YugabyteDB Voyager

Product Brief

Accelerate Cloud Native Adoption with End-to-End Migration

At A Glance

YugabyteDB Voyager simplifies the migration of data from the most common legacy and single-cloud relational databases to YugabyteDB.

Database migration has traditionally been complex, leading to delayed modernization and cloud native initiatives. YugabyteDB Voyager is purpose-built for distributed databases, enabling enterprises to embrace the full potential of distributed SQL. It is a unified product that accelerates cloud native adoption, simplifies operations, and reduces risks.

Key Capabilities

- Unified end-to-end migration product
- Supports legacy and single-cloud databases
- 100% free and open source
- Simple, one-click installer
- Distributed-SQL aware
- Supports partition tables on all source databases
- Schema analysis and migration
- Direct import from CSV files
- Resume data import on failures
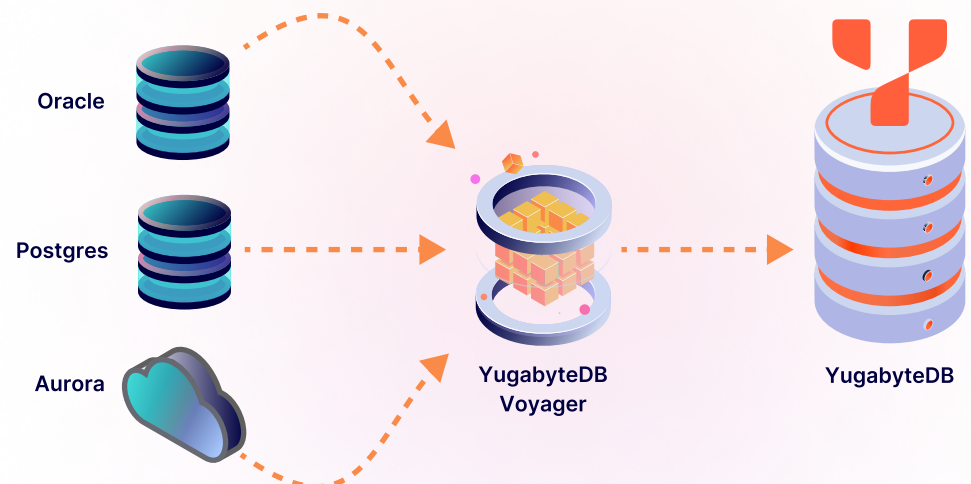- Public, private and hybrid cloud support

Introduction

YugabyteDB Voyager accelerates cloud native adoption by enabling the safe and easy migration of application data from your legacy databases and single-cloud databases to YugabyteDB, the cloud native distributed SQL database for business-critical applications.

Database migration has traditionally been complex. It requires multiple steps, each bringing risk and opportunity for error, while depending on esoteric or proprietary features. This complexity has delayed many data modernization efforts, resulting in suboptimal productivity, increased risk, and higher operational costs as the business has to continue connecting modern apps and microservices to legacy databases.

YugabyteDB Voyager eliminates this major pain-point for organizations by freeing critical application data from the dependencies and limitations of classic monolithic databases and popular single-cloud databases.

YugabyteDB Voyager is a unified, open source product that capitalizes on the strengths of distributed databases. It manages the end-to-end database migration journey, including cluster preparation, schema migration, and data migration. The powerful data migration engine easily and safely migrates data to a distributed data fabric that can be deployed on public, private, or hybrid clouds. It supports moving data to YugabyteDB Managed, YugabyteDB Anywhere, and the core open source database, YugabyteDB.

The Benefits Of YugabyteDB Voyager

Organizations today want to accelerate their cloud native and application modernization initiatives. However, database migration is often the biggest barrier, slowing or in many cases preventing projects from moving forward.

YugabyteDB Voyager encompasses the entire lifecycle of the database migration cycle in a single unified product from assessment to migration—and even includes optimization when migrating to DBaaS with the YugabyteDB Managed Performance Advisor.

This unified solution helps organizations simplify data migration.

