Custos Secrets: a Service for Managing User-Provided Resource Credential Secrets for Science Gateways

Isuru Ranawaka
Cyberinfrastructure Integration Research Center,
Indiana University
USA
isjarana@iu.edu

Nuwan Goonasekera
Melbourne Bioinformatics,
University of Melbourne
Australia
ngoonasekera@unimelb.edu.au

Enis Afgan
Department of Biology,
Johns Hopkins University
USA
enis.afgan@jhu.edu

Jim Basney
National Center for Supercomputing Applications,
University of Illinois
USA
jbasney@illinois.edu

Suresh Marru
Cyberinfrastructure Integration Research Center,
Indiana University
USA
smarru@iu.edu

Marlon Pierce
Cyberinfrastructure Integration Research Center,
Indiana University
USA
marpierc@iu.edu

ABSTRACT

Custos is open source software that provides user, group, and resource credential management services for science gateways. This paper describes the resource credential, or secrets, management service in Custos that allows science gateways to safely manage security tokens, SSH keys, and passwords on behalf of users. Science gateways such as Galaxy are well-established mechanisms for researchers to access cyberinfrastructure and, increasingly, couple it with other online services, such as user-provided storage or compute resources. To support this use case, science gateways need to operate on behalf of the users to connect, acquire, and release these resources, which are protected by a variety of authentication and access mechanisms. Storing and managing the credentials associated with these access mechanisms must be done using "best of breed" software and established security protocols. The Custos Secrets Service allows science gateways to store and retrieve these credentials using secure protocols and APIs while the data is protected at rest. Here, we provide implementation details for the service, describe the available APIs and SDKs, and discuss integration with Galaxy as a use case.

CCS CONCEPTS

• **Security and privacy** → *Access control*; *Distributed systems security*; **Web application security**.

KEYWORDS

secrets management, science gateways, Galaxy, cyberinfrastructure, cybersecurity, open source software, custos

PEARC '22, July 10–14, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9161-0/22/07…$15.00
https://doi.org/10.1145/3491418.3535177

ACM Reference Format:
Isuru Ranawaka, Nuwan Goonasekera, Enis Afgan, Jim Basney, Suresh Marru, and Marlon Pierce. 2022. Custos Secrets: a Service for Managing User-Provided Resource Credential Secrets for Science Gateways. In *Practice and Experience in Advanced Research Computing (PEARC '22), July 10–14, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3491418.3535177

1 INTRODUCTION

Science gateways hide complexities in accessing scientific computing and data infrastructure, providing user interfaces that are more task-oriented and simpler to use [9]. Traditionally, this meant a graphical user interface in front of one or more HPC clusters. More recently, with the broader adoption of cloud computing and specialized services or APIs, science gateways increasingly broker interactions between multiple third-party services on behalf of their users as well as managing secure connections to community accounts on remote HPC resources. This means that users need to be able to connect their private drop boxes, access their own virtual machines, or import data securely from their own sources into their science gateway account. Enabling this necessitates that science gateways securely manage the secret credentials to those external resources on behalf of the users and support seamless access. However, managing such user secrets is a challenging task because, unlike passwords that can be one-way encrypted, these secrets need to be stored in a readable fashion by the gateways framework. This often means storing unencrypted values in a database. This introduces systemic risk in the event of a server compromise, as the data is not encrypted at rest, there is no audit trail of secret access, and it is not possible to rotate or revoke secrets.

