**Technical Writeup: SQL On-Premises Database Migration to Azure Blob Storage and Dataverse with Key Vault Integration**

**Abstract**

This technical writeup outlines the steps and best practices for migrating an on-premises SQL database to Azure Blob Storage and subsequently to Dataverse, a Microsoft Power Platform service. The migration process includes data transformation steps and secure storage of client credentials using Azure Key Vault for enhanced security.

**Table of Contents**

1. Introduction
2. Prerequisites
3. Data Migration to Azure Blob Storage
4. Data Transformation
5. Azure Key Vault Integration
6. Data Migration to Dataverse
7. Conclusion

**1. Introduction**

Migrating on-premises databases to cloud-based platforms offers numerous benefits, including scalability, reliability, and reduced infrastructure costs. This writeup focuses on migrating an SQL on-premises database to Azure Blob Storage and then to Dataverse, a powerful Microsoft Power Platform service that enables data management and application building.

**2. Prerequisites**

Before initiating the migration process, ensure you have the following prerequisites in place:

* Azure Subscription: Access to an Azure subscription where you can create Azure resources.
* Azure Key Vault: An Azure Key Vault instance to securely store client credentials (client ID and secrets).
* SQL On-Premises Database: The database you want to migrate.
* Azure Blob Storage: An Azure Blob Storage account for storing intermediate data during migration.
* Dataverse Environment: A Dataverse environment where you want to migrate the transformed data.

**3. Data Migration to Azure Blob Storage**

**3.1. Azure Data Factory**

Azure Data Factory is a cloud-based ETL (Extract, Transform, Load) service that can be used to migrate data from your on-premises SQL database to Azure Blob Storage. Here are the steps involved:

1. Create an Azure Data Factory instance.
2. Configure Linked Services:
   * SQL Server Linked Service: Configure connection details for your on-premises SQL database.
   * Azure Blob Storage Linked Service: Configure connection details for your Azure Blob Storage account.
3. Create a Pipeline:
   * Use Copy Data activity to copy data from the SQL database to Azure Blob Storage.

**4. Data Transformation**

To perform data transformations, you can use Azure Data Factory or other ETL tools depending on your specific requirements. Data transformations may include data cleansing, data enrichment, and data format changes. Ensure that the transformed data is saved in a suitable format within Azure Blob Storage.

**5. Azure Key Vault Integration**

Securely store client credentials (client ID and secrets) in Azure Key Vault to enhance security during migration and Dataverse integration:

1. Create an Azure Key Vault instance.
2. Add secrets for client ID and secrets used for Dataverse authentication.
3. Grant appropriate access permissions to the Azure Data Factory or other services to access the Key Vault secrets securely.

**6. Data Migration to Dataverse**

**6.1. Dataverse Integration**

Microsoft Power Platform's Dataverse enables data storage, management, and application development. Follow these steps to migrate data from Azure Blob Storage to Dataverse:

1. Create a Dataverse environment or use an existing one.
2. Create tables and entities in Dataverse that match the structure of your transformed data.
3. Use Power Query or Power Automate to load data from Azure Blob Storage into Dataverse.
4. Configure authentication in Power Query or Power Automate to use the credentials stored in Azure Key Vault.

**7. Conclusion**

Migrating an SQL on-premises database to Azure Blob Storage and Dataverse can be a complex but highly rewarding process. By following the steps outlined in this writeup and leveraging Azure Key Vault for secure credential storage, you can ensure a smooth and secure migration process. This migration opens up opportunities for utilizing Dataverse's powerful features for data management and application development within the Microsoft Power Platform ecosystem.