DataKitchen

DataKitchen DataOps Observability:

Technical Product Overview

DataKitchen's *DataOps Observability* provides visibility across your data operations so you can see how everything performs in real time, respond quickly to problems, and make improvements. The application monitors every **data journey** from data source to customer value, from any development environment to production, across every tool, team, environment, and customer.

DataOps Observability ingests events from your data journey tools and infrastructure. Events include pipeline or tool start, end time, current state, infrastructure status, log events, and data quality test results. It associates those events with a specific data journey instance, then uses a rules engine to compare them against expectations and determines if any events meet predefined conditions to trigger actions, such as alerts, new events, or tests. These details are displayed as data journeys in the application.

DataOps Observability provides data teams with critical features and benefits.

- Data journeys allow monitoring of every data process from source to customer value.
- Production expectations featuring data, tool, and infrastructure testing reduce embarrassing errors to zero, making root cause analysis possible.
- Real-time alerting and immediate responses to failures mean quick fixes before harmful code reaches production or bad data reaches your customer.
- Testing development tools and data increases the delivery rate and lowers the risk of deploying new analytic insights.
- An intuitive, role-based user interface allows all stakeholders, from developers to customers, to be on the same page.
- Off-the-shelf connections and an OpenAPI enable fast implementations without replacing your existing tools or changes to existing pipelines.
- Event storage and historical dashboards enable you to catch negative trends early and track improvements.

Understand Your DataOps Issues

If you work in a data-related field, you already know there are problems in your data processes and problems across the industry that still need to be fully resolved. The cost of continually implementing temporary fixes to these problems is far greater than implementing a DataOps Observability solution.

Poor data quality: A "hope and pray" culture is common. Data teams never know if their tooling or integrations will break and end up "firefighting" when things inevitably go wrong. It's embarrassing - and potentially costly - when customers find and report data issues the data team should have noticed.

Lack of visibility: With thousands of data journeys running and hundreds of tools supporting them, it is challenging to know if your entire data estate is working correctly at any given time. You can't answer basic questions about the status of your production processes. And your teams and stakeholders have yet to share a shared context about what's happening.

Work disruptions: Daily work is interrupted to find and fix errors, resulting in delayed features, innovations, and customer deliveries.

Team frustration: Data engineers feel pressure and leave their jobs for more productive environments or careers.

DataOps Observability gives you the tools to minimize or resolve these issues. It offers visibility across your operations so you can see, in real-time, how everything is going.

Essential Product Features

The *DataOps Observability* system ingests events from the tools and assets in your data estate. It correlates and filters this information to display and alert you to the status of your operations and the problem areas that require action. *Observability* can trigger pre-defined responses, such as sending events to a pipeline, running tests, or generating tickets in a system like ServiceNow.

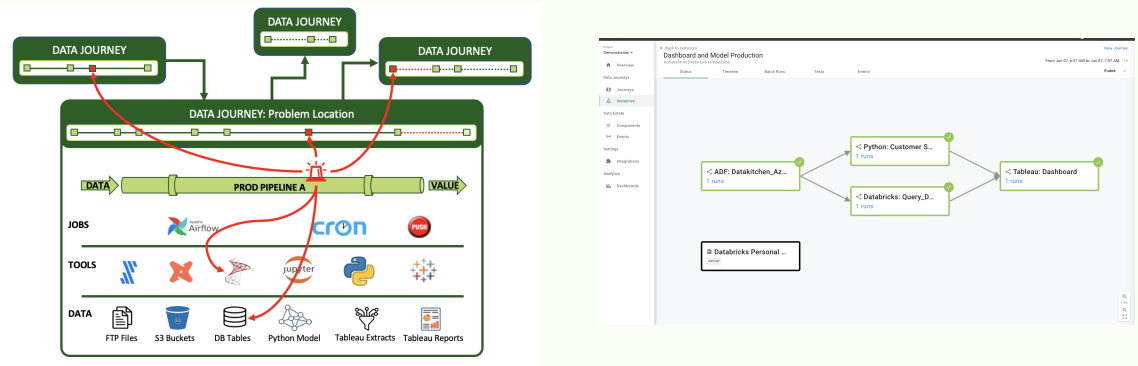
The following product features and capabilities in the *Observability* product can help resolve your DataOps issues.

