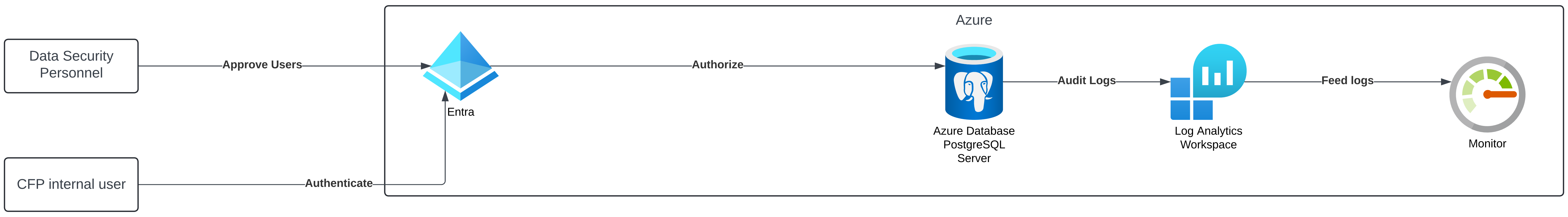
Direct database access

The diagram illustrates a high-level architecture for direct database access in an Azure environment. It shows the flow of user authentication and authorization for accessing an Azure Database for PostgreSQL server. The process begins with Data Security Personnel approving users, who then authenticate through Entra (formerly Azure AD). Entra authorizes access to the Azure Database PostgreSQL Server. The database generates audit logs, which are sent to a Log Analytics Workspace for analysis. Finally, these logs are fed into a monitoring system, likely for real-time alerting and reporting. This setup demonstrates a secure access management process with built-in auditing capabilities, aligning with the compliance requirements mentioned in the task description.



CFP applications access

This diagram illustrates a solution for secure database access and auditing for CFP applications in an Azure environment. Here's a description of the components and their interactions:

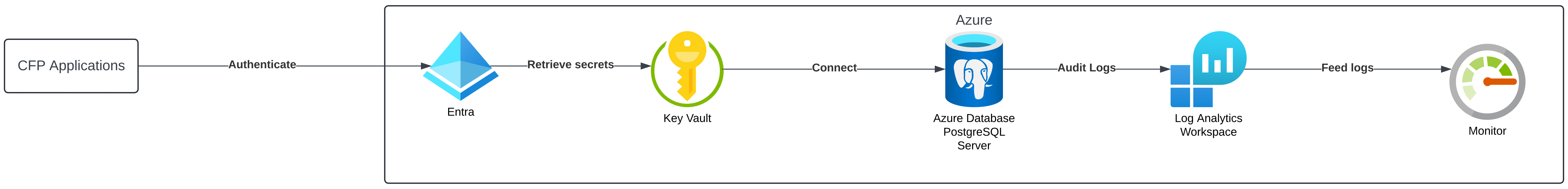
1. CFP Applications: These are the entry point, authenticating with Azure Entra (formerly Azure AD) for secure access.
2. Azure Entra: Handles authentication for the applications and retrieves secrets from the Key Vault.
3. Key Vault: Stores and manages sensitive information like database credentials, supporting the requirement for password rotation.
4. Azure Database PostgreSQL Server: The main database server, which generates audit logs for all operations.
5. Log Analytics Workspace: Collects and analyzes the audit logs from the database server.
6. Monitor: Receives log data for real-time monitoring and alerting.

This architecture addresses several key requirements:

1. End-to-end auditing: All database operations are logged and sent to Log Analytics for comprehensive auditing.
2. Secure authentication: Applications authenticate through Azure Entra, ensuring robust identity management.
3. Credential management: Using Key Vault allows for centralized management of database credentials, facilitating automated password rotation every 30 days without causing downtime.
4. Compliance: The combination of Entra, Key Vault, and comprehensive logging supports compliance requirements.
5. Zero downtime: By using Key Vault for credential management, password rotations can be performed without interrupting application access.

Considerations for implementation:

- Configure Key Vault to automatically rotate database credentials every 30 days.
- Implement a system for Data Security Personnel to approve new user requests, possibly through an automated workflow integrated with Entra.
- Ensure all applications are configured to retrieve database credentials from Key Vault instead of storing them locally.
- Set up alerts in Azure Monitor based on the audit logs to detect any unusual or unauthorized database access attempts.



