SC-300: Microsoft Identity and Access Administrator Study Notes

Author & Creator: Niraj Kumar
LinkedIn: Niraj Kumar

This comprehensive guide is designed to assist you in mastering the concepts required for the **Microsoft Certified: Identity and Access Administrator Associate (SC-300)** exam. It covers the implementation of identity management solutions, authentication, access control, and identity governance using Microsoft Entra ID.

Module 1: Identity Management & Configuration

This module establishes the core foundation of Microsoft Entra ID. It covers the essential tasks of configuring your tenant, managing the lifecycle of users and groups, securing device access, and implementing role-based access control (RBAC) to delegate administrative permissions effectively.

1.1 Tenant Setup

This section focuses on the initial configuration and management of your Microsoft Entra ID tenant. You will learn how to create a new tenant, manage its properties, configure custom domains, and customize the user experience with company branding. These foundational steps are critical for establishing a secure and professional identity environment.

1.1.1 Introduction to Microsoft Entra ID (formerly Azure Active Directory)

- **Management Interfaces:**
  - **Azure Portal (`portal.azure.com`):** The primary visual interface for managing users, groups, applications, and security.
  - **Automation:** Tasks can also be performed via PowerShell, Azure CLI, ARM Templates, or Bicep.
- **Account Structure:**
  - **User:** The identity used to log in. Every user must belong to at least one Tenant.
  - **Tenant:** A dedicated instance of Entra ID representing an organization.
    - **Default Domain:** Assigned by Microsoft upon creation (e.g., `username.onmicrosoft.com`).
    - **Limits:** A single user can create up to 200 tenants and belong to up to 500.
    - **Cost:** Creating a tenant is free.
  - **Subscription:** The billing entity required to create Azure resources (like VMs). A tenant can exist without a subscription (purely for identity management).

1.1.2 Create a New Entra ID Tenant

- **Purpose:**
  - Create a dedicated environment where you are the **Global Administrator**.
  - Useful for testing, development, or separating organizational data.

[linkendin.com/in/nirajkum]

- **Navigation:**
  - Go to **Microsoft Entra ID** in the Azure Portal.
  - Select **Manage Tenants** (often found in the top menu or overview blade).
  - Click **Create**.
- **Configuration Steps:**
  1. **Tenant Type:** Choose **Microsoft Entra ID** (Standard).
  2. **Configuration:**
     - **Organization Name:** The display name for the tenant (e.g., "Contoso Dev").
     - **Initial Domain Name:** Must be globally unique. Format: `yourname.onmicrosoft.com`.
     - **Country/Region:** Determines **Data Residency** (where data is physically stored). This **cannot be changed** after creation.
  3. **Review + Create:** Complete the Captcha verification and wait for provisioning.

1.1.3 Switch Tenants & Tenant Overview

- **Tenant Overview Blade:**
  - **Tenant ID:** A unique GUID required for scripting (PowerShell/CLI) and configuration.
  - **License Information:** Displays current license level (e.g., Azure AD Free, Premium P1/P2).
- **The Global Administrator Role:**
  - **Assignment:** Automatically assigned to the user who creates the tenant.
  - **Scope:** Highest level of access; full control over all identity resources (Users, Groups, Roles).
  - **Requirement:** Every tenant must have at least one Global Administrator.
- **Switching Contexts:**
  - **Why Switch?** To manage different environments (e.g., Corporate vs. Dev/Test) where your permissions differ.
  - **Methods:**
    1. **Settings Menu:** Click the **Gear icon** > **Directories + subscriptions** > **Switch** next to the desired tenant.
    2. **Profile Menu:** Click your **Avatar** (top right) > **Switch directory** > Select from Favorites or All Directories.

[linkendin.com/in/nirajkum]

1.1.4 Set a Custom Domain

- **Why use a Custom Domain?**
  - **Professionalism:** Allows users to sign in with their corporate email (e.g., `user@example.com`) instead of the default `user@tenant.onmicrosoft.com`.
  - **User Experience:** Simplifies login credentials to match existing email addresses.
- **Configuration Steps:**
  1. **Add Domain:**
     - Navigate to **Custom domain names** in the Entra ID blade.
     - Click **Add custom domain** and enter your domain name (e.g., `getcloudskills.com`).
  2. **Verify Ownership:**
     - Entra ID provides a **TXT** or **MX** record.
     - Add this record to your **Domain Registrar's DNS settings** (e.g., GoDaddy, Namecheap).
     - **Propagation:** DNS changes can take time (minutes to hours).
     - Click **Verify** in the Azure Portal.
  3. **Make Primary (Optional):**

        - Select the verified domain and click **Make primary**.
        - This sets it as the default suffix for new users.
- **Note:** You must have access to the DNS records of the domain to complete verification.

1.1.5 Manage Azure AD Company Branding & Tenant Properties

- **Company Branding:**
    - **Purpose:** Enhances user trust by replacing default Microsoft styling with organization-specific visuals on sign-in pages and the My Account portal.
    - **Key Settings:**
        - **Visuals:** Background image, Banner logo, Custom color.
        - **Text:** Sign-in page text, Username hint.
        - **Localization:** Ability to configure branding based on user locale.
- **User Settings:**
    - **App Registrations:** Toggle to allow or block users from registering non-admin applications.
    - **Portal Access:** "Restrict access to Azure AD administration portal" prevents non-admins from accessing the Entra ID administrative interface.
- **Tenant Properties:**
    - **Display Name:** Can be modified (e.g., renaming a dev tenant).
    - **Region/Country:** Immutable once the tenant is created.
    - **Global Admin Access:** Toggle to grant the current user management access to all Azure subscriptions (Access management for Azure resources).
    - **Metadata:** Privacy statement and contact information.

[linkedin.com/in/nirajkum]

1.2 Users, Groups, and Devices

This section covers the core identity objects within Microsoft Entra ID. You will learn how to manage the lifecycle of user identities, organize them into groups for efficient access management, and register devices to ensure secure access to organizational resources. Additionally, it touches on administrative units for delegated administration and license assignment.

1.2.1 User Management

- **Core Concepts:**
    - **Authentication:** The process of verifying identity (User ID + Password + MFA). Once validated, Azure AD issues a token to the application.
    - **Authorization:** The process of determining what permissions the user has after they are authenticated.
- **Creating a New User:**
    - **Navigation:** Go to **Users** > **New user** > **Create new user**.
    - **Mandatory Fields:**
        - **User name (UPN):** The login ID (e.g., `user@domain.com`). You can choose the default `onmicrosoft.com` domain or a verified custom domain.
        - **Name:** The display name (e.g., "Test User Two").
    - **Optional Settings:**
        - **Password:** Choose to auto-generate or manually set a password.
        - **Groups/Roles:** Can be assigned during creation (though often done later).

        - **Settings:** Block sign in (Enable/Disable account), Usage location (Important for licensing).
        - **Job Info:** Job title, Department, Manager.
- **Default Permissions:**
    - A newly created user has **no access rights** to Azure resources or applications by default. They must be explicitly granted access via roles or groups.
- **User Management & Lifecycle:**
    - **Source of Authority:** Users can be **Cloud-only** (created in Azure Portal) or **Synced** (from on-premises AD via Azure AD Connect).
    - **Administrative Actions:**
        - **Reset Password:** Allows admins to reset credentials if a user is locked out.
        - **Revoke Sessions:** Invalidates existing tokens, forcing the user to sign in again (useful for compromised accounts).
        - **Edit Properties:** Update biographical data, contact info, and job details.
- **Limits:** There are limits to the number of directory objects (users, groups) a tenant can hold, though these limits are high.

1.2.2 Group Management

- **Purpose:** Groups allow administrators to manage users collectively rather than individually. They are used for assigning permissions to resources or applications in bulk.
- **Group Types:**
    - **Security:** Used for managing access to resources (apps, data). Can contain users, devices, groups, and service principals.
    - **Microsoft 365:** Used for collaboration (Teams, SharePoint, Outlook). Includes a shared mailbox, calendar, etc.
- **Creating a Group:**
    - **Azure AD Roles Assignment:**
        - When creating a group, you can enable **"Azure AD roles can be assigned to the group"**.
        - **Important:** Once set to **Yes**, this cannot be changed back to No. This allows the group to hold admin roles (e.g., Helpdesk Administrator), simplifying role management.
- **Group Roles:**
    - **Owners:** Users who can manage the group settings and membership. Owners do not have to be members of the group (i.e., they don't necessarily inherit the group's access rights).
    - **Members:** The users (or devices) that belong to the group and inherit permissions assigned to it.
- **Membership Types:**
    - **Assigned:**
        - Administrators manually select and add specific users to the group.
        - Membership only changes when an admin manually adds or removes a user.
    - **Dynamic User:**
        - Membership is determined by a **Dynamic Query** based on user properties (e.g., `user.department -eq "Sales"` or `user.userPrincipalName -contains "student"`).
        - **Automation:** When a user's attributes change to match the rule, they are automatically added. If they no longer match, they are removed.

