**Research and Development Document**

**ON**

**Azure Multifactor Authentication (MFA)**

**as**

**Internship Project**

****

****

**By**

**Purva Sharma**

**(CT\_CSI\_CI\_4869)**

**Under**

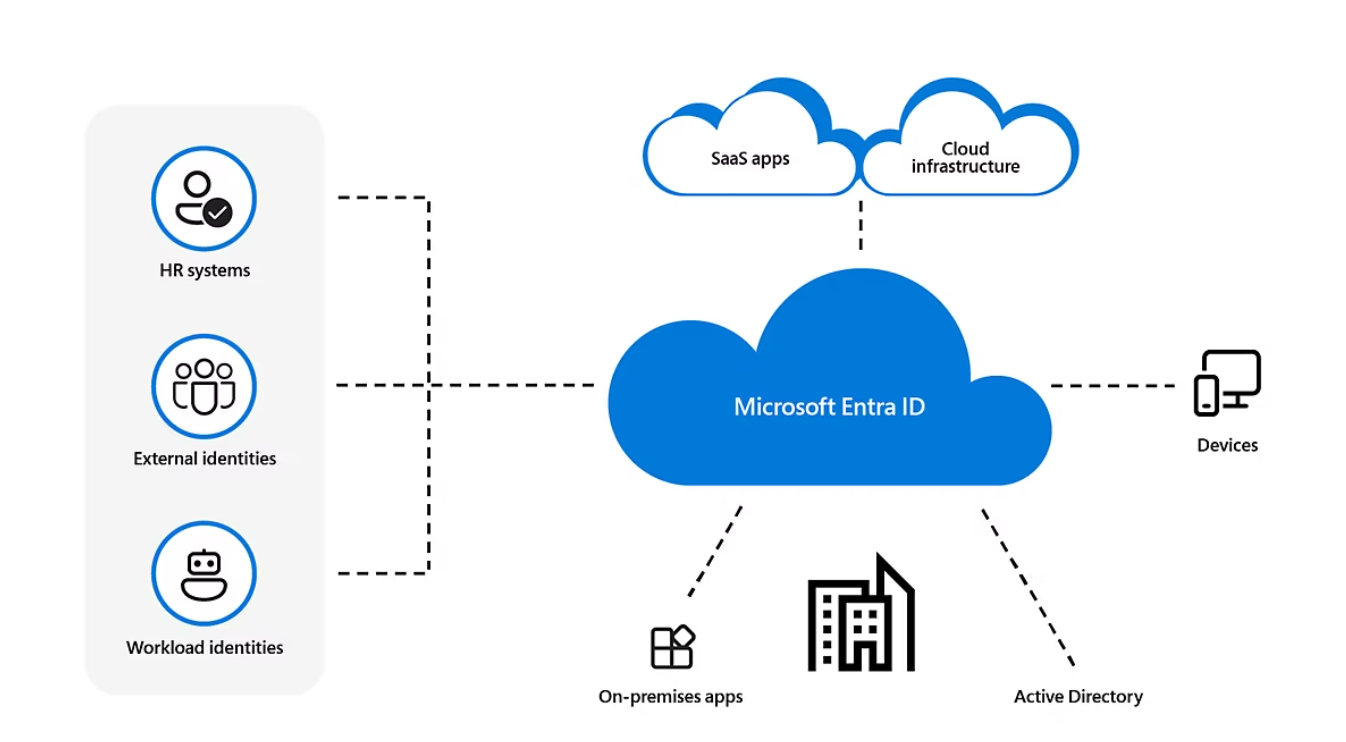
**Celebal Summer Internship**

**in**

**Cloud Infra & Security**

**Azure Multifactor Authentication (MFA)** is a security feature provided by **Microsoft Entra ID (formerly Azure Active Directory)** that requires users to verify their identity using more than one method of authentication. It adds an additional layer of protection to user sign-ins and transactions beyond just a username and password.

**Microsoft Entra ID –** also known as **Azure Active Directory (AD),** is a cloud-based identity and access management service that helps organizations secure and manage user identities and access to resources



**Use of Azure Active Directory**

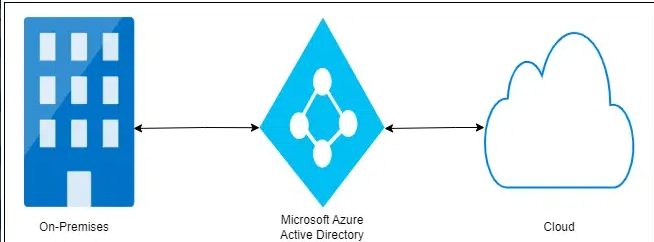
Suppose you have a large organization with a lot of developers. Some Azure services must be available to all developers for them to perform their responsibilities. When the administrator gives them a unique username and password for each service, they can access services like databases, virtual machines, or Azure storage services. It might be challenging for administrators and employees to manage many user logins at once.

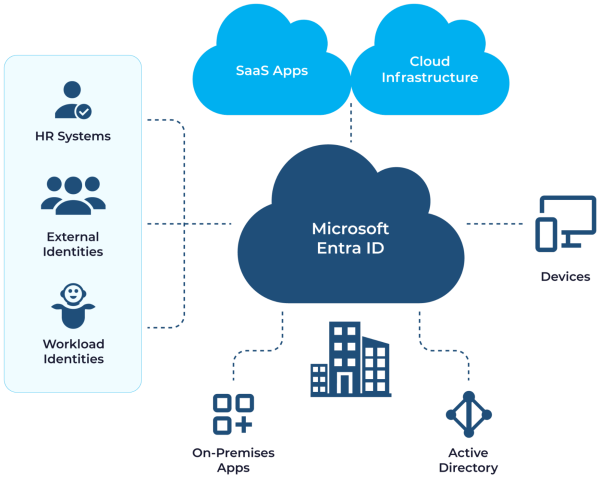
Azure Active Directory (AD) enters the scene in this situation. Administrators can easily manage numerous user logins with Azure AD. To access each service, administrators must provide a single login and password in Azure. It is used by:

* Administrators
* Developers
* Users

**Structure of Azure AD/ Microsoft Entra ID:**

Azure Active Directory (Azure AD) is structured as a cloud-based directory and identity management service with a flat hierarchy. It organizes resources into tenants, where each tenant represents a dedicated and isolated instance of Azure AD. Within a tenant, **users**, **groups**, and **applications** are managed**. Users are individual accounts**, **groups are collections of users**, and **applications are registered entities that Azure AD can authenticate**. Additionally, administrators can set up roles and permissions to control access and enforce policies across these resources.





**Working of Azure AD/Microsoft Entra ID:**

Azure Active Directory (Azure AD) simplifies identity and access management in the cloud. Users authenticate with Azure AD credentials, enabling secure access to applications and services. Single sign-on (SSO) streamlines user experience by allowing access to multiple resources with one login. Robust security features like multifactor authentication (MFA) and access policies ensure secure access control. Azure AD Connect facilitates seamless integration between on-premises and cloud environments for unified identity management.

**Licensing of Microsoft Entra ID:**

**Licenses of Entra ID are:**

* **Microsoft Entra ID Free:**Provides user and group management, on-premises directory synchronization, basic reports, self-service password change for cloud users, and single sign-on across Azure, Microsoft 365, and many popular SaaS apps.
* **Microsoft Entra ID P1**: In addition to the Free features, P1 also lets your hybrid users access both on-premises and cloud resources. It also supports advanced administration, such as dynamic membership groups, self-service group management, Microsoft Identity Manager, and cloud write-back capabilities, which allow self-service password to reset for your on-premises users.
* **Microsoft Entra ID P2**: includes features in addition to the features included in Free and P1. P2 includes Microsoft Entra ID Protection to help provide risk-based Conditional Access to your apps and critical company data and Privileged Identity Management to help discover, restrict, monitor administrators, their access to resources and to provide just-in-time access when needed.

