Table of Contents

[ACCESS MANAGEMENT 3](#_Toc155301974)

[AZURE ACTIVE DIRECTORY 3](#_Toc155301975)

[ON PREM VERSUS AZURE AD 4](#_Toc155301976)

[AD TERMINOLOGIES 4](#_Toc155301977)

[TRUST BETWEEN SUBSCRIPTION AND AD 4](#_Toc155301978)

[CONFIGURING THE TRUST BETWEEN AD AND SUBSCRIPTION 5](#_Toc155301979)

[TYPES OF PERMISSIONS IN AZURE 6](#_Toc155301980)

[A. RBAC ROLES (IAM – IDENTITY AND ACCESS MANAGEMENT) 7](#_Toc155301981)

[B. DIRECTORY ROLES (AD ROLES) 7](#_Toc155301982)

[CREATING USER 7](#_Toc155301983)

[ROLE BASED ACCESS CONTROL (RBAC) 10](#_Toc155301984)

[SCOPE OF RBAC 10](#_Toc155301985)

[TYPES OF RBAC ROLES 11](#_Toc155301986)

[AZURE AD ROLES 16](#_Toc155301987)

[GRANTING AD ROLES 16](#_Toc155301988)

[CREATING USER GROUP 17](#_Toc155301989)

[CREATING DYNAMIC USER GROUP 18](#_Toc155301990)

[JOINING A VM TO AD 18](#_Toc155301991)

[MULTIFACTOR AUTHENTICATION 20](#_Toc155301992)

[SETTING UP PER USER MFA 20](#_Toc155301993)

[INVITING GUEST USER 20](#_Toc155301994)

[USER CREATION / DELETION IN BULK 21](#_Toc155301995)

[ADMINISTRATIVE UNITS 22](#_Toc155301996)

[LAB -ADMINISTRATIVE UNITS 22](#_Toc155301997)

[AZURE AD CUSTOM DOMAIN 24](#_Toc155301998)

[STEPS TO ADD CUSTOM DOMAIN 24](#_Toc155301999)

[MULTIFACTOR AUTENTICATION(MFA) 26](#_Toc155302000)

[ENABLING MFA FOR USER 26](#_Toc155302001)

[CONDITIONAL ACCESS POLICY 29](#_Toc155302002)

[RESOURCE TAGS 29](#_Toc155302003)

[COST MANAGEMENT ON TAGS 30](#_Toc155302004)

[MOVING RESOURCES ACROSS RESOURCE GROUPS 30](#_Toc155302005)

[MOVING RESOURCES ACROSS SUBSCRIPTIONS 30](#_Toc155302006)

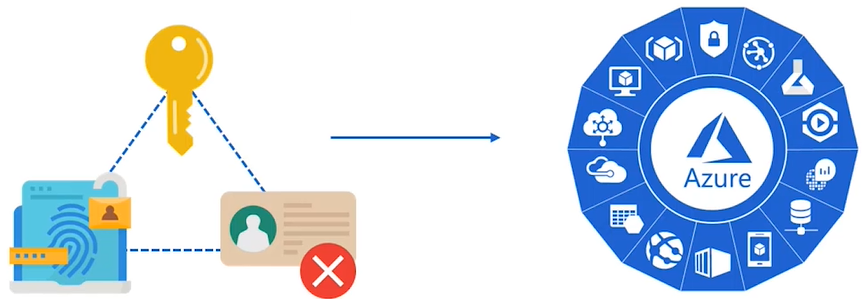
[LOCKING RESOURCES 31](#_Toc155302007)

[SETTING UP RESOURCE LOCK 32](#_Toc155302008)

[LOCKS AND MOVING RESOURCES 32](#_Toc155302009)

# ACCESS MANAGEMENT

* Access Management in cloud refers to a process that allows, denies, or restrict access to services or resources.
* This also includes, deciding who gets the access and to what extent, in cloud.



* Access Management in Azure can be achieved by RBAC(Role Based Access Control)

# AZURE ACTIVE DIRECTORY

* Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service.
* It serves as a comprehensive identity and access management solution for managing user identities and controlling access to resources in the Azure cloud environment and other integrated applications.
* The key benefit of Azure Active Directory is that we don’t need to implement any infrastructure.

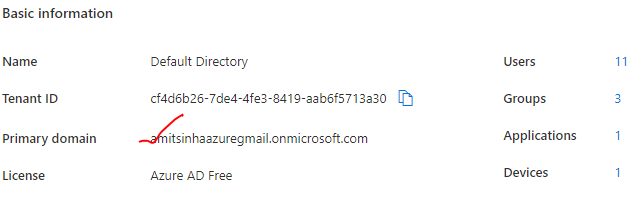
KEY FEATURES AND CAPABILITIES OF AZURE ACTIVE DIRECTORY INCLUDE:

* IDENTITY MANAGEMENT:
  + Azure AD provides a centralized identity management system to manage user identities and access across various applications and services.
  + It allows administrators to create, manage, and synchronize user accounts, groups, and roles.
* SINGLE SIGN-ON (SSO):
  + Azure AD enables users to sign in once with their Azure AD credentials and access multiple applications and services without the need for separate sign-ins.
  + This enhances user experience and simplifies access management for administrators.
* MULTI-FACTOR AUTHENTICATION (MFA):
  + Azure AD supports multi-factor authentication, adding an extra layer of security to user sign-ins.
  + MFA requires users to provide additional verification, such as a code sent to their mobile device, in addition to their password.
* APPLICATION MANAGEMENT:
  + Azure AD provides application management capabilities, allowing administrators to register and manage applications in Azure AD.
  + This includes managing access permissions, defining single sign-on configurations, and enforcing security policies.
* ROLE-BASED ACCESS CONTROL (RBAC):
  + Azure AD integrates with Azure RBAC, allowing administrators to assign granular access permissions to users and groups for Azure resources.
  + RBAC enables the principle of least privilege, ensuring users only have access to the resources they need.
* SEAMLESS INTEGRATION:
  + Azure AD integrates with a wide range of Microsoft and third-party applications, including Office 365, Azure services, and thousands of Software-as-a-Service (SaaS) applications.
  + This enables organizations to centrally manage access to various resources.
* CONDITIONAL ACCESS:
  + Azure AD's Conditional Access feature allows administrators to define policies that control access to resources based on specific conditions, such as user location, device compliance, or risk level.
  + This helps enforce security controls and prevent unauthorized access.

## ON PREM VERSUS AZURE AD

|  |  |
| --- | --- |
| ON PREM WINDOWS ACTIVE DIRECTORY | AZURE AD SERVICE |
| * In the on Prem set up we need to have a dedicated directory server aka Domain Controller. * On these domain controllers we need to enable/ install ADDS (Active Directory Domain Service). * These ADDS services need a domain name. * Now using ADDS - we can able create uses, group and join devices (desktop/ laptops etc.) under the ADDS domain. * The access control on the uses, group and devices are managed by Group Policies. | * In Azure provides a managed service for Active directory – hence we don’t need a dedicated directory server. * When we sign up for Azure, by default a **tenant** is created (tenant is also referred as directory). * *In the on prem side – a dedicated directory servers need to be set up for each organization, On the other hand Azure AD service is a distributed directory service.* * Since it’s a distributed service(global), the organization are uniquely identified by using tenant.   Note:   1. A tenant needs a primary domain name. By default. The tenant can have custom domain name as well. Adding the custom domain needs verifiction. |

**PRIMARY DOMAIN NAME**



## AD TERMINOLOGIES

|  |  |
| --- | --- |
| TENANT | * A tenant is a unique representation of an organization in Azure Active Directory. * When we create an azure account by default, we get an instance of Azure AD. Every such instance is called tenant. * It a dedicated and trusted instance of Azure AD |
| DEFAULT DIRECTORY | * Each tenant has a dedicated and trusted Azure AD directory. * This includes tenant user, group and used to perform identity and access management of Azure resources. |

* In Azure Portal, apart from default directory we create a new directory as well. Creating a new directory is same s creating a new tenant itself.

WHY AN ORGAINIZATION WILL CREATE NEW DIRECTORY RATHER THAN USING THE DEFAULT DIRECTORY?

***User-case***: Let's say if an organization has many other companies under that organization, so they can create directories for each of the companies in that organization. Hence rather than creating the users under this one directory, we can create uses company’s directory.

## TRUST BETWEEN SUBSCRIPTION AND AD

* The subscription is used for grouping of resources and used for the billing aspect. All the resources are built against a particular subscription.

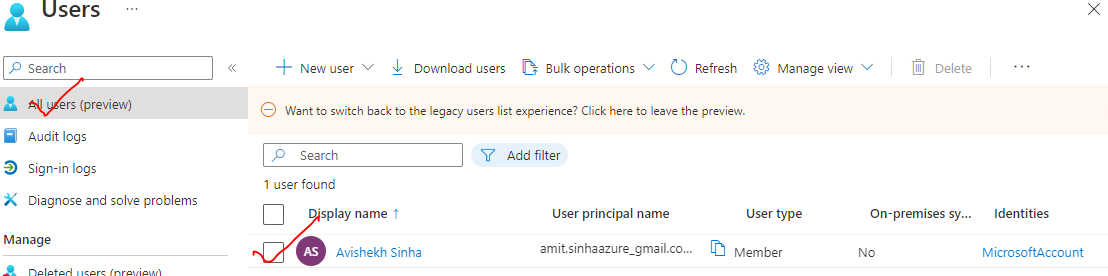
|  |  |
| --- | --- |
|  | * ***At any point in time, one subscription can trust one directory*** - so users are defined in this directory can be given access on resources that are defined in * It means that whatever users are defined in this directory can be given access on to the resources defined in that subscription. * Multiple subscriptions can also trust same active directory |

## CONFIGURING THE TRUST BETWEEN AD AND SUBSCRIPTION

|  |  |  |
| --- | --- | --- |
|  | Step 1: CREATE AN ACTIVE DIRECTORY(TENANT)   * Select Active Directory Service 🡪 Manage Tenant 🡪 Create * Select the Tenant type as “Azure Active Directory”   *Note - So if a company wants to have another directory in place for managing users separately, they can go out and create another as the Active Directory*.   * Once a new directory is created then we can go ahead and create user into the AD. * We can also change the directory for a subscription * The users in the AD will get an access to the resources created in that subscription | |
|  | To change the directory for a subscription   1. Go to Subscription🡪 Change Directory 🡪 Select the directory | |
|  | | |
| DELETING A AD(TENANT) – Go to Active Directory 🡪 Manage Tenant 🡪 Select the tenant 🡪 Delete | | |
|  | |  |

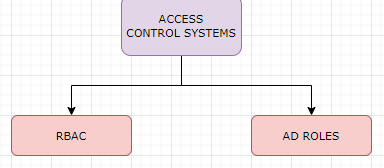
## CREATING USER IN AD

* When a user is created in azure- user can able to login to azure .
* While creating a user – user can be assigned with directory pernission (show below)
* Now when newly created user logs in - by default he will not have permission to view anything. **To give the access to the resources , resource group or subscription – We have to make use of RBAC permissions.**



|  |  |  |
| --- | --- | --- |
|  | | 1. To Create User🡪 New user 🡪 Create 2. The default domain is “*onmicrosoft.com*” 3. **The newly created user can login to Azure Portal, but by default it will have no access to any resource** 4. The Access to the user must be given explicitly |
|  | | **ASSIGNING DIRECTORY ROLES**   * Note – while creating the user – directory roles can be assigned to the user. These roles will enable user to control active directory. |
|  | | |
|  | **ASSIGNING DIRECTORY ROLE**   * Go to the user 🡪 Assigned Roles 🡪 Add Assignment. * Example - if “Global Reader” permission has been given – then user can able to read the active directory in Azure. | |

## TYPES OF ACCESS CONTROL SYSTEMS



Note :

**RBAC and AD roles serve different purposes within Azure. RBAC focuses on managing access to Azure resources, while AD roles manage access to Azure AD resources and services.**

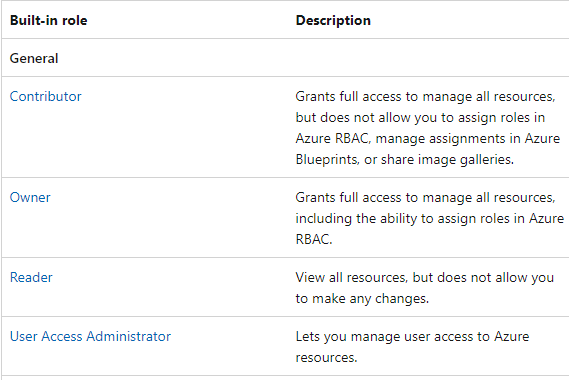
| **RBAC (Role-Based Access Control)** | **AD (Active Directory) Roles** |
| --- | --- |
| RBAC is a built-in access control system in Azure. | Azure AD roles are specific to Azure Active Directory. |
| **It manages access to Azure resources at different scopes (subscription, resource group, or resource level).** | **It manages access to Azure AD resources such as user accounts, groups, and applications.** |
| Uses roles to define sets of permissions for Azure resources. | Uses roles to define permissions for Azure AD resources. |
| Provides flexibility to assign granular permissions to users, groups, or applications. | Assigns roles to users or groups to control access levels within Azure AD. |
| **Offers built-in roles like Owner, Contributor, Reader, User Access Administrator, etc.** | **Offers built-in roles like Global Administrator, User Administrator, Application Administrator, etc.** |
| Permissions can be assigned directly to users, groups, or applications. | Permissions are assigned to users or groups in Azure AD, which then apply to Azure AD resources. |
| Can be used to manage access to resources across various Azure services. | Primarily used to manage access to Azure AD resources and services. |
| RBAC permissions can be customized, and new roles can be created. | AD roles can be customized within Azure AD, but new roles cannot be created. |

### RBAC ROLES (IAM – IDENTITY AND ACCESS MANAGEMENT)

* **RBAC in Azure is a built-in access control system that allows us to grant permissions to users, groups, or applications at different scopes (subscription, resource group, or resource level).**
* RBAC uses roles to define sets of permissions. Azure provides several built-in roles, such as Owner, Contributor, Reader, and User Access Administrator, with varying levels of access.
* We can assign these roles to users or groups to control what they can do with Azure resources. For example, we can assign the Contributor role to a user, allowing them to manage and make changes to resources within a specific scope.

|  |  |
| --- | --- |
| **RBAC TERMINOLOGIES** | |
| SCOPE | * The subject on which the RBAC permission are applied. * It can be a resource, Resource group and Subscription |
| SECURITY PRINCIPLE | * To whom the permission is given. It can be a User, Group or Managed Identity |
| ROLE | * What permissions are given. Below are the basic built in RBAC roles. |
| SECURITY |  |

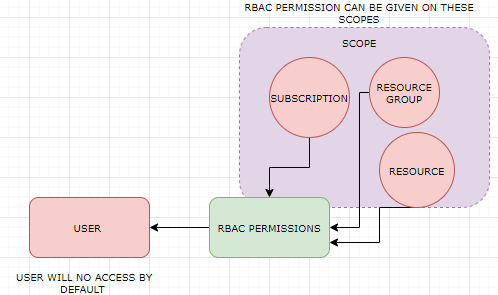
#### BASIC BUILT-IN RBACK ROLES



### DIRECTORY ROLES (AD ROLES)

* AD roles in Azure refer to the roles defined within Azure Active Directory, which is a cloud-based identity and access management service.
* **Azure AD roles are used to manage access to Azure AD resources, such as user accounts, groups, and applications.**
* Azure AD provides various **built-in roles, including Global Administrator, User Administrator, Application Administrator**, and many more.
* These roles determine the level of access and permissions users have within Azure AD. For example, the Global Administrator role has full control over Azure AD, while the User Administrator role can manage user accounts.