1.2.3 Device Management

- **Overview:**

[linkendin.com/in/nirajkum]

  - Device management enables organizations to secure and manage devices (Windows, macOS, iOS, Android, Linux) accessing corporate resources.
  - It is a prerequisite for implementing **Conditional Access** policies based on device compliance.
- **Device Identity Types:**
  - **Azure AD Registered:**
    - **Scenario:** Bring Your Own Device (BYOD).
    - **Description:** Personal devices where the user signs in with a personal account but adds a Work/School account to access apps (e.g., Teams, Outlook).
    - **Authentication:** User authenticates to the app, not the OS.
  - **Azure AD Joined:**
    - **Scenario:** Corporate-owned devices.
    - **Description:** The device is owned by the organization. The user signs in to the OS using their Entra ID credentials.
    - **Benefit:** Full control over the device, SSO to cloud and on-premises resources.
- **Registration Process (Windows Example):**
  1. Navigate to **Settings** > **Accounts** > **Access work or school**.
  2. Click **Connect**.
  3. Enter the corporate email (UPN) and password.
  4. The device is registered, and policies are applied.
  5. **Verification:** In Azure Portal > **Devices**, the device appears with the status "Azure AD Registered".
- **Device Settings:**
  - Located under **Devices** > **Device settings** in the portal.
  - **Users may join devices to Azure AD:** Controls who can perform Azure AD Join.
  - **Require Multi-Factor Authentication (MFA) to join devices:** Enhances security during the join process.
  - **Maximum number of devices per user:** Limits the number of devices a user can register (e.g., 20 or 50).

1.2.4 Administrative Units

- **Concept:**
  - An **Administrative Unit (AU)** is a container used to logically group Azure AD resources (users, groups, devices) for the purpose of delegating administrative permissions.
  - It allows you to restrict the scope of an administrative role to a specific subset of the organization.
- **Use Case:**
  - **Regional Administration:** Useful for organizations with independent divisions or geographic regions (e.g., "Central Region", "Northeast Region").
  - **Delegation:** Instead of giving a Help Desk Administrator global access to reset passwords for *everyone* in the tenant, you can restrict them to reset passwords only for users in the "Central Region" AU.
- **Configuration Steps:**
  1. **Create Administrative Unit:**
     - Navigate to **Administrative units** > **Add**.
     - Provide a **Name** (e.g., "Central Unit") and **Description**.
  2. **Add Members:**

[linkendin.com/in/nirajkum]

        - Open the created AU.
        - Select **Members** > **Add member**.
        - Select the specific users, groups, or devices that belong to this logical unit.
    3. **Assign Roles (Scoped Administration):**
        - Select **Roles and administrators** within the AU blade.
        - Choose a role (e.g., **Helpdesk Administrator**, **User Administrator**, **Password Administrator**).
        - Assign a user (e.g., a regional IT staff member) to this role.
        - **Result:** This admin can now perform their role's actions *only* on the members of this specific Administrative Unit.
- **Licensing Requirement:**
    - Assigning roles at the scope of an administrative unit requires an **Azure AD Premium P1 or P2** license for the administrator being assigned the role.

1.2.5 Assign Azure AD Premium Licenses to Users

- **Overview:**
    - Azure AD (Entra ID) operates on a tiered licensing model (Free, Premium P1, Premium P2).
    - Upgrading to Premium tiers unlocks advanced security and governance features.
- **License Tiers & Features:**
    - **Azure AD Free:**
        - Basic identity management.
        - **MFA:** Limited to Mobile App authenticator. (Global Admins get SMS/Call).
        - **Object Limit:** 500,000 directory objects.
    - **Premium P2:**
        - **Advanced MFA:** SMS/Phone call for all users, fraud alerts, MFA reports, custom greetings.
        - **Identity Governance:** Privileged Identity Management (PIM), Access Reviews.
        - **Self-Service Password Reset (SSPR):** With on-premises writeback.
        - **No object limit.**
- **Managing Licenses:**
    - **Navigation:** Go to **Billing** > **Licenses** > **All products**.
    - **View Availability:** Shows total licenses purchased vs. assigned.
- **Assignment Methods:**
    1. **Direct Assignment:** Manually assign a license to an individual user profile.
    2. **Group-Based Licensing:**
        - Assign a license to a security group (e.g., "Teachers").
        - **Benefit:** All current and future members of the group automatically inherit the license. Removing a user from the group removes the license.
        - **Efficiency:** Greatly simplifies license management for large organizations.
- **Strategic Licensing:**
    - **Selective Assignment:** You do not need to license every user in the tenant.
    - **Cost Optimization:** Assign Premium licenses only to users who require specific features (e.g., Administrators needing PIM, or specific departments needing SSPR), while keeping others on the Free plan.

[linkendin.com/in/nirajkum]

1.3 Roles and Administration

This section focuses on the Role-Based Access Control (RBAC) model within Microsoft Entra ID. You will learn how to assign built-in administrative roles to users to delegate tasks effectively while adhering to the principle of least privilege. Additionally, it covers the creation of custom roles to meet specific organizational requirements that built-in roles may not cover.

1.3.1 Assign Admin Roles

- **Azure AD Roles vs. Azure Resource Roles:**
  - **Azure AD Roles:** Purely administrative roles used to grant access to manage Azure AD resources (users, groups, apps) and other Microsoft services (Office 365, Intune, Dynamics).
  - **Azure Resource Roles (RBAC):** Used to manage Azure resources like Virtual Machines, Web Apps, and Storage Accounts.
- **Built-in Roles:**
  - Azure AD comes with over 75 built-in roles to cover various administrative tasks.
- **Key Roles:**
  - **Global Administrator:**
    - The highest level of access (Company Administrator).
    - Has permissions to manage all aspects of Azure AD and Microsoft services.
    - Can read, write, create, delete, and update all resources.
  - **Granular Roles:** Specific roles designed for limited tasks.
    - **Guest Inviter:** Can only invite external users (guests).
    - **Password Administrator:** Can reset passwords for non-admins.
    - **License Administrator:** Can manage product licenses.
    - **Intune Administrator:** Manages device configuration and compliance.
- **Security Best Practice: Principle of Least Privilege:**
  - Do not assign Global Administrator to everyone in the IT team.
  - Assign users the specific role(s) required to perform their job functions.
  - **Benefit:** Reduces the attack surface (if an account is compromised) and prevents accidental changes to critical configurations.
  - Users can be assigned **multiple roles** if their job requires permissions from different areas (e.g., Directory Writers + Exchange Administrator).

1.3.2 Define Custom Roles

- **Prerequisites:**
  - Creating custom roles requires an **Azure AD Premium P1 or P2** license.
  - If you are on the Free tier, the "New custom role" button will be grayed out.
- **Why use Custom Roles?**
  - While there are over 70 built-in roles, they may be too broad for specific security requirements.
  - **Principle of Least Privilege:** Custom roles allow you to grant *only* the specific permissions needed for a job function, without including dangerous permissions (like Delete).
- **Strategy for Creation:**
  - Start by analyzing a built-in role that is similar to what you need (e.g., Group Administrator).
  - Identify which permissions to keep and which to remove (e.g., Keep "Update", Remove "Create" and "Delete").
- **Example Scenario: "Group Updater" Role**

[linkedin.com/in/nirajkum]

  - **Goal:** Create a role that can update group properties and membership but **cannot create or delete groups**.
  - **Steps:**
    1. Navigate to **Roles and administrators**.
    2. Select **New custom role**.
    3. **Basics:** Enter a name (e.g., "Group Updater") and description.
    4. **Permissions:**
       - Search for the resource (e.g., "groups").
       - Select specific permissions from the list (e.g., `microsoft.directory/groups/update`, `microsoft.directory/groups/members/update`).
       - Ensure "Create" and "Delete" permissions are **not** selected.
    5. **Review + Create:** Finalize the role.
  - **Result:** Users assigned to this role can manage group day-to-day operations without the risk of accidental deletion or unauthorized creation of new groups.
- **Cloning:** You cannot clone a built-in role directly to create a custom role; you must start from scratch or clone an existing custom role.

Module 2: External Identities & Hybrid Identity

This module explores how to securely collaborate with external users (partners, customers) using Azure AD B2B and B2C. It also covers the implementation of hybrid identity solutions, synchronizing on-premises Active Directory with Microsoft Entra ID using Azure AD Connect to provide a seamless single sign-on experience.

2.1 External Collaboration (B2B & B2C)

This section details the mechanisms for enabling secure collaboration with external parties. It covers Azure AD Business-to-Business (B2B) for partner collaboration and Azure AD Business-to-Consumer (B2C) for customer-facing identity management, including configuration settings and user lifecycle management.

2.1.1 External Collaboration Settings

[linkedin.com/in/nirajkum]

- **Overview of External Identities:**
  - **B2B (Business-to-Business):** Designed for collaborating with partners and vendors. These users are invited as **Guest Users** into the directory. They can access resources like Teams, SharePoint, and enterprise apps.
  - **B2C (Business-to-Consumer):** Designed for customer-facing applications. Users sign in with social identities (Google, Facebook) or local accounts to access custom applications.
- **External Collaboration Settings:**
  - Located under **External Identities** > **External collaboration settings**. These settings control how guest users interact with the tenant.
- **1. Guest User Access:**
  - Determines the level of visibility guest users have regarding other users and groups in the directory.
  - **Same as members:** Guests have full read access to all directory data (not recommended).
  - **Limited access:** Guests can see their own profile but have limited visibility into other users (cannot see group memberships of others).

