

Domain 2: Manage identities and governance in Azure

Understand Microsoft Entra ID

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/1-introduction>

Introduction

Completed

Welcome to the Microsoft Entra ID learning module! Microsoft Entra ID is a cloud-based identity and access management service provided by Microsoft. Microsoft Entra ID is a comprehensive solution for managing identities, enforcing access policies, and securing your applications and data in the cloud and on-premises.

This module aims to equip you with a comprehensive understanding of the following:

- Describe Microsoft Entra ID.
- Compare Microsoft Entra ID to Active Directory Domain Services (AD DS).
- Describe how Microsoft Entra ID is used as a directory for cloud apps.
- Describe Microsoft Entra ID P1 and P2.
- Describe Microsoft Entra Domain Services.

Whether you're a beginner or an experienced IT professional, this module provides you with the knowledge and skills necessary to understand Microsoft Entra ID effectively. So, let's explore the exciting world of Microsoft Entra ID!

2. Examine Microsoft Entra ID

[https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/2-examine-azure-active-d
irectory](https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/2-examine-azure-active-directory)

Examine Microsoft Entra ID

Completed

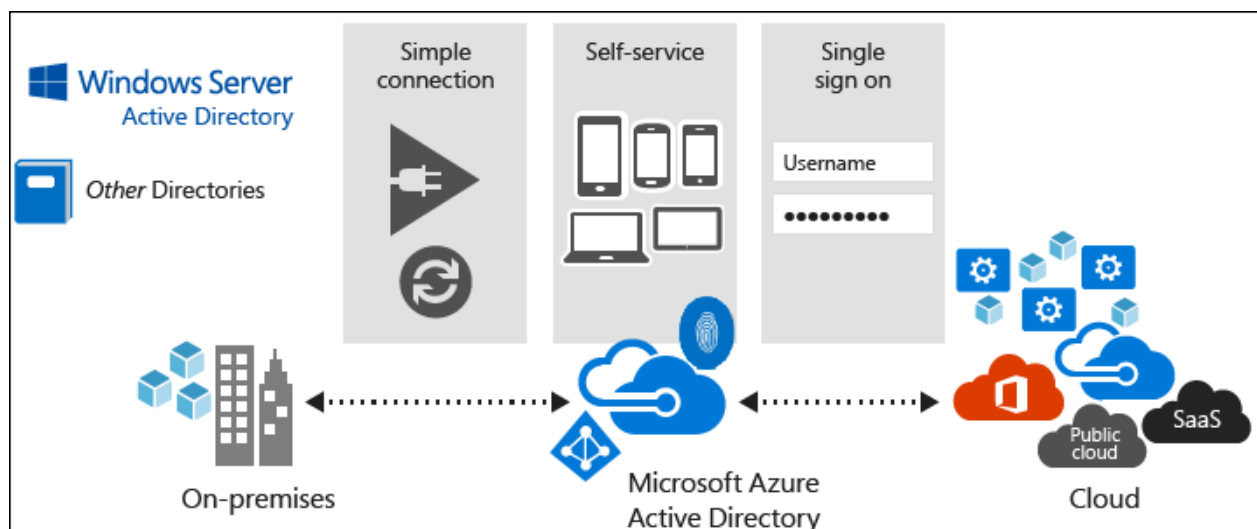
Students should be familiar with Active Directory Domain Services (AD DS or traditionally called just "Active Directory"). AD DS is a directory service that provides the methods for storing directory data, such as user accounts and passwords, and makes this data available to network users, administrators, 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.

You can use Microsoft Entra ID to provide more secure access to cloud-based resources for organizations and individuals by:

- Configuring access to applications
- Configuring single sign-on (SSO) to cloud-based SaaS applications
- Managing users and groups
- Provisioning users
- Enabling federation between organizations
- Providing an identity management solution
- Identifying irregular sign-in activity
- Configuring multi-factor authentication
- Extending existing on-premises Active Directory implementations to Microsoft Entra ID
- Configuring Application Proxy for cloud and local applications
- Configuring Conditional Access for users and devices



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.

Note

By default, when you create a new Azure subscription by using a Microsoft account, the subscription automatically includes a new Microsoft Entra tenant named Default Directory.

Some of the more advanced identity management features require paid versions of Microsoft Entra ID, offered in the form of Basic and Premium tiers. Some of these features are also automatically included in Microsoft Entra instances generated as part of Microsoft 365 subscriptions. Differences between Microsoft Entra versions are discussed later in this module.

Implementing Microsoft Entra ID isn't the same as deploying virtual machines in Azure, adding AD DS, and then deploying some domain controllers for a new forest and domain. Microsoft Entra ID is a different service, much more focused on providing identity management services to web-based apps, unlike AD DS, which is more focused on on-premises apps.

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. It's the world's largest multi-tenant directory, hosting over a million directory services instances, with billions of authentication requests per week. 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.

At any given time, an 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.

Note

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. The Microsoft Entra tenant serves as the security boundary and a container for Microsoft Entra objects such as users, groups, and applications. A single Microsoft Entra tenant can support multiple Azure subscriptions.

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. The process of joining devices to Microsoft Entra differs considerably from the process of joining computers to AD DS. The Microsoft Entra schema is also easily extensible, and its extensions are fully reversible.

The lack of support for the traditional computer domain membership means that you can't use Microsoft Entra ID to manage computers or user settings by using traditional management techniques, such as Group Policy Objects (GPOs). Instead, Microsoft Entra ID and its services define a concept of modern management. Microsoft Entra ID's primary strength lies in providing directory services; storing and publishing user, device, and application data; and handling the authentication and authorization of the users, devices, and applications. The effectiveness and efficiency of these features are apparent based on existing deployments of cloud services such as Microsoft 365, which rely on Microsoft Entra ID as their identity provider and support millions of users.

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. However, this isn't a significant shortcoming, because OUs in AD DS are used primarily for Group Policy scoping and delegation. You can accomplish equivalent arrangements by organizing objects based on their group membership.

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.

3. Compare Microsoft Entra ID and Active Directory Domain Services

Compare Microsoft Entra ID and Active Directory Domain Services

Completed

You could view Microsoft Entra ID simply as the cloud-based counterpart of AD DS; however, while Microsoft Entra ID and AD DS share some common characteristics, there are several significant differences between them.

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.

Note

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.

4. Examine Microsoft Entra ID as a directory service for cloud apps

<https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/4-examine-azure-directory-service-cloud-apps>

Examine Microsoft Entra ID as a directory service for cloud apps

Completed

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.

The process of implementing Microsoft Entra ID support for custom applications is rather complex and beyond the scope of this course. However, the Azure portal and Microsoft Visual Studio 2013 and later make the process of configuring such support more straightforward.

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.

For more information, see [Configure your App Service app to use Microsoft Entra login](#).

5. Compare Microsoft Entra ID P1 and P2 plans

<https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/5-compare-azure-premium-p1-p2-plans>

Compare Microsoft Entra ID P1 and P2 plans

Completed

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.** It simplifies the administration of groups where users are given the rights to create and manage the groups. End users can create requests to join other groups, and group owners can approve requests and maintain their groups' memberships.

- **Advanced security reports and alerts.** You can monitor and protect access to your cloud applications by viewing detailed logs that show advanced anomalies and inconsistent access pattern reports. Advanced reports are machine learning based and can help you gain new insights to improve access security and respond to potential threats.
- **Multi-factor authentication.** Full multi-factor authentication (MFA) works with on-premises applications (using virtual private network [VPN], RADIUS, and others), Azure, Microsoft 365, Dynamics 365, and third-party Microsoft Entra gallery applications. It doesn't work with non-browser off-the-shelf apps, such as Microsoft Outlook. Full multi-factor authentication is covered in more detail in the following units in this lesson.
- **Microsoft Identity Manager (MIM) licensing.** MIM integrates with Microsoft Entra ID P1 or P2 to provide hybrid identity solutions. MIM can bridge multiple on-premises authentication stores such as AD DS, LDAP, Oracle, and other applications with Microsoft Entra ID. This provides consistent experiences to on-premises line-of-business (LOB) applications and SaaS solutions.
- **Enterprise SLA of 99.9%.** You're guaranteed at least 99.9% availability of the Microsoft Entra ID P1 or P2 service. The same SLA applies to Microsoft Entra Basic.
- **Password reset with writeback.** Self-service password reset follows the Active Directory on-premises password policy.
- **Cloud App Discovery feature of Microsoft Entra ID.** This feature discovers the most frequently used cloud-based applications.
- **Conditional Access based on device, group, or location.** This lets you configure conditional access for critical resources, based on several criteria.
- **Microsoft Entra Connect Health.** You can use this tool to gain operational insight into Microsoft Entra ID. It works with alerts, performance counters, usage patterns, and configuration settings, and presents the collected information in the Microsoft Entra Connect Health portal.

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

- **Microsoft Entra ID Protection.** This feature provides enhanced functionalities for monitoring and protecting user accounts. You can define user risk policies and sign-in policies. In addition, you can review users' behavior and flag users for risk.
- **Microsoft Entra Privileged Identity Management.** This functionality lets you configure additional security levels for privileged users such as administrators. With Privileged Identity Management, you define permanent and temporary administrators. You also define a policy workflow that activates whenever someone wants to use administrative privileges to perform some task.

Note

Plans change frequently. Check Microsoft's website for the current plans and capabilities.

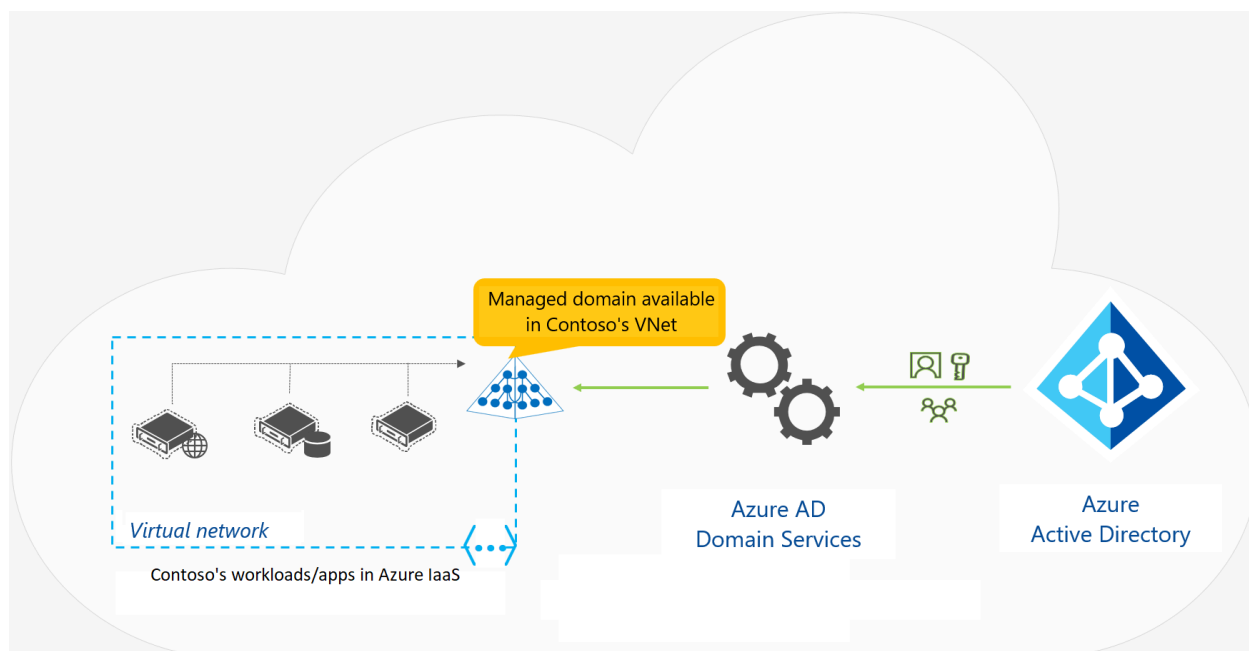
6. Examine Microsoft Entra Domain Services

Examine Microsoft Entra Domain Services

Completed

In most organizations today, line-of-business (LOB) applications are deployed on computers and devices that are domain members. These organizations use AD DS–based credentials for authentication, and Group Policy manages them. When you consider moving these apps to run in Azure, one key issue is how to provide authentication services to these apps. To satisfy this need, you can choose to implement a site-to-site virtual private network (VPN) between your local infrastructure and the Azure IaaS, or you can deploy replica domain controllers from your local AD DS as virtual machines (VMs) in Azure. These approaches can entail additional costs and administrative effort. Additionally, the difference between these two approaches is that with the first option, authentication traffic will cross the VPN, while in the second option, replication traffic will cross the VPN and authentication traffic stays in the cloud.

Microsoft provides Microsoft Entra Domain Services as an alternative to these approaches. This service, which runs as part of the Microsoft Entra ID P1 or P2 tier, provides domain services such as Group Policy management, domain joining, and Kerberos authentication to your Microsoft Entra tenant. These services are fully compatible with locally deployed AD DS, so you can use them without deploying and managing additional domain controllers in the cloud.



Because Microsoft Entra ID can integrate with your local AD DS, when you implement Microsoft Entra Connect, users can utilize organizational credentials in both on-premises AD DS and in Microsoft Entra Domain Services. Even if you don't have AD DS deployed locally, you can choose to use

Microsoft Entra Domain Services as a cloud-only service. This enables you to have similar functionality of locally deployed AD DS without having to deploy a single domain controller on-premises or in the cloud. For example, an organization can choose to create a Microsoft Entra tenant and enable Microsoft Entra Domain Services, and then deploy a virtual network between its on-premises resources and the Microsoft Entra tenant. You can enable Microsoft Entra Domain Services for this virtual network so that all on-premises users and services can use domain services from Microsoft Entra ID.

Microsoft Entra Domain Services provides several benefits for organizations, such as:

- Administrators don't need to manage, update, and monitor domain controllers.
- Administrators don't need to deploy and manage Active Directory replication.
- There's no need to have Domain Admins or Enterprise Admins groups for domains that Microsoft Entra ID manages.

If you choose to implement Microsoft Entra Domain Services, you need to be aware of the service's current limitations. These include:

- Only the base computer Active Directory object is supported.
- It's not possible to extend the schema for the Microsoft Entra Domain Services domain.
- The organizational unit (OU) structure is flat and nested OUs aren't currently supported.
- There's a built-in Group Policy Object (GPO), and it exists for computer and user accounts.
- It's not possible to target OUs with built-in GPOs. Additionally, you can't use Windows Management Instrumentation filters or security-group filtering.

By using Microsoft Entra Domain Services, you can freely migrate applications that use LDAP, NTLM, or the Kerberos protocols from your on-premises infrastructure to the cloud. You can also use applications such as Microsoft SQL Server or Microsoft SharePoint Server on VMs or deploy them in the Azure IaaS, without needing domain controllers in the cloud or a VPN to local infrastructure.

You can enable Microsoft Entra Domain Services by using the Azure portal. This service charges per hour based on the size of your directory.