## ROLE BASED ACCESS CONTROL (RBAC)



* RBAC enables administrators to manage access to Azure resources by assigning roles to users, groups, or applications.
* RBAC allows for fine-grained control over permissions, ensuring that individuals have the appropriate level of access based on their responsibilities within an organization.

### SCOPE OF RBAC

* With RBAC in Azure, roles can be assigned at different scopes. The scope of RBAC are
  + **MANAGEMENT GROUP**
  + **SUBSCRIPTION**
  + **RESOURCE GROUP**
  + **INDIVIDUAL RESOURCE LEVELS.**

**THE ROLE-BASED ACCESS CAN BE GIVEN AT**

|  |  |
| --- | --- |
| RESOURCE LEVEL | The access can be to a resource – for example - storage account or VM |
| RESOURCE GROUP LEVEL | * Access can be given at resource group level * If access is given at resource group level – it will give access to all resources in that resource group. |
| SUBSCRIPTION LEVEL | * If access is given at subscription level – it will give access to all resources in that subscription |

### TYPES OF RBAC ROLES

1. **BUILT IN ROLES**
2. **CUSTOM ROLES**

#### BUILD IN ROLES

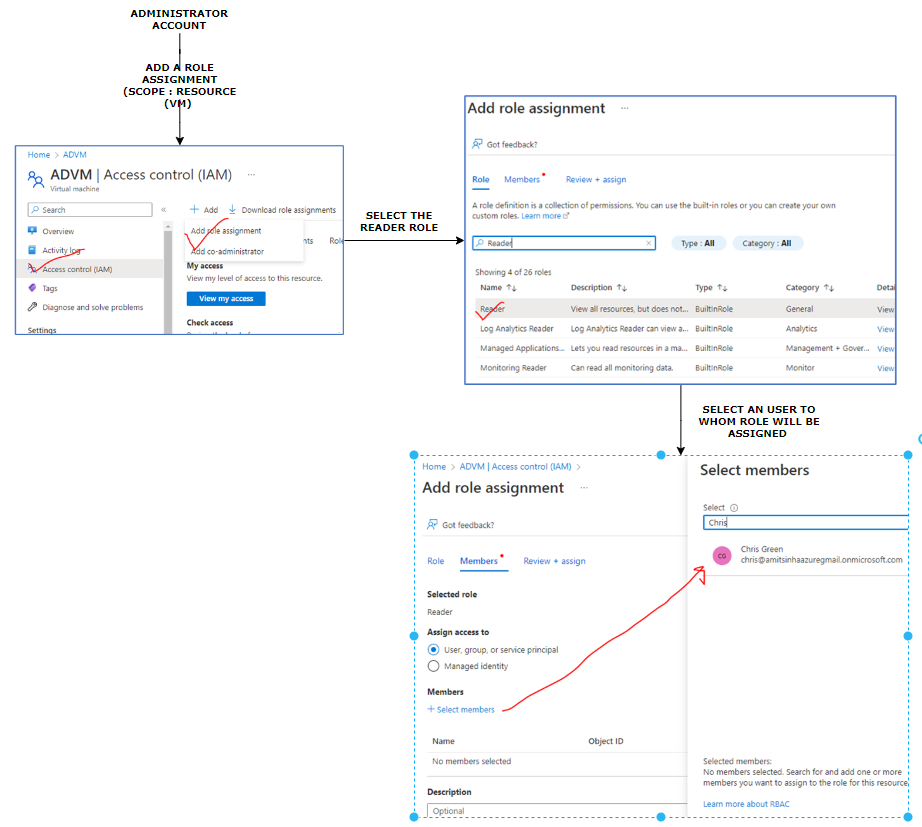
* In Azure, there are several built-in RBAC (Role-Based Access Control) roles available that can be used to manage access and permissions to Azure resources.
* **These roles provide predefined sets of permissions that can be assigned to users, groups, or applications**.
* Azure has number of in-built roles - <https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles>, but we can also create our own custom roles as well.

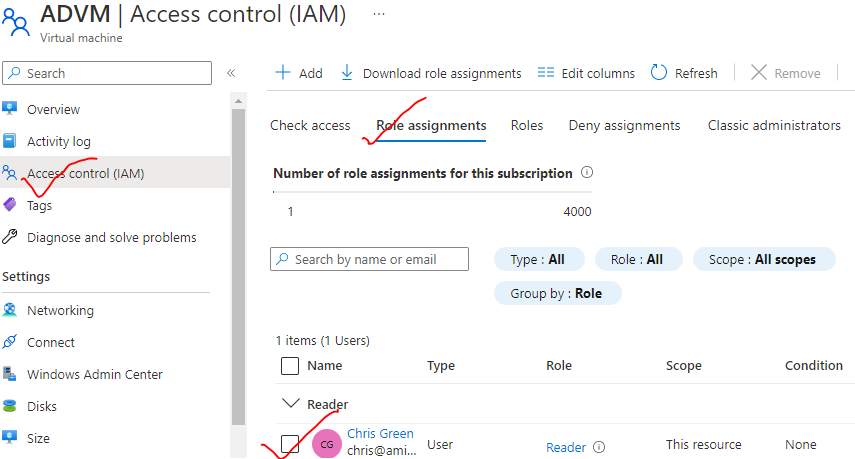
**COMMONLY USED BUILT-IN RBAC ROLES IN AZURE**

|  |  |
| --- | --- |
| **OWNER** | * This role has full access to manage all resources within a scope, including the ability to modify permissions, delete resources, and manage resource settings. |
| **CONTRIBUTOR** | * This role can create and manage all types of Azure resources, but cannot grant access to other users or manage permissions. |
| **READER** | * This role has read-only access to view resources and their properties, but cannot make any changes or perform any actions. |
| **USER ACCESS ADMINISTRATOR** | * This role can manage user access to Azure resources, including granting and revoking access permissions. |
| **VIRTUAL MACHINE CONTRIBUTOR** | * This role has permissions to manage virtual machines, including starting, stopping, and restarting them. |
| **NETWORK CONTRIBUTOR** | * This role can manage networking resources, such as virtual networks, subnets, and network security groups. |
| **STORAGE ACCOUNT CONTRIBUTOR** | * This role can manage storage accounts and their associated resources, such as blobs, files, and queues. |
| **SQL SERVER CONTRIBUTOR** | * This role has permissions to manage SQL servers and databases, including creating, deleting, and modifying them. |

##### ROLE BASED ASSIGNMENT – READER ROLE

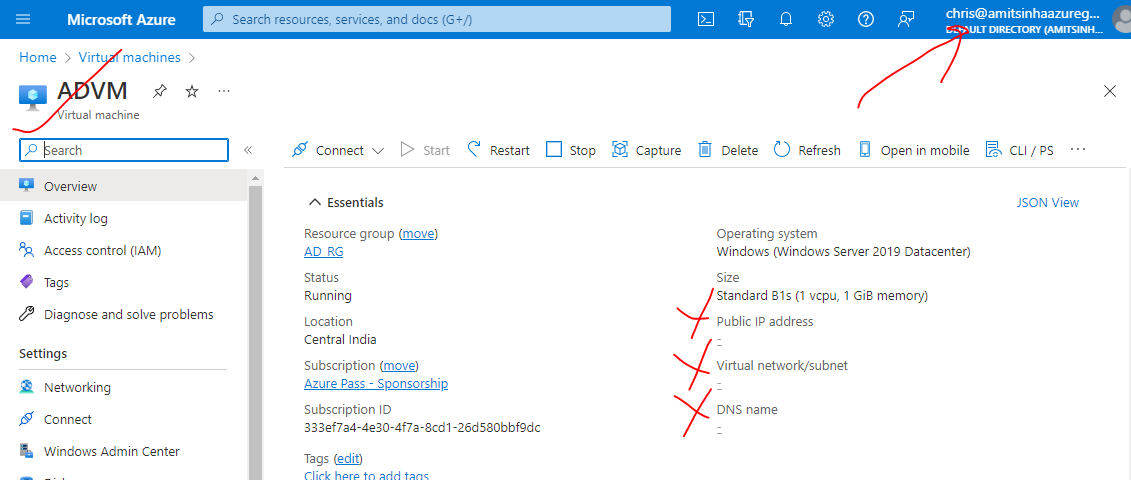
* Example – if we want to give access the Reader permission on VM level (Resource level)
* *Navigate to VM 🡪 IAM🡪 Add 🡪 Add Role Assignment*





WHEN USER LOGS IN AZURE PORTAL

* As the user has reader permission on VM only – hence he have will no access to public IP , VNET and DNS – as they are separate resources in Azure- which needs a separate role assignment.

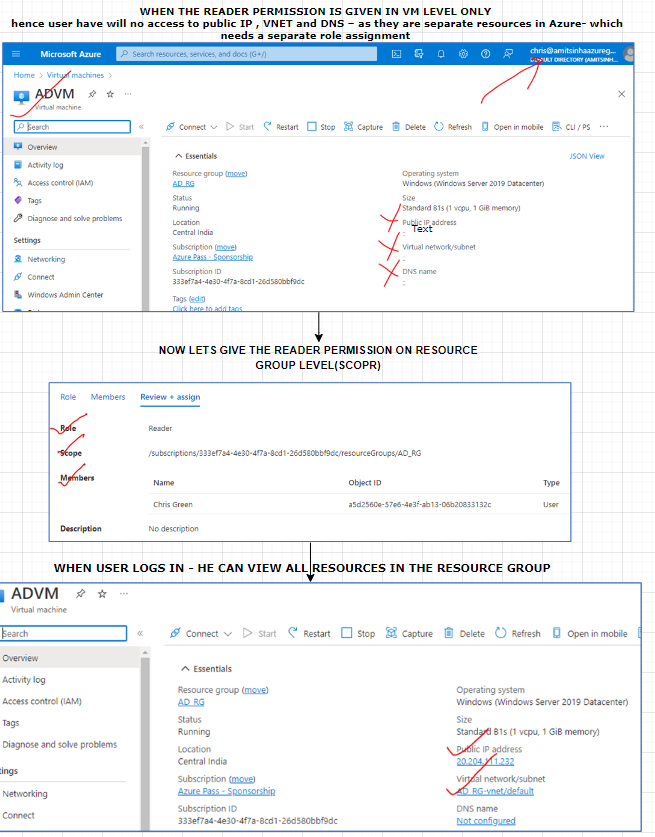


#### ROLE ASSIGNMENT IN DIFFERENT RBAC SCOPES

|  |  |
| --- | --- |
| RESOURCE LEVEL | * The access can be to a resource – for example - storage account or VM |
| RESOURCE GROUP LEVEL | * Access can be given at resource group level * If access is given at resource group level – it will give access to all resources in that resource group. |
| SUBSCRIPTION LEVEL | * If access is given at subscription level – it will give access to all resources in that subscription |

##### ROLE BASED ASSIGNMENT – RESOURCE GROUP LEVEL

* **IF WE ASSIGN A ROLE AT A RESOURCE GROUP LEVEL, IT WILL BE APPLIED TO (INHERITED BY) ALL THE RESOURCES WITHIN THE RESOURCE GROUP.**



##### ROLE BASED ASSIGNMENT – SUBSCRIPTION LEVEL

* If we assign a role to a the identity (user/ group) at the subscription level, then the role gets applied to or inherited to all resource groups that are part of that subscription.

