# AZ-104T00A – Administer Identity

When you set up an IT environment, one of the first things you need to do is give access to the people who need it. This typically includes system administrators, software developers, and end-users. The solution is to use an identity and access management system. Microsoft has long been a leader in the identity space. This leadership goes back to the introduction of Active Directory (AD) with Windows 2000 almost 23 years ago. At a very high level, Active Directory allows you to create and manage user accounts. Then when a user logs in to one of your IT systems, the system will ask for their username and password and then verify these credentials with Active Directory before letting the user in. This process is known as authentication and I think everybody in this virtual room does that a couple of times a day, let’s say so.

## Configure Azure Active Directory

Describe Azure Active Directory Benefit and features.

Microsoft in 2010 moved into cloud identity with the introduction of Azure and Azure Active Directory (Azure AD), which is now used by more than 5 million companies around the world. When you set up an environment on Azure, instead of using Active Directory, you use Azure Active Directory. It’s quite similar, but it’s used for accessing cloud applications. That includes cloud applications outside of Azure, too, such as Microsoft 365.

**SSO functionalities**

Of course, if you already have an on-premises Active Directory implementation, you don’t have to recreate all of your users and groups in Azure Active Directory. Instead, you can synchronize your accounts between the two systems using Azure AD Connect and I’m going to show you this feature to make the class a little more interesting otherwise hearing me talk forever, I would fall asleep too.1 This will create the accounts on Azure for you and keep them in sync when changes are made to the accounts in Active Directory. It will allow you to use single sign-on (or SSO), which means that users only need to log in once to access both their on-premises environment and their Azure environment. You can also use SSO to access Microsoft 365.

**Multi-factor Authentication**

In the past, authentication was mostly handled by getting each person to enter a user ID and password, but these days, you’ll likely want to check more than that to make sure the person isn’t a hacker. One way to get more assurance is to use multifactor authentication (or MFA). In addition to asking the user for something they know, such as their password, you also require something the user has, such as a mobile phone number, or something the user is, such as a human being with a fingerprint. You’ve probably had to go through multifactor authentication yourself. The most common way to prove you have something unique to you is to get an access code sent to your mobile device and then enter that code during the login process. The most common way to prove you are who you say you are is to put your finger or thumb on a fingerprint reader. Another way is to let a device scan your face. Azure AD does support MFA in the ”what you have” category with: 1) Access codes sent by SMS. 2) Access Codes sent by Voice call. 3) Microsoft Authenticator App. 4) Hardware keys, Including FIDO2 security keys and OATH Hardware tokens. 5) OATH software tokens. In the “what you are” category, it currently supports only Windows Hello for Business. Which can use facial and fingerprint recognition.

**Identity Protection**

Since most organizations are constantly under threat from hackers trying to gain access to their resources, Microsoft provides a feature called Identity Protection. It performs automated risk detection by looking for signs of an intrusion attempt. It looks for things like users logging in from anonymous IP addresses or from unexpected locations. It also looks for things like “password spray”, which is where an attacker tries the same password on many accounts. You can configure Identity Protection so that when it detects these types of risky login attempts, it automatically takes an action, such as requiring multifactor authentication or blocking the login attempt, or you could turn off the automated remediation feature and investigate the detected risks manually. Or, if you have your own security information and event management (or SIEM) system, then you could export the risk detection data to it and deal with the identity risks using that tool.

**Privilege Identity Management**

While it’s important to try to prevent hackers from gaining access to regular user accounts, it’s especially important to prevent them from accessing administrator accounts because those accounts have privileged access. To help with this, Microsoft offers the Privileged Identity Management (or PIM) service. Another reason to use PIM is to help prevent legitimate administrators from accidentally causing issues. PIM focuses on restricting administrator access to only the people who need it, putting extra requirements in place before administrators can perform privileged actions, and keeping an audit trail of what administrators have done. One way to ensure that only the people who truly need administrator privileges have them is to conduct access reviews. This requires managers to review the list of administrators on a regular basis. To control the use of administrator accounts, PIM provides just-in-time access. It works in this way: Certain users are designated as eligible to perform administrator tasks but don’t have those permissions all the time. If an eligible user needs to perform an administrator task, then they have to request activation of an elevated role. At this point, several different things could happen depending on how you configure it. First, the user might have to perform multifactor authentication. Then they have to enter a reason for the activation request. Finally, if the role requires activation approval, then they’ll have to wait for an approver to activate their role. This activation will only last for a limited amount of time, so after the activation expires, they will have to go through the same process again.

