**2.1 - Understand Microsoft Entra ID**

**Examine Microsoft Entra ID**

**Active Directory Domain Services (AD DS or traditionally called just "Active Directory")** is a directory service that provides the methods for storing directory data, such as user accounts and passwords, and makes this data available to network users, administrators, and other devices and services. It runs as a service on Windows Server, referred to as a domain controller.

**Microsoft Entra ID** is part of the platform as a service (PaaS) offering and operates as a Microsoft-managed directory service in the cloud. It’s not a part of the core infrastructure that customers own and manage, nor is it an Infrastructure as a service offering. While this implies that you have less control over its implementation, it also means that you don’t have to dedicate resources to its deployment or maintenance.

With Microsoft Entra ID, you also have access to a set of features that aren’t natively available in AD DS, such as support for multi-factor authentication, identity protection, and self-service password reset.

Microsoft Entra constitutes a separate Azure service. Its most elementary form, which any new **Azure subscription includes automatically, doesn't incur any extra cost and is referred to as the Free tier**. If you subscribe to any Microsoft Online business services (for example, Microsoft 365 or Microsoft Intune), you **automatically get Microsoft Entra ID with access to all the Free features**.

**Microsoft Entra schema**

The Microsoft Entra schema contains fewer object types than that of AD DS. Most notably, it doesn't include a definition of the computer class, although it does include the device class.

Microsoft Entra ID doesn't include the organizational unit (OU) class, which means that **you can't arrange its objects into a hierarchy of custom containers**, which is frequently used in on-premises AD DS deployments.

Objects of the Application and servicePrincipal classes represent applications in Microsoft Entra ID. **An object in the Application class contains an application definition** and an **object in the servicePrincipal class constitutes its instance in the current Microsoft Entra tenant**. Separating these two sets of characteristics allows you to define an application in one tenant and use it across multiple tenants by creating a service principal object for this application in each tenant. **Microsoft Entra ID creates the service principal object when you register the corresponding application in that Microsoft Entra tenant**.

**Microsoft Entra tenants**

Unlike AD DS, Microsoft Entra ID is multi-tenant by design and is implemented specifically to ensure isolation between its individual directory instances.

The term tenant in this context typically represents a company or organization that signed up for a subscription to a Microsoft cloud-based service such as Microsoft 365, Intune, or Azure, each of which uses Microsoft Entra ID. However, from a technical standpoint, the term **tenant represents an individual Microsoft Entra instance**. Within an Azure subscription, you can create multiple Microsoft Entra tenants. Having multiple Microsoft Entra tenants might be convenient if you want to test Microsoft Entra functionality in one tenant without affecting the others.

**Azure subscription must be associated with one, and only one, Microsoft Entra tenant.** This association allows you to grant permissions to resources in the Azure subscription (via RBAC) to users, groups, and applications that exist in that particular Microsoft Entra tenant.

You can **associate the same Microsoft Entra tenant with multiple Azure subscriptions**. This allows you to use the same users, groups, and applications to manage resources across multiple Azure subscriptions.

Each Microsoft Entra tenant is assigned the **default Domain Name System (DNS)** domain name, consisting of a unique prefix. The prefix, derived from the name of the Microsoft account you use to create an Azure subscription or provided explicitly when creating a Microsoft Entra tenant, is **followed by the onmicrosoft.com suffix**. Adding **at least one custom domain name** to the same Microsoft Entra tenant is possible and common. This name utilizes the DNS domain namespace that the corresponding company or organization owns.

**Characteristics of AD DS**

AD DS is the traditional deployment of Windows Server-based Active Directory on a physical or virtual server. Although AD DS is commonly considered being primarily a directory service, it’s only one component of the Windows Active Directory suite of technologies, which also includes Active Directory Certificate Services (AD CS), Active Directory Lightweight Directory Services (AD LDS), Active Directory Federation Services (AD FS), and Active Directory Rights Management Services (AD RMS).

