August 13, 2025

DATABRICKS DATA OBSERVABILITY BUYER'S GUIDE

The Definitive Data Platform Leader's Guide to Selecting the Right Fit for Your Needs

unravel

Introduction

The observability market is at an inflection point. According to Gartner's 2024 Market Guide for Data Observability Tools:

- By 2026, 50% of enterprises implementing distributed data architectures will have adopted data observability tools, up from less than 20% in 2024
- Traditional, static, event-based monitoring is insufficient for managing complex data systems effectively
- Enterprises now demand comprehensive solutions that provide real-time visibility, proactive alerting, and actionable recommendations, not just isolated metrics. Yet, the market is currently fragmented, with no standard definition for data observability
- Most vendors focus on specific areas of observation rather than providing comprehensive solutions

This guide helps you cut through the noise and select a solution that delivers measurable business impact across all critical observability domains.

Understanding Data Observability

Data observability goes beyond traditional monitoring - it's the foundation for proactive data operations.

It enables organizations to:

- Understand the state and health of their data ecosystems
- Continuously monitor data pipelines, quality, and infrastructure health
- Alert and troubleshoot issues proactively
- Analyze and optimize performance and costs
- Prevent data errors and system downtime before they impact business decisions

Gartner identifies five key observation areas

Data Content

Data quality, anomaly detection, schema changes

Data Flow and Pipeline

Pipeline monitoring, execution tracking, broken pipeline detection

Infrastructure and Compute

Resource monitoring, workload analysis, performance tracking

User, Usage, and Utilization

User behavior, access patterns, query analysis

Financial Allocation

Cost monitoring, spend analysis, resource optimization

Your observability solution options

While the need for effective observability solutions is clear, the exact nature and form of observability required for managing modern data platforms is decidedly opaque. The term "observability" can refer to a wide ranging landscape of home grown and off-the-shelf point solutions, each with a different area of focus. This analysis will focus on six of the most common types of tooling commonly considered for modern data platform observability purposes:

1. Do it Yourself (DIY) Homegrown Solutions
2. Generic FinOps Tools
3. Generic DevOps Observability Tools
4. Databricks Native Tools
5. Point Solutions for Data Quality and Observability
6. AI-Native Data Observability and FinOps

1

Do-It-Yourself (DIY) Homegrown Solutions

Custom-built monitoring using open-source technologies (like Prometheus, Grafana, ELK stack) combined with stock BI tools (like Tableau, Power BI) often struggles to scale with Databricks' complexity. While attractive for specialized needs, most enterprises outgrow DIY as data volumes and performance requirements increase.

Best Fit
Large engineering teams with specialized resources prioritizing control over cost, pace of innovation, and convenience.

Strengths

Complete customization

- Create custom dashboards for specific Databricks cluster metrics
- Implement organization-specific cost allocation rules

No additional licensing costs

- Use Grafana for visualization and Prometheus for metrics collection
- Implement custom scripts using Databricks' REST APIs

Complete control and flexibility over implementation

- Ability to modify monitoring parameters as business needs change
- Freedom to integrate with internal systems using custom APIs

Tailored to niche business requirements

- Industry-specific data quality monitoring
- Custom charge-back models for different business units

Weaknesses

Requires significant internal engineering resources

- A Dedicated team is needed for development and maintenance, often costing more in FTEs than a commercial solution
- Specialized knowledge of both monitoring tools and Databricks internals

Limited scalability

- Custom code and dashboards often break when cluster sizes increase dramatically
- Difficult to adapt to multi-region or multi-cloud deployments

Lack of advanced analytical capabilities

- No built-in machine learning for anomaly detection, sophisticated predictive recommendations
- Manual root cause analysis is required for most issues

High maintenance drag

- Constant upgrades needed as Databricks releases new features
- Troubleshooting broken data pipelines often requires checking multiple systems

Fragmented visibility:

- Siloed tools for cost, performance, and data quality
- No unified view across all five observation areas

Real-world scenario

A financial services company built its monitoring solution using Grafana dashboards and custom Python scripts. While initially effective, maintenance demands skyrocketed as their Databricks usage expanded from 10 to 100 clusters, ultimately spending more on engineering resources than a commercial solution would have cost. This mirrors industry trends: most enterprises outgrow DIY as Databricks' complexity increases, leading to higher costs and slower innovation.

