Data Pipeline Monitoring Checklist 2026

Enterprise Data Solutions
Comprehensive Pipeline Monitoring & Observability Framework

Version: 2.0 **Last Updated:** November 2026 **Type:** Checklist & Templates

Website: EnterpriseDataSolutions.co.nz
Email: Contact@EnterpriseDataSolutions.co.nz

Table of Contents

Introduction

Purpose of This Checklist

This Data Pipeline Monitoring Checklist provides a comprehensive framework to ensure your data pipelines are production-ready, observable, and maintainable. It covers 60+ monitoring checkpoints across data quality, performance, reliability, and observability dimensions.

Who Should Use This Checklist?

Stakeholder	Primary Use
Data Engineers	Pipeline development and monitoring implementation
Platform Engineers	Infrastructure and observability setup
DataOps Teams	Operational monitoring and incident response
Data Quality Analysts	Quality monitoring and validation
Engineering Managers	Production readiness assessments
SRE Teams	Reliability and availability monitoring

Pipeline Monitoring Principles

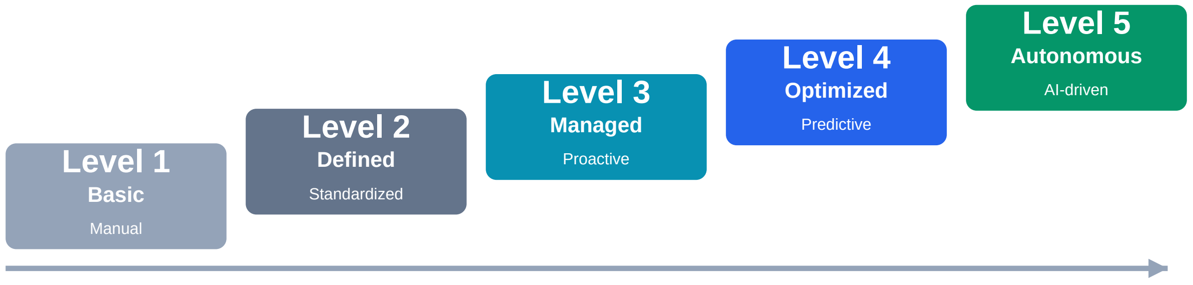
Principle	Description
Proactive Detection	Identify issues before they impact downstream consumers
End-to-End Visibility	Monitor the complete data journey from source to destination
Actionable Alerts	Every alert should trigger a specific action
Data-Aware Monitoring	Monitor data content, not just infrastructure
Cost Efficiency	Balance monitoring coverage with operational costs
Automation First	Automate detection, alerting, and remediation where possible

Pipeline Monitoring Maturity Model

Maturity Levels

Level	Name	Description	Key Capabilities
1	Basic	Manual monitoring, reactive response	Job success/failure tracking; Manual data validation; Email notifications
2	Defined	Standardized monitoring, basic automation	Centralized logging; Basic metrics dashboards; SLA tracking
3	Managed	Comprehensive monitoring, proactive detection	Data quality rules; Anomaly detection; Automated alerting; Runbooks
4	Optimized	Predictive monitoring, self-healing	ML-based anomaly detection; Auto-remediation; Cost optimization; Full lineage
5	Autonomous	AI-driven operations, continuous improvement	Predictive failure analysis; Self-tuning pipelines; Intelligent routing

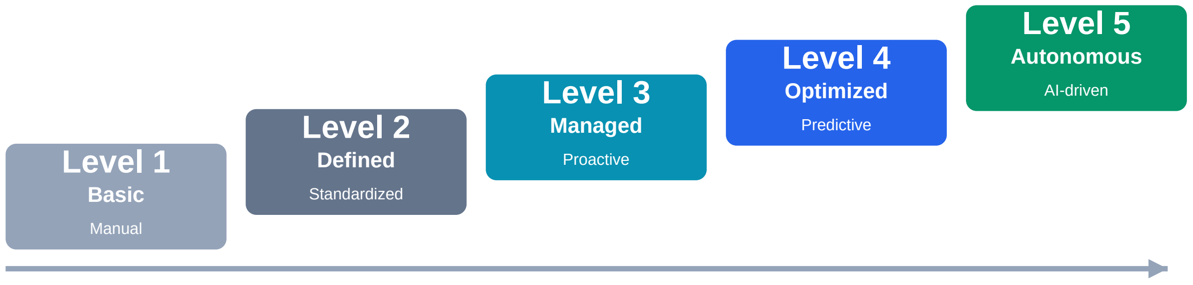
Monitoring Maturity Journey



Maturity Assessment Scorecard

Dimension	Level 1	Level 2	Level 3	Level 4	Level 5	Current
Job Monitoring	Manual checks	Success/fail tracking	SLA monitoring	Predictive alerts	Self-healing	
Data Quality	None	Manual validation	Automated rules	Anomaly detection	ML quality models	
Alerting	Email only	Basic routing	Smart routing	Context-aware	Predictive	
Incident Response	Ad-hoc	Basic runbooks	Full runbooks	Auto-remediation	Self-healing	
Observability	Logs only	Logs + metrics	Full tracing	Correlation	AI-powered	
Cost Monitoring	None	Monthly review	Real-time tracking	Optimization	Auto-scaling	

