively predict and manage cloud costs associated with industrial data workloads, including data ingestion, storage, processing, and analytics. This requires understanding historical patterns, production schedules, and business forecasts to make informed decisions about resource allocation and optimization.

Best practices

- MIDACOST01-BP01 Implement data-driven cost management using AWS cost tools and manufacturing data
- MIDACOST01-BP02 Configure automated cost monitoring and alerts for manufacturing workloads

MIDACOST01-BP01 Implement data-driven cost management using AWS cost tools and manufacturing data

Create reliable cost forecasts by combining AWS usage data with manufacturing schedules to enhance resource provisioning and budget planning accuracy. This involves analyzing production patterns, seasonal variations, and historical cloud usage to make informed decisions about resource allocation and cost optimization.

Desired outcome: Develop precise monthly and quarterly cost forecasts by combining AWS usage data with manufacturing schedules to improve forecast reliability for resource provisioning and budget planning.

Common anti-patterns:

- Relying solely on default AWS cost reports without implementing manufacturing-specific cost allocation tags
- Making resource provisioning decisions based on short-term usage data
- Failing to account for seasonal production variations when forecasting cloud costs
- Using the same forecasting approach for all types of manufacturing workloads without considering their unique characteristics
- Neglecting to correlate cloud spending with production output metrics
- Setting static budgets without considering manufacturing cycles and production schedules
- Making Reserved Instance or Savings Plan commitments without analyzing historical usage patterns
- Ignoring the impact of planned maintenance windows and product launches on resource requirements

Benefits of establishing this Best Practice:

- Improved budget planning and cost predictability
- Better alignment between IT spending and OT production needs
- Reduced risk of over-provisioning or under-provisioning resources
- Enhanced ability to optimize costs during varying production cycles

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

To systematically analyze and optimize costs:

- Configure AWS Cost Explorer to track resource usage by manufacturing workload
- Set up cost allocation tags that map to specific production lines and processes
- Create monthly reports comparing AWS resource utilization with production output

- Use AWS Budgets to set alerts based on predicted usage thresholds
- Integrate production scheduling data from your MES/ERP systems with AWS cost management tools
- Review and adjust resource allocation quarterly based on collected metrics

Implementation steps

1. Enable detailed cost and usage reporting for all cloud resources.
2. Create cost allocation tags aligned with manufacturing processes.
3. Establish a system to collect and analyze production schedule data.
4. Implement forecasting models that consider:
   - Seasonal production variations
   - Planned maintenance windows
   - New product launches
   - Historical resource utilization patterns
5. Set up regular review cycles to validate forecasts against actual usage.
6. Take advantage of cost saving mechanisms like AWS Savings Plans and Spot Instances.

Key AWS services

- AWS Cost Explorer
- AWS Budgets
- AWS Supply Chain
- Amazon SageMaker AI Canvas
- AWS Data Exports with Quick

Resources

Related documents:

- Analyzing your costs and usage with AWS Cost Explorer
- Managing your costs with AWS Budgets
- Demand Planning

- Time Series Forecasts in Amazon SageMaker AI Canvas
- Cloud Financial Management with AWS

MIDACOST01-BP02 Configure automated cost monitoring and alerts for manufacturing workloads

Set up a comprehensive alerting system that notifies teams within 24 hours when costs exceed thresholds, generates cost reports by production line, identifies waste, and maintains cost visibility across manufacturing operations. This includes setting up progressive alerting using different severity levels and implementing automated remediation for common cost-related issues.

Desired outcome: Set up a comprehensive alerting system that:

- Notifies teams within 24 hours when costs exceed defined thresholds
- Generates daily/weekly cost reports by production line
- Identifies resource waste and cost anomalies automatically
- Maintains cost visibility across manufacturing operations

Common anti-patterns:

- Setting up generic alerts without considering manufacturing-specific cost patterns
- Creating too many alerts that lead to notification fatigue
- Failing to establish baseline costs before implementing monitoring
- Not differentiating between production and non-production environment alerts
- Sending alerts to a general distribution list instead of specific responsible teams
- Using the same thresholds for different types of manufacturing workloads
- Implementing alerts without defined response procedures
- Focusing only on total cost without considering cost per unit of production
- Not accounting for shift patterns in alert configurations

Benefits of establishing this Best Practice:

- Early detection of cost anomalies

- Reduced manual monitoring effort
- Improved cost visibility across teams
- Faster response to cost-related issues

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Before setting up automation, verify that you have:

- Identified key stakeholders who need cost alerts (operations, finance, IT teams)
- Determined cost thresholds for different manufacturing processes
- Mapped your AWS resources to specific production lines or cells
- Established baseline costs for normal operations

Then, implement monitoring systems that:

- Track daily or weekly cost variations against production schedules
- Alert relevant teams when costs deviate your set thresholds (for example, 20%) or more from baseline
- Generate automated reports showing cost per unit of production
- Monitor resource utilization during different manufacturing shifts

Implementation steps

1. Define cost thresholds or budgets for different manufacturing workload components.
2. Configure automated alerts for:
   - Budget overruns
   - Unusual usage patterns
   - Idle resources
   - Storage growth rates
3. Create automated reports for:
   - Daily, weekly, or monthly cost trends
   - Resource utilization and production output

- Cost per manufacturing line, cell, or product

4. Establish escalation procedures for cost-related incidents.

Key AWS services

- AWS Cost Explorer
- AWS Budgets
- AWS CloudTrail
- AWS CloudWatch
- Amazon Simple Notification Service
- AWS Pricing Calculator
- AWS Lambda

Resources

Related documents:

- Detecting unusual spend with AWS Cost Anomaly Detection

- Cost Optimization with AWS
- Logging AWS Cost Management API calls with AWS CloudTrail
- Create a billing alarm to monitor your estimated AWS charges

Data lifecycle

MIDACOST02: How does your organization manage the full lifecycle of manufacturing data resources?

Effective management of industrial data resources throughout their lifecycle is crucial for cost optimization. This includes tracking resource usage, implementing proper decommissioning processes, and improving compliance with data retention policies.

MIDACOST02-BP04, BP05, and BP06 are closely related and focus on different aspects of managing manufacturing data and resources: retention policies, cloud resource policies, and cost controls. When implemented together, they provide a comprehensive approach to managing your manufacturing data lifecycle while optimizing costs.

Best practices

- MIDACOST02-BP01 Track resources over their lifetime
- MIDACOST02-BP02 Implement manufacturing-aware resource decommissioning process
- MIDACOST02-BP03 Automate production-aware resource decommissioning
- MIDACOST02-BP04 Implement manufacturing-specific data retention policies
- MIDACOST02-BP05 Develop cloud resource policies aligned with manufacturing operations
- MIDACOST02-BP06 Implement manufacturing-aware cost controls

MIDACOST02-BP01 Track resources over their lifetime

Implement complete visibility and control over resource lifecycle costs from creation to deletion, with comprehensive tagging aligned to manufacturing processes. This includes monitoring resource utilization against production metrics, implementing clear ownership and purpose documentation, and regularly reviewing resource usage patterns. Essential for understanding total cost of ownership and identifying optimization opportunities.