2

Generic FinOps Tools

Cloud cost management platforms are designed for the financial governance of cloud resources across multiple providers. These tools offer unified cost tracking, budget controls, and compliance features, making them a popular choice for organizations seeking high-level cloud cost visibility. Examples: CloudHealth, IBM Apptio Cloudability, Flexera, Kubecost, CloudZero.

Best Fit
Organizations that require only high-level cloud cost visibility without the need for more granular visibility and ongoing optimization.

Strengths

Cost tracking and financial management

- High-level tracking of overall Databricks spend by cluster
- Historical spend analysis with forecasting capabilities

Multi-cloud support

- Unified view of costs across AWS, Azure, and GCP Databricks deployments
- Cross-cloud cost optimization or migration recommendations

Mature financial reporting

- Automated monthly billing reports for chargeback and showback
- Budget alerts and anomaly detection for spending

Governance and compliance

- Role-based access control for financial data
- Audit trails for spending approvals and budget planning

Integration with financial systems

- Export capabilities for ERP systems
- Automated invoice reconciliation

Weaknesses

Limited Databricks-specific insights

- Unable to correlate Spark job failures with increased costs
- No understanding of Databricks-specific concepts like Photon, Delta Lake optimizations

Lack of deep technical performance analysis

- Can't identify inefficient queries driving up costs
- Unable to recommend performance tuning opportunities

Minimal data quality monitoring

- No visibility into data corruption issues
- Cannot detect schema drift or data pipeline failures that can result in costly downstream delays

Superficial observability

- Only tracks VM-level costs, not job-level efficiency
- No visibility into user behavior patterns

Reactive rather than proactive

- Identifies cost spikes after they occur
- Limited predictive capabilities for resource planning

Real-world scenario

A retail company implemented CloudHealth to track Databricks costs but found they still needed separate tools for performance monitoring and troubleshooting. When faced with escalating costs, they could see the increase but couldn't identify which specific workloads or users were causing the problem without additional tools. Ultimately, they had to deploy additional tools for performance monitoring and troubleshooting, highlighting the gap between high-level financial visibility and actionable, Databricks-specific insights.

3

Generic DevOps Observability Tools

Infrastructure and application monitoring platforms designed for DevOps teams to monitor overall IT infrastructure health, offering broad support for generic needs. While they excel at infrastructure monitoring, they are not tailored for the unique challenges of data platforms like Databricks. Examples: DataDog, Dynatrace, New Relic, AppDynamics, Splunk, Elastic Observability.

Best Fit
Teams prioritizing infrastructure optimization and reliability over optimization, reliability, and ongoing management of the Databricks platform.

Strengths

Comprehensive infrastructure monitoring

- Detailed metrics on CPU, memory, disk I/O for Databricks clusters
- Network flow visualization between components

Real-time performance tracking

- Live dashboards giving immediate visibility into cluster performance
- Latency tracking for critical processes

Robust alerting capabilities

- Customizable alert thresholds for key metrics
- Incident management and escalation workflows

APM capabilities

- Distributed tracing across microservices
- Code-level performance insights for some languages

Broad technology support

- Monitoring capabilities for databases, containers, and cloud services
- Extensibility through plugins and custom integrations

Weaknesses

Not specialized for data platforms

- Lack of deep understanding of Spark execution plans
- No insights into data skew or partition optimization are critical for performance tuning and avoiding cost wastage

Limited data-specific insights

- Cannot correlate data quality issues with performance problems
- No visibility into data lineage or impact analysis

Expensive for comprehensive coverage

- Agent-based pricing models become costly at scale
- Data volume-based pricing models are expensive for big data platforms

Complexity in the Databricks environment configuration

- Difficult to deploy agents on ephemeral clusters
- Challenging to maintain consistency across auto-scaling environments

Gaps in data platform context

- Limited understanding of Spark applications and jobs
- No visibility into Delta Lake operations or optimizations

Real-world scenario

A healthcare company deployed Dynatrace to monitor their Databricks environment, but struggled to correlate infrastructure metrics with data pipeline failures. While they could see when a cluster was under stress, they couldn't easily determine which specific data transformations were causing the problems or how to optimize them, ultimately requiring additional tools to close these gaps.