**External Identities**

Azure AD has a feature called External Identities that takes care of people outside your organization. It’s commonly used for working with partners, suppliers, and customers. It allows external users to "bring their own identities". They log into a separate identity provider first and can then gain guest access to some of your applications. This separate identity provider could be their organization’s own identity system or a social platform such as Facebook or Google.The External Identities feature includes three components: B2B collaboration, B2B direct connect, and Azure AD B2C. With B2B (or business-to-business) collaboration, external users are represented in your Azure AD directory as guest users. With B2B direct connect, you establish a mutual trust relationship with another Azure AD organization. This means external users are not represented in your directory because your directory trusts identities in their directory. B2B direct connect currently supports only Microsoft Teams shared channels. With Azure AD B2C (or business-to-consumer), you can publish your own consumer-facing applications and give your customers access to them. Azure AD B2C is actually a separate service that’s built on the same technology as Azure AD

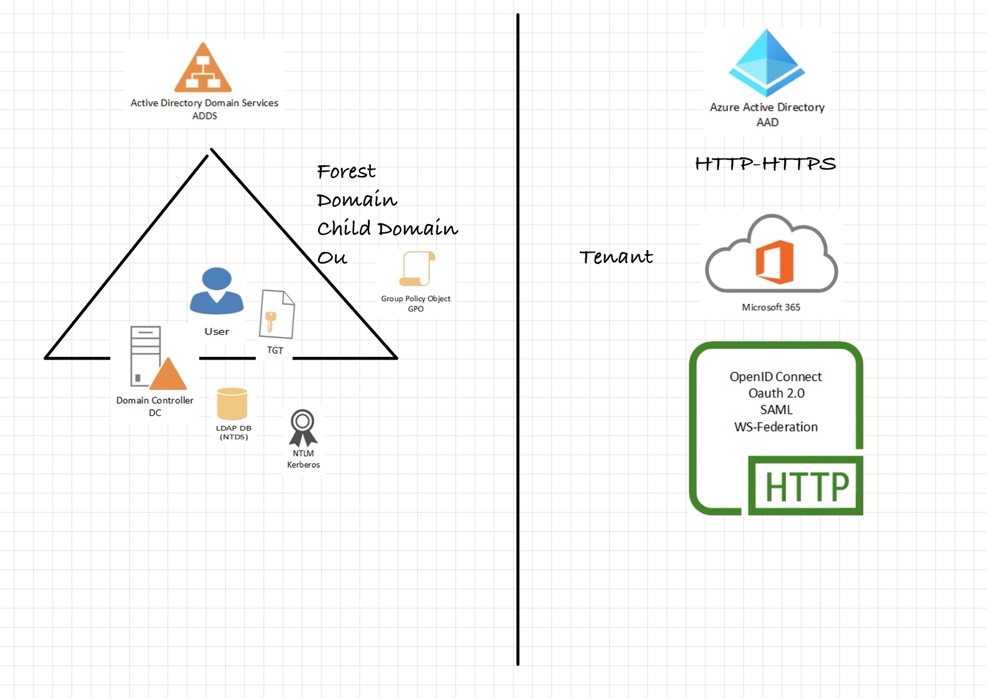
**Self-Service Support**

Another cool feature provided by Azure AD is Self-service support. Azure AD lets you to delegate tasks to company employees that might otherwise be completed by admins with higher access privileges.

Describe Azure AD concepts.