When comparing AD DS with Microsoft Entra ID, it’s important to note the following characteristics of AD DS:

* AD DS is a true directory service, with a hierarchical X.500-based structure.
* AD DS uses Domain Name System (DNS) for locating resources such as domain controllers.
* You can query and manage AD DS by using Lightweight Directory Access Protocol (LDAP) calls.
* AD DS primarily uses the Kerberos protocol for authentication.
* AD DS uses OUs and GPOs for management.
* AD DS includes computer objects, representing computers that join an Active Directory domain.
* AD DS uses trusts between domains for delegated management.

You **can deploy AD DS on an Azure virtual machine** to enable scalability and availability for an on-premises AD DS. However, deploying AD DS on an Azure virtual machine doesn't make any use of Microsoft Entra ID.

**Deploying AD DS on an Azure virtual machine requires one or more extra Azure data disks because you shouldn't use drive C for AD DS storage.** These disks are needed to store the AD DS database, logs, and the sysvol folder. The Host Cache Preference setting for these disks must be set to None.

**Characteristics of Microsoft Entra ID**

Although Microsoft Entra ID has many similarities to AD DS, there are also many differences. It’s important to realize that using Microsoft Entra isn’t the same as deploying an Active Directory domain controller on an Azure virtual machine and adding it to your on-premises domain.

When comparing Microsoft Entra ID with AD DS, it’s important to note the following characteristics of Microsoft Entra ID:

* Microsoft Entra ID is primarily an identity solution, and it’s designed for internet-based applications by using HTTP (port 80) and HTTPS (port 443) communications.
* Microsoft Entra ID is a multi-tenant directory service.
* Microsoft Entra users and groups are created in a flat structure, and there are no OUs or GPOs.
* You can't query Microsoft Entra ID by using LDAP; instead, Microsoft Entra ID uses the REST API over HTTP and HTTPS.
* Microsoft Entra ID doesn't use Kerberos authentication; instead, it uses HTTP and HTTPS protocols such as SAML, WS-Federation, and OpenID Connect for authentication, and uses OAuth for authorization.
* Microsoft Entra ID includes federation services, and many third-party services such as Facebook are federated with and trust Microsoft Entra ID.

**Examine Microsoft Entra ID as a directory service for cloud apps**

When you deploy cloud services such as Microsoft 365 or Intune, you also need to have directory services in the cloud to provide authentication and authorization for these services. Because of this, each cloud service that needs authentication will create its own Microsoft Entra tenant. When a single organization uses more than one cloud service, it’s much more convenient for these cloud services to use a single cloud directory instead of having separate directories for each service.

It’s now possible to have one identity service that covers all Microsoft cloud-based services, such as Microsoft 365, Azure, Microsoft Dynamics 365, and Intune. Microsoft Entra ID provides developers with centralized authentication and authorization for applications in Azure by using other identity providers or on-premises AD DS. Microsoft Entra ID can provide users with an SSO experience when using applications such as Facebook, Google services, Yahoo, or Microsoft cloud services.

In particular, you can enable Microsoft Entra authentication for the Web Apps feature of Azure App Service directly from the Authentication/Authorization blade in the Azure portal. By designating the Microsoft Entra tenant, you can ensure that only users with accounts in that directory can access the website. It’s possible to apply different authentication settings to individual deployment slots.

**Compare Microsoft Entra ID P1 and P2 plans**

The Microsoft Entra ID P1 or P2 tier provides extra functionality as compared to the Free and Office 365 editions. However, premium versions require additional cost per user provisioning. Microsoft Entra ID P1 or P2 comes in two versions P1 and P2. You can procure it as an extra license or as a part of the Microsoft Enterprise Mobility + Security, which also includes the license for Azure Information Protection and Intune.

Microsoft provides a free trial period that can be used to experience the full functionality of the Microsoft Entra ID P2 edition.