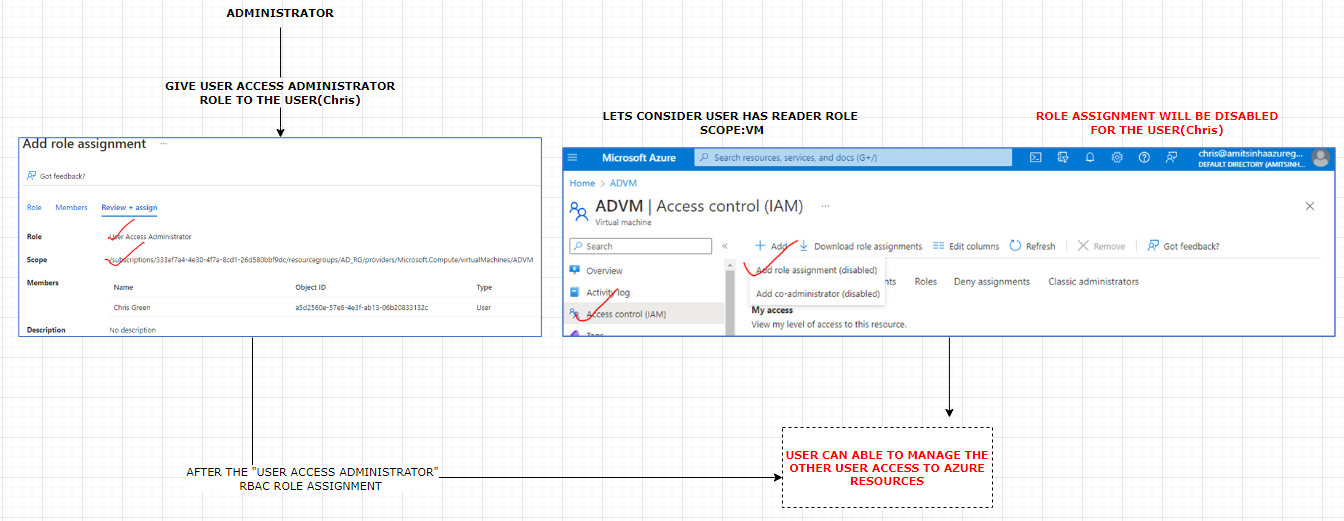
##### ROLE BASED ASSIGNMENT – CONTRIBUTOR LEVEL

* The Contributor role allows users to manage and make changes to Azure resources, except for granting access to others.

|  |  |
| --- | --- |
| Grants full access to manage all resources but does not allow to assign roles in Azure RBAC, manage assignments in Azure Blueprints, or share image galleries. | * When a user is assigned a role of reader – then no operation can be performed by the user except view. * To enable this capability – we can assign contributor role to the user. * For example - VM contributor role (To perform VM operations). In the below example - we are assigning *VM contributor Role* on resource group level |

##### ROLE BASED ASSIGNMENT – USER ACCESS ADMIN ROLE

* The role allows the user to give access to other users / add role-based access to another user. The “User Access Admin Role” – enable the user for role-based assignments.



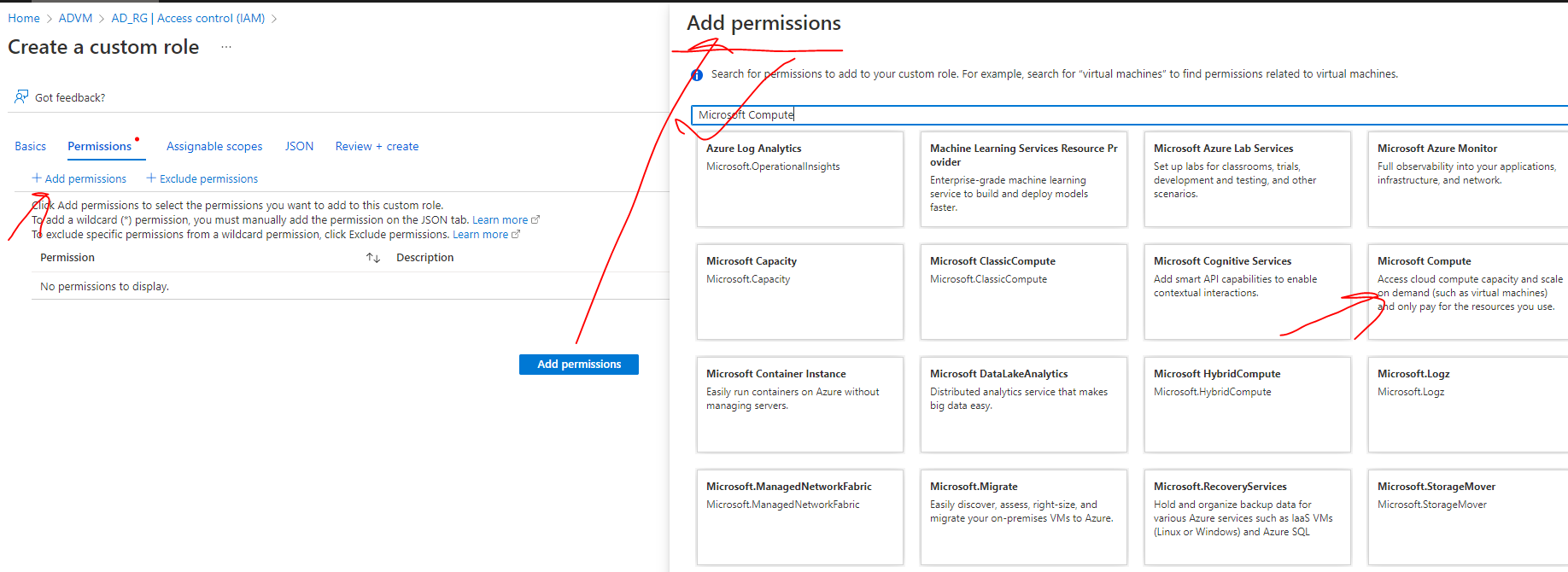
#### RBAC - CUSTOM ROLES

* <https://learn.microsoft.com/en-us/azure/role-based-access-control/custom-roles-portal>
* There are three ways that we can start to create a custom role.
  + **CLONE AN EXISTING ROLE**
  + **START FROM SCRATCH**
  + **START WITH A JSON FILE**
* The easiest way is to find an existing role that has most of the permissions you need and then clone and modify best on your scenario.
* In the Azure portal, **open a management group, subscription, or resource group** where we want the custom role to be assignable and then open **Access control (IAM)**.

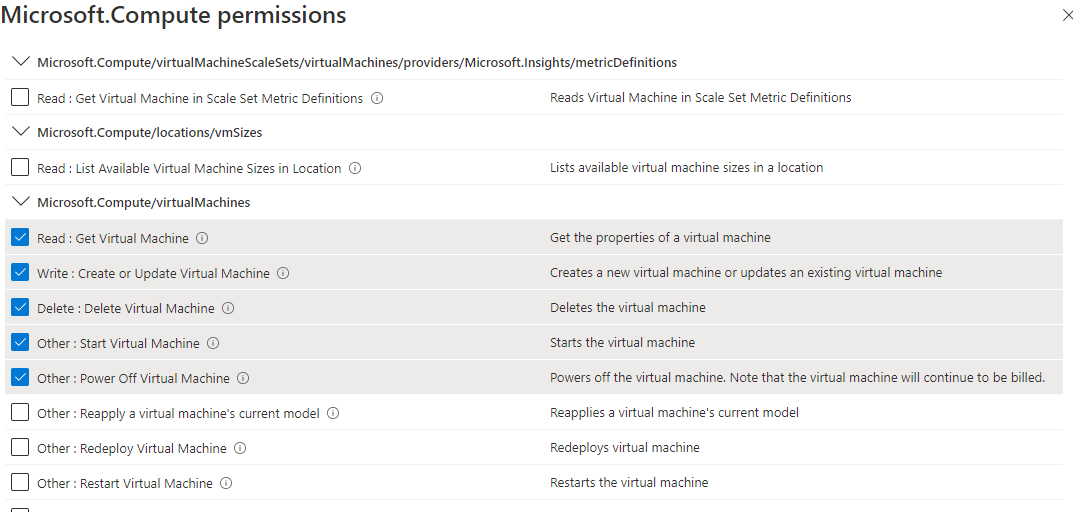
##### CREATING CUSTOM ROLES – START FROM SCRATCH

|  |  |
| --- | --- |
|  | * In the Azure portal, **open a management group, subscription, or resource group** where we want the custom role to be assignable and then open **Access control (IAM)**. |
|  | * Let’s select “start from scratch”🡪 next |

* Let’s say we want to create role for a VM 🡪 select Microsoft Compute



* Select the required permission from the Compute permission list



|  |  |
| --- | --- |
|  | * Once the custom role is created – it can be then assigned to any level |

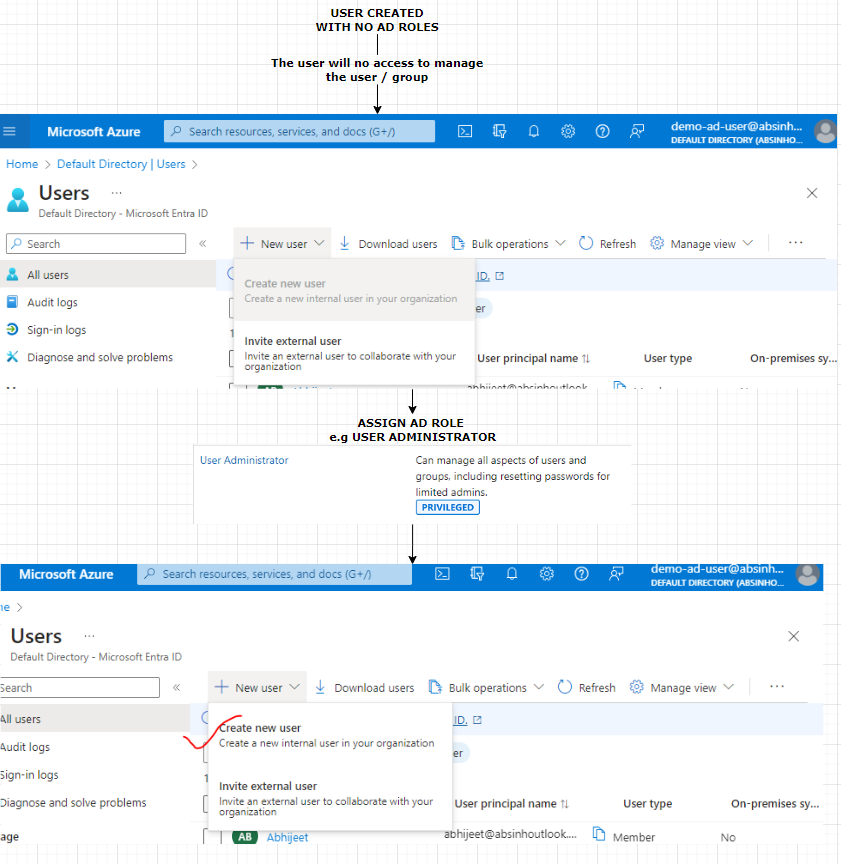
## CREATING USER GROUP

|  |  |
| --- | --- |
|  | * Go to Azure AD 🡪 Groups 🡪 New Group * Note – members can be added while creating the group itself or later too. |
| * To add member to the group 🡪 Go to the group 🡪 Members 🡪 Add members | |

## AZURE AD ROLES

* AD roles are purely meant to carry out tasks in Azure Active Directory only. AD roles are dedicated to managing the Active directory itself. for example, to create an application, create a user, create a group etc. For example - In an organization, these activities are they typically delegated to administrators.
* On the other hand, the RBAC are used to give a user access to resources within Azure.
* <https://learn.microsoft.com/en-us/azure/active-directory/roles/permissions-reference>

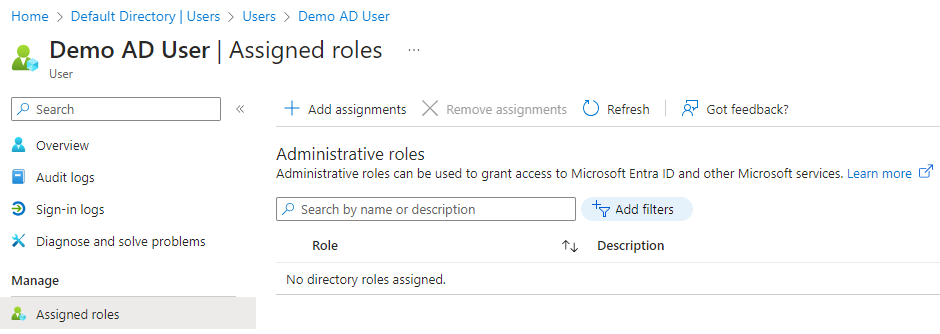
### ASSIGNING AD ROLES



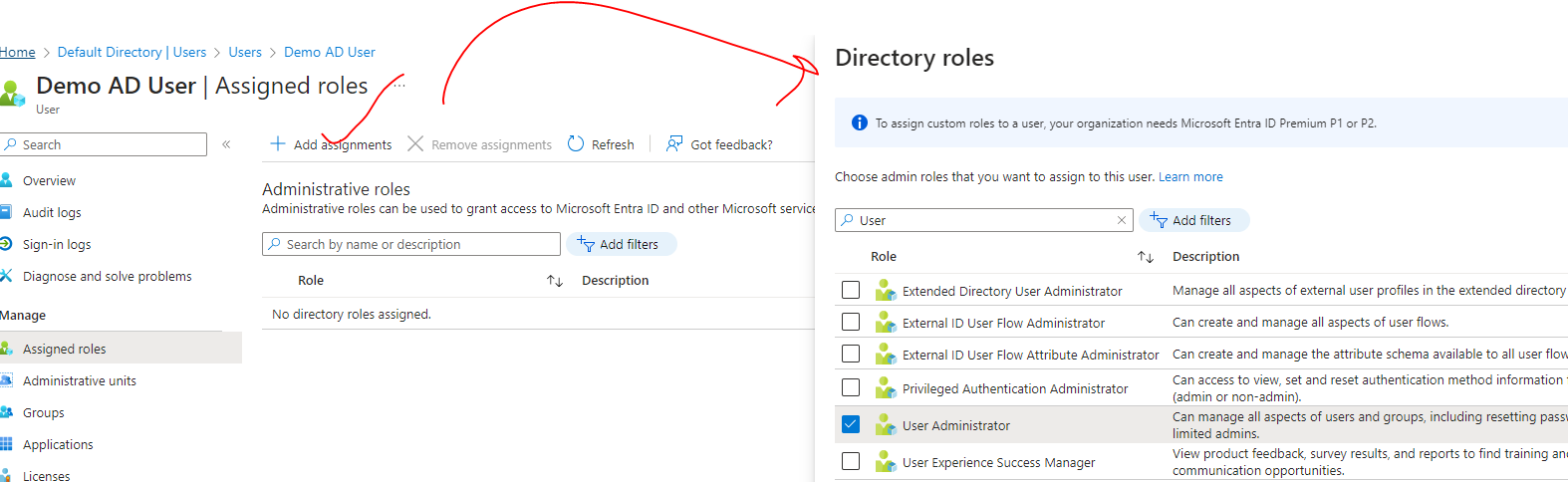
#### ASSIGNING AD ROLE TO USER

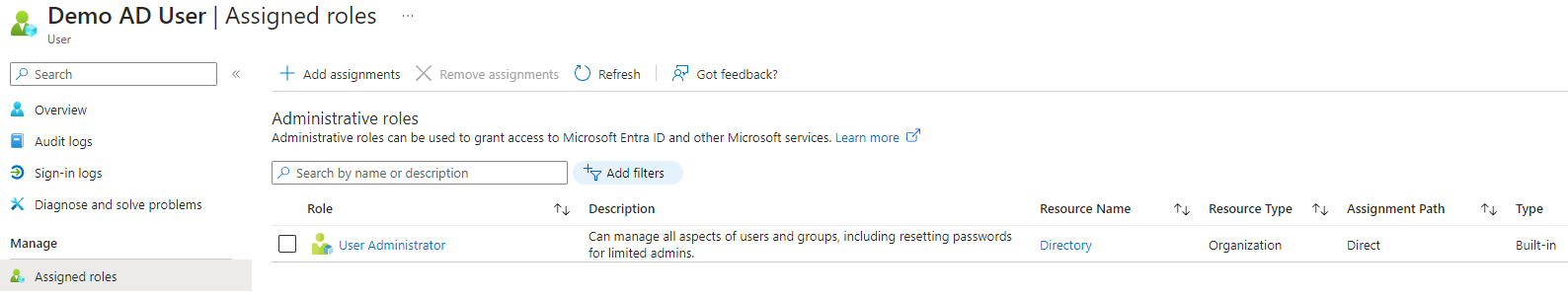
|  |
| --- |
| NOTE: For a user   1. Assigned Roles: THE IS USED TO ASSIGN AD ROLES. 2. Azure role assignments: THE IS USED TO ASSIGN RBAC ROLE TO THE USER |