There are some concepts you need to make your own for the exam. **Identity:** An identity is an object that can be authenticated. The identity can be a user with username and password, but can be also applications or other servers that require authentication by using secret keys or certificates. Azure AD is the underlying product that provides the identity services. **Account:** An account is an identity that has data associated with it. You can’t have an account without an identity. **Azure AD account:** An Azure AD account is an identity that’s created through Azure AD or another Microsoft cloud service, such as MS 365. Identities are stored in Azure AD and are accessible to your organization’s cloud service subscriptions. The Azure AD account is also called a work or school account. **Azure Tenant (directory)**: An Azure tenant is a single dedicated and trusted instance of Azure AD. Each tenant (also called a directory) represents a single organization. When your organization signs up for a Microsoft cloud service subscription, a new tenant is automatically created. **Azure subscription**: An Azure subscription is used to pay for Azure cloud services. A subscription is linked to a credit card. Each subscription is joined to a single tenant. You can have multiple subscriptions.

Compare Active Directory Domain Services to Azure Active Directory

Microsoft has spent resources into making AD and Azure AD work together. The concept is to extend the identity that lives on-premises to the cloud by synchronizing the identities. We saw the Azure AD Connect before. Microsoft has also invested in extending those identities to enable scenarios such as single sign-on by using Active Directory Federation Services (ADFS), which is deployed in many large enterprises. **WHITEBOARD**: AD DS is the traditional deployment of Windows Server-based AD on a physical or virtual server. AD DS is commonly considered to be primarily a directory service, but it’s only one component of the Windows AD suite of technologies. The suite also includes Active Directory Certificate Services (AD CS), Active Directory Lightweight Directory Services (AD LS), Active Directory Federation Services (AD FS) and Active Directory Rights Management Services (AD RMS).

Azure AD is similar to AD DS but there are significant differences: AD DS is primarily a directory service, while Azure AD is a full **identity solution**. Azure AD is designed for internet-based applications that use HTTP and HTTPS communications. Azure AD is based on HTTP and HTTPS protocols. Azure AD tenants cannot be queried by using LDAP. Azure AD uses the REST API over HTTP and HTTPS, and because on HTTP and HTTPS, it doesn’t use Kerberos Authentication. It uses protocols like SAML (Security Assertion Markup Language, an XML-based open-standard for transferring identity data between two parties: an identity provider and a service provider), WS-Federation (Web-Service Federation is a protocol that can be used to negotiate the issuance of a token for your applications such as Windows identity Foundation-based app and for identity providers such as AD FS and Open-ID Connect for Authentication (that is an identity layer built on top of the OAuth 2.0 framework that allows third-party applications to verify the identity of the end-user and to obtain basic user profile information using JSON web tokens JWTs) and OAuth (Open Authorization) for authorization. Azure AD includes **federation services**, and many third-party services like Facebook or Google as said before with B2b and B2C features. Another difference is that Azure AD users and Groups are created in a flat structure. There are no organizational units (OUs) or Group Policy objects (GPOs). Azure AD is a **managed service**. You manage only users, groups and policies. If you deploy AD DS with virtual machines by using Azure, you manage many other tasks, including deployment, configuration, virtual machines, patching and other backend processes.

Select Azure Active Directory Editions

Azure AD comes in 4 editions: Free, Microsoft 365 Apps, Premium 1 and Premium 2. The Free Edition is included with an Azure subscription. The Premium editions are available through a Microsoft Enterprise Agreement, the Open Volume License Program, and the Cloud Solution Providers program. Azure and Microsoft 365 subscribers can also buy Azure AD Premium P1 and P2 Online. Azure AD Free provides user and group management, on-prem directory synchronization and basic reports. Azure AD Microsoft 365 apps is included with Microsoft 365 and provides Identity and Access management for Microsoft 365 apps including branding, MFA, group access management and self-service password reset for cloud users. Azure AD Premium P1 lets your hybrid user access both on-prem and cloud resources. This edition supports advanced administration like dynamic groups, self-service group management and cloud write-back capabilities. P1 also includes Microsoft Identity Manager, an on-prem identity and access management suite. The extra feature in P1 allow self-service password reset for your on-prem users. Azure AD Premium P2, in addition to Free and P1 features, offers Azure AD Identity protection to help provide risk-based Conditional Access to your apps and critical company data. Privileged Identity Management is included to help discover, restrict and monitor administrators and their access to resources and to provide just-in-time access when needed. Prices are for Azure AD Premium P1 6 USD user/month with annual commitment and for Azure AD Premium P2 9 USD user/month with annual commitment