Desired outcome: Complete visibility and control over resource lifecycle costs from creation to deletion.

Common anti-patterns:

- Creating resources without implementing a consistent tagging strategy from day one
- Failing to assign clear ownership of resources during provisioning
- Using generic tags that do not reflect manufacturing-specific contexts (production line, cell, product)
- Neglecting to track resource dependencies, leading to orphaned resources after decommissioning
- Maintaining resources without clear business justification
- Not implementing automated cleanup procedures for temporary resources
- Tracking only active resources while ignoring deprecated or archived industrial data

- Using the same lifecycle management approach for all data types regardless of their criticality or retention requirements
- Failing to consider data compliance requirements when implementing lifecycle policies
- Not accounting for seasonal production variations when evaluating resource utilization
- Implementing lifecycle tracking tools without proper team training and documentation
- Allowing multiple teams to create resources without centralized visibility and governance

Benefits of establishing this Best Practice:

- Improved resource utilization
- Reduced waste from unused resources
- Better understanding of resource ROI
- Enhanced cost allocation accuracy

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Before you begin, you will need:

- Inventory of all manufacturing systems (for example, SCADA, MES, and PLM)
- Mapping of data flows between shop floor and enterprise systems
- Compliance requirements for data retention in your industry

Key decisions needed:

- Resource tagging strategy aligned with production lines and processes
- Lifecycle stages specific to manufacturing data and systems
- Thresholds for resource utilization in different production scenarios

Establish a systematic approach to track resources throughout their entire lifecycle, from provisioning to decommissioning, with appropriate governance controls for optimal utilization and cost management in your manufacturing environment.

Consider the following:

- Mapping cloud resources to specific production lines or product types
- Tracking resource usage against production output metrics
- Implementing different lifecycle policies for operational versus analytical data
- Aligning resource reviews with production cycles or shift patterns

Regularly review resource utilization in the context of manufacturing KPIs and adjust your tracking approach based on changes in production processes or compliance requirements.

Implementation steps

1. Implement comprehensive tagging strategy aligned with manufacturing processes.
2. Track resource creation, modification, and usage patterns in relation to production cycles.
3. Monitor resource dependencies, especially between OT and IT systems.
4. Document resource ownership and purpose, involving both IT and operations teams.
5. Conduct regular reviews of resource utilization metrics against production output.
6. Implement automated reporting on resource lifecycle stages, integrated with manufacturing dashboards.

Key AWS services

- AWS Config
- AWS Systems Manager
- AWS Resource Groups
- AWS Tag Editor
- AWS Cost Explorer
- AWS Application Cost Profiler
- AWS Trusted Advisor

Resources

Related documents:

- Tagging AWS Resources and Tag Editor

- Evaluating Resources with AWS Config Rules

- AWS Resource Groups

MIDACOST02-BP02 Implement manufacturing-aware resource decommissioning process

Systematically remove unused resources while preserving critical manufacturing data, maintaining production system integrity, and complying with industrial requirements. This involves careful consideration of dependencies between manufacturing systems, data retention requirements, and proper archival procedures before resource removal.

Desired outcome: Systematic removal of unused resources while preserving critical manufacturing data, maintaining production system integrity, and complying with industrial requirements.

Common anti-patterns:

- Decommissioning resources without checking their connection to active production lines
- Failing to preserve quality control and compliance data before resource removal
- Not considering seasonal manufacturing patterns when identifying unused resources
- Decommissioning without checking impact on OT or IT integrated systems
- Removing resources without validating manufacturing regulatory requirements
- Failing to archive production performance data and custom configuration settings before decommissioning
- Not considering maintenance and repair history requirements

Benefits of establishing this Best Practice:

- Reduced costs from unnecessary resource retention
- Minimized risk of accidental data loss
- Clear process for resource retirement
- Compliance with data governance requirements

Level of risk exposed if this best practice is not established: High

Implementation guidance

Establish formal processes for identifying and safely decommissioning resources in your manufacturing setup that are no longer needed, while meeting data preservation requirements and managing dependencies.

Implementation steps

1. Create decommissioning criteria based on:
    - Resource utilization thresholds
    - Business value assessment
    - Data retention requirements
2. Establish approval workflows.
3. Document dependencies and impact analysis.
4. Create backup and archival procedures.
5. Implement verification steps post-decommissioning.
6. Consider manufacturing-specific decommissioning criteria:
    - Production line changeovers
    - End of product lifecycle
    - Equipment replacement cycles
    - Historical data retention for quality compliance and machine learning

Key AWS services

- AWS Backup
- Amazon S3 Lifecycle policies
- AWS Organizations
- Amazon CloudWatch
- AWS Glue Data Catalog

Resources

Related documents:

- Amazon Simple Storage Service: Examples of S3 Lifecycle configurations
- AWS Backup
- Detecting unusual spend with AWS Cost Anomaly Detection

MIDACOST02-BP03 Automate production-aware resource decommissioning

Implement automated identification and removal of unused resources synchronized with production schedules, product lifecycles, and manufacturing compliance requirements. This automation includes safety checks, rollback procedures, and consideration of maintenance windows to help prevent disruption to manufacturing operations.

Desired outcome: Automated identification and removal of unused resources synchronized with production schedules, product lifecycles, and manufacturing compliance requirements.

Common anti-patterns:

- Implementing automated removal without considering production schedules
- Using the same automation rules for both IT and OT resources
- Not incorporating manufacturing compliance checks in automation
- Failing to account for interdependencies with MES, SCADA, or other manufacturing systems
- Automated decommissioning during production hours
- Not maintaining audit trails for regulated manufacturing processes
- Bypassing quality management system validations

Benefits of establishing this best practice:

- Reduced manual intervention
- Consistent application of decommissioning policies
- Immediate cost savings from unused resource removal
- Reduced human error

Level of risk exposed if this best practice is not established: High

Implementation guidance

Create automated systems that can safely identify, tag, notify relevant stakeholders, and finally remove resources that are no longer needed, with appropriate safeguards to help prevent disruption to manufacturing operations.

Implementation steps

1. Define automation rules for resource identification.
2. Create automated workflows for:
   - Resource tagging
   - Notification of stakeholders
   - Backup creation
   - Resource termination
3. Implement safety checks and rollback procedures.
4. Monitor automation effectiveness.
5. Include manufacturing-specific automation rules:
   - Production schedule-aware decommissioning
   - Product lifecycle milestones
   - Equipment maintenance windows
   - Shift pattern considerations

Key AWS services

- AWS Lambda
- Amazon EventBridge
- AWS Config Rules
- AWS Systems Manager Automation
- AWS Step Functions
- Amazon SNS

Resources

Related documents:

- AWS Systems Manager Automation
- Amazon EventBridge
- AWS Step Functions
- Building Lambda functions with Python

MIDACOST02-BP04 Implement manufacturing-specific data retention policies

Implement cost-effective industrial data management that balances retention requirements for production data, quality records, and compliance needs with optimized storage costs. This includes implementing tiered storage strategies and automated archival processes.

Desired outcome: Cost-effective industrial data management that balances retention requirements for production data, quality records, and compliance needs with optimized storage costs.