The following features are available with the **Microsoft Entra ID P1** edition:

* + Self-service group management.
  + Advanced security reports and alerts
  + Multi-factor authentication
  + Microsoft Identity Manager (MIM) licensing
  + Enterprise SLA of 99.9%
  + Password reset with writeback
  + Cloud App Discovery feature of Microsoft Entra ID
  + Conditional Access based on device, group, or location
  + Microsoft Entra Connect Health

In addition to these features, the **Microsoft Entra ID P2 license** provides extra functionalities:

* Microsoft Entra ID Protection
* Microsoft Entra Privileged Identity Management

**2.2 - Create, configure, and manage identities**

Access to cloud-based workloads is controlled centrally in two ways. First by providing a definitive identity for each user that they use for every service. Then second by ensuring employees and vendors have enough access to do their jobs.

**Create, configure, and manage users**

Every user who needs access to Azure resources needs an Azure user account. A user account contains all the information needed to authenticate the user during the sign-on process. Once authenticated Microsoft Entra ID builds an access token to authorize the user and determine what resources they can access and what they can do with those resources.

You use the **Microsoft Entra ID** dashboard in the Azure portal to work with user objects. Keep in mind that you can only work with a single directory at a time. You can use the **Directory + Subscription** panel to switch directories. The dashboard also has a **Switch directory** button in the toolbar which makes it easy to switch to another available directory.

**View users**

To view the Microsoft Entra users, select the Users entry under Identity - then open the All Users view. Notice the **User Type column to see members and guests.**

**Typically, Microsoft Entra ID defines users in three ways:**

**Cloud identities** - These users exist only in Microsoft Entra ID. Examples are administrator accounts and users that you manage yourself. Their source is Microsoft Entra ID or External Microsoft Entra directory if the user is defined in another Microsoft Entra instance but needs access to subscription resources controlled by this directory. When these accounts are removed from the primary directory, they're deleted.

**Directory-synchronized identities** - These users exist in an on-premises Active Directory. A synchronization activity that occurs via Microsoft Entra Connect brings these users in to Azure. Their source is Windows Server AD.

**Guest users** - These users exist outside Azure. Examples are accounts from other cloud providers and Microsoft accounts such as an Xbox LIVE account. Their source is Invited user. This type of account is useful when external vendors or contractors need access to your Azure resources. Once their help is no longer necessary, you can remove the account and all of their access.

**Restore or remove a recently deleted user with Microsoft Entra ID**

After you delete a user, the account remains in a suspended state for 30 days. During that 30-day window, the user account can be restored, along with all its properties. After that 30-day window passes, the permanent deletion process is automatically started.

You can view your restorable users, restore a deleted user, or permanently delete a user using Microsoft Entra ID user interface.

**You can't restore a permanently deleted user.**

You must have one of the following **roles to restore or permanently delete** users.

* Global administrator
* Partner Tier-1 Support
* Partner Tier-2 Support
* User administrator

**Create, configure, and manage groups**

A Microsoft Entra group helps organize users, which makes it easier to manage permissions. Using groups lets the resource owner (or Microsoft Entra directory owner), assign a set of access permissions to all the members of the group, instead of having to provide the rights one-by-one. Groups allow us to define a security boundary and then add and remove specific users to grant or deny access with a minimum amount of effort. Even better, Microsoft Entra ID supports the ability to define membership based on rules - such as what department a user works in, or the job title they have.

Microsoft Entra ID allows you to define two different types of groups.

* **Security groups** - the most common type of groups and are used to manage member and computer access to shared resources for a group of users. For example, you can create a security group for a specific security policy. By doing it this way, you can give a set of permissions to all the members at once, instead of having to add permissions to each member individually. This option requires a Microsoft Entra administrator.
* **Microsoft 365 groups** - provide collaboration opportunities by giving members access to a shared mailbox, calendar, files, SharePoint site, and more. This option also lets you give people outside of your organization access to the group. This option is available to users as well as admins.

**View available groups**

You can view all groups through the Groups item under **Identity** in the **Microsoft Entra admin center**. A new Microsoft Entra ID deployment won't have any groups defined.