Configure Azure AD Device Identities

Azure Active Directory includes the ability to manage device identity, which enables single sign-in to devices and the applications and services managed through Azure Active Directory that are accessed from that device. Managed devices include both enterprise and bring-your-own-device (BYOD) scenarios. This allows users to work from any device, including personal devices, all while protecting corporate intellectual property with the necessary regulatory and compliance controls. When associating devices with Azure AD, you have three options:

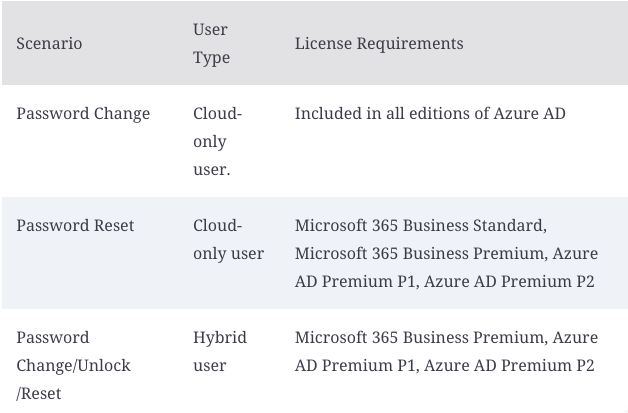
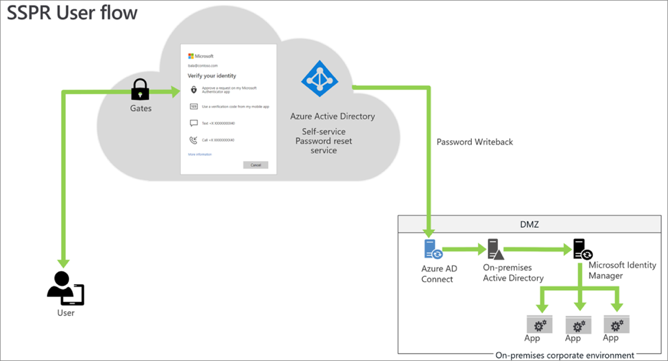
**Azure AD Registered Devices**. Registration of devices would be appropriate for personal devices (BYOD scenario). Azure AD registered devices are signed in to using a local account like a Microsoft account on a Windows 10 or newer device. These devices have an Azure AD account for access to organizational resources. Access to resources in the organization can be limited based on that azure AD account and Conditional Access policies applied to the device identity. Azure AD registration can be accomplished when accessing a work application for the first time or manually using Windows 10 or Windows 11 Settings menu.

**Azure AD Joined Devices.** The Azure AD join feature works with SSO to provide access to organizational apps and resources, and to simplify Windows Deployments of work-owned devices. Your users won’t have extra authentication prompts when they access work resources. The SSO functionality is available even when users aren’t connected to the domain network. Starting in Windows 10, your user can securely synchronize their user settings and app settings data to joined devices with a feature known as **Enterprise state roaming** that reduces the time to configure a new device. When your users access **Microsoft Store for Business** by using an Azure AD account, they can choose from an inventory of applications pre-selected by your organization. **Windows Hello** provides your users with secure and convenient access to work resources from joined devices. Another benefit is the **restriction of access** Restrict user access to apps from only joined devices that meet your compliance policies. Another feature is that Joined devices have **seamless access to on-prem resources** when the device has line of sight to the on-prem domain controller, so they can still authenticate to on-prem servers like file, print and other applications. Administrators can secure and control these devices using MDM tool like Intune or in co-management scenarios using Microsoft Endpoint Configuration Manager. Consider AD join scenarios like: 1) You want to transition to cloud-based infrastructure using Azure AD and MDM like Intune. 2) You can’t use an on-prem domain join, for example, if you need to get mobile devices such as tablets and phones under control. 3) Your users primarily need to access Microsoft 365 or other SaaS apps integrated with Azure AD. 4) You want to manage a group of users in Azure AD instead of in Active Directory. This scenario can apply, for example, to seasonal workers, contractors or students. 5) You want to provide joining capabilities to workers who work from home or are in remote branch offices with limited on-prem infrastructure. You can configure Azure AD join for all Windows 11 and Windows 10 devices except for home editions.