  - **Restricted access:** Guests can only see their own profile and absolutely nothing else in the directory.
- **2. Guest Invite Settings:**
  - Controls who is authorized to invite external users.
  - **Anyone in the organization:** Includes Member users and even Guest users (if enabled).
  - **Member users and admins:** Standard users can invite guests.
  - **Only users with specific admin roles:** Restricts invitations to roles like **Guest Inviter** or **Global Administrator**.
  - **No one:** Disables the ability to invite guests entirely.
- **3. Collaboration Restrictions (Domain Restrictions):**
  - Controls which domains are permitted or blocked for invitations.
  - **Allow invitations to any domain:** (Default) No restrictions.
  - **Deny invitations to specified domains:** Block specific domains (e.g., competitors or public email providers like `gmail.com`).
  - **Allow invitations only to specified domains:** Whitelist approach. Invitations can *only* be sent to the listed domains.

2.1.2 Invite External Users

- **Concept:**
  - Inviting external users (B2B) allows them to access your organization's resources using their existing credentials (BYOID - Bring Your Own Identity).
  - They are added as **Guest** users in the directory.
- **Step-by-Step: Inviting a User:**
  1. **Navigation:** Go to **Users** > **New user** > **Invite external user**.
  2. **Basics:**
     - **Email:** Enter the external email address (e.g., `partner@gmail.com` or `colleague@othercompany.com`).
     - **Display Name:** Name of the user.
     - **Personal Message:** (Optional) A custom message included in the email invitation.
  3. **Review + Invite:** Sends the email.
- **The Redemption Process (Guest Experience):**
  1. **Email:** The user receives an email with a link to "Accept invitation".
  2. **Consent:** Clicking the link triggers a consent prompt, asking permission for the organization to read their profile.
  3. **Access:** Once accepted, the user is redirected to the My Apps portal (`myapplications.microsoft.com`) or the specific resource URL.
  4. **Initial State:** By default, the user may not see any applications until access is explicitly granted.
- **Granting Access to Applications:**
  - Guest users do not automatically get access to apps.
  - **Steps to Assign an App:**
    1. Navigate to **Enterprise applications**.
    2. Select (or add) an application (e.g., Adobe Creative Cloud, Salesforce, or a custom app).
    3. Go to **Users and groups** > **Add user/group**.
    4. Select the Guest User and assign them.
  - **Result:** The application will now appear in the guest user's My Apps dashboard.

[linkedin.com/in/nirajkum]

2.1.3 Bulk Invite External Users

- **Overview:**
  - Azure AD provides **Bulk Operations** to manage large sets of users efficiently, avoiding manual entry for each individual.
  - **Available Operations:** Bulk Create, Bulk Invite, Bulk Delete, and Download Users.
- **Bulk Invite via CSV (Portal):**
  - **Scenario:** Inviting multiple external partners at once.
  - **Process:**
    1. Navigate to **Users** > **Bulk operations** > **Bulk invite**.
    2. **Download Template:** Azure provides a specific CSV template.
    3. **Edit CSV:**
       - Fill in the required columns (e.g., **Email address to invite**).
       - Optional columns include **Redirection url** and custom messages.
       - *Important:* Do not modify the file structure or headers.
    4. **Upload:** Upload the saved CSV file to trigger the bulk invitation process.
- **Programmatic Invitation (PowerShell):**
  - **Scenario:** When user data is in a database or requires logic/transformation before inviting.
  - **Cmdlet:** `New-AzureADMSInvitation` (Azure AD PowerShell module).
  - **Method:** Write a script to iterate through a list of users and execute the invitation command for each.

2.1.4 Manage External Users

- **Management Parity:**
  - Managing external (Guest) users in the Azure Portal offers a similar experience to managing internal (Member) users.
  - **Common Administrative Tasks:**
    - **Security:** Revoke sessions (force sign-out), Reset password (if applicable to the account type), and Delete the account.
    - **Monitoring:** View **Sign-in logs** and **Audit logs** to track user activity and access history.
- **Access & Authorization:**
  - **Zero Trust Start:** Guest users start with no access (blank "My Apps" portal) until resources are explicitly assigned.
  - **Assignments:**
    - **Groups:** Add guests to groups for easier management.
    - **Applications:** Assign guests to Enterprise Applications (e.g., Adobe, Salesforce).
    - **Admin Roles:** Guests *can* be assigned administrative roles (e.g., **Helpdesk Administrator**) to manage resources or support users within your tenant.
- **On-Premises Limitations:**
  - Guest users are **Cloud-only** objects in the inviting tenant.
  - They are **not** synced back to the on-premises Active Directory.
  - **Result:** They cannot log in to on-premises domain-joined Windows devices or authenticate against local AD domain controllers.
- **Bulk Operations:**
  - Guest users can be managed via bulk operations (e.g., Bulk Delete) using CSV templates, just like member users.

[linkedin.com/in/nirajkum]

2.1.5 B2C Social Media Users

- **Overview:**
  - **B2C (Business-to-Consumer):** Focuses on "end users" or customers who register for applications using their existing social identities or email addresses, rather than a partner organization account.
  - **Goal:** Simplify the sign-up and sign-in process by allowing users to bring their own identity (BYOID).
- **Identity Providers (IdPs):**
  - Located under **External Identities** > **All identity providers**.
  - **Default Providers:**
    - **Azure Active Directory:** Standard authentication.
    - **Microsoft Account:** Allows users with Outlook, Hotmail, or Live accounts to sign in.
    - **Email One-Time Passcode (OTP):**
      - **Mechanism:** Users without a supported account receive a code via email to sign in (Magic Link).
      - **Configuration:** Can be enabled, disabled, or scheduled for future enforcement.
  - **Social Identity Providers:**
    - **Google / Facebook:**
      - Allows users to sign in with their Google or Facebook accounts.
      - **Setup:** Requires creating a developer application in the respective platform (Google/Facebook) to obtain a **Client ID** and **Client Secret**, which are then configured in Azure AD.
  - **Generic Providers (SAML / WS-Fed):**
    - **Purpose:** Integrate with any Identity Provider that supports SAML or WS-Federation protocols (e.g., LinkedIn, Twitter, or a custom IdP).
    - **Federation:** Establishes a trust relationship where Azure AD accepts authentication tokens from the external provider.

[linkendin.com/in/nirajkum]

2.2 Hybrid Identity

This section focuses on integrating on-premises Active Directory environments with Microsoft Entra ID. You will explore the concepts of hybrid identity, including directory synchronization using Azure AD Connect, and the various authentication methods available to provide a seamless single sign-on experience for users across on-premises and cloud resources.

2.2.1 Introduction to Hybrid Identity

- **Concept:**
  - **Hybrid Identity:** The integration of on-premises identity infrastructure (Windows Server Active Directory) with Microsoft Entra ID (cloud).
  - **Goal:** Provide a common user identity for authentication and authorization to all resources, regardless of location (on-premises or cloud).
- **Azure AD Connect:**
  - **Role:** The bridge between on-premises AD and Azure AD. It is an agent installed on a Windows Server in the local network.

- - **Function:** Synchronizes identity objects (Users, Groups, Contacts) from on-premises AD to Azure AD.
  - **Direction:** On-premises AD is the **Source of Authority** (Master). Changes made on-premises are pushed to the cloud during synchronization cycles (default every 30 minutes).
  - **Source Anchor:** An immutable attribute (e.g., `mS-DS-ConsistencyGuid` or `objectGUID`) used to uniquely identify and link an object between the two directories.
- **Prerequisites:**
  - **Routable Domain:** The on-premises User Principal Name (UPN) suffix (e.g., `@contoso.com`) should match a verified custom domain in Azure AD. If it uses a non-routable domain (e.g., `.local`), users will sync as `@tenant.onmicrosoft.com`.
- **Authentication Methods:**
  1. **Password Hash Synchronization (PHS):**
     - **Mechanism:** Synchronizes a hash of the user's on-premises password hash to Azure AD.
     - **Auth Location:** Authentication happens entirely in the **Cloud**.
     - **Pros:** Simplest to deploy, high availability (not dependent on on-prem uptime for auth), supports leaked credential detection.
     - **Default:** This is the default method.
  2. **Pass-Through Authentication (PTA):**
     - **Mechanism:** Validates passwords against the on-premises Active Directory via a lightweight agent installed on-premises.
     - **Auth Location:** Authentication happens **On-Premises**.
     - **Pros:** Enforces on-premises account policies (e.g., logon hours) immediately.
     - **Cons:** Dependent on on-premises infrastructure availability.
  3. **Federation (AD FS):**
     - **Mechanism:** Redirects authentication to a separate federation server (e.g., AD FS) or third-party provider (e.g., PingFederate).
     - **Auth Location:** Authentication happens on the **Federation Server**.
     - **Pros:** Supports advanced scenarios like smart card auth, third-party MFA integration, or complex conditional access policies not supported natively.
     - **Cons:** High complexity and infrastructure maintenance.
- **Seamless Single Sign-On (Seamless SSO):**
  - Automatically signs users in when they are on their corporate devices connected to the corporate network.
  - Can be enabled with PHS or PTA.
- **Monitoring & Health:**
  - **Azure AD Connect Health:** A robust monitoring tool to view the health of the sync engine and federation infrastructure (AD FS).
  - **Alerts:** Notifies admins of sync errors or infrastructure downtime.
  - **Licensing:** Basic sync is free, but advanced monitoring (Connect Health) often requires **Azure AD Premium P1**.
- **Troubleshooting Synchronization:**
  - **Common Errors:**
    - **Duplicate Attributes:** `UserPrincipalName` or `ProxyAddresses` must be unique across the entire tenant. If a synced user conflicts with an existing cloud user, a sync error occurs.
    - **Data Validation:** Ensure on-premises data meets Azure AD complexity and formatting requirements.