7. Module assessment

<https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/7-knowledge-check>

Module assessment

Completed

8. Summary

<https://learn.microsoft.com/en-us/training/modules/understand-azure-active-directory/8-summary>

Summary

Completed

Active Directory provides the core service of identity management. AD DS is the traditional on-premises solution, whereas Microsoft Entra ID is the cloud-based solution. Microsoft Entra ID is frequently adopted at first to facilitate authentication for cloud-based apps, but is capable of providing authentication services for the entire infrastructure. While they provide similar solutions, each offer different capability and are often used together to provide a best-of-breed solution. Microsoft Entra ID is offered as a free service, with paid tiers for additional capabilities, depending on an organization's needs.

Learn more

- [What is Microsoft Entra ID?](#)
- [Compare Active Directory to Microsoft Entra ID](#)

Create, configure, and manage identities

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/1-introduction>

Introduction

Completed

Transitioning workloads to the cloud involves more than just moving servers, websites, and data. Companies need to think about how to secure those resources, by defining authorized users. Next, companies need to ensure that users only have access to data that they need, that user authorization is limited only create services available to them, and that users only perform operations authorized for them to perform. 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.

Microsoft Entra ID, the Microsoft cloud-based identity, and access management service, helps make these challenges easier to solve. Microsoft Entra ID provides end-to-end identity management; including single sign-on and multifactor authentication to help protect your users and your data. In this module, you learn the basics of creating, configuring, and managing users and groups of users. You also learn to manage licenses and device registration.

Learning objectives

In this module, you'll:

- Create, configure, and manage users
- Create, configure, and manage groups
- Manage licenses
- Configure and manage device registration
- Explore custom security attributes and automatic provisioning

Prerequisites

- Basic understanding of identity management

- Some experience with Active Directory a plus
- Experience with Zero Trust helpful

2. Create, configure, and manage users

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/2-users>

Create, configure, and manage users

Completed

Every user who needs access to resources needs a user account in Microsoft Entra ID. 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 admin center** 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 admin center 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. Take a minute to access the admin center and view your users. Notice the **User Type** column to see members and guests, as the following figure depicts.

Display name	User principal name	User type	Country or region	Job title
RN Robina Nauman	RobiNa	Member	United States	Retail Manager
MA Marcos Amboade	MarcAm	Member	United States	Marketing Assistant
CG Chris Green	ChrisG	Member		
CB Carmelo Barese	CarmeB	Member	United States	HR Manager
UJ Umar Javaid	UmaJav	Member	United States	Designer
UF Uxía Feal	UxiaFe	Member	United States	Developer
VS Vishnu Surendran	VishnS	Member	United States	Sales Rep

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 brings these users into Microsoft Entra ID. **Microsoft Entra Cloud Sync** is the recommended synchronization tool for most organizations—it uses a lightweight cloud-managed agent and supports multiple disconnected forests. **Microsoft Entra Connect Sync** remains available for complex scenarios such as device synchronization or groups with more than 50,000 members. Their source is **Windows Server AD**.
- **Guest users** - These users exist outside your organization. Examples are accounts from other cloud providers and Microsoft accounts. Their source is **Invited user**. This type of account is useful when external vendors or contractors need access to your organization's resources. Once their help is no longer necessary, you can remove the account and all of their access.

3. Exercise - assign licenses to users

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/3-exercise-assign-licenses-users>

Exercise - assign licenses to users

Completed

Exercise environment needs - this lab assumes you have a basic Microsoft Entra tenant with at least User Administrator rights to complete it. You can get a free trial subscription for at [Try Azure for Free](#).

Create a new user in Microsoft Entra ID

You can skip creating this user if you created the same user in the earlier module.

1. Browse to the Identity menu in the [Microsoft Entra admin center](#).
2. In the left navigation, under select **Users**, then **All Users**.
3. Within the Users page, on the menu, select + **New user** and **Create new user**.
4. Create a user using the following information:

Setting	Value
User principal name	ChrisG
Name	Chris Green
First name	Chris
Last name	Green
Password	make up a unique password

5. When complete, verify the account for Chris Green is shown in the **All users** list.

Create a security group in Microsoft Entra ID

1. Browse to the Microsoft Entra admin center screen.
2. In the left navigation, under **Identity**, select **Groups** and then **All groups**.
3. In the Groups screen, on the menu, select **New group**.
4. Create a group using the following information:

Setting	Value
Group type	Security
Group name	Marketing
Membership type	Assigned
Owners	Assign your own administrator account as the group owner
Members	Chris Green

Home > Licenses | Overview > Groups | All groups >

New Group

Got feedback?

Group type * ⓘ
Security

Group name * ⓘ
Marketing

Group description ⓘ
This is a new group created for the SC-300 training content.

Microsoft Entra roles can be assigned to the group ⓘ
Yes No

Membership type ⓘ
Assigned

Owners
No owners selected

Members
No members selected

Create

Add members

Try changing or adding filters if you don't see what you're looking for.

Search ⓘ
mar
2 results found

All Users Groups Devices Enterprise applications

	Name	Type	User principal name
☐	Marcos Amboade	User	MarcAm
☐	Marykutty Robin	User	MarykR

Selected (0)
Reset
No items selected

Select

5. When complete, verify the group named **Marketing** is shown in the **All groups** list.

Assign a license to a group

License assignment to groups is managed through the Microsoft 365 admin center.

1. Go to the Microsoft 365 admin center at <https://admin.microsoft.com>.
2. Select **Billing** from the menu on the left.
3. Select **Licenses**.
4. From the list of licenses you have available, select one.
5. Select **Groups** from the list near the top of the screen.
6. On the Groups page, select **+ Assign license**.
7. Search for and select the **Marketing** group you created earlier.
8. Select the **Assign** button at the bottom of the dialog.
9. You should get a message that licenses were successfully assigned.

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.

Important

You can't restore a permanently deleted user.

Required permissions

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

4. Exercise - restore or remove deleted users

Exercise - restore or remove deleted users

Completed

Exercise environment needs - this lab assumes you have a basic Microsoft Entra tenant with at least User Administrator rights to complete it. You can get a free trial subscription at [Try Microsoft Azure for free](#).

Remove a user from Microsoft Entra ID

1. Browse to the [Microsoft Entra admin center](#).
2. In the left navigation, under **Identity**, select **Users**.
3. In the **Users** list, select the check box for a user to delete. For example, select **Chris Green**.

Tip

Selecting users from the list allows you to manage multiple users at the same time. If you select the user, to open that user's page, you'll only be managing that individual user.

The screenshot shows the Microsoft Entra admin center interface. On the left is a navigation pane with 'Users' selected. The main area displays a list of 20 users, with 'Chris Green' selected. The 'Delete' button in the top right corner is highlighted with a red box.

	Display name ↑	User type	On-premises sy...
☐	Adele Vance	Member	No
☐	Alex Wilber	Member	No
☒	CG Chris Green	Member	No
☐	Diego Siciliani	Member	No
☐	Grady Archie	Member	No
☐	Henrietta Mueller	Member	No
☐	Isaiah Langer	Member	No
☐	Johanna Lorenz	Member	No

4. With the user account selected, on the menu, select **Delete user**.

5. Review the dialog box and then select **OK**.

Restore a deleted user

You can see all the users that were deleted less than 30 days ago. These users can be restored.

1. In the Users page, in the left navigation, select **Deleted users**.
2. Review the list of deleted users and select the user you deleted.

Important

By default, deleted user accounts are permanently removed from Microsoft Entra ID automatically after 30 days.

3. On the menu, select **Restore user**.
4. Review the dialog box and then select **OK**.
5. In the left navigation, select **All users**.
6. Verify the user was restored.

5. Create, configure, and manage groups

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/5-groups>

Create, configure, and manage groups

Completed

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 let you 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 access to shared resources. Members of a security group can include users, devices, and service principals. 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 and 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 has no groups defined.

Groups | All groups ...
Microsoft Entra ID for workforce

« New group Download groups Refresh Manage view ▾

Search

14 groups found

☐	Name ↑	Group type
☐	AD AAD DC Administrators	Security
☐	AC All Company	Microsoft 365

Settings

- General
- Expiration
- Naming policy

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

- **Assigned** - members are added and maintained manually.
- **Dynamic User** - users are added and removed automatically based on rules that evaluate user attributes such as department, job title, or location.
- **Dynamic Device** - devices are added and removed automatically based on rules that evaluate device attributes. Applies to security groups only; Microsoft 365 groups support dynamic users but not dynamic devices.

Dynamic groups

With dynamic membership, Microsoft Entra ID automatically adds or removes users or devices from a group based on rules you define. When a member's attributes change—for example, a user moves to a different department—all dynamic membership rules in the tenant are reevaluated, and the user is added to or removed from groups accordingly.

Dynamic membership requires a **Microsoft Entra ID P1** license (or Intune for Education for device-based rules).

Dynamic membership rules

Save

Discard

Got feedback?

Configure Rules

You can use the rule builder or rule syntax text box to create or edit a dynamic membership rule. [Learn more](#)

And/Or	Property	Operator	Value	
And	<Choose a Property>	<Choose an Operat...	Add a value	

+ Add expression

+ [Get custom extension properties](#)

Some items could not be displayed in the rule builder. [Learn more](#)

Rule syntax

user.objectId -ne null

Edit

For example, you can create a rule that automatically adds all users whose **Department** attribute equals "Marketing" to a Marketing security group, keeping membership current without manual updates.

6. Exercise - add groups in Microsoft Entra ID

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/6-exercise-add-groups-azure-active-directory>

Exercise - add groups in Microsoft Entra ID

Completed

Exercise environment needs - this lab assumes you have a basic Microsoft Entra tenant with at least User Administrator rights to complete it. You can get a free trial subscription at [Try Microsoft Azure for free](#).

Create a Microsoft 365 group in Microsoft Entra ID

1. Browse to the [Microsoft Entra admin center](#).

2. In the left navigation, under **Identity**, select **Groups**.
3. In the Groups page, on the menu, select **New group**.
4. Create a group using the following information:

Setting	Value
Group type	Microsoft 365
Group name	Northwest Sales
Membership type	Assigned
Owners	Assign your own administrator account as the group owner
Members	Assign a member of this group

New Group ...

 Got feedback?

Group type * ⓘ

Microsoft 365

Group name * ⓘ

Northwese Sales

Group email address * ⓘ

NorthweseSales @<<domain/tenant>>.onmicrosoft.com

Group description ⓘ

Group containing members of the Sales team in Northwest region

Microsoft Entra roles can be assigned to the group ⓘ

Yes

No

Membership type * ⓘ

Assigned

Owners

1 owner selected

Members

1 member selected

5. When complete, verify the group named **Northwest sales** is shown in the **All groups** list.
6. You have to refresh the **All groups** a couple of times for the new group to show up.

7. Configure and manage device registration

Configure and manage device registration

Completed

With the proliferation of devices of all shapes and sizes and the proliferation of bring-your-own-device (BYOD), IT professionals are faced with two somewhat opposing goals:

- Allow end users to be productive wherever and whenever and on any device
- Protect the organization's assets

To protect these assets, IT-staff needs to first manage the device identities. IT-staff can build on the device identity with tools like Microsoft Intune to ensure standards for security and compliance are met. Microsoft Entra ID enables single sign-on to devices, apps, and services from anywhere through these devices.

- Your users get access to your organization's assets they need.
- Your IT-staff gets the controls they need to secure your organization.

Microsoft Entra registered devices

The goal of Microsoft Entra registered devices is to provide your users with support for the BYOD or mobile device scenarios. In these scenarios, a user can access your organization's Microsoft Entra ID controlled resources using a personal device.

Microsoft Entra registered	Description
Definition	Registered to Microsoft Entra ID without requiring organizational account to sign in to the device
Primary audience	Applicable to Bring your own device (BYOD), and Mobile devices
Device ownership	User or Organization
Operating systems	Windows 10 or newer, macOS 10.15 or newer, iOS 15 or newer, Android, Linux (Ubuntu 20.04/22.04/24.04 LTS, Red Hat Enterprise Linux 8/9 LTS)
Device sign in options	End-user local credentials, Password, Windows Hello, PIN, Biometrics
Device management	Mobile Device Management (example: Microsoft Intune), Mobile Application Management
Key capabilities	SSO to cloud resources, Conditional Access when enrolled in Intune, Conditional Access via App protection policy



Microsoft Entra registered devices are signed in to using a local account like a Microsoft account on a Windows 10 or newer device, but additionally have a Microsoft Entra account attached for access to organizational resources. Access to resources in the organization can be further limited based on that Microsoft Entra account and Conditional Access policies applied to the device identity.

Administrators can secure and further control these Microsoft Entra registered devices using Mobile Device Management (MDM) tools like Microsoft Intune. MDM provides a means to enforce organization-required configurations like requiring storage to be encrypted, password complexity, and security software kept updated.

Microsoft Entra ID registration can be accomplished when accessing a work application for the first time or manually using the Windows 10 or Windows 11 Settings menu.

Scenarios for registered devices

A user in your organization wants to access tools for email, reporting time-off, and benefits enrollment from their home PC. Your organization has these tools behind a Conditional Access policy that requires access from an Intune compliant device. The user adds their organization account and registers their home PC with Microsoft Entra ID and the required Intune policies are enforced giving the user access to their resources.

Another user wants to access their organizational email on their personal Android phone infected by a root-kit. Your company requires a compliant device and created an Intune compliance policy to block any rooted devices. The employee is stopped from accessing organizational resources on this device.

Microsoft Entra joined devices

Microsoft Entra joined is intended for organizations that want to be cloud-first or cloud-only. Any organization can deploy Microsoft Entra joined devices no matter the size or industry. Microsoft Entra joined enables access to both cloud and on-premises apps and resources.

Microsoft Entra joined	Description
Definition	Joined only to Microsoft Entra ID requiring organizational account to sign in to the device
Primary audience	Suitable for both cloud-only and hybrid organizations
Device ownership	Organization
Operating systems	All Windows 10 and Windows 11 devices (except Home editions); Windows Server 2019 and newer VMs in Azure (Server Core not supported); macOS 13 or newer (preview)
Device management	Mobile Device Management (example: Microsoft Intune)
Key capabilities	SSO to both cloud and on-premises resources, Conditional Access, Self-service Password Reset and Windows Hello PIN reset

Microsoft Entra joined devices are signed in to using an organizational Microsoft Entra account. Access to resources in the organization can be further limited based on that Microsoft Entra account

and Conditional Access policies applied to the device identity.

Administrators can secure and further control Microsoft Entra joined devices using Mobile Device Management (MDM) tools like Microsoft Intune or in co-management scenarios using Microsoft Endpoint Configuration Manager. These tools provide a means to enforce organization-required configurations like requiring storage to be encrypted, password complexity, software installations, and software updates. Administrators can make organization applications available to Microsoft Entra joined devices using Configuration Manager.

Microsoft Entra joined can be accomplished using self-service options like the Out of Box Experience (OOBE), bulk enrollment, or Windows Autopilot.

Microsoft Entra joined devices can still maintain single sign-on access to on-premises resources when they are on the organization's network. Microsoft Entra joined devices authenticate to on-premises servers like for file, print, and other applications.

