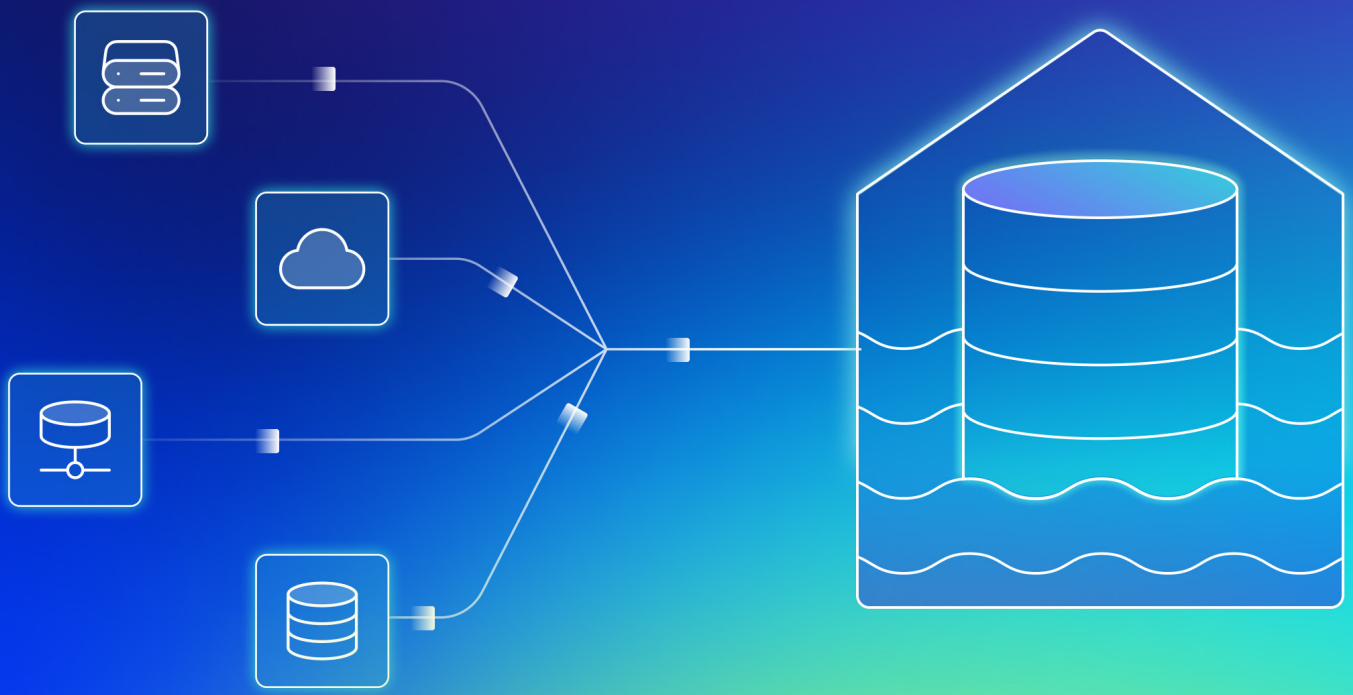
FOR DREMIO SOFTWARE

# Dremio's Well-Architected Framework

## A Guide for Best Practices for Optimized Performance & Reliability



dremio

# Dremio's Well-Architected Framework

## A Guide for Best Practices for Optimized Performance & Reliability

Dremio's Well-Architected Framework helps design and operate solutions with Dremio Software, providing insights from customer experiences. The framework is composed of pillars that provide design principles and concrete best practices that are based on those principles.

This document is a summary version of Dremio's Well-Architected Framework, designed to provide a concise overview of these essential principles and best practices. It encapsulates key insights from the lessons learned over time, having helped hundreds of customers achieve success with Dremio.

The framework follows five common cloud provider pillars (AWS, Microsoft, Google) and includes a sixth pillar specific to Dremio:

1. **Security**
2. **Performance Efficiency**
3. **Cost Optimization**
4. **Reliability**
5. **Operational Excellence**
6. **Self-service Semantic Layer**

For a more comprehensive exploration of each pillar, along with in-depth examples and detailed guidance, the full Well-Architected Framework document can be found online in Dremio Documentation. This summary serves as a practical reference to quickly grasp the core concepts and start implementing best practices in your Dremio deployments.