Complete Data Journey Observability

You need visibility across the breadth and depth of your data estate. *Observability* can monitor every step in—and relationship among—multiple data journeys (representations of groups of

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

related pipelines), even crossing organizational boundaries. And it can track execution up and down any complex stack of jobs, tools, and data in each pipeline.



Monitor activity across your data estate and drill down to transaction details.

Data Journey Components Represent Your Data Estate/Toolchain

Components represent the resources, engines, and tools you use daily to deliver data analytic assets. Components can include batch pipeline runs executed by orchestrators, streaming pipelines in event-driven systems, datasets like database tables and files, and storage or computing infrastructure. Observability can integrate with any technology you use now or in the future.

Every event the system receives or creates is associated with a specific component.

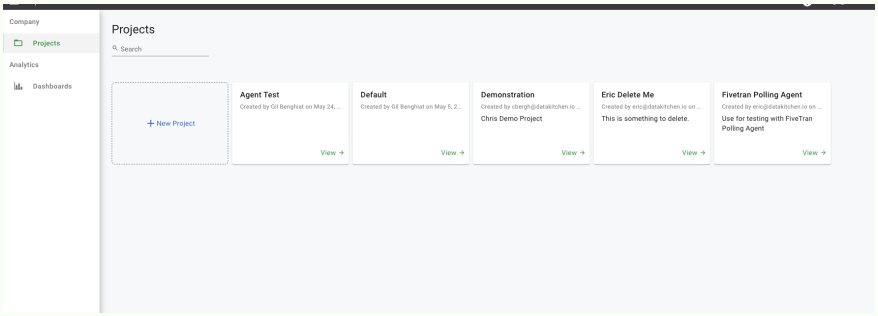
Component Type	Definition	Example
Batch pipeline	A batch, or finite, tool for data engineering.	An Apache Airflow DAG.
Streaming pipeline	A stream of event data	An event-based process enabled by, e.g., Apache Kafka.
Dataset	A specific collection of data.	An Amazon Redshift table or a folder in an Amazon S3 bucket.
Server	Storage or computing infrastructure.	An Amazon EC2 server is running Airflow jobs.

Data Journey Components

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

Data Journey Projects

Different teams and projects can manage many Data Journeys in an organization. DataOps Observability groups all Data Journeys into a Project for simplifying and sharing your Data Journey information.



Data Journeys Grouped into Projects

Capturing Data Journey Events

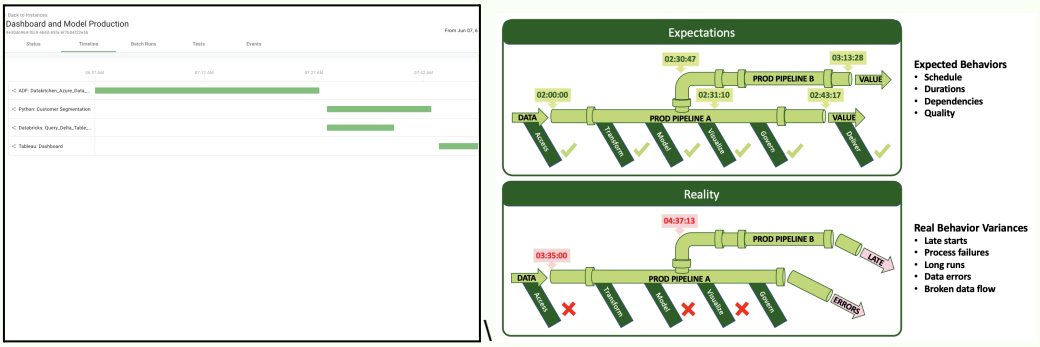
DataOps Observability collects event data through its API. Its events cover the significant activities of your data estate. Event data reflect the state, actions, and quality of your pipelines and data tools, not the data those tools act upon.

Event type	Description
Run Status & Schedule	Describes a change in status (i.e., running, completed, completed with warnings, failed) for a specified batch pipeline run.
Message Log	The Message Log logs a string message related to the pipeline and optionally related to a specific task. Logs capture failure, warning, or debugging messages from external tools and scripts.
Metric Log	The Metrics Log captures the value of a user-defined datum of interest, such as a row count, cost, or CPU percentage. This can be used for tracking the value of a metric through a run or for comparing the value of a metric across multiple runs.
Test Outcomes	The Test Outcome describes the results of a test or a set of tests executed on an external testing tool, such as DataKitchen's DataOps TestGen.
Dataset Operation	Reports a read or writes operation on a specified dataset component.