The Galaxy application [8] is a popular open source science gateway framework used primarily for performing bioinformatics analyses. Galaxy supports the execution of jobs across numerous resources, including HPC clusters and clouds, and handles all aspects of data staging. The application provides access to hundreds of domain-specific tools, has graphical support for building workflows, and facilitates sharing of research objects between users. There are large production deployments for Galaxy in the United States

(https://usegalaxy.org/), European Union (https://usegalaxy.eu/), and Australia (https://usegalaxy.org.au/) that support about 50,000 annual users executing several million jobs each year. There are more than 150 additional public Galaxy deployments focused on specific scientific subdomains that are maintained by universities and research groups around the world. With the growth in diversity of uses and evolution of online services, Galaxy users increasingly need to connect their accounts to private or protected resources. Examples include connecting their Galaxy workspace to private resources, such as buckets on Amazon S3 or virtual machines procured on Jetstream [7], as well as accessing password-protected data services, such as CloudStor or Basespace. However, Galaxy has thus far not had a secure mechanism for handling such secrets, relying on unencrypted user preferences or simply not enabling support for tools requiring authentication.

Inspired by the Galaxy use case, as part of the Custos Project [11], we built a secrets management service that science gateways can use to securely handle user-provided secrets. The Custos project (https://usecustos.org/) develops open source software and operates a suite of managed services that help science gateways handle federated user authentication, identity management, group management, and now secrets. The Custos Authentication Service brokers federated user identity between more than 4,500 identity providers (IdPs) and science gateways via CILogon [1]. The service allows users to use their home institution or online social identity to access science gateways while minimizing implementation and management details from the science gateway developers. The Groups Management Service allows users to create groups and manage group membership, which can be used for resource sharing. The goal of Custos is to integrate these capabilities into end-to-end science gateway usage scenarios. A particularly useful scenario is sharing secrets amongst group members. For example, a professor can share access to cloud credentials with students without ever disclosing the secrets in plain text.

The Custos Secrets Service is a managed service that securely stores and controls access to user tokens, passwords, SSH keys, and key-value pairs. The Secrets Service is built on top of the HashiCorp Vault secrets engine and provides highly secured secret storage with strong encryption algorithms. The Secrets Service can be accessed via a REST API or Python and Java SDKs. The Secrets Service offers a holistic user experience where authentication, secrets, and group access controls are uniformly managed.

With the end of support of the Globus Toolkit and X.509 proxy certificates, there is a critical need for science gateways to migrate from X.509 specific approaches like the now-deprecated MyProxy and migrate to approaches that support multiple modern credential types. Building on HashiCorp Vault enables management of multiple types of credentials via widely supported open source software. The original use case that motivated MyProxy, namely the need to bridge from browser-based authentication to non-browser authentication methods used by HPC clusters and other computational resources, still exists. However, science gateways now need a solution that bridges from federated campus (browser-based) authentication, via InCommon, to current methods such as OAuth tokens and SSH keys rather than X.509 proxy certificates.

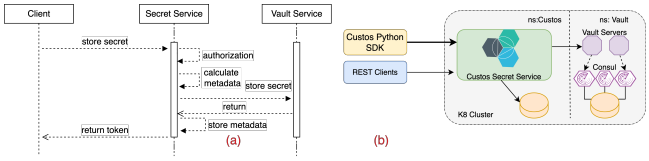

Figure 1: (a) Sequence diagram for message flow for storing secrets in Custos. (b) Implementation and deployment of secrets management service.

The overall Custos-managed service is a multi-tenanted deployment of the open source Custos software operated by the Cyberinfrastructure Integration Research Center, part of the Pervasive Technology Institute at Indiana University (IU). The service is deployed on IU's Intelligent Infrastructure in the data center facilities on the Bloomington and Indianapolis campuses, providing physical security and geographically distributed failover. The services have gone through a security review as well as a penetration testing procedure by TrustedCI.org staff.

2 CUSTOS SECRETS SERVICE ARCHITECTURE

Figure 1 (a) provides a sequence diagram depicting the message flow of storing a secret in the Custos Secrets Service. Prior to the steps shown, a user of a client gateway has authenticated through Custos and obtained an OAuth2 access token, as described in [11]. First, the Secrets Service authorizes requests using OAuth2 access tokens obtained in this step. Custos uses these tokens to identify a particular tenant gateway and the gateway's user. Second, it calculates metadata such as timestamps and internal secret IDs. Third, the secret is stored in a HashiCorp Vault deployment co-located with the Secrets Service, and the calculated metadata are stored in a separate co-located database. Finally, the service returns a querying token to the user of the client gateway.