The second characteristic of a group that you need to be aware of is the **Membership Type**. This specifies **how individuals members are added to the group**. The two types are:

* **Assigned** - members are added and maintained manually.
* **Dynamic** - members are added based on rules, creating a Dynamic Group. These groups are still either a security group or Microsoft 365 group, just their members are controlled by rule.

**Dynamic groups**

The final type of group is a dynamic group, which the name implies, the membership is generated by a formula each time the group is used. A dynamic group includes any recipient in Active Directory with attribute values that match its filter. If a recipient's properties are modified to match the filter, the recipient could inadvertently become a group member and start receiving messages that are sent to the group. Well-defined, consistent account provisioning processes will reduce the chances of this issue occurring.

**2.3 - Describe the core architectural components of Azure**

**Azure physical infrastructure**

**Physical infrastructure**

The physical infrastructure for Azure starts with datacenters. Conceptually, the datacenters are the same as large corporate datacenters. They’re facilities with resources arranged in racks, with dedicated power, cooling, and networking infrastructure.

As a global cloud provider, Azure has datacenters around the world. However, these individual datacenters aren’t directly accessible. Datacenters are grouped into Azure Regions or Azure Availability Zones that are designed to help you achieve resiliency and reliability for your business-critical workloads.

**Regions**

A region is a geographical area on the planet that contains at least one, but potentially multiple datacenters that are nearby and networked together with a low-latency network. Azure intelligently assigns and controls the resources within each region to ensure workloads are appropriately balanced.

There are also some global Azure services that don't require you to select a particular region, such as Microsoft Entra ID, Azure Traffic Manager, and Azure DNS.

**Availability Zones**

Availability zones are physically separate datacenters within an Azure region. Each availability zone is made up of one or more datacenters equipped within dependent power, cooling, and networking. An availability zone is set up to be an isolation boundary. If one zone goes down, the other continues working. Availability zones are connected through high-speed, private fiber-optic networks.

To ensure resiliency, a minimum of three separate availability zones are present in all availability zone-enabled regions. However, not all Azure Regions currently support availability zones.

Availability zones are primarily for VMs, managed disks, load balancers, and SQL databases. Azure services that support availability zones fall into three categories:

**Zonal services**: You pin the resource to a specific zone (for example, VMs, managed disks, IP addresses).

**Zone-redundant services**: The platform replicates automatically across zones (for example, zone-redundant storage, SQL Database).

**Non-regional services**: Services are always available from Azure geographies and are resilient to zone-wide outages as well as region-wide outages.

**Region pairs**

Most Azure regions are paired with another region within the same geography (such as US, Europe, or Asia) at least 300 miles away. This approach allows for the replication of resources across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages that affect an entire region. For example, if a region in a pair was affected by a natural disaster, services would automatically fail over to the other region in its region pair.

Not all Azure services automatically replicate data or automatically fall back from a failed region to cross-replicate to another enabled region. In these scenarios, recovery and replication must be configured by the customer.

Most regions are paired in two directions, meaning they are the backup for the region that provides a backup for them (West US and East US back each other up). However, some regions, such as West India and Brazil South, are paired in only one direction. In a one-direction pairing, the Primary region does not provide backup for its secondary region. So, even though West India’s secondary region is South India, South India does not rely on West India. West India's secondary region is South India, but South India's secondary region is Central India. Brazil South is unique because it's paired with a region outside of its geography. Brazil South's secondary region is South Central US. The secondary region of South Central US isn't Brazil South.

**Sovereign Regions**

In addition to regular regions, Azure also has sovereign regions. Sovereign regions are instances of Azure that are isolated from the main instance of Azure. You may need to use a sovereign region for compliance or legal purposes.

Azure sovereign regions include:

USD oD Central, US Gov Virginia, US Gov Iowa and more: These regions are physical and logical network-isolated instances of Azure for U.S. government agencies and partners. These datacenters are operated by screened U.S. personnel and include additional compliance certifications.