**In addition to Microsoft Entra ID licenses, you can enable additional identity management capabilities with licenses for other Microsoft Entra products, including:**

* **Microsoft Entra ID Governance. Microsoft Entra ID Governance is an advanced set of identity governance capabilities for Microsoft Entra ID P1 and P2 customers.**
* **"Pay as you go" feature licenses. You can also get licenses for features such as Microsoft Entra Domain Services, and Microsoft Entra customer identity and access management solution (CIAM). CIAM can help you provide identity and access management solutions for your customer-facing apps. For more information, see our next-generation solution for external identities, Microsoft Entra External ID.**

**Basic Terminology of Entra ID:**

* **Identity:** A thing that can get authenticated. An identity can be a user with a username and password. Identities also include applications or other servers that might require authentication through secret keys or certificates.
* **Account:** An identity that has data associated with it. You can’t have an account without an identity.
* **Microsoft Entra account:** An identity created through Microsoft Entra ID or another Microsoft cloud service, such as Microsoft 365. Identities are stored in Microsoft Entra ID and accessible to your organization's cloud service subscriptions. This account is also sometimes called a Work or school account.
* **Tenant:** A dedicated and trusted instance of Microsoft Entra ID. The tenant is automatically created when your organization signs up for a Microsoft cloud service subscription. These subscriptions include Microsoft Azure, Microsoft Intune, or Microsoft 365. This tenant represents a single organization and is intended for managing your employees, business apps, and other internal resources.

**1. Configure & manage Azure Multifactor Authentication (MFA)**

**Enabling MFA for Users:**

Azure MFA can be enforced using:

* **Per-user MFA:** Manually enable MFA for individual users.
* **Conditional Access Policies (recommended):** Define policies based on user location, device state, and more.

## **Supported MFA Methods Diagram:**

MFA Verification

Authenticator App

SMS

Call

FIDO2 key

**2. Two-Factor Authentication (2FA)**

**2FA** is a type of **MFA** that requires two forms of identity from different categories:

* Something you know (password)
* Something you have (mobile device or hardware token)

## **Comparison Table:**

|  |  |
| --- | --- |
| **Authentication Type** | **Description** |
| Password | Knowledge-based |
| OTP via SMS/App | Possession-based |
| Biometrics | Inherence-based |
|  |  |

**Benefits of 2FA:**

* **Blocks unauthorized access**
* **Reduces identity theft**
* **Required for many regulatory frameworks**

**Authentication**

**Authentication is the process of challenging a person, software component, or hardware device for credentials to verify their identity or prove they're who or what they claim to be. Authentication typically requires the use of credentials**

**Multifactor authentication (MFA)** is a security measure that requires users to provide more than one piece of evidence to verify their identities, such as:

* Something they know, for example a password.
* **Something they are, like a biometric (fingerprint or face).**

**Single sign-on (SSO)** allows users to authenticate their identity once and then later silently authenticate when accessing various resources that rely on the same identity. Once authenticated, the IAM system acts as the source of identity truth for the other resources available to the user. It removes the need for signing on to multiple, separate target systems.

**3. Methods of Two-Factor Authentication**

**1. Microsoft Authenticator App**

* Sends push notification
* Time-based One Time Password (TOTP)

**2. SMS Verification**

* One-time passcode sent via text

**3. Phone Call**

* Automated call with code

**4. FIDO2 Security Keys**

* USB/NFC-based hardware authentication

**5. Windows Hello**

* Biometric sign-in for Windows 10/11

**Difference between MFA and 2FA**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Two-Factor Authentication (2FA)** | **Multi-Factor Authentication (MFA)** |
| Number of Factors | Requires two distinct factors (e.g., password + code from phone) | Requires two or more factors |
| Flexibility | Less flexible; limited to two factors | More flexible; allows for various combinations of factors |
| Security Level | Generally provides good security, especially compared to single-factor authentication | Considered more secure than 2FA due to additional layers of verification |
| User Experience | Usually more streamlined and user-friendly due to fewer steps | May introduce more friction to the login process, depending on the number and type of factors |
| Examples | Password + SMS code, Password + Authenticator app code | Password + Fingerprint + Security key |
| Relationship to MFA | A specific type of MFA | Encompasses any authentication method with two or more factors |

**4. Setup Self-Service Password Reset (SSPR)**

**Self-Service Password Reset (SSPR)** is an Azure AD feature that allows users to reset their own passwords without administrator or helpdesk intervention, as long as they verify their identity using configured authentication methods.

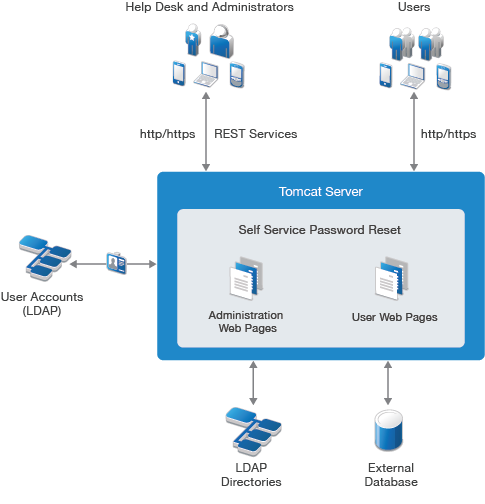
**Allows users to reset passwords securely without admin help.**

**Benefits of SSPR:**

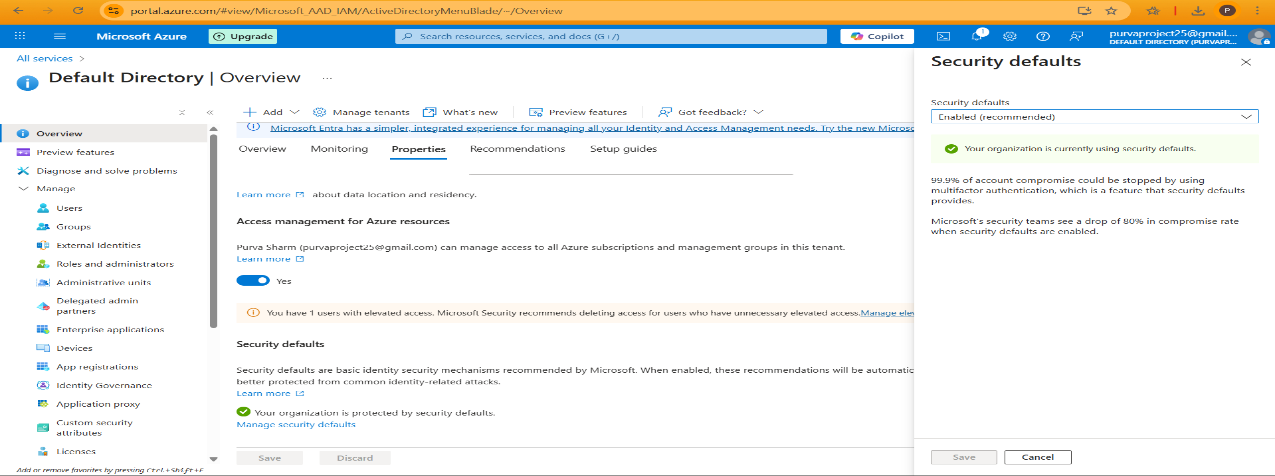
* Reduces support/helpdesk workload
* Enables users to unlock accounts or reset passwords anytime
* Improves security with MFA during reset
* Enhances productivity and user satisfaction

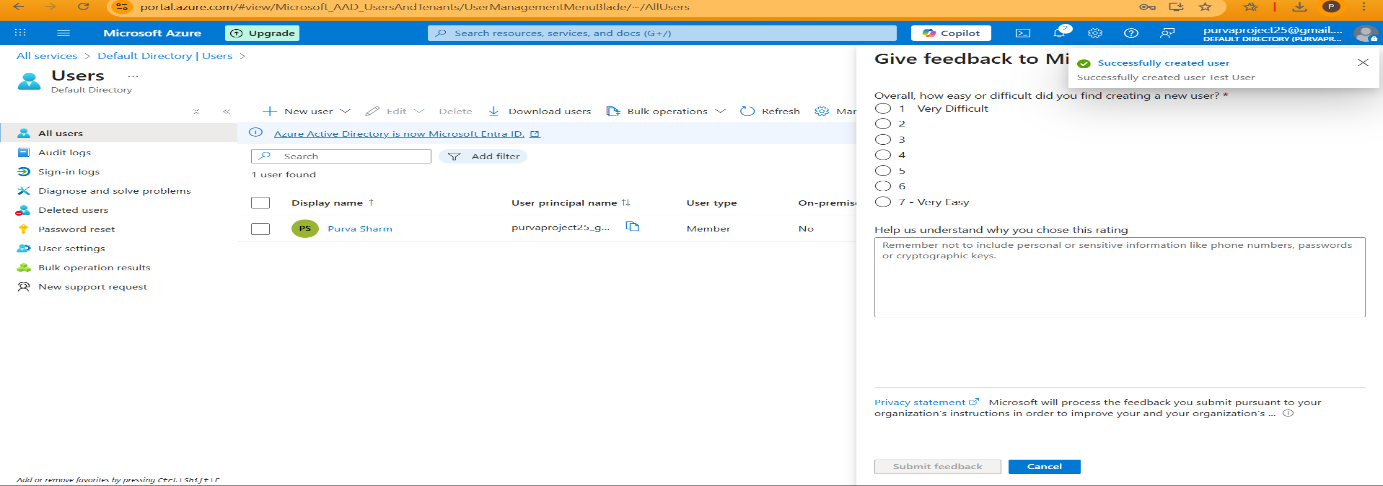
**How SSPR Works:**

1. User forgets password and visits the reset page:  
   https://passwordreset.microsoftonline.com
2. User verifies identity using one or more authentication methods (e.g., phone, email, security questions, Microsoft Authenticator).
3. User sets a new password securely.