Scenarios for joined devices

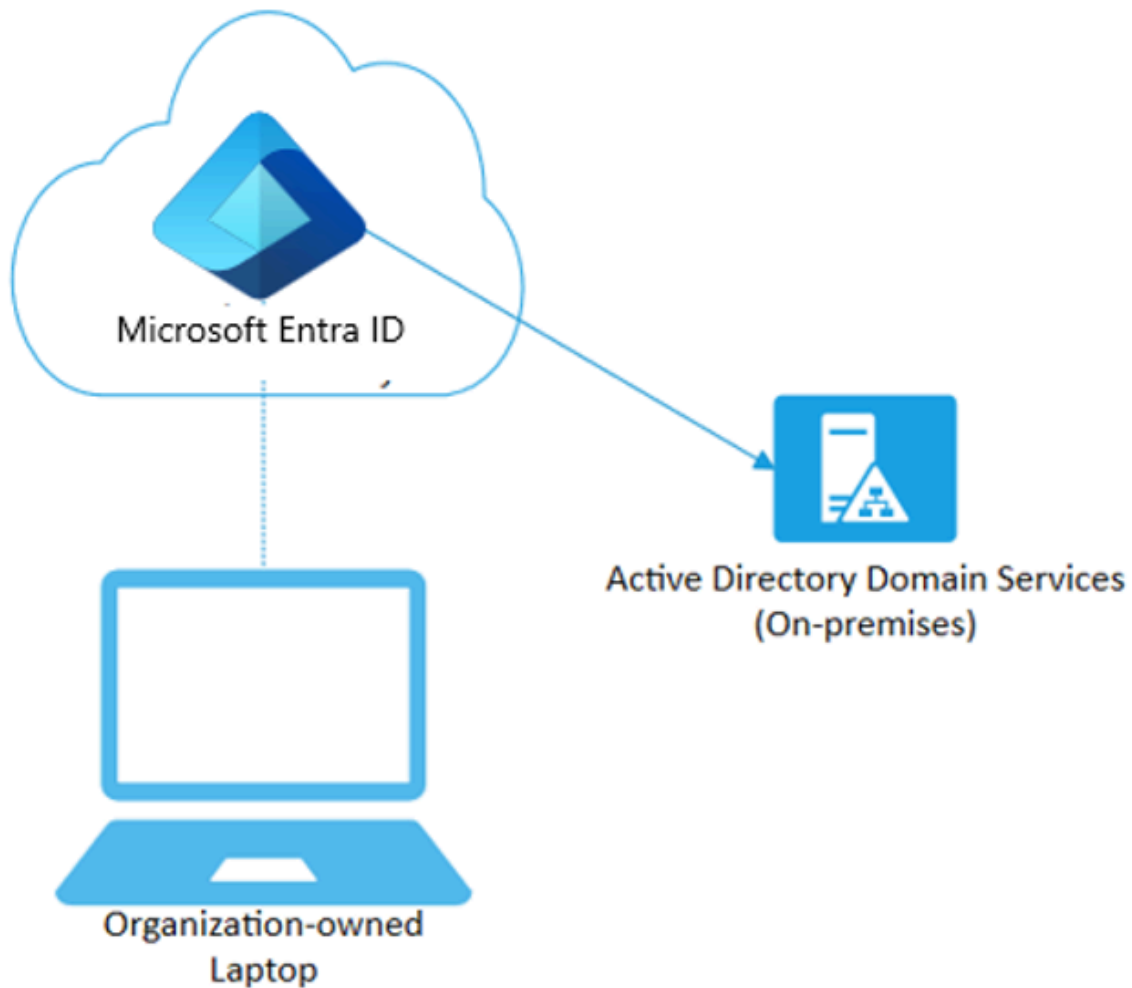
Although Microsoft Entra joined is primarily intended for organizations that don't have an on-premises Windows Server Active Directory infrastructure, you can certainly use it in scenarios where:

- You want to transition to cloud-based infrastructure using Microsoft Entra ID and MDM like Intune.
- You can't use an on-premises domain join, for example, if you need to get mobile devices such as tablets and phones under control.
- Your users primarily need to access Microsoft 365 or other SaaS apps integrated with Microsoft Entra ID.
- You want to manage a group of users in Microsoft Entra ID instead of in Active Directory. This scenario can apply, for example, to seasonal workers, contractors, or students.
- You want to provide joining capabilities to workers in remote branch offices with limited on-premises infrastructure.

You can configure Microsoft Entra joined devices for all Windows 10 and Windows 11 devices except for the Home editions.

The goal of Microsoft Entra joined devices is to simplify:

- Windows deployments of work-owned devices
- Access to organizational apps and resources from any Windows device
- Cloud-based management of work-owned devices
- Users to sign in to their devices with their Microsoft Entra ID or synced Active Directory work or school accounts.



Microsoft Entra Joined can be deployed by using many different methods.

Hybrid Microsoft Entra joined devices

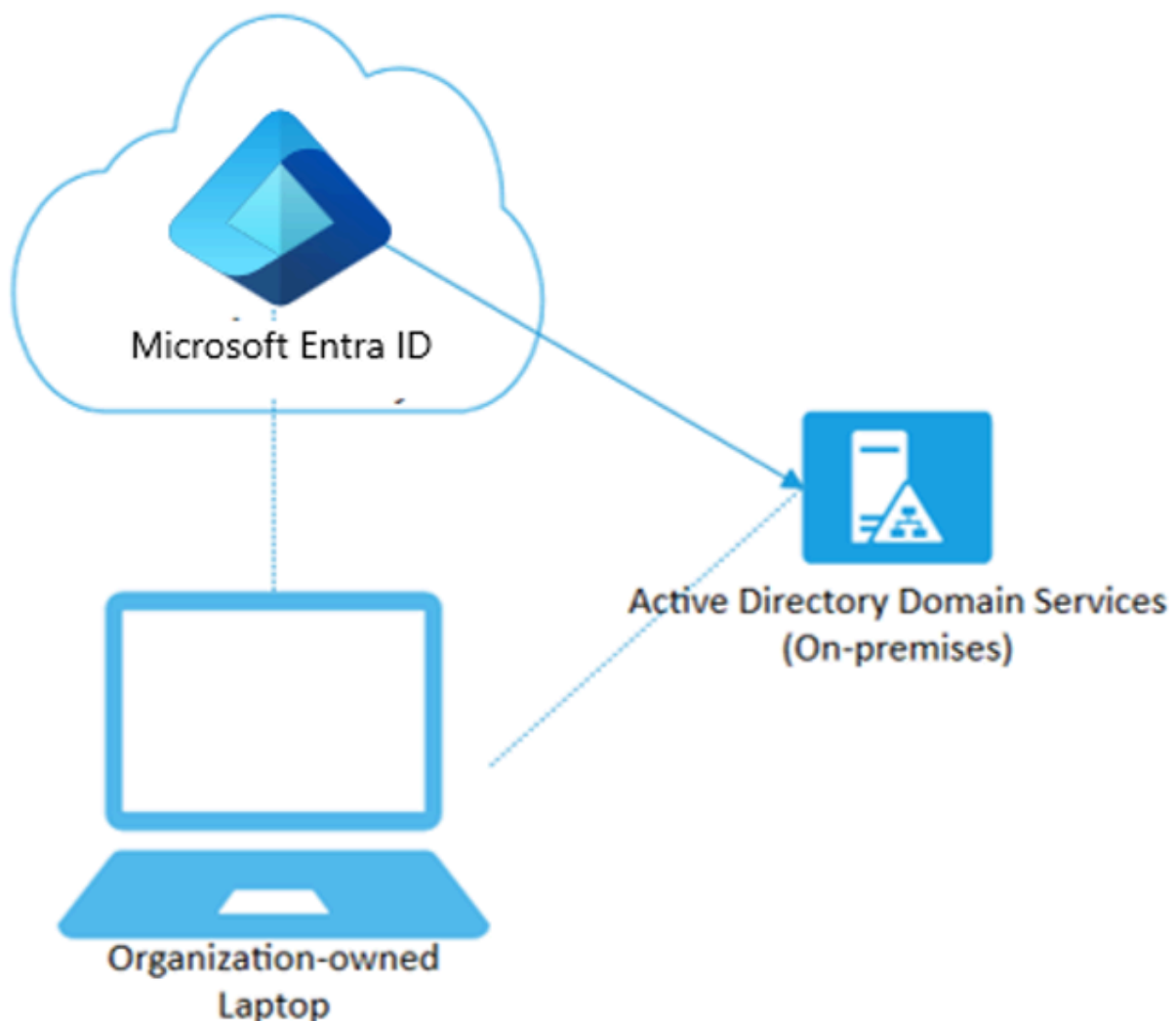
For more than a decade, many organizations have used the domain join to their on-premises Active Directory to enable:

- IT departments to manage work-owned devices from a central location.
- Users to sign in to their devices with their Active Directory work or school accounts.

Typically, organizations with an on-premises footprint rely on imaging methods to configure devices, and they often use **Configuration Manager** or **group policy (GP)** to manage them.

If your environment has an on-premises AD footprint and you also want benefit from the capabilities provided by Microsoft Entra ID, you can implement hybrid Microsoft Entra joined devices. These devices are devices that are joined to your on-premises Active Directory and registered with your Microsoft Entra directory.

Hybrid Microsoft Entra joined	Description
Definition	Joined to on-premises AD and Microsoft Entra ID requiring organizational account to sign in to the device
Primary audience	Suitable for hybrid organizations with existing on-premises AD infrastructure
Device ownership	Organization
Operating systems	Windows 10, Windows 11 (except Home editions), Windows Server 2016, 2019, and 2022
Device sign in options	Password or Windows Hello for Business
Device management	Group Policy, Configuration Manager standalone, or co-management with Microsoft Intune
Key capabilities	SSO to both cloud and on-premises resources, Conditional Access, Self-service Password Reset and Windows Hello PIN reset



Scenarios for hybrid joined

Use Microsoft Entra hybrid joined devices if:

- You have Win32 apps deployed to these devices that rely on Active Directory machine authentication.
- You want to continue to use Group Policy to manage device configuration.
- You want to continue to use existing imaging solutions to deploy and configure devices.

Device writeback (no longer supported)

Device writeback is no longer supported and is no longer a recommended approach for hybrid identity scenarios. It is replaced by **Cloud Kerberos Trust**, which allows Microsoft Entra joined and hybrid joined devices to authenticate to on-premises resources without requiring device objects to be written back to on-premises Active Directory.

For organizations planning new hybrid deployments, use Cloud Kerberos Trust to enable on-premises SSO and Windows Hello for Business in hybrid environments. See [Configure Microsoft Entra Kerberos for on-premises single sign-on](#) for guidance.

8. Manage licenses

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/8-manage-licenses>

Manage licenses

Completed

Microsoft paid cloud services, such as Microsoft 365, Enterprise Mobility + Security, Dynamics 365, and other similar products, require licenses. These licenses are assigned to each user who needs access to these services. To manage licenses, administrators use the [Microsoft 365 admin center](#) or PowerShell and Microsoft Graph API. Microsoft Entra ID is the underlying infrastructure that supports identity management for all Microsoft cloud services. Microsoft Entra ID stores information about license assignment states for users.

Without group-based licensing, assigning licenses at the individual user level makes large-scale management difficult. For example, to add or remove user licenses based on organizational changes, such as users joining or leaving the organization or a department, an administrator often must write a complex PowerShell script. This script makes individual calls to the cloud service.

To address those challenges, Microsoft Entra ID now includes group-based licensing. You can assign one or more product licenses to a group. Microsoft Entra ID ensures that the licenses are assigned to all members of the group. Any new members who join the group are assigned the appropriate licenses. When they leave the group, those licenses are removed. This licensing management

eliminates the need for automating license management via PowerShell to reflect changes in the organization and departmental structure on a per-user basis.

License requirements

You must have one of the following licenses to use group-based licensing:

- Paid or trial subscription for Microsoft Entra ID Premium P1 and greater
- Paid or trial edition Office 365 Enterprise E3 or greater

Required number of licenses

For any groups assigned a license, you must also have a license for each unique member. While you don't have to assign each member of the group a license, you must have at least enough licenses to include all of the members. For example, if you have 1,000 unique members who are part of licensed groups in your tenant, you must have at least 1,000 licenses to meet the licensing agreement.

Features

Here are the main features of group-based licensing:

- Licenses can be assigned to any security group in Microsoft Entra ID. Security groups can be synced from on-premises by using Microsoft Entra Cloud Sync (recommended) or Microsoft Entra Connect Sync. You can also create security groups directly in Microsoft Entra ID (also called cloud-only groups), or automatically via the Microsoft Entra dynamic group feature.
- When a product license is assigned to a group, the administrator can disable one or more service plans in the product. Typically, this assignment is done when the organization isn't yet ready to start using a service included in a product. For example, the administrator might assign Microsoft 365 to a department, but temporarily disable the Viva Engage service.
- All Microsoft cloud services that require user-level licensing are supported. This support includes all Microsoft 365 products, Enterprise Mobility + Security, and Dynamics 365.
- Group-based licensing is currently available only through the [Microsoft 365 admin center](#).
- Microsoft Entra ID automatically manages license modifications that result from group membership changes. Typically, license modifications are effective within minutes of a membership change.
- A user can be a member of multiple groups with license policies specified. A user can also have some licenses that were directly assigned, outside of any groups. The resulting user state is a combination of all assigned product and service licenses. If a user is assigned same license from multiple sources, the license is consumed only once.
- In some cases, licenses can't be assigned to a user. For example, there might not be enough available licenses in the tenant, or conflicting services are assigned at the same time. Administrators have access to information about users for whom Microsoft Entra ID couldn't fully process group licenses. They can then take corrective action based on that information.

Some Microsoft services aren't available in all locations. The administrator, before assigning a license to a user, should specify usage location in the User Profile.

For group license assignment, any users without a usage location specified inherit the location of the directory. If you have users in multiple locations, we recommend that you always set usage location as part of your user creation. Usage location helps ensure the result of license assignment is always correct and users don't receive services in locations that aren't allowed.

9. Exercise - change group license assignments

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/9-exercise-change-group-license-assignments>

Exercise - change group license assignments

Completed

Exercise environment needs - this lab assumes you have a basic Microsoft Entra tenant with at least User Administrator rights to complete it. You can get a free trial subscription at [Try Microsoft Azure for free](#).

Change group license assignment

1. Open <https://entra.microsoft.com> to get to the Microsoft Entra admin center.
2. In the left navigation, open **Groups**.
3. Select **All groups**, then select one of the available groups.
4. In the left navigation, under **Manage**, select **Licenses**.

You see a list of any license assignments that are currently made. And you find that you have to use the Microsoft 365 Admin Center to make any updates.

5. Review the current assignments and then, on the menu, select **+ Assignments**.
6. Open <https://admin.microsoft.com> to open the Microsoft 365 admin center.
7. Select **Billing**. Then select **Licenses**.
8. Select an available license from the list.
9. Select **Groups** from the menu near the top of the page.
10. Select the **+ Assign licenses** option.
11. Pick the group you were looking at earlier in Microsoft Entra. Then select the **Assign** button at the bottom of the page.
12. On the group's Licenses page, review the change. You should be able to see the change in both the Microsoft Entra admin center and the Microsoft 365 admin center.

Identify and resolve license assignment problems for a group in Microsoft Entra ID

Group-based licensing in Microsoft Entra ID introduces the concept of users in a licensing error state. In this section, we explain the reasons why users might end up in this state.

