Contents

[Terminology: 2](#_Toc50571641)

[Active Directory ( AD ) : 2](#_Toc50571642)

[Difference between Active Directory (AD) and Active Directory Domain Services (AD DS) 2](#_Toc50571643)

[LDAP: 3](#_Toc50571644)

[Kerberos 3](#_Toc50571645)

[Azure Active Directory (Azure AD) 3](#_Toc50571646)

[Azure AD Domain Services (Azure AD DS) 4](#_Toc50571647)

[Oauth 2 4](#_Toc50571648)

[SAML 4](#_Toc50571649)

[Managed vs. Self-Managed Domains 4](#_Toc50571650)

[Managed Domain 4](#_Toc50571651)

[Self-Managed Domain 4](#_Toc50571652)

[User Roles 5](#_Toc50571653)

[Classic subscription administrator roles 6](#_Toc50571654)

[Azure account and Azure subscriptions 7](#_Toc50571655)

[Azure roles 7](#_Toc50571656)

[Azure AD roles 8](#_Toc50571657)

[Differences between Azure roles and Azure AD roles 8](#_Toc50571658)

[Do Azure roles and Azure AD roles overlap? 9](#_Toc50571659)

[Global Administrator 9](#_Toc50571660)

[Why would you need to elevate your access? 10](#_Toc50571661)

[Azure management groups 11](#_Toc50571662)

[Important facts about management groups 11](#_Toc50571663)

[Root management group for each directory 11](#_Toc50571664)

[Important facts about the Root management group 11](#_Toc50571665)

[Management group access 12](#_Toc50571666)

[Issues with breaking the role definition and assignment hierarchy path 13](#_Toc50571667)

[Resource groups 14](#_Toc50571668)

[Azure Resource Manager 14](#_Toc50571669)

[Terminology 15](#_Toc50571670)

[Understand scope 16](#_Toc50571671)

[Practical Tips 17](#_Toc50571672)

[Azure Key vault: 18](#_Toc50571673)

# Terminology:

Comparison by MS Azure of various services

<https://docs.microsoft.com/en-us/azure/active-directory-domain-services/compare-identity-solutions>

## Active Directory ( AD ) :

<https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap/>

Active Directory is a **directory services implementation** by **Microsoft** that provides all sorts of functionality like authentication, group and user management, policy administration and more.

Active Directory (AD) supports both [Kerberos](https://www.varonis.com/blog/kerberos-authentication-explained/) and LDAP – Microsoft AD is by far the most common directory services system in use today. AD provides Single-SignOn (SSO) .

AD and Kerberos are not cross platform, which is one of the reasons companies are implementing access management software to manage logins from many different devices and platforms in a single place. AD does support LDAP, which means it can still be part of your overall access management scheme.

Active Directory is just one example of a **directory service** that supports LDAP. There are other flavors, too: Red Hat Directory Service, OpenLDAP, Apache Directory Server, and [more](https://en.wikipedia.org/wiki/Directory_service#LDAP_implementations).

### Difference between Active Directory (AD) and Active Directory Domain Services (AD DS)

In Windows Server 2000 and **Windows Server 2003 Microsoft used the term Active Directory (AD).** Starting with Windows Server 2008**, Microsoft broke down the services provided by Active Directory into individual components**, such as

* **AD DS (Active Directory Domain service)**
* **AD FS (Active Directory Federation service)**
* **AD LDS (Active Directory Lightweight Directory service)**
* **AD RMS (Active Directory Rights Management service)**
* **AD CS (Active Directory Certificate service)**

Therefore, AD DS is simply the Directory Services component of the Active Directory. Other components included in the newer editions of Windows Servers are AD Federation Services, AD Lightweight Directory Services, AD Rights Management Services, and AD Certificate Services. Together all these services fit under the AD umbrella. It’s important to note that although earlier editions of Windows Servers (2000 and 2003) didn’t use the term AD DS, the directory services are primarily the same in the newer editions of Windows Servers (starting 2008).

Note: AD servers can be installed in cloud VMs and can be accessed in networks in same domain.

## LDAP:

<https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap/>

LDAP (Lightweight Directory Access Protocol) is an open and cross platform protocol used for **directory services authentication**.

LDAP provides the communication language that applications use to communicate with other **directory services servers**. Directory services store the users, passwords, and computer accounts, and share that information with other entities on the network.

LDAP is a way of speaking to Active Directory.

LDAP is a protocol that many different directory services and access management solutions can understand.

The relationship between AD and LDAP is much like the relationship between Apache and HTTP:

* HTTP is a web protocol.
* Apache is a web server that uses the HTTP protocol.
* LDAP is a directory services protocol.
* [Active Directory](https://www.varonis.com/blog/active-directory-domain-naming-best-practices/) is a directory server that uses the LDAP protocol.

## Kerberos

Kerberos is a network **authentication protocol** designed by MIT to securely establish user identity over an insecure network. It takes care of network security issues such as password sniffing.

Usually, **Kerberos is used for user authentication and LDAP for user authorization**. Even though LDAP can also be used for authentication, Kerberos is prefered here since it is more secure. When a user logs in to a workstation, it contacts the Kerberos server to authenticate the user, and the LDAP server to get the user home and group details.

## Azure Active Directory (Azure AD)

Azure Active Directory (Azure AD) is Microsoft’s cloud-based identity and access management service.

Azure AD is a cloud-based identity service that offers the following:

* Cloud-based identification & authentication
* User and computer management
* Mobile Device Management (MDM)
* Access to Software as a service (SaaS) applications, Microsoft Azure portal, and Office 365 services

**Azure AD is primarily an identity solution**, and it is designed for Internet-based applications by using HTTP (port 80) and HTTPS (port 443) communications.

Azure AD users and groups are created in a flat structure, and there are no OUs (Organizational Unit) or GPOs (Group Policy Objects).

