INTERNSHIP PROJECT REPORT

Project Title: Azure Key Vault – Cryptographic Key & Secret Management

Submitted by: Oleti Raghu Sai Varun

Internship: Smarted Innovations (Training by Embrizon Team)

Date: June 2025

DECLARATION

I, Oleti Raghu Sai Varun, hereby declare that the project report titled “Azure Key Vault – Cryptographic Key & Secret Management” is the outcome of my own work carried out during my internship at Smarted Innovations (Training by Embrizon Team).

This project has been completed under the guidance provided by the training team and is submitted as a part of the internship deliverables. The work presented is original and has not been submitted elsewhere for any academic or professional purpose.

Signature: Oleti Raghu Sai Varun
Date: June 2025

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Smarted Innovations** and the **Embrizon Training Team** for giving me the opportunity to undertake this internship and work on real-time cloud projects using Microsoft Azure.

I am especially thankful to my mentors and trainers at Embrizon for their constant support, valuable guidance, and detailed feedback throughout the course of this project. Their mentorship played a vital role in enhancing my understanding of secure cloud service configuration and management.

I would also like to extend my heartfelt thanks to my family and friends for their continuous encouragement and moral support during the completion of this project.

This internship has been a valuable learning experience and a significant step toward my professional development in the field of **Artificial Intelligence, Machine Learning, and Cloud Technologies**.

Oleti Raghu Sai Varun
Date: June 2025

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

In today's digital landscape, the security of sensitive data such as passwords, connection strings, cryptographic keys, and application secrets has become more important than ever. With the rapid adoption of cloud computing, it is critical to have a centralized and secure method of storing and accessing confidential information across distributed environments.

Microsoft Azure Key Vault is a cloud-based service designed to securely store and manage sensitive information such as secrets, keys, and certificates. It provides a secure environment for applications and users to interact with these elements without directly exposing them to developers, operators, or third-party tools. Key Vault enables tight control over who has access, how information is used, and ensures traceability through auditing and logging features.

This project was undertaken as part of my internship at Smarted Innovations (training by Embrizon Team), where I gained hands-on experience with Azure Key Vault. The goal was to configure and use a Key Vault instance to create a cryptographic key and store a password securely as a secret. Additionally, access to the vault was managed using **Role-Based Access Control (RBAC)** to align with modern cloud security standards.

The implementation was carried out through the Azure Portal interface, which allowed easy monitoring, auditing, and management of vault operations. This report documents the steps taken to complete the task, the outcomes achieved, and screenshots of each critical stage.

Overall, the project provided valuable insight into how enterprise-grade applications can use Azure Key Vault to enhance security, prevent data leaks, and comply with best practices for cloud governance.

CHAPTER 2: OBJECTIVE

The objective of this project is to gain hands-on experience in securely managing secrets and cryptographic keys using **Microsoft Azure Key Vault**, a key component of Azure's security infrastructure. The project is aimed at understanding how to securely store and manage sensitive data such as passwords and encryption keys in the cloud, without exposing them directly to applications or users.

This task involves creating an Azure Key Vault, configuring role-based access using **RBAC (Role-Based Access Control)**, and performing practical operations such as generating a cryptographic key and storing a secret (password). The purpose is to simulate real-world use cases where applications need secure, centralized, and scalable access to confidential information.

The project also aims to familiarize the intern with:

- Key Vault structure and access controls
- Role assignments for secure access
- Secret and key lifecycle management
- Navigating and managing vault contents through the Azure Portal

By completing this objective, the intern builds a solid foundation in cloud security and secret management — essential skills for modern cloud-based and AI-integrated applications.