Figure 1 (b) depicts the implementation and deployment architecture of the Custos Secrets Service. Custos provides REST endpoints and Python and Java SDKs to connect to the Custos Secrets Service. Custos services are implemented as gRPC services (https://grpc.io/), and internal communications use gRPC client stubs. REST endpoints are exposed as public endpoints and implemented using an Envoy Proxy (https://www.envoyproxy.io/), which provides REST-to-gRPC transcoding. Python and Java SDK are implemented on top of generated gRPC client stubs; bindings and SDKs for other programming languages can be generated (https://grpc.io/docs/languages/). All Custos services are implemented and deployed according to the microservices architecture following cloud native practices [11]. The Custos microservices are deployed on a Kubernetes cluster under a Custos namespace. We use Rancher [2] and helm charts [3] to configure and manage the Kubernetes cluster deployment.

We have chosen Vault for secret storage due to its capabilities such as providing strong encryption algorithms, flexible deployment options, secure communication between vault services, and pluggable data storage. Custos provides gateway-specific services on top of Vault. We have created a highly available Vault service deployment as a part of the Custos deployment with the HashiCorp's

Consul as the database. All Vault APIs are restricted for internal communications and only Custos Services can communicate with internal vault servers. External clients are forbidden to directly communicate with Vault APIs.

Building on Vault, the Custos Secrets Service provides capabilities such as controlled access to secrets for users and user groups. The service also maintains different levels of isolation to separate gateway operator-level secrets, user group-level secrets, and individual user-level secrets. Furthermore, the Custos Secrets Service captures essential usage metadata and stores them in a separate database. This enables logging, auditing, and observability capabilities for gateways without exposing the secrets. The Custos Secrets Service manages two secret engines for gateways, one for operational secrets and one for administrative secrets. Administrative secrets are created inside the Custos Secrets Service to manage Custos's internal communications between Keycloak, CILogon or Vault APIs. These secrets are not visible to end users. Operational secrets are stored by end users directly within the Custos Secrets Service.

The Custos Secrets Service provides APIs to store secrets as key value pairs, generate and store SSH keys, and store certificate and password credentials. These API methods are typically used by gateway developers (such as Galaxy), not by end users. [5] lists the REST endpoints. These APIs are designed for general purpose applications and can be adapted as required. All endpoints support GET, PUT, POST, and DELETE operations, and each request must contain metadata related to request routing and authorization.

In addition to basic key-value storage, Custos supports SSH key generation using Java SSH libraries. Client gateways can request a new SSH key or store existing SSH keys using the following endpoint. See [5] for an overview of these methods. The returned token can be used to fetch the generated or stored SSH keys. Similar endpoints can be used to store and fetch X509 certificates.

In addition to REST endpoints, Custos provides language bindings for Python and Java clients, which are based on generated gRPC client stubs. [4] describes how to use the Custos Python SDK to store a secret in Custos. Once authenticated to the Custos services through "IdentityManagementClient" an access token is obtained to use with the ResourceSecretManagementClient ("Custos Secrets Service") APIs. The Custos Service authorizes the user and executes relevant secret management methods.

3 GALAXY USE CASES AND INTEGRATION

We demonstrate the utility of the Secrets Service via integration with the Galaxy application. To address the problems described in the introduction, we have added integration with the Secrets Service, which allows sensitive information to be securely passed between Galaxy and Custos and subsequently made available to integrated tools or data sources. This allows Galaxy users to provide a secret in the Galaxy interface and have Galaxy use that secret to access services requiring authentication. In turn, Galaxy will act on behalf of the user and automatically handle data retrieval or storage or, in the future, compute resource procurement and release (Figure2).

