iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

FLOWGRAPH-DRIVEN REPLICATION
OPTIMIZING SAP HANA SDI FOR REAL-TIME DATA LAKE INGESTION ON AWS

Sivasankararao Kavuri
Data Migration Lead

1. EXECUTIVE SUMMARY
SAP HANA Smart Data Integration (SDI) provides the connectivity and transformation layer many enterprises rely on to move data from on-premises HANA systems into cloud environments. When the target is a real-time, AWS-hosted data lake, the flowgraph layer, rather than the adapter or network layer, is frequently the dominant factor determining whether a replication pipeline meets its latency target under production volume. This article focuses specifically on flowgraph-level optimization techniques and their interaction with AWS target service selection, sizing, and cost.
The article presents a reference architecture for bridging SDI with AWS-native data lake services, a structured set of flowgraph optimization techniques, change data capture tuning guidance, a comparison of AWS landing targets, a benchmarking methodology with illustrative results, sizing and cost models, and governance and monitoring guidance specific to an AWS-hosted target environment. It is deliberately table-dense, since the underlying optimization decisions are best communicated as structured comparisons rather than narrative alone.
Across the illustrative benchmarks presented in this article, disciplined flowgraph optimization, informed target selection, and CDC tuning combined to reduce end-to-end replication latency by 40 to 65 percent relative to an unoptimized baseline, at broadly comparable infrastructure cost.

2. INTRODUCTION
2.1 PURPOSE AND CONTEXT
As enterprises extend on-premises HANA landscapes into AWS-hosted data lake architectures, the temptation is to treat SDI as a fixed-throughput connector and to solve latency problems entirely at the network or target infrastructure layer. In practice, flowgraph design choices, node ordering, partitioning strategy, and the balance between in-flow transformation and target-side processing, frequently have a larger effect on end-to-end latency than infrastructure sizing alone. This article treats flowgraph optimization as a first-class architectural concern.
2.2 SCOPE
The scope covers SDI flowgraph design and tuning, AWS target service selection among Amazon S3, Amazon Kinesis Data Streams, Amazon Managed Streaming for Apache Kafka (Amazon MSK), and AWS Glue Data Catalog integration, along with the sizing, cost, monitoring, and governance considerations specific to that combination. It builds on, but does not repeat in full, the general HANA SDI bridging framework covered in prior published guidance, and instead concentrates on the flowgraph and AWS-specific optimization layer.

In Scope	Out of Scope
SDI flowgraph node-level optimization	General SDI installation and licensing procedures
AWS target service selection and comparison	Non-AWS cloud provider target services
CDC tuning for real-time streams	Application-level business logic within source systems
Sizing, cost, and monitoring for the AWS-hosted target	Detailed AWS Identity and Access Management policy authoring syntax

Table 1. *Scope boundary for this article, distinguishing flowgraph and AWS-target optimization from general SDI setup.*
2.3 RELATIONSHIP TO THE GENERAL SDI BRIDGING FRAMEWORK
Prior published guidance on bridging on-premises HANA systems with cloud data lake architectures establishes a general four-layer framework spanning connectivity, provisioning, transformation, and consumption. This article assumes that general framework as a starting point and drills specifically into the transformation layer, where the flowgraph resides, and into the AWS-specific instantiation of the connectivity and consumption layers. Readers seeking guidance on governance, cataloging, or security principles at a general, provider-agnostic level should treat that prior guidance as the foundation this article builds upon, rather than expecting this article to restate it in full.
The relationship between the two is best understood as general framework and provider-specific profile: the four-layer structure remains constant, while the specific technology choices, tuning parameters, and benchmarks in this

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

article are AWS-specific and would differ in detail, though not in underlying structure, for a different cloud provider target.

3. Flowgraph Architecture Recap

A flowgraph is a directed pipeline of nodes executed by the SDI Data Provisioning Server, each node performing a discrete operation such as reading from a source, filtering rows, joining inputs, or writing to a target. Understanding the relative cost profile of each node type is a prerequisite for the optimization techniques presented in Section 6.

3.1 FLOWGRAPH EXECUTION MODEL

Nodes execute as a pipeline, with data flowing from source nodes through intermediate transformation nodes to target nodes. Depending on node type and configuration, execution can occur row by row or in batches, and this distinction materially affects both throughput and the latency profile of a given flowgraph.

