



# Using Snowflake as Your Time Series Database

When migrating your operational data to the cloud, you can choose from multiple time-series databases as your cloud persistence layer. Snowflake has several inbuilt capabilities that are a prerequisite to a good time series database. It offers infinite scalability, cost-effective storage, and fast query response times both for point reads and analytical queries. Using the same database both for operational and EDW workflows will vastly simplify cross-functional analytics that require integrated data sets. Can Snowflake fulfill your time-series data needs too? In this whitepaper, we deep dive into this topic and argue that the answer to the above question is a strong yes if you leverage DeepIQ as the data engineering layer.

## Background

Snowflake has several important features that make it a good candidate for time-series data.

1. Snowflake is a hybrid columnar type of database and the values of a tag in a time-series table can be sequentially stored one after the other rather than row by row. This is a good design pattern for time series data because it matches the standard consumption pattern and minimizes the “extra” data that Snowflake must load to provide the results of a query.
2. Snowflake micro-partitions data automatically according to its ingestion order by default. As time series data is typically ingested in time sequence, Snowflake will partition the data accordingly. This again matches typical consumption patterns for time-series data (reading the values of a tag or tag over a given time) and allows Snowflake to efficiently “prune” that underlying storage files storing the time series data. Consequently, a table containing a decade’s worth of time series data can be queried just as quickly as one containing just a week’s worth.
3. The separation of storage from compute, and the “immutable” data design choices that are a part of Snowflake make things such as simultaneous ingestion of high-rate time series data and analysis of that data a non-issue. Users can stand up a small compute instance to support continuous data ingestion of thousands of rows per second and at the same time against that same growing data set leverage a large multi-processor set of compute instances to process that data for machine learning use cases, etc.

Despite these architectural advantages, there are important issues that need to be addressed for Snowflake to replace your existing time-series persistence layer. These issues are described below:

## 1. Ingestion

Firstly, due to strict network security restrictions on control networks, operational data sources, including SCADA, Historian, or Control systems, may not be directly accessible from your cloud tenant. Even if the operation network is accessible, standard data extraction software does not support the protocols or framework required for connectivity to the sources.

## 2. Real-Time Analytics

Secondly, the volume and velocity of time series data are an order of magnitude higher than the standard IT sources. Many times, these raw data sources may have to be enhanced with provenance, data cleansing, or machine learning model results before persistence. To support these use cases, you will need data ingestion software that scales to high data volumes and velocities and can enrich data during the ingestion process.

## 3. Data Engineering

Additionally, time series data has unique transformation requirements. For example, you will have to impute to remove missing data, interpolate data to the required frequency and remove high-frequency noise. To perform these analytics at scale, you will need to develop complex user-defined functions.

DeeplQ provides a straightforward answer to all these challenges. In this whitepaper, we use a sample use case to illustrate how DeeplQ and

Snowflake provide an ideal solution for a real-world use case.

## DeeplQ Introduction

DeeplQ is a self-service {Data + AI} Ops app built for the industrial world. DeeplQ simplifies industrial analytics by automating the following three tasks:

- Ingesting operational and geospatial data at scale into your cloud platform.
- Implementing sophisticated time series and geospatial data engineering workflows; and
- Building state of the art ML models using these datasets.

With DeeplQ's DataStudio, moving your IoT data into Snowflake from your diverse industrial technology and network landscape is a simple task with no requirement for code development. This whitepaper will explain how you can build a production-grade data pipeline with real-time streaming data from your industrial systems into your Snowflake in minutes. Using a real-world use case, we will explain how to build these data pipelines and conclude with a brief discussion about the data apps you can build with this technology.

## Use Case

In this use case example, the data source is an on-premise OPC server that is connected to a control system. Our objective is to move this streaming data into Snowflake. However, the raw data frequency is irregular, making it difficult to process. Our first goal is to persist this raw data in a Snowflake data warehouse.