When you assign licenses directly to individual users, without using group-based licensing, the assignment operation might fail. For example, when you execute the PowerShell cmdlet `Set-MgUserLicense` on a user object, the cmdlet can fail for many reasons that are related to business logic. For example, there might be an insufficient number of licenses or a conflict between two service plans that can't be assigned at the same time. The problem is immediately reported back to you.

When you're using group-based licensing the same errors can occur, but they happen in the background while the Microsoft Entra service is assigning licenses. For this reason, the errors can't be communicated to you immediately. Instead, they're recorded on the user object and then reported via the administrative portal. The original intent to license the user is never lost, but it's recorded in an error state for future investigation and resolution.

Not enough licenses

Problem: There aren't enough available licenses for one of the products specified in the group. You need to either purchase more licenses for the product or free up unused licenses from other users or groups.

To see how many licenses are available, go to **Microsoft Entra - Identity - Billing**, then **Licenses**, then **All products**.

To see which users and groups are consuming licenses, select a product. Under **Licensed users**, you see a list of all users who have licenses assigned directly or via one or more groups. Under **Licensed groups**, you see all groups with product licenses assigned.

PowerShell: PowerShell cmdlets report this error as *CountViolation*.

Service plans that conflict

Problem: One of the products specified in the group contains a service plan that conflicts with another service plan that's already assigned to the user via a different product. Some service plans are configured in a way that they can't be assigned to the same user as another, related service plan.

Consider the following example. A user has a license for Office 365 Enterprise *E1* assigned directly, with all the plans enabled. The user is added to a group that has the Office 365 Enterprise *E3* product assigned to it. The E3 product contains service plans that can't overlap with the plans that are included in E1, so the group license assignment fails with the **Conflicting service plans** error. In this example, the conflicting service plans are:

- SharePoint Online (Plan 2) conflicts with SharePoint Online (Plan 1).
- Exchange Online (Plan 2) conflicts with Exchange Online (Plan 1).

To solve this conflict, you need to disable two of the plans. You can disable the E1 license directly assigned to the user. Or, you need to modify the entire group license assignment and disable the plans in the E3 license. Alternatively, you might decide to remove the E1 license from the user if it's redundant in the context of the E3 license.

The decision about how to resolve conflicting product licenses always belongs to the administrator. Microsoft Entra ID doesn't automatically resolve license conflicts.

PowerShell: PowerShell cmdlets report this error as *MutuallyExclusiveViolation*.

Other products depend on this license

Problem: One of the products specified in the group contains a service plan that must be enabled for another service plan, in another product, to function. This error occurs when Microsoft Entra ID attempts to remove the underlying service plan. For example, this can happen when you remove the user from the group.

To solve this problem, you need to make sure that the required plan is still assigned to users through some other method or that the dependent services are disabled for those users. After doing that, you can properly remove the group license from those users.

PowerShell: PowerShell cmdlets report this error as *DependencyViolation*.

Usage location isn't allowed

Problem: Some Microsoft services aren't available in all locations because of local laws and regulations. Before you can assign a license to a user, you must specify the **Usage location** property for the user. You can specify the location under the **User**, then **Profile**, then **Edit** the section in the Azure portal.

When Microsoft Entra ID attempts to assign a group license to a user whose usage location isn't supported, it fails and records an error on the user.

To solve this problem, remove users from unsupported locations from the licensed group. Alternatively, if the current usage location values don't represent the actual user location, you can modify them so that the licenses are correctly assigned next time (if the new location is supported). You can specify the usage location under the user's **Properties** tab in the [Microsoft Entra admin center](#).

PowerShell: PowerShell cmdlets report this error as *ProhibitedInUsageLocationViolation*.

Note

When Microsoft Entra ID assigns group licenses, any users without a specified usage location inherit the location of the directory. We recommend that administrators set the correct usage location values on users before using group-based licensing to comply with local laws and regulations.

Duplicate proxy addresses

If you use Exchange Online, some users in your organization might be incorrectly configured with the same proxy address value. When group-based licensing tries to assign a license to such a user, it fails and shows "Proxy address is already being used."

After you resolve any proxy address problems for the affected users, make sure to force license processing on the group to ensure that the licenses can now be applied.

Microsoft Entra Mail and ProxyAddresses attribute change

Problem: While updating license assignment on a user or a group, you might see that the Microsoft Entra Mail and ProxyAddresses attribute of some users are changed.

Updating license assignment on a user causes the proxy address calculation to be triggered, which can change user attributes.

LicenseAssignmentAttributeConcurrencyException in audit logs

Problem: User has LicenseAssignmentAttributeConcurrencyException for license assignment in audit logs. When group-based licensing tries to process concurrent license assignment of the same license to a user, this exception is recorded on the user. This typically happens when a user is a member of more than one group with same assigned license. Microsoft Entra ID retries processing the user license and will resolve the issue. There's no action required from the customer to fix this issue.

More than one product license assigned to a group

You can assign more than one product license to a group. For example, you can assign Office 365 Enterprise E3 and Enterprise Mobility + Security to a group to easily enable all included services for users.

Microsoft Entra ID attempts to assign all licenses that are specified in the group to each user. If Microsoft Entra ID can't assign one of the products because of business logic problems, it won't assign the other licenses in the group either. An example is if there aren't enough licenses for all, or if there are conflicts with other services that are enabled on the user.

You can see the users who failed to get assigned and check which products are affected by this problem.

When a licensed group is deleted

You must remove all licenses assigned to a group before you can delete the group. However, removing licenses from all the users in the group can take time. There can be failures if user has a dependent license assigned. If a user has a license that's dependent on a license, which is being removed due to group deletion, the license assignment to the user is converted from inherited to direct.

For example, consider a group that has Office 365 E3/E5 assigned with a Skype for Business service plan enabled. Also imagine that a few members of the group have Audio Conferencing licenses assigned directly. When the group is deleted, group-based licensing tries to remove Office 365 E3/E5 from all users. Because Audio Conferencing is dependent on Skype for Business, for any users with Audio Conferencing assigned, group-based licensing converts the Office 365 E3/E5 licenses to direct license assignment.

Manage licenses for products with prerequisites

Some Microsoft Online products you might own are *add-ons*. Add-ons require a prerequisite service plan to be enabled for a user or a group before they can be assigned a license. With group-based licensing, the system requires that both the prerequisite and add-on service plans be present in the same group to ensure that any users who are added to the group can receive the fully working product. Let's consider the following example:

Microsoft Workplace Analytics is an add-on product. It contains a single service plan with the same name. We can only assign this service plan to a user, or group, when one of the following prerequisites is also assigned:

- Exchange Online (Plan 1)
- Exchange Online (Plan 2)

If we try to assign this product on its own to a group, the portal returns a notification message. If we select the item details, it shows the following error message:

License operation failed. Make sure that the group has necessary services before adding or removing a dependent service. **The service Microsoft Workplace Analytics requires Exchange Online (Plan 2) to be enabled as well.**

To assign this add-on license to a group, we must ensure that the group also contains the prerequisite service plan. For example, we might update an existing group that already contains the full Office 365 E3 product, and then add the add-on product to it.

It's also possible to create a standalone group that contains only the minimum required products to make the add-on work. It can then be used to license only selected users for the add-on product. Based on the previous example, you would assign the following products to the same group:

- Office 365 Enterprise E3 with only the Exchange Online (Plan 2) service plan enabled
- Microsoft Workplace Analytics

From now on, any users added to this group consume one license of the E3 product and one license of the Workplace Analytics product. At the same time, those users can be members of another group that gives them the full E3 product, and they still consume only one license for that product.

Tip

You can create multiple groups for each prerequisite service plan. For example, if you use both Office 365 Enterprise E1 and Office 365 Enterprise E3 for your users, you can create two groups to license Microsoft Workplace Analytics: one that uses E1 as a prerequisite and the other that uses E3. This lets you distribute the add-on to E1 and E3 users without consuming more licenses.

Force the group license process to resolve errors

Depending on what steps taken to resolve the errors, it might be necessary to manually trigger the processing of a group to update the user state.

For example, if you free up some licenses by removing direct license assignments from users, you need to trigger the processing of groups that previously failed to fully license all user members. To reprocess a group, go to the group pane, open **Licenses**, and then select the **Reprocess** button on the toolbar.

Force the user license process to resolve errors

Depending on what steps taken to resolve the errors, it might be necessary to manually trigger the processing of a user to update the user's state.

For example, after you resolve duplicate proxy address problem for an affected user, you need to trigger the processing of the user. To reprocess a user, go to the user pane, open **Licenses**, and then select the **Reprocess** button on the toolbar.

How to migrate users with individual licenses to group licenses

You can have existing licenses deployed to users in the organizations via direct assignment; that is, using PowerShell scripts or other tools to assign individual user licenses. Before you begin using group-based licensing to manage licenses in your organization, you can use this migration plan to seamlessly replace existing solutions with group-based licensing.

Keep in mind that you should avoid a situation in which migrating to group-based licensing results in users temporarily losing their currently assigned licenses. Any process that result in removal of licenses should be avoided to remove the risk of users losing access to services and their data.

Recommended migration process

1. You have existing automation (for example, PowerShell) managing license assignment and removal for users. Leave it running as is.
2. Create a new licensing group (or decide which existing groups to use) and make sure that all required users are added as members.
3. Assign the required licenses to those groups; your goal should be to reflect the same licensing state your existing automation (for example, PowerShell) is applying to those users.
4. Verify that licenses are applied to all users in those groups. This application can be done by checking the processing state on each group and by checking Audit Logs.
 - You can perform a random check of a few individual users by looking at their license details. You see that they have the same licenses assigned "directly" and "inherited" from groups.
 - You can run a PowerShell script to [verify how licenses are assigned to users](#).
 - When the same product license is assigned to the user both directly and through a group, only one license is consumed by the user. Hence no more licenses are required to perform migration.
5. Verify that no license assignments failed by checking each group for users in error state.

Consider removing the original direct assignments. We recommend that you do it gradually, and monitor the outcome on a subset of users first. If you leave the original direct assignments on users, when the users leave their licensed groups they retain the directly assigned licenses, which might not be what you want.

An example

An organization has 1,000 users. All users require Office 365 Enterprise E3 licenses. Currently the organization has a PowerShell script running on premises, adding and removing licenses from users as they come and go. However, the organization wants to replace the script with group-based licensing so licenses can be managed automatically by Microsoft Entra ID.

Here is what the migration process could look like:

1. Using the Azure portal, assign the Office 365 E3 license to the **All users** group in Microsoft Entra ID.
2. Confirm that license assignment has completed for all users. Go to the overview page for the group, select **Licenses**, and check the processing status at the top of the **Licenses** page.
 - Look for "Latest license changes have been applied to all users" to confirm processing has completed.
 - Look for a notification on top about any users for whom licenses were not successfully assigned. Did we run out of licenses for some users? Do some users have conflicting license plans that prevent them from inheriting group licenses?

3. You need to check a few users to verify that they have both the direct and group licenses applied. Go to the profile page for a user, select Licenses, and examine the state of licenses.
- This is the expected user state during migration:

PRODUCTS	STATE	ENABLED SERVICES	ASSIGNMENT PATHS
Azure Active Directory Premium P1	Active with errors	3/3	Inherited (asd,All Users)
Enterprise Mobility + Security E3	Active	6/6	Direct, Inherited (AniGroup,All Users)
Microsoft Azure Active Directory Standa...	Active	2/2	Direct, Inherited (All Users)
Microsoft Azure Multi-Factor Authentic...	Active	1/1	Inherited (All Users)
Microsoft Dynamics CRM Online	Active	5/5	Direct
Office 365 E3	Active	17/17	Direct, Inherited (AniGroup)
Power BI (free)	Active	1/1	Direct

4. After confirming that both direct and group licenses are equivalent, you can start removing direct licenses from users. You can test this by removing them for individual users in the portal and then run automation scripts to have them removed in bulk. Here's an example of the same user with the direct licenses removed through the portal. Notice that the license state remains unchanged, but we no longer see direct assignments.

PRODUCTS	STATE	ENABLED SERVICES	ASSIGNMENT PATHS
Azure Active Directory Premium P1	Active with errors	3/3	Inherited (asd,All Users)
Enterprise Mobility + Security E3	Active	6/6	Direct, Inherited (AniGroup,All Users)
Microsoft Azure Active Directory Standa...	Active	2/2	Direct, Inherited (All Users)
Microsoft Azure Multi-Factor Authentic...	Active	1/1	Inherited (All Users)
Microsoft Dynamics CRM Online	Active	5/5	Direct
Office 365 E3	Active	17/17	Inherited (AniGroup)
Power BI (free)	Active	1/1	Direct

Change license assignments for a user or group in Microsoft Entra ID

This section describes how to move users and groups between service license plans in Microsoft Entra ID. The goal is to ensure that there's no loss of service or data during the license change. Users should switch between services seamlessly. The license plan assignment steps in this section describe changing a user or group on Office 365 E1 to Office 365 E3, but the steps apply to all license plans. When you update license assignments for a user or group, the license assignment removals and new

assignments are made simultaneously so that users don't lose access to their services during license changes or see license conflicts between plans.

Before you update the license assignments, verify certain assumptions are true for all of the users or groups to be updated. If the assumptions aren't true for all of the users in a group, the migration might fail for some. As a result, some of the users might lose access to services or data. Ensure that:

- Users have the current license plan that's assigned to a group and inherited by the user and not assigned directly.
- You have enough available licenses for the license plan you're assigning. If you don't have enough licenses, some users might not be assigned the new license plan. You can check the number of available licenses.
- Always confirm users don't have assigned service licenses that can conflict with the desired license or prevent removal of the current license. For example, a license from a service such as Workplace Analytics or Project Online that has a dependency on other services.
- If you manage groups on-premises and sync them into Microsoft Entra ID via Microsoft Entra Connect, then you add or remove users by using your on-premises system. It can take some time for the changes to sync with Microsoft Entra ID to be picked up by group licensing.
- If you're using Microsoft Entra dynamic group memberships, you add or remove users by changing their attributes, but the update process for license assignments remains the same.

10. Exercise - change user license assignments

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/10-exercise-change-user-license-assignments>

Exercise - change user license assignments

Completed

Exercise environment needs - this lab assumes you have a basic Microsoft Entra tenant with at least User Administrator rights to complete it. You can get a free trial subscription at [Try Microsoft Azure for free](#).

Create a new user in Microsoft Entra ID

1. Browse to the [Microsoft Entra admin center](#).
2. In the left navigation, under **Identity**, select **Users**.
3. In the Users page, on the menu, select **+ New user** and then **Create new user**.

4. Create a user using the following information:

Setting	Value
User name	DominiqueK
Name	Dominique Koch
First name	Dominique
Last name	Koch
Password	Make a unique password for the user
Usage location	Select your preferred usage location

5. When complete, verify the account for Dominique Koch is shown in the **All users** list.

Update user license assignments

License assignment to individual users is managed through the Microsoft 365 admin center.

1. Open the [Microsoft 365 admin center](#).
2. Select **Billing**, then select **Licenses**.
3. Select an available license from the list.
4. Select **Licensed users** from the menu near the top of the page.
5. Select **+ Assign licenses**.
6. Search for and select **Dominique Koch**, then select **Assign** at the bottom of the page.
7. When complete, verify the license is listed on Dominique Koch's profile in the Microsoft Entra admin center under **Identity** > **Users** > select the user > **Licenses**.

