Azure IAM Lab – Role-Based Access & Least Privilege

Author: Alan  
Date: May 2025  
Version: 1.0

# 1. Executive Summary

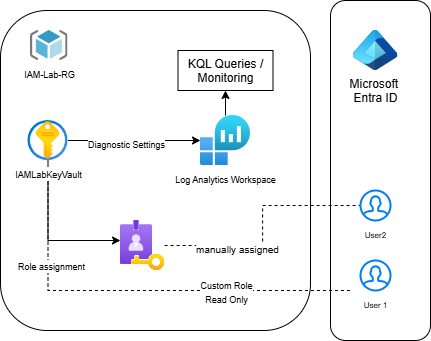
This lab demonstrates how to implement and enforce least privilege access in Microsoft Azure using Role-Based Access Control (RBAC), Azure AD Privileged Identity Management (PIM), and logging via Azure Monitor. It walks through setting up scoped access, time-bound role elevation, and visibility through audit logs and Log Analytics.

# 2. Objectives

- Practice least privilege using custom RBAC roles  
- Configure role assignments at different scopes  
- Enable and use Azure AD PIM for just-in-time access  
- Monitor and log access and role changes  
- Understand the risks of overprivileged access

# 3. Architecture Overview

This lab includes a protected resource (such as a Key Vault), Azure AD users/groups, a custom role, and use of Azure AD PIM. Activity Logs are forwarded to Log Analytics for monitoring.



# 4. Implementation Steps

1. Create the Resource Group and deploy a protected resource (Key Vault or Storage Account)

A screenshot of a computer

AI-generated content may be incorrect.

Why: This will group all your IAM-related resources in one place.

2. Create a Key Vault (the protected resource)

A screenshot of a computer

AI-generated content may be incorrect.

3. Create Azure AD test users

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Why? 2 users are enough to demonstrate the RBAC

3. Create a custom role (e.g., Reader scoped to Key Vault)

A screenshot of a computer

AI-generated content may be incorrect.

Why? By creating custom roles, we specify what resource group we want to grant access to, enforcing the least privilege principle

4. Assign roles to users at different scopes

A screenshot of a group of members

AI-generated content may be incorrect.

Why? Now User1 can read secrets in the Key Vault — *and nothing else*.

A screenshot of a computer

AI-generated content may be incorrect.

User2 was manually assigned Contributor role scoped to the Key Vault. This simulates elevated access normally managed via PIM (PIM wasn’t enable for this lab due subscription and cost limitations)

5. Enable diagnostic logs and link to Log Analytics

A screenshot of a computer

AI-generated content may be incorrect.

Why? Logs need to be sent to Log Analytics Workspace to be retrieved

6. Run KQL queries to analyze access and elevation activity

A screenshot of a computer

AI-generated content may be incorrect.

# 5. Security Considerations

- Review built-in roles to enforce least privilege  
- Limit role assignments to the lowest necessary scope  
- Use PIM to reduce persistent privileges (was manually assigned Contributor role scoped to the Key Vault. This simulates elevated access normally managed via PIM)  
- Regularly review access logs and role activations

# 6. Findings / Results

# Custom role successfully limited user access to a single Key Vault, enforcing read-only access to secrets. The Contributor role was manually assigned to simulate elevated access in place of PIM (not available in the current tenant). After a short delay, User2 was able to create, update, and delete secrets — confirming scoped access worked as expected. Diagnostic settings were configured, and Log Analytics was connected. While access logs initially took time to appear, KQL queries helped validate secret operations like SecretGet and SecretSet.

# 7. Conclusion

The IAM Lab proves how Azure RBAC, PIM, and logging tools can be used to implement and enforce true least privilege. The configuration supports security best practices while offering flexibility and traceability in access control.