* To assign AD roles to a user(Demo AD User) . Navigate to the user 🡪 Assigned roles



Step 1: Click on assigned roles 🡪 Select any AD roles (For example: **User Administrator**)





### BUILT IN AD ROLES

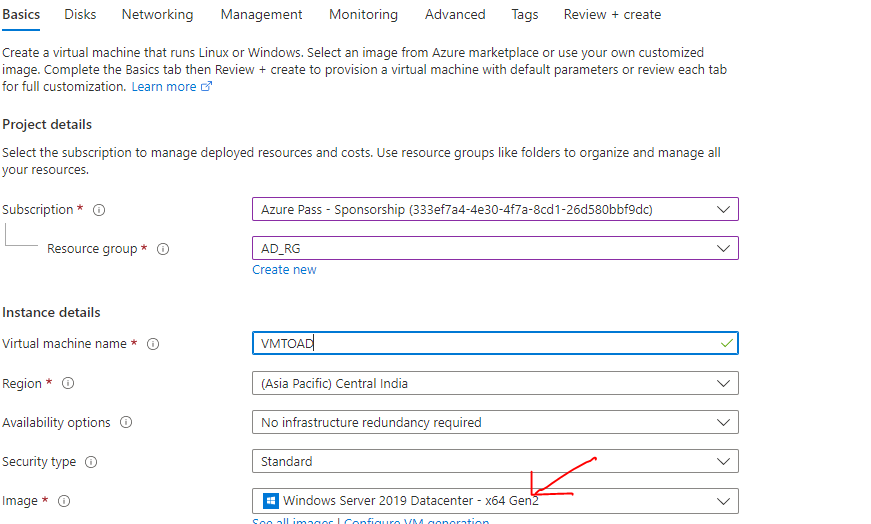
* <https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/permissions-reference>

|  |  |
| --- | --- |
|  | * Navigate to Azure Active Directory 🡪 Roles and Administrators * Select the AD role e.g., User Administrator (<https://learn.microsoft.com/en-us/azure/active-directory/roles/permissions-reference#user-administrator> ) * Select the user (e.g **UserD**) to which AD roles must be assigned. * The AD roles can also be assigned to User Groups as well * The AD roles can be removed using Remove Assignment option, |
|  | |

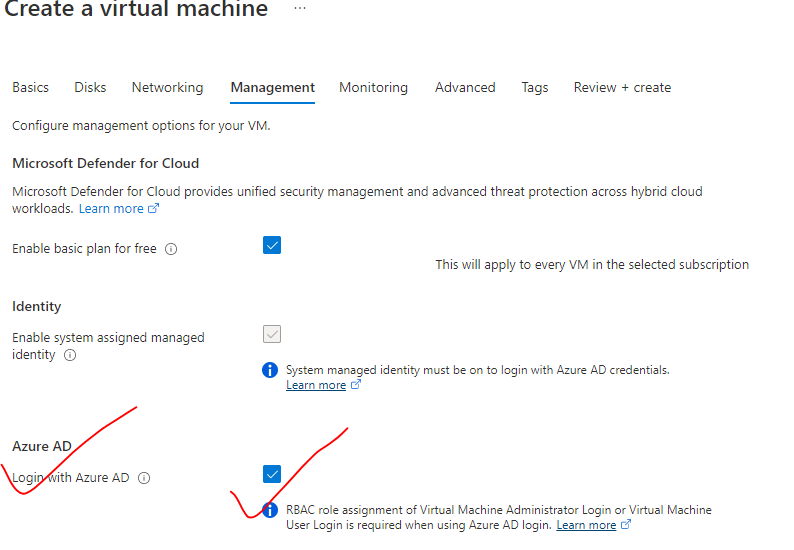
## JOINING A VM TO AD

* Windows 10 and based device, if you have Windows server 2019 can be joined to an Active Directory.

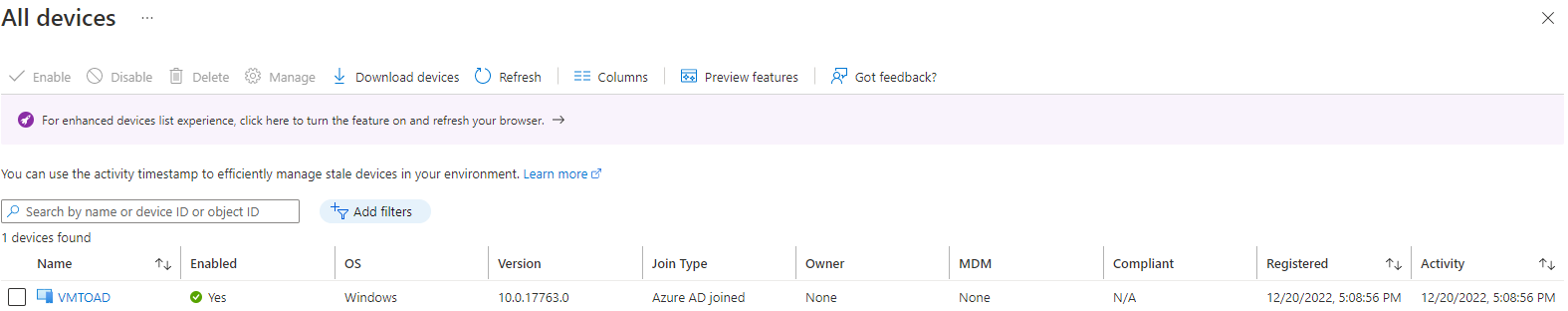
Step 1: Provision a Window 10 / Server VM



Step 2: Register the device with AD



* To check the connected device – Active Directory 🡪 Devices

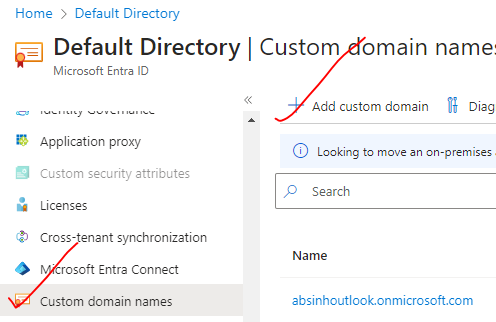


## AZURE AD CUSTOM DOMAIN

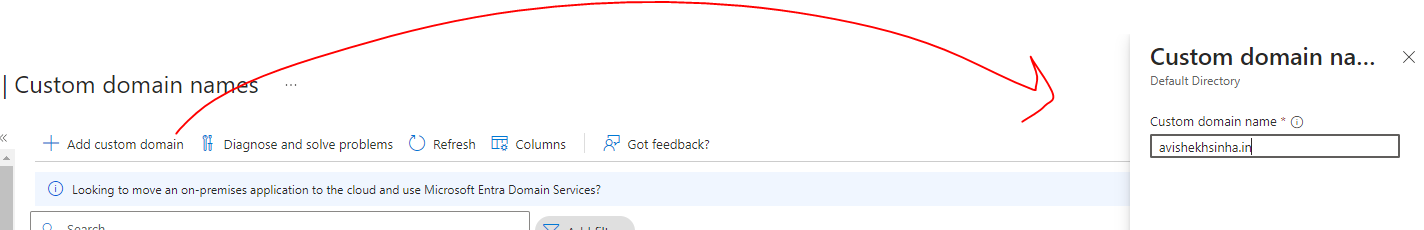
* In Azure Active Directory (Azure AD), we can add and configure a custom domain to use with the Azure AD tenant.
* This allows us to use our own domain name for user sign-in and other services within Azure AD.

### STEPS TO ADD CUSTOM DOMAIN

1. Under the Azure Active Directory menu, select "Custom domains" 🡪 Click on the "Add custom domain" button to start adding a new custom domain

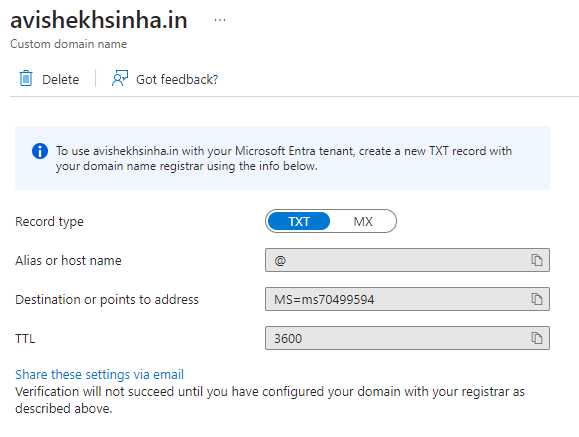


1. Enter the domain name we want to use, such as "avishekh.sinha.in," and click "Add domain."



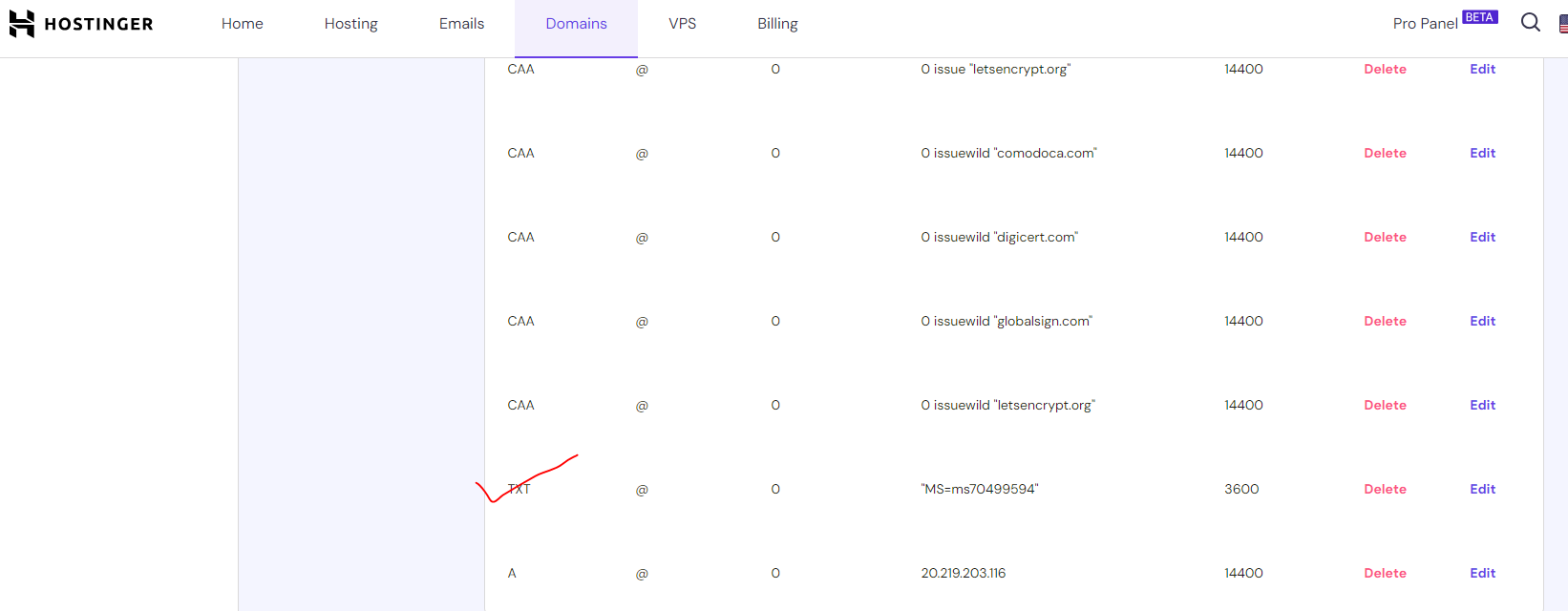
1. VERIFY DOMAIN OWNERSHIP

Azure AD will provide instructions for verifying the domain ownership. This typically involves adding a TXT or MX record to your DNS configuration with the provided values.