4

Databricks Native Tools

Built-in tools and features provided by Databricks for monitoring their platform, including cluster monitoring, query history, and Databricks SQL query monitoring for quick visibility into platform health and usage without additional deep observability or auto-optimizations.

Best Fit

Organizations using Databricks for ML/AIML primarily needing native data quality assurance rather than more comprehensive visibility, optimization, and reliability of the full Databricks platform and supporting infrastructure.

Strengths

Tight platform integration

- Native understanding of Databricks components
- No additional deployment or configuration needed

Basic performance and cost metrics

- Cluster utilization dashboards
- Job run histories and duration tracking

No additional tool procurement

- Included with Databricks licensing
- Simplified vendor management

Growing capabilities

- Recent enhancements include Lakehouse Monitoring for data quality and SQL warehouse monitoring
- Continuous enhancements with platform updates

Native understanding of Databricks concepts

- Built-in visibility into Delta Lake operation metrics
- Photon acceleration visibility

Weaknesses

Limited depth of observability

- Basic metrics without advanced correlation
- Minimal long-term historical trend analysis or multi-dimensional drill-downs

Lacks advanced predictive capabilities

- No AI-driven recommendations
- Limited forecasting and trend analysis

Minimal cross-platform insights

- No integrated view of upstream/downstream systems
- Isolated view of Databricks only and no external source integrations (e.g., cloud bills)

Basic alerting mechanisms

- Limited customization of alerts and no capability to build enterprise-grade workflows involving multiple third-party vendors
- Minimal integration with external notification systems

Fragmented tooling

- Different interfaces for cluster monitoring versus SQL monitoring
- No unified view across all observation areas

Real-world scenario

A manufacturing company relied on Databricks' native monitoring tools but found they couldn't effectively analyze long-term trends or get proactive recommendations. When troubleshooting performance issues, they had to piece together information from multiple interfaces manually, and they could not set up sophisticated alerts to identify potential problems.

5

Point Solutions for Data Quality and Observability

Specialized platforms focused primarily on data quality, content monitoring, and in some cases, pipeline observability. These tools are designed to automate the detection of data issues, ensure data integrity, and provide detailed insights into the health of the data. Examples: Monte Carlo, Acceldata, Soda, Bigeye, Anomalo, Lightup, Metaplane.

Best Fit
Teams that primarily need cross-platform data quality enforcement. It is often used in combination with more comprehensive, AI-native Data Observation and FinOps tools.

Strengths

Strong focus on data quality

- Automated detection of schema changes and data anomalies
- Statistical analysis of data distributions

Detailed data content monitoring

- Continuous tracking of null values and anomalies
- Data freshness and recency monitoring

Rule-based anomaly detection

- Custom business rules for data validation
- Automated testing of data integrity

Metadata management

- Tracking of data lineage
- Impact analysis for downstream dependencies

Specialized data profiling

- Column-level statistics and pattern recognition
- Trend analysis of data distributions

Weaknesses

Narrow observability scope

- Limited or no infrastructure monitoring
- Minimal cost optimization capabilities

Primarily focused on data content

- Most tools cover only 1-2 of Gartner's five observation areas
- Limited visibility into infrastructure and compute optimization

Limited infrastructure and cost optimization

- Cannot recommend resource sizing changes
- No visibility into idle cluster costs

Fragmented capabilities

- Need to combine multiple tools for comprehensive coverage
- Different tools for quality versus performance

Varying levels of Databricks integration

- Some tools have limited support for Delta Lake
- Varying depth of integration with Databricks features

Real-world scenario

A media company implemented Monte Carlo for data quality monitoring but still struggled with performance and cost issues in their Databricks environment. While they could detect data quality problems, they lacked visibility into which users were running inefficient queries or how to optimize their cluster configurations, ultimately requiring additional tools to address these gaps.

6

AI-Native Data Observability and FinOps

Purpose-built, AI-driven data observability platforms designed specifically for modern data platforms like Databricks, delivering unified, actionable insights across all Gartner-defined observability domains. Example: Unravel Data.

Best Fit

Large users of Databricks with mission-critical reliance requiring comprehensive, full-stack optimization, performance, and reliability, along with AI-native automation capabilities to drive actionability at scale.