Node Type	Typical Execution Style	Relative CPU Cost
Source / Target	Streaming, row or micro-batch	Low to moderate
Projection / Filter	Streaming, row-level	Low
Join	Buffered, depends on join strategy	Moderate to high
Union	Streaming, row-level	Low
Aggregation	Buffered, requires grouping state	High
Data Quality Transform	Streaming or buffered, rule-dependent	Moderate

Table 2. *Relative execution style and CPU cost profile by flowgraph node type.*

3.2 NODE PLACEMENT AND LATENCY CONTRIBUTION

Node Position in Flow	Typical Latency Contribution	Optimization Priority
Early filter or projection	Very low, reduces downstream volume	Low, already efficient
Mid-flow join	Moderate to high, depends on cardinality	High
Mid-flow aggregation	High, requires buffering	High
Late-stage target write	Depends on target throughput	Moderate

Table 3. *Typical latency contribution by node position, used to prioritize optimization effort in Section 6.*

4. Real-Time Ingestion Requirements for AWS Data Lakes

4.1 LATENCY TIERS

Tier	Target End-to-End Latency	Representative Use Case
Tier 1 - Near real-time	Under 30 seconds	Operational dashboards, fraud-adjacent monitoring
Tier 2 - Fast refresh	30 seconds to 5 minutes	Intraday analytics, inventory visibility
Tier 3 - Standard refresh	5 to 60 minutes	Daily operational reporting
Tier 4 - Batch	Hours	Historical analytics, model training data sets

Table 4. *Latency tier definitions used to classify ingestion requirements for AWS-hosted data lake consumers.*

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

4.2 AWS LANDING TARGET OVERVIEW

AWS Service	Primary Role	Typical Latency Tier Served
Amazon S3	Durable object storage landing zone	Tier 2 to Tier 4
Amazon Kinesis Data Streams	Real-time event ingestion buffer	Tier 1 to Tier 2
Amazon MSK (Managed Kafka)	Real-time event streaming backbone	Tier 1 to Tier 2
AWS Glue	Batch and micro-batch transformation, cataloging	Tier 3 to Tier 4
Amazon Redshift	Analytical query target for curated data	Tier 3 to Tier 4

Table 5. *AWS service roles and the ingestion latency tier each service most commonly serves.*

4.3 MAPPING BUSINESS REQUIREMENTS TO LATENCY TIERS

Latency tier classification, per Table 3, should be driven by an explicit business requirement rather than by default assumption. The mapping below illustrates how common business requirements translate into a specific tier.

Business Requirement	Mapped Latency Tier
Real-time fraud or exception monitoring	Tier 1
Operational dashboard refreshed during the business day	Tier 1 or Tier 2
Intraday inventory or order visibility	Tier 2
Daily management reporting	Tier 3
Historical trend analysis or model training	Tier 4

Table 6. *Illustrative mapping of common business requirements to the latency tiers defined in Table 3.*

5. Reference Architecture: SDI to AWS

The reference architecture places the SDI Data Provisioning Agent on premises, adjacent to the HANA system, with outbound connectivity to AWS. Depending on the target latency tier, the flowgraph writes either directly to a streaming target such as Amazon Kinesis Data Streams or Amazon MSK, or to Amazon S3 as a landing zone for subsequent AWS Glue processing.

5.1 ARCHITECTURE LAYERS

Layer	AWS-Side Component	Primary Responsibility
Connectivity	AWS Direct Connect or VPN, TLS termination	Secure, reliable network path
Provisioning	SDI remote source and adapter configuration	Data movement mode selection
Streaming ingress	Amazon Kinesis Data Streams or Amazon MSK	Real-time buffering and fan-out
Landing storage	Amazon S3	Durable raw and curated zone storage
Cataloging and transformation	AWS Glue Data Catalog and Glue ETL jobs	Metadata registration and batch transformation
Consumption	Amazon Redshift or Amazon Athena	Analytical query access

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Table 7. *Reference architecture layers mapping the on-premises to AWS bridge, extending the general framework with AWS-specific components.*

5.2 NETWORK PATH OPTIONS

Connectivity Option	Typical Latency Impact	Typical Use Case
AWS Direct Connect	Lowest, dedicated path	Tier 1 and Tier 2 workloads at meaningful volume
Site-to-site VPN over internet	Moderate, variable	Tier 2 and Tier 3 workloads, lower volume
Public internet with TLS only	Higher, most variable	Non-production or pilot workloads only

Table 8. *Network connectivity options between the on-premises Data Provisioning Agent and AWS, with latency implications.*

5.3 FAILOVER AND HIGH AVAILABILITY CONSIDERATIONS

A Tier 1 replication pipeline should not depend on a single Data Provisioning Agent host without a defined recovery path, since an agent outage directly interrupts every source it hosts. High availability design should address both the on-premises agent tier and the AWS-side target service.