Common anti-patterns:

- Applying generic IT data retention policies to manufacturing data
- Failing to differentiate between operational data and long-term quality records
- Overlooking industry-specific regulations (for example, FDA, ISO) in retention policies
- Storing manufacturing data indefinitely without a defined purpose
- Not considering data dependencies in retention schedules (for example, keeping raw data but deleting related metadata)
- Implementing retention policies without input from production and quality teams

Benefits of establishing this best practice:

- Alignment with regulatory requirements
- Optimized storage costs
- Clear data lifecycle management
- Reduced risk of compliance violations

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Implement comprehensive data retention policies that store manufacturing data only as long as necessary for operational, regulatory, and business purposes while optimizing storage costs.

Implementation steps

1. Document regulatory requirements.
2. Define data classification schemes.
3. Create retention schedules.
4. Implement automated archival processes.
5. Set up compliance monitoring.
6. Regular policy review and updates.

Key AWS services

- Amazon S3 Lifecycle policies
- Amazon Glacier
- AWS Backup
- AWS Storage Gateway
- Amazon Macie
- AWS CloudTrail

Resources

Related documents:

- Amazon Simple Storage Service: Managing the lifecycle of objects
- Amazon Glacier
- AWS Backup
- Amazon Macie

MIDACOST02-BP05 Develop cloud resource policies aligned with manufacturing operations

Create well-defined policies for cloud resource provisioning, usage, and management that reflect specific manufacturing processes, compliance requirements, and cost optimization goals. These policies should consider both IT and OT needs while maintaining operational efficiency.

Desired outcome: Well-defined policies for cloud resource provisioning, usage, and management that reflect specific manufacturing processes, compliance requirements, and cost optimization goals.

Common anti-patterns:

- Creating generic cloud policies without considering manufacturing-specific needs
- Implementing policies that hinder rapid scaling during production spikes
- Overlooking OT or IT integration in policy development
- Failing to involve key stakeholders (for example, production managers and quality control) in policy creation
- Not accounting for different policy needs across various manufacturing stages (design, production, and maintenance)
- Implementing strict cost-saving policies that compromise manufacturing system reliability

Benefits of establishing this best practice:

- Standardized resource management
- Clear governance framework
- Aligned business and IT objectives
- Improved cost control

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Develop and implement comprehensive policies that reflect your organization's specific manufacturing requirements while improving cost optimization.

Implementation steps

1. Document organizational requirements:
   - Manufacturing process needs
   - Compliance requirements
   - Cost optimization targets
2. Create policy frameworks for:
   - Resource provisioning
   - Access control
   - Cost allocation
   - Data management
3. Establish review and approval processes.
4. Implement policy enforcement mechanisms.

Key AWS services

- AWS Organizations
- AWS Control Tower
- Service Catalog
- AWS IAM
- AWS Config
- AWS CloudFormation

Resources

Related documents:

- AWS Organizations
- AWS Control Tower
- Service Catalog
- AWS Config

MIDACOST02-BP06 Implement manufacturing-aware cost controls

Establish effective guardrails that help prevent unnecessary spending while maintaining operational efficiency and flexibility for production demands. This includes implementing approval workflows that don't hinder urgent production needs and differentiating between cost controls for different environments (production, development, testing).

Desired outcome: Effective guardrails that help prevent unnecessary spending while maintaining operational efficiency and flexibility for production demands.

Common anti-patterns:

- Applying blanket cost controls without considering critical manufacturing systems
- Implementing rigid resource limits that don't account for production variability
- Neglecting to create separate cost control policies for research and development, production, and quality assurance environments
- Failing to align cost control measures with manufacturing cycles and seasonal demands
- Implementing approval workflows that cause delays in scaling resources for urgent production needs
- Not differentiating between cost controls for operational data and long-term compliance data storage
- Implementing strict policies that hinder engineering research and development or applying overly permissive policies that lead to over provisioning
- Not training employees on best practices of deploying right-sized infrastructure/services that balance cost and performance

Benefits of establishing this Best Practice:

- Avoided cost overruns
- Controlled resource provisioning
- Enhanced budget compliance
- Improved cost predictability

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Establish mechanisms to monitor, control, and optimize cloud spending for manufacturing workloads while verifying that critical operational systems maintain necessary resources.

Implementation steps

1. Define cost control mechanisms:
    - Budget thresholds
    - Resource limits
    - Approval workflows
2. Implement automated enforcement.
3. Create exception processes.
4. Monitor control effectiveness.
5. Regular review and adjustment.

Key AWS services

- AWS Budgets
- AWS Cost Explorer
- AWS Service Quotas
- AWS Organizations
- AWS CloudFormation
- AWS Control Tower

Resources

Related documents:

- Managing your costs with AWS Budgets
- Analyzing your costs and usage with AWS Cost Explorer
- AWS Service Quotas
- AWS Cost Management

Migration optimization

MIDACOST03: How do you optimize your migration strategy to reduce infrastructure costs for manufacturing-specific workloads?

Developing an effective migration strategy for industrial workloads requires a thorough analysis of current on-premises systems, including product lifecycle management (PLM), manufacturing execution systems (MES), supervisory control and data acquisition (SCADA) systems, and other manufacturing-specific applications. This involves understanding unique manufacturing requirements, assessing OT and IT integration needs, considering production continuity, and planning for post-migration optimization for cost reduction without compromising operational efficiency.

Best practices

- MIDACOST03-BP01 Analyze all components of your industry workloads for migration and modernization

MIDACOST03-BP01 Analyze all components of your industry workloads for migration and modernization

Develop an optimized migration strategy that reduces costs while maintaining or improving operational capabilities for manufacturing-specific workloads. This includes analyzing OT and IT dependencies, assessing modernization opportunities, and planning migrations that minimize production impact.

Desired outcome: Optimized migration strategy that reduces costs while maintaining or improving operational capabilities.

Common anti-patterns:

- Migrating manufacturing systems without analyzing OT and IT dependencies
- Implementing direct migration strategies for all workloads while disregarding potential optimization opportunities
- Migrating critical production systems first without proper testing

- Overlooking data transfer costs between cloud and on-premises manufacturing equipment
- Planning migration without input from shop floor operations teams
- Ignoring regulatory compliance requirements specific to manufacturing processes
- Making migration decisions based solely on IT costs without considering production impact
- Not accounting for legacy manufacturing systems' integration requirements

Benefits of establishing this best practice:

- Reduced infrastructure costs
- Improved operational efficiency
- Optimized resource utilization
- Clear migration roadmap

Level of risk exposed if this best practice is not established: High

Implementation guidance

Before starting, verify that you have:

- Complete inventory of IT/OT systems and their dependencies
- Manufacturing process maps and criticality levels
- Compliance and regulatory requirements documented

Key decisions needed:

- Which systems are candidates for migration based on criticality
- Migration strategy selection (rehost, replatform, or refactor) per system
- Migration sequence aligned with production impact
- Cost-saving validation thresholds

Then, analyze industrial workloads by documenting current infrastructure costs, mapping OT and IT system dependencies (like SCADA to MES connections), and analyzing data transfer patterns. Evaluate options including rehost, replatform, or modernize based on each component's needs.

