**Modak Nabu**TM **2.5**

**RELEASE NOTES**

Contents

[1 Overview 3](#_Toc77362203)

[2 Glossary of terms used 3](#_Toc77362204)

[3 Major Additions 4](#_Toc77362205)

[3.1 Monitoring Dashboard for Pipelines 4](#_Toc77362206)

[3.2 Executive Dashboard 4](#_Toc77362207)

[3.3 Advanced options for ingestion 4](#_Toc77362208)

[3.4 Tags in Data Connections and Data Pipelines 5](#_Toc77362209)

[3.5 Email Notification for Pipelines 5](#_Toc77362210)

[3.6 Files Ingestion 5](#_Toc77362211)

[3.7 Modak AlmarenTM artifact based ingestion workflow 6](#_Toc77362212)

[3.8 New Data Connections 6](#_Toc77362213)

[3.9 Enhanced landing pages for datastore, data connection and data pipelines 6](#_Toc77362214)

[3.10 Manage Compute Engines 7](#_Toc77362215)

[3.11 Enhancements to scheduling cron jobs 7](#_Toc77362216)

# Overview

Nabu converges data discovery, data ingestion, data preparation, data catalogue and data profiling into a single enterprise platform with metadata being primary driver.

Metadata is captured and kept up to date via a suite of crawlers that gather and store technical metadata for each identified data asset. This metadata is utilized by BotWorks – a framework comprising asynchronous, independent computing units chained together in dynamic flow patterns, to orchestrate various tasks in Nabu, such as data ingestion and data profiling. The data can be explored through Nabu search interface. Similar data is identified through data fingerprinting technology.

Nabu 2.5 adds more self-service capabilities for regular data engineering activities and provides improved visibility on the progress/status of such activities.

# Glossary of terms used

|  |  |
| --- | --- |
| Data Connection | Any existing repository of data in an enterprise. This can be relational database, file systems or cloud services. In earlier versions of Nabu, these were named ‘Data Places’ |
| Datastore | Logical grouping of data connections and selected schemas/tables/views. |
| Kosh | Metadata repository of Nabu. |
| Crawl | Process of capturing metadata from tables/files in a data connection and storing them in Kosh. |
| Ingest | Process of moving data from a source data connection to a target data connection. |
| Profile | Profiling captures statistics including range of values in a column, frequency distribution, minimum and maximum values, cardinality, and number of null values. |
| Index | Indexing of metadata and data using Apache Solr makes it searchable through Nabu search interface |
| Entity | An ‘Entity’ is a collection of similar columns in a datastore. Similar columns are identified using data fingerprinting. Data within these columns is searchable using Nabu Search. |
| Facet | A collection of similar columns which are grouped together using data fingerprinting. Difference between an entity and a facet is that the data in the columns grouped together under a facet is not searchable. |
| Filter | Filter is the parameter on which search results of an entity can be filtered. |
| Synonym | Alias for an entity. Search results of a synonym are the same as that of the entity for which it is an alias |
| Fieldstore | Fieldstore is an attribute of the entity which can be displayed in search results for an entity. |

# Major additions

## Monitoring dashboard for pipelines

Large scale data movement in enterprises requires setting up pipelines from different sources to a target destination such as a data lake or a data warehouse. Some of the pipelines may run for long and data engineering/DataOps teams need to monitor them regularly and take appropriate action as required.

The monitoring dashboard provides unified interface to monitor the pipelines scheduled in Nabu and helps in troubleshooting. The dashboard shows details about a pipeline such as its status, time taken for a run, status of previous runs, source(s) and destination for a pipeline.

It also shows details at each object level (table/file) – their status, size, time taken for ingestion. In case of failed tables/files, users can view details of the stage at which the ingestion workflow failed. The link to view Spark logs for ingestion jobs is also provided on the dashboard for quick diagnosis.

Those tables/files in a pipeline which failed to get ingested can be ‘retried’ directly from the monitoring dashboard, rather than scheduling a pipeline again.

## Executive dashboard

The executive dashboard is a single pane of glass for business stakeholders to view the status/progress related to various activities for data preparation.

The dashboard provides information on data crawling (such as number of data connections, tables, files, collections, size of data crawled), pipelines (number of tables, files, collections, and size of data moved through pipelines) and data profiling (number of tables, files, size of data profiled)

Recent and upcoming jobs for data crawling, ingestion, profiling, and indexing can be viewed on the executive dashboard. It also allows sharing custom views of dashboard with different stakeholders.

The details on the executive dashboard can be contextualized by filtering information based on tags. These tags (in the form of key-value pair) are user defined and are applied on data connections, pipelines (see section 3.4 for details) and datastores. Users can also view aggregate results based on a tag category.