Azure AD cannot be queried through LDAP; instead, Azure AD uses the REST API over HTTP and HTTPS.

Azure AD does not use Kerberos authentication; instead, it uses HTTP and HTTPS protocols such as **SAML, WS-Federation, and OpenID Connect for authentication (and OAuth for authorization).**

Azure AD includes federation services, and many third-party services (such as Facebook) are federated with and trust Azure AD.

## Azure AD Domain Services (Azure AD DS)

<https://techcommunity.microsoft.com/t5/itops-talk-blog/what-are-the-differences-between-azure-active-directory-and/ba-p/917392>

**The Azure AD DS is a managed AD DS service in the cloud.** In other words, if you want the traditional AD DS running in the cloud, you can take advantage of the Azure AD DS service by running AD DS under Azure AD. This means that you will be able to use traditional AD DS features, such as Kerberos and NTLM authentication, Group Policies (which aren’t supported in Azure AD), LDAP, etc.

## Oauth 2

[OAuth 2](https://tools.ietf.org/html/rfc6749) is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, and DigitalOcean.

## SAML

Security Assertion Markup Language (SAML) is an XML-based, open-standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider.

## Managed vs. Self-Managed Domains

For organizations who are interested in running traditional AD DS services in the cloud, Microsoft offers a couple of methods. You can either use a managed domain or a self-managed domain. Here’s the difference.

### Managed Domain

A managed domain is something that you will create in the cloud using AD DS and Microsoft will create and manage the associated resources as necessary.

### Self-Managed Domain

A self-managed domain is an AD DS environment that you can create in the cloud using the traditional tools. For example, you will use Virtual Machines (VMs) to install the AD DS domain controllers, member servers, etc. This is a self-managed domain so you (not Microsoft) will be responsible for managing the domain just like you do in your on-premises environment.

# User Roles

<https://docs.microsoft.com/en-us/azure/role-based-access-control/rbac-and-directory-admin-roles>

AAA stand for Authentication, Authorization, and Access Control. In simple terms

**Authentication**: is basically, **who are you** i.e. your verification with user and password.

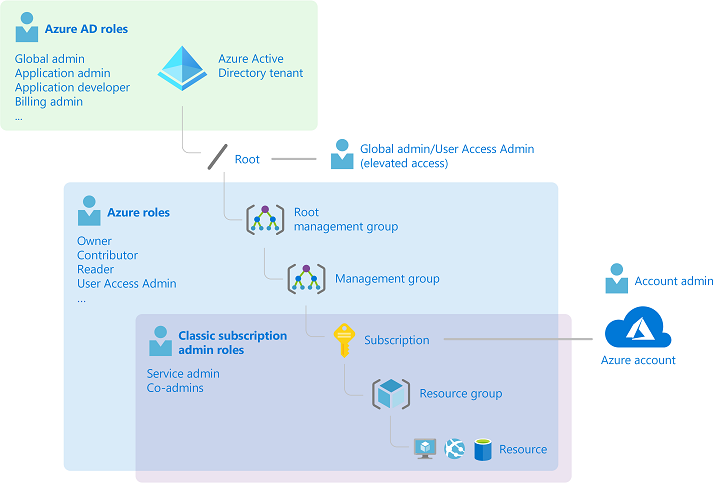
**Authorization**: is **What can you do**, i.e. A user can be admin, developer, Ops, RA user or even Guest. The level of authorization that is to be given to a user is determined by the metadata concerning the user’s account. Such data can indicate if the user is a member of the ‘Administrators’ or ‘Customers,’ or it can indicate if the user has paid-subscription for some content.

**Access Control**: Once we establish who the user is and what they can access to, we need to actively prevent that user from accessing anything they should not. Thus, we can see access control as the merger of authentication and authorization plus some additional measures like IP-based restrictions.

Main types of access control are **DAC (discretionary access control)**, **RBAC (role-based access control)**, **ABAC (attribute based access control)** and **MAC (mandatory access control)**.

Azure has three kinds of roles

* Classic subscription administrator roles
* Azure roles
* Azure Active Directory (Azure AD) roles



## Classic subscription administrator roles

Account Administrator, Service Administrator, and Co-Administrator are the three classic subscription administrator roles in Azure. Classic subscription administrators have full access to the Azure subscription.

The account that is used to sign up for Azure is automatically set as both the **Account Administrator** and **Service Administrator**. Then, additional **Co-Administrators** can be added.

| **Classic subscription administrator** | **Limit** | **Permissions** | **Notes** |
| --- | --- | --- | --- |
| Account Administrator | 1 per Azure account | * Access the [Azure Account Center](https://account.azure.com/Subscriptions) * Manage all subscriptions in an account * Create new subscriptions * Cancel subscriptions * Change the billing for a subscription * Change the Service Administrator | Conceptually, the billing owner of the subscription. The Account Administrator has no access to the Azure portal. |
| Service Administrator | 1 per Azure subscription | * Manage services in the [Azure portal](https://portal.azure.com) * Cancel the subscription * Assign users to the Co-Administrator role | By default, for a new subscription, the Account Administrator is also the Service Administrator. The Service Administrator has the equivalent access of a user who is assigned the Owner role at the subscription scope. The Service Administrator has full access to the Azure portal. |
| Co-Administrator | 200 per subscription | * Same access privileges as the Service Administrator, but can’t change the association of subscriptions to Azure directories * Assign users to the Co-Administrator role, but cannot change the Service Administrator | The Co-Administrator has the equivalent access of a user who is assigned the Owner role at the subscription scope. |

### Azure account and Azure subscriptions

An Azure account represents a billing relationship. An Azure account is a user identity, one or more Azure subscriptions, and an associated set of Azure resources.

Azure subscriptions help you organize access to Azure resources. They also help you control how resource usage is reported, billed, and paid for.