Create cost-benefit analyses comparing current versus projected cloud costs, with particular attention to manufacturing-specific requirements and production continuity.

Implementation steps

1. Conduct comprehensive workload assessment:
    - Current infrastructure costs
    - Application dependencies
    - Performance requirements
    - Data transfer patterns
2. Evaluate migration options:
    - Rehost (lift and shift)
    - Replatform
    - Refactor or modernize
3. Create a cost-benefit analysis.
4. Develop a migration timeline.
5. Plan for optimization post-migration.

Key AWS services

- AWS Migration Hub
- AWS Application Discovery Service
- AWS Migration Evaluator
- AWS Application Migration Service

Resources

Related documents:

- AWS Migration Hub
- AWS Application Discovery Service
- AWS Migration Evaluator
- AWS Application Migration Service

Forecast resource provisioning

MIDACOST04: How do you leverage historical manufacturing data to forecast resource provisioning?

Accurate resource provisioning in manufacturing environments relies on analyzing historical data from both IT and OT systems. This includes understanding production cycles, seasonal variations, and peak usage periods to create data-driven forecasting models. Effective use of historical data enables optimized capacity planning and cost management aligned with actual manufacturing operations.

Best practices

- MIDACOST04-BP01 Perform an analysis on the historical manufacturing workloads

MIDACOST04-BP01 Perform an analysis on the historical manufacturing workloads

Make data-driven resource provisioning decisions based on accurate historical usage patterns in manufacturing environments. This involves analyzing at least one full production cycle's data, considering seasonal variations, and accounting for planned maintenance windows in resource forecasting.

Desired outcome: Data-driven resource provisioning decisions based on accurate historical usage patterns in manufacturing environments.

Common anti-patterns:

- Using IT-only metrics without considering manufacturing operations data
- Basing forecasts on insufficient historical data (needs at least one full production cycle)
- Ignoring seasonal production variations in resource planning
- Not differentiating between development, testing, and production environment needs
- Failing to account for planned maintenance windows in resource forecasting
- Using the same forecasting model for both batch and continuous production processes

- Overlooking equipment upgrade cycles in long-term resource planning
- Not considering quality control and compliance requirements in resource forecasting

Benefits of establishing this best practice:

- Improved capacity planning
- Reduced overprovisioning
- Better alignment with production patterns
- Optimized resource costs

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Before you begin, you will need:

- At least one full production cycle's data
- Historical resource usage patterns across seasons
- Manufacturing schedules and maintenance windows documented

Key decisions needed:

- Forecast horizon based on production cycles
- Resource allocation thresholds for different workload types
- Scaling trigger points aligned with manufacturing needs
- Data retention requirements for compliance

Systematically collect and analyze resource utilization data from manufacturing systems to identify usage patterns and correlations with production cycles. Use these insights to create forecasting models that align with actual manufacturing operations, considering both IT and OT systems for comprehensive resource planning.

Implementation steps

1. Collect historical data on:

- Resource utilization
- Production cycles
- Seasonal variations
- Peak usage periods

2. Analyze patterns and trends:
   - Daily/weekly/monthly patterns
   - Production correlation
   - Seasonal impacts
3. Create baseline metrics.
4. Develop forecasting models.
5. Validate predictions against actual usage.

Key AWS services

- Quick
- AWS Cost Explorer
- Amazon CloudWatch
- AWS Systems Manager
- Amazon SageMaker AI

Resources

Related documents:

- Quick
- Analyzing your costs and usage with AWS Cost Explorer
- Metrics in Amazon CloudWatch
- AWS Systems Manager Inventory
- AWS Compute Optimizer
- Use the SageMaker AI AI DeepAR forecasting algorithm
- Time Series Forecasts in Amazon SageMaker AI Canvas

Balancing resources

MIDACOST05: How do you balance resource provisioning to optimize costs while meeting manufacturing demands?

Balancing resource provisioning in industrial settings requires implementing strategies that can handle variable workloads common in manufacturing. This involves using buffering and throttling approaches to manage data surges from production lines and equipment sensors, as well as implementing dynamic resource provisioning that aligns with production schedules and utilization patterns. The goal is to optimize costs while verifying that critical manufacturing operations remain unaffected.

Best practices

- MIDACOST05-BP01 Implement a buffering or throttling approach
- MIDACOST05-BP02 Implement dynamic resource provisioning

MIDACOST05-BP01 Implement a buffering or throttling approach

Implement balanced resource utilization that handles varying workload demands while maintaining cost efficiency for manufacturing systems. This includes prioritizing critical processes while queuing less time-sensitive tasks and implementing appropriate scaling triggers aligned with production cycles.

Desired outcome: Balanced resource utilization that handles varying workload demands while maintaining cost efficiency.

Common anti-patterns:

- Implementing throttling on time-critical manufacturing processes
- Using the same buffering strategy for all types of industrial data
- Overlooking real-time requirements of production monitoring systems
- Setting queue limits without considering production batch sizes
- Implementing aggressive throttling that impacts quality data collection
- Not accounting for upstream and downstream dependencies in manufacturing processes
- Using standard IT buffering patterns without adapting to manufacturing needs

Benefits of establishing this best practice:

- Controlled resource consumption
- Avoided system overload
- Optimized costs during peak periods
- Improved system stability

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Before you begin, you will need:

- Documented critical and non-critical manufacturing processes
- Peak resource utilization patterns for different production phases
- Response time requirements for various manufacturing systems

Key decisions needed:

- Resource allocation priorities for critical vs. non-critical processes
- Throttling thresholds for different types of manufacturing workloads
- Queue configurations for deferrable processes
- Scaling triggers aligned with production cycles and peaks

Implement buffering and throttling mechanisms to manage cloud resource utilization during manufacturing peaks. Design a system that prioritizes critical processes (for example, real-time monitoring, quality control) for immediate resource access, while queuing less time-sensitive tasks (for example, batch analytics, report generation). Use auto-scaling for baseline capacity but implement throttling to help prevent non-critical tasks from consuming resources needed for production-critical operations.

Consider the following:

- Using Spot Instances for interruptible, non-critical workloads
- Implementing reserved capacity for predictable, critical processes
- Using serverless technologies for sporadic, scalable tasks

Regularly review and adjust your buffering and throttling strategies based on changing production patterns and business needs.