PILLAR 1/6

# Security

Dremio's Security pillar is essential to ensuring that your data is secured properly when using Dremio to query your data lakehouse. The security components are especially important to architect and design your data platform. After your workloads are in production, you should review your security components on a regular basis to ensure compliance and eliminate threats.

### CORE PRINCIPLES

- Leverage Industry-standard Identity Providers and Authorization Systems: Integrate with leading providers (e.g., Azure AD, OpenID, Okta) using multi-factor authentication and SSO.
- Design for Least Privilege Access to Objects: Ensure self-service access is granted only to necessary data.

### BEST PRACTICES

- Protect Access Credentials: Use identity providers to avoid sharing passwords. Secure LDAP integration with CA-signed certificates.
- Leverage Role-Based Access Controls: Manage access with roles instead of individual user privileges to protect data integrity and simplify architecture.

PILLAR 2/6

# Performance Efficiency

Dremio is a powerful platform that can process large amounts of data. To get the best performance out of your Dremio environment, you should follow these design principles and implementation best practices.

## Dimensions of Performance Optimization

When optimizing Dremio clusters for performance, several factors should be considered. Queries submitted to Dremio must be planned by the coordinator before being routed for execution. There is always one master coordinator and, optionally, additional scale-out coordinators that assist with planning JDBC/ODBC queries. The coordinator generates a query plan that can be used to route the query to one of the engines that are part of the cluster.

Some queries can be well-written, and some can consume inordinately high resources from the start. Those queries can be rewritten and optimized on their own without regard to the larger cluster.

Beyond individual queries, we look at the execution environment of executor nodes. Those nodes have individual constraints of memory and CPU. Executors in Dremio are also part of an engine that groups executors together to process queries in parallel across multiple machines. The size of the engine that a query runs on can affect its performance and ability to handle additional queries.

### CORE PRINCIPLES

- Perform Regular Maintenance: Regularly maintain clusters for optimal performance.
- Scale-out Coordinators: Add secondary coordinators for planning high-volume queries.
- Optimize for Efficiency: Optimize semantic layer and queries before scaling.
- Optimize Engines: Use multiple engines for workload isolation and query buffering.

### BEST PRACTICES

- Clean the KV Store: Regularly clean to prevent metadata fragmentation and improve performance.
- Right-size Scale-out Coordinators: Add coordinators based on query volume to improve query planning concurrency and reduce wait times.
- Design Semantic Layer for Workload Performance: Use data reflections for query optimization and performance.
- Improve Query Performance: Analyze query history and profiles to identify and resolve bottlenecks.
- Rebalance Workload Management Rules: Adjust query cost thresholds to balance query distribution.
- Right-size Engines and Executors: Adjust the number of executors in an engine either manually or via Dremio's auto-scaling feature,, or add new engines based on workload to avoid saturation.

### LEVERAGE REFLECTIONS TO IMPROVE PERFORMANCE

- Iterative Development: Build use cases in Dremio's semantic layer iteratively, without Reflections initially, to identify and analyze slow queries for performance improvement.
- Optimizing with Reflections: Apply raw Reflections on Views with joins or large datasets (e.g., JSON/CSV) to leverage Apache Iceberg materialization for faster query execution and better performance.
- Offloading Heavy Queries: Use raw Reflections to offload analytical queries from operational data stores, reducing the load on OLTP databases and enhancing query performance.
- Joining On-Premises and Cloud Data: Implement raw Reflections for joining on-premises data with cloud data to mitigate latency issues and improve query performance.

## Optimize Metadata Refresh Performance

Dedicated Metadata Refresh Engine: Implementing a dedicated metadata refresh engine in your Dremio cluster isolates metadata refresh activities from other workloads, ensuring business-critical tasks are not impacted and that metadata refreshes complete efficiently and on time.