Monitoring Maturity Journey



Pre-Production Checklist

Pipeline Design Review

#	Checkpoint	Status	Owner	Notes
1.1	Pipeline architecture documented	[ ]		
1.2	Data flow diagram created	[ ]		
1.3	Source and target systems identified	[ ]		
1.4	Data schema documented and versioned	[ ]		
1.5	Error handling strategy defined	[ ]		
1.6	Retry logic implemented	[ ]		
1.7	Idempotency ensured	[ ]		
1.8	Dead letter queue configured	[ ]		
1.9	Backfill strategy documented	[ ]		
1.10	Pipeline dependencies mapped	[ ]		

Testing Requirements

#	Checkpoint	Status	Owner	Notes
2.1	Unit tests for transformations	[ ]		
2.2	Integration tests with mock data	[ ]		
2.3	End-to-end tests in staging	[ ]		
2.4	Data quality tests implemented	[ ]		
2.5	Performance tests completed	[ ]		
2.6	Failure scenario tests	[ ]		
2.7	Recovery procedure tests	[ ]		
2.8	Load tests at 2x expected volume	[ ]		
2.9	Schema evolution tests	[ ]		
2.10	Rollback procedure tested	[ ]		

Operational Readiness

#	Checkpoint	Status	Owner	Notes
3.1	Monitoring dashboards created	[ ]		
3.2	Alerting rules configured	[ ]		
3.3	On-call rotation assigned	[ ]		
3.4	Runbooks documented	[ ]		
3.5	SLAs/SLOs defined	[ ]		
3.6	Escalation paths documented	[ ]		
3.7	Access controls configured	[ ]		
3.8	Secrets management setup	[ ]		
3.9	Logging standards implemented	[ ]		
3.10	Cost estimates documented	[ ]		

Documentation Requirements

#	Checkpoint	Status	Owner	Notes
4.1	README with pipeline overview	[ ]		
4.2	Configuration documentation	[ ]		
4.3	Data dictionary updated	[ ]		
4.4	Operational procedures documented	[ ]		
4.5	Troubleshooting guide created	[ ]		
4.6	Change history maintained	[ ]		
4.7	Data lineage documented	[ ]		
4.8	Business context documented	[ ]		

Data Quality Monitoring

Data Quality Dimensions

Dimension	Definition	Example Checks	Priority
Completeness	Required data elements are present	Null checks, required field validation	Critical
Accuracy	Data correctly represents reality	Range checks, reference validation	Critical
Consistency	Data is uniform across systems	Cross-system reconciliation	High
Timeliness	Data arrives within expected window	Freshness checks, SLA monitoring	High
Validity	Data conforms to defined formats	Format validation, domain checks	High
Uniqueness	No unintended duplicates	Duplicate detection, key validation	Medium
Integrity	Relationships are maintained	Referential integrity checks	Medium

Completeness **Accuracy** **Consistency** **Timeliness** **Validity** **Uniqueness** **Integrity**

Data Quality Rules Template

Rule ID	Rule Name	Dimension	Table/Dataset	Column(s)	Logic	Threshold	Severity	Alert Channel
DQ-001	Primary Key Uniqueness	Uniqueness			COUNT(*)= COUNT(DISTINCT pk)	100%	Critical	PagerDuty
DQ-002	Null Check - Required Fields	Completeness			NULL count = 0	100%	Critical	PagerDuty
DQ-003	Valid Date Range	Validity			date BETWEEN min AND max	99.9%	High	Slack
DQ-004	Referential Integrity	Integrity			All FKs exist in parent	100%	Critical	PagerDuty
DQ-005	Valid Email Format	Validity			Regex match	95%	Medium	Email
DQ-006	Numeric Range Check	Accuracy			value BETWEEN min AND max	99%	High	Slack
DQ-007	Freshness Check	Timeliness			max(timestamp) > threshold	100%	Critical	PagerDuty
DQ-008	Row Count Variance	Completeness			Variance < threshold	+/- 20%	High	Slack
DQ-009	Schema Validation	Validity			Schema matches expected	100%	Critical	PagerDuty
DQ-010	Cross-System Reconciliation	Consistency			Source count = Target count	100%	High	Slack

Data Quality Monitoring Matrix

Check Type	Real-Time	Batch	Frequency	Tool Examples
Schema validation	Yes	Yes	Every run	Great Expectations, dbt
Null/completeness	Yes	Yes	Every run	Great Expectations, Soda
Uniqueness	No	Yes	Daily	dbt tests, custom SQL
Referential integrity	No	Yes	Daily	dbt tests, custom SQL
Statistical profiling	No	Yes	Daily/Weekly	Monte Carlo, Anomalo
Anomaly detection	Yes	Yes	Continuous	Monte Carlo, Databand
Freshness monitoring	Yes	Yes	Continuous	Elementary, custom
Volume monitoring	Yes	Yes	Every run	Datadog, custom

Data Quality Dashboard Metrics

Metric	Description	Target	Alert Threshold
Overall Quality Score	Weighted average across dimensions	> 98%	< 95%
Critical Rule Pass Rate	% of critical rules passing	100%	< 100%
Tables with Issues	Count of tables failing checks	0	> 0
Data Freshness	Time since last update	< SLA	> SLA
Schema Drift Incidents	Unexpected schema changes	0	> 0
Quality Trend	Week-over-week quality change	Stable/Improving	Declining