****

**5. Setting up for Roles and Licences**



******

**Steps:**

1. **Verify License**
   * **Go to: Azure AD > Licenses > Overview**
   * **Ensure Azure AD Premium P2 is assigned to users and admins.**
2. **Check Required Roles**
   * **You must be either:**
     + **Global Administrator**
     + **Privileged Role Administrator**
3. **Consent to PIM**
   * **Go to: Azure AD > Privileged Identity Management**
   * **Click Consent to PIM if prompted.**
4. **Enable PIM for Azure AD Roles**
   * **Go to: PIM > Azure AD roles > Manage > Roles**
   * **Click Discover roles > Select > Click Enable PIM.**

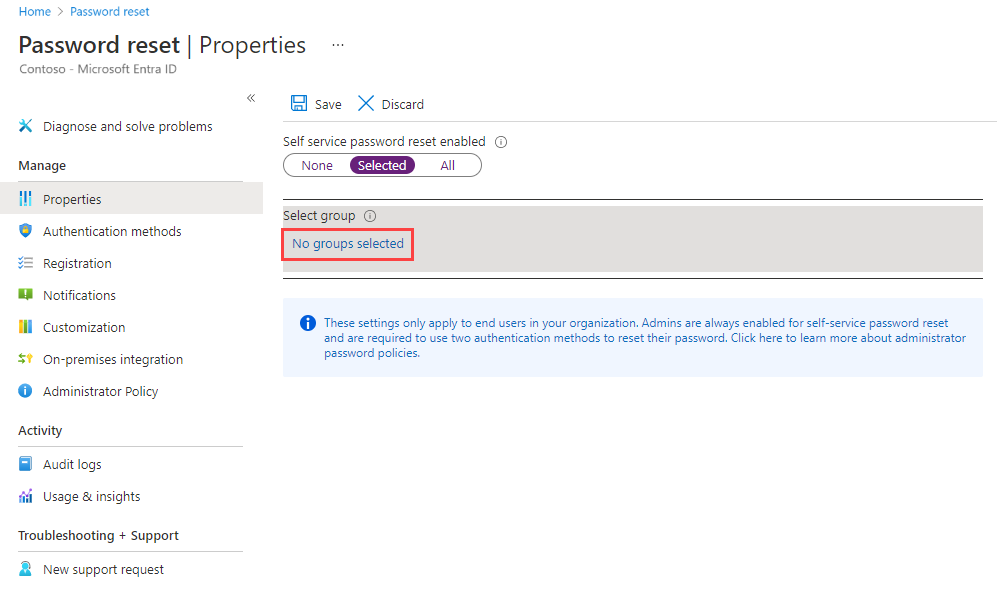
**6. Deploy Self-Service Password Reset (SSPR)**

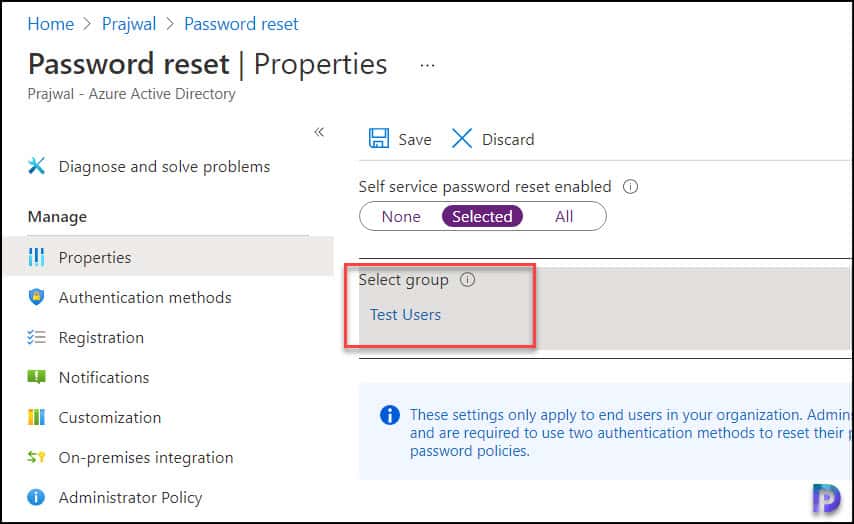
**Configuration Steps:**

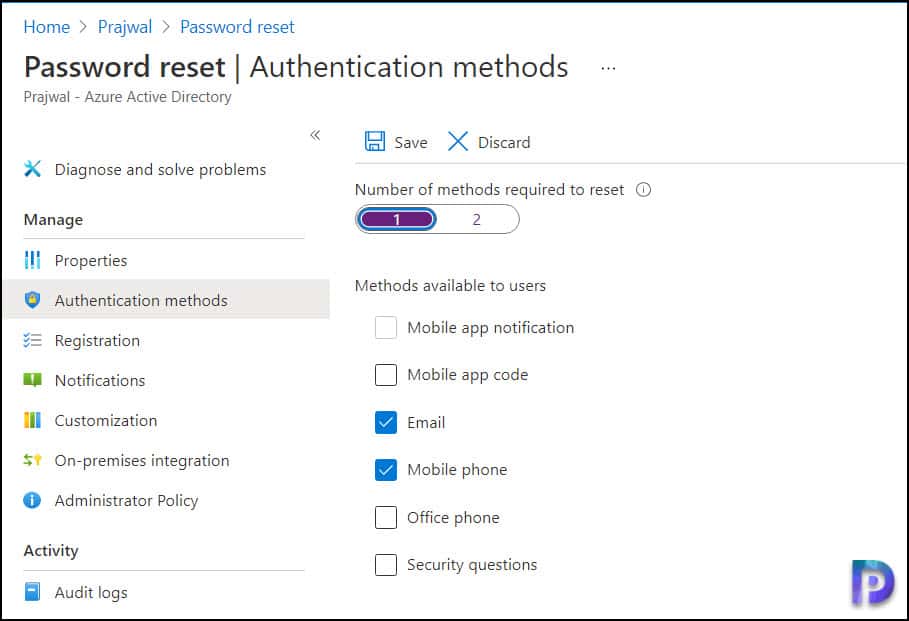
1. Navigate to Azure AD > Password Reset
2. Choose target group (e.g., interns, developers)
3. Define number of authentication methods
4. Customize helpdesk contact info for locked users

**User Experience:**

* Visit https://passwordreset.microsoftonline.com
* Authenticate using MFA
* Reset password securely







**7. Implement and Manage Azure MFA Settings**

**Access MFA Settings**

1. Go to Azure Portal → Microsoft Entra ID
2. Navigate to Security → Multifactor Authentication
3. Click Additional cloud-based MFA settings

Available Settings:

* App Passwords – Used for older apps not supporting modern auth
* Trusted IPs – Skip MFA for known corporate networks
* Verification Options – Choose allowed methods
* Remember Multi-Factor Authentication – Skip MFA for remembered devices (configurable)