[linkedin.com/in/nirajkum]

2.2.2 Setup Azure AD Connect

- **Objective:** Install and configure the Azure AD Connect agent on a Windows Server to synchronize on-premises identities to the cloud.
- **Prerequisites:**
  - **Server:** A domain-joined Windows Server (standard or datacenter).
  - **Credentials:**
    - **Azure AD:** Global Administrator or Hybrid Identity Administrator.
    - **On-Premises:** Enterprise Administrator (for creating the AD account used by the service).
- **Installation Steps (Express vs. Custom):**
  1. **Connect to Azure AD:** Launch the installer and sign in with your Azure AD Global Admin credentials.
  2. **Connect to Directories:** Enter the credentials for the on-premises Active Directory Forest.
  3. **Domain and OU Filtering:**
     - **Domains:** Select the verified domains to sync.
     - **Filtering:** You can choose to sync specific Organizational Units (OUs) or specific groups.
     - *Example:* Syncing only the "Instructors" group rather than the entire directory.
  4. **Optional Features:**
     - **Password Hash Synchronization:** Selected as the sign-on method.
     - **Password Writeback:** (Optional) Allows cloud password changes to write back to on-prem AD. (Disabled in this demo).
  5. **Configure:** The wizard configures the sync engine and starts the initial synchronization.
- **Post-Installation:**
  - **Synchronization Cycle:** The scheduler runs every 30 minutes by default to sync changes (new users, password updates) from on-prem to cloud.
  - **Verification:**
    - Go to **Azure Portal** > **Users**.
    - Verify that on-premises users appear with the "Directory synced" status set to **Yes**.
  - **Monitoring:** Use **Azure AD Connect Health** to monitor the sync status and troubleshoot errors.

[linkedin.com/in/nirajkum]

Module 3: Authentication & Access Management

This module covers the critical aspects of securing user identities through robust authentication methods and access controls. You will learn about implementing Multi-Factor Authentication (MFA), passwordless strategies, and Self-Service Password Reset (SSPR). Furthermore, it dives into securing access using Conditional Access policies and managing enterprise applications to ensure only authorized users can access organizational resources.

3.1 Authentication Methods

This section explores the various authentication methods available in Microsoft Entra ID to secure user access. It covers the configuration and enforcement of Multi-Factor Authentication (MFA), the implementation of passwordless authentication strategies, and the use of password protection policies. Additionally, it details how to enable Self-Service Password Reset (SSPR) to empower users to manage their own credentials.

3.1.1 Introduction to Azure MFA

- **Overview:**
  - **Multi-Factor Authentication (MFA):** A security process that requires more than one method of authentication from independent categories of credentials to verify the user's identity.
  - **Goal:** Protect user identities. If a password is compromised, the attacker still cannot access the account without the second factor (e.g., the user's phone).
  - **Scope:** This course focuses on **Azure MFA (Cloud-based)**. The on-premises **MFA Server** is deprecated and not covered.
- **Authentication Factors:**
  - **Something you know:** Password.
  - **Something you have:** Phone (SMS, Call, Mobile App), Hardware token.
  - **Something you are:** Biometrics (Fingerprint, Face ID).
- **Common Methods:**
  - **Microsoft Authenticator App:** Generates time-based codes or push notifications.
  - **SMS/Text Message:** Sends a code to the registered mobile number.
  - **Phone Call:** Automated call to verify identity.
  - **Email:** Often used for guest users or specific scenarios.
- **Enabling MFA:**
  - **Per-User MFA (Legacy/Basic):**
    - Enabled individually for specific users via a dedicated portal link.
    - **Status:** Disabled -> Enabled -> Enforced (after registration).
    - **User Experience:** Once enabled, the user is forced to register for MFA methods upon their next sign-in.
  - **Organizational Strategy:**
    - Instead of enabling per-user, it is better to have a strategy based on risk, roles (admins), or login frequency.
    - This is typically handled via **Conditional Access** (covered later) or Security Defaults.
- **Configuration Location:**
  - In the Azure Portal, search for **"Multifactor Authentication"** to access service-level settings (fraud alerts, session settings, trusted IPs).

[linkendin.com/in/nirajkum]

3.1.2 MFA Settings

- **MFA Server:**
  - **Status:** Deprecated as of July 1, 2019. New deployments are not supported.
  - **Focus:** This course focuses on Azure MFA (Cloud).
- **Service Settings (Cloud MFA):**
  - Accessed via the "Additional cloud-based MFA settings" link in the portal.
  - **Trusted IPs:**
    - Allows users to skip MFA when signing in from a known corporate network location (Public IP CIDR ranges).
    - *Note:* Private IPs (e.g., 10.x.x.x) cannot be used here.
  - **Verification Options:**
    - **Call to phone:** Automated voice call.
    - **Text message to phone:** SMS with a code.

        - **Notification through mobile app:** Push notification (Approve/Deny) via Microsoft Authenticator.
        - **Verification code from mobile app:** Time-based One-Time Password (TOTP) via Microsoft Authenticator.
    - **Remember Multi-Factor Authentication:**
        - Allows users to mark a device as "Trusted" to skip MFA for a set number of days (1 to 365).
- **Azure Portal MFA Settings:**
    - **Account Lockout:**
        - Configures temporary account lockout after a sequence of failed MFA attempts (e.g., wrong code entered multiple times).
        - **Settings:** Number of denials before lockout, lockout duration, and reset counter duration.
    - **Block/Unblock Users:**
        - Manual administrative action to block a specific user from using MFA (e.g., if a device is lost or stolen).
    - **Fraud Alert:**
        - Allows users to report fraudulent MFA requests (e.g., receiving a push notification when they are not trying to sign in).
        - **Action:** Can be configured to automatically block the user when fraud is reported.
        - **Notifications:** Configure email recipients (e.g., Security Team) to receive alerts when fraud is reported.
    - **OATH Tokens:**
        - Support for hardware tokens (e.g., YubiKey, RSA key fobs) that generate OATH TOTP codes.
        - Admins can upload seed files to register these devices for users.
    - **Phone Call Settings:**
        - **Caller ID:** Configure a specific phone number to display when Azure calls users for MFA.
        - **Custom Greetings:** Upload audio files for custom greetings during the MFA call.

3.1.3 Passwordless Authentication

- **Concept:**
    - **The Goal:** Remove the password from the login experience entirely.
    - **Security vs. Convenience:**
        - **Passwords:** Convenient but Low Security.
        - **MFA:** High Security but Inconvenient.
        - **Passwordless:** High Security AND Convenient.
- **Three Main Passwordless Methods:**
    1. **Windows Hello for Business:**
        - **Type:** Device-specific biometric/PIN.
        - **Mechanism:** Replaces passwords with strong two-factor authentication on Windows 10/11 devices.
        - **Security:** Uses biometrics (Face, Fingerprint) or PIN tied to the device's TPM (Trusted Platform Module).
    2. **Microsoft Authenticator App (Phone Sign-In):**
        - **Type:** Software-based.
        - **Mechanism:** The user enters their username on the computer. A number is displayed on the screen. The user must open the app on their phone, select the matching number, and approve via biometric/PIN.

[linkendin.com/in/nirajkum]

        - **Benefit:** Turns the phone into a secure token.
    3. **FIDO2 Security Keys:**
        - **Type:** Hardware device (USB, NFC, Bluetooth).
        - **Mechanism:** "Fast Identity Online". Users plug in a key (e.g., YubiKey) and touch it (often with a fingerprint) to authenticate.
        - **Use Case:** High security environments, shared workstations (kiosks, hospitals) where mobile phones are restricted or not practical.
- **Configuration:**
    - Navigate to **Security** > **Authentication methods**.
    - **Enable Methods:** Select the specific method (e.g., Microsoft Authenticator) and toggle "Enable" to **Yes**.
    - **Targeting:** Assign to **All users** or specific **Groups**.
- **User Experience (Number Matching):**
    - To prevent "MFA Fatigue" (users blindly approving requests), Microsoft enforces **Number Matching**. The user must physically see the login screen to know which number to select on their phone.