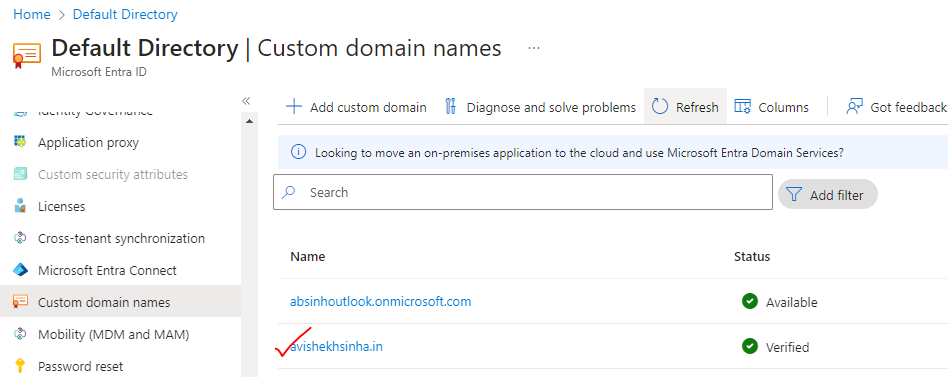


Add the text record in the in the

**TXT RECORD FROM DNS PROVIDER**



1. VERIFY DOMAIN:
   * Once we have added the necessary DNS records, click on the "Verify" button in the Azure portal to initiate the domain verification process.
   * Azure AD will verify the DNS records and confirm domain ownership.



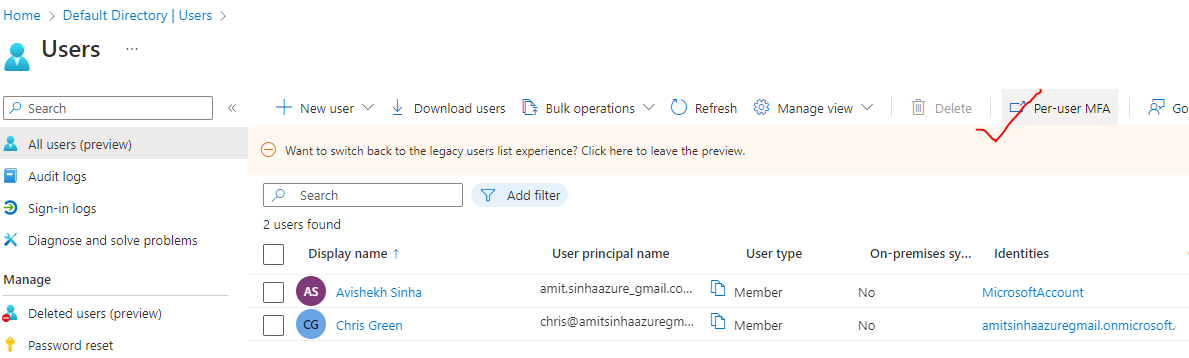
1. CONFIGURE USER SIGN-IN:
   * After the domain is verified, we can configure user sign-in using the custom domain.
   * **We have the option to set the custom domain as the default domain or allow users to sign in with both the custom domain and the initial <your-domain>.onmicrosoft.com domain.**

## SELF SERVICE PASSWORD RESET

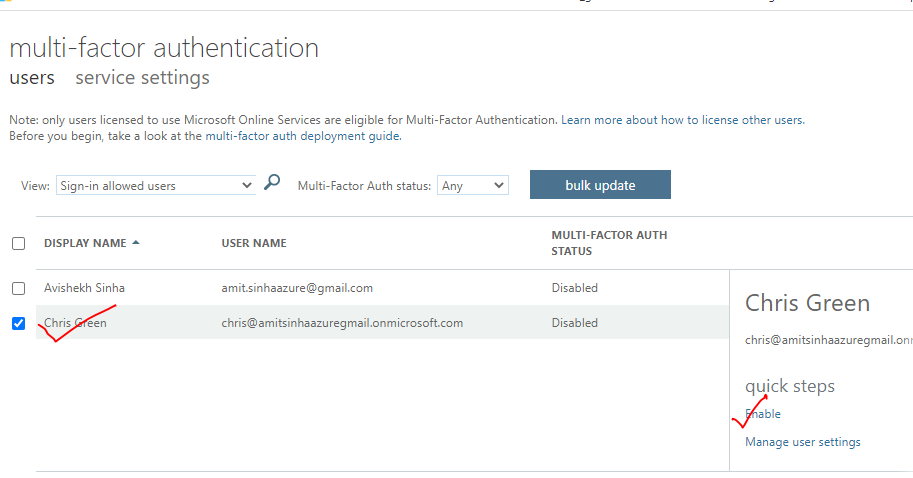
## MULTIFACTOR AUTHENTICATION

### SETTING UP PER USER MFA

* Go to Active directory 🡪 Per user MFA



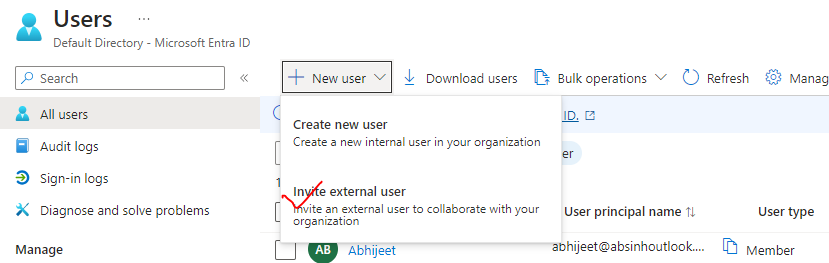
* Select user 🡪 Enable

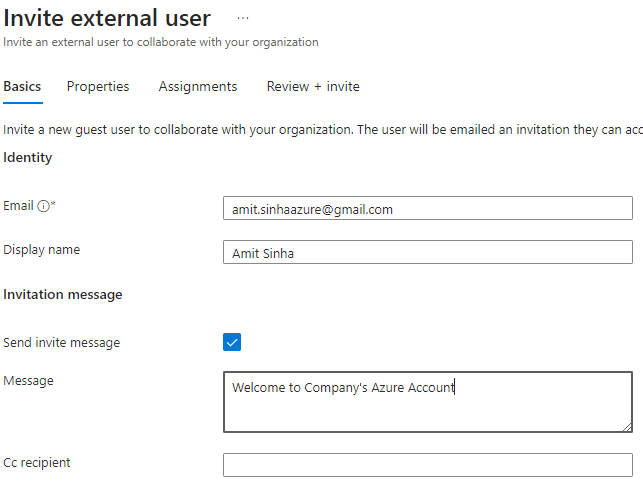


## INVITING GUEST USER

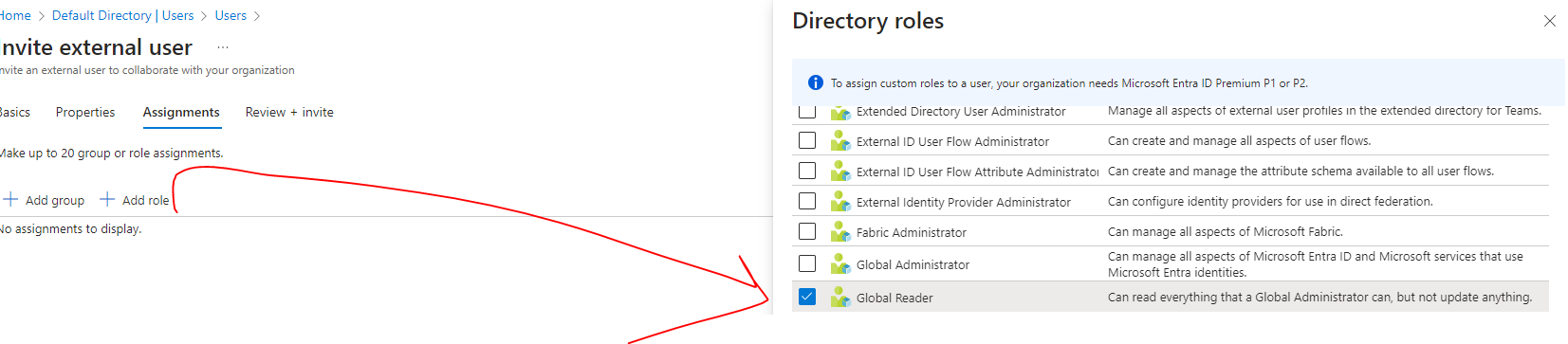
* Inviting a guest user in Azure Active Directory (AD) refers to the process of granting access to external users, who do not belong to your organization, to collaborate and access resources within your Azure AD tenant.
* When we invite a guest user, we are essentially extending an invitation for them to join our Azure AD environment as a guest. This allows them to sign in with their own credentials from a different organization or as an individual user.
* Guest users can be invited to collaborate on various Azure services, such as Azure DevOps, Azure B2B collaboration, Azure Portal, Azure Active Directory, and more. By inviting guest users, we can enable collaboration and resource access for partners, vendors, contractors, and other external users, while maintaining control over their permissions and access levels within your Azure AD tenant.

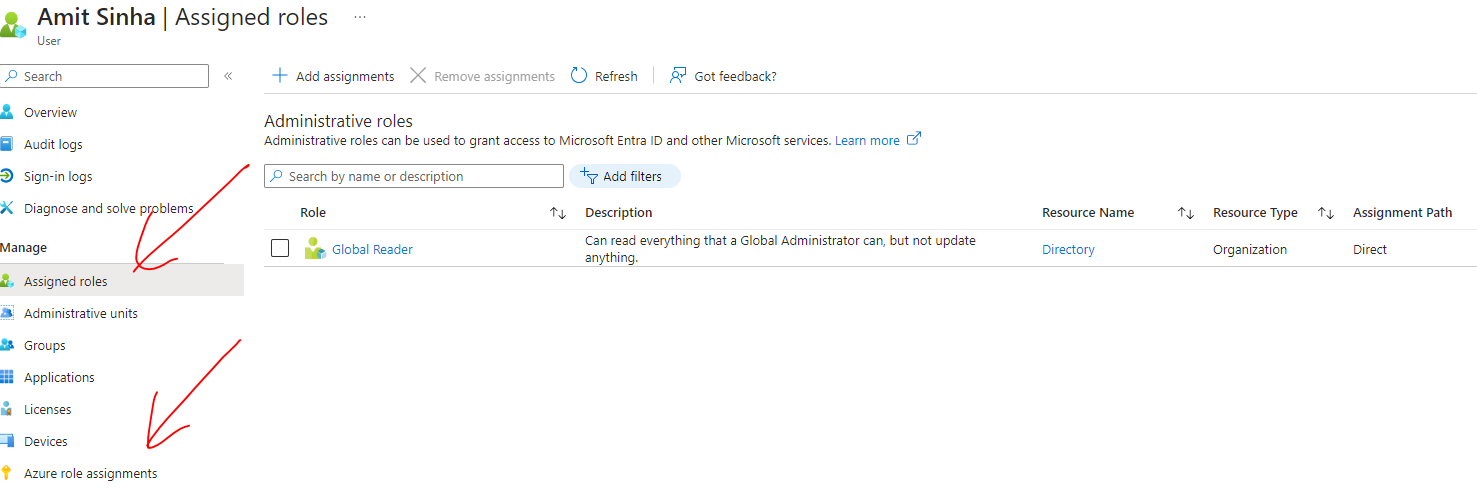
### CREATING GUEST USER



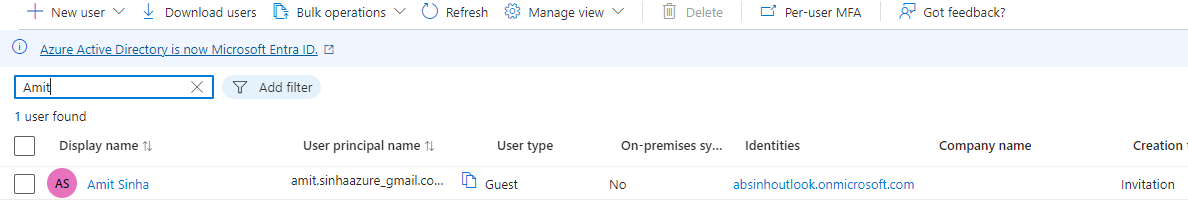


The external user can be assigned either AD roles or assign RBAC role

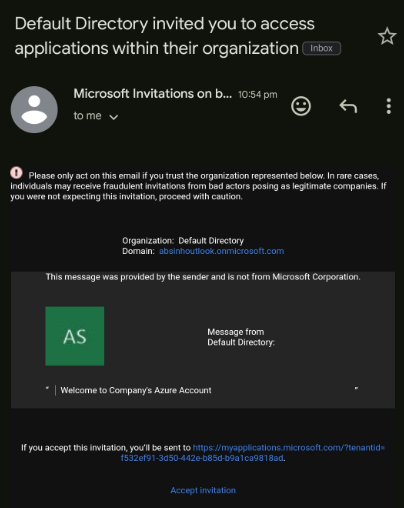




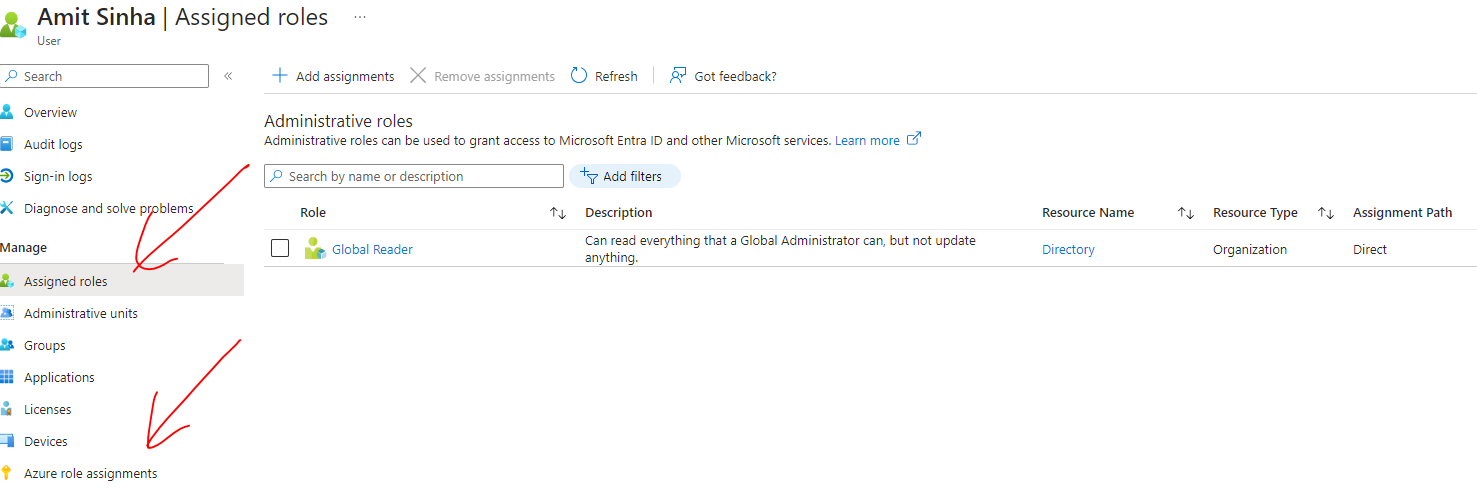
* **The user will be a guest user**



* The user will receive an email to accept the invitation - Accept the invitation 🡪 Log in to the Azure Account