Each subscription is associated with an Azure AD directory. To find the directory the subscription is associated with, open **Subscriptions** in the Azure portal and then select a subscription to see the directory.

**Multiple subscriptions can trust the same Azure AD directory. Each subscription can only trust a single directory.**

A subscription trusts Azure AD to authenticate users, services, and devices.

Accounts and subscriptions are managed in the [Azure Account Center](https://account.azure.com/Subscriptions).

## Azure roles

Azure RBAC is an authorization system built on [Azure Resource Manager](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview) that provides fine-grained access management to Azure resources, such as compute and storage.

| **Azure role** | **Permissions** | **Notes** |
| --- | --- | --- |
| [Owner](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#owner) | * Full access to all resources * Delegate access to others | The Service Administrator and Co-Administrators are assigned the Owner role at the subscription scope Applies to all resource types. |
| [Contributor](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#contributor) | * Create and manage all of types of Azure resources * Create a new tenant in Azure Active Directory * Cannot grant access to others | Applies to all resource types. |
| [Reader](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#reader) | * View Azure resources | Applies to all resource types. |
| [User Access Administrator](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#user-access-administrator) | * Manage user access to Azure resources | The Global Administrator( AD role) will be granted the [User Access Administrator](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#user-access-administrator) role (an Azure role) on all subscriptions for a particular tenant. |

Only the Azure portal and the Azure Resource Manager APIs support Azure RBAC. Users, groups, and applications that are assigned Azure roles cannot use the [Azure classic deployment model APIs](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/deployment-models).

In the Azure portal, role assignments using Azure RBAC appear on the **Access control (IAM)** blade. This blade can be found throughout the portal, such as management groups, subscriptions, resource groups, and various resources.

## Azure AD roles

Azure AD roles are used to manage Azure AD resources in a directory such as create or edit users, assign administrative roles to others, reset user passwords, manage user licenses, and manage domains.

| **Azure AD role** | **Permissions** | **Notes** |
| --- | --- | --- |
| [Global Administrator](https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-assign-admin-roles#company-administrator-permissions) | * Manage access to all administrative features in Azure Active Directory, as well as services that federate to Azure Active Directory * Assign administrator roles to others * Reset the password for any user and all other administrators | The person who signs up for the Azure Active Directory tenant becomes a Global Administrator. |
| [User Administrator](https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-assign-admin-roles#user-administrator) | * Create and manage all aspects of users and groups * Manage support tickets * Monitor service health * Change passwords for users, Helpdesk administrators, and other User Administrators |  |
| [Billing Administrator](https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-assign-admin-roles#billing-administrator) | * Make purchases * Manage subscriptions * Manage support tickets * Monitors service health |  |

In the Azure portal, you can see the list of Azure AD roles on the **Roles and administrators** blade.

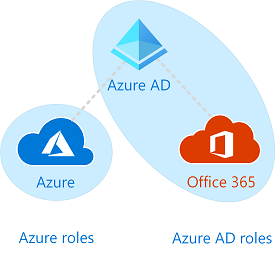
### Differences between Azure roles and Azure AD roles

At a high level, Azure roles control permissions to manage Azure resources, while Azure AD roles control permissions to manage Azure Active Directory resources.

| **Azure roles** | **Azure AD roles** |
| --- | --- |
| Manage access to Azure resources | Manage access to Azure Active Directory resources |
| Supports custom roles | Supports custom roles |
| Scope can be specified at multiple levels (management group, subscription, resource group, resource) | Scope is at the tenant level |
| Role information can be accessed in Azure portal, Azure CLI, Azure PowerShell, Azure Resource Manager templates, REST API | Role information can be accessed in Azure admin portal, Microsoft 365 admin center, Microsoft Graph, AzureAD PowerShell |

### Do Azure roles and Azure AD roles overlap?

By default, Azure roles and Azure AD roles do not span Azure and Azure AD. However, if a Global Administrator elevates their access by choosing the **Access management for Azure resources** switch in the Azure portal, the Global Administrator will be granted the [User Access Administrator](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#user-access-administrator) role (an Azure role) on all subscriptions for a particular tenant. The User Access Administrator role enables the user to grant other users access to Azure resources. This switch can be helpful to regain access to a subscription. For more information, see [Elevate access to manage all Azure subscriptions and management groups](https://docs.microsoft.com/en-us/azure/role-based-access-control/elevate-access-global-admin).



Several Azure AD roles span Azure AD and Microsoft Office 365, such as the Global Administrator and User Administrator roles. For example, if you are a member of the Global Administrator role, you have global administrator capabilities in Azure AD and Office 365, such as making changes to Microsoft Exchange and Microsoft SharePoint. However, by default, the Global Administrator doesn't have access to Azure resources.

## Global Administrator

<https://docs.microsoft.com/en-us/azure/role-based-access-control/elevate-access-global-admin>

Once you have created a Azure account make one user as Global Administrator, this acts as like a root user in Linux system which has access to everything.

When you elevate your access, you will be assigned the [User Access Administrator](https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#user-access-administrator) role in Azure at root scope (/). This allows you to view all resources and assign access in any subscription or management group in the directory. User Access Administrator role assignments can be removed using Azure PowerShell, Azure CLI, or the REST API.

You should remove this elevated access once you have made the changes you need to make at root scope.