3.1.4 Password Protection

- **Overview:**
    - Azure AD Password Protection detects and blocks known weak passwords and their variants.
    - It consists of a **Global Banned Password List** (maintained by Microsoft) and a **Custom Banned Password List** (maintained by you).
- **Smart Lockout:**
    - **Goal:** Protect against brute-force attacks by locking out potential attackers while ensuring valid users can still access their accounts.
    - **Configuration:**
        - **Lockout threshold:** The number of failed attempts allowed before the account is locked (Default: 10).
        - **Lockout duration:** The length of time the account remains locked in seconds (Default: 60).
    - **Intelligence:** Azure AD uses "smart" logic to distinguish between a likely attacker and the genuine user, potentially locking out the attacker's IP while leaving the user's access from a trusted location intact.
- **Custom Banned Password List:**
    - **Purpose:** Prevent users from using organization-specific terms in their passwords, which are easily guessable (e.g., Company Name, "Password123", local sports teams).
    - **Normalization:** The algorithm checks for common character substitutions (e.g., "P@ssw0rd" matches "Password").
    - **Configuration:** Enter a list of strings (one per line) to ban.
- **On-Premises Integration:**
    - **Password Protection for Windows Server AD:**
        - You can extend Azure AD password policies to your on-premises Active Directory Domain Controllers.
        - **Requirement:** Install the Azure AD Password Protection proxy and agent on on-premises servers.
    - **Modes:**
        - **Audit:** Logs when a user sets a weak password but does not block it.

linkendin.com/in/nirajkum

        - **Enforced:** Actively blocks users from setting passwords that violate the policy.

3.1.5 Self-Service Password Reset (SSPR)

- **Overview:**
    - **Self-Service Password Reset (SSPR)** allows users to reset their own passwords without contacting the help desk.
    - **Default State:** Disabled for standard users ("None"). Enabled by default for Administrators.
- **Enabling SSPR:**
    - Navigate to **Password reset** > **Properties**.
    - **Self Service Password Reset Enabled:**
        - **None:** Disabled.
        - **Selected:** Enable for specific groups (Recommended for piloting).
        - **All:** Enable for everyone in the tenant.
- **Authentication Methods:**
    - Define how users prove their identity before resetting a password.
    - **Number of methods required:** Choose 1 or 2.
    - **Available Methods:**
        - Mobile app notification / code.
        - Email.
        - Mobile phone (SMS).
        - Office phone (Call).
        - **Security Questions:** Can be used for SSPR (unlike MFA).
- **Registration:**
    - **Require users to register when signing in:** If set to **Yes**, users are prompted to set up their authentication data (phone/email) the next time they log in.
- **Notifications:**
    - **Notify users on password resets:** Sends an email to the user confirming the change (Security Alert).
    - **Notify all admins when other admins reset their password:** Alerts the admin team of privileged account changes.
- **Hybrid Identity (Password Writeback):**
    - **Scenario:** If users are synced from on-premises AD.
    - **Requirement: Password Writeback** must be enabled in Azure AD Connect.
    - **Function:** When a user resets their password in the cloud, Azure AD writes the new password back to the on-premises Active Directory in real-time. Without this, the on-prem password remains unchanged.
- **Administrator Policy:**
    - Admins always have SSPR enabled (cannot be turned off).
    - They are required to use strong authentication methods (Email, Phone, or App).

3.1.6 Enable Tenant Restrictions

- **Concept:**
    - **Tenant Restrictions** allow organizations to control access to SaaS applications based on the Azure AD tenant the applications use for single sign-on.

[linkendin.com/in/nirajkum]

    - **Goal:** Prevent data exfiltration by ensuring users on the corporate network can only log in to permitted external tenants (e.g., preventing login to a personal Outlook.com account or a competitor's tenant).
- **How it Works:**
    - **Network Level:** It is not a policy configured on the user object or device directly in Azure AD. Instead, it relies on **Traffic Inspection** at the corporate proxy or firewall.
    - **Header Injection:** The proxy inserts a specific HTTP header (`Restrict-Access-To-Tenants`) into outgoing traffic destined for Azure AD login endpoints (`login.microsoftonline.com`, `login.windows.net`, etc.).
    - **Header Content:** The header includes a list of allowed Tenant IDs or domains.
    - **Azure AD Enforcement:** When Azure AD receives the request, it checks the header. If the target tenant is not in the allowed list, Azure AD blocks the sign-in attempt.
- **Configuration:**
    1. **Proxy Configuration:** Configure the on-premises proxy/firewall (e.g., F5, Palo Alto, Zscaler) to enable SSL inspection and inject the required headers.
        - `Restrict-Access-To-Tenants`: `<Allowed-Tenant-List>`
        - `Restrict-Access-Context`: `<Your-Directory-ID>` (Used for reporting).
    2. **Azure Portal (Reporting):**
        - Navigate to **Identity** > **External Identities** > **Tenant restrictions**.
        - Here you can view reports of blocked sign-in attempts if the `Restrict-Access-Context` header is configured correctly.
- **User Experience:**
    - If a user tries to sign in to a blocked tenant (e.g., personal Xbox/Skype), they receive a message stating: "Access to this organization has been restricted by your administrator."

3.2 Security & Conditional Access

This section focuses on securing access to organizational resources using advanced security features. You will learn how to implement Azure AD Security Defaults for baseline protection, configure Tenant Restrictions to control access to external SaaS applications, and master Conditional Access policies to enforce granular access controls based on signals like user, location, device, and application. Additionally, it explores Azure AD Identity Protection to detect and remediate identity-based risks.

3.2.1 Azure AD Security Defaults

- **Overview:**
    - **Security Defaults** is a set of basic identity security mechanisms recommended by Microsoft.
    - **Default State:** Enabled by default for all new tenants created after October 2019.
    - **Target Audience:** Organizations that want a baseline level of security without configuring complex Conditional Access policies (often smaller businesses or those on the Free tier).
- **What Security Defaults Enforce:**
    1. **Unified MFA Registration:** All users are required to register for Multi-Factor Authentication using the Microsoft Authenticator app. They have a **14-day grace period** to register after their first sign-in.
    2. **MFA for Administrators:** Users with highly privileged roles (e.g., Global Admin, Security Admin, Exchange Admin) are required to perform MFA every time they sign in.

[linkendin.com/in/nirajkum]

   3. **MFA for All Users:** Standard users are prompted for MFA when necessary (based on usage patterns), though not necessarily every time.
   4. **Blocking Legacy Authentication:** Older protocols that do not support MFA (like POP3, IMAP, SMTP, and older Office clients) are blocked to prevent password spray attacks.
   5. **Privileged Activities:** Access to the Azure Portal, Azure CLI, and PowerShell is restricted to authenticated users with MFA.

- **Security Defaults vs. Conditional Access:**
  - **Mutually Exclusive:** You cannot use both simultaneously.
  - **Limitation:** Security Defaults are "all or nothing." You cannot exclude specific accounts (like a Break Glass account) or customize the rules.
  - **Transition:** To implement custom **Conditional Access** policies (which offer granularity), you must first **disable** Security Defaults.
- **Disabling Security Defaults:**
  1. Navigate to **Microsoft Entra ID** > **Properties**.
  2. Click the link **Manage security defaults** (at the bottom).
  3. Set **Enable security defaults** to **No**.
  4. Select a reason for disabling (e.g., "My organization is using Conditional Access").
  5. Click **Save**.

3.2.2 Azure AD Conditional Access

- **Concept:**
  - **Conditional Access** is the policy engine of Microsoft Entra ID (Zero Trust). It analyzes signals to make access decisions and enforce organizational policies.
  - **Logic:** "If this (Conditions), Then that (Access Controls)."
- **Key Components (Signals & Decisions):**
  - **Assignments (Who):**
    - Defines which users or groups the policy applies to.
    - **Best Practice:** Always exclude a "Break Glass" (emergency access) account to prevent locking yourself out of the tenant.
  - **Cloud Apps or Actions (What):**
    - **Cloud Apps:** Select specific apps (e.g., Azure Portal, Office 365) or "All Cloud Apps".
    - **User Actions:** Register security information, Register or join devices.
  - **Conditions (When):**
    - **User Risk:** (Requires Identity Protection) Likelihood that a user identity is compromised (e.g., leaked credentials). Levels: High, Medium, Low.
    - **Sign-in Risk:** (Requires Identity Protection) Likelihood that a specific sign-in is suspicious (e.g., anonymous IP, impossible travel).
    - **Device Platforms:** Android, iOS, Windows, macOS.
    - **Locations:** Trusted locations (corporate network) vs. Untrusted locations.
    - **Client Apps:** Browser vs. Mobile apps vs. Legacy authentication.
  - **Access Controls (Grant/Block):**
    - **Block Access:** Deny the sign-in.
    - **Grant Access:** Allow access if requirements are met:
      - Require Multi-Factor Authentication (MFA).
      - Require device to be marked as compliant (Intune).
      - Require Hybrid Azure AD joined device.