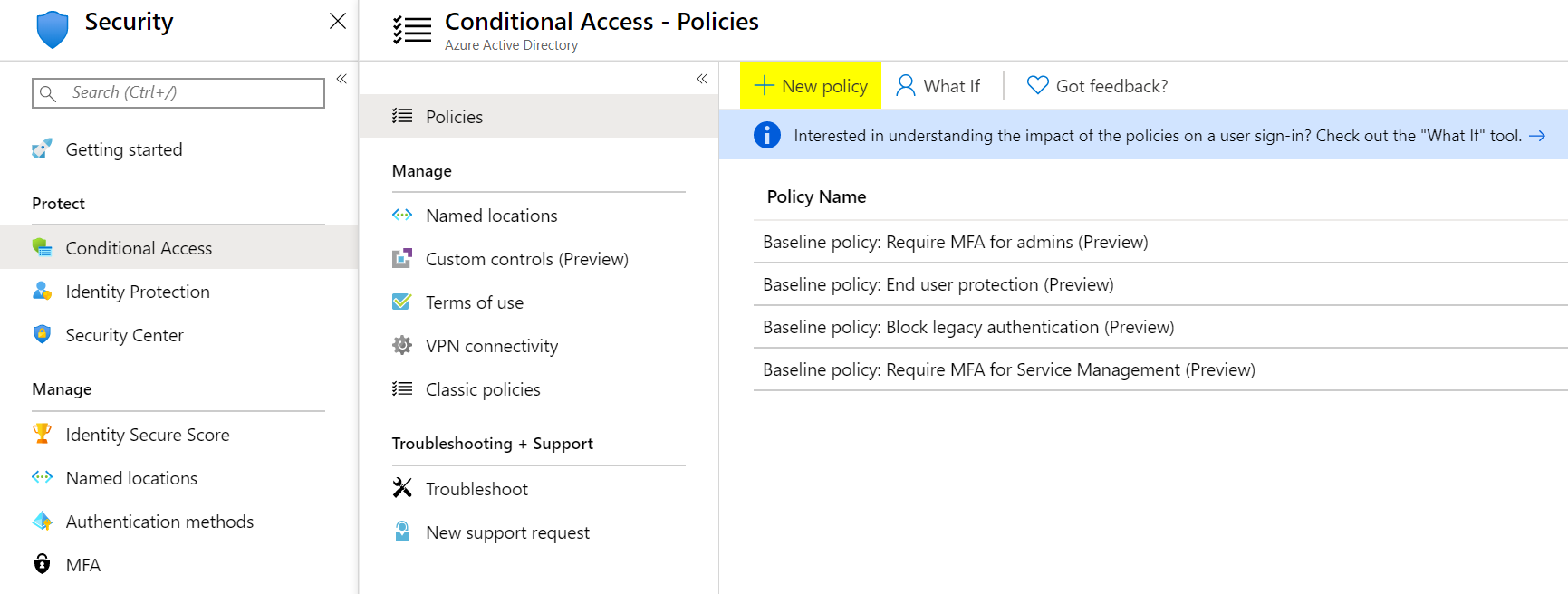
**Configure MFA Registration**

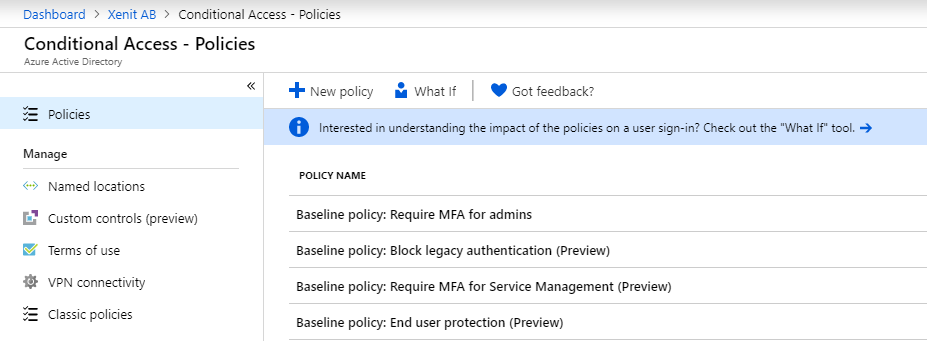
1. Go to Microsoft Entra ID → Security → Authentication Methods
2. Select Microsoft Authenticator
3. Enable for All or specific users
4. Configure registration requirements and method priority

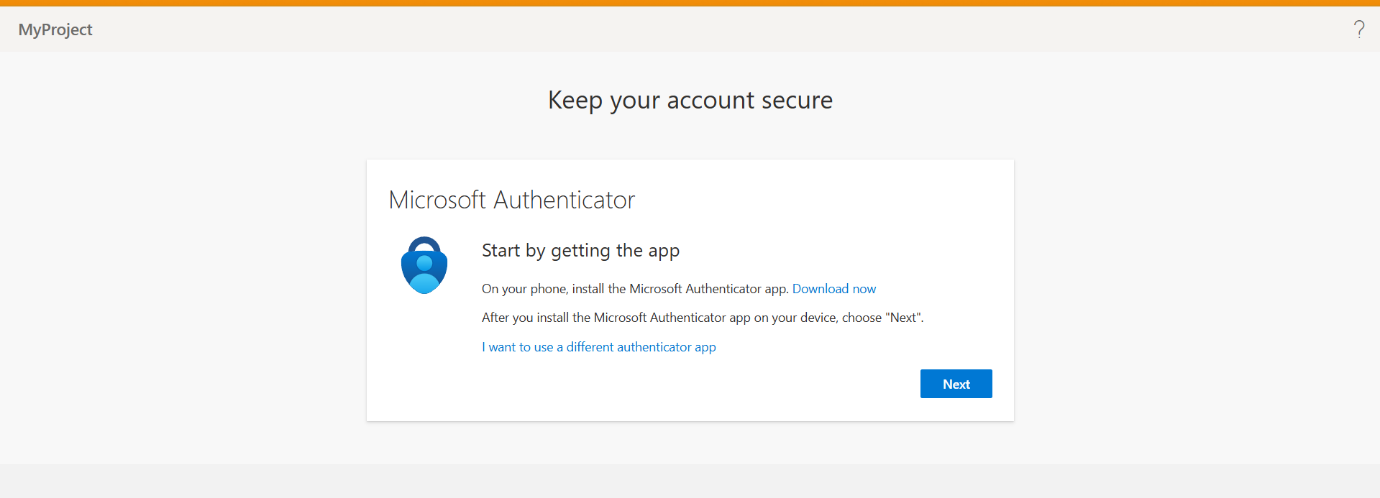
**Enforcing MFA via Conditional Access (Recommended Method)**

1. Go to **Microsoft Entra ID → Security → Conditional Access**
2. Click **+ New Policy**
3. Name: Enforce MFA for All Users
4. **Assignments:**
   * Users: All users (or specific group)
   * Cloud apps: All apps (or select apps)
5. **Access Controls → Grant → Require multi-factor authentication**
6. Enable Policy → **On → Create**

This is the most flexible and enterprise-grade way to enforce MFA.







**Monitoring MFA Usage and Reporting**

* + **View Reports**

1.Go to **Azure Portal → Microsoft Entra ID → Sign-in logs**

2.Filter by **Authentication requirement**

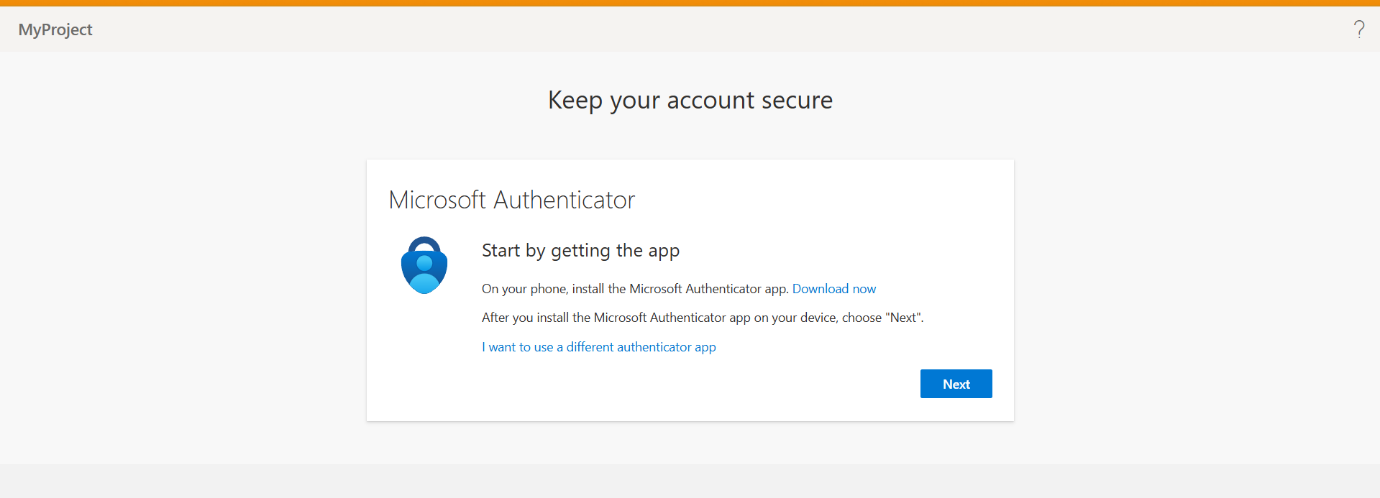
3.View columns like MFA Required, MFA Result