CHAPTER 3: TOOLS AND TECHNOLOGIES USED

This project involved hands-on interaction with Microsoft Azure services, primarily focused on secure secret and key management using **Azure Key Vault**. The following tools and technologies were used throughout the implementation:

🔹 1. Microsoft Azure Portal

A web-based platform provided by Microsoft for managing Azure resources. It was used to:

- Create and configure the Key Vault
- Manage role-based access control (RBAC)
- Generate cryptographic keys
- Store and retrieve secrets

🔹 2. Azure Key Vault

The core Azure service used in this project. It provides:

- Secure storage for secrets, keys, and certificates
- Access control using RBAC
- Centralized secret management for cloud applications

🔹 3. Azure Identity and Access Management (IAM)

RBAC roles were assigned via IAM to grant appropriate permissions. It enabled:

- Secure access to the Key Vault
- Fine-grained control over who can create, read, or manage keys and secrets

🔹 4. Web Browser (Microsoft Edge / Chrome)

Used to access the Azure Portal interface and monitor all vault operations visually.

🔹 5. Operating System: Windows 10/11

The project was executed from a personal computer running Windows OS, which supports full access to the Azure platform via browser.

CHAPTER 4: KEY VAULT OVERVIEW

In any cloud-based application or infrastructure, managing sensitive information such as passwords, encryption keys, and API tokens is crucial for maintaining security and compliance. Microsoft Azure provides Azure Key Vault, a centralized and secure cloud service designed to protect and control access to such secrets, keys, and certificates.

Azure Key Vault helps prevent accidental data exposure by ensuring that sensitive information is not embedded in application code or stored insecurely. It supports both manual management via the Azure Portal and automated access through applications and scripts using managed identities or service principals.

Each Key Vault is a logical container for a set of cryptographic keys, secrets, and certificates. Access to these items is tightly controlled using Access Policies or the more modern Role-Based Access Control (RBAC) system. In this project, RBAC was used to securely assign the appropriate permissions to the user.

The main capabilities of Azure Key Vault include:

Secure storage and access for secrets (like passwords or connection strings)

Creation and management of cryptographic keys (RSA, EC)

Controlled access to keys and secrets through RBAC or Access Policies

Logging and auditing of operations via Azure Monitor

For this project, a Key Vault named mykeyvaultrsv was created and used. It served as the central location to:

Generate and store a cryptographic key (MyCryptoKey)

Securely store a password as a secret (MyPasswordSecret)

Control who could access these objects through RBAC role assignments

By using Azure Key Vault, cloud applications and users can securely retrieve secrets and keys without hardcoding sensitive information, thereby aligning with industry security best practices.

CHAPTER 5: IMPLEMENTATION STEPS

This chapter outlines the complete step-by-step implementation of Azure Key Vault. The project was executed using the Azure Portal and followed a structured process, from setting up access control to securely storing secrets and keys.

- **Step 1: Open Azure Key Vault**

1. Log in to https://portal.azure.com
2. Navigate to Key Vaults from the search bar
3. Select the existing vault named mykeyvaultrsv

This opened the vault's main dashboard where all configurations were managed.

- **Step 2: Configure Access Control (RBAC)**

1. In the vault's left menu, click on Access control (IAM)
2. Click + Add → Add role assignment
3. In the role list, select Key Vault Administrator
4. Under "Assign access to", choose User, group, or service principal
5. Click + Select members and choose your own Azure user account
6. Click Review + assign to finalize the role assignment

This granted full permission to manage keys and secrets in the vault.

- **Step 3: Create a Cryptographic Key**

1. In the left menu, select Keys
2. Click + Generate/Import
3. Configure the new key:
    - Method: Generate

- Name: MyCryptoKey
- Key Type: RSA
- RSA Key Size: 2048

4. Click Create

The new key MyCryptoKey was successfully created and listed in the vault.

- **Step 4: Store a Secret**

1. Go to Secrets in the vault menu
2. Click + Generate/Import
3. Enter:
    - Name: MyPasswordSecret
    - Value: ABCD@1234567
4. Click Create

This stored the password securely in the vault and made it available for controlled use.

- **Step 5: View the Stored Secret (Optional)**

1. Click on the secret name MyPasswordSecret
2. Open Current Version
3. Click Show Secret Value to view the stored password

This step verified that the secret was stored correctly and accessible to authorized users.