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

Data Journey Expectations and Alerts

You can define your production expectations to reduce embarrassing errors to zero. You can measure the variances during and after each run by establishing baselines—concrete expectations for run schedules and durations, data quality, and dependencies.



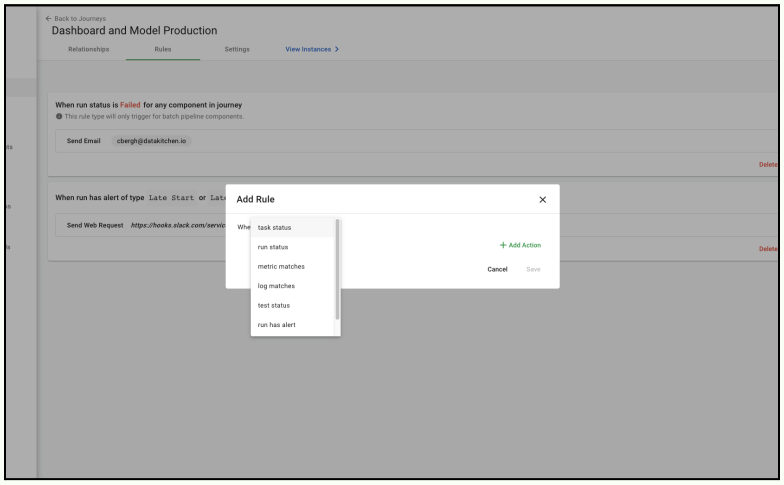
Set baseline expectations in Observability, then view the variances with each run.

Additionally, *DataOps Observability* accepts log scraping data from other tools and can consolidate real-time and logged events into critical alerts.

Event Engine to Apply Rules and Actions

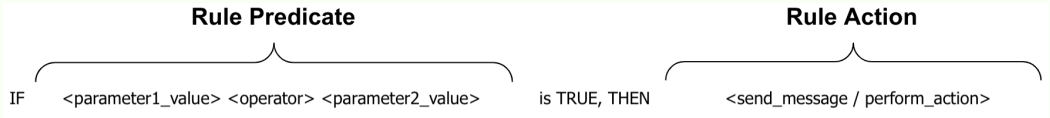
The vent information that *Observability* captures can invariably be a lot of data for someone to monitor and sift through. The DataKitchen product includes an event engine that can react to what's happening in the journeys and cut down the signal-to-noise ratio.

Given a set of rules, the event engine can trigger notifications through email, Slack, and other services, send alerts to your operations people, issue commands to start or end pipeline runs and prompt other actions when the difference between expected results and actual runs exceeds your tolerance thresholds.



Define rules for when a task or run does not meet expectations.

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved



Rule format overview

Rule Examples

- When test status is passed or any component in Journey, then send an email to <email address one>, <email address two>
- When the log matches level = any and the message contains ^This exact message only$, for <a specific batch pipeline>, send a webhook request to <URL>, then specify the payload and, optionally, the headers.
- When metric matches key = <key> and value = <value> for <a specific dataset>, then send email to <email address>
- When the run has an alert of type Late Start, Late End for any component* in the journey, send a webhook request to <URL>, then specify the payload and, optionally, the headers.
- For more examples, see the documentation.

Development Data & Tool Testing

You can catch and fix errors in production and during development by implementing development data and tool testing. The downstream effects are increased delivery rates and reduced risk of deploying new insights.

Monitor development pipelines and tools to catch errors before they reach production.

Observability enables you to watch the schedule and duration of your runs. Still, more importantly, you can check that the contents of your pipelines—the data, the models, the integrations, the reports, and the outputs your customers see—are as accurate, complete, and