Performance Monitoring

Pipeline Performance Metrics

Metric	Description	How to Measure	Target	Alert Threshold
Execution Time	Total pipeline run duration	End time - Start time	< SLA	> 1.5x baseline
Throughput	Records processed per second	Records / Duration	> baseline	< 0.5x baseline
Latency	Time from source to destination	Target timestamp - Source timestamp	< SLA	> SLA
Resource Utilization	CPU, memory, disk usage	System metrics	60-80%	> 90%
Queue Depth	Messages waiting to process	Queue metrics	< threshold	> threshold
Error Rate	Failed records / Total records	Error count / Total	< 0.1%	> 1%
Retry Rate	Retried operations / Total	Retry count / Total	< 5%	> 10%

Performance Monitoring Checklist

#	Checkpoint	Status	Owner	Notes
5.1	Baseline performance metrics established	[ ]		
5.2	Execution time monitoring configured	[ ]		
5.3	Throughput tracking implemented	[ ]		
5.4	Resource utilization monitored	[ ]		
5.5	Query performance tracked	[ ]		
5.6	Network latency monitored	[ ]		
5.7	Memory usage alerts configured	[ ]		
5.8	Disk I/O monitoring enabled	[ ]		
5.9	Connection pool metrics tracked	[ ]		
5.10	Performance trends visualized	[ ]		

Performance Baseline Template

Pipeline	Metric	Baseline	Min Acceptable	Max Acceptable	Measurement Period
	Execution Time				
	Records/Second				
	CPU Usage %				
	Memory Usage %				
	Error Rate %				

Query Performance Tracking

Query ID	Description	Avg Duration	P95 Duration	P99 Duration	Data Scanned	Optimization Status

Reliability & Availability

Reliability Metrics

Metric	Definition	Formula	Target	Measurement
Availability	% time pipeline is operational	(Total Time - Downtime) / Total Time	99.9%	Monthly
Success Rate	% of runs completing successfully	Successful Runs / Total Runs	99.5%	Weekly
MTBF	Mean Time Between Failures	Total Uptime / Number of Failures	> 168 hrs	Monthly
MTTR	Mean Time To Recovery	Total Downtime / Number of Incidents	< 30 min	Monthly
Data Loss	Records lost or corrupted	Lost Records / Total Records	0%	Per run
Recovery Point	Data age at recovery	Time since last good state	< 1 hour	Per incident

Reliability Checklist

#	Checkpoint	Status	Owner	Notes
6.1	High availability architecture implemented	[ ]		
6.2	Failover mechanisms tested	[ ]		
6.3	Data backup strategy configured	[ ]		
6.4	Point-in-time recovery enabled	[ ]		
6.5	Circuit breakers implemented	[ ]		
6.6	Rate limiting configured	[ ]		
6.7	Dependency health checks active	[ ]		
6.8	Graceful degradation defined	[ ]		
6.9	Chaos testing performed	[ ]		
6.10	Disaster recovery plan tested	[ ]		

Failure Mode Analysis Template

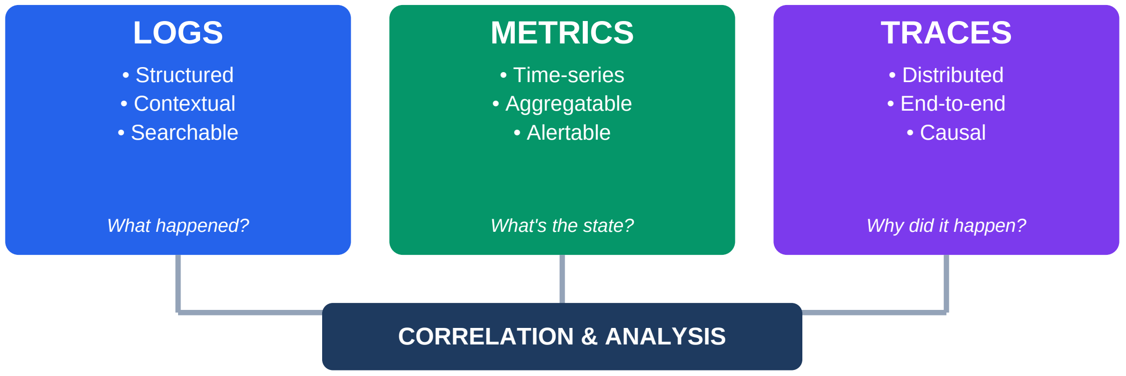
Failure Mode	Probability	Impact	Detection Method	Mitigation	Recovery Action
Source system unavailable	Medium	High	Health check	Retry + fallback	Backfill from replica
Network timeout	Medium	Medium	Connection monitoring	Retry with backoff	Automatic retry
Schema change	Low	Critical	Schema validation	Schema registry	Manual intervention
Data volume spike	Medium	Medium	Volume monitoring	Auto-scaling	Pause + capacity add
Corrupt data	Low	High	Quality checks	Quarantine	Reprocess from source
Resource exhaustion	Medium	High	Resource monitoring	Auto-scaling	Scale + restart
Dependency failure	Medium	High	Dependency health	Circuit breaker	Graceful degradation

