

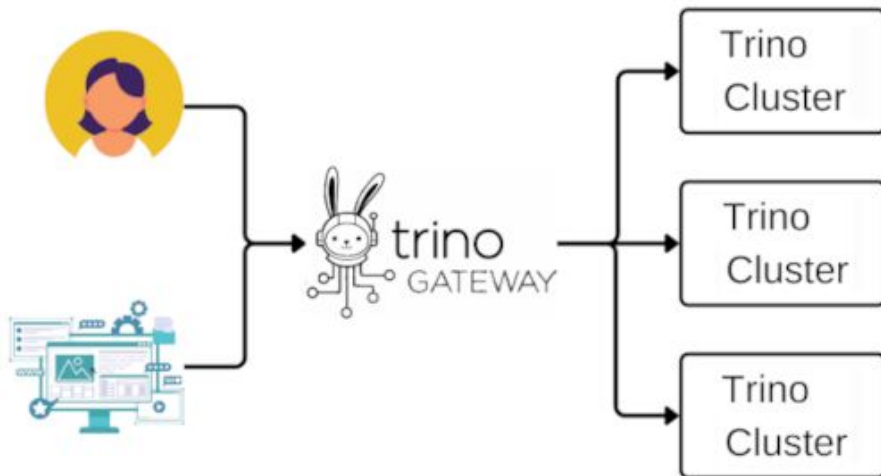
Status and next steps

Presented at the Trino Contributor Congregation in June 2024

What is Trino Gateway?



Trino Gateway is a load balancer, proxy server, and configurable routing gateway for multiple Trino clusters.



More at <https://trinodb.github.io/trino-gateway/>

Project overview



- Evolved from lyft/presto-gateway
- Contributed by Bloomberg, and indirectly Lyft in July 2023
- Presented at [Trino Summit 2023](#)
- Project lead Manfred Moser
- Subproject maintainers Jaeho Yoo, Will Morrison, and Vishal Jadhav
- Contributions from Yuya Ebihara, Star Poon, Andy Tsu, and many others
- Public developer sync every two weeks
- Approx. 300 commits of varying size
- Seven releases so far

Significant improvements



- Huge code, Java version, and dependency upgrades to Trino standards
- Adoption of Airbase and Airlift and removal of Dropwizard
- Access control, authentication, and other security improvements
- New web-based user interface
- New router modules
- New Docker container
- New Helm chart
- New documentation website



Immediate next steps

- Complete the refactor from Dropwizard to Airlift
- Improve configuration to align more with Trino
- Test and polish user interface
- Improve Helm chart and overall deployment experience
- Add more documentation for different use cases
- Establish regular release cadence
- Improve Trino cluster health check logic
- Add support for observability with OpenTelemetry and JMX

Long terms ideas - single pane of glass



A collection of *potential* improvements and existing plans:

- Fully support as cluster of stateless nodes
- Expand user interface with more data and insights
- More router modules including based on query cost estimate, programmable logic, query shape, and used catalogs
- Add support for dynamic catalog storage in database and use for multiple clusters
- Add UI for access control CRUD operations (routing rules)??/
- Test and improve performance, potentially including other backend database
- Add caching layer for metadata, result sets, and other aspects
- Add SPI and user interface for pluggable Trino cluster management tasks like scaling up and down, adding catalogs, and more



Discussion topics

- Add and use Trino API endpoint with worker and running query count without authentication
 - Currently using unofficial endpoint from UI with authentication
- Approaches for shared libraries and concerns
 - Authentication layer
 - Authorization layer
 - Resource groups (already exists, for better or worse)
 - Planner and cost estimation
 - Caching
 - Reuse some coordinator code and workload
- Overlap and impact from Trino connector (cross cluster joins)

Final words



We have come a long way already, ...
but there is an even longer road ahead.

Are you ready to help us and participate?