**Hybrid Azure AD joined Devices** – Organizations with existing Active directory implementations can benefit from some of the functionality provided by Azure AD by implementing hybrid Azure AD joined devices. These devices are joined to your on-premises and registered with Azure AD. Hybrid Azure Ad joined devices require network line of sight to your on-prem domain controllers periodically. Without this connection, devices become unusable. Consider Azure AD hybrid joined if: 1) You support down-level devices running 8.1. 2) You want to continue to use Group Policy to manage device configuration. 3) You want to continue to use existing imaging solutions to deploy and configure devices. 4) You have Win32 apps deployed to these devices that rely on Active Directory machine authentication.

Implement Azure Active Directory Self-Service Password reset

Self Service Password Reset is an Azure AD feature that enables users to reset their password without contacting IT staff for help. The user can quickly unblock themselves and continue working no matter where they are or time of day. By allowing the employees to unblock themselves, your organization can reduce the non-productive time and high support costs for most common password-related issues. So, we can say that manage cost is one of the key benefits of enabling SSPR. The other benefits are: Intuitive user experience with one-time user registration process that allows users to reset passwords and unblock accounts on-demand from any device or location. It provides also Flexibility and security because Administrators can change settings to accommodate new security requirements and roll these changes out to users without disrupting their sign-in. At least it provides also robust auditing and usage tracking including information of each step of the password reset process. These logs are available from an API and enable the user to import the data into a Security Incident and Event Monitoring (SIEM) system of choice (like Splunk in our case). There are basically 3 requirements for SSPR feature: 1) SSPR requires an Azure AD account with Global Administrator privileges to manage SSPR options. These accounts can always reset their own passwords, no matter what options are configured. 2) SSPR uses a security group to limit the users who have SSPR privileges. 3) All user accounts in your organization must have a valid license to use SSPR and the license required.



**Consider who can reset their password.** Decide which users in your organization should be able to use the feature. In Azure Portal, there are three options for the SSPR feature: None, Selected and All. The Selected option is useful for creating specific groups who have SSPR enabled. You can create groups for testing or proof of concept before applying the feature to a larger group. **Consider your authentication method.** Determine how many authentication methods are required to reset a password and select the authentication options for users: 1) Your system must require at least one authentication method to reset a password. 2) A strong SSPR plan offers multiple authentication methods for the users. Options include email notification, text message or security code sent to the user’s mobile or office phone. You can also offer the user a set of security questions. 3) You can require security questions to be registered for the users in your Azure AD tenant. 4) You can configure how many correctly answered questions are required for a successful password reset. **Consider combining methods for stronger security.** Security questions can be less secure than other authentication methods. Some users might know the answer for a particular user’s question or the questions might be easy to solve.

Summary and Resources – Configure Azure Active Directory

To keep you awake and active, I invite you to answer these questions. Don’t be shy, it serves as a review for everyone to digest what we have said so far. Those who answer will be entitled to a double coffee.

1. Describe the following concepts: identity, account, Azure AD account, Azure AD Account, Azure AD tenant, and Azure subscription. How are these different?

**Answer:** Identity is an object that can be authenticated. An Account is an identity that has data associated with it. An Azure AD account is an identity created through Azure AD or another Microsoft cloud service. An Azure AD tenant is a dedicated and trusted instance of Azure AD, A Tenant is automatically created when your organization signs up for a Microsoft cloud service subscription​. An Azure subscription is used to pay for Azure cloud services**.**

1. How is Azure Active Directory different from Azure Active Directory Domain Services?

**Answer:** Azure AD is primarily an identity solution and designed for HTTP and HTTPS communications. Azure AD can be queried with a REST API, instead of LDAP. Azure AD uses federation services, and many third-party services (such as Facebook). Azure AD users and groups are created in a flat structure. Azure AD does not have Organizational Units (OUs) or Group Policy Objects (GPOs).

