White Paper

# Open Source Migration

## 4 Tips For Migrating From Proprietary to Open Source Solutions

## Overview

As organizations today develop their application stacks to meet customer demand, they are facing an explosion of data they have to manage—and a growing need for powerful data layer technologies to operate as a part of their stack. IDC predicts that more data will be created in the next three years than in the previous 30 years combined, placing pressure on engineers, architects, and IT leaders to ensure they have the most efficient architectures in place to handle their growing data layer needs.

As organizations decide which data layer technologies to embrace, they generally have to reach an important decision: whether to adopt proprietary or open source software. To make things even more complicated, there are also "open core" solutions which can appear to be open source but are actually proprietary solutions drawing from open source code. The decision organizations make at this juncture has far-reaching consequences for long-term architecture, data management, staffing, and budget.

In Instaclustr's work with hundreds of customers, and with more than 100 million node hours of hands-on operational experience, we've seen a growing trend of organizations leaving proprietary technologies for open source.

**IDC predicts that more data will be created in the next three years than in the previous 30 years combined**

[1] https://www.idc.com/getdoc.jsp?containerId=prUS46286020

instaclustr

Organizations make this choice for reasons such as reducing costs, maximizing flexibility and reducing lock-in, and benefiting from the vibrant and innovative developer communities around open source.

True open source solutions have gained popularity with developers across the board. In a 2020 survey by StackOverflow of over 65,000 developers, Redis, PostgreSQL, and Elasticsearch were the three “most-loved” databases. Meanwhile, the proprietary databases IBM DB2 and Oracle were the two “most-dreaded.”

If you’re currently using a proprietary data layer solution and are thinking of moving to true open source, this white paper is for you. We’ve helped more than 100 customers migrate from proprietary technologies to open source solutions. Ultimately, migrating to open source brings far-reaching benefits for your data layer and broader application stack in the long term. Read on to explore four valuable strategies for ensuring a successful move to open source.

# 1. Identify Alternatives to Proprietary Features

If you’re currently using a proprietary data solution, you may be concerned what life will be like without certain features. The good news is there are often open source alternatives to proprietary features you’re accustomed to using. Disentangling your architecture from these features can be achieved by finding suitable open source or custom-built alternatives with the same functionality (and where you won’t be charged license fees).

The following are some examples of proprietary features from proprietary vendors and their open source alternatives:

- **Confluent Replicator:**

  Many organizations relying on Confluent solutions use Confluent Replicator to replicate topics across clusters. The open source alternative to Confluent Replicator is Apache MirrorMaker 2, which similarly lets you mirror clusters and supports various cluster topologies. MirrorMaker 2 also includes a number of enhancements over MirrorMaker 1, which was a less complete solution.

- **DSE Search:**

  Many companies working with Datastax use Datastax Enterprise (DSE) Search to index, search, and analyze their data. There are numerous open source alternatives to DSE Search, including SASI indexes, Apache Lucene Index, or OpenSearch (the open source replacement for Elasticsearch).

[2] https://insights.stackoverflow.com/survey/2020#overview

- **Oracle Spatial:**

  This feature of Oracle allows developers and analysts to store and analyze geospatial information. When moving to PostgreSQL, Oracle Spatial can be replaced with PostGIS, an optional PostgreSQL extension that acts as a spatial database and can be queried with SQL.

These are just some of a wide number of open source technologies that can take the place of proprietary features that some vendors license and charge for. For more niche use cases where an open source solution is not available, it's often not too difficult to build (either in-house or with a consultant) the capability you need. This process is still cost-effective over the long term as it lets you avoid paying license fees.

**One of the top 10 U.S. banks saw a 5% performance improvement after migrating from Datastax Enterprise to open source Apache Cassandra without any changes in their data model or architecture.**

## 2. Document Your Migration Plan

After identifying the open source solutions that will replace otherwise proprietary features you rely on, the next steps are finalizing and implementing a migration plan that minimizes or eliminates application downtime. There are a number of effective approaches to migration you can adopt depending on the vendor you are leaving and the open source solutions you are adopting.

While examining the migration process from a specific proprietary solution to an open source alternative would require a more in-depth document, the following are steps you can expect to take in any such migration:

- **Gap Analysis and Health Check:**

  Assess any gaps in your software needs and address them with open source or custom-built alternatives (as discussed in the previous section).  Review the health of your cluster and assess any gaps between your current configurations and processes and the best practices for your new solution. This should involve a review of your infrastructure, monitoring, data model, and usage, among other areas.

- **Determine Your Migration Pattern:**

  There are a number of different approaches you can take for executing a migration—from changing over one node at a time in a cluster, to creating a new, mirrored cluster and removing the old. Some applications or technology changes might require more in-depth application-specific strategies. Decide on the migration pattern that is the best fit for your use case; advice from someone with experience in similar migrations can be invaluable.

- **Pre-Migration Setup:**

  Remove any dependencies you have on proprietary-only features and replace them with your planned alternatives.

- **Complete the Migration:**

  Execute your chosen migration pattern and de-provision any clusters or other infrastructure that are no longer needed.