Dependency Health Monitoring

Dependency	Type	Health Check	Frequency	Timeout	Fallback
Source Database	Database	SELECT 1	30s	5s	Read replica
API Endpoint	HTTP	GET /health	60s	10s	Cache
Message Queue	Queue	Connection check	30s	5s	Dead letter
Target Warehouse	Database	Connection check	60s	10s	Buffer to S3
Metadata Store	Database	SELECT 1	60s	5s	Local cache

Observability Framework

Three Pillars of Observability



Logging Standards

Log Field	Type	Required	Description	Example
timestamp	ISO 8601	Yes	Event time	2026-01-15T10:30:00.000Z
level	string	Yes	Severity level	INFO, WARN, ERROR, DEBUG
service	string	Yes	Service name	customer-pipeline
pipeline_id	string	Yes	Pipeline identifier	pl-customer-daily-001
run_id	string	Yes	Execution run ID	run-2026-01-15-001
trace_id	string	Yes	Distributed trace ID	abc123def456
span_id	string	Yes	Span identifier	span-789
message	string	Yes	Log message	Processing batch 5 of 10
environment	string	Yes	Deployment environment	production
records_processed	number	No	Count processed	10000
duration_ms	number	No	Operation duration	1250
error_type	string	No	Error classification	ValidationError
error_message	string	No	Error details	Invalid date format

Logging Checklist

#	Checkpoint	Status	Owner	Notes
7.1	Structured logging implemented (JSON)	[ ]		
7.2	Log levels used appropriately	[ ]		
7.3	Correlation IDs propagated	[ ]		
7.4	PII/sensitive data excluded	[ ]		
7.5	Log retention policy configured	[ ]		
7.6	Log aggregation centralized	[ ]		
7.7	Log search and analysis enabled	[ ]		
7.8	Log-based alerts configured	[ ]		

Metrics Standards

Metric Type	Naming Convention	Example	Use Case
Counter	{service}_{action}_total	pipeline_records_processed_total	Total counts that only increase
Gauge	{service}_{measurement}	pipeline_queue_depth	Values that go up and down
Histogram	{service}_{action}_duration_seconds	pipeline_execution_duration_seconds	Distribution of values
Summary	{service}_{measurement}_quantile	pipeline_latency_quantile	Pre-calculated percentiles

Core Pipeline Metrics

Metric Name	Type	Labels	Description
pipeline_runs_total	Counter	pipeline, status, environment	Total pipeline executions
pipeline_records_processed_total	Counter	pipeline, stage, environment	Records processed
pipeline_records_failed_total	Counter	pipeline, error_type, environment	Failed records
pipeline_execution_duration_seconds	Histogram	pipeline, environment	Execution time distribution
pipeline_data_freshness_seconds	Gauge	pipeline, dataset, environment	Seconds since last update
pipeline_queue_depth	Gauge	pipeline, queue, environment	Messages waiting
pipeline_active_jobs	Gauge	pipeline, environment	Currently running jobs
pipeline_last_success_timestamp	Gauge	pipeline, environment	Unix timestamp of last success

Tracing Implementation

Trace Component	Purpose	Implementation
Trace ID	Unique identifier for entire flow	Generate at entry, propagate everywhere
Span ID	Unique identifier for operation	Generate per operation
Parent Span ID	Link child to parent	Reference parent in child
Operation Name	What the span represents	descriptive, hierarchical
Tags	Key-value metadata	pipeline, environment, version
Logs	Events within span	Important milestones
Baggage	Cross-service context	user_id, tenant_id

Observability Tool Stack

Category	Tool Options	Selection Criteria
Log Aggregation	Elasticsearch, Splunk, Datadog, CloudWatch	Volume, cost, search needs
Metrics	Prometheus, Datadog, CloudWatch, Grafana Cloud	Cardinality, retention, cost
Tracing	Jaeger, Zipkin, Datadog APM, AWS X-Ray	Language support, sampling
Dashboards	Grafana, Datadog, New Relic, Kibana	Integration, sharing, alerts
Data Observability	Monte Carlo, Soda, Great Expectations, Elementary	Data-specific needs, coverage

Alerting Strategy

Alert Severity Levels

Severity	Definition	Response Time	Notification Channel	Examples
P1 - Critical	Production down, data loss imminent	< 15 minutes	PagerDuty + Phone	Pipeline failure, data corruption
P2 - High	Significant impact, SLA at risk	< 1 hour	PagerDuty + Slack	Performance degradation, quality failure
P3 - Medium	Minor impact, workaround available	< 4 hours	Slack	Warning thresholds, anomalies
P4 - Low	Informational, no immediate action	Next business day	Email + Ticket	Trends, optimization opportunities

P1
Critical
< 15 min

P2
High
< 1 hour

P3
Medium
< 4 hours

P4
Low
Next day

Alert Design Principles

Principle	Description	Implementation
Actionable	Every alert should trigger a specific action	Include runbook link, context
Relevant	Alert the right people at the right time	Use routing rules, schedules
Timely	Detect and notify quickly	Real-time monitoring, low latency
Accurate	Minimize false positives	Tune thresholds, use anomaly detection
Contextual	Provide enough information to act	Include metrics, logs, suggestions
Unique	Avoid duplicate alerts for same issue	Deduplication, aggregation