* + **Usage & Insights**

1.Navigate to **Microsoft Entra → Protection → Authentication methods**

2.Click **Usage & Insights**

3.Review registered methods by users



**Account Lockout Protection**

To prevent brute-force attacks:

1. Go to Azure Portal → Microsoft Entra ID → Security → Authentication Methods
2. Choose Microsoft Authenticator or Password Protection
3. Set Lockout threshold, Duration, and Reset interval

Azure intelligently locks accounts after multiple failed attempts.

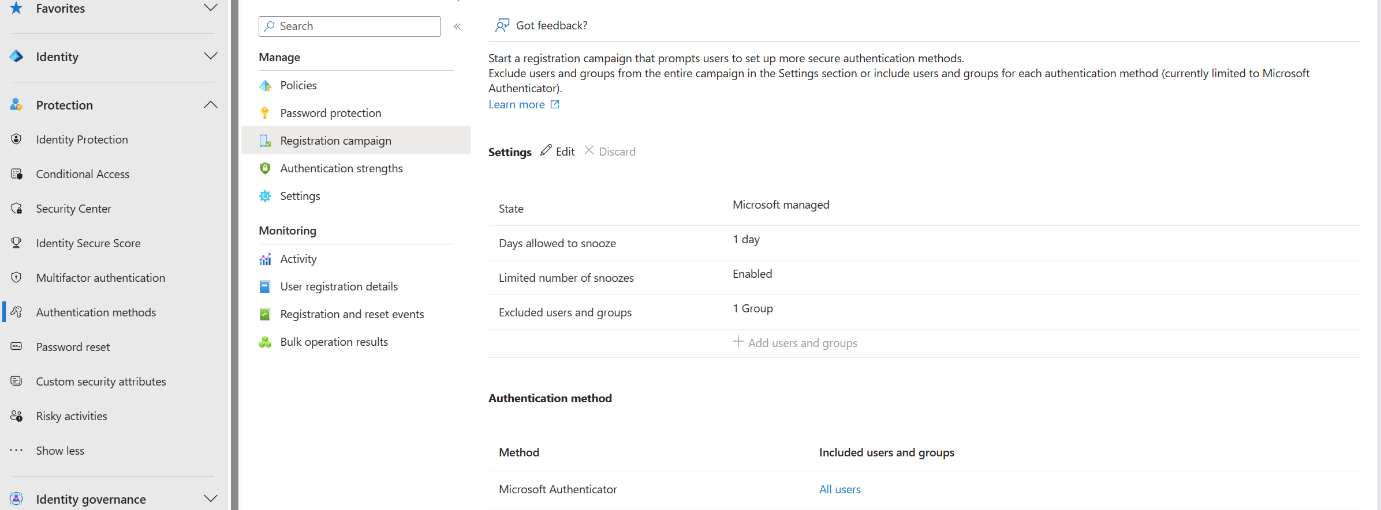
**Extending MFA to On-Premise and 3rd-Party Apps**

* Use Azure AD Application Proxy to enable MFA for on-prem apps.
* Integrate MFA with:
  + VPN devices
  + Remote Desktop Gateway
  + Salesforce, Dropbox, etc. via Enterprise Applications

Go to Azure AD **→** Enterprise applications → New Application → Add Gallery App

**Best Practices for Managing MFA**

* Enforce MFA for all users using Conditional Access
* Use Authenticator App over SMS
* Enable registration campaign to onboard users
* Regularly monitor sign-in logs for MFA failures
* Educate users about phishing-resistant MFA



**8. Account Lockout Settings**

**Purpose:**

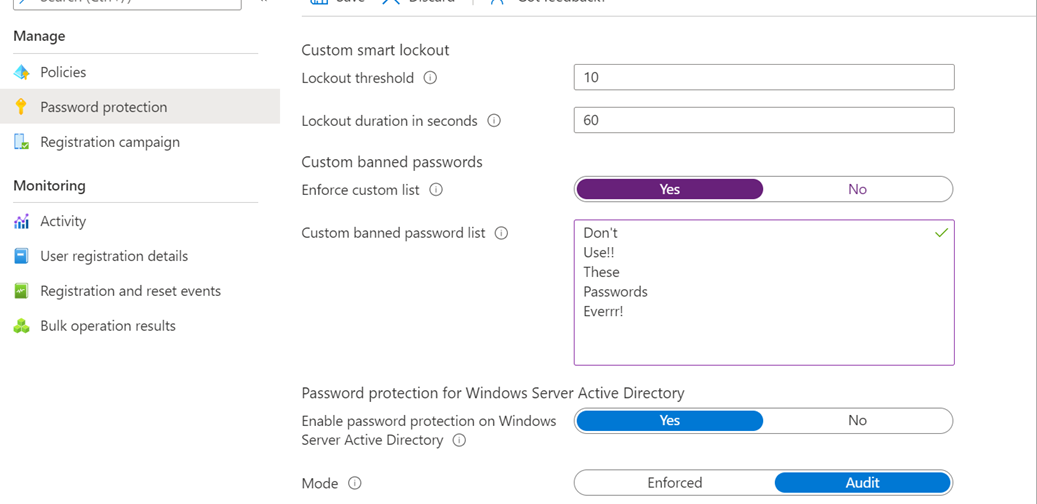
Prevent brute-force MFA attacks

Settings Location:

* Azure AD > Security > Authentication methods > MFA > Account Lockout

Configurable Options:

* Lockout threshold: Attempts before lock (default: 10)
* Lockout duration: Duration in minutes (default: 1)



Implementing and managing Azure MFA is **critical to securing modern cloud environments**. With proper configuration using Conditional Access, registration policies, and reporting, organizations can protect user accounts and comply with security standards.

**References**

* [Microsoft Docs – Azure MFA](https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks)
* [Microsoft Entra ID – Conditional Access](https://learn.microsoft.com/en-us/entra/identity/conditional-access/)
* [Authentication Methods in Azure](https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-authentication-methods)

**9. Manage MFA Settings for Users**

**Ways to Manage:**

1. Per-user MFA configuration
2. User registration campaigns
3. Conditional Access for MFA enforcement

**Tools for Admins:**

* Authentication Methods Policy
* PowerShell & Graph API scripting

**Why Extend Azure MFA?**

* Strengthen authentication for legacy or hybrid apps
* Prevent breaches across all access points
* Meet compliance standards like NIST, HIPAA
* Centralize authentication across cloud and on-prem systems

**Architecture of Extending Azure MFA**

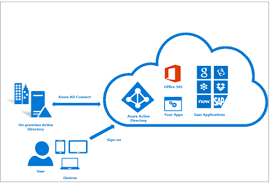
Azure MFA supports various integration methods, primarily through:

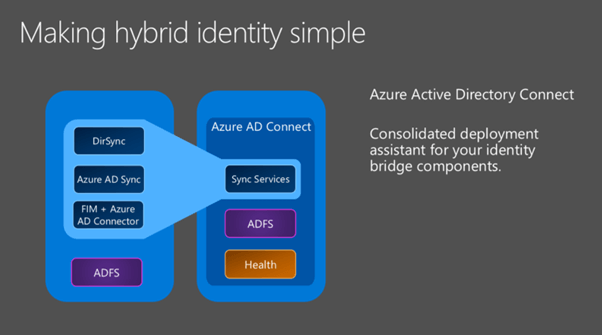
* **NPS Extension (RADIUS)**
* **Azure AD Application Proxy**
* **SAML / OIDC Federation**
* **Conditional Access policies**

## **10. Supported Third-party and On-Prem Systems**

These allow MFA enforcement outside Azure cloud apps

| **System** | **Integration Method** |
| --- | --- |
| Cisco / Fortinet VPN | NPS + RADIUS |
| Remote Desktop Gateway | NPS Extension |
| On-Prem Web Apps (IIS) | Application Proxy |
| Salesforce / Dropbox | SAML Federation |
| Custom Apps | OAuth/OpenID or Legacy Federation |
| Linux Servers (SSH) | PAM Modules or Jump Servers |





On Prem integration with MS AZURE 1

**11. Monitor Azure MFA Activity**

Monitoring Azure MFA activity is essential to ensure users are authenticating securely, to detect any unusual login behavior, and to audit compliance. Microsoft Entra provides various monitoring and reporting capabilities.