China East, China North, and more: These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft doesn't directly maintain the datacenters.

**2.4 - Azure Policy initiatives**

**Operation flows of Azure Resource Manager**

Azure Resource Manager includes two scenarios for handling Azure requests: Greenfield and Brownfield. As you deploy resources, Azure Resource Manager understands when to create new resources and when to update existing resources.

**Greenfield** refers to a scenario where an Azure Policy (policy-first) exists, and when you're creating or updating an Azure resource.

**Brownfield** is the scenario where the resources exist already (resource-first), and you're assigning a new Azure Policy to those resources.

**Azure Policy definitions**

Azure Policy definition describes resource compliance conditions and the action or effects that take place if those conditions are met. The policy consists of two parts:

A condition that compares a resource property field or a value, accessed by using aliases, to a required value.

The effect determines what happens when the policy rule is evaluated to match the condition. For each new resource, an updated resource, or an existing resource, the effects behave differently.

**Azure Policy resources**

Azure Policy enforces organizational standards and assesses compliance at scale. It evaluates Azure resources and actions by comparing their properties to business rules, providing an aggregated view of the environment's overall state. This policy allows for detailed analysis down to each resource and policy level with granularity. **Six policy resources** are available in Azure, and multiple different concepts apply to these Azure policy resources.

**Definitions** - Azure Policy definitions describe resource compliance conditions and the effect to take if a condition is met. Several settings determine which resources are evaluated by any Azure Policy.

**Initiatives** - Azure Policy initiatives, also known as a policy set, allow you to group several policy definitions to simplify assignments and management because you work with the initiatives as a single item. Initiatives offer a streamlined and automated approach to governance, allowing organizations to manage and monitor compliance at scale.

**Assignments** - Policy assignments define which resources are evaluated by a policy definition or initiative. Policy assignments can be done in the portal, an API call, or through the command line interface.

**Exemptions**

Use the Policy exemptions feature to exempt a resource hierarchy or an individual resource from evaluation of initiatives or definitions. Resources that are exempt count toward overall compliance but can't be evaluated or have a temporary waiver. They're created as a child object on the resource hierarchy, or the individual resource granted the exemption.

Policy exemptions aren't created during assignment time, but after, and the effect is still the same as an excluded scope.

**Attestations** - Policy attestations are used by Azure Policy to set compliance states of resources or scopes targeted by manual policies. Each applicable resource requires one attestation for each manual policy assignment. For ease of management, manual policies should be designed to target the scope that defines the boundary of resources whose compliance state needs to be attested.

**Remediations** - The policy remediation task feature is used to bring resources into compliance based on a definition and assignment. Resources that are noncompliant to a modify or deployIfNotExists definition assignment can be brought into compliance by using a remediation task. Resources that are newly created or updated that are applicable to a deployIfNotExists or modify definition assignment are automatically remediated.

**Evaluation of resources through Azure Policy**

A significant benefit of Azure Policy is the insight and controls that it provides over resources in a subscription or management group of subscriptions. This control can be used to prevent resources from being created in the wrong location, enforce common and consistent tag usage, or audit existing resources for appropriate configurations and settings. Before you review compliance data and react to it, you need to understand the evaluation triggers, timings, and the resource compliance states.

**Evaluation triggers**

Evaluations of assigned policies and initiatives happen as the result of various events:

* A policy or initiative is newly assigned to a scope
* A policy or initiative already assigned to a scope is updated
* A resource is deployed to or updated in a scope with an assignment through Azure Resource Manager, REST API, or a supported SDK
* A subscription (resource type Microsoft.Resources/subscriptions) is created or moved in a management group hierarchy with an assigned policy definition that targets the subscription resource type
* A policy exemption is created, updated, or deleted
* Standard compliance evaluation cycle
* The machine configuration resource provider is updated with compliance details by a managed resource
* On-demand scan

**Evaluation timing**

When you're working with policy assignments in Azure, you need to understand the behavior and timing of compliance scans, especially in Brownfield scenarios, where new policies are applied to existing resources. Compliance scans through Azure policies are triggered by various methods:

* Automatic full scan - A full compliance scan is triggered automatically every 24 hours.
* Manual scan for Brownfield scenarios - In cases where a new policy is applied to existing resources (Brownfield scenarios), you can manually trigger a compliance scan by running az policy state trigger-scan.