11. Create custom security attributes

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/11-create-custom-security-attributes>

Create custom security attributes

Completed

Home > Security | Getting started >

Custom attributes

[+ Add attribute set](#) [Refresh](#) [Got feedback?](#)

Search attribute set name

Attribute set name	Description
Banker	This person can write checks against the bank

New attribute set

Add an attribute set to group and manage related custom security attributes. All custom security attributes must be a part of an attribute set. [Learn more](#)

Attribute set name *

Description

Maximum number of attributes

New attribute

Add a custom security attribute (key-value pair) to your directory that you can later assign to Microsoft Entra objects, such as users or applications. [Learn more](#)

Attribute name *

Description

Data type *

Allow multiple values to be assigned ☐ Yes ☒ No

Only allow predefined values to be assigned ☐ Yes ☒ No

Predefined values

Value	Is active?
No results	

What is a custom security attribute?

Custom security attributes in Microsoft Entra ID are business-specific attributes (key-value pairs) that you can define and assign to Microsoft Entra objects. These attributes can be used to store information, categorize objects, or enforce fine-grained access control over specific Azure resources.

Why use custom security attributes?

- Extend user profiles, such as add Hourly Salary to all my employees.
- Ensure only administrators can see the Hourly Salary attribute in my employees' profiles.
- Categorize hundreds or thousands of applications to easily create a filterable inventory for auditing.
- Grant users access to the Azure Storage blobs belonging to a project.

What can I do with custom security attributes?

- Define business-specific information (attributes) for your tenant.
- Add custom security attributes to Microsoft Entra users and enterprise applications (service principals).
- Manage Microsoft Entra objects using custom security attributes with queries and filters.
- Provide attribute governance so attributes determine who can get access.

Custom security attributes are **not** supported in Microsoft Entra Domain Services, SAML token claims, or JSON Web Token (JWT) claims.

Features of custom security attributes

- Available tenant-wide
- Include a description
- Support different data types: Boolean, integer, string
- Support single value or multiple values
- Support user-defined free-form values or predefined values

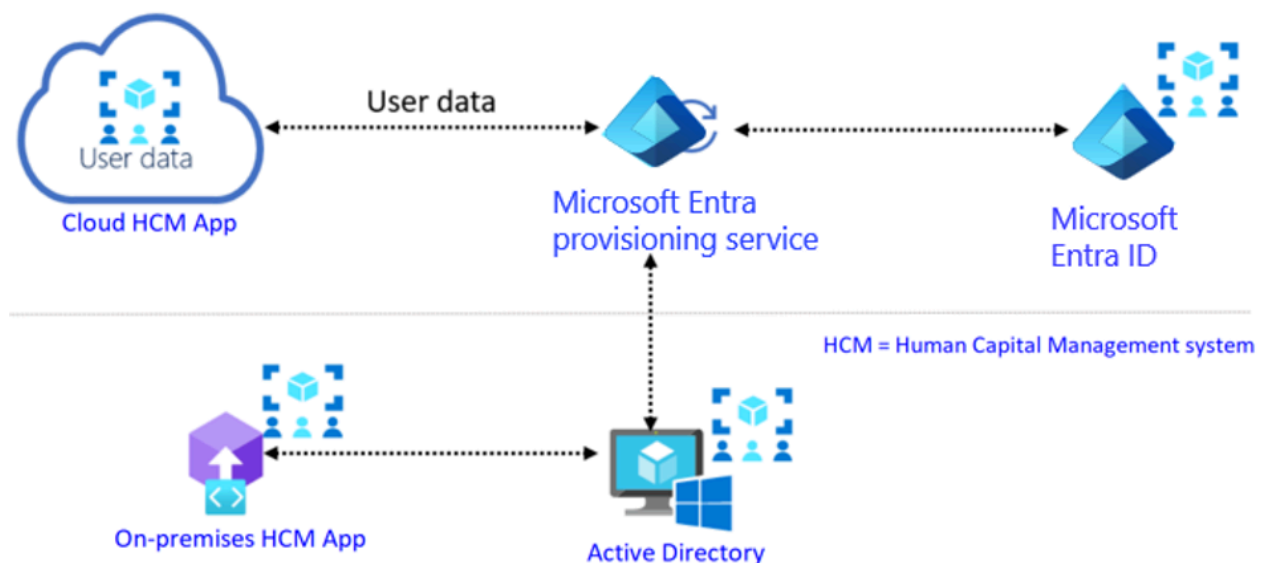
- Assign custom security attributes to directory synced users from an on-premises Active Directory

12. Explore automatic user creation

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/12-explore-automatic-user-creation>

Explore automatic user creation

Completed



Components of SCIM (System for Cross-Domain Identity Management)

- **HCM system** - Applications and technologies that enable Human Capital Management process and practices that support and automate HR processes throughout the employee lifecycle.
- **Microsoft Entra Provisioning Service** - Uses the SCIM 2.0 protocol for automatic provisioning. The service connects to the SCIM endpoint for the application, and uses the SCIM user object schema and REST APIs to automate provisioning and deprovisioning of users and groups.
- **Microsoft Entra ID** - User repository used to manage the lifecycle of identities and their entitlements.

- **Target system** - Application or system that has SCIM endpoint and works with the Microsoft Entra provisioning to enable automatic provisioning of users and groups.

Why use SCIM?

System for Cross-Domain Identity Management (SCIM) is an open standard protocol for automating the exchange of user identity information between identity domains and IT systems. SCIM ensures that employees added to the Human Capital Management (HCM) system automatically have accounts created in Microsoft Entra ID or Windows Server Active Directory. User attributes and profiles are synchronized between the two systems, updating, or removing users based on the user status or role change.

The key is keeping your identity systems up to date. If a user can be automatically deprovisioned from Microsoft Entra ID as soon as they're removed from your HR systems, you have less worry about a possible breach.

API-driven inbound provisioning

Not all HR systems expose a SCIM endpoint. For these scenarios, Microsoft Entra ID supports **API-driven inbound provisioning**, which reached general availability in March 2024. Instead of requiring the source system to push data via SCIM, any automation tool, or script can retrieve workforce data from any system of record and send it to the Microsoft Entra provisioning API. Supported authoritative sources include Workday, SAP SuccessFactors, and any custom HR system integrated via the API. This approach gives organizations flexibility to automate identity lifecycle management regardless of their HR platform's native integration capabilities.

13. Module assessment

<https://learn.microsoft.com/en-us/training/modules/create-configure-manage-identities/13-knowledge-check>

Module assessment

Completed

14. Summary and resources

Summary and resources

Completed

You completed this module, you are able to:

- Create, configure, and manage users
- Create, configure, and manage groups
- Manage licenses
- Configure and manage device registration
- Explore custom security attributes and automatic account provisioning

Resources

Use these resources to discover more:

- [Quickstart: Create and assign a user account](#)
- [Bulk create users in Microsoft Entra ID](#)
- [Create a basic group and add members using Microsoft Entra ID](#)
- [Create or update a dynamic membership group in Microsoft Entra ID](#)
- [What is Microsoft Entra Cloud Sync?](#)
- [Manage license requests](#)
- [Assign licenses to users - Microsoft 365 Admin Center](#)
- [Plan Microsoft Entra device deployment](#)
- [API-driven inbound provisioning concepts](#)

Describe the core architectural components of Azure

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/1-introduction>

Introduction

Completed

In this module, you'll be introduced to the core architectural components of Azure. You'll learn about Azure's physical layout: datacenters, availability zones, and regions; and you'll learn about Azure's management structure: resources and resource groups, subscriptions, and management groups.

Learning objectives

After completing this module, you'll be able to:

- Describe Azure regions, region pairs, and sovereign regions.
- Describe Availability Zones.
- Describe Azure datacenters.
- Describe Azure resources and Resource Groups.
- Describe subscriptions.
- Describe management groups.
- Describe the hierarchy of resource groups, subscriptions, and management groups.

2. What is Microsoft Azure

What is Microsoft Azure

Completed

Azure is a continually expanding set of cloud services that help you meet current and future IT challenges. Azure gives you the freedom to build, manage, and deploy applications on a massive global network using your favorite tools and frameworks.

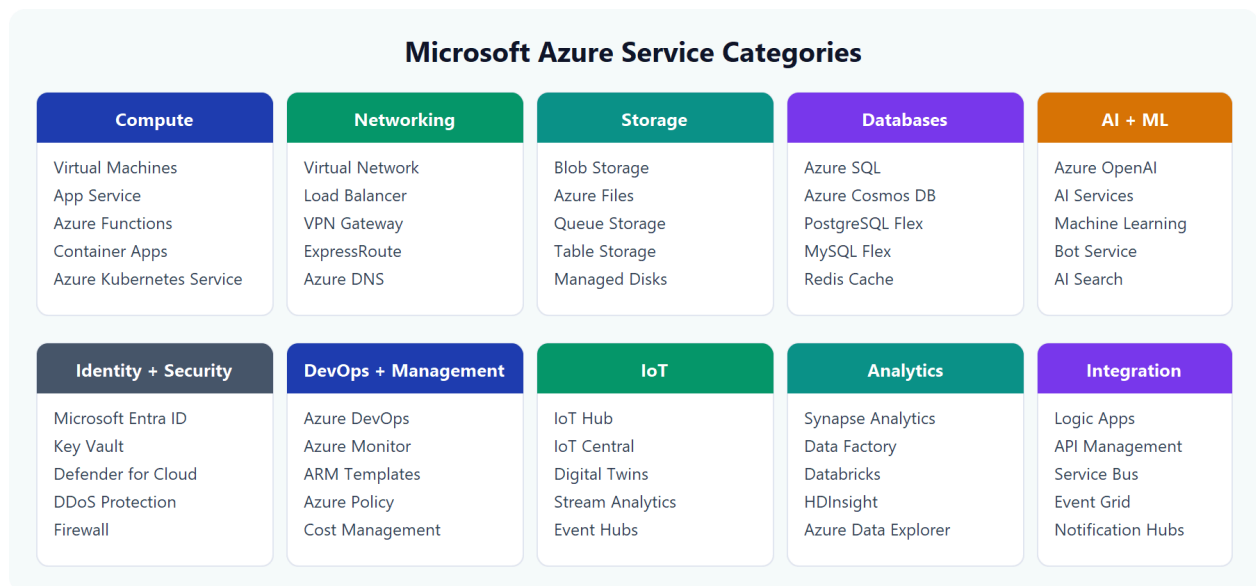
What does Azure offer?

Limitless innovation. Build intelligent apps and solutions with advanced technology, tools, and services to take your operations to the next level. Seamlessly unify your technology to simplify platform management and deliver innovations efficiently and securely on a trusted cloud.

- **Bring ideas to life:** Build on a trusted platform to advance your team's capabilities with industry-leading AI and cloud services.
- **Seamlessly unify:** Efficiently manage all your infrastructure, data, analytics, and AI solutions across an integrated platform.
- **Innovate on trust:** Rely on trusted technology from a partner who's dedicated to security and responsibility.

What can I do with Azure?

Azure provides hundreds of services that enable you to do everything from running your existing applications on virtual machines to exploring new software paradigms, such as intelligent bots and generative AI.



Many teams start exploring the cloud by moving their existing applications to virtual machines (VMs) that run in Azure. Migrating your existing apps to VMs is a good start, but the cloud is much more than a different place to run your VMs.

As your skills grow, you can modernize one workload at a time, such as moving from manually managed servers to managed databases, autoscaling web apps, or event-driven services.

Practical example

Suppose your organization runs an internal app with seasonal demand spikes. In Azure, you can host the app on virtual machines or managed app services, store data in managed databases, and monitor health from a centralized dashboard. As demand increases, you can scale resources up or out and then scale back when demand drops so you're not paying for unused capacity year-round.

For example, Azure provides Azure AI services and Azure OpenAI Service so you can add language, vision, speech, and generative AI capabilities to your applications. It also provides Azure Machine Learning, Internet of Things (IoT) services, and storage solutions that dynamically grow to accommodate massive amounts of data. Azure services enable solutions that aren't feasible without the power of the cloud.

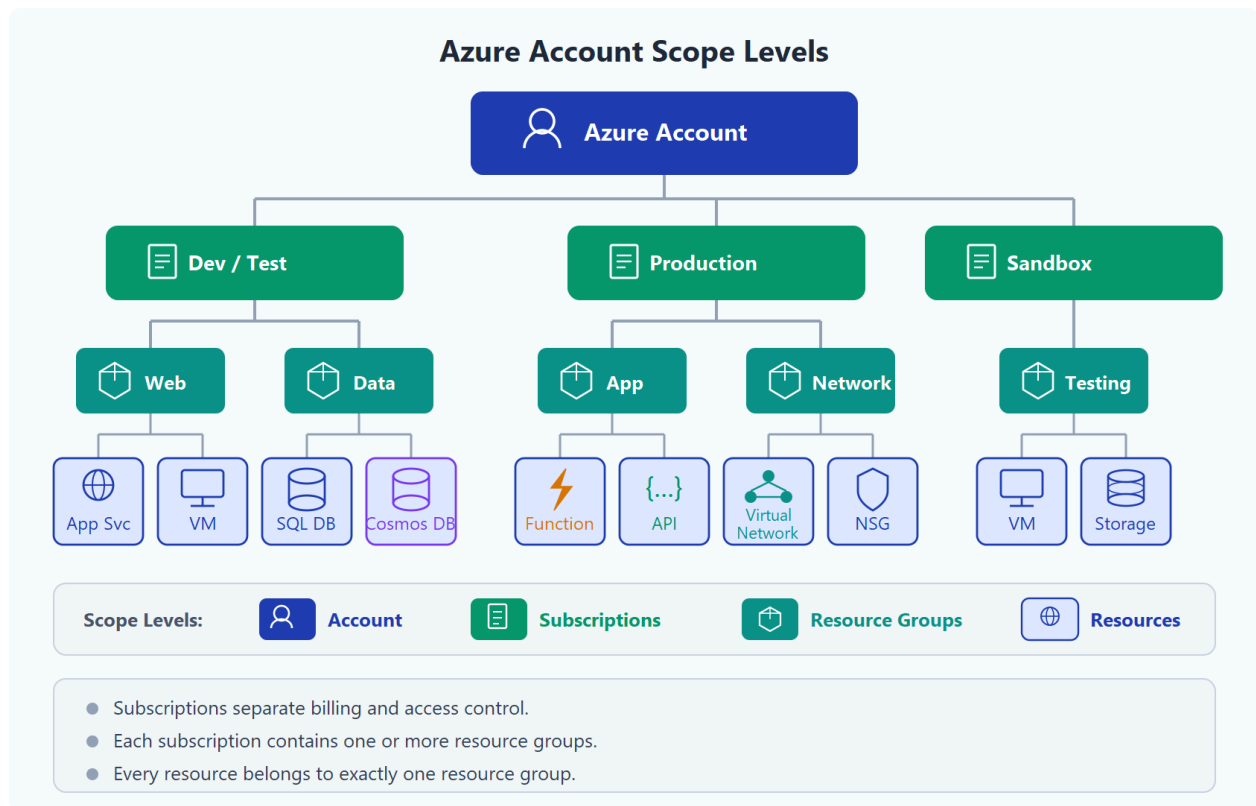
3. Get started with Azure accounts

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/3-get-started-azure-accounts>

Get started with Azure accounts

Completed

To create and use Azure services, you need an Azure subscription. When you're working with your own applications and workloads, you create an Azure account, and a subscription is created for you. After you've created an Azure account, you're free to create additional subscriptions. For example, your team might use a single Azure account and separate subscriptions for development, testing, and production workloads. After you've created an Azure subscription, you can start creating Azure resources within each subscription.