**Why Monitor MFA Activity?**

* Detect suspicious login attempts
* Ensure all users are registered for MFA
* Investigate MFA failures or bypasses
* Meet audit and compliance requirements
* Create alerts for high-risk sign-ins

**Monitoring Tools:**

* Azure Sign-in Logs
* MFA Usage Reports
* Conditional Access Insights

**Key Metrics:**

* User sign-in attempts
* Method usage statistics
* Failure and success trends

**Audit User MFA Registration Status**

Steps:

1. Go to **Azure Portal → Microsoft Entra ID → Users**
2. Select **Per-user MFA** or go to:
   * **Authentication Methods → Registration campaign**
3. Or, use:
   * **Microsoft Entra → Protection → Authentication methods → Registration**

* **On activation, require Microsoft Entra Conditional Access authentication context**: You can require users who are eligible for a role to satisfy Conditional Access policy requirements. For example, you can require users to use a specific authentication method enforced through Authentication Strengths, elevate the role from an Intune-compliant device, and comply with terms of use.
* Require approval to activate: You can require approval for activation of an eligible assignment. The approver doesn't have to have any roles. When you use this option, you must select at least one approver.

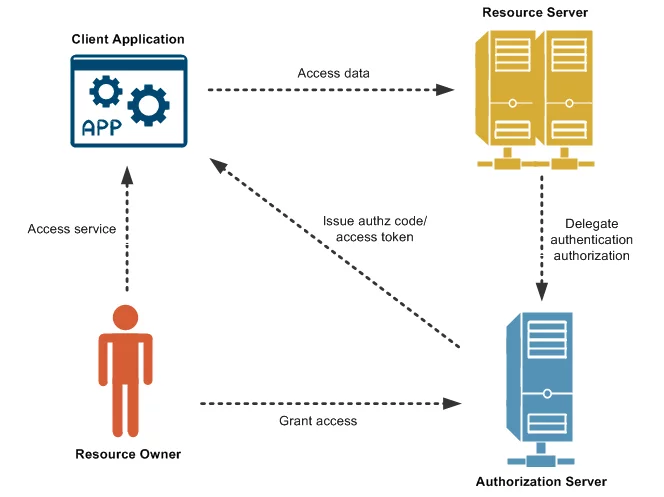
**12. OAuth Tokens**

* **What is OAuth?**

**OAuth (Open Authorization)** is a secure open standard for **authorization** that allows one application to access data/resources on behalf of a user **without revealing their password**.

Example: Logging into a third-party app using your Microsoft or Google account — the app accesses only the authorized parts of your profile.

Tokens issued to users/applications to access services without using passwords repeatedly are called OAuth Tokens.



* **OAuth 2.0 Grant Types**

OAuth 2.0 defines different **authorization flows (grant types)** for various use cases:

| **Grant Type** | **Use Case** |
| --- | --- |
| **Authorization Code** | For web and mobile apps |
| **Client Credentials** | App-to-app communication |
| **Implicit** | For browser-based apps (not recommended now) |
| **Password** | For legacy scenarios (deprecated) |
| **Device Code** | For devices without browsers (TVs, CLI) |

**What are OAuth Tokens?**

In OAuth, **tokens are used to access resources** instead of using a password. When a user grants access, the authorization server issues tokens to the client app.

OAuth tokens are digital keys that grant third-party applications limited access to user resources on another service, without needing the user's actual username and password. These tokens act as secure intermediaries, allowing applications to interact with specific resources on behalf of a user. They are a core part of modern identity management and authorization processes.

**What they are:**

* **Delegated Authorization:**

OAuth tokens enable a user to authorize a third-party application to access their data on another service (like Google or Facebook) without directly sharing their credentials with that application.

* **Secure Access:**

Instead of giving the application your password, you grant it a token that allows it to access only the resources you've specified (e.g., reading your email, posting to your wall).

* **Different Types:**

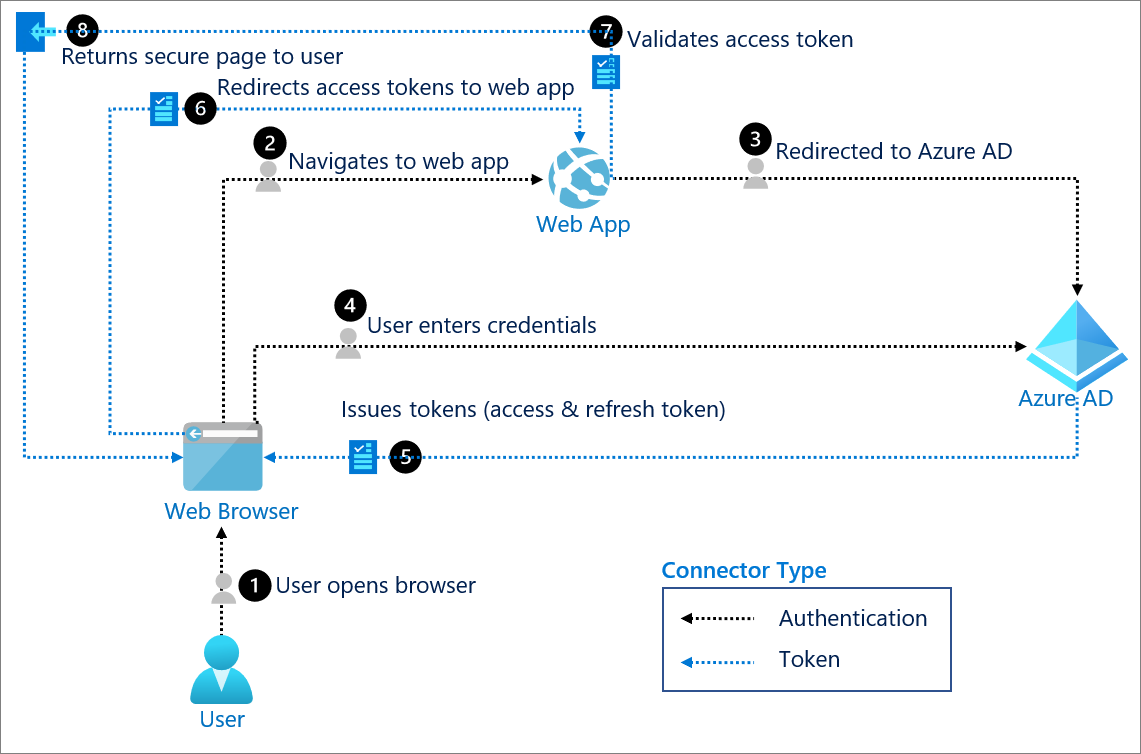
OAuth 2.0 commonly uses access tokens, refresh tokens, and ID tokens. Access tokens are used for immediate access, refresh tokens can be used to get new access tokens, and ID tokens are used to provide information about the user's authentication.

* **Varying Formats:**

Access tokens can come in various formats, including JSON Web Tokens (JWTs).

* **Scope of Access:**

Tokens are associated with specific scopes, defining the level of access the application has to the user's resources.



**How they work (Simplified):**

1. **Authorization Request:**

A user initiates an action that requires access to a third-party service (e.g., logging in to an app with Google).

1. **Redirection:**

The user is redirected to the service's authorization server (e.g., Google's login page).

1. **Authentication:**

The user authenticates with the service.

1. **Authorization Grant:**

The user grants permission for the application to access specific resources.

1. **Token Issuance:**

The service issues an access token (and potentially a refresh token) to the application.

1. **Resource Access:**

The application uses the access token to access the user's resources on the service.