## Advanced options for ingestion

Advanced options for ingestion provide ability to change defaults for running pipelines. The advanced options for pipelines include following categories:

1. **Basic Configuration:** This allows changing default values for pipeline flow timeout, number of retry attempts for failed objects, number of parallel source connections.
2. **Schema Drift:** Users can choose to be notified when schema drift is detected. Additionally, the following actions can be taken when schema drift is detected:
3. Drop existing table at destination and create another table with new schema
4. Create table at destination with the new schema definition and create backup of existing table
5. Keep existing table at destination as is, and create another table with new schema
6. Ignore tables from pipeline where schema drift is detected.
7. **Manage Inconsistent Data Types:** The mappings for those data types that are not consistent at the source and at the destination is shown. The users have the option to change default mappings.
8. **Ignore column data types:** Certain column data types (such as BLOB/CLOB), which may not be needed for analytical purposes, can be ignored from the pipelines for quicker execution.
9. **Pipeline Pre-Conditions:** Pre-conditions for execution of a pipeline can be defined in terms of success/failure/completion of other pipelines.
10. **Manage unsupported data types:** If mappings for any data type are not available, then the user can choose between different options to handle them. The unsupported data types can be ignored from the pipeline, or can be ingested as null, or as custom text, or as-is.

## Tags in data connections and data pipelines

To add context to data engineering activities, Nabu 2.5 provides ability to add tags to data connections and pipelines. This is in addition to the existing capability to add tags to a datastore, table, column, and entity.

The tags are defined as key-value pair (comprising tag category and tag value) that provide business or operational context. These tags can then be used as filters in monitoring dashboard and executive dashboard.

## Email notification for pipelines

Users can select to be notified on the status of pipelines through email. The email ids that need to be notified in case of pipeline success, or failure can be provided while creating/modifying the pipeline. The email notifications contain pipeline status and summary of successful/failed objects.

## Files ingestion

Pipelines for ingesting semi structured files such as CSV, TSV into data warehouses like Hive, Redshift and Azure Synapse can be created in Nabu 2.5. The pipelines are created in a similar manner as the pipelines for tables, with the additional option to ingest only new or modified files. Users can select the files to be ingested from a data connection by manually selecting file names or applying regex or like filters on file names.

## Modak AlmarenTM artifact based ingestion workflow

With enterprises using diverse sources of data and adding newer sources frequently, it is essential that these sources can be quickly on-boarded into Nabu, to be used as source or destination for data movement.

With this objective the ingestion workflow in Nabu has been templatized. The workflow uses Modak AlmarenTM artifacts for carrying out ingestion. Modak AlmarenTM is a low code framework for writing Spark jobs.

Adding an ingestion workflow for a new source-destination combination now involves creating the artifacts and deploying them with the templatized workflow. These artifacts provide the connection details for source and destination and data type mappings.

## New data connections

Support for few more data sources has been added, out of the box. Nabu 2.5 provides metadata crawlers for DB2, Teradata, Netezza, Salesforce, HTTP, in addition to the existing metadata crawlers for SAS, MongoDB, Oracle, SQL Server, Hive, PostgreSQL, MySQL, and cloud services from AWS, GCP and Azure. The user interface for adding data connections has been redesigned to make it quicker and easier to add data connections.

## Enhanced landing pages for data connections, data pipelines and datastores

The landing pages for data connections, pipelines and datastores have been enhanced to provide greater functionality.

On the data connection landing page, users can view when the metadata was last crawled, status of last crawling job. Users can schedule crawling from the landing page itself and can also create ‘duplicate’ data connections to quickly edit information while creating new data connections.

On the ingestion landing page, users can view status of last run of the pipeline, date of last run, next scheduled run. Users can schedule pipeline run from the landing page itself and can also create ‘duplicate’ pipelines to quickly edit information when creating new pipelines.

On the datastore landing page, users can view status of last profiling done, date of last profiling and next scheduled profiling. Users can schedule profiling, indexing and datastore refresh from landing page itself and can also create ‘duplicate’ datastores to quickly edit information when creating new datastores.

## Manage Compute Engines

Nabu provides ability to select different compute engines for ingestion jobs. In Nabu 2.5, users can select compute engines for profiling and indexing jobs as well.

The user interface for creating, modifying, and deleting compute engines is provided in Nabu 2.5.

## Enhancements to scheduling cron jobs

The user interface for scheduling cron jobs has been simplified. The options to schedule jobs at regular intervals (daily, weekly, monthly) have been made more intuitive.

The interface provides added functionality to schedule a job to run only once, in addition to the existing functionality to schedule recurring jobs. Moreover, the users can select their own time zones while scheduling the jobs and do not have to schedule jobs in server time zones. The new interface retains the ability to schedule jobs by providing cron expression directly.