Component	High Availability Approach
Data Provisioning Agent	Standby agent host with rapid failover, or active-active split by source group
Network path	Redundant Direct Connect circuits or a VPN fallback path
Amazon Kinesis Data Streams	Inherently managed for availability by the service; size shards with headroom
Amazon MSK	Multi-AZ broker deployment with a replication factor of at least 3
Amazon S3	Inherently managed for durability and availability by the service

Table 9. *High availability approach by component for a flowgraph-driven replication pipeline targeting AWS.*

6. Flowgraph Optimization Techniques

Building on the node cost profile in Section 3, this section presents specific optimization techniques ordered by typical impact on end-to-end latency.

6.1 NODE-LEVEL OPTIMIZATION TECHNIQUES

Technique	Applies To	Illustrative Latency Benefit
Push filters as early as possible	Filter nodes	10 to 25 percent volume reduction downstream
Replace mid-flow join with pre-joined source view where feasible	Join nodes	20 to 40 percent latency reduction
Limit aggregation scope to the minimum required grouping keys	Aggregation nodes	15 to 30 percent latency reduction
Split large flowgraphs into smaller, independently scheduled flows	Overall flow structure	Improves parallelism, variable benefit
Avoid unnecessary data type conversions mid-flow	Projection nodes	5 to 10 percent CPU reduction

Table 10. *Node-level flowgraph optimization techniques with illustrative latency or resource benefit.*

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

6.2 PARTITIONING AND PARALLELISM

Where a source table can be logically partitioned, for example by date range or business unit, running multiple parallel flowgraph instances against distinct partitions can materially improve throughput, provided the target service can accept concurrent writes without contention.

Partitioning Strategy	Parallelism Benefit	Target Compatibility Consideration
Date-range partitioning	High, evenly distributes historical load	Works well with S3 prefix-based landing
Business unit or region partitioning	Moderate, depends on data skew	Requires balanced partition sizing
Hash-based partitioning on key column	High, most even distribution	Requires stable, well-distributed key

Table 11. *Partitioning strategies for parallel flowgraph execution and their target compatibility considerations.*

6.3 PUSH-DOWN VERSUS IN-FLOW TRANSFORMATION

Transformation Location	Advantage	Trade-Off
In-flow (within the flowgraph)	Single consistent pipeline, easier lineage tracking	Higher SDI compute load
Push-down to source query	Reduces data volume before it leaves the source	Limited to source-supported operations
Downstream in AWS Glue	Leverages elastic AWS compute for heavy transformation	Adds a processing hop and latency

Table 12. *Comparison of transformation placement options across the flowgraph, source, and AWS-side processing layers.*

6.4 FLOWGRAPH TESTING STRATEGY

Every optimization technique in this section should be validated through a defined test sequence before being promoted to production, rather than assessed on developer intuition alone.

Test Stage	Purpose
Unit test on sample data	Confirms transformation logic correctness on a small, known data set
Volume test at production scale	Confirms throughput and latency under realistic load
Failure injection test	Confirms behavior when a source, network, or target dependency is unavailable
Regression benchmark against prior baseline	Confirms the change is a genuine improvement per Section 9

Table 13. *Flowgraph testing stages applied to every optimization change before production promotion.*

7. Change Data Capture Tuning for Real-Time Streams

7.1 KEY CDC CONFIGURATION PARAMETERS

Parameter	Effect	Illustrative Starting Value
Polling or log-read interval	Controls how frequently new changes are captured	1 to 5 seconds for Tier 1

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Batch size per capture cycle	Balances throughput against per-cycle latency	500 to 5,000 rows
Commit frequency to target	Controls durability versus throughput trade-off	Every cycle for Tier 1, batched for Tier 3
Parallel capture threads	Increases throughput for high-volume sources	2 to 4 threads per source

Table 14. *Key change data capture configuration parameters and illustrative starting values by latency tier.*

7.2 LATENCY CONTRIBUTORS IN THE CDC PATH

Latency Contributor	Typical Share of End-to-End Latency	Primary Mitigation
Source log read interval	10 to 20 percent	Reduce polling interval for Tier 1 workloads
Flowgraph transformation processing	20 to 40 percent	Apply Section 6 node-level optimizations
Network transit to AWS	10 to 25 percent	Use AWS Direct Connect for Tier 1 workloads
Target service write acknowledgment	15 to 30 percent	Select target service matching the latency tier per Table 5

Table 15. *Typical distribution of end-to-end latency across the CDC path, with primary mitigation for each contributor.*

7.3 INITIAL LOAD STRATEGY AHEAD OF ONGOING CAPTURE

Every change data capture pipeline requires an initial full load of existing data before ongoing capture begins, and the strategy for that initial load materially affects both cutover risk and source system impact.

