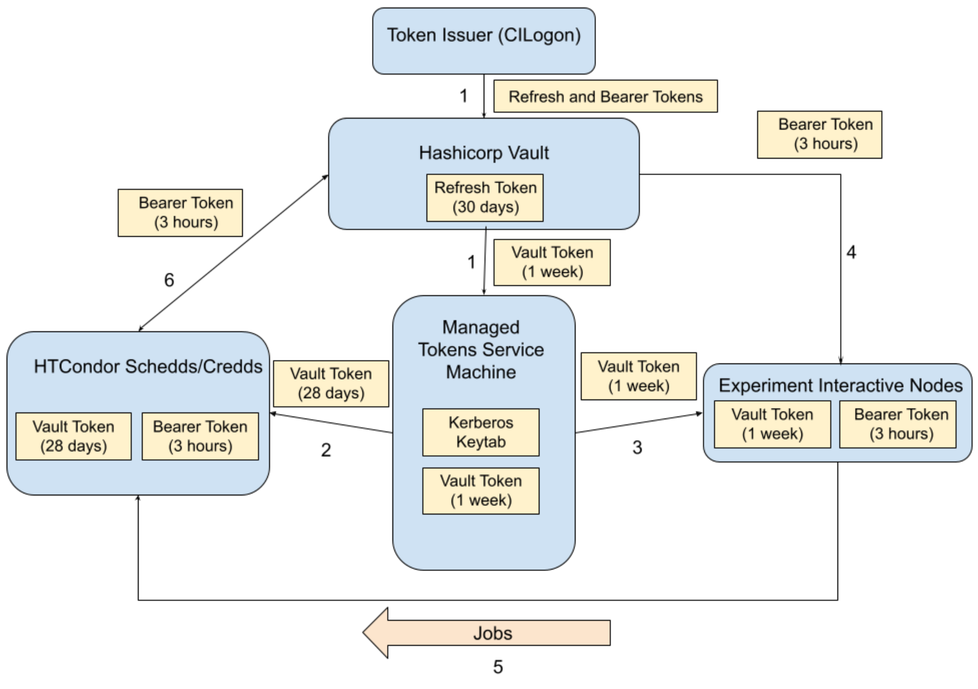
FERMILAB-POSTER-24-0298-CSAID

A *Managed Tokens* Service for Securely Keeping and Distributing Grid Tokens

Shreyas Bhat, Dave Dykstra, Fermi National Accelerator Laboratory (USA)

Background

Fermilab is transitioning authentication and authorization for grid operations to using bearer tokens based on the WLCG Common JWT (JSON Web Token) Profile. One of the functionalities that Fermilab experimenters rely on is the ability to automate batch job submission, which in turn depends on the ability to securely refresh and distribute the necessary credentials to experiment job submit points. Given that Fermilab has numerous experiments, each with their own unique credentials, there was a need for a common system to manage the tokens for best security and to eliminate duplicate effort.