Alert Configuration Template

Alert Name	Metric/Condition	Threshold	Duration	Severity	Channel	Runbook
Pipeline Failure	pipeline_runs_total{status="failed"}	> 0	0m	P1	PagerDuty	RB-001
Long Running Job	pipeline_execution_duration_seconds	> 2x baseline	5m	P2	Slack	RB-002
Data Freshness	pipeline_data_freshness_seconds	> SLA	5m	P1	PagerDuty	RB-003
High Error Rate	error_rate	> 1%	5m	P2	Slack	RB-004
Quality Check Failure	quality_checks_failed	> 0	0m	P2	Slack	RB-005
Resource Exhaustion	cpu_utilization	> 90%	10m	P2	Slack	RB-006
Queue Backup	queue_depth	> 10000	5m	P2	Slack	RB-007
Dependency Down	dependency_health	!= healthy	2m	P1	PagerDuty	RB-008

Alert Routing Matrix

Alert Category	Business Hours	After Hours	Weekend
P1 - Critical	On-call Primary	On-call Primary	On-call Primary
P2 - High	On-call Primary	On-call Primary	On-call Secondary
P3 - Medium	Team Slack Channel	Email Queue	Email Queue
P4 - Low	Email Queue	Email Queue	Email Queue

Alert Checklist

#	Checkpoint	Status	Owner	Notes
8.1	Alert severity levels defined	[ ]		
8.2	On-call rotation established	[ ]		
8.3	Escalation paths documented	[ ]		
8.4	Alert routing configured	[ ]		
8.5	Runbooks linked to alerts	[ ]		
8.6	Alert deduplication enabled	[ ]		
8.7	Alert history retention configured	[ ]		
8.8	Alert acknowledgment workflow defined	[ ]		
8.9	Alert fatigue review scheduled	[ ]		
8.10	Alert testing procedures established	[ ]		

SLA & SLO Templates

SLA Definition Template

SLA Component	Definition
Service Name	
Pipeline(s) Covered	
Service Owner	
Customer/Stakeholder	
Effective Date	
Review Frequency	

Service Level Objectives

SLO ID	Metric	Target	Measurement Window	Consequence
SLO-001	Availability	99.9%	Monthly	Credit
SLO-002	Data Freshness	< 1 hour	Per batch	Notification
SLO-003	Data Quality Score	> 99%	Daily	Review
SLO-004	Completeness	100%	Per batch	Reprocessing
SLO-005	Latency (p99)	< 4 hours	Weekly	Investigation

Error Budget

SLO	Target	Error Budget (Monthly)	Current Usage	Status
Availability 99.9%	99.9%	43.2 minutes		
Freshness < 1 hour	99%	7.2 hours		
Quality > 99%	99%	1% of records		

SLO Examples by Pipeline Type

Pipeline Type	Metric	SLO Target	Rationale
Real-time Streaming	Latency p99	< 5 seconds	User-facing applications
	Availability	99.99%	Critical path
	Error rate	< 0.1%	Data integrity
Batch ETL - Critical	Completion time	By 6 AM	Downstream dependencies
	Data quality	> 99.9%	Financial reporting
	Availability	99.9%	Business continuity
Batch ETL - Standard	Completion time	Within SLA window	Operational needs
	Data quality	> 99%	Analytics quality
	Availability	99.5%	Acceptable risk
ML Feature Pipeline	Freshness	< 1 hour	Model relevance
	Completeness	100%	Model accuracy
	Latency	< 15 minutes	Training schedules

SLI Measurement Methods

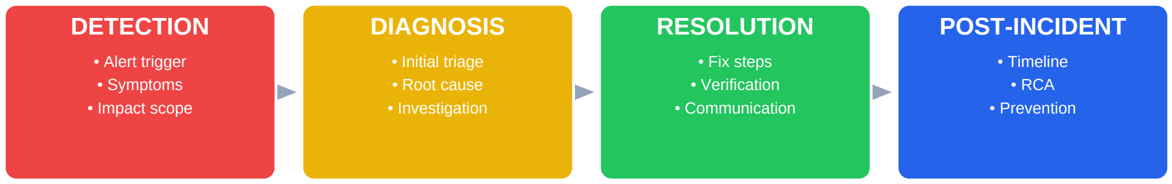
SLI	Measurement Method	Data Source	Calculation
Availability	Successful runs / Total runs	Orchestrator logs	(successful / total) * 100
Latency	Time from trigger to completion	Pipeline metrics	p99(end_time - start_time)
Freshness	Time since last data update	Target metadata	now() - max(updated_at)
Error Rate	Failed records / Total records	Quality metrics	(errors / total) * 100
Throughput	Records processed per time	Pipeline metrics	sum(records) / time_window

SLA Reporting Template

Period	Pipeline	SLO	Target	Actual	Status	Notes
		Availability				
		Freshness				
		Quality				
		Latency				

Incident Response Runbooks

Runbook Template Structure



RB-001: Pipeline Failure

Runbook ID: RB-001

Title: Pipeline Execution Failure

Severity: P1 - Critical

Last Updated: November 2026

Detection

Attribute	Details
Alert Name	Pipeline Failure
Metric	pipeline_runs_total{status="failed"}
Threshold	> 0
Notification	PagerDuty - Primary On-call