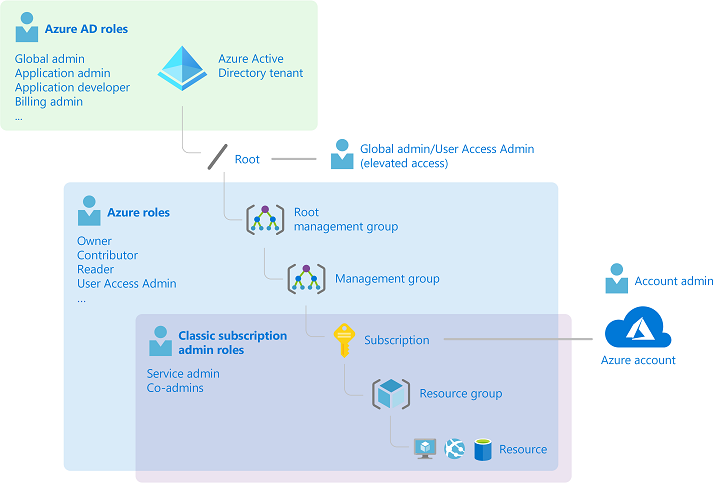
This feature is only available to users who are assigned the **Global Administrator** role in Azure AD.

When you view the Access control (IAM) pane, you'll notice that you have been assigned the **User Access Administrator** role at root scope.

### Why would you need to elevate your access?

If you are a Global Administrator, there might be times when you want to do the following actions:

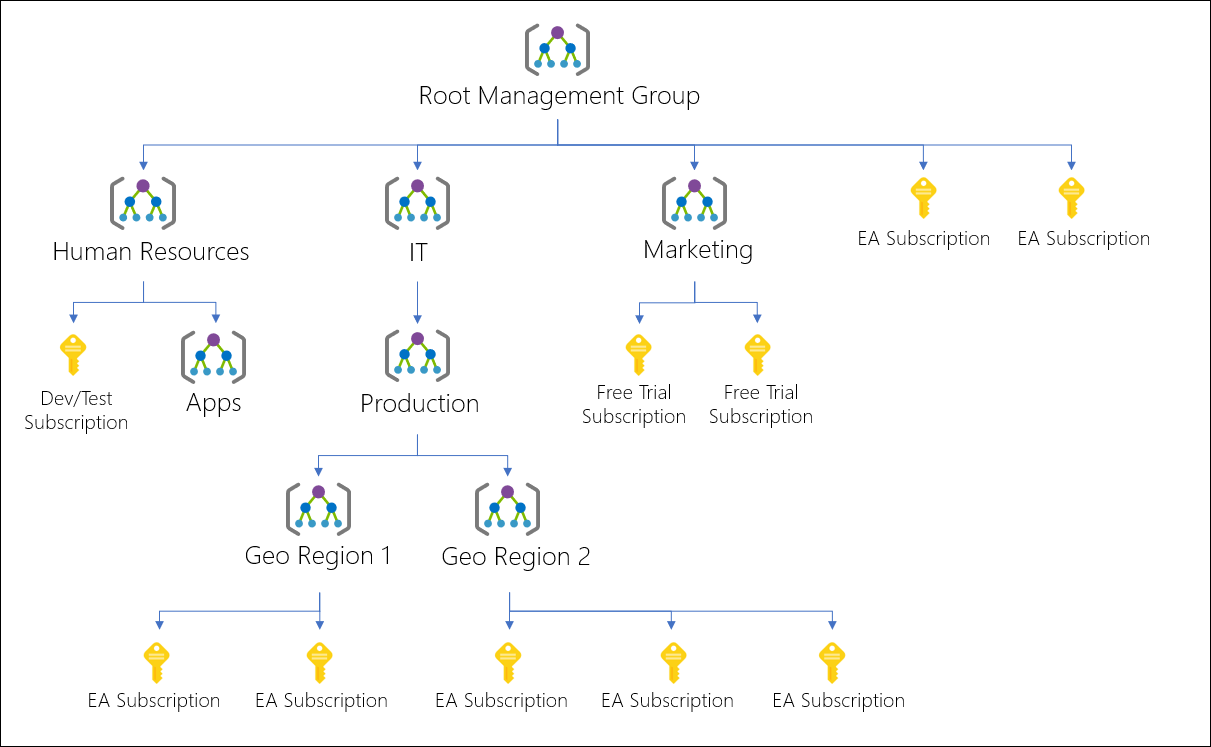
* Regain access to an Azure subscription or management group when a user has lost access
* Grant another user or yourself access to an Azure subscription or management group
* See all Azure subscriptions or management groups in an organization
* Allow an automation app (such as an invoicing or auditing app) to access all Azure subscriptions or management groups



# Azure management groups

## Important facts about management groups

* 10,000 management groups can be supported in a single directory.
* A management group tree can support up to six levels of depth.
  + This limit doesn't include the Root level or the subscription level.
* Each management group and subscription can only support one parent.
* Each management group can have many children.
* All subscriptions and management groups are within a single hierarchy in each directory.



## Root management group for each directory

Each directory is given a single top-level management group called the "Root" management group. This root management group is built into the hierarchy to have all management groups and subscriptions fold up to it. This root management group allows for global policies and Azure role assignments to be applied at the directory level.

### Important facts about the Root management group

* By default, the root management group's display name is **Tenant root group**. The ID is the Azure Active Directory ID.
* To change the display name, your account must be assigned the Owner or Contributor role on the root management group. See [Change the name of a management group](https://docs.microsoft.com/en-us/azure/governance/management-groups/manage#change-the-name-of-a-management-group) to update the name of a management group.
* **The root management group can't be moved or deleted, unlike other management groups**.
* **All subscriptions and management groups fold up to the one root management group** within the directory.
  + All resources in the directory fold up to the root management group for global management.
  + New subscriptions are automatically defaulted to the root management group when created.
* **All Azure customers can see the root management group**, but not all customers have access to manage that root management group.
  + Everyone who has access to a subscription can see the context of where that subscription is in the hierarchy.
  + No one is given default access to the root management group. Azure AD Global Administrators are the only users that can elevate themselves to gain access. Once they have access to the root management group, the global administrators can assign any Azure role to other users to manage  
    it.
* In SDK, the root management group, or 'Tenant Root', operates as a management group.

