
🧑🏽‍💻 Role: Cloud Security Engineer (Azure)

🔧 Project Title: Simulated Cloud Misconfigurations in Azure (Storage + IAM)

🛠️ What You Did (Step-by-Step)

✅ 1. Created Two Azure Storage Accounts

You provisioned two storage accounts in Azure to simulate a **secure environment** vs a **misconfigured (vulnerable) environment**:

- **Storage Account 1**: `securestorage<name>`
  - Purpose: Demonstrate a locked-down, secure configuration.
- **Storage Account 2**: `misconfiguredstorage<name>`
  - Purpose: Simulate real-world vulnerabilities.

✅ 2. Configured Blob Container Access

Azure Storage Accounts allow file/object storage via **Blob Containers**.

You created one container in each account:

Container Name	Access Level	Purpose
`private-container`	🔒 Private (no anonymous access)	Secure – Only authenticated users can access data
`public-container`	🌍 Public (anonymous read access enabled)	Misconfigured – Anyone can access blob data via URL

✅ 3. Modified Storage Account Settings

You explicitly changed the **storage account-level policy** to:

- Enable **public access** for the misconfigured account.
- Keep it **disabled** for the secure one.

This replicates a common mistake: forgetting to disable public access at the account level.

✅ 4. Assigned IAM Roles (RBAC Misconfiguration)

You tested Azure **Identity and Access Management (IAM)** by assigning different role-based access levels:

Role Name	Assigned To	Purpose	Risk Level
`Storage Blob Data Reader`	(User A / You)	Secure – Can only **view** blob data	✅ Secure
`Storage Blob Data Owner`	(User A / You)	Misconfigured – Can **view, edit, delete**, and change access settings	❌ High Risk

This shows how **overly permissive roles** can lead to:

- Accidental data deletion
- Unauthorized sharing
- Escalation of access

💡 What Are Azure Storage Accounts?

- Azure **Storage Accounts** are cloud containers for storing:
    - Blobs (files)

  - Queues
  - Tables
  - File Shares

- You used **Blob Storage** specifically, which is ideal for unstructured data (like documents, images, or backups).

✅ Summary (Put This on Your Resume/Portfolio)

> **Designed and deployed a secure vs. misconfigured Azure cloud environment to demonstrate common storage and IAM misconfigurations. Configured public blob containers, enforced/disabled public access, and applied role-based access controls (RBAC) to simulate real-world privilege escalation and data leakage scenarios.**

Let me know if you'd like this turned into a downloadable **PDF**, **Markdown report**, or slide deck.

Great question — by adding users in **Microsoft Entra ID** (formerly Azure Active Directory) under the **Default Directory**, here's what you did **as a Cloud Engineer** — broken down with clarity:

🧑🏾‍💻 Action Taken: User Identity Creation in Microsoft Entra ID

🛠️ What You Did:

You **created new users** in your tenant's **Azure Active Directory (AAD)** — now branded as **Microsoft Entra ID**.

🧩 What This Means Technically

- You added **individual user identities** under your **Default Directory** (the root Entra tenant).
- These users can now be:
  - Assigned **IAM roles** (e.g., Reader, Owner, Contributor)
  - Given **access to Azure resources**
  - Used to simulate different roles in security testing
- They are **directory-bound identities** that can authenticate to Azure services.

🔐 Why This Matters in a Security Project

You were simulating a real cloud team environment where:

- Different users have **different levels of access**
- You can **test misconfigurations** like:
  - Assigning **too much access** to a low-level user
  - Seeing if a **non-admin can escalate privileges**
  - Testing **anonymous blob access** vs authenticated user access

---

✅ What You Did (Cloud Engineer Perspective)

> **Provisioned multiple user identities in Microsoft Entra ID (formerly Azure AD) under the Default Directory for role-based access control testing and simulation of real-world IAM misconfigurations in Azure. Enabled secure and insecure access scenarios for resource access validation.**