**Resource compliance states**

When initiative or policy definitions are assigned, Azure Policy determines which resources are applicable. Then, it evaluates those resources that aren't excluded or exempted. Evaluation provides one of the compliance states to each resource based on conditions in the policy rule and each resource's adherence to those requirements.

* Non-compliant
* Compliant
* Error (for template or evaluation error)
* Conflicting (two or more policy assignments in the same scope with contradicting rules, such as two policies appending the same tag with different values)
* Protected (resource covered under an assignment with a denyAction effect)
* Exempted Unknown (default state for definitions with a manual effect)

When multiple resources or policies have varying compliance states, the overall compliance state is assessed individually for each resource and for each policy assignment. Azure Policy ranks each compliance state so that one wins over another in this situation. The rank order of the states is as given in the previous list of compliance states.

The compliance percentage is determined by dividing Compliant, Exempt, and Unknown resources by total resources. Total resources include resources with Compliant, Non-compliant, Unknown, Exempt, Conflicting, and Error states.

**Enforcement Mode**

enforcementMode is a property of a policy assignment that lets you deactivate the enforcement of certain policy effects. This mode allows you to test the policy's outcome on existing resources without initiating the policy effect or triggering entries in the Azure Activity log. The enforcementMode can be changed to Enabled after the policy is thoroughly tested.

This scenario is commonly referred to as What If and aligns to safe deployment practices. The enforcementMode is different from the disabled effect. The disabled effect prevents resource evaluation from happening at all while enforcementMode lets the evaluation happen without the effect taking place.

**2.5 - Azure RBAC**

**Azure subscriptions**

Azure subscription is associated with a single Microsoft Entra directory. Users, groups, and applications in that directory can manage resources in the Azure subscription. The subscriptions use Microsoft Entra ID for single sign-on (SSO) and access management. You can extend your on-premises Active Directory to the cloud by using Microsoft Entra Connect. This feature allows your employees to manage their Azure subscriptions by using their existing work identities. When you disable an on-premises Active Directory account, it automatically loses access to all Azure subscriptions connected with Microsoft Entra ID.

**What's Azure RBAC?**

Azure RBAC is an authorization system built on Azure Resource Manager that provides fine-grained access management for resources in Azure. With Azure RBAC, you can grant the exact access that users need to do their jobs. For example, you can use Azure RBAC to let one employee manage virtual machines in a subscription, while another manages SQL databases within the same subscription.

You can grant access by assigning the appropriate Azure role to users, groups, and applications at a certain scope. The scope of a role assignment can be a management group, subscription, a resource group, or a single resource. A role assigned at a parent scope also grants access to the child scopes contained within it.

**Azure RBAC in the Azure portal**

In several areas in the Azure portal, you'll see a pane named Access control (IAM), also known as identity and access management. On this pane, you can see who has access to that area and their role. Using this same pane, you can grant or remove access.

**Role assignment**

Once you have determined the who, what, and where, you can combine those elements to grant access. A role assignment is the process of binding a role to a security principal at a particular scope for the purpose of granting access. To grant access, you'll create a role assignment. To revoke access, you'll remove a role assignment.

**How does Azure RBAC work?**

You can control access to resources using Azure RBAC by creating role assignments, which control how permissions are enforced. To create a role assignment, you need three elements: a security principal, a role definition, and a scope. You can think of these elements as who, what, and where.