<https://www.youtube.com/watch?v=jOprhCxnEAg>

## Management group access

Azure management groups support [Azure role-based access control (Azure RBAC)](https://docs.microsoft.com/en-us/azure/role-based-access-control/overview) for all resource accesses and role definitions. These permissions are inherited to child resources that exist in the hierarchy. Any Azure role can be assigned to a management group that will inherit down the hierarchy to the resources.

The following chart shows the list of roles and the supported actions on management groups.

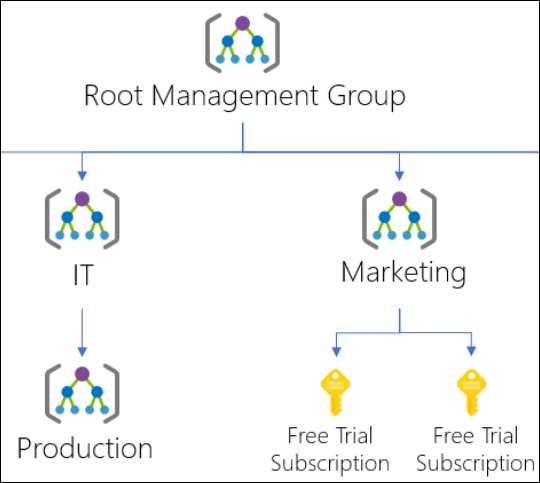
| Management group access | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Azure Role Name** | **Create** | **Rename** | **Move\*\*** | **Delete** | **Assign Access** | **Assign Policy** | **Read** |
| Owner | X | X | X | X | X | X | X |
| Contributor | X | X | X | X |  |  | X |
| MG Contributor\* | X | X | X | X |  |  | X |
| Reader |  |  |  |  |  |  | X |
| MG Reader\* |  |  |  |  |  |  | X |
| Resource Policy Contributor |  |  |  |  |  | X |  |
| User Access Administrator |  |  |  |  | X | X |  |

\*: MG Contributor and MG Reader only allow users to do those actions on the management group scope.  
\*\*: Role Assignments on the Root management group aren't required to move a subscription or management group to and from it. See [Manage your resources with management groups](https://docs.microsoft.com/en-us/azure/governance/management-groups/manage) for details on moving items within the hierarchy.

### Issues with breaking the role definition and assignment hierarchy path

Role definitions are assignable scope anywhere within the management group hierarchy. A role definition can be defined on a parent management group while the actual role assignment exists on the child subscription. Since there's a relationship between the two items, you'll receive an error when trying to separate the assignment from its definition.

For example, let's look at a small section of a hierarchy for a visual.



Let's say there's a custom role defined on the Marketing management group. That custom role is then assigned on the two free trial subscriptions.

If we try to move one of those subscriptions to be a child of the Production management group, this move would break the path from subscription role assignment to the Marketing management group role definition. In this scenario, you'll receive an error saying the move isn't allowed since it will break this relationship.

There are a couple different options to fix this scenario:

* Remove the role assignment from the subscription before moving the subscription to a new parent MG.
* Add the subscription to the Role Definition's assignable scope.
* Change the assignable scope within the role definition. In the above example, you can update the assignable scopes from Marketing to Root Management Group so that the definition can be reached by both branches of the hierarchy.
* Create an additional Custom Role that will be defined in the other branch. This new role will require the role assignment to be changed on the subscription also.

**Task :**

<https://docs.microsoft.com/en-us/azure/governance/management-groups/overview#moving-management-groups-and-subscriptions>

# Resource groups

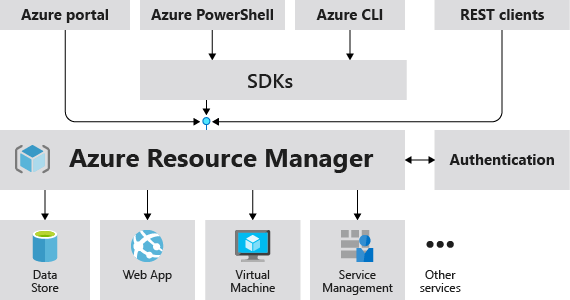
There are some important factors to consider when defining your resource group:

* All the resources in your group should share the same lifecycle. You deploy, update, and delete them together. If one resource, such as a server, needs to exist on a different deployment cycle it should be in another resource group.
* Each resource can only exist in one resource group.
* Some resources can exist outside of a resource group. These resources are deployed to the [subscription](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-subscription), [management group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-management-group), or [tenant](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-tenant). Only specific resource types are supported at these scopes.
* You can add or remove a resource to a resource group at any time.
* You can move a resource from one resource group to another group. For more information, see [Move resources to new resource group or subscription](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/move-resource-group-and-subscription).
* A resource group can contain resources that are located in different regions.
* A resource group can be used to scope access control for administrative actions.
* A resource can interact with resources in other resource groups. This interaction is common when the two resources are related but don't share the same lifecycle (for example, web apps connecting to a database).

# Azure Resource Manager

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account.

When a user sends a request from any of the Azure tools, APIs, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.



## Terminology

**resource** - A manageable item that is available through Azure. Virtual machines, storage accounts, web apps, databases, and virtual networks are examples of resources. Resource groups, subscriptions, management groups, and tags are also examples of resources.

**resource group** - A container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for your organization.

**resource provider** - A service that supplies Azure resources. For example, a common resource provider is **Microsoft.Compute**, which supplies the virtual machine resource. **Microsoft.Storage** is another common resource provider. See [Resource providers and types](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/resource-providers-and-types).

**Resource Manager template** - A JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group, subscription, management group, or tenant.

**declarative syntax** - Syntax that lets you state "Here is what I intend to create" without having to write the sequence of programming commands to create it. The Resource Manager template is an example of declarative syntax.