If you're new to Azure, you can sign up for a free account on the Azure website to start exploring at no cost to you. When you're ready, you can choose to upgrade your free account. You can also create a new subscription that enables you to start paying for Azure services you need beyond the limits of a free account.

Create an Azure account

You can purchase Azure access directly from Microsoft by signing up on the Azure website or through a Microsoft representative. You can also purchase Azure access through a Microsoft partner. Cloud Solution Provider partners offer a range of complete managed-cloud solutions for Azure.

What is the Azure free account?

The Azure free account includes:

- Free access to popular Azure products for 12 months.
- A credit to use for the first 30 days.
- Access to more than 65 services that are always free.

The [Azure free account](#) is an excellent way for new users to get started and explore. To sign up, you need a phone number, a credit card, and a Microsoft or GitHub account. The credit card information is used for identity verification only. You won't be charged for any services until you upgrade to a paid subscription.

What is the Azure free student account?

The Azure free student account offer includes:

- Free access to certain Azure services for 12 months.
- A credit to use in the first 12 months.
- Free access to certain software developer tools.

The [Azure free student account](#) is an offer for students that gives \$100 credit and free developer tools. Also, you can sign up without a credit card.

If you're practicing by creating resources in Azure, monitor usage and remove resources you no longer need to avoid unexpected costs.

4. Describe Azure physical infrastructure

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/5-describe-azure-physical-infrastructure>

Describe Azure physical infrastructure

Completed

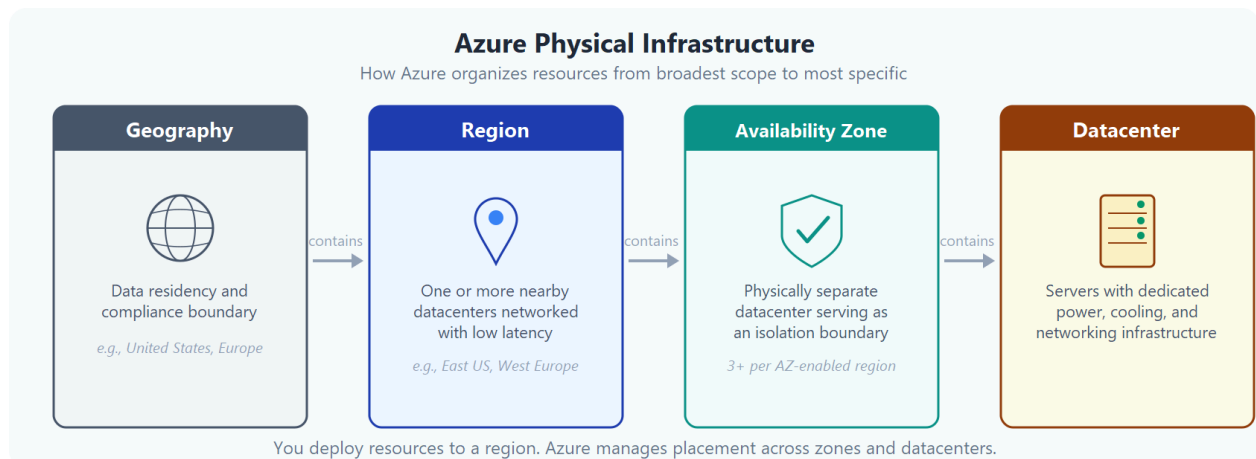
Azure's core architectural components can be broken down into two main groupings: the physical infrastructure and the management infrastructure. This unit covers the physical side — how Azure organizes its datacenters, regions, and availability zones to deliver reliable services worldwide.

Physical infrastructure

The physical infrastructure for Azure starts with datacenters. These datacenters are facilities with servers arranged in racks, with dedicated power, cooling, and networking infrastructure — similar to an on-premises datacenter, but at a much larger scale.

As a global cloud provider, Azure has datacenters around the world. However, you don't interact with individual datacenters directly. Instead, datacenters are grouped into Azure Regions and Azure Availability Zones that provide resiliency and reliability for your workloads.

The [Global infrastructure](#) site gives you a chance to interactively explore the underlying Azure infrastructure.



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.

When you deploy a resource in Azure, you'll often need to choose the region where you want your resource deployed.

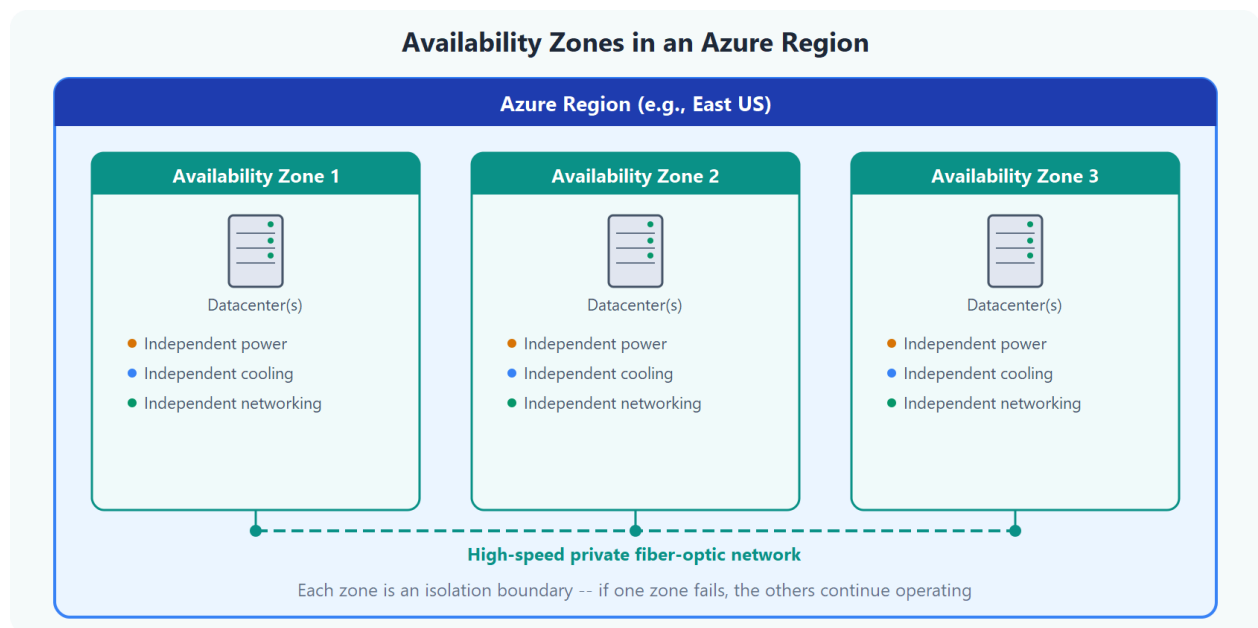
Note

Some services or virtual machine (VM) features are only available in certain regions, such as specific VM sizes or storage types. 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 with independent 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.



Important

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.

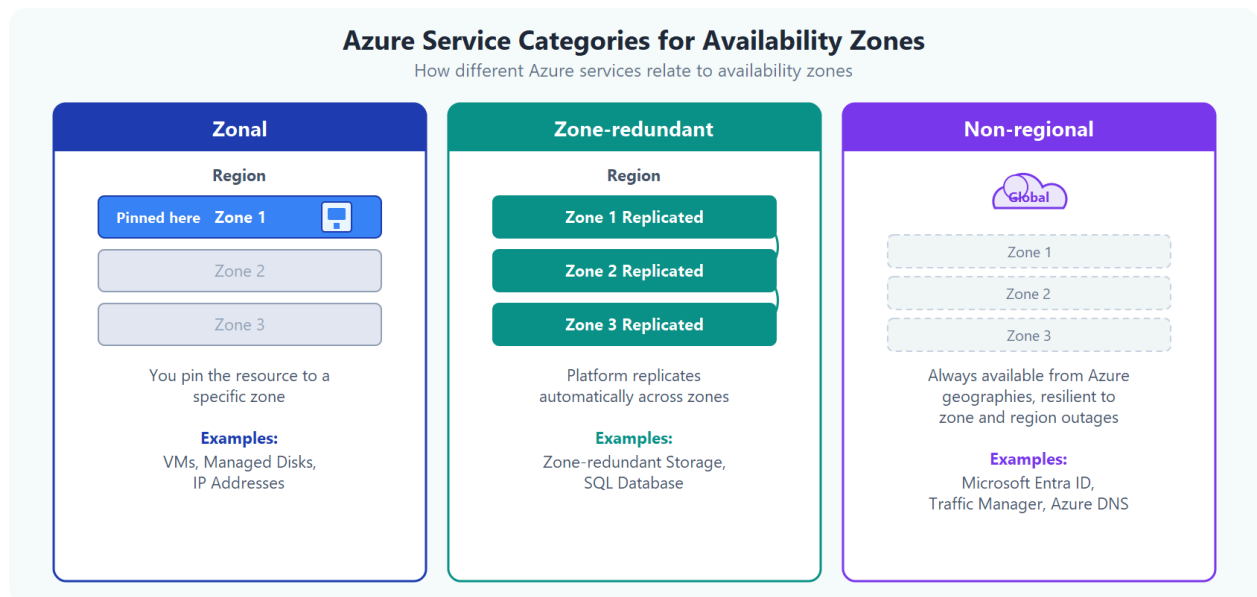
Use availability zones for your workloads

When you run your own on-premises infrastructure, setting up redundancy means buying and maintaining duplicate hardware. With Azure, you can protect your workloads by spreading them across availability zones within a region.

You place your VMs, storage, databases, and other resources in one availability zone and replicate them to other zones within the same region. Keep in mind that there could be a cost to duplicating your services and transferring data between zones.

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.



Even with the additional resiliency that availability zones provide, it's possible that an event could be so large that it impacts multiple availability zones in a single region. To provide even further resiliency, Azure has Region Pairs.

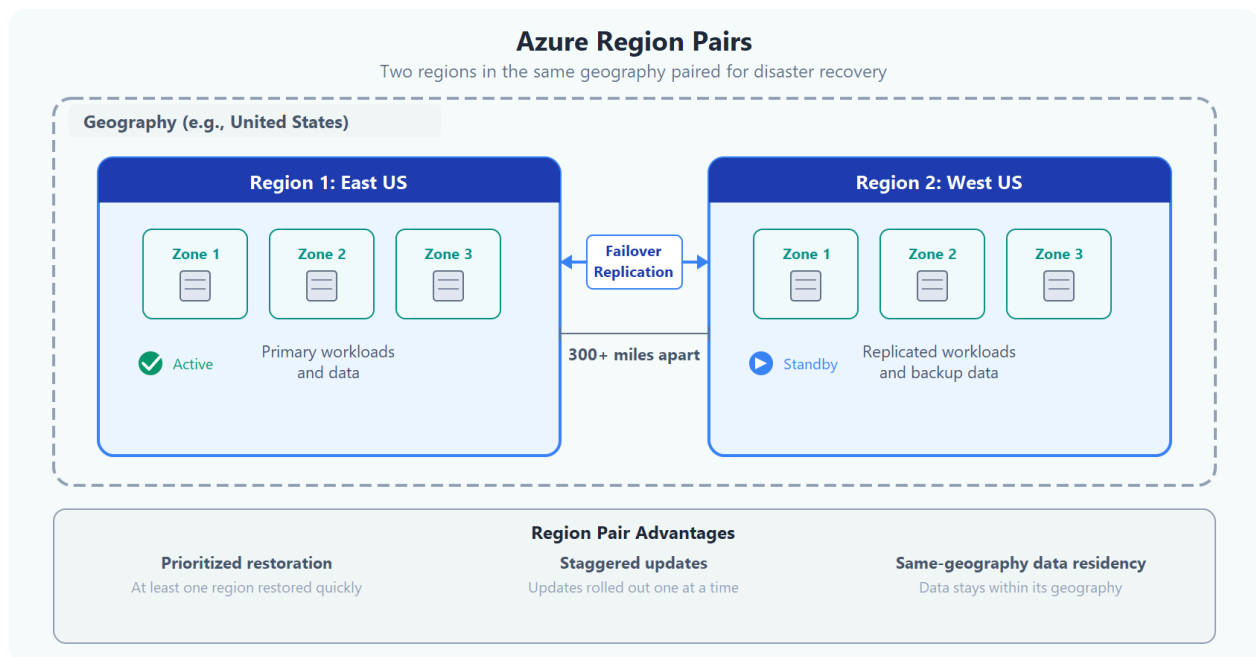
Region pairs

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.

Important

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.

Examples of region pairs in Azure are West US paired with East US and Southeast Asia paired with East Asia. Because the pair of regions is directly connected and far enough apart to be isolated from regional disasters, you can use them to provide reliable services and data redundancy.



Additional advantages of region pairs:

- If an extensive Azure outage occurs, one region out of every pair is prioritized to make sure at least one is restored as quickly as possible for applications hosted in that region pair.
- Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application outage.
- Data continues to reside within the same geography as its pair (except for Brazil South) for data-residency and compliance purposes.

Important

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 Brazil South, are paired in only one direction. In a one-direction pairing, the Primary region does not provide backup for its secondary region. 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. Additionally, some regions (such as Italy North, Poland Central, and Israel Central) don't have a traditional region pair and instead rely on availability zones and geo-redundant storage for resiliency.

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:

- US DoD Central, US Gov Virginia, US Gov Arizona, 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.

5. Describe Azure management infrastructure

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/6-describe-azure-management-infrastructure>

Describe Azure management infrastructure

Completed

The management infrastructure includes Azure resources and resource groups, subscriptions, and accounts. Understanding this hierarchy helps you organize resources, control who can access what, and manage costs as your Azure usage grows.

Azure resources and resource groups

A resource is the basic building block of Azure. Anything you create, provision, or deploy is a resource. VMs, virtual networks, databases, and Azure AI services are all examples of resources.



Resource groups are groupings of resources. Every resource must belong to exactly one resource group. You can move some resources between groups, but a resource is only associated with one

group at a time. Resource groups can't be nested, and they can't be renamed after creation, so choose a clear naming convention from the start.