Implementation steps

- Identify and categorize manufacturing workloads:
  - Critical real-time processes (for example, process control, safety systems)
  - Time-sensitive operations (for example, quality inspections, inventory updates)
  - Deferrable tasks (for example, long-term analytics, reporting)
- Design resource allocation strategies:
  - Priority-based access for critical systems
  - Queueing mechanisms for non-critical operations
  - Load balancing across production lines or facilities
- Implement OT-aware monitoring:
  - Set up real-time monitoring for critical production KPIs
  - Configure alerts based on manufacturing thresholds
  - Integrate with SCADA or MES for comprehensive visibility
- Establish OT-IT integrated scaling mechanisms:
  - Automatic scaling triggered by production volumes
  - Resource reservation for planned production increases
  - Gradual scale-down aligned with shift changes or maintenance windows
- Conduct regular performance and cost reviews:
  - Analyze resource utilization against production output
  - Identify opportunities for optimization without impacting OT
  - Adjust strategies based on changing manufacturing requirements
- Implement feedback loops with shop floor:
  - Gather input from operators on system performance
  - Align IT resource adjustments with production schedules
  - Continuously refine based on real-world manufacturing impact

Key AWS services

- Amazon SQS

- Amazon Kinesis
- AWS Auto Scaling
- Amazon API Gateway

Resources

Related documents:

- Amazon Simple Queue Service
- Amazon Kinesis Data Streams Developer Guide
- AWS Auto Scaling
- Throttle requests to your REST APIs for better throughput in API Gateway

MIDACOST05-BP02 Implement dynamic resource provisioning

Enable automated resource scaling that matches manufacturing workload demands while optimizing costs. This includes implementing warm pools for faster scaling, considering application warm-up times, and aligning scaling policies with production schedules and peak processing times.

Desired outcome: Automated resource scaling that matches manufacturing workload demands while optimizing costs.

Common anti-patterns:

- Implementing automatic scaling without considering production schedule requirements
- Setting scaling thresholds without consulting manufacturing operations teams
- Using the same scaling policies for both production and non-production workloads
- Neglecting warm-up times for manufacturing applications when scaling
- Implementing aggressive scale-in policies that could impact production monitoring
- Not accounting for data retention requirements when scaling storage resources
- Ignoring the impact of scaling on integrated manufacturing systems
- Setting up dynamic provisioning without consideration for compliance requirements

Benefits of establishing this best practice:

- Optimized resource utilization
- Reduced manual intervention
- Cost-efficient scaling
- Improved responsiveness

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Before you begin, you will need:

- Detailed production schedules and patterns
- Peak resource usage data by workload type
- System warm-up and response time requirements

Key decisions needed:

- Scaling thresholds for different manufacturing workloads
- Resource retention periods based on production needs
- Performance impact limits for critical systems
- Cost optimization targets by workload type

Design your manufacturing workloads to automatically adjust resource provisioning based on current demand and production schedules. Implement a data-driven approach that correlates IT resource needs with manufacturing operations, providing appropriate safeguards for critical production systems and consideration for startup times and warm pools.

Implementation steps

1. Define scaling metrics:
    - Production demand indicators
    - Resource utilization thresholds
    - Cost constraints
2. Configure auto scaling policies:
    - Scale-out conditions

- Scale-in conditions
- Cool-down periods

3. Implement monitoring.
4. Set up cost tracking.
5. Perform regular policy review and optimization.

Key AWS services

- AWS Auto Scaling
- Amazon EC2 Auto Scaling
- AWS Lambda
- Amazon CloudWatch

Resources

Related documents:

- AWS Auto Scaling
- Amazon EC2 Auto Scaling User Guide
- AWS Lambda: Configuring reserved concurrency for a function
- AWS Lambda: Configuring provisioned concurrency for a function PDF RSS
- Amazon CloudWatch: Using Amazon CloudWatch alarms
- Predictive scaling for Amazon EC2 Auto Scaling

Sustainability

The sustainability pillar provides you with the discipline to address the long-term environmental, economic, and societal impact of your business activities. You can find extensive prescriptive guidance on your implementation's sustainability in the Well-Architected Sustainability Pillar whitepaper.

Manufacturing organizations must focus on sustainability within their design, production processes and supply chain operations to reduce their impact on the environment and to encourage sustainable practices. Focusing on these areas helps manufacturers adapt their operations to industry sustainability best practices, adopt new environmentally friendly technologies and materials, and plan for the business impacts of potential future regulatory requirements related to environmental protection and resource conservation.

Topics

- Design principles
- Region selection
- Alignment to demand
- Data management
- Hardware and services
- Process and culture

Design principles

The following section defines a set of high-level design principles for manufacturing sustainability in AWS cloud environments.

- **Implement real-time monitoring and control only for critical production processes:** High-precision and complex manufacturing generally requires high performance computing, networks, and storage for real-time process monitoring and control. However, less complex production lines and operations typically do not require such intensive real-time capabilities. Manufacturers should evaluate their processes, risk profiles, and requirements to make the optimal real-time monitoring investment choices. But being thoughtful in implementing real-time monitoring generally provides sustainability and cost/efficiency benefits by avoiding over investment in capabilities that aren't needed across all operations.

- **Implement efficient data ingestion and edge processing:** Implement edge computing capabilities to filter, aggregate, and process data locally before transmission to the cloud. This approach reduces network bandwidth usage and associated energy consumption while enabling real-time insights where needed. The edge-first strategy verifies that only relevant data is transmitted to the cloud, optimizing both performance and sustainability for manufacturing operations.
- **Implement intelligent resource scaling for compute needs:** Implement auto scaling to automatically adjust compute resources based on actual manufacturing workload demands. Configure scaling policies for compute instances and other resources to match production patterns. This provides optimal resource utilization during peak production hours while scaling down during off-hours to reduce unnecessary energy consumption. well-designed auto scaling not only improves sustainability but also optimizes operational costs.
- **Use tiered storage for data requiring long term archiving:** Manufacturing records, product specifications, quality data, and regulatory compliance documentation must often be retained for 10-20 years or more. Implement a tiered storage strategy that optimizes storage costs by aligning retrieval needs with appropriate storage options. Frequently accessed data should use higher performance storage, while archival data can use lower cost cold storage tiers designed for long term retention with infrequent access patterns.
- **Implement smart data backup policies:** Avoid unnecessary duplication of backup data across multiple sites and storage tiers. Only back up data that is difficult or impossible to recreate from other sources. Leverage fully managed backup services with efficient deduplication and compressed storage.
- **Implement sustainability monitoring and optimization:** Create a comprehensive monitoring strategy to track and improve the environmental impact of manufacturing operations. Deploy resource utilization tracking tools, carbon footprint monitoring solutions, and cost management systems to identify optimization opportunities across your infrastructure. Regularly assess infrastructure using established sustainability frameworks and best practices. Establish clear sustainability KPIs and conduct periodic reviews to drive continuous improvement in environmental performance while maintaining operational efficiency. This data driven approach enables manufacturers to make informed decisions that reduce environmental impact and optimize resource usage across their infrastructure.

Region selection

MIDASUS01: How do you select a Region to optimize Manufacturing workloads for sustainability?

Manufacturing organizations should select regions that minimize environmental impact and support sustainable operations. By incorporating sustainability into their core strategy, manufacturers can optimize workloads for environmental efficiency and build resilience against evolving environmental regulations. This proactive approach not only reduces carbon footprint but also positions organizations to meet future sustainability requirements, achieve cost efficiencies, and maintain competitive advantage in an increasingly environmentally conscious world.

Best practices

- MIDASUS01-BP01 Select Regions that offer services required by Manufacturing organizations that maximizes the reduction of your carbon footprint

MIDASUS01-BP01 Select Regions that offer services required by Manufacturing organizations that maximizes the reduction of your carbon footprint

Choose Regions with lower carbon footprint for your manufacturing workloads while meeting technical, compliance, and performance requirements.