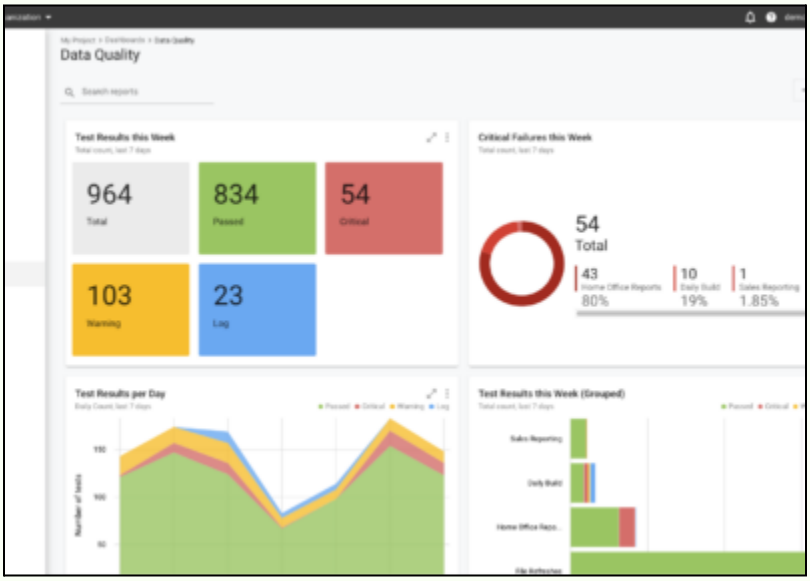
© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

up-to-date as expected. With *Observability,* you can implement automated tests the product consumes as events.

Tests that evaluate your data and its artifacts check data inputs, transformation results, model predictions, and report consistency. They range from typical software development tests (unit, functional, regression tests, etc.) to custom data tests (location balance, historical balance, data conformity, data consistency, business logic tests, and statistical process control). These tests can be developed and executed via any method, including frameworks like DataKitchen's TestGen or dbt.

Event Storage for Diagnosis and Historical Analysis

With *Observability*, you get real-time details and a store of run data over time. Use this information to diagnose problems and surface trends for statistical process control analysis.



Use historical data to diagnose errors and catch negative trends.

Success is not just about knowing what happens during the operation; it's also about what happens after the process. By storing run data, Observability allows your analysts to look at variances from a historical perspective and make adjustments. Finding trends and patterns can help your team optimize your data pipelines and predict and prevent future problems.

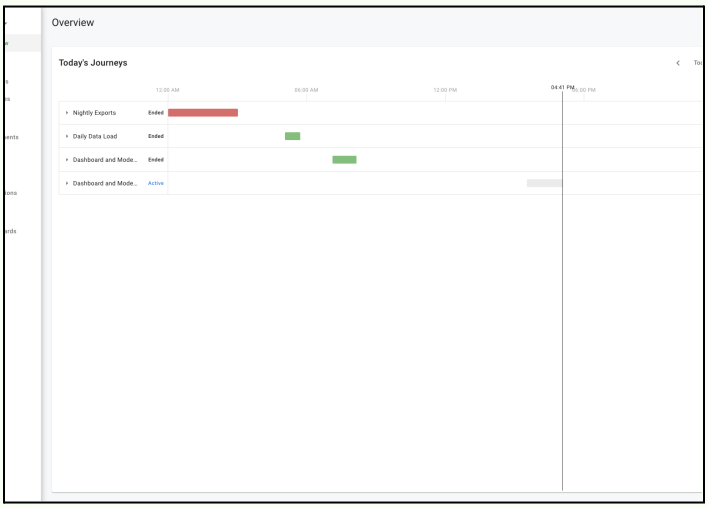
With this information, you can create ***historical dashboards*** to simplify root-cause diagnosis and enable your teams to locate the sources of issues quickly.

Role-Based User Interface for Everyone

Accessible User Interface for Everyone

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

Easy to understand and navigate, your DataOps Observability tenant is designed to be accessible to all. The user interface allows everyone on the team---IT, managers, data engineers, scientists, analysts, and your business customer---to be on the same page. And quite literally, too. Use the unique URLs the system generates to quickly share specific data, save significant instances, and bookmark events for future insight.



View your pipeline runs in real-time and address problems quickly.

Data Toolchain Integrations with Observability

Tool Integrations with OpenAPI

Pre-built integrations and an OpenAPI specification (OAS) enable quick integrations without replacing existing tools. You can start transmitting events to and receiving commands from *Observability* right away.

Integrating your infrastructure and tools with the Observability system lets you publish event information to the system's REST API. As Observability ingests those events, it translates, collates, and displays the critical event information you need to see.

DataKitchen recommends three methods for publishing events to Observability's Event Ingestion API: (1) use the DataKitchen Integration Agent agent to request events from your resources. DataKitchen provides a growing list of off-the-shelf agents for many of the most popular data tools available today. DataKitchen also provides a 15-business-day SLA for delivering **new (or updated) connectors** to commercial tools in your data estate/toolchain. (2) post events directly to the API; (3) publish via a client SDK generated by Swagger Codegen,

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

The Event Ingestion API can auto-generate client SDKs for interacting with the API. See Swagger Codegen for more details. Following this process, DataKitchen developed a client library in Python available for you to use for your integrations.