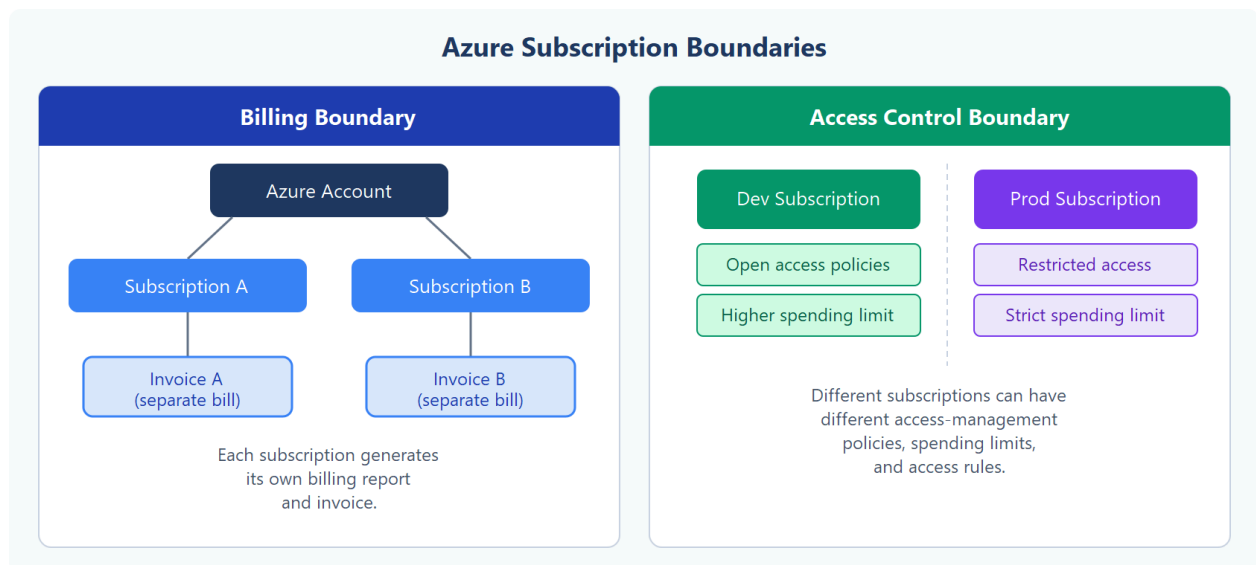
Actions you apply to a resource group affect all resources inside it. Deleting a resource group deletes everything in it. Granting or denying access applies to all its resources.

For example, if you're setting up a temporary dev environment, grouping all the resources together lets you delete the entire group when you're done. If you're running multiple projects, create a separate resource group for each so each team only sees and manages its own resources.

There are no hard rules for structuring resource groups — choose the approach that works best for your situation.

Azure subscriptions

In Azure, subscriptions are a unit of management, billing, and scale. Subscriptions let you organize resource groups and control billing separately from access.



Using Azure requires an Azure subscription. A subscription provides access to Azure products and services and serves as a billing unit. An Azure subscription links to an Azure account, which is an identity in Microsoft Entra ID or in a directory that Microsoft Entra ID trusts.

An account can have multiple subscriptions, but only one is required. In a multi-subscription account, you can configure different billing models and access policies. There are two types of subscription boundaries:

- **Billing boundary:** Determines how an Azure account is billed. You can create multiple subscriptions for different billing requirements. Azure generates separate billing reports and invoices for each subscription.
- **Access control boundary:** Azure applies access-management policies at the subscription level. For example, you might create one subscription for your development work and another for production, each with different spending limits and access rules.

Create additional Azure subscriptions

You might create additional subscriptions to separate:

- **Environments:** Subscriptions for lifecycle stages such as sandbox, development, test, and production. Access control occurs at the subscription level, making this a natural boundary.
- **Team and workload boundaries:** Give each project its own subscription so costs are easy to track, or separate sandbox environments from production.
- **Billing:** Create subscriptions to track costs separately — for instance, one for production workloads and another for development and testing.

Azure management groups

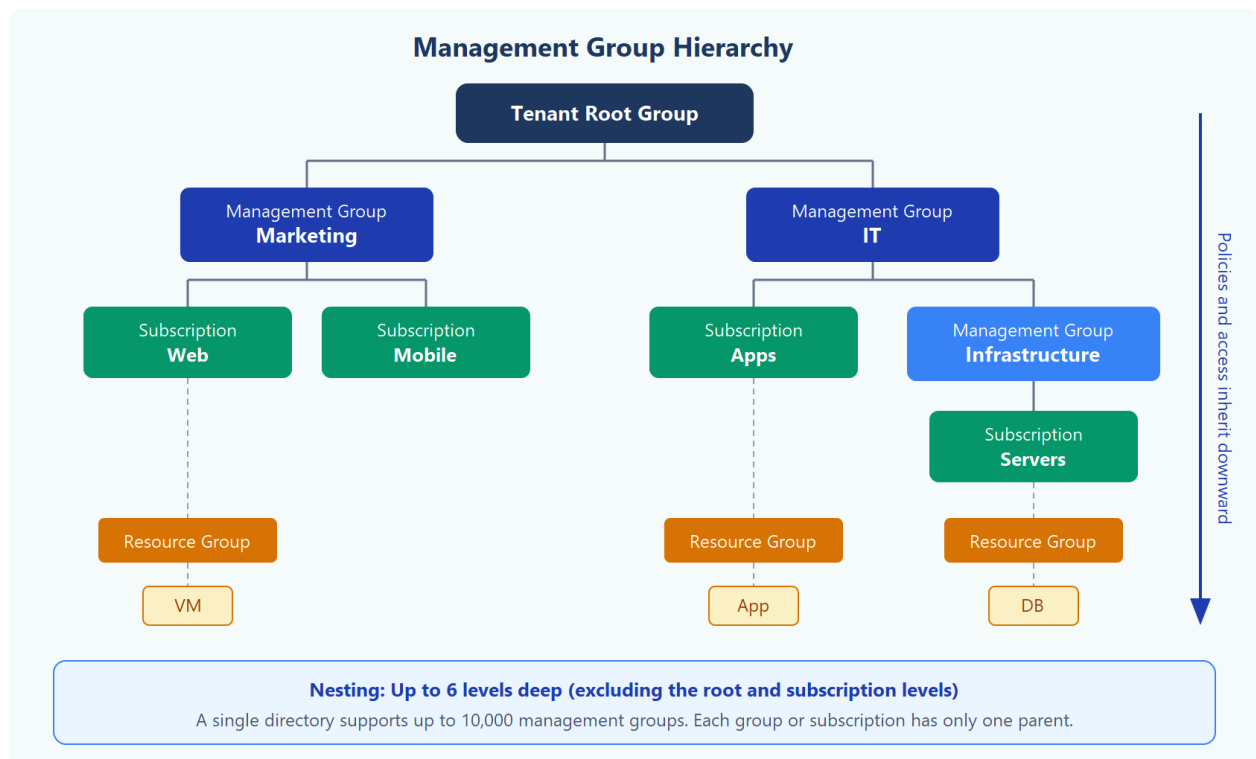
Resources go into resource groups, and resource groups go into subscriptions. For a small environment, that's enough. But when you have many subscriptions across multiple teams or geographies, you need a way to manage access and policies at a higher level.

Azure management groups sit above subscriptions. You organize subscriptions into management groups and apply governance conditions — like access policies or compliance rules — to the group. All subscriptions in a management group automatically inherit those conditions, just as resources inherit settings from their resource group. Management groups can be nested up to six levels deep (not counting the root level or the subscription level), letting you build a hierarchy that mirrors your organization.

Every Microsoft Entra tenant has a single top-level Tenant Root Group. All other management groups and subscriptions fold up to this root group, which lets you apply governance policies globally.

Management group, subscriptions, and resource group hierarchy

You can build a flexible structure of management groups and subscriptions to organize your resources into a hierarchy for unified policy and access management.



Examples of how you could use management groups:

- **Apply a policy across subscriptions.** You could limit VM locations to the US West Region in a group called Production. This policy inherits to all subscriptions under that management group and applies to all VMs in those subscriptions. The resource or subscription owner can't override it, which strengthens governance.
- **Grant access to multiple subscriptions at once.** By placing subscriptions under a management group, you can create one Azure RBAC assignment on the group. All sub-management groups, subscriptions, resource groups, and resources underneath inherit those permissions — no need to script Azure RBAC across individual subscriptions.

Important facts about management groups:

- A single directory supports up to 10,000 management groups.
- Each management group and subscription can have only one parent.

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/8-knowledge-check>

Module assessment

Completed

7. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-core-architectural-components-of-azure/9-summary>

Summary

Completed

In this module, you learned about the physical and management structure of Microsoft Azure. You were introduced to the relationship between datacenters, availability zones, and regions. You explored how the infrastructure supports the benefits of the cloud, such as high availability and reliability. You also learned about the management infrastructure of Azure. You explored how resources and resource groups are related, and how subscriptions and management groups can help manage resources.

Learning objectives

You should now be able to:

- Describe Azure regions, region pairs, and sovereign regions.
- Describe Availability Zones.
- Describe Azure datacenters.
- Describe Azure resources and Resource Groups.
- Describe subscriptions.
- Describe management groups.
- Describe the hierarchy of resource groups, subscriptions, and management groups.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Build a concept map that connects regions, Availability Zones, datacenters, resources, resource groups, subscriptions, and management groups."
- "Design a subscription and management-group structure for an organization with multiple departments and compliance boundaries."

- "Explain region pairs and sovereign regions, then recommend a resiliency approach for a workload with strict data residency requirements."

Azure Policy initiatives

1. Azure Policy design principles

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/azure-policy-design-principles>

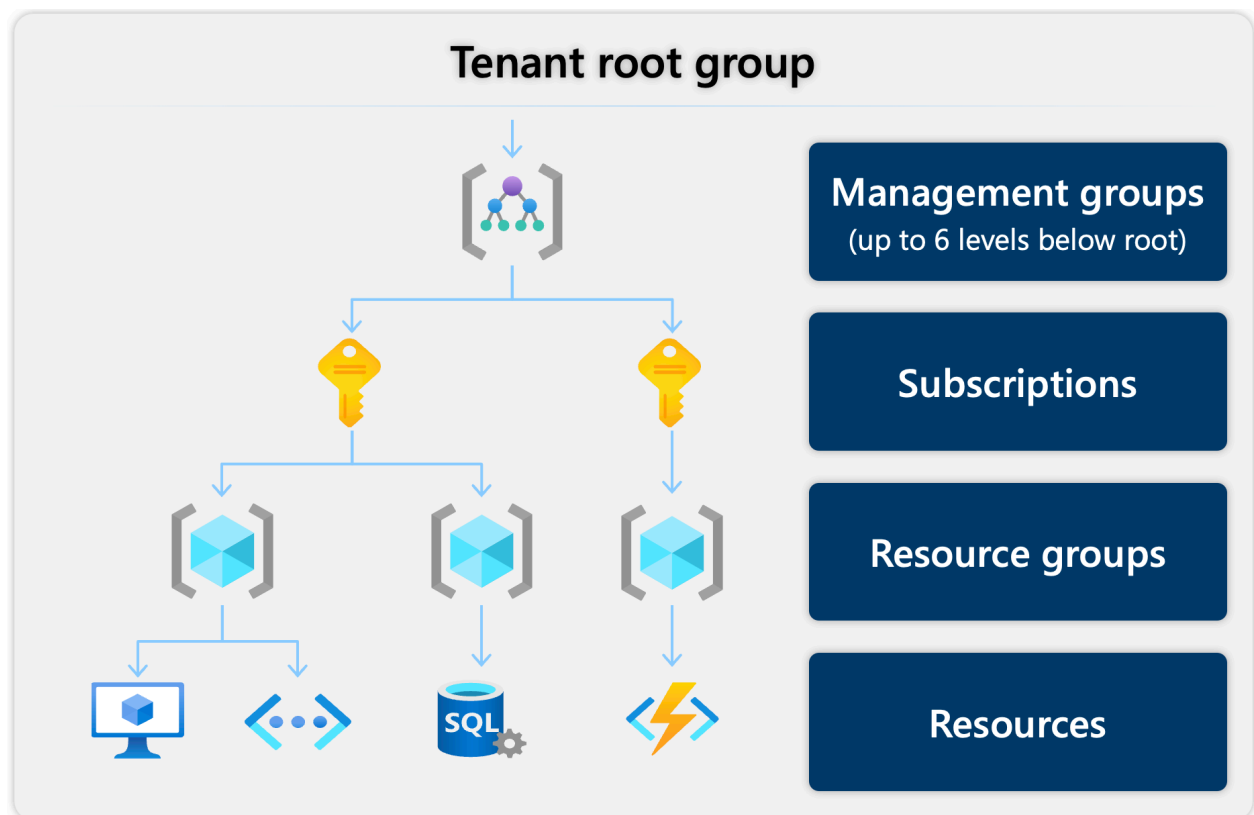
Azure Policy design principles

Completed

Governance provides mechanisms and processes to maintain control over your applications and resources in Azure. It involves planning your policy in Azure Policy and setting strategic priorities. While designing your policy, you must organize your cloud-based resources to secure, manage, and track costs that are related to your workloads.

Hierarchy for governance

Azure provides four levels of management to establish proper governance: Management groups, Subscriptions, Resource groups, and Resources. You can build a flexible structure of management groups and subscriptions to organize your resources into a hierarchy for unified policy and access management. The following diagram shows an example of creating a hierarchy for governance by using management groups.



The Azure structure starts with the tenant root group at the top, followed by a hierarchy of management groups, which can extend to six levels beneath the root. The definition of each level in the management hierarchy and relationship between these levels is as follows:

Concept	Description
Resource	A resource is the basic building block of Azure, and it includes instances of services that you create, provision, deploy, and so on. Virtual machines (VMs), virtual networks, databases, AI services, and so on, are considered resources in Azure.
Resource groups	Resource groups are groupings of resources. When you create a resource, you must place it into a resource group. While a resource group can contain many resources, a single resource can only be in one resource group at a time. When you apply an action to a resource group, that action applies to all resources in the resource group. If you delete a resource group, all resources are deleted. If you grant or deny access to a resource group, all resources in the resource group are also granted or denied access.
Subscriptions	In Azure, subscriptions are a unit of management, billing, and scale. Similar to how resource groups are a way of logically organizing resources, subscriptions allow you to logically organize your resource groups and facilitate billing. Each subscription has limits or quotas on the number of resources that you can create and use. Organizations can use subscriptions to manage costs and the resources that users, teams, and projects create. Using Azure requires an Azure subscription. An Azure subscription provides you with authenticated and authorized access to Azure products and services. It also allows you to

Concept	Description
	provision resources. An Azure subscription links to an Azure account, which is an identity in Microsoft Entra ID or in a directory that Microsoft Entra ID trusts.
Management groups	Azure management groups provide a level of scope above subscriptions. If you have many subscriptions, you might need a way to efficiently manage access, policies, and compliance for those subscriptions. You organize subscriptions into containers called management groups and then apply governance conditions to the management groups. Management groups give you enterprise-grade management at a large scale, no matter what type of subscriptions you might have. Management groups can be nested.

You can apply management settings at any of these levels of scope. The level that you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For example, when you apply a policy to the subscription, the policy is applied to all resource groups and resources in that subscription. When you apply a policy on the resource group, that policy is applied to that resource group and all its resources. However, another resource group won't have that policy assignment. All subscriptions within a management group automatically inherit the conditions that are applied to the management group.