[linkedin.com/in/nirajkum]

- **Policy State:**
  - **Report-only:** Logs the policy result without enforcing it (Audit mode).
  - **On:** Enforces the policy.
  - **Off:** Disabled.
- **Example Scenario: Risky Accounts Policy**
  - **Goal:** Require MFA for users with Medium or High risk.
  - **Steps:**
    1. **Name:** "Risky Accounts".
    2. **Users:** Select specific group (e.g., "Students").
    3. **Cloud Apps:** Select "All cloud apps".
    4. **Conditions:** Under **User risk**, configure "Yes" and select **High** and **Medium**.
    5. **Grant:** Select **Grant access** and check **Require multi-factor authentication**.
    6. **Enable Policy:** Set to **On**.

3.2.3 Test Conditional Access

- **The "What If" Tool:**
  - **Purpose:** A simulation tool within the Conditional Access blade that allows administrators to predict the impact of policies on a user's sign-in without actually performing the sign-in.
  - **Use Case:** Essential for troubleshooting existing policies and validating new policies before they are enforced (to prevent accidental lockouts).
- **How to Use:**
  1. Navigate to **Conditional Access** > **What If**.
  2. **Select User:** Choose the specific user you want to test (e.g., a Student or a Teacher).
  3. **Configure Conditions:** Simulate the scenario by setting:
     - **Cloud Apps:** Which app are they accessing?
     - **IP Address:** Are they coming from a trusted or untrusted location?
     - **Device Platform:** Windows, iOS, Android?
     - **Risk Level:** Simulate a "High" user risk or sign-in risk to see if risk-based policies trigger.
  4. **Run Evaluation:** Click **What If**.
- **Results:**
  - **Policies that will apply:** Lists policies where the user matches all assignments and conditions. Shows the Grant/Block controls that would be enforced.
  - **Policies that will not apply:** Lists policies that were evaluated but didn't trigger. Crucially, it provides the **reason** (e.g., "User not included in policy" or "Condition not matched").

3.2.4 AD Identity Protection

- **Overview:**
  - **Identity Protection** automates the detection and remediation of identity-based risks using Microsoft's threat intelligence and machine learning.
  - **Licensing:** Requires **Azure AD Premium P2**.
  - **Core Function:** It assesses risk at two levels: **User Risk** and **Sign-in Risk**.
- **1. User Risk Policy:**
  - **Definition:** Detects the probability that a user's identity has been compromised (e.g., credentials found on the dark web).
  - **Configuration:**

[linkedin.com/in/nirajkum]

        - **Assignments:** Select users/groups (e.g., All Users).
        - **Conditions:** Set the risk level threshold (e.g., High).
        - **Controls:**
            - **Block Access:** Prevent the user from signing in.
            - **Allow access and require password change:** The user must perform MFA and then reset their password to self-remediate the compromised identity.
- **2. Sign-in Risk Policy:**
    - **Definition:** Detects the probability that a specific sign-in attempt is suspicious (e.g., impossible travel, anonymous IP address, unfamiliar sign-in properties).
    - **Configuration:**
        - **Assignments:** Select users/groups.
        - **Conditions:** Set the risk level threshold (e.g., Medium and above).
        - **Controls:**
            - **Block Access.**
            - **Allow access and require multi-factor authentication:** If the user passes the MFA challenge, the risk is considered remediated (the user proved they are who they say they are despite the suspicious location).
- **3. MFA Registration Policy:**
    - **Purpose:** Ensures users are registered for MFA so they can actually respond to the risk-based challenges defined above.
    - **Behavior:** When enabled, users are prompted to set up MFA upon their next interactive sign-in and have a **14-day grace period** to complete registration.
- **Reporting:**
    - **Risky users:** A report showing users currently flagged as at-risk.
    - **Risky sign-ins:** A log of specific authentication events that triggered risk detections.

[linkendin.com/in/nirajkum]

3.3 Application Management

This section covers the integration and management of applications within Microsoft Entra ID. You will learn how to register applications, configure Enterprise Applications for Single Sign-On (SSO), manage user consent settings, and monitor application usage to ensure secure and seamless access to organizational resources.

3.3.1 Introduction to Enterprise Application Integration

- **Overview:**
    - **Objective:** Implement Access Management for Apps (10-15% of exam).
    - **Concept:** Use Microsoft Entra ID (Azure AD) as the central **Identity Provider (IdP)** for third-party SaaS applications (Service Providers).
    - **Single Sign-On (SSO):** Enables users to access external applications (like Salesforce, AWS, Dropbox) using their corporate Entra ID credentials.
- **Benefits:**
    - **Security:** Centralized control over access. If a user leaves, disabling their Entra ID account revokes access to all integrated apps immediately.
    - **User Experience:** Users only need to remember one set of credentials.
    - **Compliance:** Logs and audit trails for application access are centralized in Entra ID.
- **The Application Gallery:**

    - **Location: Enterprise applications** > **New application**.
    - **Content:** A repository of thousands of pre-integrated applications (e.g., AWS, Google Cloud, Oracle, SAP, Adobe, Atlassian, Box, DocuSign, Cisco Webex, GitHub).
    - **Function:** These gallery apps come with pre-configured settings to simplify the setup of SSO and user provisioning.
- **Integration Workflow:**
    1. **Add Application:** Select the app from the Azure AD Gallery.
    2. **Configure SSO:** Set up the trust relationship (usually SAML or OIDC) between Azure AD and the application. This often requires configuration on both sides (Azure AD and the App's admin console).
    3. **Assign Users:** Grant specific users or groups access to the application.
- **Scenario Example (AWS):**
    - By integrating AWS with Azure AD, a user can log in to the AWS Management Console using their Azure AD account.
    - Azure AD authenticates the user and passes their role information to AWS, determining what they can do within the AWS environment.

3.3.2 Application Controls

- **Overview:**
    - Beyond simple Grant/Block access, Azure AD can enforce granular controls *within* an application session using **Conditional Access App Control**.
    - This integrates with **Microsoft Defender for Cloud Apps** (formerly MCAS) to act as a reverse proxy for real-time monitoring and control.
- **Session Controls:**
    - Located under the **Session** section of a Conditional Access policy (not Grant/Block).
    - **Use Conditional Access App Control:**
        - **Monitor only:** Logs user activities in the app.
        - **Block downloads:** Prevents users from saving files to their local device.
        - **Use custom policy:** Allows for granular rules defined in Defender for Cloud Apps (e.g., "Block download of files containing credit card numbers").
- **Example Scenario: Restricting Dropbox**
    - **Goal:** Allow users to access Dropbox to view files but prevent them from downloading files to unmanaged devices.
    - **Configuration:**
        1. **Policy Name:** "Block Dropbox Downloads".
        2. **Users:** All Users.
        3. **Cloud Apps:** Select **Dropbox**.
        4. **Session:** Check **Use Conditional Access App Control** and select **Block downloads** (or Custom Policy).
    - **Result:** When a user logs into Dropbox via Azure AD, their session is routed through the Defender for Cloud Apps proxy. If they try to download a file, the action is blocked in real-time.

Module 4: Identity Governance

[linkedin.com/in/nirajkum]

This module delves into the advanced capabilities of Microsoft Entra Identity Governance. You will learn how to automate access lifecycle management using Entitlement Management, ensure that access rights remain appropriate over time with Access Reviews, and manage and secure privileged roles using Privileged Identity Management (PIM) to enforce the principle of least privilege.

4.1 Entitlement Management

This section covers Microsoft Entra Entitlement Management, a key identity governance feature. You will learn how to simplify access management by bundling resources (group memberships, application access, SharePoint sites) into 'access packages' that users can request. This automates the access request, approval, and lifecycle management process for both internal and external users.

4.1.1 Introduction to Entitlement Management and Packages

- **Overview:**
  - **Identity Governance:** This topic covers the planning and implementation of identity governance strategies, which accounts for **25-30%** of the SC-300 exam.
  - **The Challenge:** Managing permissions at scale is difficult. It is often impossible to predetermine exactly what permissions every user needs, leading to permission creep or administrative bottlenecks.
  - **The Solution: Entitlement Management** automates access request workflows, access assignments, reviews, and expiration.
- **Navigation:**
  - Located in the Azure Portal under **Identity Governance** > **Entitlement management**.
- **Key Concepts:**
  - **Access Packages:**
    - **Definition:** A bundle of all the resources a user needs to work on a project or perform a specific role.
    - **Resource Types:** Can include:
      - **Groups and Teams:** (e.g., Microsoft 365 Groups).
      - **Applications:** (e.g., Enterprise Applications, SaaS apps).
      - **SharePoint Sites:** (e.g., Online sites).
      - **Licenses:** Software licenses can be included.
    - **Workflow:**
      1. **Request:** Users (internal or external) request access to the package.
      2. **Approval:** The request goes through an approval workflow (e.g., Manager or Project Lead approval).
      3. **Provisioning:** Upon approval, the user is automatically given access to all resources in the bundle.
    - **Lifecycle & Expiration:**
      - Access is **time-bound** (e.g., 30 days).
      - **Automatic Revocation:** When the assignment expires, access to all resources is automatically removed. No manual cleanup is required.
    - **External User Cleanup:** If a guest user (B2B) has no other access rights in the tenant other than this expired package, their guest account can be automatically removed from the directory.

