Approach Paper for Customizing Redmine and Superset for Automatic Gmail Login via Keycloak

Table of Content

Technical Challenges

Background, Context, and Motivation

The goal is to integrate Keycloak with Redmine and Superset to enable Single Sign-On (SSO) using Gmail accounts, with Keycloak acting as the Identity Provider (IdP) and leveraging OAuth 2.0 for authentication. This integration aims to enhance security by centralising authentication, streamline the login process by providing a seamless experience across both platforms, and offer a unified authentication mechanism, improving overall user satisfaction and productivity.

Current System

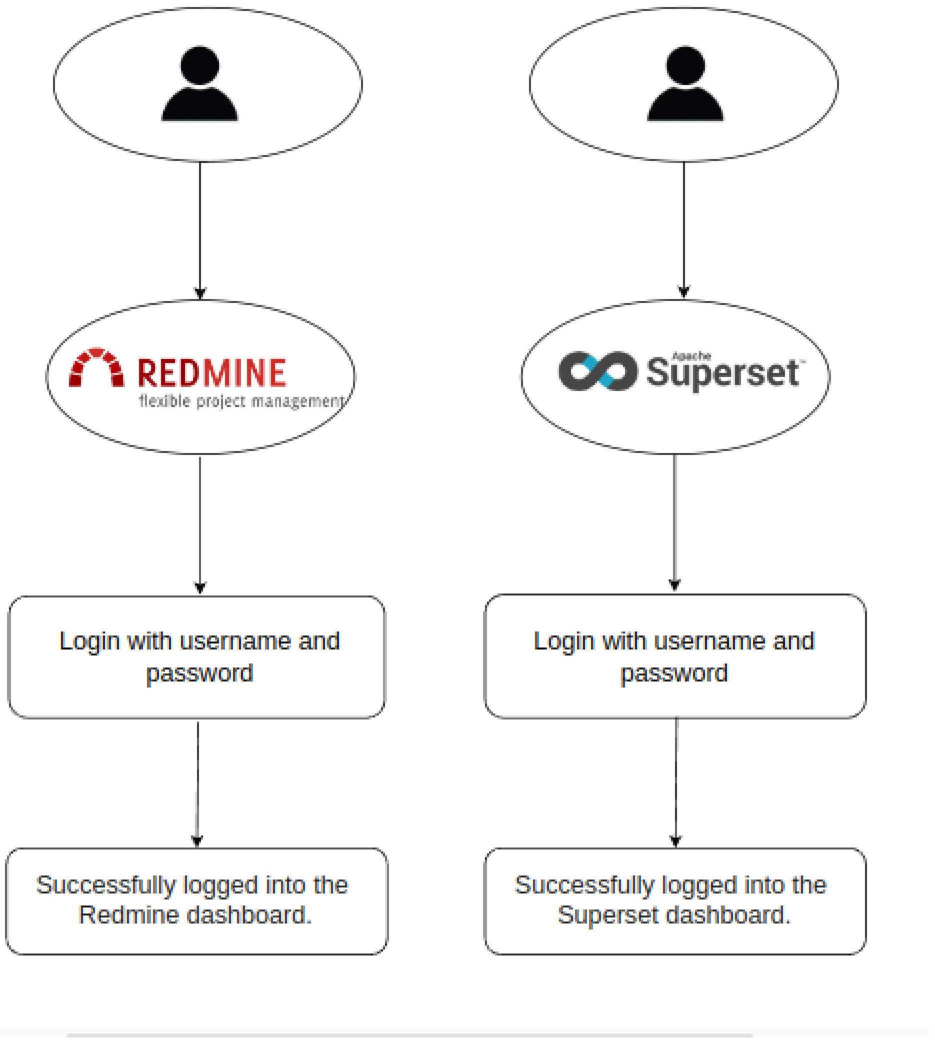
Current System Overview

- **Redmine**: A project management tool used for issue tracking and project management.
- **Superset**: A data exploration and visualisation tool.
- **Authentication**: Users currently login to Redmine and Superset using separate native authentication mechanisms.

Current Scenario

- Users manage separate credentials.
- No centralised authentication.
- Increased security risks and administrative overhead.

Diagrammatic Representation of Current System


REDMINE
flexible project management
Apache Superset
Login with username and password
Login with username and password
Successfully logged into the Redmine dashboard.
Successfully logged into the Superset dashboard.

Proposed Solution

Scope of Work

Integrate Keycloak as the SSO provider for both Redmine and Superset, enabling users to log in automatically using their Gmail accounts. This will involve configuring Keycloak to support SAML 2.0 and setting up Redmine and Superset to authenticate users via Keycloak.