Accelerate Cloud Native Adoption

Remove the common barriers to data modernization and accelerate the evolution to a cloud native architecture with a well-defined migration process. Voyager guides users through a step-by-step process to migrate not just a single database at a time, but hundreds of different databases if required.

Simplify Migration Operations

Eliminate complex planning and manual processes with a unified, open source product that understands the unique strengths of distributed databases. It assesses migration readiness, analyzes schema, recommends changes, and performs a safe data migration with horizontal scalability, automatic sharding and parallel data ingestion.

Reduce Risks

Discard legacy tools that are not distributed-database aware and eliminate manual, error-prone scripts with this purpose-built solution. Voyager provides a proven, repeatable process to analyze and execute the migration process across over ten of the most popular legacy and single-cloud relational databases.

End-To-End Migration Engine

YugabyteDB Voyager is designed to provide complete database migration lifecycle management to simplify and automate migration from legacy and single-cloud databases to YugabyteDB. The migration product is based on consistent, well-defined steps, each providing specific status information and guidelines on the next step in the migration process.

There are six key steps to this proven migration journey—all powered by YugabyteDB Voyager:

1. **Prepare:** Install YugabyteDB Voyager with the one-click installer that now supports air-gapped and docker-based installations. Verify that the user has sufficient privileges and connectivity to both source database(s) and YugabyteDB.
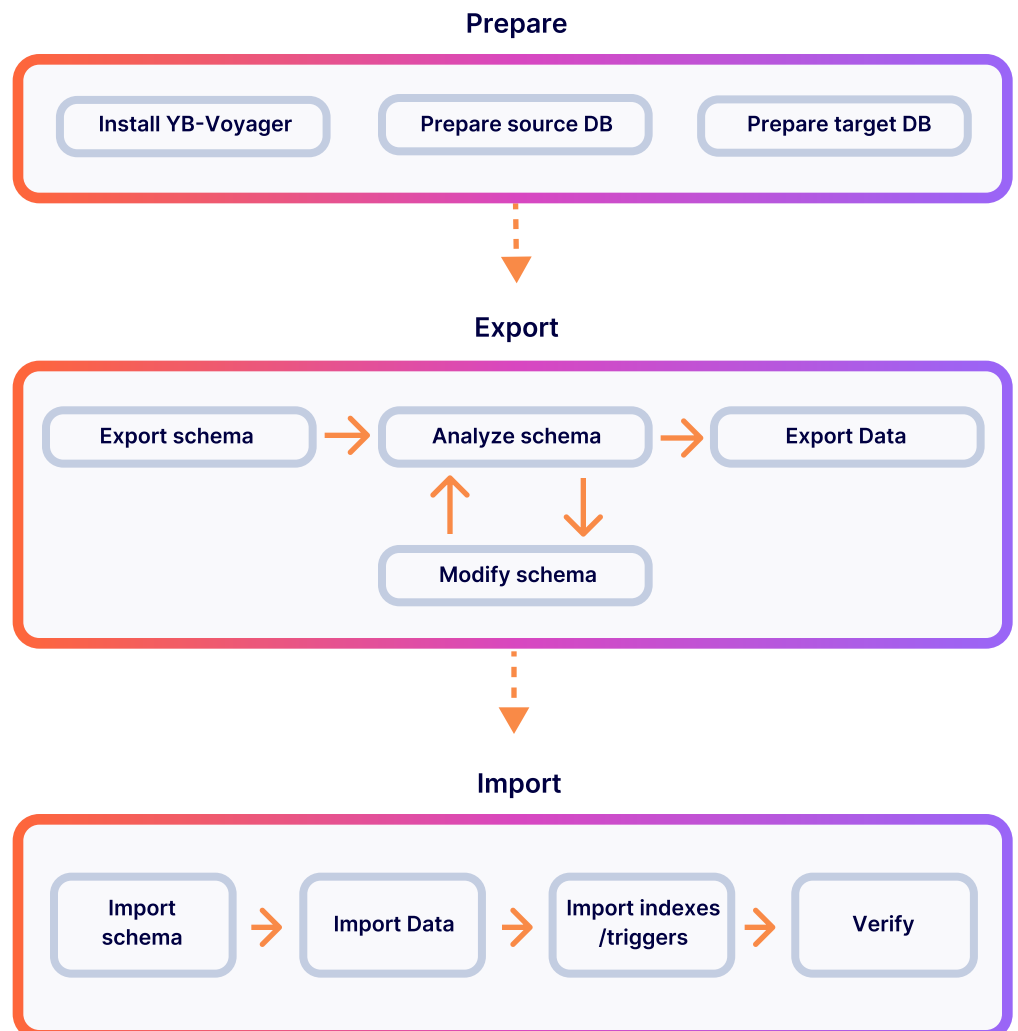
"To simplify the migration from our legacy databases, we were excited to embrace YugabyteDB Voyager. We completed a successful PoC that allowed us to load 120M records with YugabyteDB Voyager - an end-to-end migration product. We look forward to moving our production workloads with YugabyteDB Voyager and accelerating our cloud-native adoption."