1. **Token Refresh:**

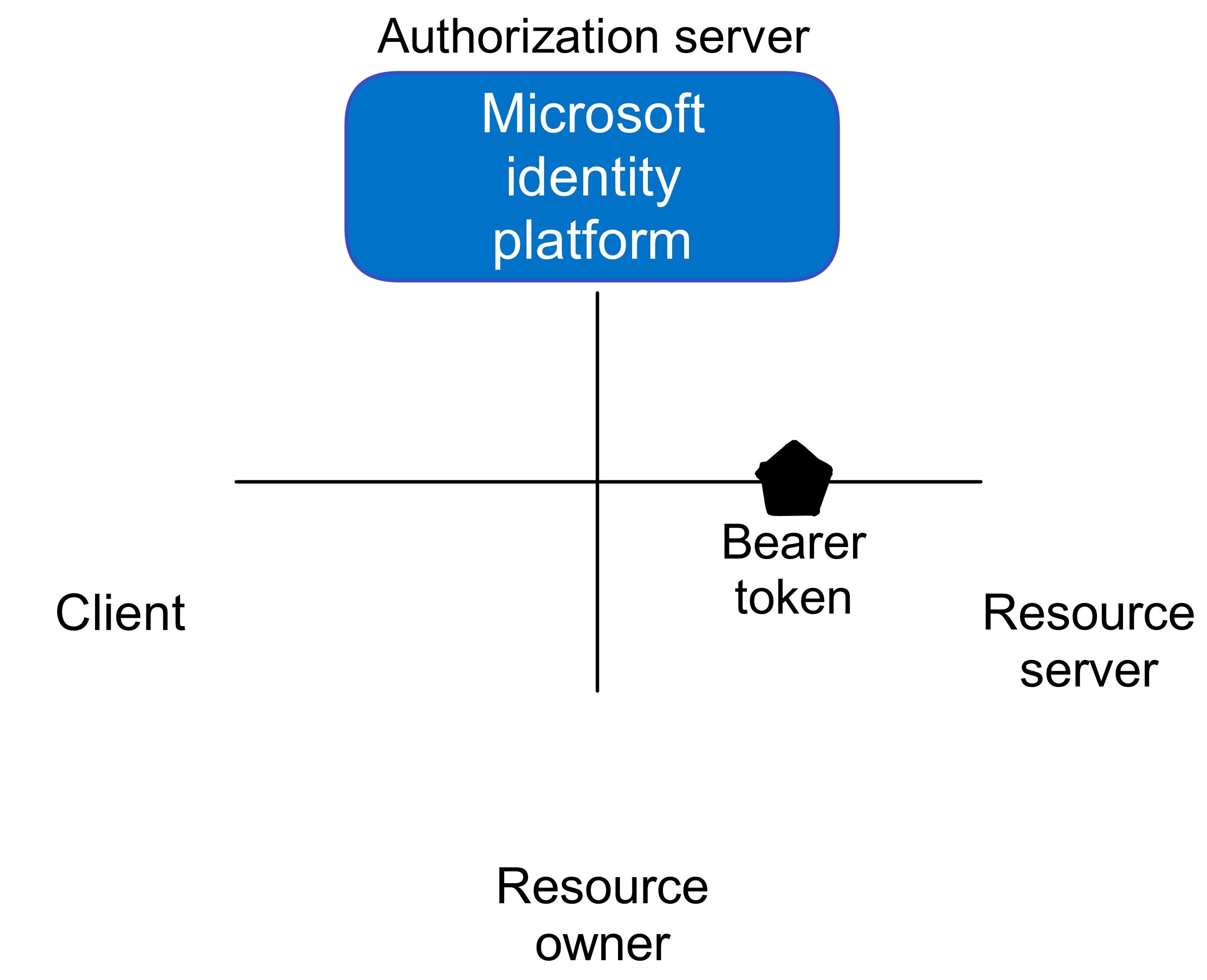
If the access token expires, the application can use the refresh token to get a new access token.

**Example:**

You might use an app to manage your social media. Instead of giving the app your Facebook password, you authorize it using OAuth. The app receives an access token, allowing it to post updates to your Facebook timeline, but only those you've authorized.

**Key Benefits:**

* **Enhanced Security:** Users don't have to share their passwords with third-party applications.
* **Granular Control:** Users can control the specific resources and actions that applications can access.
* **Simplified User Experience:** OAuth simplifies the login process, making it easier to use third-party applications.

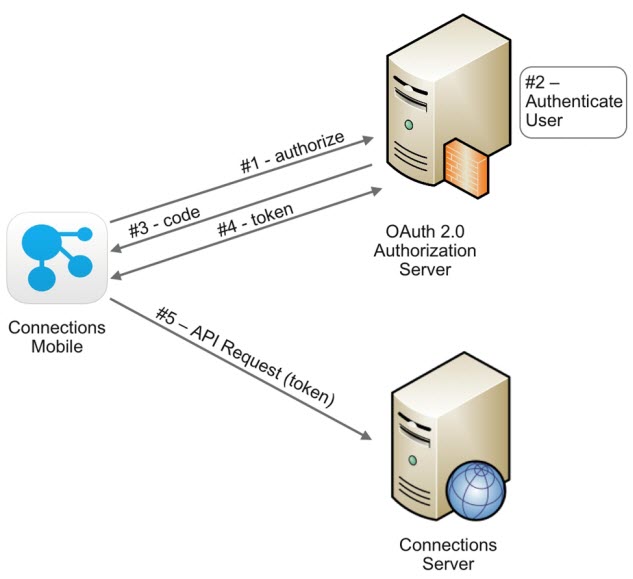


**Types of OAuth Tokens**

| **Token Type** | **Purpose** |
| --- | --- |
| **Access Token** | Grants access to protected resources |
| **Refresh Token** | Used to obtain a new access token after expiration |
| **ID Token** | (OpenID Connect only) Contains user identity information in JWT format |

***How OAuth Works (Flow Diagram)***

**

****

**OAuth Tokens in Microsoft Azure (Microsoft Entra ID)**

In Azure, OAuth is implemented via Microsoft Identity Platform (Microsoft Entra ID).Why Use OAuth in Azure?

* Secure API Access: Applications authenticate using tokens, not credentials.
* Supports SSO: Enables seamless access across cloud apps.
* Works with MFA: Integrates with Conditional Access and MFA for security.
* Token-Based Authorization: Improves scalability and stateless communication.