Desired outcome: Manufacturing workloads deployed in regions that minimize carbon footprint while maintaining operational excellence, compliance requirements, and optimizing for latency to manufacturing facilities.

Benefits of establishing this best practice:

- Reduced environmental impact and energy costs for manufacturing IT operations.
- Enhanced sustainability reporting for regulatory compliance and improved brand reputation.
- Strategic alignment with carbon reduction goals prepares the organization for evolving environmental regulations in manufacturing.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Select cloud regions with lower carbon footprint that also satisfy your manufacturing compliance requirements, data sovereignty needs, and performance thresholds.
- Deploy manufacturing workloads on energy-efficient computing systems within these sustainable regions to maximize environmental benefits.
- Configure automatic scaling policies that align with manufacturing production patterns to verify that computing resources are only active when needed.
- Implement edge processing at manufacturing facilities to reduce data transfer volumes between factories and cloud regions.
- Use efficient data transfer mechanisms and compression techniques when moving manufacturing data between regions to minimize network impact.
- Consider hybrid deployment models for manufacturing workloads that must remain geographically close to production facilities while still benefiting from cloud sustainability features.

Implementation steps

1. **Carbon footprint assessment:**
   - Conduct an environmental impact assessment of your current manufacturing workload deployment using the AWS Customer Carbon Footprint Tool
   - Map manufacturing compliance and technical requirements against available lower-carbon regions
   - Create a phased migration plan for manufacturing workloads to greener regions, prioritizing non-critical applications first
2. **Deploy energy-efficient computing:**
   - Deploy EC2 Graviton instances with Auto Scaling configurations that align with production schedules and peak processing times
   - Configure Amazon EC2 Auto Scaling groups based on manufacturing production patterns
3. **Optimize edge processing:**
   - Implement AWS IoT Greengrass at manufacturing facilities to optimize edge processing and reduce unnecessary data transfers

- Configure IoT rules to filter and aggregate manufacturing data at the edge
- Set up AWS DataSync for efficient transfer of required manufacturing data between regions

4. **Implement hybrid solutions:**
   - Deploy AWS Outposts or AWS Local Zones for manufacturing workloads requiring low-latency access to production facilities
   - Configure AWS Direct Connect for high-throughput, low-latency connectivity between manufacturing sites and sustainable regions
   - Implement Amazon S3 Transfer Acceleration for optimized cross-regional data movement when required
5. **Establish monitoring and governance:**
   - Create Amazon CloudWatch dashboards to track resource utilization and carbon metrics across regions
   - Establish sustainability KPIs and monitoring dashboards to track carbon reduction progress
   - Implement quarterly reviews to reassess regional deployment decisions based on sustainability performance metrics
6. **Continuous optimization:**
   - Use AWS Cost Explorer and Sustainability reports to identify further optimization opportunities
   - Regularly review and update regional deployment strategy as cloud provider sustainability features evolve

Key AWS services

- AWS Customer Carbon Footprint Tool
- Amazon EC2 Auto Scaling
- AWS Graviton
- AWS DataSync
- AWS IoT Greengrass
- AWS Outposts
- Amazon CloudWatch
- AWS Cost Explorer
- AWS Direct Connect

- AWS Local Zones

Resources

- AWS Customer Carbon Footprint Tool
- AWS Graviton Processors
- Get started with Amazon EC2 Auto Scaling
- AWS DataSync
- AWS Outposts Gives Manufacturers the Power of AWS On Premises
- AWS Local Zones

Alignment to demand

MIDASUS02: How do you optimize manufacturing workloads to align with production demands?

Manufacturing IT infrastructures support various operational patterns - from enterprise planning systems and product design workflows to shop floor operations and seasonal demand variations. By optimizing cloud resources to match these actual production demands, manufacturers can reduce environmental impact without compromising operational efficiency.

Best practices

- MIDASUS02-BP01 Actively manage workloads and resource allocation based on production demands

MIDASUS02-BP01 Actively manage workloads and resource allocation based on production demands

Identify critical and non-critical manufacturing systems, then align computing resources with actual production schedules and operational requirements to reduce waste while providing reliability for the essential manufacturing systems.

Desired outcome: Cloud resources that efficiently scale with manufacturing operations, prioritizing time-sensitive shop floor systems while optimizing resource usage for enterprise applications, resulting in reduced energy consumption, lower costs, and improved environmental sustainability.

Benefits of establishing this best practice:

- Reduced energy consumption and carbon footprint by removing over provisioning of resources.
- Lower operational costs while maintaining reliability for the production systems.
- Enhanced sustainability reporting metrics with quantifiable improvements in cloud resource efficiency aligned with manufacturing operations.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Analyze production schedules and system criticality to categorize manufacturing applications as time-critical (shop floor control, real-time monitoring) and non-time-critical (reporting functions, data processing) using workload assessment tools.
- Implement auto scaling mechanisms for all manufacturing systems based on their specific demand patterns, verifying that enterprise planning and design systems maintain necessary availability while optimizing resource allocation.
- Use batch processing systems for scheduling background data processing jobs like quality analysis, production reporting, and maintenance analytics during periods of lower resource demand.
- Deploy high performance computing (HPC) solutions and run resource-intensive engineering workloads such as computational fluid dynamics (CFD) and computer aided engineering (CAE) simulations during off-peak production hours to optimize resource utilization.

Implementation steps

1. **Assessment and classification:**
   - Conduct workload assessment of manufacturing applications using AWS Well-Architected Tool
   - Document peak usage patterns using Amazon CloudWatch
   - Classify applications into real time and batch processing categories
2. **Demand pattern mapping:**

- Create demand heat maps using Amazon CloudWatch metrics
- Identify off-peak windows for non-time-critical workloads

3. **Resource optimization configuration:**
   - Configure AWS Auto Scaling policies with appropriate thresholds
   - Implement scaling plans that align with production schedules
   - Define resource constraints to help prevent over-provisioning
4. **Workload scheduling implementation:**
   - Create AWS Batch job configurations for non-critical processing tasks
   - Configure AWS ParallelCluster for engineering simulations during off-hours
   - Implement prioritization logic for computing resources
5. **Monitoring and continuous improvement:**
   - Deploy CloudWatch dashboards to track resource utilization efficiency
   - Establish KPIs using AWS Cost and Usage Reports
   - Create sustainability improvement reporting and quarterly review process

Key AWS services

- AWS Auto Scaling
- Amazon EC2
- AWS Batch
- AWS ParallelCluster
- Amazon CloudWatch
- AWS Cost and Usage Reports

Resources

- Step and simple scaling policies for Amazon EC2 Auto Scaling
- AWS Batch
- AWS ParallelCluster
- Amazon EC2 instance types

Data management

MIDASUS03: How do you optimize data movement and access patterns?

In manufacturing environments, data is constantly generated across factory floors, production lines, and edge devices. Moving this data inefficiently across networks increases energy consumption and costs, while reducing application performance. Optimizing data movement and access patterns is crucial for sustainable manufacturing operations.