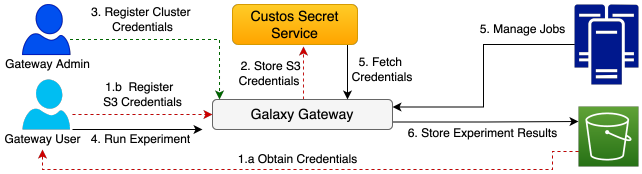

Figure 2: Galaxy integration with Custos to obtain secrets such as access tokens to integrate with user-provided third-party services.

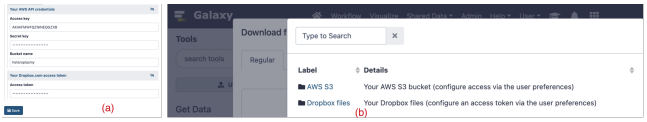

Figure 3: (a) User preferences section of the Galaxy user profile page allowing users to supply and securely store their secrets for accessing specific services. (b) With secrets stored for specific services, namely AWS S3 and Dropbox.com in the figure, a user is able to browse those data repositories from within Galaxy and import data into their workspace.

Galaxy supports a notion of user preferences as key-value pairs that gather arbitrary user settings and can be made available elsewhere in Galaxy (see Figure 3). These settings can now include secrets such as API access keys for a cloud provider, a username and password for an online database, or an expiring service account token that can be automatically renewed. Once the user provides secrets, they use the relevant tool in Galaxy and select an appropriate value from the drop-down menu that is automatically populated from their preferences. Meanwhile, Galaxy automatically handles the retrieval of the necessary values from Custos and supplies it to the service using secure protocols.

To enable this, we have enhanced the Galaxy user preference mechanism to read from and write specific properties to Custos. In Galaxy, user preference properties are annotated with a type in the appropriate configuration file. We have enhanced the "password" type to use the Secrets Service as the data store, making it a secure field. We have also introduced a new type of user preference property called "secret." The key difference is that the type "password" results in the client UI rendering a password field, with the password value being accessible to the client. However, the type "secret" introduces an additional layer of security by making the field write-only for the client, meaning that the unencrypted value cannot be retrieved by the client, and is available only to Galaxy's backend code.

A second place where we have integrated the Secrets Service is with Galaxy's file sources. File source is an abstraction in Galaxy that enables it to interface with external file systems and file system–like databases. File sources allow a user to browse data repositories as well as import and export data. Many data sources require login credentials for a particular file system. We have enhanced file sources to be able to directly access the values from Custos Secrets, enabling access to private repositories.

4 RELATED WORK

Credential management systems include MyProxy, Java Keystore, LDAP implementations, and HashiCorp Vault. MyProxy is an X.509 credential management system that is primarily used in grid computing. XSEDE uses MyProxy as a main credentialing mechanism for users with an approved computational allocation. MyProxy manages only X.509 credentials, hence, an alternative credential management system is required to handle credentials such as SSH keys and OAuth tokens such as SciTokens [12]. Besides, MyProxy does not have any user credentials to shared community accounts mapping, which is required by many science gateway use cases. OAuth for MyProxy (OA4MP) provides an OAuth 2.0 interface to MyProxy but is still X.509 specific without support for SSH keys and other needed credential types. The Java Keystore is a popular mechanism for storing locally accessed keys. OpenLDAP is another solution for managing credentials. However, science gateways need to handle heterogeneous credentials types. HashiCorp Vault is an open source product to secure, store, and control access tokens, and to protect secrets and other sensitive data. The HashiCorp Vault should have its own deployments for applications. For example, WLCG workflows use HashiCorp Vault for managing security tokens [6].

5 CONCLUSION AND FUTURE WORK

The Custos project has enabled a cross-gateway software collaboration in which we can develop core open source software that integrates the best available providers or services, such as CILogon, Keycloak, and Vault, into a single framework that encapsulates end-to-end science gateway use cases.