* Security principal (who) is just a fancy name for a user, group, or application to which you want to grant access.
* Role definition (what) is a collection of permissions. It's sometimes just called a role. A role definition lists the permissions the role can perform such as read, write, and delete. Roles can be high-level, like Owner, or specific, like Virtual Machine Contributor.
* Scope (where) is the level where the access applies. This is helpful if you want to make someone a Website Contributor but only for one resource group.

**Azure RBAC is an allow model**

Azure RBAC is an allow model. This means that when you're assigned a role, Azure RBAC allows you to perform certain actions such as read, write, or delete. If one role assignment grants you read permissions to a resource group, and a different role assignment grants you write permissions to the same resource group, then you'll have read and write permissions on that resource group.

Azure RBAC has something called NotActions permissions. You can use NotActions to create a set of not allowed permissions. The access a role grants—the effective permissions—is computed by subtracting the NotActions operations from the Actions operations.

**2.6 - Allow users to reset their password with Microsoft Entra self-service password reset**

**Why use self-service password reset (SSPR)?**

In Microsoft Entra ID, any user can change their password if they're already signed in. But if they're not signed in, forgot their password, or it's expired, they'll need to reset their password. With SSPR, users can reset their passwords in a web browser or from a Windows sign-in screen to regain access to Azure, Microsoft 365, and any other application that uses Microsoft Entra ID for authentication.

SSPR reduces the load on administrators because users can fix password problems themselves without having to call the help desk. Also, it minimizes the productivity impact of a forgotten or expired password. Users don't have to wait until an administrator is available to reset their password.

**How SSPR works**

The user initiates a password reset either by going directly to the password-reset portal, or by selecting the Can't access your account link on a sign-in page. The reset portal takes these steps:

**Localization**: The portal checks the browser's locale setting and renders the SSPR page in the appropriate language.

**Verification**: The user enters their username and passes a CAPTCHA to ensure that it's a user and not a bot.

**Authentication**: The user enters the required data to authenticate their identity. They might enter a code or answer security questions.

**Password reset**: If the user passes the authentication tests, they can enter a new password and confirm it.

**Notification**: A message is sent to the user to confirm the reset.

**Authenticate a password reset**

It's critical to verify a user's identity before you allow a password reset. Malicious users might exploit any weakness in the system to impersonate that user. Azure supports six different ways to authenticate reset requests.

As an administrator, you can choose the methods to use when you configure SSPR. Enable two or more of these methods so that users can choose the ones they can easily use.

**Configure notifications**

Administrators can choose how users are notified of password changes. There are two options you can enable:

**Notify users on password resets**: The user who resets their own password is notified to their primary and secondary email addresses. If the reset was done by a malicious user, this notification alerts the user, who can take mitigation steps.

**Notify all admins when other admins reset their password**: All administrators are notified when another administrator resets their password.

**License requirements**

There are two editions of Microsoft Entra ID, Premium P1 and Premium P2. The password-reset functionality you can use depends on your edition.

Any user who is signed in can change their password, regardless of the edition of Microsoft Entra ID.

What if you're not signed in, and you've forgotten your password or your password has expired? In this case, you can use SSPR in Microsoft Entra ID P1 or P2. It's also available with Microsoft 365 Apps for business or Microsoft 365.

In a hybrid situation, where you have Active Directory on-premises and Microsoft Entra ID in the cloud, any password change in the cloud must be written back to the on-premises directory. This writeback support is available in Microsoft Entra ID P1 or P2. It's also available with Microsoft 365 Apps for business.

**SSPR deployment options**

You can deploy SSPR with password writeback by using Microsoft Entra Connect or cloud sync, depending on user needs. You can deploy each option side-by-side in different domains to target different sets of users. This helps existing users on-premises to write back password changes, while adding an option for users in disconnected domains because of a company merger or split. Users from an existing on-premises domain can use Microsoft Entra Connect, while new users from a merger can use cloud sync in another domain.

Cloud sync can also provide higher availability, because it doesn't rely on a single instance of Microsoft Entra Connect.