Password rotation

This diagram illustrates a solution for automated password rotation and auditing for the Azure Database PostgreSQL Server used by CFP. Here's a description of the components and their interactions:

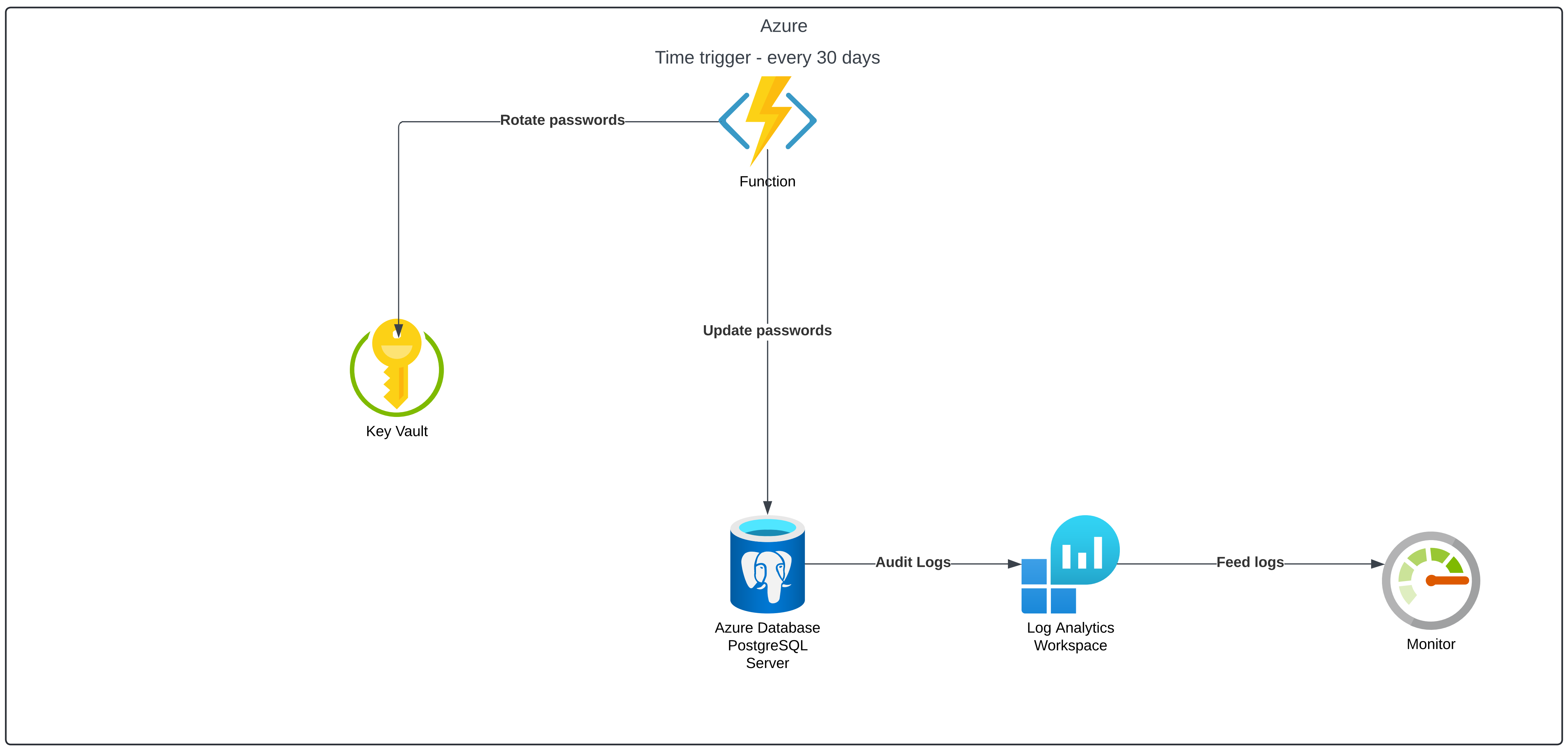
1. Azure Function: Triggered every 30 days, this function initiates the password rotation process.
2. Key Vault: Stores and manages the database credentials securely. The Azure Function rotates the passwords stored here.
3. Azure Database PostgreSQL Server: The main database server. Its passwords are updated by the Azure Function using the new credentials from Key Vault.
4. Log Analytics Workspace: Collects and analyzes audit logs from the database server.
5. Monitor: Receives log data for real-time monitoring and alerting.

This architecture addresses the key requirements:

1. Automated password rotation: The Azure Function, triggered every 30 days, rotates passwords stored in Key Vault and updates them in the database, meeting the requirement for regular credential updates.
2. Zero downtime: By using Key Vault and carefully orchestrating the password update process, the solution can maintain continuous database access for applications.
3. End-to-end auditing: All database operations, including password changes, are logged and sent to Log Analytics for comprehensive auditing.
4. Compliance: The combination of regular password rotation, secure credential storage in Key Vault, and comprehensive logging supports compliance requirements.

Considerations for implementation:

- The Azure Function should be designed to handle potential failures during the rotation process, ensuring database accessibility is maintained.
- Implement proper error handling and notification mechanisms in the Azure Function to alert administrators of any issues during password rotation.
- Configure applications to retrieve database credentials from Key Vault instead of storing them locally, allowing seamless updates without application changes.
- Set up alerts in Azure Monitor based on the audit logs to detect any unusual activities or failed rotation attempts.
- Ensure the Azure Function has appropriate permissions to access Key Vault and update database credentials.
- Implement a separate process for Data Security Personnel to approve new user requests, possibly through an automated workflow integrated with Azure AD (Entra).


Azure
Time trigger - every 30 days
Rotate passwords
Function
Update passwords
Key Vault
Azure Database PostgreSQL Server
Audit Logs
Log Analytics Workspace
Feed logs
Monitor