Best practices

- MIDASUS03-BP01 Implement edge data and cross-region movement strategies
- MIDASUS04-BP01 Design backup strategies that focus on critical data

MIDASUS03-BP01 Implement edge data and cross-region movement strategies

Implement strategies that process information at the edge when possible and strategically manage cross-Region transfers in manufacturing environments. This approach reduces unnecessary network traffic, lowers energy consumption, and improves operational efficiency in factory environments.

Desired outcome: Reduced data transfer across networks, optimized energy usage, faster access to manufacturing data, and improved application performance with lower carbon impact.

Benefits of establishing this best practice:

- Decreased network bandwidth consumption and associated energy usage
- Reduced carbon footprint from data centers and networking equipment
- Lower latency for manufacturing applications requiring real-time data
- Cost savings from reduced data transfer fees

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Process data at the source or edge locations to filter, compress, or aggregate data before transmitting to centralized storage, reducing unnecessary data movement and processing requirements.
- Implement efficient data transfer mechanisms when cross-region or cross-zone data movement is necessary, using compression, batching, and optimized transfer strategies.
- Store data in locations geographically closest to where it will be processed and accessed most frequently to minimize network latency and reduce energy consumed during data transit.
- Apply data lifecycle management strategies to automatically tier, archive, or delete data based on access patterns, compliance requirements, and business value, reducing storage footprint and associated energy costs.

Implementation steps

1. **Edge processing implementation:**
   - Deploy AWS IoT Greengrass to process sensor data locally at the edge
   - Configure data filtering rules to send only aggregated results to the cloud
   - Set up Lambda functions for edge-based data processing and reduction
2. **Efficient data transfer configuration:**
   - Implement Amazon S3 Transfer Acceleration and AWS Global Accelerator for cross-region data movement
   - Use Amazon CloudFront to cache frequently accessed data closer to end users
3. **Geographic data optimization:**
   - Store data in AWS Regions closest to production facilities
   - Configure Amazon S3 lifecycle policies for efficient data management
   - Monitor data access patterns using CloudWatch to identify optimization opportunities

Key AWS services

- AWS IoT Greengrass
- Amazon S3 Transfer Acceleration
- AWS Global Accelerator

- Amazon CloudFront

Resources

- AWS IoT Greengrass
- Configuring fast, secure file transfers using Amazon S3 Transfer Acceleration
- AWS Global Accelerator
- AWS for the Edge

MIDASUS04: How do you optimize your data backup strategies for sustainability in a manufacturing environment?

Manufacturing environments generate vast amounts of data from engineering, supply chain, production lines, IoT sensors, and quality control systems. Without proper backup strategies, organizations may unnecessarily duplicate data and consume excessive resources, impacting their sustainability goals.

Best practices

- MIDASUS04-BP01 Design backup strategies that focus on critical data

MIDASUS04-BP01 Design backup strategies that focus on critical data

Manufacturing data varies in criticality from essential production recipes and regulatory compliance records to temporary operational logs. Design backup strategies that prioritize truly valuable data while minimizing resource consumption.

Desired outcome: A data backup system that balances business continuity needs with sustainability goals, reducing storage requirements, energy consumption, and carbon footprint while maintaining manufacturing operational resilience.

Benefits of establishing this best practice:

1. Reduced storage costs and energy consumption
2. Lower carbon footprint from decreased data center resources

3. Optimized network bandwidth usage
4. Improved recovery time for truly critical manufacturing data

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

- Categorize data by importance (critical production recipes, compliance records, operational logs) to determine appropriate backup strategies for each category.
- Establish tiered backup strategies based on data importance, with more frequent and comprehensive backups for critical data while implementing less resource intensive approaches for lower priority data.
- Implement data compression, deduplication, and efficient formats to minimize storage footprint and processing requirements for backups across the data categories.
- Store manufacturing data in AWS Regions closest to production facilities to reduce energy used for data transfer and improve access speeds for operational systems.

Implementation steps

1. **Data classification and assessment:**
   - Conduct comprehensive data audits across Amazon EBS volumes, Amazon EFS file systems, and on-premises data connected through AWS Storage Gateway
   - Document recovery time objectives (RTOs) and recovery point objectives (RPOs) for each data category in AWS Backup
2. **Tiered backup strategy design:**
   - Configure AWS Backup plans with different frequencies for critical EBS volumes, EFS file systems, and Storage Gateway volumes
   - Implement lifecycle policies in Amazon S3 to automatically transition infrequently accessed backups to cold storage classes
3. **Storage optimization configuration:**
   - Enable compression and deduplication features in AWS Backup for EBS snapshots and EFS backups
   - Configure Amazon EFS Infrequent Access storage class for rarely accessed file data
   - Implement AWS Storage Gateway with deduplication enabled to reduce backup data footprint

4. **Geographic distribution setup:**
   - Deploy EBS and EFS backups to AWS Regions closest to primary usage locations
   - Configure AWS Storage Gateway to cache frequently accessed data locally while storing backups in energy-efficient regions
5. **Performance monitoring:**
   - Create CloudWatch dashboards to track backup storage utilization across EBS, EFS, and Storage Gateway
   - Establish quarterly review processes using AWS Trusted Advisor storage recommendations

Key AWS services

- AWS Backup
- Amazon S3
- Amazon Glacier
- Amazon EBS
- Amazon EFS
- AWS Storage Gateway
- Amazon CloudWatch
- AWS Trusted Advisor

Resources

- AWS Backup
- Amazon Simple Storage Service: Managing the lifecycle of objects
- Cloud Storage on AWS
- Optimize Siemens Teamcenter with Amazon FSx for NetApp ONTAP

Hardware and services

MIDASUS05: How do you optimize hardware resources for manufacturing workloads?

Efficient hardware utilization in manufacturing environments helps reduce energy consumption while maintaining the performance requirements of critical operations. By properly selecting and configuring compute resources, organizations can minimize their environmental impact while improving reliable manufacturing operations across design, simulation, planning, and execution systems.

Best practices

- MIDASUS05-BP01 Implement optimized compute, storage and networking hardware

MIDASUS05-BP01 Implement optimized compute, storage and networking hardware

Manufacturing workloads have diverse requirements - from compute intensive simulations, storage to real time shop floor systems. Choosing the right hardware architecture and service models helps minimize environmental impact while meeting manufacturing performance demands.

Desired outcome: Manufacturing workloads running on optimally sized and energy-efficient infrastructure that minimizes waste and carbon footprint while meeting performance requirements for critical operations, resulting in quantifiable sustainability improvements without compromising production reliability.

Benefits of establishing this best practice:

1. Reduced energy consumption and carbon emissions through removal of over provisioning and inefficient hardware choices.
2. Lower operational costs through optimized resource utilization.
3. Improved sustainability reporting metrics that demonstrate concrete steps toward environmental goals without sacrificing manufacturing performance.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Analyze hardware requirements for manufacturing workloads including PLM (Product Lifecycle Management), MES (Manufacturing Execution Systems), and ERP (Enterprise Resource Planning) applications based on vendor specifications and performance needs, selecting minimal resources necessary to meet functional requirements.

Implement right sizing strategies for all compute resources supporting manufacturing operations, verifying that production systems use the most efficient configurations based on actual workload demands rather than over-provisioned specifications.