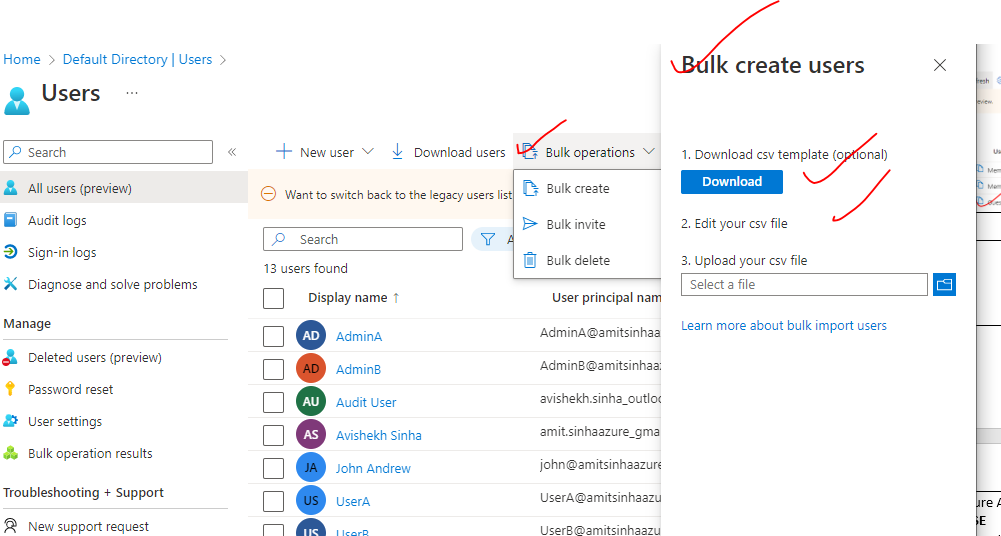


* After accepting the invitation – The guest user can login to the Azure Account.
* Initially the guest user don’t have access to any resources. Access has to be given explicitly



## USER CREATION / DELETION IN BULK

* For bulk creation/deletion of user, Navigate to Active directory 🡪 Bulk Operations
* Download the CSV and add the data to delete /create 🡪 Upload the CSV
* Sample - <https://github.com/avishekhsinhaRepo/Docs/blob/master/Azure/Documents/AD/UserCreateTemplate.csv>



## CONDITIONAL ACCESS POLICY

* Documenaation: <https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview>

## ADMINISTRATIVE UNITS

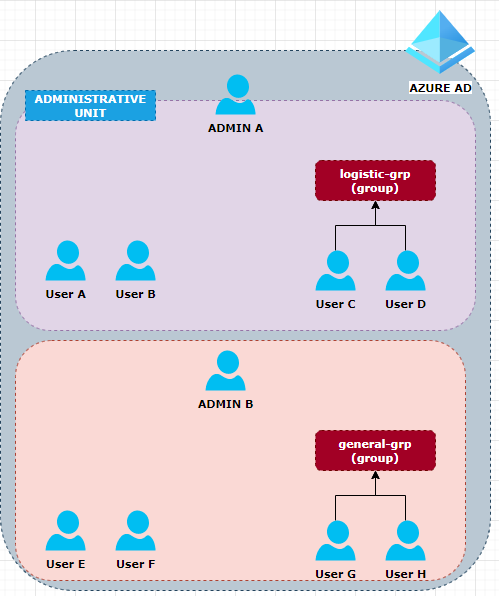
* Administrative Units in Azure Active Directory (Azure AD) provide a way to organize and manage users and resources within your organization.
* It is a logical grouping that helps us delegate administrative tasks and apply policies to specific subsets of users and resources.

**ADMINISTRATIVE UNITS ALLOW YOU TO:**

* ORGANIZE USERS
  + We can group users based on different criteria, such as department, location, or business unit, into separate Administrative Units.
  + This makes it easier to manage and delegate administrative tasks to specific administrators or helpdesk teams.
* DELEGATE ADMINISTRATION:
  + We can assign administrative roles and permissions to specific Administrative Units, allowing administrators to manage users and resources within their assigned unit without having access to the entire Azure AD tenant.
* APPLY POLICIES:
  + Administrative Units can be used to apply policies and settings, such as password policies or conditional access policies, to specific groups of users or resources.

|  |  |
| --- | --- |
|  | * In Azure AD we create uses and assign roles/permission to them   **USE-CASE**   * Let's say that we have multiple departments in a company, and we want to create a admin for each department – who will only go ahead and manage the subset of users within that particular department. * Since Azure AD is a global directory. If we start creating users in global AD – it will be difficult to manage them all * Hence to manage the users department wise – we can make use administrative units for each department. |

### LAB -ADMINISTRATIVE UNITS



**PART OF ADMINISTRATIVE UNIT**

* **Two users who are going to be part of an administrative unit + Admin User that is going to behave as the user administrator for the administrative unit** +**Two more users that are going to be part of a group(logistic-grp).**

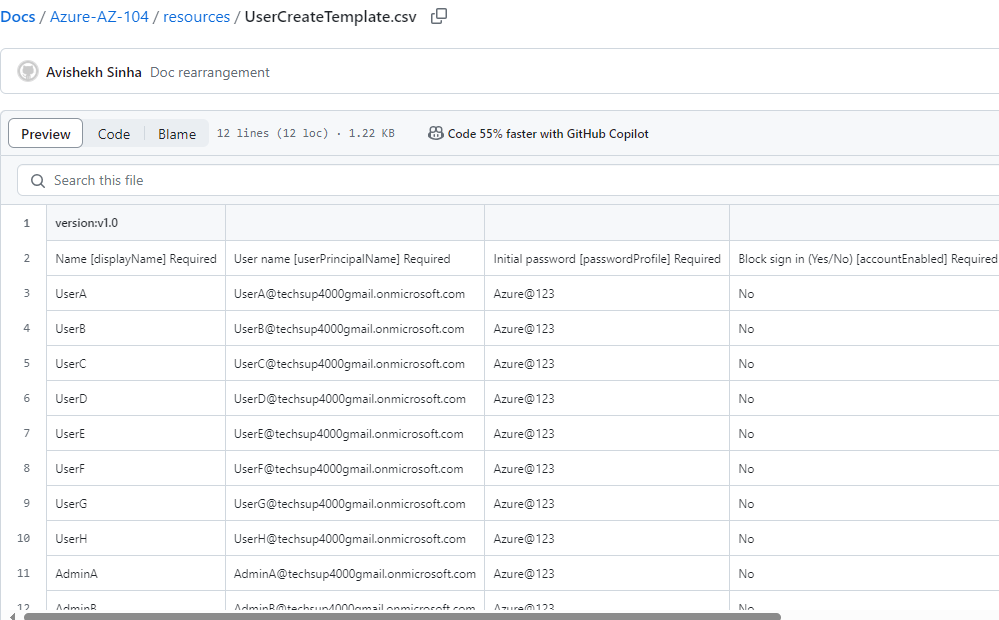
**NOT PART OF ADMINISTRATIVE UNIT**

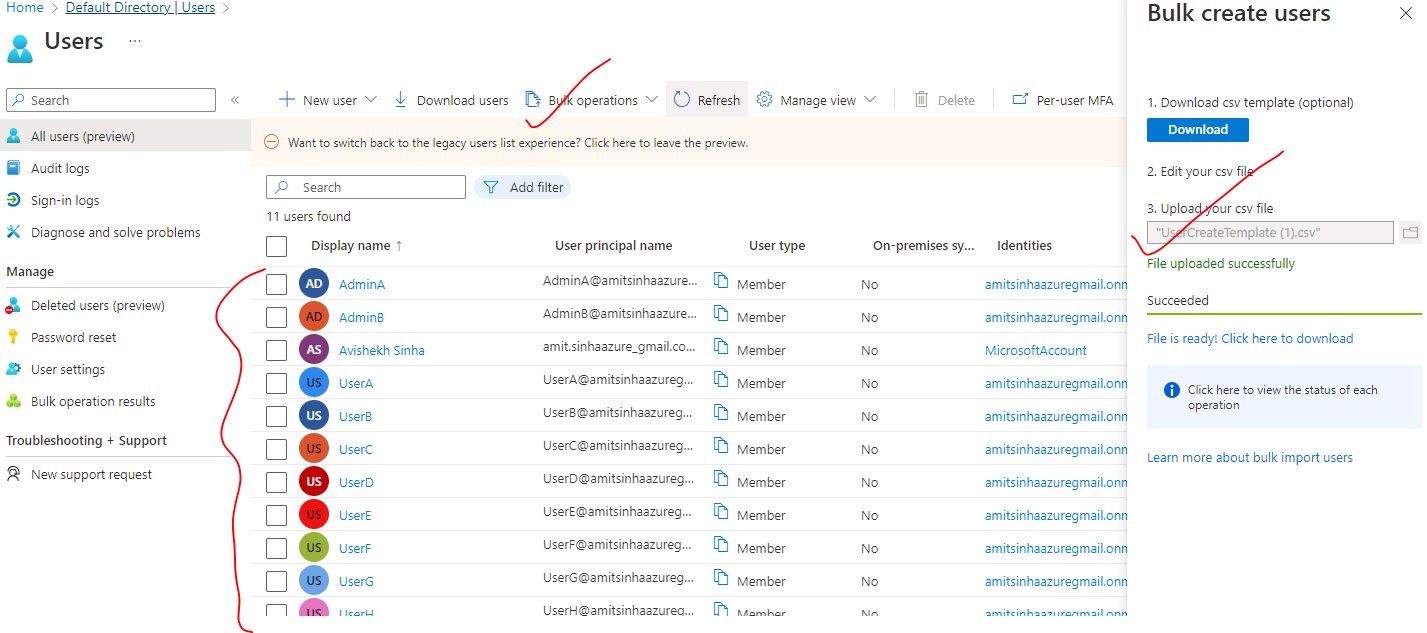
* **We're going to create some other users who are not going to be part of that unit + Some of these users are going to be part of another group known as general-grp.**

***Step 1:*** *Create user (Bulk Operation) – for the bulk creation of user Azure provides a template(as .xlsx) 🡪 Download the template 🡪 Add the data of users 🡪 Upload the sheet*

[Docs/Azure-AZ-104/resources/UserCreateTemplate.csv at master · avishekhsinhaRepo/Docs (github.com)](https://github.com/avishekhsinhaRepo/Docs/blob/master/Azure-AZ-104/resources/UserCreateTemplate.csv)

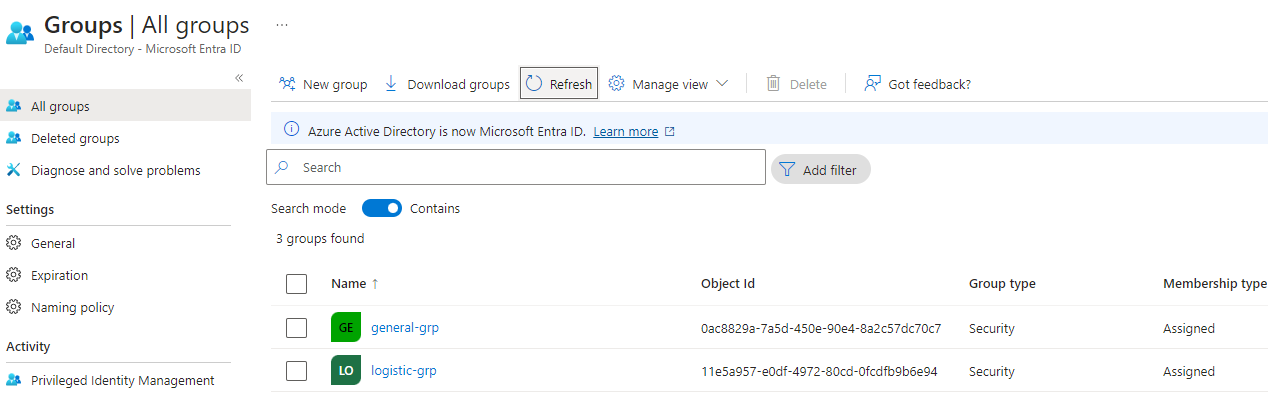
**\*\*\* update the domain name based on your domain account**





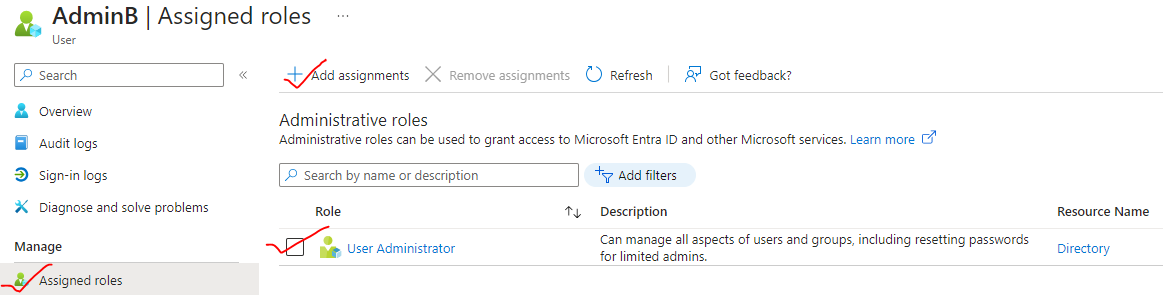
**Step 2:** Create the user groups

|  |  |
| --- | --- |
| Logistic Group – User C and User D | General Group – User G and User H |
|  |  |