Strengths

Comprehensive cross-domain visibility

- Unified coverage across all five Gartner observability areas
- Correlation of issues across multiple domains for faster resolution

AI-powered insights and automation

- Automatic root cause analysis for failed jobs and performance issues
- ML-based anomaly detection and pattern recognition
- Proactive notifications of potential issues before they impact users

Deep Databricks integration

- Support for Delta Lake, Photon engine, and Unity Catalog
- Optimization recommendations specific to Databricks environments
- Integration with Databricks workflows and jobs

Granular cost optimization

- Job-level and query-level cost attribution
- Specific recommendations with estimated savings
- Idle resource detection and elimination
- Cost tracking by business unit, department, and user

Actionable recommendations

- Concrete, implementable guidance with expected impact
- Query optimization suggestions with before/after comparisons
- Automated remediation capabilities
- Self-service optimization tools for users

Weaknesses

Higher initial investment

- Higher pricing compared to some point solutions or native tools
- Higher value proposition for large-scale Databricks users

Organizational adoption challenges

- May necessitate process changes to leverage capabilities fully
- Learning curve for maximizing platform value

Implementation complexity

- Configuration required for maximum value
- Integration with existing toolchains and workflows

Feature richness can be overwhelming

- Requires prioritization of which features to adopt first
- Training needed to leverage advanced capabilities

Evolving technology area

- Market definitions and maturing standards
- Vendor landscape continues to develop

Real-world scenario

A global financial services company implemented Unravel Data in their Databricks environment and achieved a 40% reduction in compute costs within three months by optimizing queries, cluster configurations, and reducing resource waste. The platform also helped identify data quality issues and link them to specific pipeline failures, avoiding downstream business impacts and significantly reducing mean time to resolution for critical incidents.

Key differentiators to a holistic approach to data observability

1. AI-Powered Insights
2. Comprehensive Coverage
3. Full-Spectrum Observability

1. AI-Powered Insights

Automatic root cause analysis for failed jobs

1. Predictive analytics for resource bottlenecks
2. ML-based anomaly detection for performance and cost
3. Automated recommendations for query optimization
4. Proactive notification of potential issues before they impact users

2. Comprehensive Coverage

All Five Critical Observation Areas

- Unifies all of Gartner's observability dimensions - data content, data flow, infrastructure, usage, and financial monitoring
- Correlates issues across multiple domains for faster resolution

End-to-End Visibility

- Traces data from ingestion through processing to consumption
- Correlates user activities with infrastructure performance
- Provides full context for troubleshooting

Support for Complex Architectures

- Multi-cloud and hybrid deployment support
- Integration with upstream and downstream systems
- Support for various Databricks deployment models

Proactive Issue Resolution

- Identifies potential bottlenecks before they impact users
- Recommends optimization strategies with expected impact
- Automates routine optimization tasks

3. Full-Spectrum Observability

1. **FinOps Optimization**
   - Granular cost attribution to users, departments, and applications
   - Idle resource detection and automatic scaling recommendations
   - Right-sizing recommendations for clusters and warehouses
   - Cost forecasting and budget planning based on historical patterns
   - Chargeback reporting with business context

2. **Application Performance Management**
   - End-to-end visibility into Spark job execution
   - Query-level performance analysis and optimization
   - Data skew detection and resolution strategies
   - Cache hit ratio optimization
   - SQL warehouse performance tuning

3. **Data Infrastructure Management**
   - Cluster configuration optimization
   - Resource utilization analysis across all workloads
   - Automated scaling recommendations
   - Bottleneck identification and remediation
   - Capacity planning and forecasting

4. **Dataset Usage Analytics**
   - Dataset-level access patterns and popularity metrics
   - Lineage mapping showing upstream and downstream dependencies
   - Impact analysis for proposed changes before they go into production
   - Storage optimization recommendations
   - Dataset access audit trails

5. **User Management and Optimization**
   - User behavior analytics identifies inefficient patterns
   - Resource consumption by user and job type
   - Query pattern analysis and optimization recommendations
   - Training opportunity identification based on user ratings
   - Self-service optimization recommendations for users