Impact Assessment

Question	Action
Which pipelines are affected?	Check orchestrator dashboard
What downstream systems depend on this data?	Reference dependency map
What is the SLA impact?	Check SLA dashboard
Are other pipelines at risk?	Review shared resources

Diagnosis Steps

Step	Action	Expected Outcome
1	Check orchestrator logs	Identify failure point
2	Review pipeline logs	Find error message
3	Check dependency health	Confirm external system status
4	Verify resource availability	Confirm capacity
5	Check recent changes	Identify deployment correlation

Common Causes & Resolutions

Cause	Symptoms	Resolution
Source unavailable	Connection timeout	Verify source health, failover to replica
Schema change	Parsing errors	Update schema mapping, notify data owner
Resource exhaustion	OOM, timeout	Scale resources, optimize query
Bad data	Validation failures	Quarantine records, notify upstream
Code bug	Unexpected errors	Rollback, hotfix
Permission issue	Access denied	Verify credentials, IAM

Resolution Steps

Step	Action	Verification
1	Identify root cause	Error categorized
2	Apply appropriate fix	Fix deployed
3	Restart pipeline	Run initiates
4	Monitor execution	Progress visible
5	Verify completion	Success status
6	Validate data quality	Quality checks pass
7	Confirm downstream receipt	Consumers acknowledge

Escalation

Condition	Escalate To	Timeframe
Unable to diagnose in 30 min	Secondary on-call	Immediate
Multiple pipelines affected	Engineering Manager	15 minutes
SLA breach imminent	Data Platform Lead	Immediate
Data loss confirmed	CDO / Stakeholders	Immediate

RB-002: Long Running Pipeline

Runbook ID: RB-002

Title: Pipeline Exceeds Expected Duration

Severity: P2 - High

Last Updated: November 2026

Detection

Attribute	Details
Alert Name	Long Running Job
Metric	pipeline_execution_duration_seconds
Threshold	> 2x baseline
Duration	5 minutes

Diagnosis Steps

Step	Action	Expected Outcome
1	Check current stage	Identify bottleneck
2	Review resource utilization	CPU, memory, I/O
3	Check data volume	Compare to typical
4	Review query execution plans	Identify slow queries
5	Check for blocking/locks	Database contention

Common Causes & Resolutions

Cause	Resolution
Data volume spike	Scale resources, partition data
Inefficient query	Optimize query, add indexes
Resource contention	Reschedule, increase resources
External API slowdown	Increase timeout, add caching
Missing index	Add/rebuild index

RB-003: Data Freshness Alert

Runbook ID: RB-003

Title: Data Freshness Exceeds Threshold

Severity: P1 - Critical

Last Updated: November 2026

Detection

Attribute	Details
Alert Name	Data Freshness
Metric	pipeline_data_freshness_seconds
Threshold	> SLA threshold
Duration	5 minutes

Diagnosis Steps

Step	Action	Expected Outcome
1	Check if pipeline is running	Verify orchestrator
2	Check last successful run	Identify gap
3	Review pipeline status	Running, failed, pending
4	Check upstream data	Source freshness
5	Verify scheduling	Cron/trigger active

Resolution Decision Tree

RB-004: Data Quality Failure

Runbook ID: RB-004

Title: Data Quality Check Failed

Severity: P2 - High

Last Updated: November 2026

Detection

Attribute	Details
Alert Name	Quality Check Failure
Metric	quality_checks_failed
Threshold	> 0
Notification	Slack - Data Quality Channel

Diagnosis Steps

Step	Action	Expected Outcome
1	Identify failed checks	List from monitoring
2	Review sample failed records	Understand pattern
3	Check source data	Upstream quality
4	Review recent changes	Code, schema, config
5	Assess impact scope	Records, consumers

Quality Issue Response Matrix

Issue Type	Severity	Response	Notification
Null in required field	High	Reject records, notify source	Data Owner
Duplicate records	Medium	Deduplicate, investigate source	Data Steward
Invalid format	Medium	Transform or reject	Data Engineer
Anomalous values	High	Quarantine, investigate	Data Quality Team
Schema mismatch	Critical	Halt pipeline, assess	Platform Team

RB-005: Resource Exhaustion

Runbook ID: RB-005

Title: Resource Utilization Critical

Severity: P2 - High

Last Updated: November 2026

Detection

Attribute	Details
Alert Name	Resource Exhaustion
Metrics	cpu_utilization, memory_utilization, disk_utilization
Threshold	> 90%
Duration	10 minutes

Immediate Actions

Resource	Immediate Action	Long-term Fix
CPU	Scale up/out	Optimize processing
Memory	Restart with larger instance	Fix memory leaks, optimize
Disk	Clear temp files, expand	Archive, retention policy
Network	Rate limit, queue	Optimize payloads, CDN

Incident Communication Template

Subject: [SEVERITY] Pipeline Incident - [Brief Description]

Section	Content
Status	Investigating / Identified / Monitoring / Resolved
Impact	Description of user/business impact
Affected Pipelines	List of pipelines
Started	Timestamp
Current Status	What we know
Next Update	Expected time
Actions Taken	What we've done
Contact	On-call engineer