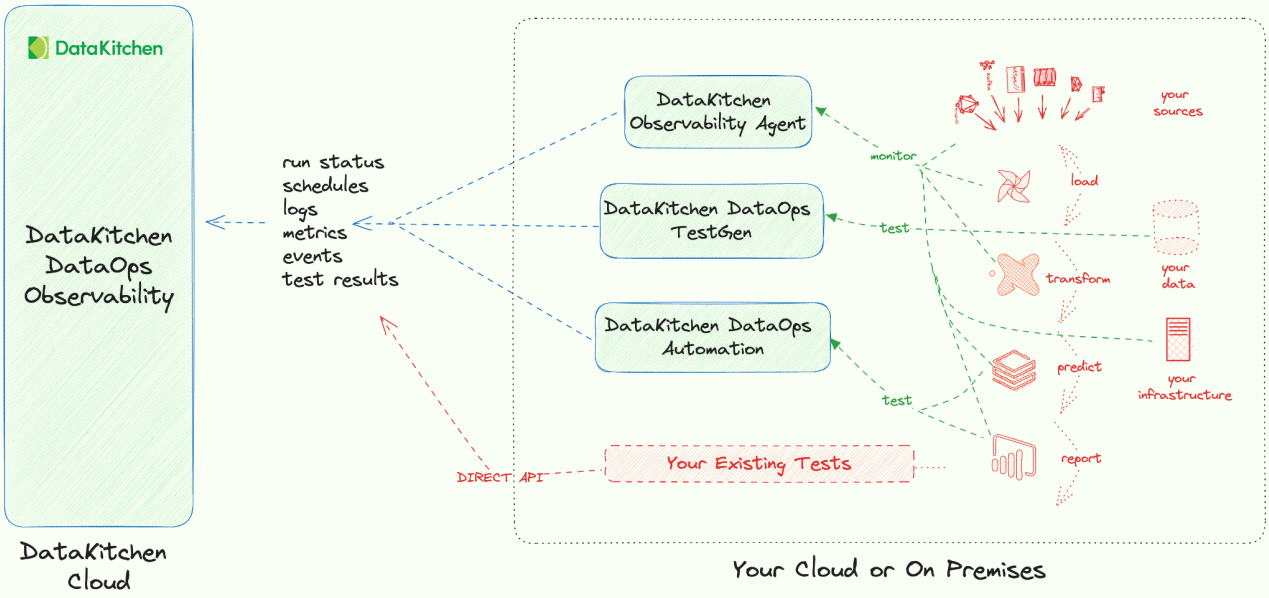
Publishing Method	Implementation Details	Best For
Observability Agent	Observed components expose or publish events in a way that an agent can forward to Observability. DataKitchen provides ready-made agents for many everyday data science and data tools.	A large number of resources are being observed where augmentation to send events is prohibitive
Direct API Requests	The observed resource publishes events directly to the Event Ingestion API	A small number of resources that are easy to augment to publish events
API requests via client SDK	The observed resource publishes events via a client library generated by Swagger Codegen in the user's programming language of choice	A small number of resources that are easy to augment to publish events

DataKitchen offers three methods to publish events; agents may be the most straightforward.

Integration with DataKitchen's DataOps TestGen and DataOps Automation Products

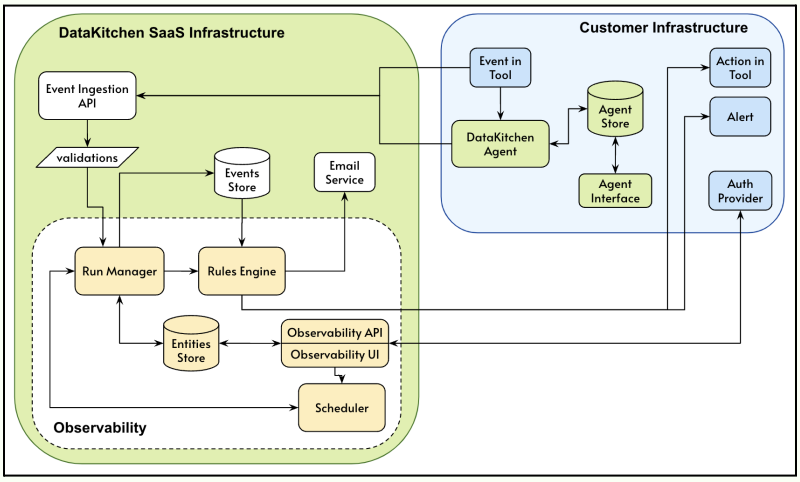
DataOps TestGen is a simple way to create and execute data quality validation tests against data in your databases. By profiling your data and automatically generating a standard set of validations, TestGen eliminates the need to write custom tests for many scenarios. When more custom tests are required, DataOps Automation provides a framework for building and executing complex tool, model, and API-level tests. Both products automatically forward test results to DataOps Observability.