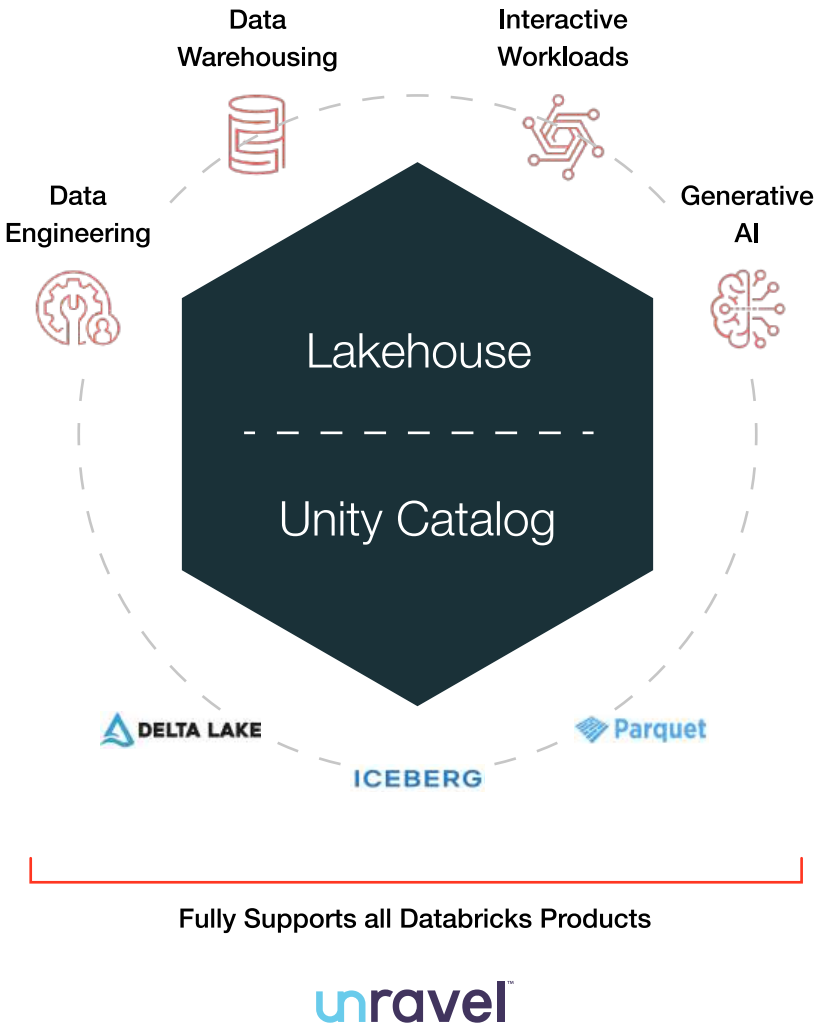
Unique Value Proposition for Databricks Environments

Deep Databricks Integration

- Native understanding of Databricks internals and architecture
- Support for Delta Lake, Photon engine, and Unity Catalog
- Insight into Databricks-specific optimization opportunities
- Integration with Databricks Workflows and Jobs
- Support for both interactive and automated workloads

Multi-Environment Monitoring

- Support for development, test, and production environments
- Cross-environment performance comparison
- Consistent metrics across all platforms
- Unified view across cloud providers
- Support for hybrid deployments



unravel

Selecting the right solution for you

To choose the right solution for your needs, evaluate each solution category against your specific requirements. For example, smaller organizations may be able to get by using existing generic DevOps tools like DataDog, while organizations with very large engineering teams and unique requirements may choose a DIY homegrown option. Most alternatives offer free trials or health check reporting to better understand ROI for your environments. The comparison matrix and evaluation framework can help you determine your best option.

Comparison Matrix

- Limited
- Moderate
- Basic
- Strong

	Data Content	Data Flow & Pipeline	Infrastructure & Compute	User, Usage & Utilization	Financial Allocation
DIY Solutions	Basic	Basic	Moderate	Limited	Basic
FinOps Tools	Limited	Limited	Basic	Limited	Strong
DevOps Tools	Limited	Moderate	Strong	Moderate	Moderate
Databricks Native	Moderate	Moderate	Moderate	Basic	Basic
Data Quality Tools	Strong	Moderate	Limited	Limited	Limited
Unravel Data	Strong	Strong	Strong	Strong	Strong

Recommendation Framework

When assessing data observability solutions for your Databricks environment, keep the following factors in mind.

1

Depth of observability

- How many of the five observation areas does the solution cover?
- What is the granularity of the insights provided?
- Can it correlate issues across different domains?
- Does it support chargeback and showback models?