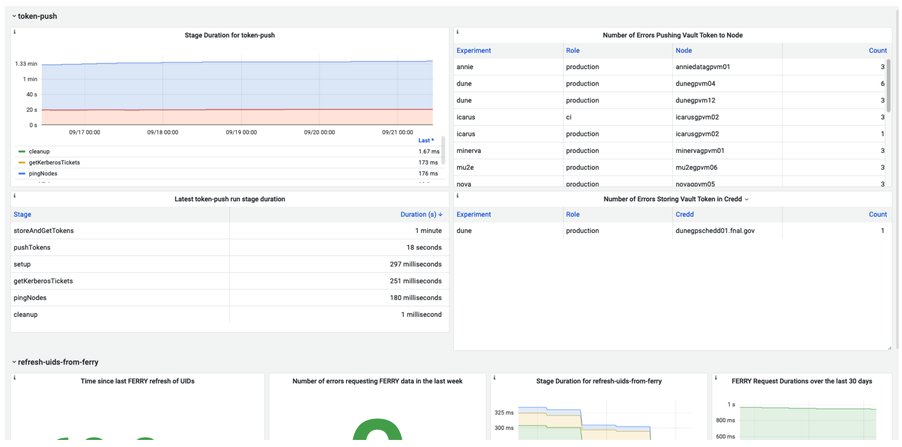


Architecture of *Managed Token Service* within Grid Infrastructure

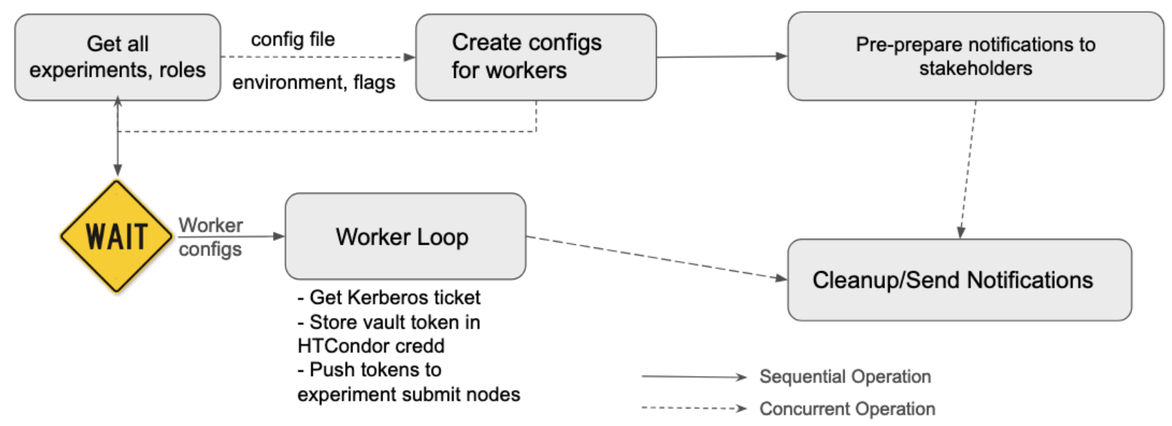
Architecture

We developed a *Managed Tokens* Service to store *Hashicorp* vault tokens in the batch system credential manager (*HTCondor credd*), and also to keep copies of these vault tokens refreshed on the user submit nodes. Other Fermilab grid software can then use *htgettoken* to obtain an access token (*JWT Bearer Token*) using this vault token, and the access token can be used in grid operations. Initial authentication is handled using OIDC authentication by the service operator and renewed using *Kerberos* keytabs.

Since Fermilab hosts many experiments, each with possibly multiple different permission sets (*capability sets*), quick scalability of this service was a top priority. Due to its ability to easily launch and synchronize multiple concurrent operations, we chose *Go* as the language to implement *Managed Tokens*. We were able to quickly design and build a system that spins out a worker thread for each experiment and operation, and synchronize their actions from the main executable, *token-push*.



Managed Tokens* Service Health Dashboard for operators in *Grafana



General program flow for *token-push*, the main *Managed Tokens* service executable

Monitoring

Given that the *Managed Tokens* service sits between many other services and components, being able to monitor the service at varying levels of detail is extremely important. To that end, this service implements the common observability layers.

- The service logs are sent to a *Grafana Loki* instance that runs on Fermilab's central monitoring infrastructure, *Landscape*.
- Metrics are collected regarding operation successes and failures, along with other performance metrics, and these are scraped by the Landscape *Prometheus* server.
- *OpenTelemetry*-compliant *traces* are collected and are viewable through a *Jaeger* instance.
- We built *Grafana* dashboards, again hosted on Landscape, that allow operators and stakeholders to ascertain the performance of the service. These dashboards also will send alerts to operators if there is an issue with the service that requires intervention.

Lessons Learned

The *Managed Tokens* service has been running in production since November 2022, with very few major issues. It has software dependencies that were rapidly evolving at the same time this product was being developed. Taking the time to design the architecture of the product before writing any code, rather than only ensuring it met stakeholder requirements, allowed us to quickly adapt to this changing landscape of the token infrastructure at Fermilab, whether through bugfixes, changes, or added features.

Since *Managed Tokens* was released with a strong monitoring infrastructure, it allowed us to scale with the confidence that we would be able to identify any issues before they disrupted service to stakeholders.

Broader Applicability

The Managed Tokens software is open-source, available at https://github.com/fermitools/managed-tokens. We welcome any contributions, pull requests, and suggestions.

Currently, the service is designed to be usable by anyone hosting experiments that use *htgettoken* to obtain access tokens, *Kerberos*-authenticated *Hashicorp Vault* to store refresh tokens, and *HTCondor* as their batch system. However, we do have plans in the future to broaden the libraries to allow for wider support for different infrastructures.

This work was produced by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.