Our ultimate objective is to generate a table with regularly spaced 1-hertz data that can be easily consumed by downstream analytic processes. With a consistent frequency, the data can be easily visualized and used to derive meaningful conclusions.

## Architecture

For this use case, we will use the architecture shown in Figure 1. We have a DeeplQ edge connector that uses an OPC UA client to connect to the on-premises streaming data source. DeeplQ edge uses an encrypted and compressed data channel to push messages to the AWS cloud and land data into the storage layer of your choice. In this use case, we will persist incoming high-volume time series data into an AWS Kinesis stream and use DeeplQ's DataStudio software to format the data and persist it into Snowflake. We then use another DataStudio workflow to read the raw data from Snowflake and create a cleaned time series table with 1-HTZ frequency data. DeeplQ's DataStudio workflow leverages Apache Spark's distributed and parallel computing framework to scale to high volumes.

*Note: We can leverage other services like Azure EventHub, Google PubSub, or Kafka instead of AWS Kinesis.*



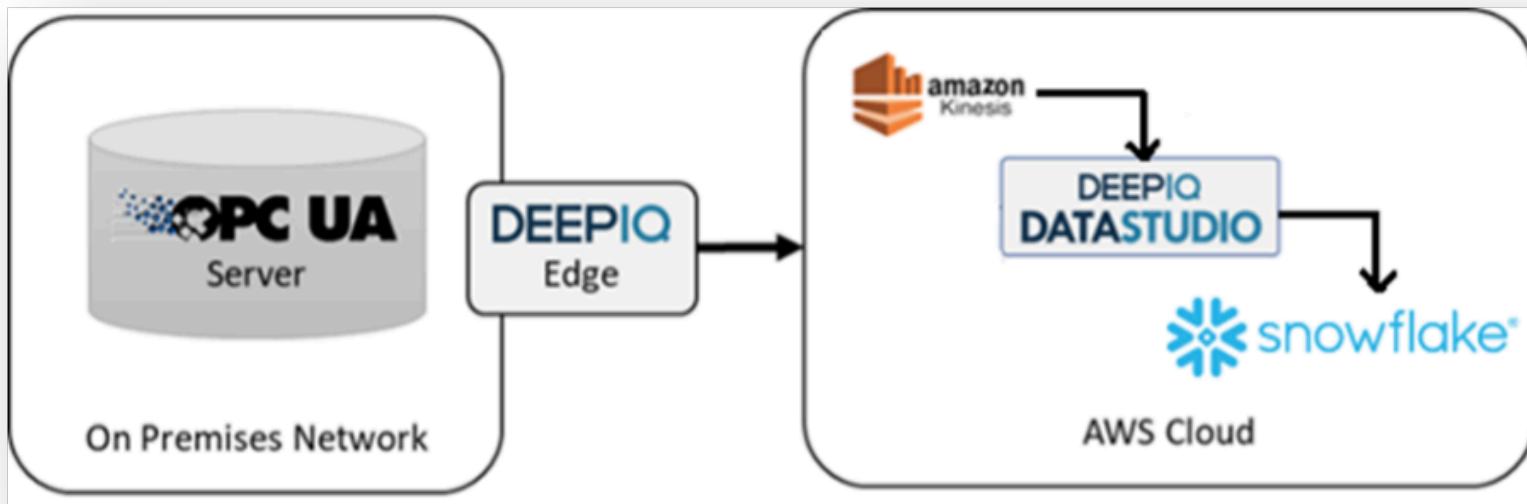


Figure 1: Dataflow from DeepIQ Edge to Snowflake on AWS using DeepIQ DataStudio

DeepIQ's edge connector allows users to parallelize data read requests and maximize throughput to the extent supported by the source system. In this whitepaper, we will focus on the throughput of the cloud with different strategies.

## Approach and Results

We use DeepIQ's DataStudio in the AWS environment for this sample use case. We will also use an installed DeepIQ Edge connector on our on-premises network where the time series data source is available. DeepIQ edge installation is also GUI based and hassle-free.