Cost Monitoring

Cost Metrics to Track

Cost Category	Metrics	Tracking Method	Review Frequency
Compute	CPU hours, instance hours	Cloud billing API	Daily
Storage	GB stored, storage class	Cloud billing API	Weekly
Data Transfer	GB transferred, egress	Cloud billing API	Daily
Query Processing	Bytes scanned, slots used	Platform metrics	Daily
Orchestration	DAG runs, task instances	Platform metrics	Weekly
Monitoring	Metrics ingested, logs stored	Tool billing	Monthly

Cost Monitoring Checklist

#	Checkpoint	Status	Owner	Notes
9.1	Cost allocation tags implemented	[ ]		
9.2	Budget alerts configured	[ ]		
9.3	Cost dashboards created	[ ]		
9.4	Anomaly detection for cost spikes	[ ]		
9.5	Resource utilization tracking	[ ]		
9.6	Idle resource identification	[ ]		
9.7	Cost per pipeline tracked	[ ]		
9.8	Cost optimization opportunities identified	[ ]		

Cost Optimization Opportunities

Opportunity	Potential Savings	Effort	Priority
Right-size compute resources	20-40%	Medium	High
Use spot/preemptible instances	60-80%	Medium	High
Optimize query patterns	30-50%	High	High
Implement data lifecycle policies	20-30%	Low	Medium
Compress data in transit	10-20%	Low	Medium
Consolidate pipelines	15-25%	High	Medium
Schedule non-critical jobs off-peak	10-20%	Low	Low

Cost Allocation Template

Pipeline	Compute	Storage	Transfer	Processing	Monitoring	Total	Budget	Variance
Total								

Security & Compliance Monitoring

Security Monitoring Checklist

#	Checkpoint	Status	Owner	Notes
10.1	Access logging enabled	[ ]		
10.2	Failed authentication alerts	[ ]		
10.3	Privileged action monitoring	[ ]		
10.4	Data access audit trail	[ ]		
10.5	Encryption at rest verified	[ ]		
10.6	Encryption in transit verified	[ ]		
10.7	Secrets rotation automated	[ ]		
10.8	Vulnerability scanning scheduled	[ ]		
10.9	Security incident alerting	[ ]		
10.10	Compliance reporting automated	[ ]		

Security Metrics

Metric	Description	Target	Alert Threshold
Failed login attempts	Authentication failures	< 5/hour	> 10/hour
Unauthorized access attempts	Denied data access	0	> 0
Secrets age	Days since rotation	< 90 days	> 90 days
Unencrypted data transfers	Non-TLS connections	0	> 0
Privileged operations	Admin actions count	Baseline	> 2x baseline
Data export events	Bulk data exports	Baseline	> 2x baseline

Compliance Monitoring Matrix

Regulation	Monitoring Requirement	Implementation	Frequency
GDPR	Data access logging	Audit logs	Continuous
	Data retention compliance	Retention policies	Daily
	Cross-border transfer tracking	Transfer logs	Continuous
HIPAA	PHI access logging	Audit logs	Continuous
	Access control verification	Access reviews	Quarterly
	Encryption validation	Security scans	Weekly
SOX	Change management	Change logs	Continuous
	Access controls	Access reviews	Quarterly
	Audit trails	Audit logs	Continuous

Data Privacy Monitoring

Check	Description	Frequency	Alert On
PII Detection	Scan for unexpected PII	Daily	PII found in non-approved locations
Consent Compliance	Verify data use matches consent	Per run	Usage outside consent scope
Retention Compliance	Check data age vs retention policy	Daily	Data past retention period
Access Rights	Verify access matches approved list	Weekly	Unauthorized access patterns

Implementation Roadmap

Phase 1: Foundation (Weeks 1-4)

Week	Activity	Deliverable	Owner	Status
1	Assess current monitoring state	Assessment report	Platform Lead	
1-2	Define monitoring requirements	Requirements document	Data Engineering	
2	Select monitoring tools	Tool selection matrix	Platform Team	
2-3	Implement centralized logging	Log aggregation	Platform Team	
3-4	Deploy basic metrics collection	Core dashboards	Platform Team	
4	Configure critical alerts	P1 alerts active	On-call Lead	

Phase 1 Success Criteria:

- ■ Logging centralized for all pipelines
- ■ Core metrics dashboards available
- ■ P1 alerts configured and tested
- ■ On-call rotation established

Phase 2: Data Quality (Weeks 5-8)

Week	Activity	Deliverable	Owner	Status
5	Define data quality rules	Rule library	Data Quality Lead	
5-6	Implement quality checks	Automated checks	Data Engineers	
6-7	Create quality dashboards	Quality scorecards	Data Quality Lead	
7-8	Configure quality alerts	Quality alerts active	On-call Lead	
8	Document quality runbooks	Runbook library	Data Quality Lead	

Phase 2 Success Criteria:

- ■ Quality rules defined for critical pipelines
- ■ Automated quality checks running
- ■ Quality dashboards operational
- ■ Quality incident runbooks documented

Phase 3: Advanced Observability (Weeks 9-12)

Week	Activity	Deliverable	Owner	Status
9	Implement distributed tracing	Tracing active	Platform Team	
9-10	Set up anomaly detection	ML-based alerts	Platform Team	
10-11	Create SLO dashboards	SLO tracking	Platform Lead	
11-12	Implement cost monitoring	Cost dashboards	FinOps	
12	Conduct monitoring review	Optimization plan	Platform Lead	