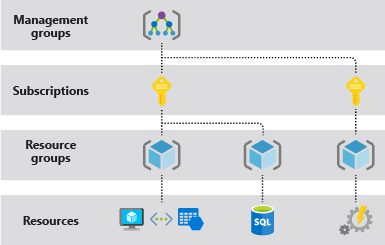
**tenant** - A Tenant, as it relates to Azure, refers to a single instance of [Azure Active Directory](https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis), or, as it is often called “Azure AD”. Azure AD is a key piece of Microsoft’s cloud platform as it provides a single place to manage users, groups and the permissions they hold in relation to applications published in Azure AD.

Azure AD Tenants are globally unique and are scoped using a domain that ends with ‘onmicrosoft.com’ (i.e. myazuread.onmicrosoft.com) and each has a ‘Tenant ID’ in the form of an [UUID/GUID](https://en.wikipedia.org/wiki/Universally_unique_identifier). Some customers choose to connect their internal Active Directory environment to Azure AD to allow single or same sign-on for their staff and will also use a custom domain instead of the default ‘onmicrosoft.com’.

Finally, **Azure AD Tenants can be associated with multiple Subscriptions (typically in larger organisations), but a Subscription can only ever be associated with a single Azure AD Tenant at any time.**

## Understand scope

Azure provides four levels of scope: [management groups](https://docs.microsoft.com/en-us/azure/governance/management-groups/overview), subscriptions, [resource groups](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview#resource-groups), and resources. The following image shows an example of these layers.