Once deployed, we are ready to build the production pipeline from the historian to the AWS cloud. This task is a three-step process.

- We configure DeepIQ Edge to publish data to an AWS Kinesis Data Stream/Topic.
- We use a DeepIQ DataStudio workflow to read data from Kinesis and write it to Snowflake.
- We use another DeepIQ DataStudio workflow using proprietary components to read and process data. These include time series data transformation and cleansing techniques such as interpolation.

We will first create a request for data using DataStudio's edge control panel. We specify the tags of interest in our data request, as shown in Figure 2.

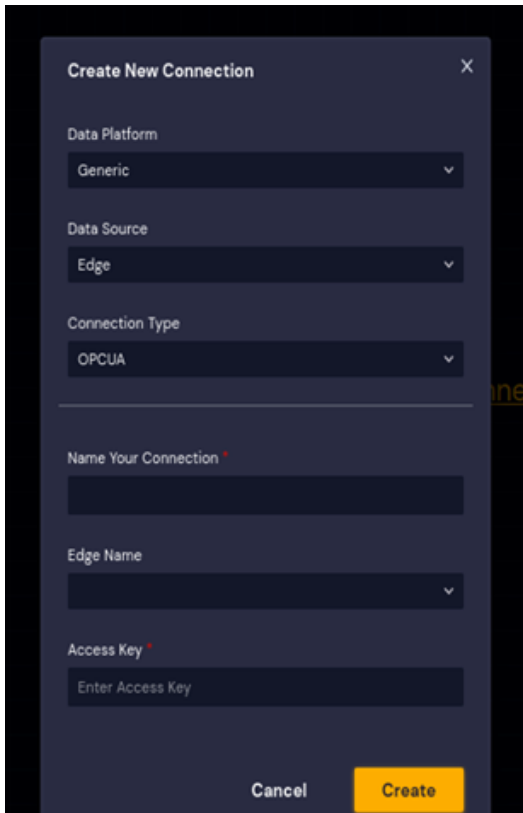
The image shows a 'Create New Connection' dialog box with a dark theme. It features several dropdown menus: 'Data Platform' set to 'Generic', 'Data Source' set to 'Edge', and 'Connection Type' set to 'OPCUA'. Below these is a text input field for 'Name Your Connection' with a red asterisk indicating it is required. This is followed by an 'Edge Name' dropdown menu and an 'Access Key' text input field, also marked with a red asterisk and containing the placeholder text 'Enter Access Key'. At the bottom, there are two buttons: a 'Cancel' button and a yellow 'Create' button.

Figure 2: Creating Connection to an OPCUA Edge Connector

Once the data request is submitted, the DeepIQ Edge software processes the request by submitting it to the OPC UA server, as shown in Figure 3. The OPC UA server will publish the data into the configured Kinesis topic.



### Create Request

edge-01 (OPC UA)

Request Name

request-opcua

Request Body \*

Copy

Clear

1

2

3

4

5

6

7

```
{  
  "command": "get_historical_data",  
  "element_root": "Pump Data Replicate",  
  "element_name": "Level 1",  
  "tags": "**",  
  "max_values": 1000  
}
```

Cancel

Create

Figure 3: Requesting and Filtering for required tags from the Edge Connector

You can monitor the process and find all the available tags in DataStudio, as shown in Figure 4.

ID	Request Name	Request Created	User	From Upstream	Status	Actions
0001	tag_1	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0002	tag_2	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0003	tag_3	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0004	tag_4	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0005	tag_5	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0006	tag_6	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0007	tag_7	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0008	tag_8	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0009	tag_9	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop
0010	tag_10	May 24, 2022 12:45 PM	admin	From Upstream	Running	Stop

Figure 4: Showing tags available in the connected Edge

DeepIQ's DataStudio supports data ingestion into Snowflake in both Spark Batch and Streaming modes.

Figure 5 is a sample workflow showing the Snowflake data ingestion in the streaming mode. We are reading data from AWS Kinesis, transforming the data to the required schema, and persisting data into a Snowflake Table.