Phase 3 Success Criteria:

- ■ End-to-end tracing operational
- ■ Anomaly detection reducing false positives
- ■ SLOs tracked and reported
- ■ Cost monitoring in place

Appendices

Appendix A: Monitoring Tool Comparison

Tool	Category	Strengths	Considerations	Pricing Model
Datadog	Full stack	Comprehensive, easy setup	Cost at scale	Per host/metrics
Prometheus + Grafana	Metrics	Open source, flexible	Self-managed	Free (infra costs)
Monte Carlo	Data observability	ML-powered, data-focused	Data platform cost	Per table
Great Expectations	Data quality	Open source, customizable	Maintenance	Free (infra costs)
PagerDuty	Alerting	Robust routing, integrations	Per user cost	Per user
Soda	Data quality	SQL-based, intuitive	Limited ML	Per checks
Elementary	dbt observability	dbt-native, free tier	dbt-specific	Freemium

Appendix B: Metric Naming Conventions

Convention	Example	Rationale
Use snake_case	pipeline_records_total	Consistency, readability
Prefix with service	etl_pipeline_duration	Namespace isolation
Suffix with unit	duration_seconds	Clarity
Use _total for counters	records_processed_total	Prometheus convention
Use descriptive names	customer_pipeline_freshness	Self-documenting

Appendix C: Log Level Guidelines

Level	When to Use	Examples
DEBUG	Detailed diagnostic info	Variable values, loop iterations
INFO	Normal operation events	Job started, batch processed
WARN	Potentially harmful situations	Retry occurred, approaching limit
ERROR	Error events, job continues	Record failed, recoverable error
FATAL	Severe errors, job must stop	Cannot connect, data corruption

Appendix D: Alert Template Examples

Prometheus Alert Rule

```
groups:
  - name: pipeline_alerts
    rules:
      - alert: PipelineFailure
        expr: pipeline_runs_total{status="failed"} > 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: "Pipeline {{ $labels.pipeline }} failed"
          description: "Pipeline {{ $labels.pipeline }} has failed in {{ $labels.environment }}"
          runbook_url: "https://runbooks.example.com/RB-001"
```

Datadog Monitor

```
{
  "name": "Pipeline Execution Failure",
  "type": "metric alert",
  "query": "sum:pipeline.runs.failed{*}.as_count() > 0",
  "message": "Pipeline failure detected.\n\nPipeline: {{pipeline.name}}\nEnvironment: {{env}}\n\nRunbook: https:/
/runbooks.example.com/RB-001",
  "tags": ["pipeline", "p1"],
  "priority": 1
}
```

Appendix E: Dashboard Template Sections

Section	Metrics to Include	Visualization
Overview	Success rate, active jobs, last run	Stat panels, status
Performance	Duration trend, throughput, latency	Time series
Data Quality	Quality score, failed checks, trends	Gauge, time series
Resources	CPU, memory, disk, network	Time series, heatmap
Alerts	Active alerts, recent incidents	Table, status
SLOs	Error budget, SLI trends	Gauge, time series
Cost	Daily/monthly cost, trends	Time series, stat

Appendix F: Glossary

Term	Definition
DAG	Directed Acyclic Graph - workflow structure
DLQ	Dead Letter Queue - storage for failed messages
MTBF	Mean Time Between Failures
MTTR	Mean Time To Recovery
SLA	Service Level Agreement - contractual commitment
SLI	Service Level Indicator - metric measuring service
SLO	Service Level Objective - target for SLI
Observability	Ability to understand system state from outputs
Runbook	Document with steps to resolve incidents
Error Budget	Allowed unreliability before action required

About Enterprise Data Solutions

Enterprise Data Solutions is New Zealand's trusted partner for data engineering and platform development. We help organizations build robust, scalable, and observable data pipelines that power modern analytics and AI initiatives.

Our Services

Service	Description
Data Pipeline Development	Design and build reliable ETL/ELT pipelines
Platform Engineering	Implement modern data platforms on cloud
DataOps Implementation	Establish CI/CD, monitoring, and automation
Data Quality Programs	Implement quality frameworks and monitoring
Performance Optimization	Tune pipelines for speed and efficiency
Incident Response Setup	Design alerting, runbooks, and on-call processes

Why Choose Enterprise Data Solutions

- Deep expertise in data engineering and platform development
- Proven methodologies refined through real-world implementations
- Local presence with global perspective
- Tool-agnostic approach - we recommend what's right for you
- End-to-end capabilities from architecture to operations

Contact Us

Enterprise Data Solutions

Channel	Contact
Website	https://www.enterprisedatasolutions.co.nz
Email	Contact@enterprisedatasolutions.co.nz
Services	Data Engineering, Platform Development, DataOps, Monitoring

Schedule a Consultation

Ready to implement production-grade monitoring for your data pipelines? Contact us to discuss your organization's needs and how this checklist can be customized for your context.

Document Control

Attribute	Value
Document Title	The Data Pipeline Monitoring Checklist 2026
Version	2.0
Classification	Public
Prepared By	Enterprise Data Solutions
Copyright	2026 Enterprise Data Solutions. This template may be customized for organizational use.

This checklist is provided by Enterprise Data Solutions. Feel free to adapt it for your organization's specific requirements.