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved



DataOps Observability Architecture

Initial releases of Observability are *limited to SaaS offerings only*. Future releases will include self-hosted implementations and additional capabilities.



- **Customer infrastructure:** Sends events from local tools to the Event Ingestion API directly or via an agent. May receive alerts and take actions based on pipeline task settings and run states.

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

- **Base infrastructure:** Collects events via the agent and sends them through the Event Ingestion API and a series of validations into an Events Store.

- **Observability application:** The pipeline run manager creates runs, manages tasks, sets run states as they occur, and sends run events to the Rules Engine for evaluation. The Observability API and Observability UI authenticate users and display pipeline run information.

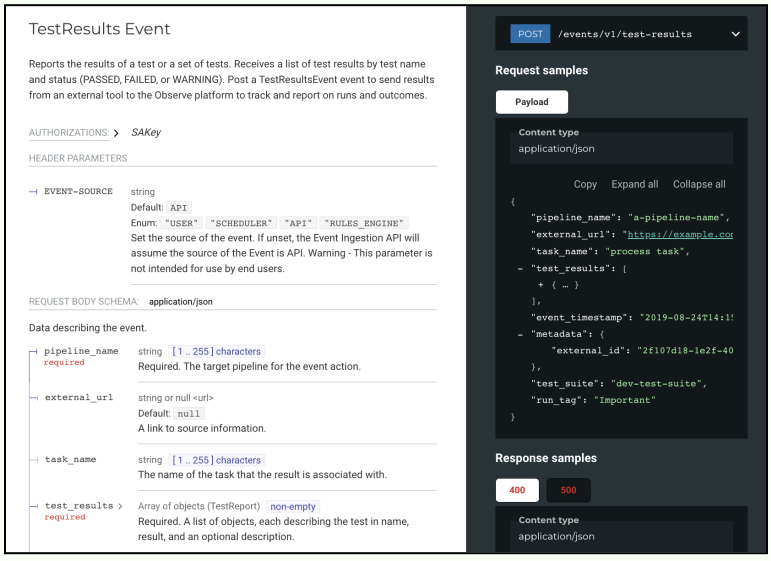
Secure Infrastructure

DataKitchen leverages the most sophisticated automation, scaling, cloud security, and authentication technologies for its DataOps Observability offering. The result is a secure and resilient service, giving you the confidence to monitor your most sensitive and complex data workloads with DataKitchen.

DataKitchen's cloud infrastructure includes features such as end-to-end encryption for data in transit and at rest, built-in redundancies, and high availability. It is subject to regular vulnerability assessments and penetration tests.

Event Ingestion API

The ***Event Ingestion API*** is a set of POST-only endpoints for receiving events from external tools, scripts, or DataKitchen's integration agent. It checks that each event is sent with a valid service account key and that the data schema of the event is valid for the specified event type.



Event Endpoints

- DatasetOperation Event
- MessageLog Event

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

- MetricLog Event
- RunStatus Event
- TestOutcomes Event

The API adheres to standard security practices by requiring authentication with service account (SA) keys. Find details in the Event Ingestion API documentation and more information in the *Observability Integration* help topics.

Observing the Observability Industry,

Many tools use the term 'Observability' today. How can you tell the difference between them? There are several key ideas to consider. First, IT hardware monitoring / Application Performance Monitoring (APM) checks, like disk and CPU, while helpful, are often lagging indicators of problems with your data journeys. Second, you need to check/validate the source data, the integrated data, and the things created from the data (like reports/models) to ensure data journey success. And third, you need the ability to test that the relevant tools, jobs, and code acting on the data are behaving as expected and that you know the 'run time lineage' of the whole system.