Deploy storage tiering mechanisms to align manufacturing data accessibility requirements with appropriate storage technologies, moving infrequently accessed historical production data to more energy efficient storage options.

Use hardware sharing approaches where possible across manufacturing systems to increase utilization rates of deployed resources, improving efficiency through consolidated infrastructure and reducing the total environmental footprint.

Implementation steps

Manufacturing application resource assessment:

- Conduct workload profiling of PLM, MES, and ERP systems using AWS Compute Optimizer to identify optimization opportunities
- Implement instance rightsizing recommendations, prioritizing Graviton based instances for better performance and efficiency

Business application resource scheduling:

- Configure Amazon EC2 Auto Scaling with scheduled actions aligned to manufacturing business hours and usage patterns
- Implement hibernation policies for non-production environments during off-hours to reduce idle resource consumption

Manufacturing data storage tiering:

- Configure Amazon S3 Lifecycle policies to automatically transition production data to appropriate storage classes based on access patterns
- Implement intelligent tiering for files, documentation, and historical manufacturing data to optimize storage costs and efficiency

Engineering workload optimization:

- Deploy AWS Batch for compute-intensive manufacturing simulations with job queuing that maximizes resource utilization
- Deploy AWS ParallelCluster for high performance computing needs like CFD, FEA, and engineering simulations

Edge computing optimization:

- Deploy AWS IoT Greengrass for local processing of shop floor data to minimize unnecessary data transfer

Key AWS services

- AWS Compute Optimizer
- AWS Graviton processors
- AWS IoT Greengrass
- AWS Batch
- AWS ParallelCluster
- Amazon EC2 Auto Scaling
- Amazon S3

Resources

- AWS Compute Optimizer
- AWS Graviton Processors
- AWS IoT Greengrass
- AWS Batch
- High-Performance Computing with AWS ParallelCluster

Process and culture

MIDASUS06: How do you integrate sustainability into manufacturing processes and culture?

Manufacturing organizations can promote sustainability through effective processes and cultural change by establishing clear sustainability goals and implementing continuous improvement processes. Building a sustainability-oriented culture requires not just technological solutions, but also organizational alignment, education, and metrics that make environmental impact visible and actionable across the enterprise.

Best practices

- MIDASUS06-BP01 Implement sustainability driven manufacturing processes

MIDASUS06-BP01 Implement sustainability driven manufacturing processes

Building a sustainability focused culture in manufacturing environments helps drive energy efficiency from the shop floor to enterprise systems. By aligning sustainability goals with operational excellence initiatives, organizations can reduce environmental impact while improving manufacturing performance.

Desired outcome: Measurably reduced environmental impact across manufacturing operations through systematized processes that track sustainability metrics, optimize resource usage, and continuously improve efficiency. Manufacturing teams at all levels actively participate in sustainability initiatives as part of standard operating procedures rather than as separate efforts.

Benefits of establishing this best practice:

1. Holistic approach to sustainability that drives long term behavioral change rather than isolated initiatives.
2. Improved cross functional collaboration as sustainability becomes a shared organizational value.
3. Enhanced employee engagement through meaningful participation in environmental stewardship aligned with organizational purpose.

Level of risk exposed if this best practice is not established: Medium

Implementation guidance

Incorporate sustainability KPIs into existing manufacturing performance dashboards, creating visibility of environmental impact alongside traditional metrics to drive a culture of sustainable operations.

Establish systematic tracking and reporting of resource utilization across manufacturing systems, enabling teams to identify inefficiencies and prioritize improvements that reduce environmental impact.

Apply advanced analytics and predictive modeling to identify process optimization opportunities that balance production objectives with sustainability goals.

Implement financial accountability mechanisms that link resource consumption to costs, encouraging more efficient practices and sustainability-minded decision making throughout operations.

Implementation steps

Sustainability dashboard integration:

- Implement manufacturing dashboards using Quick that combine traditional metrics (OEE, quality, throughput) with sustainability KPIs

Resource utilization tracking:

- Deploy AWS Systems Manager to monitor resource consumption and maintain efficient software versions
- Create automated reports using Quick to highlight opportunities for improved resource efficiency across manufacturing systems

Predictive process optimization:

- Utilize Amazon SageMaker AI to develop models that identify process improvements that reduce resource consumption
- Implement data pipelines using AWS Glue to make manufacturing data available for sustainability analysis

Financial accountability framework:

- Configure AWS Cost Explorer with sustainability focused tags to track resource usage against production output
- Implement AWS Budgets to set sustainability targets and alert teams when consumption thresholds are approached

Key AWS services

- Quick
- AWS Systems Manager
- Amazon SageMaker AI
- AWS Cost Explorer
- AWS Budgets
- AWS Resource Groups & Tag Editor

Resources

- Quick: Publishing dashboards
- AWS Systems Manager Automation
- Amazon SageMaker AI

Conclusion

The goal of the Modern Industrial Data Lens for the Well-Architected Framework is to provide architectural best practices for designing and operating reliable, secure, efficient, and cost-effective regulated Manufacturing services workloads on AWS. In operational excellence, we outline best practices around how people, process, and operating models need to be aligned so that workloads running on AWS can support critical Manufacturing services business services. Architectures for manufacturing workloads need to incorporate security and evidence-based compliance design patterns. Manufacturing customers also need to continuously monitor, measure, and test failure and recovery in the cloud to achieve their business resiliency and performance objectives. These objectives can be met with significant cost savings by right-sizing and establishing governance models around consumption and monitoring of AWS resources.

This framework can improve security, resiliency, and operational efficiency for manufacturing customers migrating and building apps on AWS and can also assist in meeting regulatory and compliance obligations.

Contributors

Contributors to this version of the Well-Architected Modern Industrial Technology Lens document include:

- Akash Bhatia, Principal Solutions Architect, Amazon Web Services
- Emilia Brzozowska, Senior Solutions Architect, Manufacturing, Amazon Web Services
- Madhuri K. Srinivasan, Senior Technical Writer, Well-Architected, Amazon Web Services
- Matthew Wygant, Sr. TPM Guidance, Amazon Web Services
- Raja GT, Worldwide Senior Solutions Architect, Manufacturing, Amazon Web Services
- Rajesh Gomatam, Worldwide Principal Solutions Architect, Manufacturing, Amazon Web Services
- Ryan Vanderwerf, Senior Solutions Architect, Manufacturing, Amazon Web Services
- Steve Blackwell, Worldwide Technology Lead, Manufacturing, Amazon Web Services
- Stewart Matzek, Senior Technical Writer, Well-Architected, Amazon Web Services
- Steven Arita, Cloud Optimization Success SA, Amazon Web Services
- Vijay Karthick Baskar, Senior Solutions Architect, Manufacturing, Amazon Web Services

References

- The journey to modern manufacturing with AWS
- Real-time operational monitoring of renewable energy assets with AWS IoT
- Achieve Production Optimization with AWS Machine Learning

Document revisions

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

Change	Description	Date
Initial release	Initial release of the Modern Industrial Data Technology Lens.	October 30, 2025

AWS Glossary

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