- Madhan Kumar,
Founder of Genxlead

2. **Analyze Schema:** Export the schema and then generate a Schema Analysis Report to review suggested changes. Make the necessary schema adjustments to properly support a distributed database environment. Rerun analysis as needed on updated schema.

3. **Export Data:** Export data into local files on the machine that has YugabyteDB Voyager installed.

4. **Import Schema:** Import the ready schema to the YugabyteDB database. Note: safe defaults are 'on' by default, so that indexes are created only after data loads, foreign key checks are disabled, and triggers are created but disabled.

5. **Import Data, Indexes and Triggers:** Import the data into the target YugabyteDB database, and once complete, import the indexes and triggers. Then at last verify (currently this is a manual step by comparing source and target databases).

6. **Verify and Optimize:** Once the migration is complete, verify that the source and target databases are correct. For migrations to YugabyteDB Managed, leverage the included Performance Advisor to intelligently optimize indexes, queries, schema, and more for enhanced performance.

Prepare

| Install YB-Voyager | Prepare source DB | Prepare target DB |

Export

Export schema → Analyze schema → Export Data

↑ ↓

Modify schema

Import

Import schema → Import Data → Import indexes /triggers → Verify

Database Modernization Service

Adopting game-changing enterprise technology with confidence can be challenging without a trusted team of experts to help. For organizations lacking the in-house skills or time, the Yugabyte Customer Success (YCS) team is here to help. Their mission is to guide you to success, from the early stages of consideringYugabyteDB adoption, through migration and modernization, and beyond.

The YCS team is composed of industry experts and veterans including Customer Engineers Architects, Migration and Modernization Engineers, and Customer Success Managers.

Key Features Of YugabyteDB Voyager

YugabyteDB Voyager Features		
Supported Source Databases	PostgreSQL • PostgreSQL 9.x - 11.x • Amazon Aurora PostgreSQL • Amazon RDS for PostgreSQL • Cloud SQL for PostgreSQL • Azure Database for PostgreSQL Oracle • Oracle 11g - 19c • Amazon RDS for Oracle	MySQL • MySQL 8.x • MariaDB • Amazon Aurora MySQL • Amazon RDS for MySQL • Cloud SQL for MySQL
Supported Target Databases	• YugabyteDB • YugabyteDB Anywhere • YugabyteDB Managed Supported versions include YugabyteDB stable version 2.14.5.0 and later and preview version 2.17.0.0 and later.	
Key Features	• Free and 100% open source • Zero to minimal changes to the source databases required in most cases • Unified CLI (command line interface) experience for all different source databases • Offline migration with source database remaining operational during most steps to minimize downtime • Auto-tuneable based on workloads by analyzing the target cluster capacity and running parallel jobs by default • Progress bars identify expected time for data export and import to complete • Built-in resiliency can resume data import in case of failures • Parallelism of data across tables • Direct data import from CSV files	

yugabyteDB