Initial Load Strategy	Advantage	Trade-Off
Single full-table extract	Simple to implement and reconcile	Highest source system impact during extract
Chunked extract by key range	Lower peak source impact, resumable on failure	More complex reconciliation logic
Parallel chunked extract	Fastest completion for large tables	Highest concurrent source load during the window

Table 16. *Initial load strategy options for change data capture sources, with advantages and trade-offs.*

8. AWS Target Service Selection

8.1 AMAZON S3 LANDING CONSIDERATIONS

Consideration	Guidance
Object size	Favor moderate object sizes; very small objects increase request overhead
Prefix design	Partition by source table and date to support efficient downstream scanning
Write frequency	Micro-batch writes every 30 to 120 seconds balance latency and object count

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Consistency	Design downstream consumers to tolerate eventual visibility of newly written objects

Table 17. *Amazon S3 landing zone design considerations for SDI-sourced data.*

8.2 AMAZON KINESIS DATA STREAMS CONSIDERATIONS

Consideration	Guidance
Shard count	Size to expected peak records per second, with headroom for burst capture cycles
Partition key selection	Choose a key that distributes load evenly across shards
Retention period	Set according to downstream consumer recovery requirements
Consumer scaling	Ensure downstream consumers scale independently of shard count where possible

Table 18. *Amazon Kinesis Data Streams configuration considerations for Tier 1 SDI-sourced ingestion.*

8.3 AMAZON MSK (MANAGED KAFKA) CONSIDERATIONS

Consideration	Guidance
Topic partitioning	Align partition count to expected consumer parallelism
Replication factor	Set to at least 3 for production durability
Retention policy	Balance storage cost against replay and recovery requirements
Broker sizing	Size to sustained throughput, not only peak burst

Table 19. *Amazon MSK configuration considerations for Tier 1 SDI-sourced ingestion at higher sustained volume.*

8.4 TARGET SERVICE COMPARISON MATRIX

Attribute	Amazon S3	Kinesis Data Streams	Amazon MSK
Best-fit latency tier	Tier 2 to Tier 4	Tier 1 to Tier 2	Tier 1 to Tier 2
Operational complexity	Low	Moderate	Higher
Relative cost at moderate volume	Low	Moderate	Moderate to high
Native AWS Glue integration	Direct, native	Via Kinesis Data Firehose or consumer	Via MSK Connect or consumer

Table 20. *Comparison of Amazon S3, Kinesis Data Streams, and Amazon MSK as SDI target services.*

8.5 CONSUMPTION LAYER OPTIONS

Once data lands in Amazon S3, either directly or after streaming buffer processing, the consumption layer determines how analytics users ultimately query it. Amazon Redshift and Amazon Athena represent the two most common options and suit different query patterns.

Consumption Option	Best-Fit Query Pattern	Cost Model
Amazon Redshift	Frequent, complex analytical queries against curated data	Provisioned or serverless compute, billed continuously or per query

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Amazon Athena	Ad hoc, infrequent queries directly against S3 data	Billed per query, based on data scanned

Table 21. *Comparison of Amazon Redshift and Amazon Athena as consumption layer options for SDI-sourced data.*

9. Performance Benchmarking Methodology

9.1 BENCHMARK DIMENSIONS

Dimension	Measurement Approach
End-to-end latency	Timestamp difference between source commit and target write acknowledgment
Sustained throughput	Rows per second sustained over a defined observation window
Resource utilization	Data Provisioning Agent CPU and memory during sustained load
Cost per unit volume	Illustrative AWS service cost divided by rows processed over the same window

Table 22. *Benchmark dimensions and measurement approach used to evaluate flowgraph and target configuration changes.*

9.2 SAMPLE BENCHMARK RESULTS

Configuration	Median Latency	Sustained Throughput	Agent CPU Utilization
Baseline, unoptimized flowgraph, S3 target	48 seconds	3,200 rows/sec	78 percent
Optimized flowgraph (Section 6), S3 target	19 seconds	4,900 rows/sec	61 percent
Optimized flowgraph, Kinesis Data Streams target	6 seconds	5,600 rows/sec	58 percent
Optimized flowgraph, Kinesis target, Direct Connect	3 seconds	5,800 rows/sec	57 percent

Table 23. *Sample benchmark results illustrating the cumulative effect of flowgraph optimization, target selection, and network path.*

9.3 BENCHMARK ENVIRONMENT SPECIFICATION

Benchmark results are only comparable across iterations if the underlying environment specification is held constant, or explicitly documented when it changes. The specification below reflects the environment used to produce the sample results in Table 18.