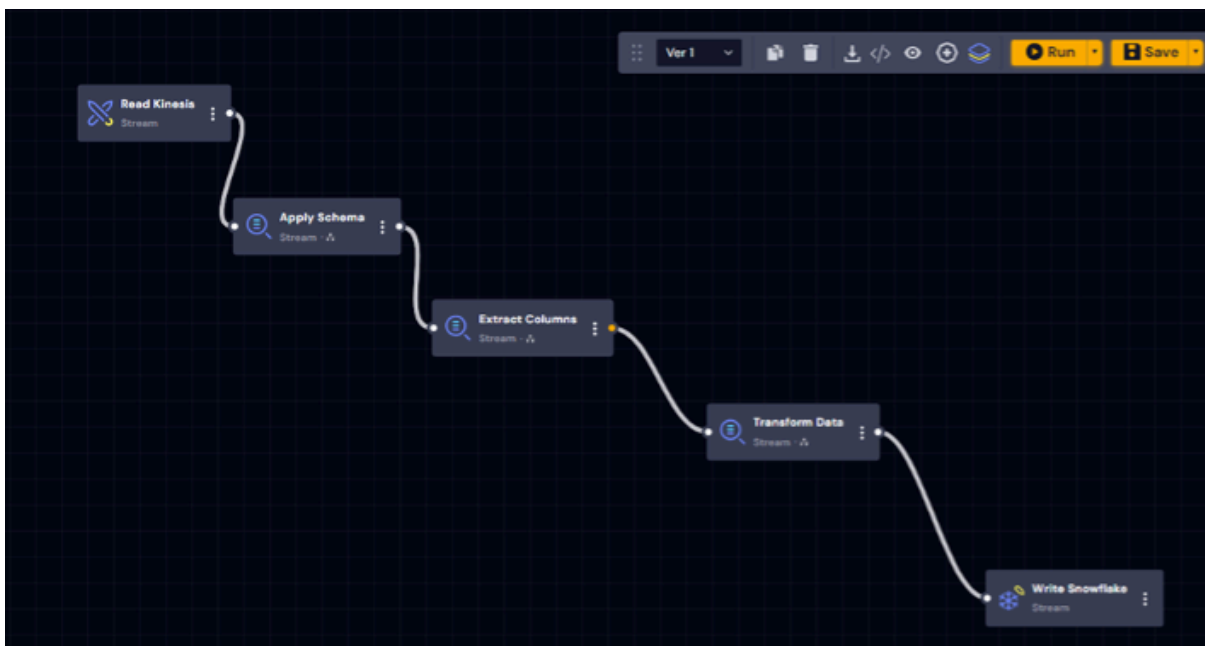


Figure 5: A simple workflow reading from an AWS Kinesis and writing to Snowflake

, let us review the performance and scalability aspects of DeepIQ DataStudio components.

Figure 6 is a DataStudio workflow that writes to a Snowflake table in high-performance batch mode. We limited the Spark cluster size to twenty cores and 70 GB to test our workflow performance. Using this workflow, we were able to ingest the 10,000 records in under one second. DataStudio uses the underlying spark cluster to submit records to Snowflake by parallelizing the load to each executor, which helped us scale out and achieve the throughput of 10,000 tags per second. We can scale the ingestion process further by increasing the partitions and executors (cluster size) for larger datasets.



Figure 6: DataStudio Workflow writing to Snowflake in high-performance Batch mode