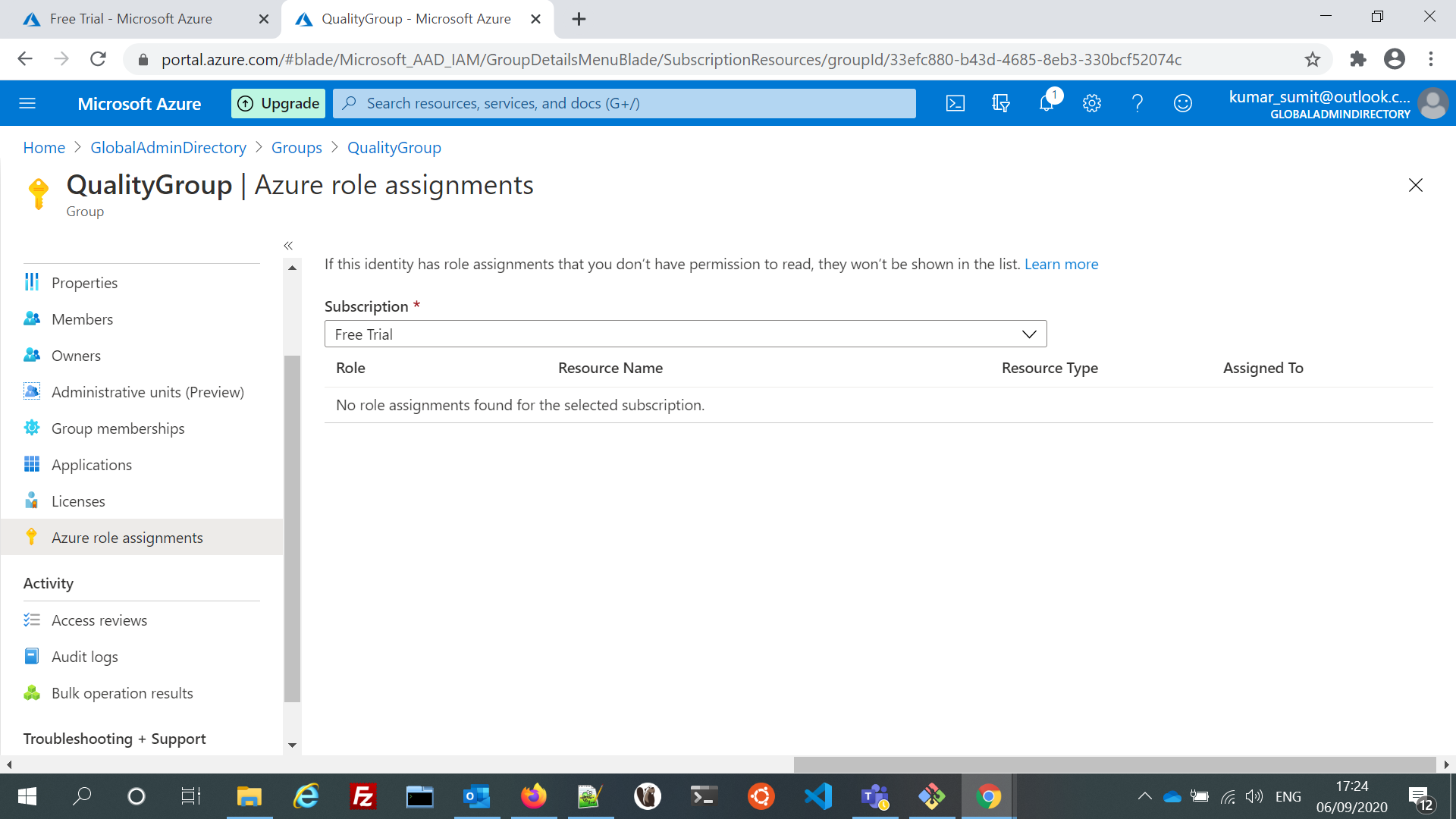


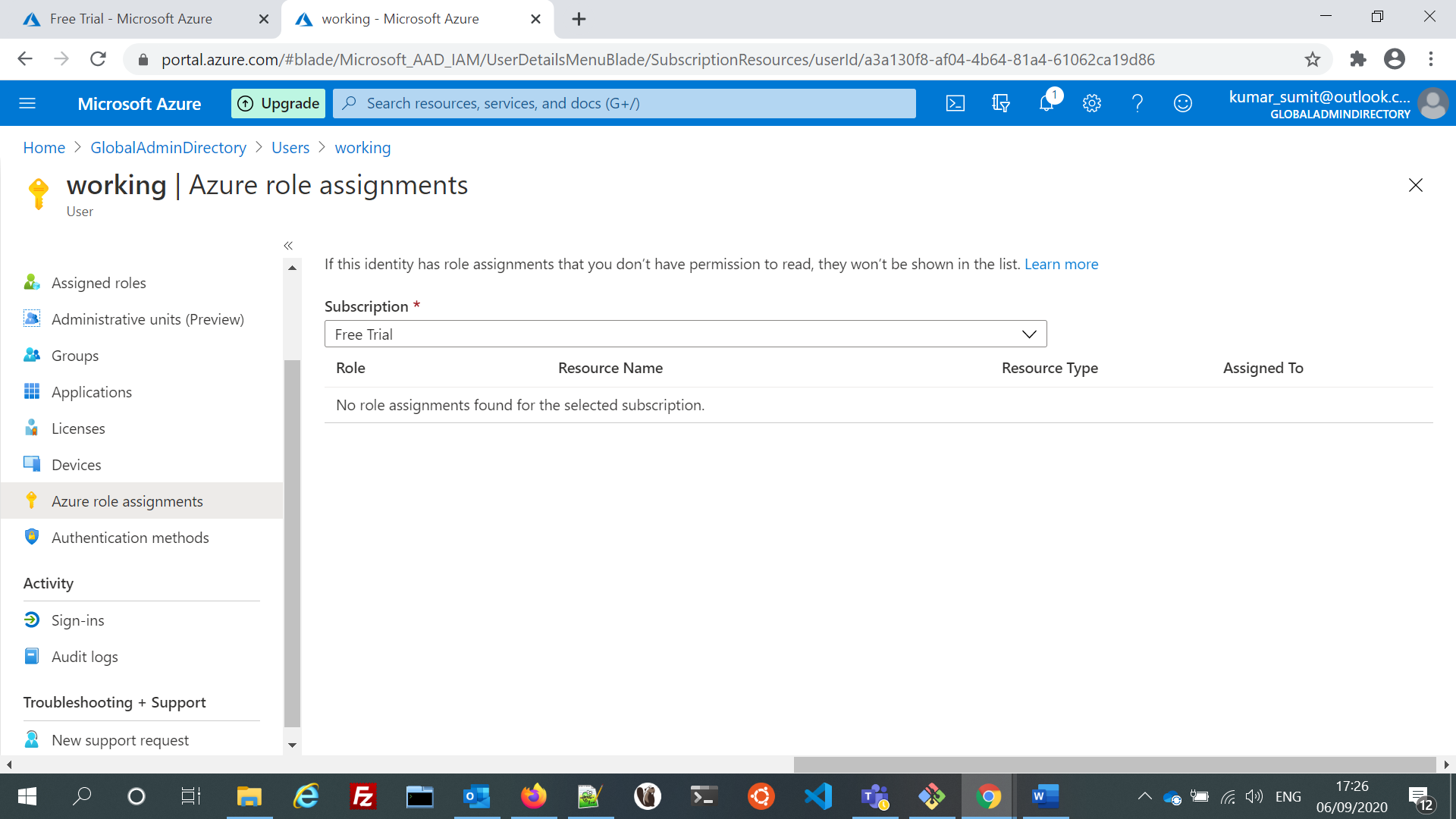
You apply management settings at any of these levels of scope. The level you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For example, when you apply a [policy](https://docs.microsoft.com/en-us/azure/governance/policy/overview) to the subscription, the policy is applied to all resource groups and resources in your subscription.

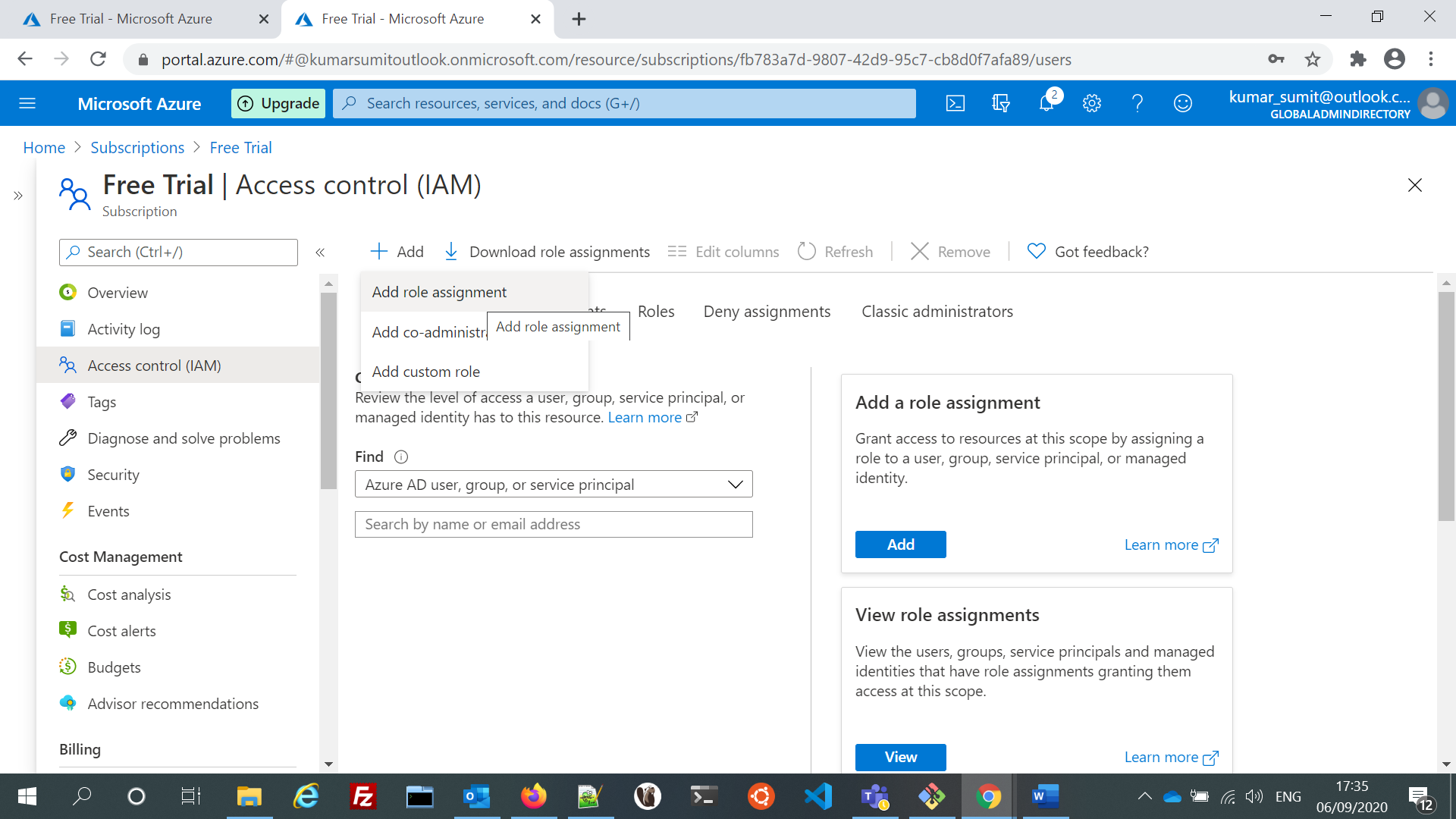
<https://www.youtube.com/watch?v=qFoHDTxkQII>