CHAPTER 6: SCREENSHOTS

This chapter contains visual proof of the implementation steps performed using the Azure Portal. Each figure corresponds to a specific step detailed in Chapter 5.

Figure 1 – Azure Key Vault Overview

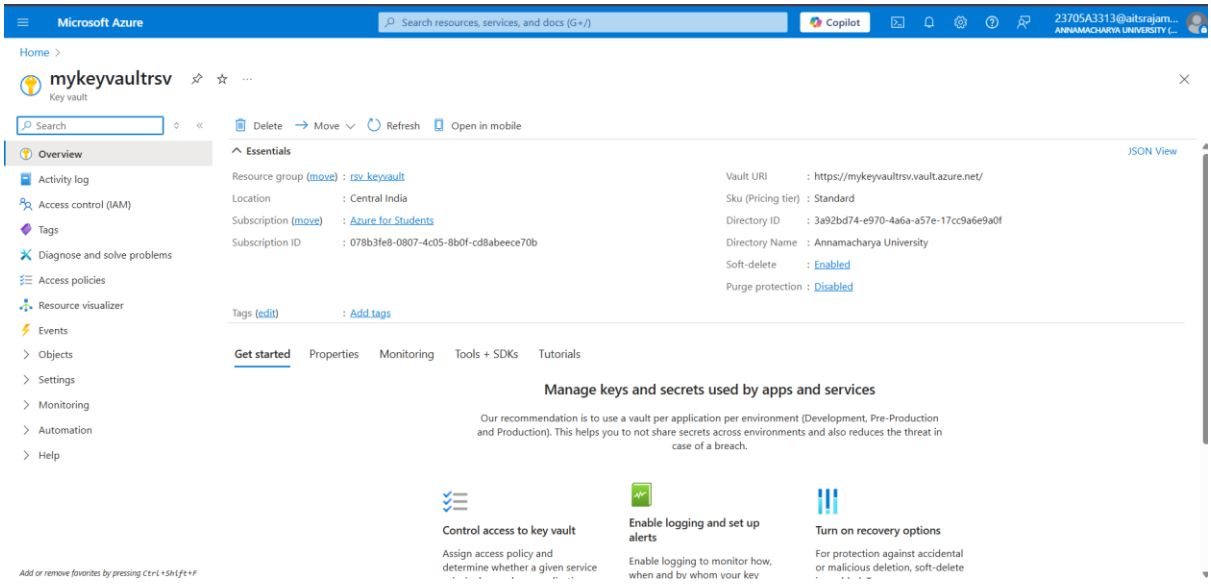


Figure 2 – RBAC Role Assignment

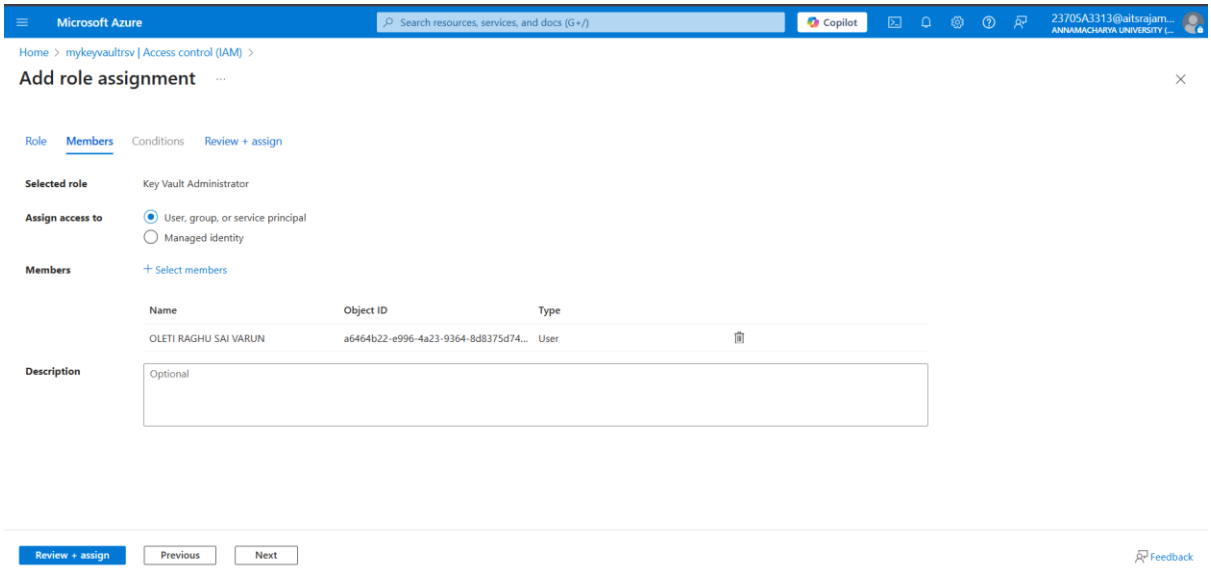


Figure 3 – Key Creation (MyCryptoKey)

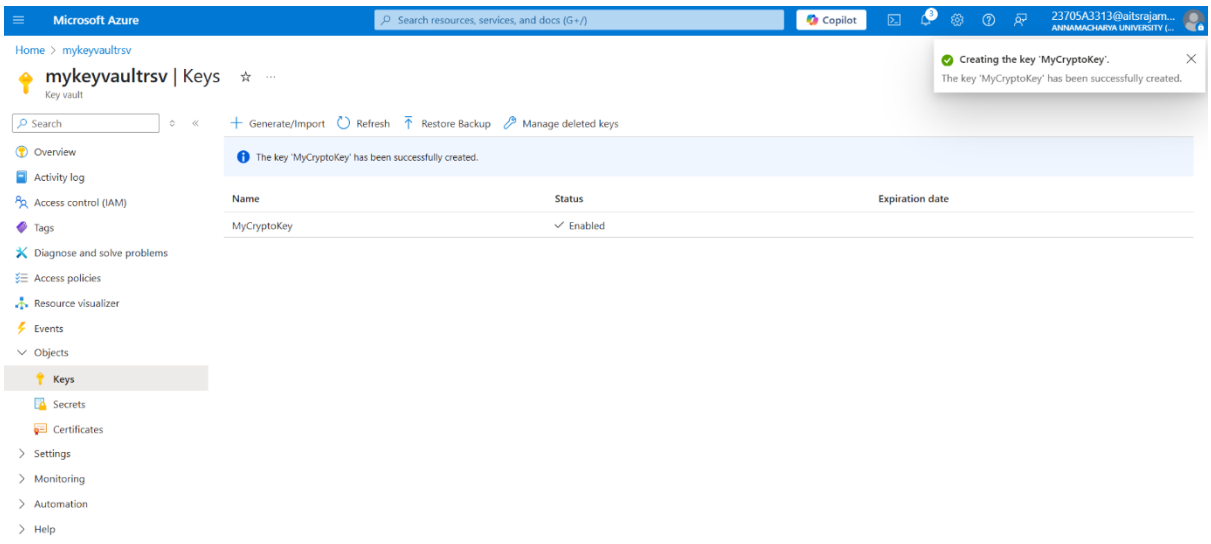


Figure 4 – Secret Storage (MyPasswordSecret)