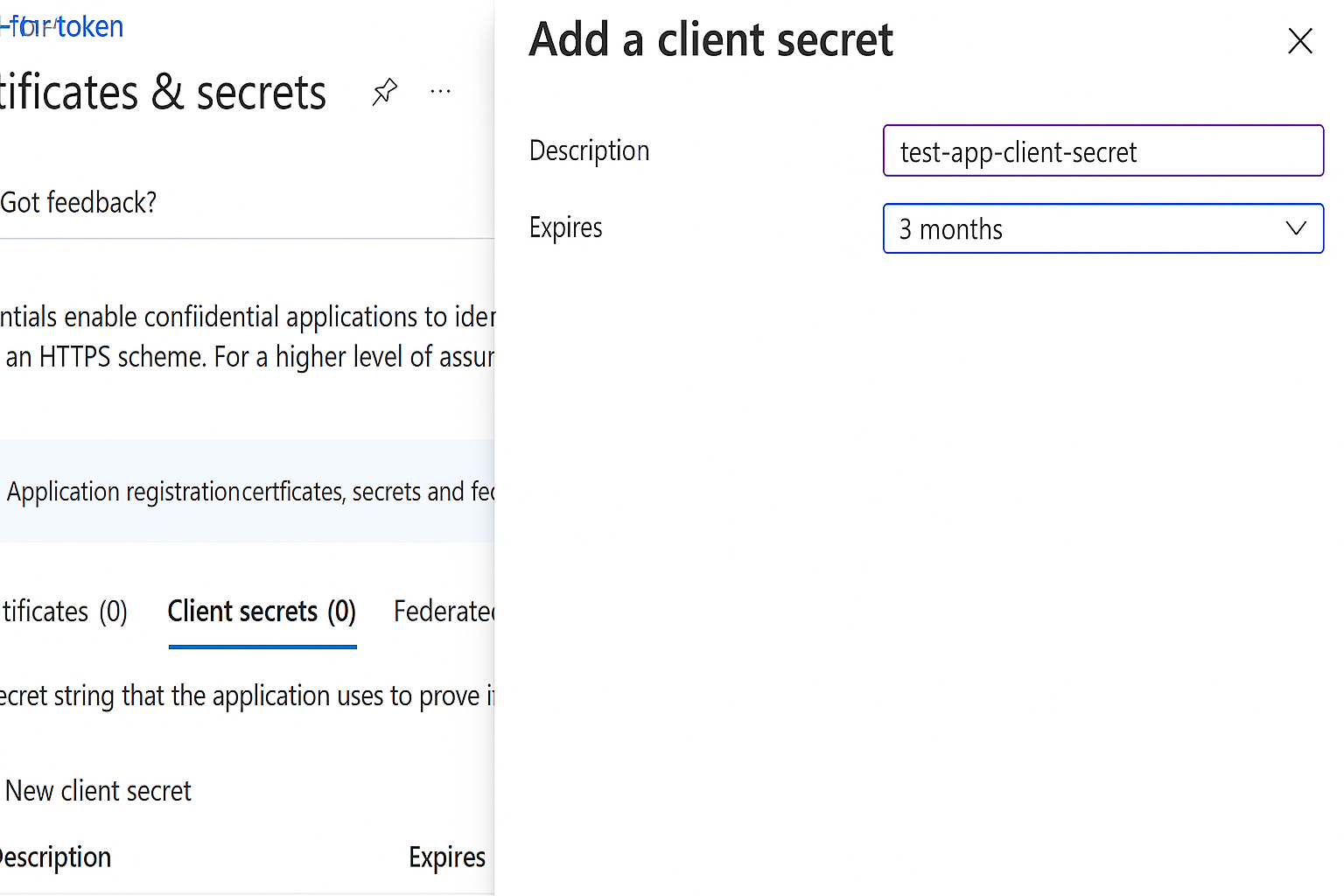
**Key Components:**

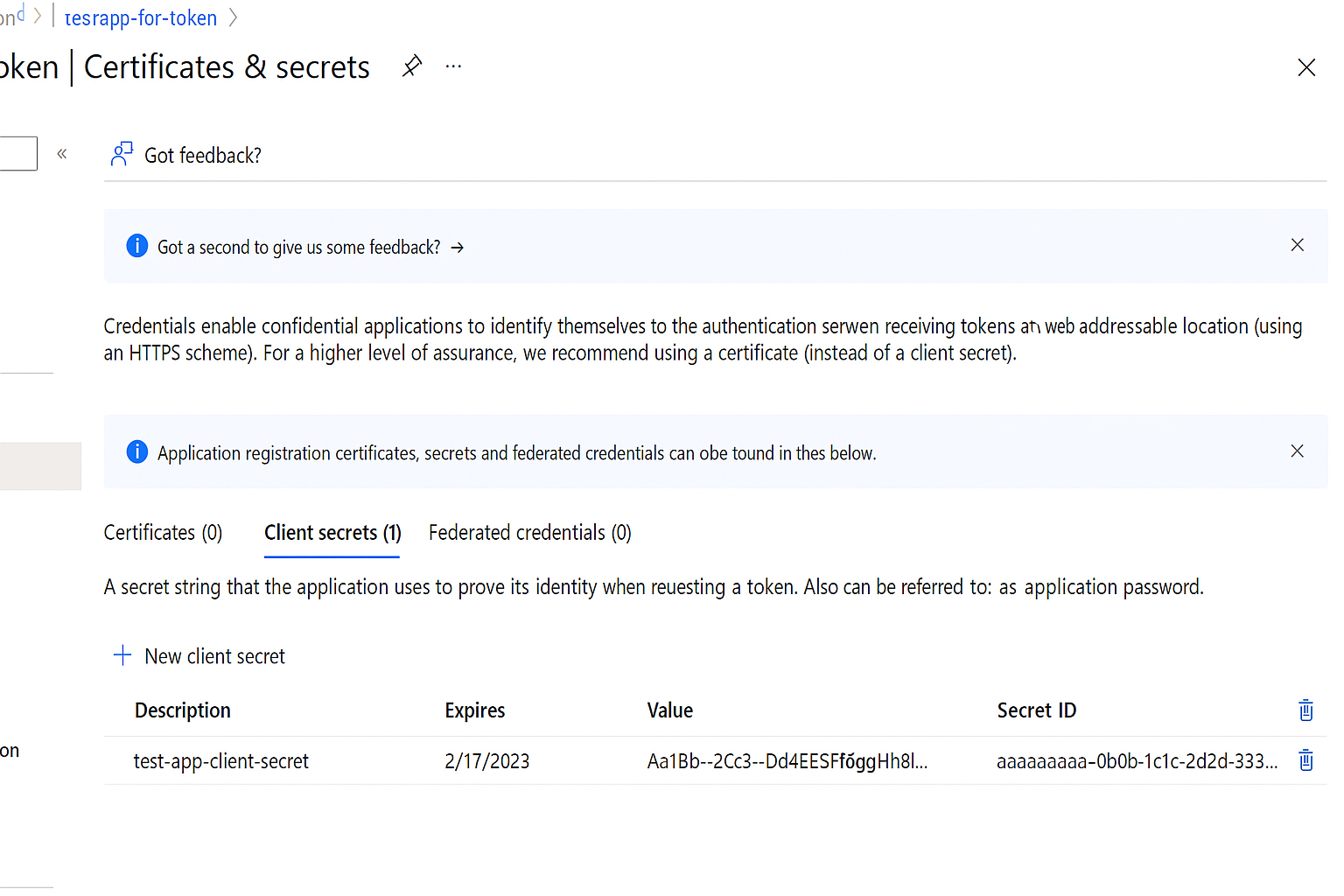
| Component | Description |
| --- | --- |
| Authorization Server | https://login.microsoftonline.com/{tenant}/oauth2/v2.0 |
| Resource Server | e.g., Microsoft Graph API |
| Client (App) | Your registered app in Azure |
| User | Person who is granting permission |
|  |  |

Azure issues OAuth 2.0 tokens when your application authenticates users or

itself to access Azure services like:

* Microsoft Graph
* Azure REST APIs
* Custom web APIs

****

****

**Security Best Practices for OAuth Tokens**

* Never expose tokens in browser URLs
* Store tokens securely (Key Vault, environment variables)
* Use short-lived access tokens
* Implement token revocation if compromised
* Always use HTTPS
* Enable Conditional Access policies in Azure
* **Use Cases in Azure**

| **Use Case** | **Token Required** | **Example** |
| --- | --- | --- |
| Access Graph API | Access Token | Get user profile |
| Automate Azure via REST | Client Credentials Token | Script deployments |
| Single Sign-On (SSO) | ID Token | Log in to 3rd-party |
| Refresh expired access | Refresh Token | Silent login |

* **Conclusion**

OAuth tokens play a vital role in secure authentication and authorization in Azure. Azure issues JWT-based tokens to apps that allow access to protected APIs, including Microsoft Graph and Azure services. With proper handling, OAuth tokens offer scalable, secure, and password-less access to modern applications.

**References**

**Azure Multifactor Authentication (MFA)**

1. Microsoft Docs – What is Azure AD Multi-Factor Authentication
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks>
2. Configure Azure AD Multi-Factor Authentication Settings
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-mfasettings>
3. Enable per-user MFA in Azure AD
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-userstates>
4. Conditional Access and MFA Integration
   * <https://learn.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-policies>

**Self-Service Password Reset (SSPR)**

1. Microsoft Docs – Self-service password reset overview
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-sspr-howitworks>
2. Set up self-service password reset in Azure AD
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/tutorial-sspr>

**Authentication Methods, Account Lockout, and Policies**

1. Authentication Methods Policy in Microsoft Entra ID
   * <https://learn.microsoft.com/en-us/entra/id-authentication/concept-authentication-methods>
2. Configure account lockout settings for Azure MFA
   * <https://learn.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-mfasettings#account-lockout>

**OAuth and Tokens**

1. Microsoft Identity Platform – OAuth 2.0 Authorization Code Flow
   * <https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow>
2. Access Tokens and Identity Tokens – Microsoft Identity Platform
   * <https://learn.microsoft.com/en-us/azure/active-directory/develop/access-tokens>

**Azure Portal & Admin Center**

1. Microsoft Entra Admin Center (Azure AD)
   * <https://entra.microsoft.com/>
2. Azure Portal
   * <https://portal.azure.com>