Environment Element	Specification
Data Provisioning Agent sizing	12 vCPU, 48 GB memory
Source table volume	Approximately 8 million rows, moderate change rate
Network path (unless noted otherwise)	Site-to-site VPN over internet
Observation window per benchmark run	4 hours of sustained load

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Table 24. *Benchmark environment specification used to produce the sample results in this article.*

10. Sizing and Cost Modeling on AWS

10.1 DATA PROVISIONING AGENT SIZING

Workload Profile	Illustrative vCPU	Illustrative Memory
Tier 1, single high-volume source	8 to 12 vCPU	32 GB
Tier 1 to Tier 2, multiple sources	12 to 16 vCPU	48 GB
Tier 3 to Tier 4, batch-oriented	4 to 8 vCPU	16 to 24 GB

Table 25. *Illustrative Data Provisioning Agent sizing by workload profile for an AWS-targeted implementation.*

10.2 ILLUSTRATIVE AWS COST COMPONENTS

Cost Component	Driver	Illustrative Relative Weight
Amazon S3 storage and requests	Object volume and write frequency	Low to moderate
Kinesis Data Streams shard hours	Shard count and retention period	Moderate
Amazon MSK broker hours	Broker count and instance size	Moderate to high
AWS Glue job execution	Job frequency and data processing units consumed	Moderate
Data transfer to AWS	Volume of replicated data	Low to moderate

Table 26. *Illustrative AWS cost components for a flowgraph-driven replication pipeline and their primary cost driver.*

10.3 COST BY VOLUME TIER

Volume Tier	Illustrative Rows per Day	Illustrative Relative Monthly Cost
Small	Under 5 million	1.0x baseline
Medium	5 to 50 million	3.5x baseline
Large	50 to 250 million	9x baseline
Very large	Over 250 million	18x baseline, typically favors Amazon MSK over Kinesis

Table 27. *Illustrative relative monthly cost by ingestion volume tier for an AWS-targeted SDI pipeline.*

10.4 RESERVED VERSUS ON-DEMAND CAPACITY

For sustained, predictable workloads, reserved or provisioned capacity options generally offer a meaningful cost advantage over on-demand pricing, at the cost of reduced flexibility to scale down quickly if source volume decreases.

Capacity Model	Cost Characteristic	Best-Fit Scenario
On-demand	Higher unit cost, no commitment	Pilot or highly variable workloads
Provisioned throughput (Kinesis)	Lower unit cost at sustained volume, requires capacity planning	Stable, well-understood Tier 1 workloads
Reserved broker capacity (MSK-adjacent compute)	Lowest unit cost at scale, requires longer-term commitment	Large, mature, sustained-volume pipelines

Table 28. *Comparison of on-demand and reserved or provisioned capacity models for AWS target services.*

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

11. Monitoring and Observability

11.1 CORE METRICS

Metric	Source	Illustrative Alert Threshold
End-to-end replication latency	SDI monitoring plus target service timestamp	Above Tier target for 5 consecutive minutes
Kinesis Data Streams iterator age	Amazon CloudWatch	Above 60 seconds
MSK consumer lag	Amazon CloudWatch or Kafka metrics	Above defined lag threshold per topic
Data Provisioning Agent CPU and memory	Agent host monitoring	Above 80 percent sustained
S3 write error rate	Amazon CloudWatch	Any sustained non-zero rate

Table 29. *Core monitoring metrics for a flowgraph-driven, AWS-targeted replication pipeline.*

11.2 ALERTING TIERING

Alert Tier	Trigger	Response Expectation
Tier A - Critical	Sustained latency breach for a Tier 1 workload	Immediate investigation, same business hour
Tier B - Warning	Resource utilization above threshold without service impact yet	Investigation within 1 business day
Tier C - Informational	Isolated transient latency spike, self-resolved	Logged for trend review only

Table 30. *Alerting tiers for the monitoring metrics defined in Table 21.*

11.3 SAMPLE OPERATIONAL DASHBOARD EXTRACT

Pipeline	Target Service	Current Latency	Status
Order Transactions	Kinesis Data Streams	4 seconds	Normal
Financial Postings	Amazon MSK	6 seconds	Normal
Inventory Movements	Kinesis Data Streams	22 seconds	Watch
Vendor Master (batch)	Amazon S3 via Glue	Completed on schedule	Normal

Table 31. *Sample daily operational dashboard extract for an AWS-targeted flowgraph replication pipeline.*