SAMPLE / PUBLIC / SNOWFLAKE\_USERS1\_PIDATA

Table ACCOUNTADMIN 8 minutes ago 100 20.0KB

Table Details Columns Data Preview Copy History

SAMPLEDB 100 Rows • Updated just now

		ATTRIBUTE	TIMESTAMP	VALUE	QUALITY
1	/Test/toplevel00/secondlevel00	Input Rate05	2022-11-03 18:24:21	307.4189	0
2	/Test/toplevel00/secondlevel00	Inlet Pressure09	2022-11-03 18:24:18	93.73811	0
3	/Test/toplevel00/secondlevel00	Outlet Pressure02	2022-11-03 18:24:21	133.4879	0
4	/Test/toplevel00/secondlevel00	Outlet Pressure03	2022-11-03 18:24:21	121.4037	0
5	/Test/toplevel00/secondlevel00	Discharge Pressure08	2022-11-03 18:24:18	41.63112	0
6	/Test/toplevel00/secondlevel00	R/Vib01	2022-11-03 18:24:21	232.3368	0
7	/Test/toplevel00/secondlevel00	Discharge Pressure05	2022-11-03 18:24:21	42.96402	0
8	/Test/toplevel00/secondlevel00	Temperature00	2022-11-03 18:24:12	188.2611	0
9	/Test/toplevel00/secondlevel00	Volume03	2022-11-03 18:24:21	277.1048	0
10	/Test/toplevel00/secondlevel00	Volume08	2022-11-03 18:24:15	278.4912	0
11	/Test/toplevel00/secondlevel00	Output Rate01	2022-11-03 18:24:21	346.1765	0
12	/Test/toplevel00/secondlevel00	Output Rate06	2022-11-03 18:24:21	358.8356	0
13	/Test/toplevel00/secondlevel00	Performance09	2022-11-03 18:24:18	384.1872	0
14	/Test/toplevel00/secondlevel00	Pressure00	2022-11-03 18:24:18	31.39744	0

Figure 7: A screenshot from Snowflake showing the data



In this use case, the data is captured with a timestamp as the index for different tags/attributes with their value and quality. We can also expand to capture additional attributes like asset names, location, etc

Now, let us look at the time series data processing needs. DeepIQ's DataStudio offers components for smoothing, approximation, and interpolating time series data with various algorithms. Figure 8 shows a workflow that reads data from the raw storage table and creates a regular-spaced data table after removing outliers. In this workflow, we use cubic interpolation to interpolate data and an STD outlier removal algorithm to remove outliers.



