



# A Strategic Guide for lakeFS OSS Users: Evaluating the Move to Enterprise

At a Certain Scale, You Pay — Whether You Realize It or Not

Open source is where you start. It gives you superior flexibility, low friction, maximum speed and control. But eventually, scale catches up with you.

When your team outgrows basic Git workflows, you move to GitHub or GitLab Enterprise. When Spark becomes business-critical, you adopt Databricks. You move to Confluent when Kafka pipelines power production systems.

Data version control is no different.

If you're running lakeFS OSS and your org is scaling — multiple teams, growing data volumes, production pipelines — you've already made the decision to pay: in engineering hours, in slow recoveries, in risk. You're building workarounds for problems that enterprise features solve natively.

Your organization is already invested - it is just what currency you use, and yes manual labor and complexity are their own currency.

**lakeFS OSS provides a powerful foundation and can take you a long way on your journey. But at some point, Enterprise is the right answer.**

This checklist will help you assess whether it's time to consider the enterprise edition of lakeFS.

☒ Click each box that describes your use case.

## Data Governance & Compliance

You require SSO, RBAC, or IAM integration to align with organizational security policies

Audit logs are needed for compliance or incident investigation

You require short-lived access tokens or IP whitelisting for security

You have external SCIM or IdP tools and want identity managed centrally

You're under pressure to enforce governance across data platforms, but OSS lacks the controls

Multiple teams or users are working in parallel on shared datasets

## Integration with the Broader Stack

You need orchestration-layer integration (e.g., Airflow, Dagster) for isolated pipeline runs

You want to use lakeFS with Iceberg and need native integration

Your notebooks or local compute require easy mounting or direct Git-style operations

You want to query versioned metadata at scale for lineage, auditing, or model reproducibility

You are managing data across multiple storage locations: multi-cloud, hybrid, or on-prem environments

You are using lakeFS with Red Hat OpenShift AI and need native integration, identity controls, and scalable versioned storage across workloads

## Scalability, Stability, and Operations

You have hit scale challenges with large commits, large object counts, or long GC cycles

You need PrivateLink or region-specific architecture for cloud networking/security

You require high availability, SLA guarantees, or external disaster recovery capabilities

You want managed GC and data retention logic aligned with business policies

## Strategic Investment

lakeFS is now part of a production ML, analytics, or ETL pipeline

You've onboarded multiple departments or domains using data versioning

Your internal platform team spends engineering cycles building what Enterprise already provides

You need faster support resolution and roadmap visibility

Score one point for each box you clicked to assess your readiness tier.

| Total Score | Readiness Tier            | Interpretation                                                              |
|-------------|---------------------------|-----------------------------------------------------------------------------|
| 0–4         | Not Yet Ready → Early     | OSS adoption; may revisit in 6–12 months                                    |
| 5–9         | Emerging Needs            | OSS is straining; custom tooling or workarounds                             |
| 10–14       | Enterprise Pressure Point | Actively feeling pain; likely building features                             |
| 15–20       | High Readiness            | Already operating like an enterprise customer; should be in Enterprise eval |