12. Security and Compliance on AWS

12.1 IDENTITY AND ENCRYPTION CONTROLS

Control	Implementation Point
Least-privilege IAM role for the replication pipeline	AWS IAM, scoped to specific S3 prefixes, streams, or topics
Encryption in transit	TLS between the Data Provisioning Agent and all AWS endpoints
Encryption at rest	AWS Key Management Service-backed encryption on S3, Kinesis, and MSK
Credential rotation	Scheduled rotation of technical account and access key credentials

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Table 32. *Identity and encryption controls for a flowgraph-driven replication pipeline targeting AWS.*

12.2 COMPLIANCE CONSIDERATIONS

Where replicated data includes regulated or sensitive fields, masking should occur in the flowgraph transformation layer before data reaches AWS, consistent with the transformation-layer masking principle established in prior published HANA SDI bridging guidance. Compliance requirements specific to data residency should also inform AWS Region selection for target services.

Consideration	Guidance
Data residency	Select AWS Region consistent with applicable regulatory requirements
Sensitive field masking	Apply in the flowgraph transformation layer prior to landing in AWS
Audit logging	Enable AWS CloudTrail logging for all services in the replication path
Retention and deletion	Align S3 lifecycle policies and Kinesis or MSK retention with policy requirements

Table 33. *Compliance considerations for regulated data flowing through an AWS-targeted replication pipeline.*

13. Data Quality and Reconciliation

Reconciliation Check	Frequency	Illustrative Tolerance
Row count, source versus S3 landing zone	Hourly for Tier 1 and Tier 2	Zero variance beyond in-flight lag
Row count, source versus streaming target	Continuous, windowed	Zero variance beyond defined lag window
Checksum or hash comparison on sampled records	Daily	Zero mismatch on sampled set
Schema drift detection	On every source schema change event	Zero undetected drift

Table 34. *Reconciliation checks and illustrative tolerances for a flowgraph-driven replication pipeline.*

13.2 SCHEMA DRIFT DETECTION APPROACH

Source schema changes, such as an added or resized column, can silently break a flowgraph or produce malformed records in the AWS target if not detected proactively. Schema drift detection should compare the current source schema against a stored baseline on a scheduled basis, and any detected difference should raise an alert rather than allowing the flowgraph to continue processing under an assumption that no longer holds.

14. Governance and Metadata Cataloging

AWS Glue Data Catalog provides a natural registration point for metadata describing SDI-sourced data landing in Amazon S3, allowing downstream AWS analytics services to discover schema and partition structure without direct dependency on the source HANA system.

Metadata Field	Purpose
Source system and table name	Traces landed data back to its HANA origin
Flowgraph identifier	Traces the specific transformation pipeline that produced the data
Landing partition scheme	Documents the S3 prefix and partitioning convention
Latency tier classification	Documents the expected freshness service level for consumers
Sensitivity classification	Documents whether masking has been applied and to which fields

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Table 35. *Recommended metadata fields for AWS Glue Data Catalog registration of SDI-sourced data.*

14.2 CATALOG GOVERNANCE ROLES

Activity	Data Platform Team	Source System Owner	Data Governance Team
Metadata field definition	Responsible	Consulted	Accountable
Catalog registration at landing	Responsible	Informed	Informed
Sensitivity classification review	Consulted	Consulted	Accountable
Periodic catalog accuracy audit	Responsible	Informed	Accountable

Table 36. *Responsibility matrix for AWS Glue Data Catalog governance activities.*

15. Implementation Methodology

1. Classify each candidate source table by target latency tier per Table 4.
2. Select the AWS target service for each classified table per the comparison in Table 15.
3. Design an initial flowgraph and benchmark it against the baseline methodology in Section 9.
4. Apply node-level, partitioning, and push-down optimization techniques from Section 6 iteratively.
5. Tune change data capture parameters per Section 7 for Tier 1 and Tier 2 sources.
6. Configure the selected AWS target service following the considerations in Section 8.
7. Establish monitoring, alerting, and reconciliation checks before production cutover.
8. Register metadata in AWS Glue Data Catalog for every landed data set.
9. Conduct a final benchmark against production-representative volume before go-live.
10. Review cost against the modeling in Section 10 and adjust target service selection if warranted.