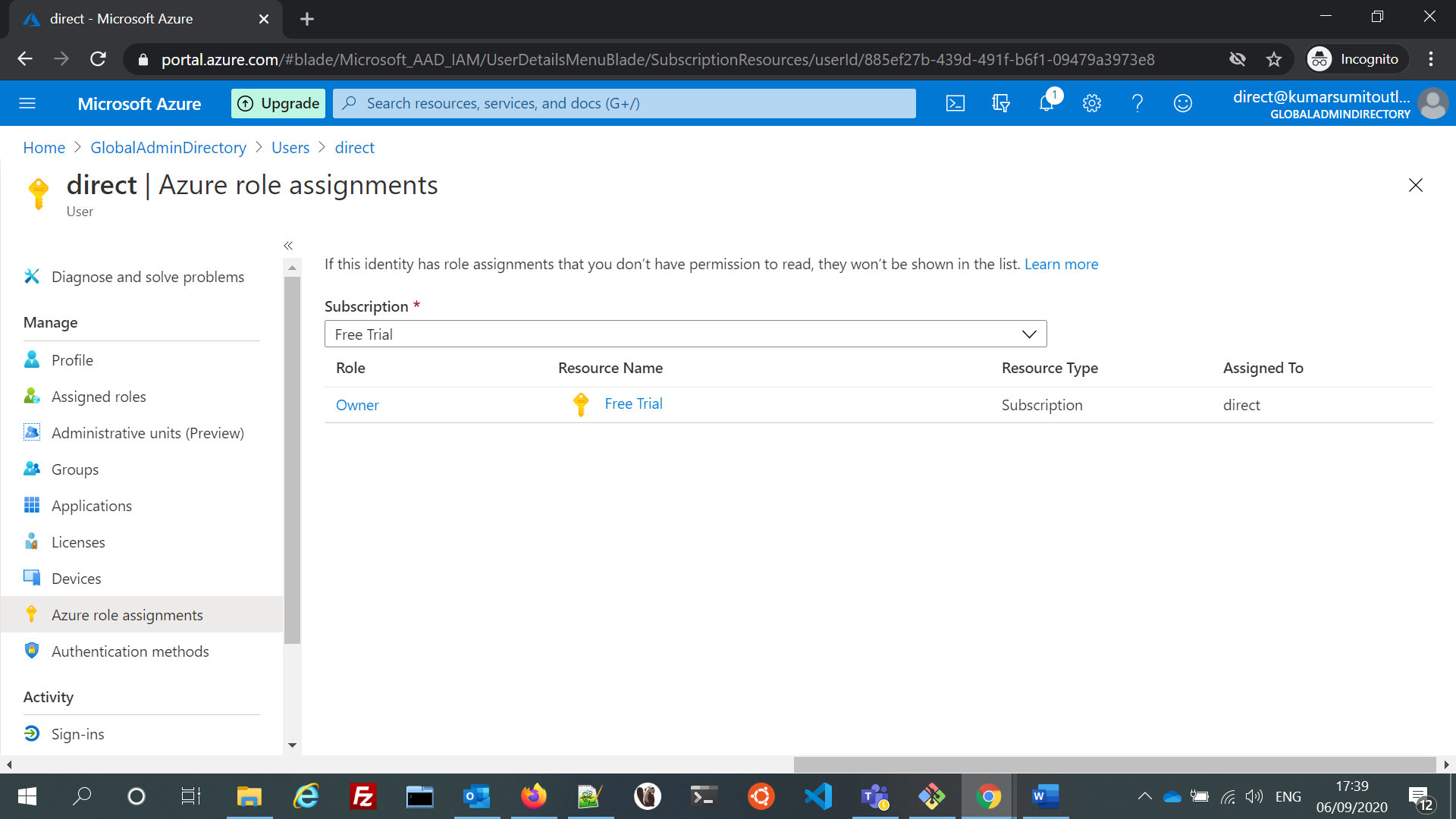
# Practical Tips

After creating user and groups from Azure Active Directory, we will have to assign a role to the user/group. Keep in mind that roles are related to subscription. As you can see in screenshots of Azure role assignments, there are no roles assigned and it is being filtered on subscription.









# Azure Key vault:

<https://www.youtube.com/watch?v=T0zpfInK7Kw>