2

AI and predictive capabilities

- Does the solution offer automated root cause analysis?
- Can it provide proactive recommendations?
- Does it learn from patterns in your environment?
- How accurate are its predictions and anomaly detection?

3

Databricks-specific features

- Does it understand Databricks-specific concepts and components?
- How deeply does it integrate with Databricks?
- Does it support all Databricks deployment models (AWS, Azure, GCP)?
- Can it monitor both interactive and automated workloads?

4

Cost optimization features

- How granular is the cost attribution?
- Does it provide actionable recommendations with estimated savings?
- Can it detect idle resources and recommend or auto-fix right-sizing?
- Does it help identify who is accountable when issues are addressed or left unaddressed?

5

Ease of implementation

- What is the deployment time and complexity?
- Does it require agents or code changes?
- How much configuration is needed?
- How much manual effort is needed to monitor cost/performance after recommendations are applied continuously?

6

Scalability

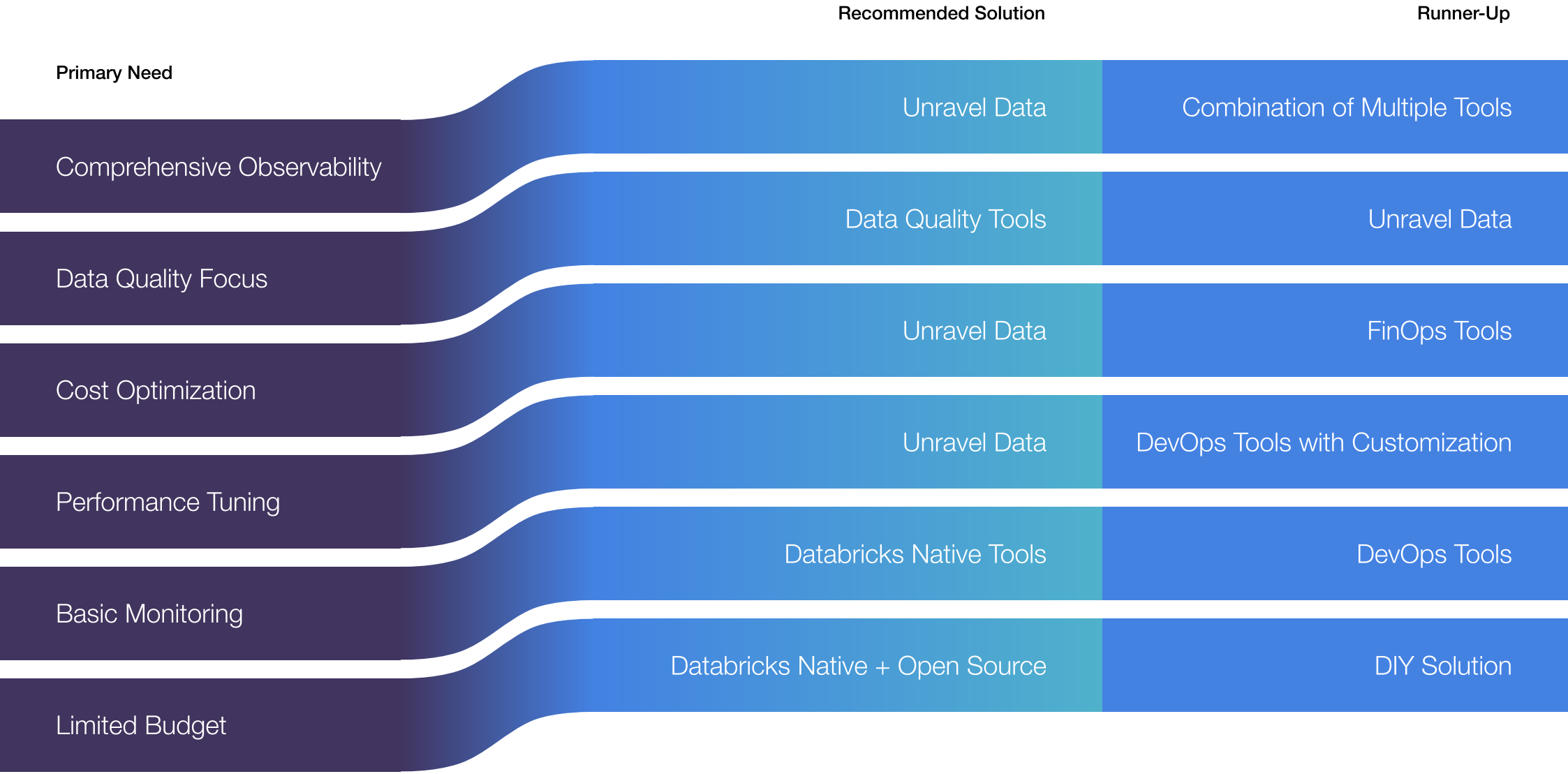
- Can it handle enterprise-scale Databricks deployments?
- How does performance scale with increased data volumes?
- Does pricing scale reasonably with usage?
- Can it support multi-region and multi-cloud deployments?