Phase	Illustrative Duration
Classification and target selection	1 to 2 weeks
Initial flowgraph build and baseline benchmark	2 to 3 weeks
Optimization iteration	2 to 4 weeks
AWS target configuration and monitoring setup	1 to 2 weeks
Final benchmark and cutover	1 to 2 weeks

Table 37. *Illustrative phase timeline for implementing a flowgraph-driven, AWS-targeted replication pipeline.*

16. Risk Factors and Mitigation

Risk Factor	Potential Impact	Mitigation
Unoptimized flowgraph deployed to production	Latency target missed under peak volume	Benchmark against Section 9 methodology before cutover
Undersized Kinesis shard count or MSK broker capacity	Throttling or consumer lag under peak load	Size against peak, not average, per Table 17
Missing reconciliation between source and AWS target	Undetected data loss	Implement checks per Table 24

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Sensitive data landing unmasked in S3	Compliance exposure	Apply masking in the flowgraph transformation layer
Network path undersized for Tier 1 workloads	Latency target structurally unreachable	Use AWS Direct Connect per Table 8 for Tier 1 workloads

Table 38. *Common risk factors in flowgraph-driven, AWS-targeted replication pipelines, with mitigations.*

17. Illustrative Multi-Week Benchmark Data

The table below illustrates how latency and throughput typically evolve over a multi-week optimization cycle for a single Tier 1 source table replicated to Amazon Kinesis Data Streams.

Week	Median Latency	P95 Latency	Sustained Throughput	Agent CPU
Week 1 (baseline)	48 seconds	94 seconds	3,200 rows/sec	78 percent
Week 2	31 seconds	62 seconds	3,900 rows/sec	71 percent
Week 3	19 seconds	38 seconds	4,600 rows/sec	65 percent
Week 4	9 seconds	21 seconds	5,300 rows/sec	60 percent
Week 5 (Direct Connect enabled)	4 seconds	11 seconds	5,700 rows/sec	58 percent
Week 6 (stabilized)	3 seconds	8 seconds	5,800 rows/sec	57 percent

Table 39. *Illustrative six-week optimization trend for a single Tier 1 source table, combining flowgraph, target, and network improvements.*

18. Industry Adoption Patterns

The patterns below are drawn from anonymized, generalized implementation experience. No specific organization is identified.

Industry Pattern	Dominant Target Service	Typical Latency Tier Focus
Manufacturing and supply chain	Amazon S3 with scheduled AWS Glue processing	Tier 2 to Tier 3
Retail and consumer, order and inventory	Amazon Kinesis Data Streams	Tier 1 to Tier 2
Financial services, transaction monitoring	Amazon MSK, with stricter masking controls	Tier 1

Table 40. *Cross-industry summary of dominant AWS target service and latency tier focus.*

Across all three patterns, the underlying decision logic remains consistent with the latency tier classification introduced in Section 4: the specific AWS target service and masking posture vary by industry, but the classification-first approach to selecting them does not.

19. Common Pitfalls

- Selecting an AWS target service before classifying the actual required latency tier.
- Benchmarking flowgraph changes against a non-production-representative data volume.
- Sizing Kinesis shards or MSK brokers against average rather than peak throughput.
- Applying data masking only after data has already landed in Amazon S3.
- Allowing flowgraph optimization work to proceed without a documented before-and-after benchmark.
- Neglecting to register AWS Glue Data Catalog metadata until after a pipeline is already in production.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

20. Best Practices and Recommendations

- Classify every source table by target latency tier before selecting an AWS target service.
- Benchmark flowgraphs against production-representative volume before and after every optimization change.
- Apply node-level optimization techniques in the priority order established in Table 9.
- Prefer AWS Direct Connect over internet-based connectivity for Tier 1 workloads.
- Size Kinesis shards and MSK broker capacity against peak, not average, throughput.
- Apply sensitive data masking in the flowgraph transformation layer, prior to any AWS landing.
- Register AWS Glue Data Catalog metadata for every landed data set as part of the standard implementation methodology, not as an afterthought.
- Maintain a rolling multi-week benchmark trend, as illustrated in Table 26, to track optimization progress objectively.

21. Future Outlook

As of mid-2021, enterprise interest in combining on-premises transactional systems with AWS-native, real-time analytics services was continuing to grow, driven in part by increasing maturity of managed streaming services and by growing familiarity with flowgraph-level optimization techniques among data platform teams. Continued investment in benchmarking discipline, standardized latency tier classification, and AWS-native cataloging integration is expected to remain a priority as flowgraph-driven replication pipelines scale in both source count and target diversity.

22. Conclusion

Flowgraph-level optimization is a distinct and often underweighted lever in SAP HANA SDI pipelines targeting AWS-hosted, real-time data lake architectures. The techniques, comparisons, and benchmarks presented in this article demonstrate that combining disciplined flowgraph design with informed AWS target service selection and CDC tuning can materially reduce end-to-end latency without a proportional increase in infrastructure cost. Teams implementing this class of pipeline are encouraged to treat benchmarking as a continuous practice, not a one-time validation step, and to classify every source table by latency tier before committing to a specific AWS target service or network connectivity option.