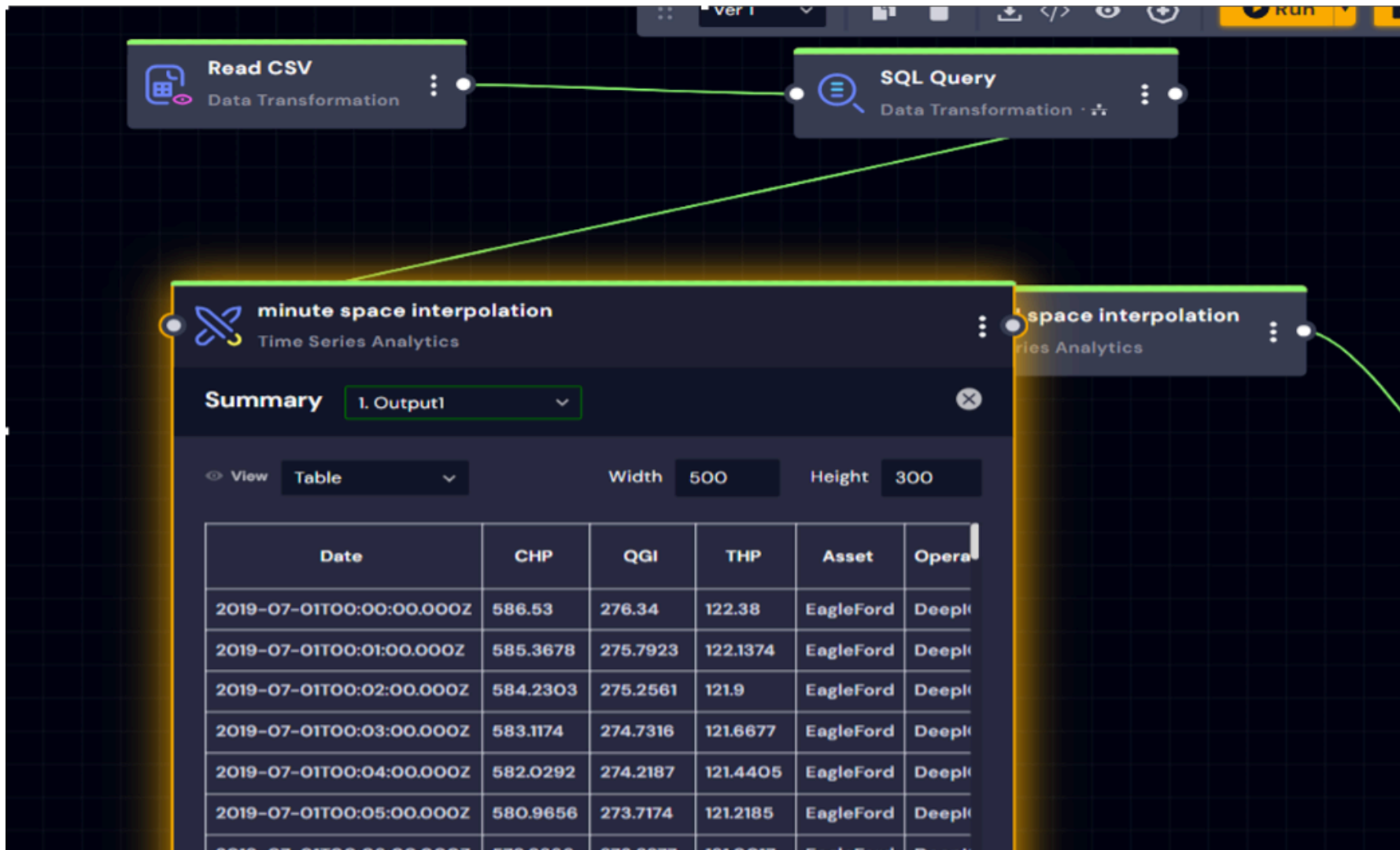


Figure 8: Workflow showing data cleansing and transformation activities

## Other Use Cases

Now that you have a data pipeline that consistently moves clean time series data into your Snowflake data warehouse, you have the foundation to build powerful data applications. In our previous whitepaper, which is available at [DeepIQ | Predictive Maintenance](#), we discussed how to implement predictive health models even with sparse failure data.

In addition to streaming data into Snowflake, DeepIQ offers the capability to connect directly to AWS Kinesis using Spark Structured Streaming. This enables you to execute machine learning models and calculate KPIs in real-time, providing you with the ability to make informed decisions quickly.

Conclusion

In this whitepaper, we explained how DeepIQ simplifies the process of persisting and analyzing time series data in Snowflake. As illustrated in the use case, Snowflake is a powerful cloud service platform that could be used as a time series database and as a single platform for both IT and OT data. With DeepIQ's DataStudio, you can realize the full potential of your Snowflake Data Warehouse by ingesting both your operational and IT data at scale and developing innovative solutions that maximize the value of this data. For more information, please get in touch with [info@deepiq.com](mailto:info@deepiq.com).

DeepIQ is on a mission to transform industrial processes by digitizing industrial expertise. Our vision is to drive end-to-end automation, enabling systems such as self-running power plants or drilling rigs using generative AI as the higher order reasoning layer operating over existing industrial automation technology stack.

[Privacy Policy](#)   [Terms of Services](#)



FEATURES

Extract  
Engineer  
Explore

INDUSTRIES

Upstream  
Midstream  
Downstream & Chemical  
Mining

SOLUTIONS

IT-OT Contextualization  
Well Construction  
Optimization  
P & ID Digitization

PARTNERS

AWS  
Azure  
Cloudera  
Google



Predictive Maintenance  
Production Optimization  
Route Optimization  
Sustainability

Databricks  
OSIsoft(AVEVA)  
Snowflake

## COMPANY

About Us  
Resources  
News  
DataStudio Deployment  
Guide  
Career  
Contact Us