7

Total cost of ownership

- What is the licensing model?
- What internal resources are required for maintenance?
- How does the solution impact infrastructure costs?
- What is the expected ROI?

Decision Matrix Based on Primary Needs



Rolling out your solution

Rolling out a data observability solution for Databricks involves a phased approach—starting with assessment and testing, followed by full deployment and continuous optimization. With clear metrics and real-world validation, organizations can drive measurable ROI through improved performance, reduced costs, and faster issue resolution.

Implementation Roadmap

For organizations considering a data observability solution for Databricks, consider this phased approach.

Phase 1: 1-2 Weeks

Assessment and Planning

- Identify current pain points, observability, and actionability gaps
- Define key metrics and KPIs for success
- Document the current Databricks architecture and workload patterns
- Establish baseline metrics for performance and cost

Phase 2: 4-6 Weeks

Proof of Concept

- Select 2-3 candidate solutions based on evaluation criteria
- Implement each solution in a limited environment
- Alternatively, opt for auto-applying the solutions (if available)
- Test against real-world scenarios and pain points
- Evaluate effectiveness against defined metrics

Phase 3: 4-8 Weeks

Solution Selection and Implementation

- Select the preferred solution based on the PoC results
- Develop an implementation plan and success criteria
- Deploy the solution across dev, test, and production environments
- Configure alerts, dashboards, and integrations

Phase 4: Ongoing

Optimization and Expansion

- Refine monitoring parameters based on initial results
- Implement recommended optimizations
- Measure and document realized benefits against the cloud bill
- Expand coverage to additional workloads and environments

ROI Considerations

When calculating the potential return on investment for data observability solutions, consider these factors.

Cost Factors

- Solution licensing costs
- Implementation and professional services
- Internal resources for management
- Training and enablement

Benefit Factors

- Infrastructure cost reduction (typically 20-40%)
- Improved performance leading to faster time-to-insight
- Reduced downtime and mean time to resolution
- Engineering time savings from automated troubleshooting

Example ROI Calculation

For a mid-sized Databricks environment ($500K annual spend)

Average infrastructure cost savings	30% = $150K/year
Engineering productivity gains	15% of 5 FTEs = $150K/year
Downtime reduction	50% reduction in 10 annual incidents = $100K/year
Total annual benefit	$400K
Total solution cost (Example)	$150K/year
Net annual benefit	$250K
RETURN ON INVESTMENT	167%

Conclusion

As Databricks environments scale, comprehensive data observability is no longer optional. It's a competitive necessity. The right solution delivers measurable cost savings, faster insights, and greater reliability. While several solution categories offer partial capabilities, Comprehensive solutions like Unravel Data stand out, consistently delivering 20-40% cost reduction and 50% faster resolution of critical incidents. It offers comprehensive coverage across all five observation areas identified by Gartner, as well as deep integration with Databricks and AI-driven optimization capabilities.

Organizations should evaluate solutions based on their specific requirements, considering factors such as the depth of observability, ease of implementation, and potential return on investment (ROI). A phased approach with a proof of concept is recommended to validate the solution's effectiveness in your specific environment.

Take the next step: request an Unravel Data free health check or proof of concept and see how quickly you can unlock hidden value in your Databricks investment.

Additional Resources

- Gartner Market Guide for Data Observability Tools (2024)
- Unravel Data case studies and documentation
- Databricks' best practices for monitoring and optimization
- Industry benchmarks for Databricks performance and cost

Disclaimer

This buyer's guide is based on Gartner's 2024 Market Guide for Data Observability Tools, industry research, and best practices. Organizations should conduct their evaluation based on their specific requirements and environments.

unravel™