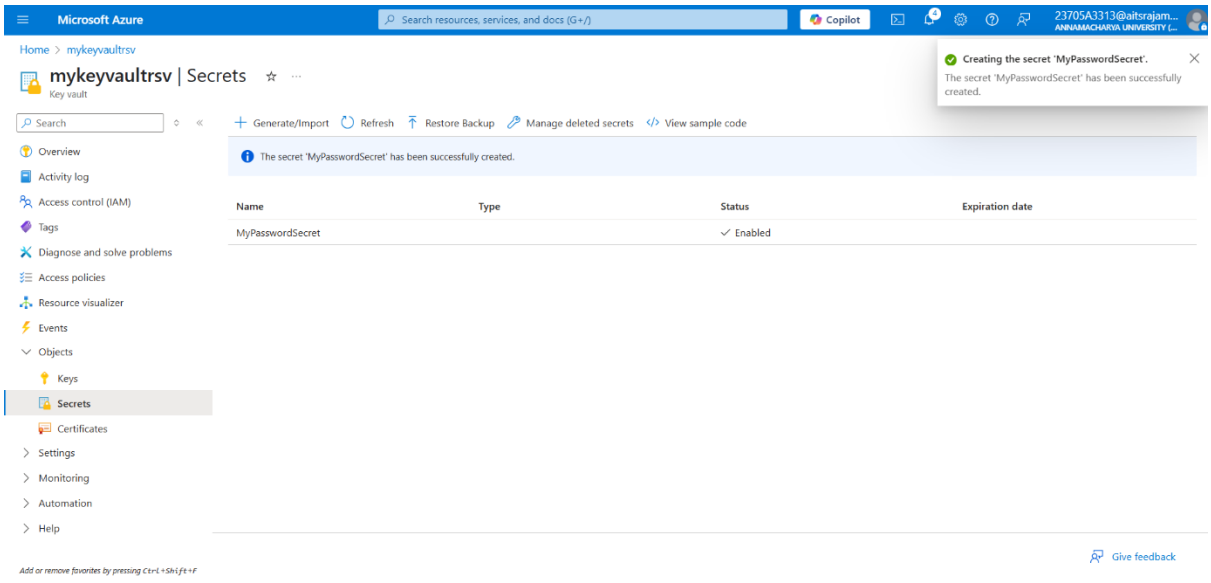
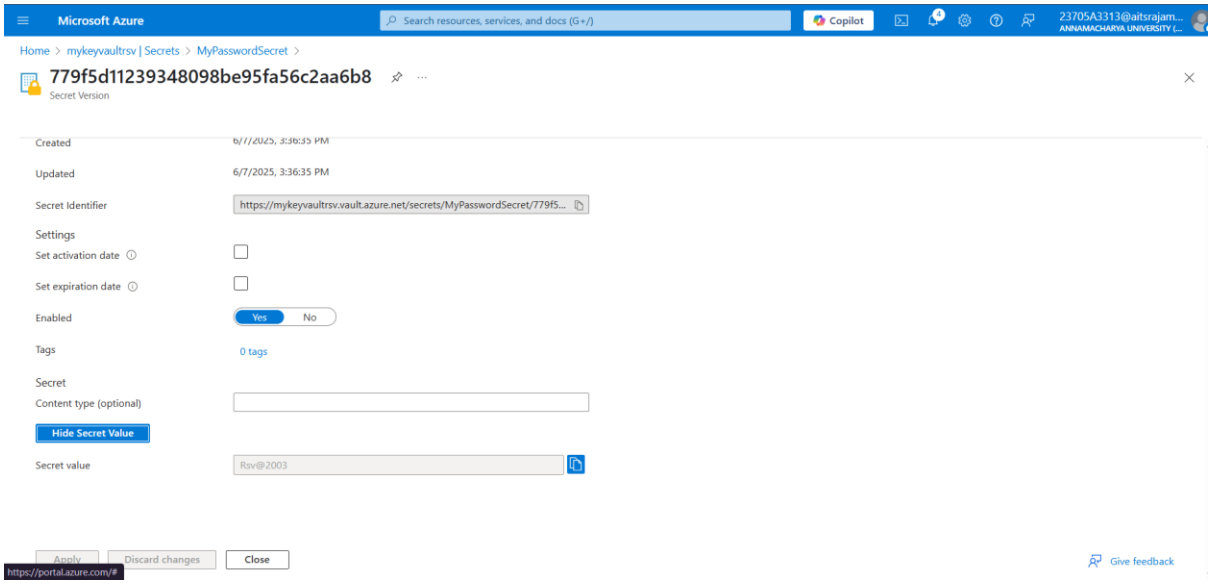


Figure 5 – Viewing Secret Value



CHAPTER 7: RESULT

The project was successfully completed using Microsoft Azure Key Vault through the Azure Portal. The configured Key Vault (mykeyvaultrsv) allowed for secure storage and access control of sensitive information.