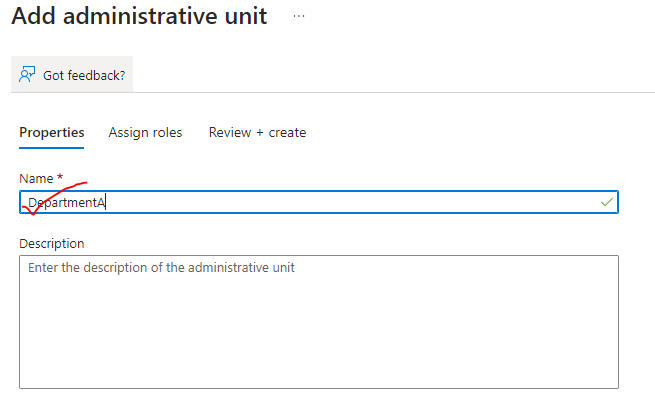


**Step 3:** ASSIGN “USER ADMINISTRATOR ROLE ” TO ADMINS

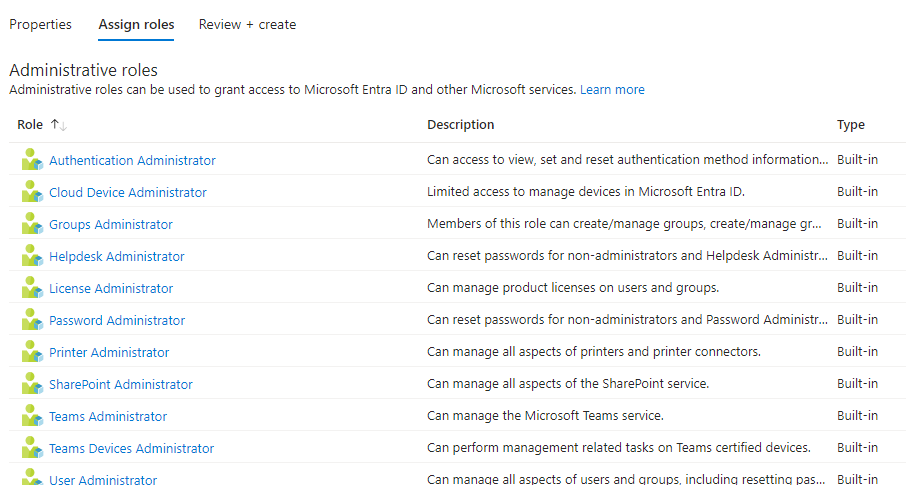
**ADMIN B**

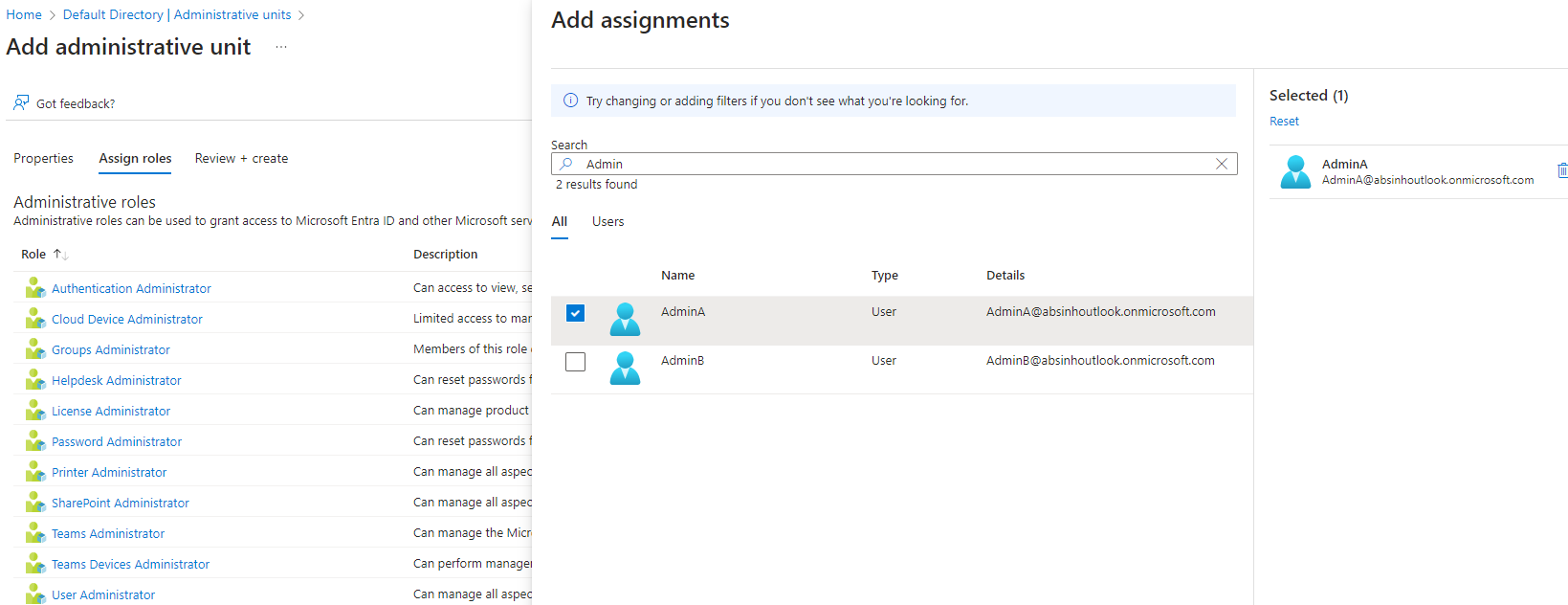


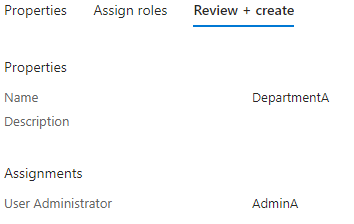
**STEP 4: CREATE THE ADMINISTRATIVE UNIT**



* There are certain roles that are available only as part of Administrative unit.
* Admin A will be assigned with “**User Administrator**” for the administrative unit







**So the entire idea is - Admin A can able to manage**

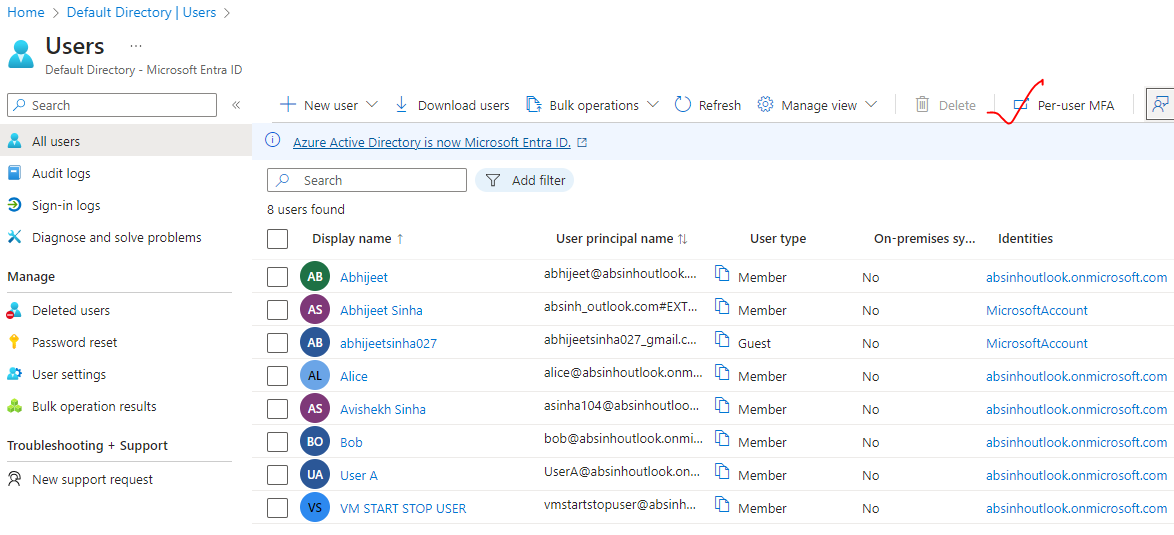
1. **Logistic Group (this group is part of Administrative Unit)**
2. **User A and User B (Will be members of Admin Unit)**
3. **But - should not have the ability to manage other users that are outside the administrative unit.**

# MULTIFACTOR AUTENTICATION(MFA)

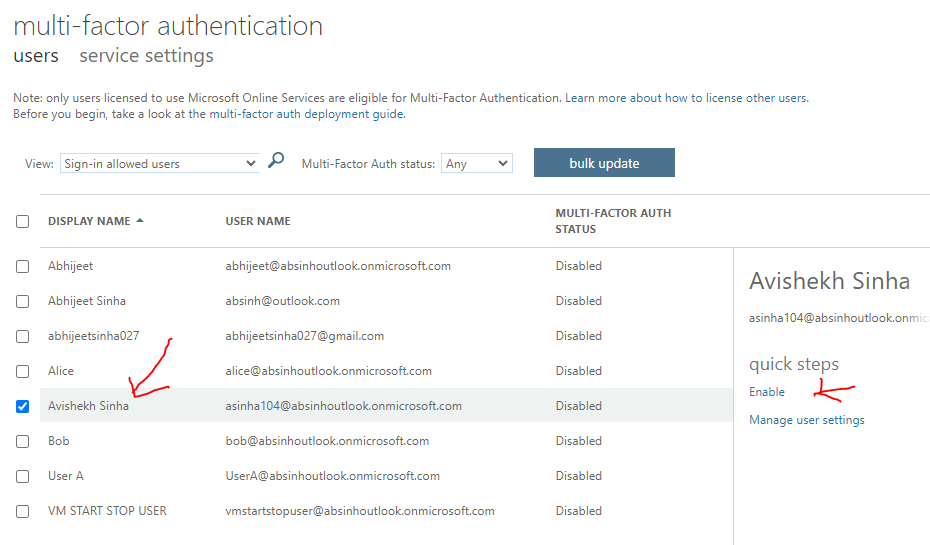
* In Azure, Multi-Factor Authentication (MFA) provides an extra layer of security to help protect user accounts from unauthorized access.
* MFA requires users to provide additional verification factors, such as a phone call, text message, or mobile app notification, in addition to their username and password.
* This helps to ensure that even if an attacker obtains a user's password, they still need the additional verification factor to gain access.

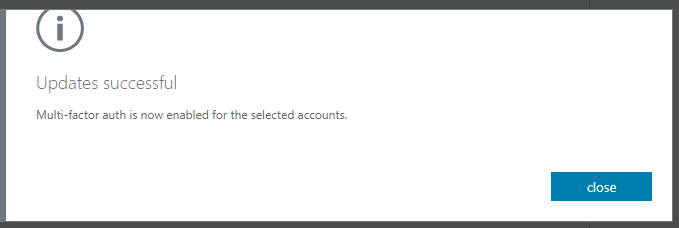
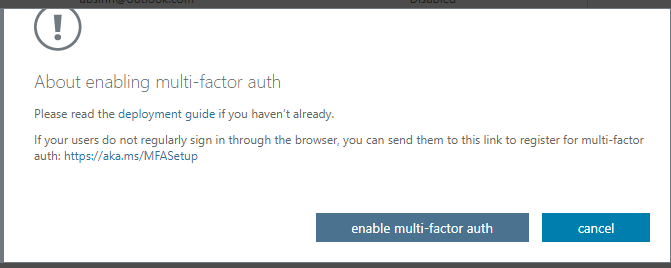
## ENABLING MFA FOR USER

* Navigate to Microsoft Entra Id 🡪 All Users 🡪 Per User MFA

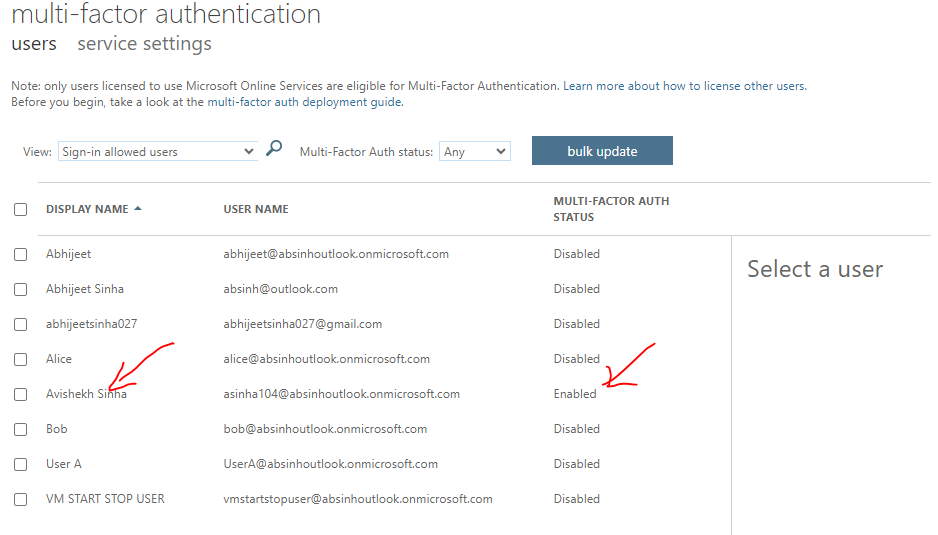


* To enable MFA for a user 🡪 Select the user 🡪 Enable🡪Enable MFA





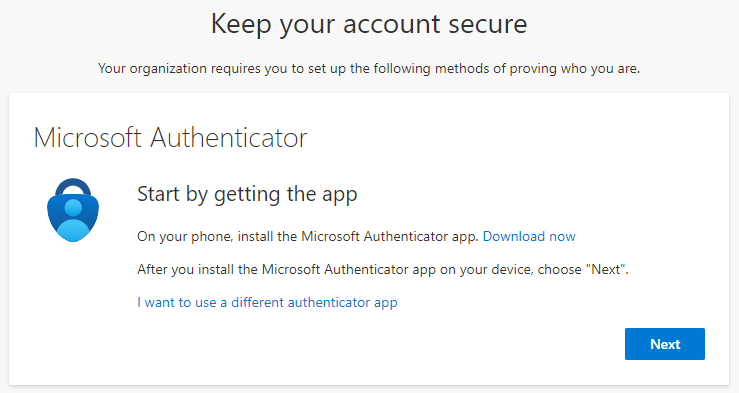
MFA ENABLED OF USER



* Let’s login with the user name for whom the MFA has been enabled. The MFA setting adds one more level of authentication. If Multi-Factor Authentication (MFA) is enabled for a user in Azure, they will need to perform additional verification steps when signing in to their account.