- **System Cleanup:**

  Perform a review of the health and performance of your cluster, and remove any legacy proprietary software.

## AdStage Migrates From Datastax to Apache Cassandra

AdStage provides an all-in-one online advertising platform to manage Google, Bing, Facebook, LinkedIn, and Twitter ad campaigns. AdStage needed a solution that allowed the platform to scale rapidly. Integrating a new customer to the platform can mean literally adding terabytes of data within a short time span.

During the early stages of AdStage's development the company participated in the DataStax Startup Program. As the company went through a considerable growth phase, AdStage was nearing the point where the company would no longer qualify for the DataStax Startup Program, and as such was facing the prospect of having to pay considerable licensing fees as the production deployment was now over 80 nodes.

The Instaclustr Technical Operations team worked with the AdStage engineering team to transition every node in both production and staging environments to the open source version of Apache Cassandra with zero downtime. Apache Cassandra delivered the operational speed that AdStage needed to ensure that the platform had low latency and the required throughput.

***"Instaclustr has been instrumental in helping us manage, scale, and optimize our Cassandra deployment, allowing us to focus on delivering a great product to our customers,"*** said Jason Wu, the former CTO and CoFounder of AdStage.

## 3. Harness the Full Benefits of Open Source

The positive impacts from migrating to open source can be far-reaching, and it's important once you've completed your migration to ensure you've maximized the value from the changes you've made. Some of the many benefits of moving to open source include:

- **Cost Savings:**
  Proprietary vendors will use specific features (even those with free open source alternatives) to force you to pay license fees that escalate with each contract renewal. Moving to open source can generate considerable cost savings due to no longer paying those fees. Even when you bring in outside support (as discussed in the next section), you can still expect to generate meaningful savings over using proprietary software. Following your migration, revisit your data infrastructure budget line by line to pull out any remaining proprietary features that you may be overpaying for and switch to open source alternatives.

- **Community Innovation:**
  The open source community is a vibrant, innovative, and collaborative space, where experts from many different companies and disciplines come together to build solutions for the benefit of all. Ultimately, this community is able to pull from a more diverse set of experts than a single proprietary vendor can. To harness this innovation, track the latest solutions coming out of this community, especially by following releases associated with the Apache Foundation or the Cloud Native Computing Foundation.

- **More Flexible Architecture:**
  Once you've migrated to open source, you get the benefit of no longer having to consider vendor licensing when planning your application architecture. Once you've removed the proprietary, licensed software from your stack, revisit your architecture plan for the coming years. Are there projects you wanted to undertake but didn't because it would have cost more for the additional proprietary features? Are there open source alternatives you can now bake into your architecture instead?

To make sure you stay with open source as you add or substitute in different solutions, look for software licensed under the Apache License 2.0, the MIT License, the 3-Clause BSD License, or similar. Periodically assess your architecture to make sure you have the most cost-effective, flexible, performant solutions for your data layer.

## 4. Get Help If You Need It

If you're unsure about moving to open source with your current resources and staffing, there are a variety of ways to get outside support to ensure a successful migration. You can:

- **Train Your Team:**
  There are experts who can train your team to confidently handle the migration process and the day-to-day management of your chosen open source data layer technology.

- **Hire a Consultant:**
  Hire an outside consultant to perform the migration for you.

- **Sign Up for a Support Plan:**
  Think of these types of plans as your "phone a friend" for open source. If you're undertaking a migration but expect you'll have a few questions along the way, a support plan could be a good option for you.

- **Outsource to a Managed Service:**
  You can hire a managed service provider to handle the daily management of your chosen open source solution. This frees you up from managing your data layer so you can focus engineering resources on the unique features of your application. Managed services providers (Instaclustr among them) often also provide consulting and support for migrations.

- **Engage With the Community:**
  A wonderful thing about the open source community is the level of collaboration and knowledge-sharing that happens between people from a wide variety of technical backgrounds. If you have a question, there's a good chance there's a forum thread that answers it.

## Conclusion: Don't Wait to Move to Open Source

If you're currently using a proprietary data layer solution, you may hesitate to make the jump to open source. Fears about the difficulty of migration, your team's ability to support the new technology, or losing out on proprietary features could be holding you back unnecessarily. With appropriate planning, the move to open source is more than achievable, and rewards you with lower-cost, performant solutions that are part of a rich ecosystem that will evolve along with your application stack.

# About Instaclustr

Instaclustr delivers reliability at scale through our integrated data platform of open source technologies such as **Apache Cassandra®**, **Apache Kafka®**, **Apache Spark™**, **Elasticsearch™**, **Redis™**, and **PostgreSQL®**.

Our expertise stems from delivering more than 100 million node hours under management, allowing us to run the world's most powerful data technologies effortlessly.

We provide a range of managed, consulting, and support services to help our customers develop and deploy solutions around open source technologies. Our integrated data platform, built on open source technologies, powers mission critical, highly available applications for our customers and help them achieve scalability, reliability, and performance for their applications.

## Build, run, and scale your app with confidence

Like what you see?

If you're looking at building a proof of concept for your application, or looking for production grade nodes, contact a member of our **Sales team** to discuss your specific needs.