1. Describe Azure AD Join and the usage cases for it.

**Answer:** Azure AD Join provides single sign on to your Azure managed SaaS apps and services. Joined devices have enterprise state roaming of user settings. AD Join provides seamless access to on-premises resources. Restricts access to apps from only compliant devices.

1. Describe the Self-Service Password Reset authentication methods can be configured for users.

**Answer:** Self-Service Password Reset authentication methods include mobile app notification, mobile app code, email, mobile phone, office phone, and security questions. A combination of authentication methods can be used.

## Configure Users and Groups

Create User Accounts

There are primarily three types of users in Azure AD – cloud-only users, external guest users and users synchronized from an on-premises directory. Cloud-only users are created and managed exclusively in Azure AD, and their attributes can be updated directly in Azure AD. You can create cloud-only users through the Azure portal, Azure PowerShell, and the Azure command-line interface (CLI). When creating new users, you must be assigned to the Global Administrator or User Administrator role. We will look at Role Based Access Control later on. The administrator can create a user within the organization or invite a guest user to provide access to organization resources: A new user account must have a display name and an associated user account name. Information and settings that describe a user are stored in the user account profile and it can have other settings like a user’s job title, and their contact email address. A user with Global administrator or User administrator privileges can pre-set profile data in user accounts, such as the main phone number for the company.

Manage User Accounts

Non-admin users can set some of their own profile data, but they cannot change their display name or account name. **Consider restore options for deleted accounts.** Include restore scenarios in your account management plan. Restore operations for a deleted account are available up to 30 days after an account is removed. After 30 days, a deleted user account can’t be restored. **Consider gathered account data.** Collect sign-in and audit log information for user accounts. Azure AD lets you gather this data to help you analyse and improve your infrastructure.

**Demo Creating User Account**

Azure Portal

**AZ Cli:**

*az ad user create –display-name xxxx –password xxxx –user-principal-name* [*xxx@techints.com*](mailto:xxx@techints.com)*)*

**PowerShell:**

*$PasswordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile*

*$PasswordProfile.Password = “PassWord123!”;*

*New-AzureADUser -DisplayName “xxxx” -PasswordProfile $PasswordProfile -UserPrincipalName* [*xxx@techints.com*](mailto:xxx@techints.com) *-AccountEnabled $true -MailNickname “xxxx” -ForceChangePasswordNextLogin*

**PowerShell on MacOS (version 7.3):**

*$password = ConvertTo-SecureString -String “PassW0rd123!” -AsPlainText -Force*

*New-AzADUser -DisplayName “xxxx\* -UserPrincipalName* [*xxx@techints.com*](mailto:xxx@techints.com) *-password $password -MailNickName “xxxx” -ForceChangePasswordNextLogin*

Perform bulk account updates.

Previously, the Azure portal was only helpful for single updates to users, which meant we had to rely on custom automation solutions (mostly using PowerShell) for updating users in bulk. Because of recent updates, you can now perform bulk operations (such as creating, inviting, and deleting users in batches) using the Azure portal. Azure PowerShell can still be used for bulk upload of user accounts. Bulk operations are three-step process: 1) Download a CSV (comma-separated values or comma-delimited) template (UserCreateTemplate**.csv**) by clicking the **Download** button on the **Bulk Create User** blade. This is a standard template with mandatory attributes, such as **Name**, **User Name**, **Initial Password**, and **Block Sign In**. You can also specify optional attributes such as **First Name**, **Last Name**, **Job Title**, and so on. 2) Edit the CSV file with bulk update values. You just need to update appropriate values and save the changes. The sample mandatory values are already included in the template for reference. 3) Upload the updated CSV file and submit the operation. Only Global Administrators or User administrators have privileges to create and delete user accounts in the Azure Portal

**Things to consider when creating user accounts**

There are some design considerations for creating and deleting user accounts. Think about what user account conventions and processes might be required by your organization. **Consider naming conventions.** Establish or implement a naming convention for your user accounts. Apply conventions to user account names, display names, and user aliases for consistency across the organization. Conventions for names and aliases can simplify the bulk create process by reducing areas of uniqueness in the CSV file. A convention for user names could begin with the user’s first name followed by a period, and end with the user’s last name, but maybe for security reasons, it would be better to abstract the username from the proper name, using random patterns. **Consider using initial passwords.** Implement a convention for the initial password of a newly created user. Design a system to notify new users about their passwords in a secure way. You may generate a random password and find a secure method to provide. **Consider strategies for minimizing errors.** View and address any errors, by downloading the results file on the Bulk operation results page in the Azure Portal. The results file contains the reason for each error. An error might be a user account that’s already been created or an account that’s duplicated. Generally, it’s easier to upload and troubleshoot smaller groups of user accounts.