The following outcomes were achieved during the implementation:

- ✅ RBAC configuration was applied by assigning the Key Vault Administrator role to the user, ensuring secure access to manage secrets and keys.
- ✅ A 2048-bit RSA cryptographic key named MyCryptoKey was generated and stored within the Key Vault.
- ✅ A secret value (ABCD@1234567) was securely stored in the Key Vault under the name MyPasswordSecret.
- ✅ The stored secret was accessed and verified through the Azure Portal interface.
- ✅ Each step was executed manually through the Azure Portal with appropriate permission controls, ensuring a secure and hands-on learning experience.

All required configurations and operations were successfully carried out without errors. Screenshots were captured at every stage to document the process, and the project met all the intended objectives.

CHAPTER 8: CONCLUSION

This project provided valuable hands-on experience with Microsoft Azure Key Vault, focusing on the secure storage and management of secrets and cryptographic keys. Through the Azure Portal, the entire process — from configuring access controls to storing and retrieving a secret — was executed successfully.

The implementation of Role-Based Access Control (RBAC) ensured that sensitive operations were accessible only to authorized users, aligning with industry best practices for cloud security. Generating a 2048-bit RSA key and securely storing a password in the form of a secret demonstrated the practical utility of Azure Key Vault in real-world cloud environments.

By completing this project, I gained a solid understanding of how centralized key and secret management works within Azure, along with how to enforce proper permissions using RBAC. These are essential skills for designing secure, scalable, and maintainable cloud-native applications.

This task not only helped strengthen my technical skills in cloud security but also gave me a real-world perspective on how critical systems manage confidential data securely in production environments.

CHAPTER 9: FUTURE ENHANCEMENTS

While the current implementation effectively demonstrates the core functionalities of Azure Key Vault, there are several opportunities for enhancement that could further improve the system's scalability, automation, and integration.

- **1. Automate Key Vault Operations Using Azure CLI or PowerShell**

Instead of using the Azure Portal, future implementations can utilize the Azure Command-Line Interface (CLI) or PowerShell scripts to automate the creation of

keys, secrets, and access policies. This approach improves consistency, efficiency, and reproducibility in real-world environments.

- **2. Integrate Key Vault with Applications**

Secrets and keys stored in the vault can be integrated into Azure-hosted web apps, Azure Functions, or containerized services. This allows secure, programmatic retrieval of sensitive information during runtime without hardcoding credentials.

- **3. Enable Logging and Monitoring**

Integrating Azure Key Vault with **Azure Monitor**, **Log Analytics**, and **Diagnostic settings** can help track vault activity, detect unauthorized access attempts, and comply with auditing requirements.

- **4. Enable Soft Delete and Purge Protection**

To prevent accidental or malicious deletion of secrets and keys, soft-delete and purge protection features should be enabled. This adds an extra layer of data protection and recovery capabilities.

- **5. Use Managed Identities for Secure App Access**

Instead of manual secret access, applications can use **Managed Identities** to securely authenticate to Azure services and retrieve secrets from Key Vault without exposing credentials.

These enhancements will help align the project with enterprise-grade security practices, improve automation, and prepare the system for more complex and integrated cloud environments.

CHAPTER 10: REFERENCES

The following official Microsoft documentation was used as a reference for the successful implementation of this project:

1. Microsoft Learn – *Store secrets in Azure Key Vault*
   🔗 https://learn.microsoft.com/en-us/training/modules/configure-and-manage-azure-key-vault/4-store-secrets-in-akv

2. Microsoft Azure Documentation – *Azure Key Vault Overview*
   🔗 https://learn.microsoft.com/en-us/azure/key-vault/general/overview
3. Microsoft Azure Documentation – *Role-Based Access Control (RBAC)*
   🔗 https://learn.microsoft.com/en-us/azure/role-based-access-control/overview