[linkendin.com/in/nirajkum]

  - **Catalogs:**
    - **Purpose:** A container used to group and organize Access Packages.
    - **Usage:** You create Access Packages and arrange them into Catalogs (e.g., "Marketing Catalog", "External Partners Catalog").

4.1.2 Create and Manage Access Packages

- **Scenario:**
  - We will create an Access Package that allows users (Students and Teachers) to voluntarily join a specific Security Group ("Team Assignment 1") for a limited duration (60 days) without requiring manual approval.
- **Step 1: Create the Resource (Prerequisite)**
  - Before creating the package, the resource must exist.
  - Navigate to **Groups** > **New group**.
  - Create a **Security** group named "Team Assignment 1". Leave members empty.
- **Step 2: Create the Access Package**
  1. **Navigation:** Go to **Identity Governance** > **Entitlement management** > **Access packages**.
  2. **Initiate:** Click **New access package**.
  3. **Basics Tab:**
     - **Name:** Enter a name (e.g., "Student Assignment Access").
     - **Description:** Optional description.
     - **Catalog:** Use the default **General Catalog**.
  4. **Resource Roles Tab:**
     - **Groups and Teams:** Click **Select groups** and choose "Team Assignment 1".
     - **Role:** Select **Member** (The permission granted within the group).
  5. **Requests Tab:**
     - **Users who can request access:** Select **For users in your directory**.
     - **Select users and groups:** Choose specific groups allowed to request this (e.g., "Students" and "Teachers").
     - **Approval:** Set **Require approval** to **No** (Automatic provisioning).
     - **Enable new requests:** Set to **Yes**.
  6. **Lifecycle Tab:**
     - **Expiration:** Set access to expire after **60 days**.
     - **Access Reviews:** Set to **No** (covered in later sections).
  7. **Review + Create:** Click **Create**.
- **Step 3: The User Experience (My Access Portal)**
  - **My Access Link:** On the Access Package overview page, copy the **My Access portal link**.
  - **Requesting Access:**
    1. The user (Student) navigates to the link.
    2. Signs in with their Azure AD credentials.
    3. Sees the available package and clicks **Request access**.
    4. Enters a **Justification** (e.g., "Need to join the project") and submits.
  - **Result:** Since approval is disabled, the status changes to "Delivered" almost immediately.
- **Step 4: Verification**
  - As an Administrator, navigate back to **Groups** > "Team Assignment 1" > **Members**.
  - Verify that the requesting user has been automatically added to the group.

[linkedin.com/in/nirajkum]

    - **Note:** After the 60-day duration, Entitlement Management will automatically remove the user from this group.

4.1.3 Create and Require Terms of Use

- **Overview:**
    - **Purpose:** Organizations often require users to accept legal terms or compliance policies before accessing corporate resources.
    - **Mechanism:** This is implemented using **Conditional Access**.
    - **Licensing:** Requires **Azure AD Premium P1** or higher.
- **Step 1: Define the Terms of Use (ToU)**
    - **Navigation:** Go to **Conditional Access** > **Terms of use**.
    - **Create New Terms:**
        - **Name:** Internal name for the policy.
        - **Display Name:** What the user sees.
        - **Document:** Upload the PDF containing the legal text.
        - **Language:** You can upload multiple PDFs for different languages (English, French, Spanish, etc.).
        - **Require users to expand:** If set to **On**, users must scroll to the bottom of the document before the "Accept" button becomes active.
        - **Expire consents:**
            - **Frequency:** You can force users to re-accept the terms on a schedule (e.g., Annually).
            - **Duration:** Set a specific date or frequency (e.g., every 365 days).
- **Step 2: Enforce via Conditional Access**
    - Creating the Terms of Use document alone does not enforce it; you must link it to a Conditional Access policy.
    - **Create Policy:**
        1. **Assignments:** Select **All Users** (or specific groups).
        2. **Cloud Apps:** Select **All Cloud Apps**.
        3. **Grant Controls:**
            - Select **Grant access**.
            - Check the box corresponding to the **Terms of Use** created in Step 1.
            - *Note:* This makes accepting the terms a requirement for the token to be issued.
- **User Experience:**
    - When a targeted user signs in to an application, the sign-in flow is interrupted.
    - A page displays the PDF document.
    - The user must view the document and click **Accept**.
    - Azure AD records the acceptance (audit trail), and the user is not prompted again until the terms expire or are updated.

4.1.4 External User Lifecycle Management

- **Overview:**
    - **Challenge:** External users (guests) often accumulate in a directory long after their project or contract has ended.

[linkendin.com/in/nirajkum]

    - **Solution:** Entitlement Management can automatically manage the lifecycle of these external users based on their access package assignments.
- **Configuration:**
    - **Navigation:** Go to **Identity Governance** > **Entitlement management** > **Settings**.
    - **Manage the lifecycle of external users:** Click **Edit**.
- **Lifecycle Settings:**
    - **Trigger:** These settings apply when an external user (who was invited through an access package) loses their **last** assignment to any access package.
    - **Block Sign-in:**
        - **Setting:** "Block external user from signing in to this directory".
        - **Default: Yes**.
        - **Effect:** If the user has no active packages, their account is disabled, preventing login.
    - **Delete User:**
        - **Setting:** "Remove external user".
        - **Default: Yes** (after **30 days**).
        - **Effect:** If the user remains blocked and receives no new assignments for the specified duration, the guest account is permanently deleted from the directory.
    - **Customization:** You can extend the number of days before deletion or choose not to delete users automatically (e.g., if they need to retain access for historical reasons or other manual assignments).

4.2 Access Reviews

This section focuses on **Access Reviews**, a critical component of Identity Governance. As organizations grow, users often accumulate permissions they no longer need ("permission creep"). Access Reviews allow organizations to efficiently manage group memberships, access to enterprise applications, and role assignments by requiring regular recertification by owners or managers. This ensures that only the right people have continued access to resources.

4.2.1 Introduction to Access Reviews

- **Overview:**
    - **Definition:** Access Reviews are an automated service within Microsoft Entra Identity Governance that enables organizations to efficiently manage group memberships, access to enterprise applications, and role assignments.
    - **Goal:** To mitigate "permission creep" by ensuring users only retain access to the resources they currently need.
- **Navigation:**
    - Located in the Azure Portal under **Identity Governance** > **Access reviews**.
- **Core Concepts:**
    - **The Workflow:**
        1. **Scope:** Define what is being reviewed (e.g., Members of the "Sales" group).
        2. **Reviewers:** Define who performs the review.
            - **Self-Review:** Users review their own access.
            - **Manager Review:** Managers review their direct reports' access.
            - **Group Owners:** Owners of the group review membership.

[linkendin.com/in/nirajkum]

        3. **Recurrence:** Set the frequency (e.g., Quarterly, Annually).
    - **Decision Logic:**
        - **Approve:** User retains access.
        - **Deny:** User loses access (can be auto-remediated).
        - **No Response:** Define a fallback action (e.g., Remove access, Keep access, or Accept recommendations).
- **Strategic Implementation:**
    - **Balancing Friction:** Security controls often introduce friction (like MFA). Access Reviews add administrative overhead.
    - **Cadence:** Choose a frequency that balances security needs with user productivity.
    - **Psychology:** If users know they can easily request access again (via Entitlement Management/Access Packages), they are more likely to honestly deny their own unneeded access during a review.

4.2.2 Create Access Reviews

- **Scenario:**
    - Create a review for the "Students" and "Teachers" groups where members must self-attest their need for continued access.
- **Step 1: Initiate Review**
    - Navigate to **Identity Governance** > **Access reviews**.
    - Click **New access review**.
- **Step 2: Review Type & Scope**
    - **Select what to review:**
        - **Teams + Groups:** Reviews membership of Security Groups or M365 Groups.
        - **Applications:** Reviews assignment to Enterprise Applications (e.g., Salesforce, Adobe).
    - **Review Scope:**
        - **Select teams + groups:** Choose specific groups (e.g., "Students", "Teachers").
    - **Scope Users:**
        - **Guest users only:** Limits review to B2B users.
        - **All users:** Includes internal member users as well.
- **Step 3: Reviews (Who performs the review?)**
    - **Select Reviewers:**
        - **Group owners:** The owners of the group perform the review.
        - **Selected user(s) or group(s):** Specific auditors.
        - **Users review their own access:** (Self-Review) Users are asked if they still need access.
        - **Managers of users:** Uses the `Manager` attribute in AD.
    - *Decision:* For this demo, we select **Users review their own access**.
- **Step 4: Settings (Recurrence & Logic)**
    - **Duration (in days):** How long the review remains open (e.g., 3 days).
    - **Recurrence:**
        - **One time:** A single ad-hoc review.
        - **Weekly/Quarterly/Annually:** Recurring compliance checks.
- **Step 5: Upon Completion Settings**
    - **Auto-apply results to resource:** If enabled, users denied access are automatically removed from the group/app.
    - **If reviewers don't respond:** (Fallback action)

[linkedin.com/in/nirajkum]

        - **No change:** Access remains.
        - **Remove access:** Strict security posture.
        - **Approve access:** Lenient.
        - **Take recommendations:** Uses system intelligence.
    - **Decision Helpers (Recommendations):**
        - **No sign-in within 30 days:** The system recommends "Deny" if the user hasn't used the account recently.
    - **Advanced:**
        - **Justification required:** User must type why they need access.
        - **Email notifications:** Alerts users/admins.
        - **Reminders:** Sends follow-up emails if review is pending.
- **Step 6: Finalize**
    - **Review Name:** Enter a descriptive name (e.g., "Student Teacher Review").
    - Click **Create**.
    - **Outcome:** Emails are sent to reviewers (or users in self-review) to begin the process.