Add a dedicated engine to isolate metadata refresh workloads from all business-critical workloads

PILLAR 3/6

# Cost Optimization

While getting the best performance possible with Dremio is important, it is also important to optimize your costs associated with managing the Dremio platform.

### CORE PRINCIPLES

- Minimize Running Executor Nodes: Scale nodes to match current load and meet service level objectives.
- Dynamically Scale Executor Nodes Up and Down: Use scale-up and scale-down features based on load.
- Eliminate Unnecessary Data Processing: Avoid unnecessary reflections and metadata processing to optimize costs.

### BEST PRACTICES

- Size Engines to Minimum Nodes Required: Use scripts to scale nodes during low usage periods.
- Remove Unused Reflections: Identify and remove unused reflections to free up resources.
- Optimize Metadata Refresh Frequency: Adjust refresh frequencies to match data update schedules, avoiding unnecessary compute resource usage.

PILLAR 4/6

# Reliability

The reliability pillar focuses on ensuring your system remains operational and can be quickly and efficiently restored in the event of unexpected downtime.

### CORE PRINCIPLES

- Set Workload Management Queue Settings : Protect the system from overload by configuring queues.
- Ensure Regular Backups Can Be Restored: Regularly run and test backups.
- Monitor and Measure Platform Activity: Regularly monitor and measure activity to ensure reliability.

### BEST PRACTICES

- Initialize Workload Management Settings: Set queue memory limits to regulate memory-intensive queries.
- Back Up Dremio: Regularly back up the KV store and critical configuration files.
- Test Restoring Dremio from a Backup: Ensure backups can be reliably restored.
- Monitor Dremio Cluster Health: Use JMX and third-party tools to monitor metrics and set alerts for critical thresholds.

PILLAR 5/6

# Operational Excellence

Following a regular schedule of maintenance tasks is key to keeping your Dremio cluster operating at peak performance and efficiency. This pillar provides details about the tasks that should be periodically completed to maintain an operationally healthy Dremio cluster.

## CORE PRINCIPLES

- Regularly Evaluate Cluster Resources: Assess resource usage as workloads grow.
- Automate Promotion of Catalog Objects from Lower Environments: Use REST APIs for automation.
- Regularly Evaluate Query Performance: Monitor and optimize high-cost queries.
- Regularly Monitor Dremio Live Metrics: Collect and act on metrics to ensure smooth operations.

## BEST PRACTICES

- Optimize Workload Management Rules: Regularly adjust query-cost thresholds based on statistical analysis.
- Configure Engines: Isolate workloads using separate engines for different types of queries.
- Optimize Query Performance: Create semantic layer views without reflections, then add reflections strategically.
- Configure Persistent Logging in Kubernetes Environments: Persist logs on disk for longer availability.
- Monitor Dremio via JMX Metrics: Set up monitoring solutions to proactively identify and resolve issues.

PILLAR 6/6

# Self-Service Semantic Layer

Dremio has a unique capability in its semantic layer, where the physical structure of the underlying data storage is mapped to how the data is ultimately consumed ultimately via SQL queries. When the semantic layer is optimally designed and maintained, the data is easy to discover, queries are easy to write, and performance is optimized.

### CORE PRINCIPLES

- Layer Views: Balance security, performance, and usability by organizing views into Preparation, Business, and Application layers.
- Annotate Datasets to Enhance Discovery and Understanding: Tag and document datasets for easier discovery and governance.

### BEST PRACTICES

- Optimize Workload Management Rules: Regularly adjust query-cost thresholds based on statistical analysis.
- Configure Engines: Isolate workloads using separate engines for different types of queries.
- Optimize Query Performance: Create semantic layer views without reflections, then add reflections strategically.
- Configure Persistent Logging in Kubernetes Environments: Persist logs on disk for longer availability.
- Monitor Dremio via JMX Metrics: Set up monitoring solutions to proactively identify and resolve issues.

## Get the Complete Well-Architected Framework

This summary serves as a practical reference to quickly grasp the core concepts and start implementing best practices in your Dremio deployments.

For a more comprehensive exploration of each pillar, along with in-depth examples and detailed guidance, the full Well-Architected Framework document can be found online in Dremio Documentation.