Create group accounts

Groups are collections of objects that make role assignments and access permissions easier to manage. A group can contain users, groups, devices or service principles. When using groups, you eliminate the need to individually assign roles or permissions. Creating groups is a similar experience and can be performed from the Azure portal, Azure PowerShell, Azure CLI and Microsoft Graph. When creating a group, there are several factors that dictate the type of group that is created and how that group behaves in Azure AD and associated workloads such as Microsoft 365. Azure AD allows your organization to define two different types of group accounts. Security Groups are used to manage member and computer access to shared resources for a group of users. Microsoft 365 groups provide collaboration opportunities. Group members have access to a shared mailbox, calendar, files, SharePoint site and more. When you add members to a group, there are different ways you can assign member access rights. As you read through these options, consider which groups are needed to support your organization, and what access rights should be applied to group members. We can have three options: 1) **Assigned.** This value allows you to select one or more users and add them to the group. Adding and removing users is performed manually. 2) **Dynamic User.** This value allows you to use dynamic group rules to automatically add and remove members. 3) **Dynamic devices.** This value allows you to use dynamic group rules to automatically add and remove devices. You can only create dynamic group if you have a Premium AD license. Otherwise, the **Membership type** option is unavailable and is set to **Assigned**. For both dynamic user and dynamic device-based groups, the rules associated with the group are evaluated on an ongoing basis. If a user or device has an attribute that matches the rule, that user or device is added to the group. If an attribute changes and the user or device no longer matches the criteria for group membership, the entity will be removed. Membership processing is not immediate. If an error occurs while processing a membership rule, an error is surfaced on the Group page in the Azure portal. You can always view the current processing status from the Group page. It is important to note that you can create a dynamic group for users or devices, but you cannot create both at the same time. You also cannot use user attributes in a device-based rule. It is possible to change the membership type of a group after it has been created, which provides an opportunity to transition from a static (or assigned) membership model to a dynamic membership model or vice-versa. When creating dynamic groups, rules can be edited in the simple rule format, where you will build the query and conditions in the rule builder, where you can build complex rules with conditional logic.

Assign Licenses to Users and Groups

<https://www.microsoft.com/security/business/identity-access-management/azure-ad-pricing?rtc=1#office-SKUChooser-q6q98uk>

Create Administrative Units

Planning for comprehensive management of your Azure Active Directory infrastructure is critical. It can be useful to restrict administrative scope by using administrative units for your organization. The division of roles and responsibilities is especially helpful for organizations that have many independent divisions. Consider, as an example, the management tasks for a large university that's composed of several different schools like Business, Engineering, and Medicine. The university has administrative offices, academic buildings, social buildings, and student dormitories. For security purposes, each business office has its own internal network for resources like servers, printers, and fax machines. Each academic building is connected to the university network, so both teachers and students can access their accounts. The network is also available to students and deans in the dormitories and social buildings. Across the university, guest users require access to the internet via the university network. The university has a team of IT admins who work together to control resource access, manage users, and set policies for the school. Some admins have greater privileges than others depending on the scope of their responsibilities. A central authority is needed to plan, manage, and oversee the complete structure. In this scenario, you can assign administrative units to make it easier to manage the organization. Consider how a central admin role can use administrative units to support the Engineering department in our scenario: 1) Create an administrative unit with only the Engineering department. 2) Populate the administrative unit with only the engineering department students, staff and resources. 3) Create a role that has administrative permissions for only Azure AD users in the Engineering department administrative unit. 4) Add the Engineering department IT team to the role, along with its scope.

Demonstration – Users and Groups