Rationale

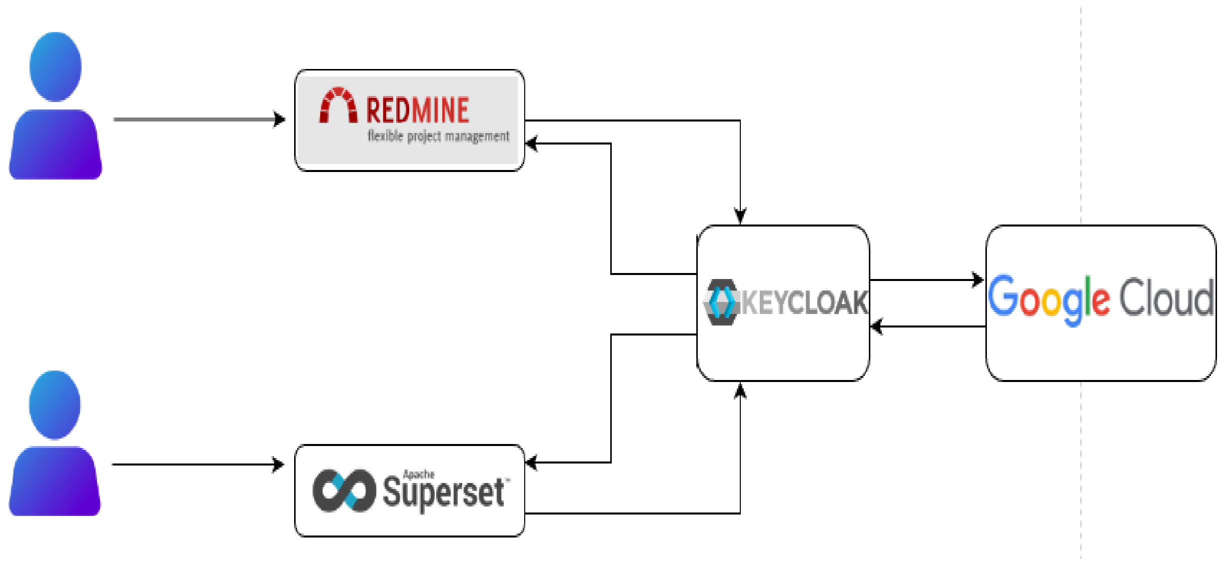
- **Security**: Centralised authentication reduces risks associated with multiple credentials.
- **User Experience**: Users benefit from a streamlined login process with a single set of credentials.
- **Efficiency**: Administrative overhead is reduced by managing user accounts centrally.

Features, Technology, and Benefits

Feature	Technology	Benefit
Gmail-based SSO	Keycloak, SAML 2.0	Seamless and secure login experience
Centralised User Management	Keycloak	Simplified user management and improved security
Integration with Redmine	Redmine SAML Plugin	Efficient project management with streamlined access
Integration with Superset	Superset SAML Integration	Enhanced data exploration with easy access

Proposed Architecture / Application Workflow

Architecture of Proposed System



Pre-requisites

Hardware Requirements

- **Servers**: To host Keycloak, Redmine, and Superset.
- **Recommended**:
  - **Redmine**: 2 CPUs, 4GB RAM, and 50GB storage
  - **Superset**: 2 CPUs, 4GB RAM, and 50GB storage
  - **Keycloak**: 2 CPUs, 4GB RAM, and 50GB storage

Software Requirements

- **Operating System**: Ubuntu 22.04 LTS (Jammy Jellyfish)
- **Keycloak**: Version 24.0.3
- **Redmine**: Version 5.1.2.stable
- **Apache Superset**: Version 4.0.1
- **Database**: PostgreSQL 16.2 for Redmine and Superset

Networking Requirements

- Stable internet connectivity for SAML 2.0 authentication with Gmail.
- Internal network setup to ensure communication between Keycloak, Redmine, and Superset.
- Secure channels (HTTP) for all authentication-related communication.

Handling Keycloak Downtime

Keycloak Downtime Management

Graceful Degradation

- Ensure that both Redmine and Superset can fall back to local authentication if Keycloak is unavailable. Implement a mechanism to detect Keycloak downtime and switch to local authentication seamlessly.

Load Balancing

- Implement load balancing and failover mechanisms for Keycloak. Utilize a load balancer to distribute traffic across multiple Keycloak instances and ensure failover capabilities.

Ensuring High Availability

High Availability Strategies

1. **Cluster Setup**: Deploy Keycloak in a clustered environment to ensure redundancy. This involves setting up multiple Keycloak nodes that can share the load and provide high availability.
2. **Database Replication**: Use database replication for Keycloak to avoid a single point of failure. Ensure that the Keycloak database is replicated across multiple servers to provide redundancy and failover capabilities.

Token Workflow

SAML 2.0 Token Workflow

1. **User Initiates Login**: User clicks on "Login with Google."
2. **Redirect to Google**: User is redirected to Google for authentication.
3. **Google Authenticates User**: User authenticates and consents.
4. **Token Exchange**: Google sends an authorization code to Keycloak.
5. **Access Token**: Keycloak exchanges the code for an access token.
6. **User Info**: Keycloak retrieves user info and authenticates the user.

By following the outlined steps and considerations, integrating Keycloak for automatic Gmail login in Redmine and Superset will enhance security and user experience while simplifying user management. The proposed solutions ensure a smooth and efficient implementation, maintaining high availability and reliability.