4.2.3 Perform an Access Review

- **The Reviewer Experience (User Side):**
    - **Notification:** When a review starts, assigned reviewers (or users in a self-review) receive an email notification with a direct link.
    - **My Access Portal:** Alternatively, users can navigate directly to **myaccess.microsoft.com**.
    - **Steps to Review:**
        1. Log in to the My Access portal.
        2. Select **Access reviews** from the left menu.
        3. Click on the pending review (e.g., "Student Teacher Review").
        4. **Decision:**
            - **Approve:** Select **Yes** if access is still required.
            - **Deny:** Select **No** if access is no longer needed.
        5. **Justification:** Enter a business reason if required (e.g., "Need access for project X until Q4").
        6. **Submit:** The decision is recorded.
- **The Administrator Experience (Monitoring):**
    - **Tracking Progress:**
        - Navigate to **Identity Governance** > **Access reviews**.
        - Select the active review to view the dashboard.
        - **Overview:** Shows the progress bar (e.g., 1 of 3 users reviewed).
        - **Results:** Detailed view of each user's status (Approved, Denied, Not Reviewed) and the system's **Recommendation** (based on sign-in activity).
    - **Management Actions:**
        - **Stop:** Ends the review immediately.
        - **Reset:** Restarts the review cycle.
        - **Delete:** Removes the review configuration.
- **Completion & Remediation:**
    - Once the duration expires (e.g., after 3 days), the configured **Upon completion** settings apply.
    - **Auto-apply:** If enabled, users who were denied or didn't respond (based on fallback settings) are automatically removed from the group or application.

[linkendin.com/in/nirajkum]

4.2.4 Access Review Licensing

- **License Requirement:**
  - Access Reviews require an **Azure AD Premium P2** (or Microsoft Entra ID P2) license.
- **Who Needs a License?**
  - **Reviewers:** Any user who performs an action in an access review needs a license. This includes:
    - **Self-Reviewers:** Users reviewing their own access.
    - **Group/App Owners:** Owners reviewing membership or assignment.
    - **Managers:** Managers reviewing direct reports.
- **Who Does NOT Need a License?**
  - **Creators:** The Administrator who creates or manages the review settings does not strictly require a license for that specific action (unless they are also a reviewer).
- **External Users (Guests):**
  - Guest users performing reviews fall under the **External Identities Monthly Active Users (MAU)** billing model.
  - They must be covered by the tenant's P2 allowance.

4.3 Privileged Identity Management (PIM)

This section covers **Privileged Identity Management (PIM)**, a service in Microsoft Entra ID that enables you to manage, control, and monitor access to important resources. PIM helps mitigate the risks of excessive, unnecessary, or misused access rights by enforcing the principle of least privilege through **Just-In-Time (JIT)** access. Instead of having permanent admin rights ("standing access"), users are made "eligible" for a role and must activate it for a specific duration when needed.

4.3.1 Introduction to Privileged Identity Management

- **Overview:**
  - **Privileged Identity Management (PIM):** A service that manages, controls, and monitors access to important resources in your organization.
  - **Just-In-Time (JIT) Access:** PIM provides time-bound access to resources using JIT. Instead of users having permanent "standing access" (admin rights 24/7), they only have privileges when they need them.
  - **Analogy:** Similar to the `sudo` command in Linux. A user operates with standard permissions and only elevates to root/admin privileges for specific tasks.
- **Key Capabilities:**
  - **Request Access:** Users must actively request to use their role.
  - **Justification:** Users can be required to enter a business justification or ticket number.
  - **Approval Workflow:** Activation can require approval from a designated manager or admin.
  - **Notifications & Auditing:** Admins are notified when roles are activated, and audit logs track all activity.
  - **Access Reviews:** Ensure continued need for access (recertification).
- **Licensing:**
  - Requires **Azure AD Premium P2** (or Microsoft Entra ID P2).
- **Navigation:**
  - Access via the Azure Portal > **Privileged Identity Management**.

[linkedin.com/in/nirajkum]

- **Implementation Strategy:**
  1. **Plan:** Don't turn it on without a plan. Define policies and roles.
  2. **Pilot:** Test with a small group of users first.
  3. **Communicate:** Educate users on the workflow changes (activating roles vs. having them permanently).
- **Assignment Types:**
  - **Active Assignment:**
    - The user has the role privileges assigned immediately.
    - Can be **Permanent** (traditional admin) or **Time-bound** (expires at a specific future date/time).
  - **Eligible Assignment:**
    - The user does *not* have the privileges by default.
    - They are "Eligible" to activate the role.
    - **Activation:** Requires the user to perform an action (MFA, justification, approval) to gain the privileges for a limited duration (e.g., 4 hours).
    - **Benefit:** Reduces the attack surface. If the account is compromised while not active, the attacker does not have admin rights.

4.3.2 Assigning Roles with PIM

- **Scope of PIM:**
  - PIM is not limited to **Azure AD Roles** (Global Admin, etc.).
  - It also manages **Azure Resources** (Subscriptions, Resource Groups, Management Groups) using Azure RBAC roles (e.g., Contributor, Virtual Machine Operator).
- **Onboarding Azure Resources:**
  - **Discovery:** Before managing resources, PIM must "Discover" them.
  - **Navigation:** PIM > **Azure resources** > **Discover resources**.
  - **Onboarding:** Select a Subscription and click **Manage resource**.
  - **Effect:** This action integrates PIM into the role assignment workflow for that subscription, enabling time-bound and eligible access.
- **Assigning a Role (PIM Workflow):**
  1. **Select Resource:** Open the managed Subscription in PIM.
  2. **Select Role:** Go to **Roles** and choose a specific RBAC role (e.g., **Virtual Machine Operator**).
  3. **Add Assignment:** Click **Add assignments**.
  4. **Select Member:** Choose the user or group.
  5. **Configure Settings:**
     - **Assignment Type:**
       - **Eligible:** User must activate to use. (Example limit: 1 year eligibility).
       - **Active:** User has access immediately but for a limited time (e.g., 6 months).
     - **Duration:** Define start and end dates.
- **PIM vs. Traditional IAM:**
  - **Traditional (IAM blade):** Allows for permanent, standing access without expiration or justification.
  - **PIM:** Enforces **Least Privilege** by requiring justification, approval (optional), and enforcing expiration dates on assignments.

[linkendin.com/in/nirajkum]

4.3.3 Emergency Break Glass Accounts

- **Overview:**
    - **Purpose:** An emergency access account (often called a "break-glass" account) is a highly privileged account used *only* when normal administrative access is unavailable.
    - **Risk Scenario:** You could be locked out of your tenant due to:
        - MFA service outages.
        - Misconfigured Conditional Access policies (e.g., blocking all IPs).
        - PIM approval workflows failing (no eligible approvers).
- **Configuration Best Practices:**
    - **Cloud-Only:** The account must be created directly in Entra ID (not synced from on-prem) to ensure access if on-premises infrastructure fails. Use the `*.onmicrosoft.com` domain.
    - **Role:** Permanently assigned the **Global Administrator** role.
    - **Exclusions:**
        - **MFA:** Do **not** register for MFA.
        - **Conditional Access:** Explicitly **Exclude** this account from *all* Conditional Access policies.
        - **PIM:** Do not require PIM activation; the role should be permanent.
- **Securing the Account:**
    - **Complex Credentials:** Use a very long, complex password (generated randomly).
    - **Split Knowledge:** Split the password into two or three parts. Give each part to a different senior administrator or store them in separate physical safes ("Two-man rule").
    - **Physical Security:** Store the credentials in a secure, fireproof location.
- **Monitoring & Maintenance:**
    - **Alerting:** Configure log monitoring to send an immediate high-priority alert (SMS/Email) to the security team whenever this account signs in.
    - **Testing:** Validate the account periodically (e.g., every 90 days) to ensure the password works and the account hasn't been blocked.

Summary:

This study guide provides a comprehensive overview of the key skills measured in the SC-300 exam. It began with the fundamentals of **Identity Management**, covering tenant configuration, user and group lifecycle, and role-based access control. We then explored **External Identities and Hybrid Identity**, detailing how to secure collaboration and synchronize on-premises directories with the cloud.

The guide emphasized **Authentication and Access Management**, focusing on MFA, passwordless methods, and the implementation of robust Conditional Access policies to secure resources. Finally, we covered **Identity Governance**, explaining how to automate access lifecycles with Entitlement Management, perform Access Reviews, and secure administrative access using Privileged Identity Management (PIM). Mastering these concepts is essential for any Identity and Access Administrator working with Microsoft Entra ID.

[linkendin.com/in/nirajkum]