STEPS

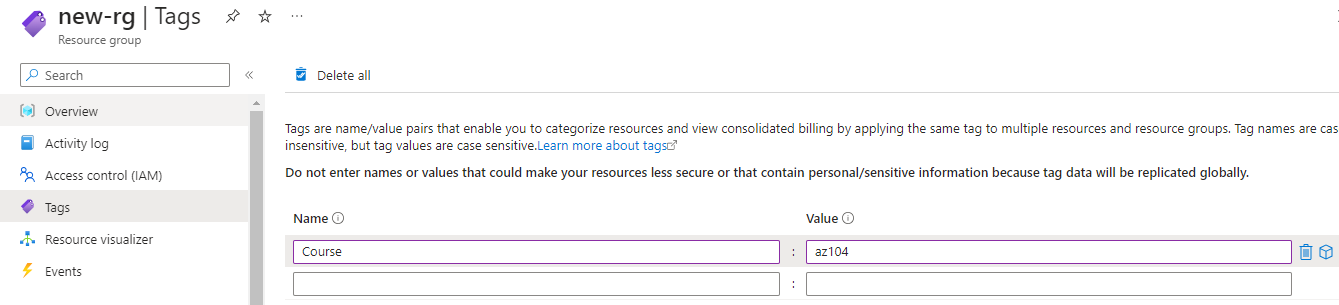
1. **Sign-In**: The user initiates the sign-in process by entering their username and password as usual.
2. **Verification Methods:** 
   1. Depending on the MFA settings and user preferences, they will be prompted to choose a verification method. This can include options like a **phone call, text message (SMS), mobile app notification**, or using a verification code generated by a mobile app or hardware token.
3. **Additional Verification**: Once the user selects their preferred verification method, they will receive a phone call, text message, or app notification, or they will generate a verification code using their mobile app or hardware token.
4. **Provide Verification** **Code**: The user will enter the verification code received through their chosen method or generated by their mobile app or hardware token.
5. **Access Granted:** If the provided verification code is correct and matches the expected code, the user will be granted access to their Azure account.



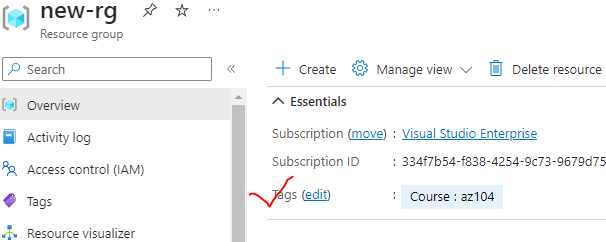
# RESOURCE TAGS

* In Azure, resource tags are metadata that we can assign to resources to organize and categorize them.
* Tags consist of name-value pairs and provide a way to add custom labels to resources.
* **We can use tags to logically group resources, track costs, apply policies, and simplify resource management**.
* Tags must be applied directly to resources and aren't implicitly inherited from the parent resource group.

**CREATING A TAG**



**TAG CREATED FOR THE RESOURCE**



**KEY POINTS ABOUT RESOURCE TAGS**

* TAG STRUCTURE:
  + Tags consist of a name and a value. The name is a string, and the value can be any string or empty.
* TAG LIMITATIONS:
  + Each resource can have multiple tags, up to a **maximum of 50 tags per resource.**
  + Tag names are case-insensitive, and the tag name-value pairs must be unique within a resource.
* ASSIGNING TAGS:
  + Tags can be assigned to resources during creation or added later.
  + We can assign tags using Azure Portal, Azure PowerShell, Azure CLI, or Azure Resource Manager templates.
* MANAGING TAGS:
  + We can manage and view tags for resources through the Azure Portal, Azure PowerShell, Azure CLI, Azure Resource Manager templates, or Azure Management APIs.
  + Tags can also be used for filtering, organizing, and querying resources.
* COST MANAGEMENT:
  + By assigning tags to resources, we can track and manage costs associated with those resources.
  + Azure Cost Management + Billing provides reporting and analysis capabilities based on resource tags.
* POLICY ENFORCEMENT:
  + Azure Policy allows us to define policies based on tags to enforce compliance and governance rules.
  + We can use policies to ensure resources have specific tags assigned or to restrict resource creation based on tags.

## COST MANAGEMENT ON TAGS

## MOVING RESOURCES ACROSS RESOURCE GROUPS

* In Azure, we can move resources across resource groups using the **Azure Portal, Azure PowerShell, Azure CLI, or Azure Resource Manager templates**. Here's how you can perform this task using each method:

Azure Portal:

1. Open the Azure Portal and navigate to the resource you want to move.
2. Select "Move" from the resource's menu.
3. Choose the "Move to another resource group" option.
4. Select the target resource group and click "OK" to initiate the move.

Azure PowerShell:

1. Open Azure PowerShell and connect to your Azure account.
2. Use the Get-AzResource cmdlet to retrieve the resource you want to move.
3. Use the Move-AzResource cmdlet to move the resource to the target resource group.  
   Example: Move-AzResource -ResourceId <resourceId> -DestinationResourceGroupName <targetResourceGroup>

Azure CLI:

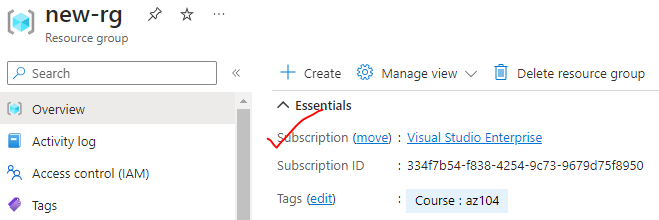
1. Open Azure CLI and sign in to your Azure account.
2. Use the az resource show command to get the details of the resource you want to move.
3. Use the az resource move command to move the resource to the target resource group.  
   Example: az resource move --ids <resourceId> --destination-group <targetResourceGroup>

## MOVING RESOURCES ACROSS SUBSCRIPTIONS

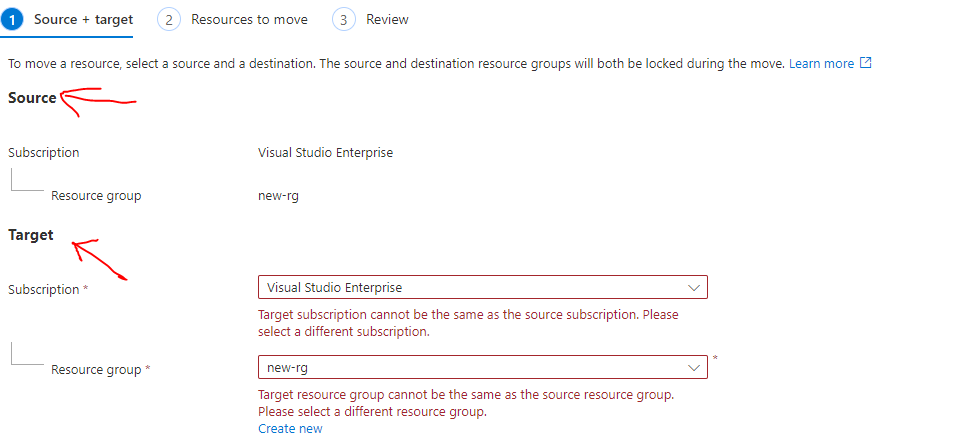
* We can move resources across subscriptions using the Azure Portal or Azure PowerShell.

USING AZURE PORTAL

* Open the Azure Portal and navigate to the resource you want to move.
* Select "Move" from the resource's menu.



* Choose the "Move to another subscription" option.
* Select the target subscription and click "OK" to initiate the move.



AZURE POWERSHELL

1. Open Azure PowerShell and connect to your Azure account.
2. Use the Get-AzResource cmdlet to retrieve the resource you want to move.
3. Use the Move-AzResource cmdlet to move the resource to the target subscription.  
   Example: Move-AzResource -ResourceId <resourceId> -DestinationSubscriptionId <targetSubscriptionId>

WHEN MOVING RESOURCES ACROSS SUBSCRIPTIONS, THERE ARE A FEW IMPORTANT CONSIDERATIONS:

* Permissions:
  + We need appropriate permissions in both the source and target subscriptions to perform the move operation.
* Limitations:
  + Not all resources can be moved across subscriptions.
  + Some resources, **like virtual networks and storage accounts**, have limitations or dependencies that may prevent the move.
* Resource Dependencies:
  + When moving resources with dependencies, we may need to move related resources as well.
  + **For example, if you're moving a virtual machine, we might need to move its associated network interface and storage account.**
* Resource Group: Moving a resource to another subscription may require moving its associated resource group as well. Make sure to consider the impact on other resources in the same resource group.

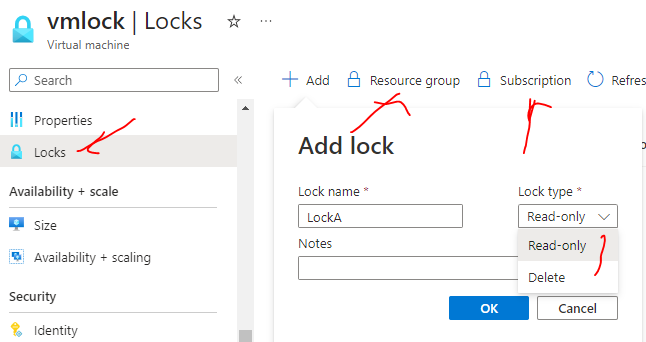
## LOCKING RESOURCES

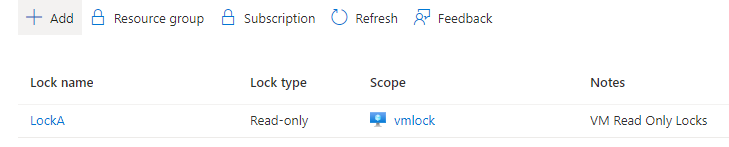
* In Azure, resource locks are a feature that allows us to **prevent accidental deletion or modification of critical resources.**
* By applying a lock to a resource or resource group, we can ensure that it cannot be deleted or modified without explicit permission. This helps in maintaining the integrity and stability of important resources.

HERE ARE A FEW KEY POINTS ABOUT RESOURCE LOCKS IN AZURE:

* TYPES OF LOCKS: There are two types of locks you can apply to resources:
  + Delete Lock (CanNotDelete): This lock prevents the resource from being deleted, but it allows other modifications.
  + Read-Only Lock(ReadOnly): This lock makes the resource read-only, preventing both deletion and modifications.
* SCOPE:
  + We can apply locks at the resource group level or at the individual resource level.
  + **Applying a lock at the resource group level automatically applies it to all resources within that group**.
* LOCK HIERARCHY:
  + Locks have a hierarchical relationship, meaning a lock applied at a parent level (resource group) is inherited by child resources unless overridden.
  + For example – the Locks on resource group level will apply the lock to all the resources in the resource group. Similarly the locks on subscription level will apply the lock to all the resources in that subscription

### SETTING UP RESOURCE LOCK





* **As “ReadOnly“ locks has been applied to the VM – we cannot perform any operation or modify any of the property of the VM.**

### LOCKS AND MOVING RESOURCES

* **If the resource has lock we still have the ability to move the resource across the resource groups.**
* **If applied a lock at the resource group level, then the lock will be inherited to the resources in the resource group. If we try to move the resource to different resource group – then we cannot move the resource because we are changing the properties of the resource group.**
* if we lock the destination resource group even then we can't move the resources to the destination resource group

## AZURE POLICY

* Azure Policy Service is a governance service in Microsoft Azure that allows us to enforce and monitor compliance with organizational standards and best practices across your Azure environment.
* It provides a centralized way to define, assign, and enforce policies that govern resource configurations and behaviors.  
    
  Here are some key features and capabilities of Azure Policy Service:
* Policy definition: Azure Policy allows you to define policies using JSON-based rules that specify the desired state and behavior of Azure resources. Policies can cover a wide range of aspects, such as resource properties, tagging, access control, network security, and more.
* Policy assignment: Once policies are defined, you can assign them to Azure subscriptions, resource groups, or management groups. This allows you to apply policies at different scopes, depending on your governance requirements.
* Compliance evaluation: Azure Policy continuously evaluates resources against assigned policies and provides compliance results. It helps identify resources that are non-compliant, allowing you to take corrective actions to bring them into compliance.
* Policy enforcement: Azure Policy can enforce compliance by blocking the creation or modification of resources that violate policy rules. It can also trigger notifications or remediation actions to rectify non-compliant resources.
* Built-in and custom policies: Azure Policy offers a range of built-in policies that cover common governance scenarios. Additionally, you can create custom policies tailored to your specific requirements using Azure Policy's JSON-based policy definition language.
* Integration with Azure DevOps and CI/CD pipelines: Azure Policy integrates with Azure DevOps and CI/CD pipelines, enabling you to include policy validation as part of your deployment and release processes.  
    
  Azure Policy Service helps you maintain control and enforce governance across your Azure environment by providing a mechanism to define, assign, and enforce policies. It promotes best practices, improves security, and ensures compliance with organizational standards and regulatory requirements.

## MANAGEMENT GROUP

* In Azure, Management Groups are a hierarchical organizational construct that allow us to manage and govern resources across multiple Azure subscriptions.
* Management Groups provide a way to apply policies, access control, and governance at scale by creating a hierarchy of groups.