23. REFERENCES

1) Akidau, T., Chernyak, S., & Lax, R. (2018). Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing. O'Reilly Media.
2) Amazon Web Services. (2019). Streaming Data Solutions on AWS with Amazon Kinesis. AWS Whitepaper.
3) Amazon Web Services. (2020). Amazon Simple Storage Service (S3) User Guide: Best Practices Design Patterns. AWS Documentation.
4) Amazon Web Services. (2020). Building a Data Lake on AWS. AWS Whitepaper.
5) Amazon Web Services. (2021). Amazon Managed Streaming for Apache Kafka Developer Guide. AWS Documentation.
6) Gartner, Inc. (2020). How to Build an Effective Data Lake Strategy. Gartner Research.
7) Gartner, Inc. (2020). Market Guide for Data Integration Tools. Gartner Research.
8) Heilman, R., Sanyal, J., Ravi, R., & Yang, J. (2017). SAP HANA Smart Data Integration and Smart Data Quality. SAP PRESS.
9) IDC. (2020). Worldwide Data Integration and Intelligence Software Market Forecast, 2020 to 2024. IDC.
10) Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media.
11) Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A Distributed Messaging System for Log Processing. Proceedings of the NetDB Workshop.
12) SAP SE. (2018). SAP HANA Data Provisioning Guide. SAP Help Portal.
13) SAP SE. (2019). SAP HANA Smart Data Integration and Smart Data Quality: Administration Guide. SAP Help Portal.
14) SAP SE. (2021, February). SAP HANA Smart Data Integration: Performance and Tuning Guide. SAP Help Portal.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Appendix A: Glossary of Key Terms

Term	Definition
SDI	SAP HANA Smart Data Integration, a data provisioning and transformation framework.
Flowgraph	A visually designed data transformation and movement pipeline executed by SDI.
CDC	Change Data Capture; a technique for continuously propagating source data changes.
Amazon Kinesis Data Streams	An AWS managed service for real-time data streaming ingestion.
Amazon MSK	Amazon Managed Streaming for Apache Kafka, a managed Kafka service on AWS.
AWS Glue	An AWS managed service for data cataloging and batch extract-transform-load processing.
Latency Tier	A classification of required end-to-end data freshness, defined in Table 3.
AWS Direct Connect	A dedicated network connection service between on-premises infrastructure and AWS.
Shard	A unit of throughput capacity within an Amazon Kinesis Data Stream.

Table 41. *Glossary of key terms referenced throughout this article.*

Appendix B: Flowgraph Optimization Checklist

- Source table classified by target latency tier before flowgraph design begins.
- Filter and projection nodes placed as early as possible in the flowgraph.
- Mid-flow joins reviewed for opportunities to pre-join at the source or eliminate entirely.
- Aggregation scope limited to the minimum required grouping keys.
- Partitioning strategy defined and validated for parallel execution where applicable.
- Push-down versus in-flow versus downstream transformation placement decided explicitly per Table 12.
- Baseline benchmark captured before any optimization changes are applied.
- Each optimization change benchmarked independently against the baseline.
- Final benchmark captured against production-representative volume before cutover.

Appendix C: Sample AWS Landing Zone Configuration Reference

Configuration Item	Sample Value
S3 bucket naming convention	org-datalake-raw-hana-<region>
S3 prefix structure	<source_system>/<table_name>/<yyyy>/<mm>/<dd>/
Kinesis stream naming convention	hana-cdc-<source_system>-<table_name>
Default shard count (Tier 1 pilot)	4 shards, scaled based on Table 26 throughput trend
Glue Data Catalog database name	hana_sdi_landing
IAM role naming convention	role-sdi-replication-<environment>

Table 42. *Sample AWS landing zone configuration reference values for a new SDI replication pipeline.*

Appendix D: Sample IAM Policy Scope Reference

The reference below summarizes the minimum IAM permission scope typically required for the replication pipeline's technical role, consistent with the least-privilege principle established in Section 12.1.

AWS Service	Minimum Required Permission Scope
Amazon S3	Write access scoped to the specific landing bucket and prefix only

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Amazon Kinesis Data Streams	PutRecord and PutRecords scoped to the specific stream only
Amazon MSK	Produce access scoped to the specific topic and cluster only
AWS Glue Data Catalog	Create and update table metadata scoped to the specific database only
AWS Key Management Service	Encrypt and decrypt scoped to the specific key used by the pipeline

Table 43. *Minimum IAM permission scope reference by AWS service for a flowgraph replication pipeline's technical role.*