Using this approach, we have integrated Custos with other science gateway systems, notably the HathiTrust Research Center's Analytics Gateway [10]. The Custos collaboration enables us to collectively tackle new use cases, such as securely integrating user-provided resources as described here. Going forward, our primary efforts will be to increase awareness of these features in the Galaxy community and expand the availability of these services to other interested gateways and potentially other types of cyberinfrastructure.

ACKNOWLEDGMENTS

This work was in part supported by the NSF grants 1840003 and 2005506, and NIH grant U24HG006620.

REFERENCES

[1] Jim Basney, Heather Flanagan, Terry Fleury, Jeff Gaynor, Scott Koranda, and Benn Oshrin. 2019. CILogon: Enabling Federated Identity and Access Management for Scientific Collaborations. *PoS* ISGC2019 (2019), 031. https://doi.org/10.22323/1.351.0031

[2] Steve Buchanan, Janaka Rangama, and Ned Bellavance. 2020. Deploying and using Rancher with Azure Kubernetes service. In *Introducing Azure Kubernetes Service*. Springer, 79–99.

[3] CNCF. 2022. *Helm.* Retrieved April 8, 2022 from https://helm.sh/

[4] Custos. 2022. *Python SDK.* Retrieved April 8, 2022 from https://cwiki.apache.org/confluence/display/CUSTOS/Use+Custos+Python+SDK

[5] Custos. 2022. *REST Endpoints.* Retrieved April 8, 2022 from https://cwiki.apache.org/confluence/display/CUSTOS/Use+Custos+REST+Endpoints

[6] Dave Dykstra, Mine Altunay, and Jeny Teheran. 2021. Secure Command Line Solution for Token-based Authentication. In *EPJ Web of Conferences*, Vol. 251. EDP Sciences, EDP Sciences, France, 02036. https://doi.org/10.1051/epjconf/202125102036

[7] David Y Hancock, Jeremy Fischer, John Michael Lowe, Winona Snapp-Childs, Marlon Pierce, Suresh Marru, J Eric Coulter, Matthew Vaughn, Brian Beck, Nirav Merchant, et al. 2021. Jetstream2: Accelerating cloud computing via Jetstream. In *Practice and Experience in Advanced Research Computing*. 1–8.

[8] Vahid Jalili, Enis Afgan, Qiang Gu, Dave Clements, Daniel Blankenberg, Jeremy Goecks, James Taylor, and Anton Nekrutenko. 2020. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. *Nucleic acids research* 48, W1 (2020), W395–W402.

[9] Katherine A Lawrence, Michael Zentner, Nancy Wilkins-Diehr, Julie A Wernert, Marlon Pierce, Suresh Marru, and Scott Michael. 2015. Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4252–4268.

[10] Isuru Ranawaka, Samitha Liyanage, Dannon Baker, Alexandru Mahmoud, Juleen Graham, Terry Fleury, Dimuthu Wannipurage, Yu Ma, Enis Afgan, Jim Basney, Suresh Marru, and Marlon Pierce. 2021. Science Gateway Integration Examples with the Custos Security Service. In *8th International Workshop on HPC User Support Tools (HUST)*. Zenodo, 9 pages. https://doi.org/10.5281/zenodo.5749727

[11] Isuru Ranawaka, Suresh Marru, Juleen Graham, Aarushi Bisht, Jim Basney, Terry Fleury, Jeff Gaynor, Dimuthu Wannipurage, Marcus Christie, Alexandru Mahmoud, et al. 2020. Custos: Security middleware for science gateways. In *Practice and Experience in Advanced Research Computing*. 278–284. https://doi.org/10.1145/3311790.3396635

[12] Alex Withers, Brian Bockelman, Derek Weitzel, Duncan Brown, Jeff Gaynor, Jim Basney, Todd Tannenbaum, and Zach Miller. 2018. SciTokens: Capability-Based Secure Access to Remote Scientific Data. In *Proceedings of Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) *(PEARC '18)*. ACM, New York, NY, USA, Article 24, 8 pages. https://doi.org/10.1145/3219104.3219135