Category	Key Challenge	DataOps Observability Difference
Data Observability Tools: *Monte Carlo, AccelData, BIgeye, etc.*	Data Observability tools test data in the database/data store. DataOps Observability does that, plus correlates that to the other critical elements of the data journey – where a fundamental understanding is required.	DataOps Observability produces source-to-consumer data journeys; these tools do not. It can correlate events and logs from these vendors and others in one place/view.
I**T Observability Tools:** *Data Dog, New Relic, AWS, Azure, GCP monitoring services, etc.*	These tools are valuable but need to be completed. They monitor servers and networks, which is needed. However, problems in data and analytic systems span hardware, software, tools, raw data, integrated data, and the code that drives those tools.	DataOps Observability is built upon the concept of data journeys and tests, which is a feature lacking in those IT Tools. DataOps Observability can correlate events and logs from these vendors with any other critical signals in one place/view.
IT Logging Tools: *Splunk, Devo, Microsoft Sentinel, Azure Monitor, etc.*	These technologies are valuable but must be completed and provide a limited set of signals within a pipeline.	DataOps Observability produces and observes data journeys and tests, and these tools do not. DataKitchen can correlate events and logs from these vendors with

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

		any other critical signals in one place/view.
Build vs. Buy	It will take many months to develop and support the backend and user experience needed to support a scalable DataOps Observability solution.	DataOps Observability offers a consumption-based pricing model that allows users to adopt the platform at their own pace. The product can also integrate internally developed observability capabilities into a consolidated view.

DataOps Observability Technical Requirements

Contact your DataKitchen representative to set up the required infrastructure for your Observability account. This configuration includes your company account, initial user accounts, authentication service connection, unique user interface URL, default account entities, and Service Account Keys for API access.

Why DataKitchen's DataOps Observability?

DataKitchen's DataOps Observability is your mission control for every data journey in your enterprise. Data journeys span from data sources to customer value, go from any development environment into production, and provide monitoring across every tool, every team, every environment, and every customer.

DataKitchen's DataOps Observability detects, localizes, and provides a shared context so problems can be understood and acted upon immediately.

In a world of complexity, failure, and frustration, data and analytics teams need to deliver insight to their customers with no errors and a high rate of change. You don't have to live with these problems. DataKitchen DataOps Observability provides the solution.

The critical feature – support for data journeys – allows monitoring of every data process from source to customer value. Production expectations reduce embarrassing errors to zero, including data and tool testing. Development cycle data validation and tool testing increase the delivery rate and lower the risk of deploying new analytic insights. Historical dashboards enable

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved

you to find the root causes of issues and constantly improve. An intuitive, role-based user interface allows all stakeholders to be on the same page. Off-the-shelf integrations and an open API enable fast implementations without replacing your existing tools.

When you aim to produce rapid, trusted customer insight, you need to start by reducing your team's hassles and embarrassment while increasing the time it has to develop and deliver. You can lower your error rates and achieve your goals by monitoring all data journeys. Keep your tools and infrastructure the same; use the DataKitchen DataOps Observability product on top of those tools.

Spend less time worrying about what may go wrong and gain more time to create by observing every data journey and taking early action to stay on track.

Additional Resources

- DataOps Observability: Taming the Chaos (blog)
- Observability Functional Brief
- What is DataOps Observability? (FAQ)
- *DataKitchen DataOps Observability Help*
- Event Ingestion API documentation

DataKitchen Software

DataOps Observability:

- **End to End Data Journey Observability** across all tools, data, and infrastructure
- **Complete Toolchain Production Monitoring and Alerting:** Ensure speed & quality of delivery through key metrics and relevant alerts
- **'Mission Control' Dashboards** and historical analytics

DataOps TestGen:

- **Simple, Fast Data Quality Test Generation and Execution**
- **Data Profiling:** database scanning, profiling, and identification of 'bad data.'
- **53 Unique Data Test Types:** algorithmically generated and user-configurable business rule-based configurable run-time data tests and execution

DataOps Automation:

- **Automated Data and Tools Testing:** custom data test development in many languages and APIs
- **Test Development Environments**: Automate the creation & management of environments to speed new feature delivery; secure vault;
- **DataOps Automation**: Common collaboration system (Kitchens) across roles & teams: git integration; Meta-Orchestration: Easily design & execute a system of tools from a single view of the process

Revised June 8, 2023

© 2023 DataKitchen, Inc., datakitchen.io All rights reserved