Introduction to Azure Resource Manager

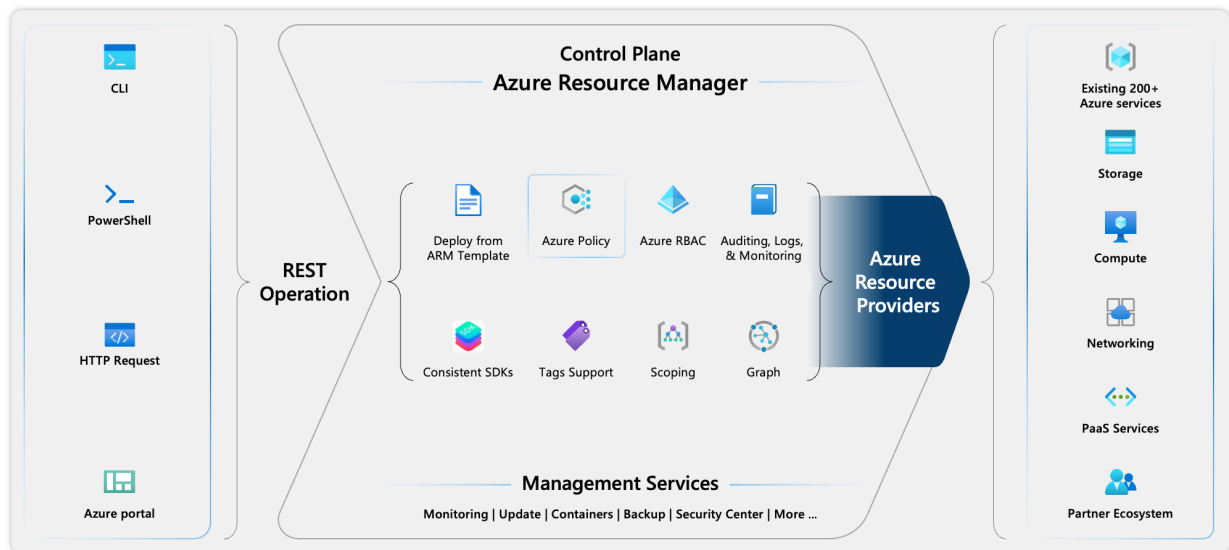
Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that allows you to create, update, and delete resources in your Azure account.

Azure operations are classified into two main types: control plane and data plane. The control plane helps you manage resources in your subscription, while the data plane allows you to access the capabilities provided by instances of specific resource types.

Control plane

Azure Policy operates in the control plane to enforce rules and compliance on your resources. Azure Resource Manager manages all control plane operations in Azure and includes the different components that are centralized between the different services. Azure Policy is integrated with Azure Resource Manager.

Azure Policy and Azure Resource Manager



Azure Resource Manager manages essential functions, such as template-based deployments, role-based access control (RBAC), auditing, monitoring, and tagging, which provides a unified management experience for Azure resources after deployment. For example, consider a scenario where you have a storage account. With Azure Resource Manager, you can create the storage account and enforce a policy that mandates encryption for all storage accounts.

When you send a control plane request through any of the Azure APIs, tools, or SDKs, Azure Resource Manager receives the request. All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs, and client SDKs. These tools use the same API to process requests, ensuring uniform results and capabilities. The Resource Manager authenticates and authorizes the request before forwarding it to the appropriate Azure resource provider that completes the operation.

Data plane

The data plane is where the actual data operations occur, and Azure Policy ensures that the resources you interact with in the data plane are compliant with your policies. Data plane operations involve direct interaction with the data stored in a resource. Continuing with the previous example, you engage with the storage account to upload or download files. This interaction is handled directly by the data plane of the storage account rather than being managed by Azure Resource Manager.

Data plane operations aren't limited to REST API. Requests are made directly to the data endpoints of services (for example, accessing data in Microsoft Azure Storage, querying an SQL database, or reading secrets from Microsoft Azure Key Vault). Each Azure service handles these requests internally, bypassing Azure Resource Manager, and directly managing the data through its resource provider. Service-specific access controls, such as RBAC or access control lists (ACLs), often manage data plane permissions. The service responds with the data or result of the data operation, ensuring that the requester has the correct permissions.

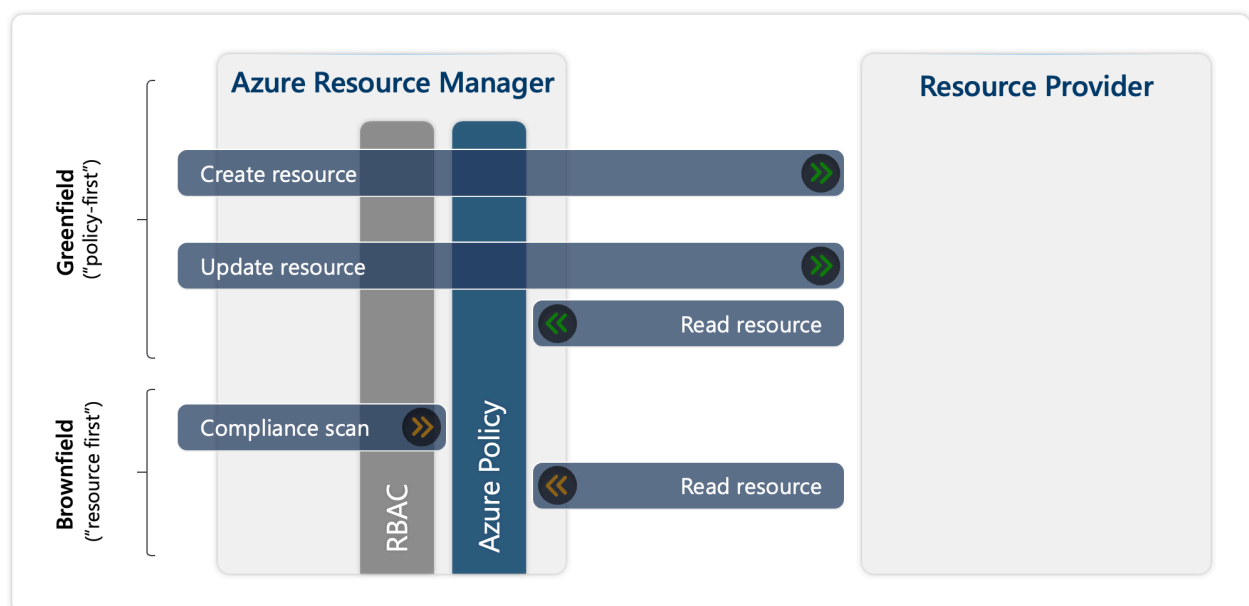
Azure Policy allows individual Azure services to implement an Azure Policy extension, enhancing policy behavior and integration with specific resource providers. Azure Policy currently supports data plane operations through the following resource provider modes:

- **Microsoft.Kubernetes.Data** - Used for managing Kubernetes clusters and components such as pods, containers, and ingresses.
- **Microsoft.KeyVault.Data** - Used for managing vaults and certificates in Azure Key Vault.
- **Microsoft.Network.Data** - Used for managing Microsoft Azure Virtual Network Manager custom membership policies by using Azure Policy.
- **Microsoft.ManagedHSM.Data** - Used for managing Azure Key Vault Managed HSM keys by using Azure Policy.
- **Microsoft.DataFactory.Data** - Used for using Azure Policy to deny Microsoft Azure Data Factory outbound traffic domain names.
- **Microsoft.MachineLearningServices.v2.Data** - Used for managing Microsoft Azure Machine Learning model deployments. This Resource Provider mode reports compliance for newly created and updated components.

For more information, see [Resource Provider Modes](#).

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.

For example, you're creating a new resource by calling an HTTPS REST API into Azure Resource Manager, that is, targeting a specific resource provider. In Azure Resource Manager, the request goes

through different layers, including role-based access control (RBAC) and Azure Policy. These layers are only two of the many other layers, but it's important to understand that Azure Policy comes after RBAC. If you don't have permissions to run a given operation, then Azure Policy isn't considered or called because it would fail on the RBAC stage. If you have permissions, the request goes through Azure Policy and is evaluated against any policy. For resource updates, you send only the changes (delta) in the request payload in the existing resource. Azure Policy requires full visibility of the current state of the resource. Azure Policy merges the delta that you're sending by reading the current state. Then, the target state is obtained as a result of the merger, and that's what gets evaluated against your policies.

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

In this case, policy evaluation happens through a compliance scan, which runs automatically every 24 hours, or it can be manually triggered. The duration of the scan is unpredictable, but after it's completed, the compliance state of existing resources is updated. To perform this evaluation, Azure Policy reads all existing resources in the scope. You can create an Azure policy that prohibits resources from being created outside of a certain region, such as West Europe. Existing resources outside this region aren't deleted but are flagged as noncompliant after the policy is applied, and future attempts to create resources outside West Europe fail.

2. Summary

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/summary>

Summary

Completed

Azure Policy is a crucial component of the governance model in the Cloud Adoption Framework for Azure, which is designed to balance control and stability with speed and results. It helps you enforce organizational and regulatory standards and assess compliance at scale through built-in and custom policies and policy initiatives.

The module covered the hierarchical organization of Azure resources, policy operations in Greenfield and Brownfield scenarios, and the various components of policy definitions. You also delved into the evaluation and effects of policies, safe deployment practices, and integration with Event Grid for automated actions based on policy state changes. Key points included:

- Importance of careful policy design
- Testing to ensure effective governance without disrupting operations
- Logical operators and conditions in policy evaluation

- Supported effect types such as *disabled*, *modify*, *deny*, *audit*, *deployIfNotExists*, and *manual*

Additionally, the module emphasized starting with *enforcementMode* deactivated for new policies to test their impact and then deploying policies in rings to gradually expand to production environments.

For more information, see the [Azure Policy](#) documentation.

3. Evaluation of resources through Azure Policy

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/azure-policies-evaluation-resources>

Evaluation of resources through Azure Policy

Completed

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

For more information, see [Evaluation triggers](#).

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*.

When you assign a new policy, a delay can occur in the policy taking effect, which can be up to 30 minutes. The Azure Resource Manager cache holds session data, and it can take time for the policy to propagate in the same session. To bypass the caching delay, you can sign out and sign back in to refresh the Azure Resource Manager cache, which ensures that the new policy is applied immediately to the defined scope.

After the scan starts, several factors influence how long it takes for a compliance scan to complete:

- **Policy definitions** - The size and complexity of the policy definitions can increase scan time.
- **Number of policies** - The more policies applied, the longer the scan might take.
- **Scope size** - The size of the resource scope assigned to the policy also plays a role.
- **System load** - Compliance scans are a low-priority operation, meaning that if the system is busy with more critical tasks, the scan might take longer. The system prioritizes interactive and high-importance operations, so scans might take several minutes, or tens of minutes, even in smaller environments.
- **Synchronous scan (Low-Priority Execution)** - Because compliance scans are synchronous and assigned a low priority in Azure's system, they're delayed if the system is busy. This scan can significantly extend the time it takes for the scan to complete, even for smaller scopes or policies.

This understanding of the compliance scan process and potential delays allows you to better manage the application and avoid unnecessary waiting, especially in environments with complex or extensive policy definitions.

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.

For more information on when the policies return these states for any particular resource, see [Azure Policy compliance 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.

The following table describes this property's values.

Mode	JSON value	Type	Remediate manually	Activity log entry	Description
Enabled	Default	string	Yes	Yes	The policy effect is enforced during resource creation or update.
Disabled	DoNotEnforce	string	Yes	No	The policy effect isn't enforced during resource creation or update.

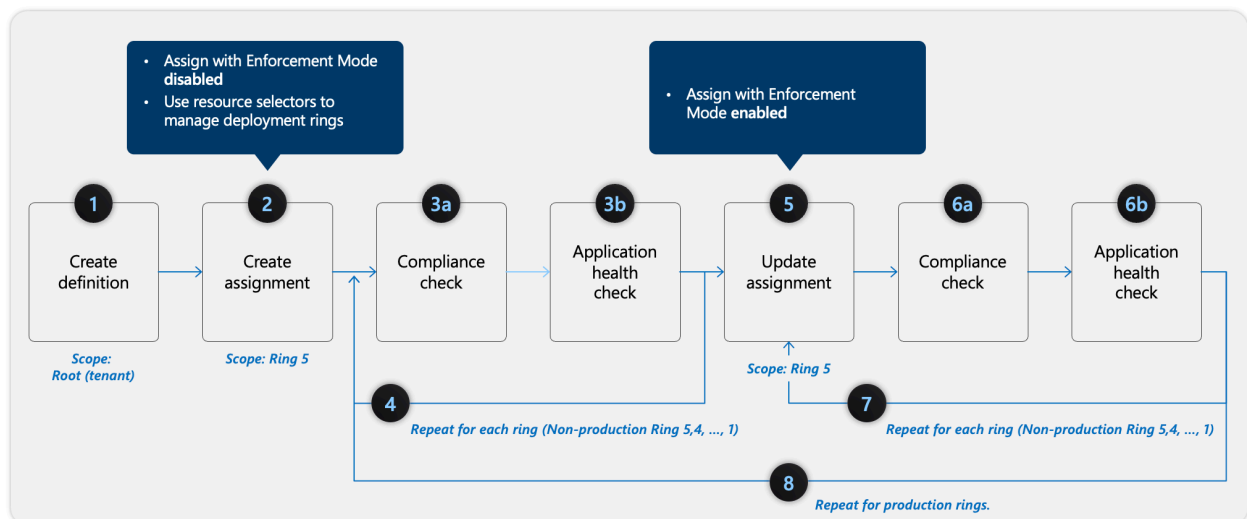
If *enforcementMode* isn't specified in a policy or initiative definition, the value *Default* is used. Remediation tasks can be started for *deployIfNotExists* policies, even when *enforcementMode* is set to *DoNotEnforce*.

Policy enforcement and safe deployment best practices

Without the appropriate knowledge of best practices and proper testing, applying a set of policy to an existing environment that's running production workloads can result in unintended behaviors of policy resources. Treating policy as code (keeping your policy definitions in source control, and whenever a change is made, testing and validating that change) allows you to automate testing and make sure that no manual error factor happens. The best practices framework focuses on minimizing the impact of policy changes while ensuring compliance, and it includes two aspects:

- **First aspect** - Start from Assignments of new policies with *enforcementMode* Disabled. When assigning policies that include deny or modify actions, beginning with *enforcementMode* Disabled allows you to view the compliance state and evaluate policy outcomes without triggering actions or denying operations. This "what-if" scenario minimizes impact and helps identify issues in the new policies or changes without disrupting the environment.
- **Second aspect** - Deploy policies in deployment rings. To control potential negative impacts, policies should be deployed gradually in smaller subsets and then in bigger sets. You can start with test and development environments and then move to production by applying the policy to a small subset first. This strategy helps in testing the policy thoroughly. Gradually expanding the scope (through deployment rings) can cover the full production environment.

The following diagram shows how to apply the safe deployment best practices framework for Azure Policy assignments.



The following steps correspond with the outlined steps in the preceding screenshot:

1. **Create definition** - Begin by defining the policy definition with the scope as root (tenant).
2. **Create assignment** - Define deployment rings (1 to 5) by using resource selectors. Assign the policy to a specific scope (such as a resource group, subscription, or management group) in Ring 5. Assign with *enforcementMode* Disabled to evaluate compliance without enforcing changes.
3. a) **Compliance check** - Verify that the policy is being applied correctly and that the desired compliance state is achieved for the resources in Ring 5.

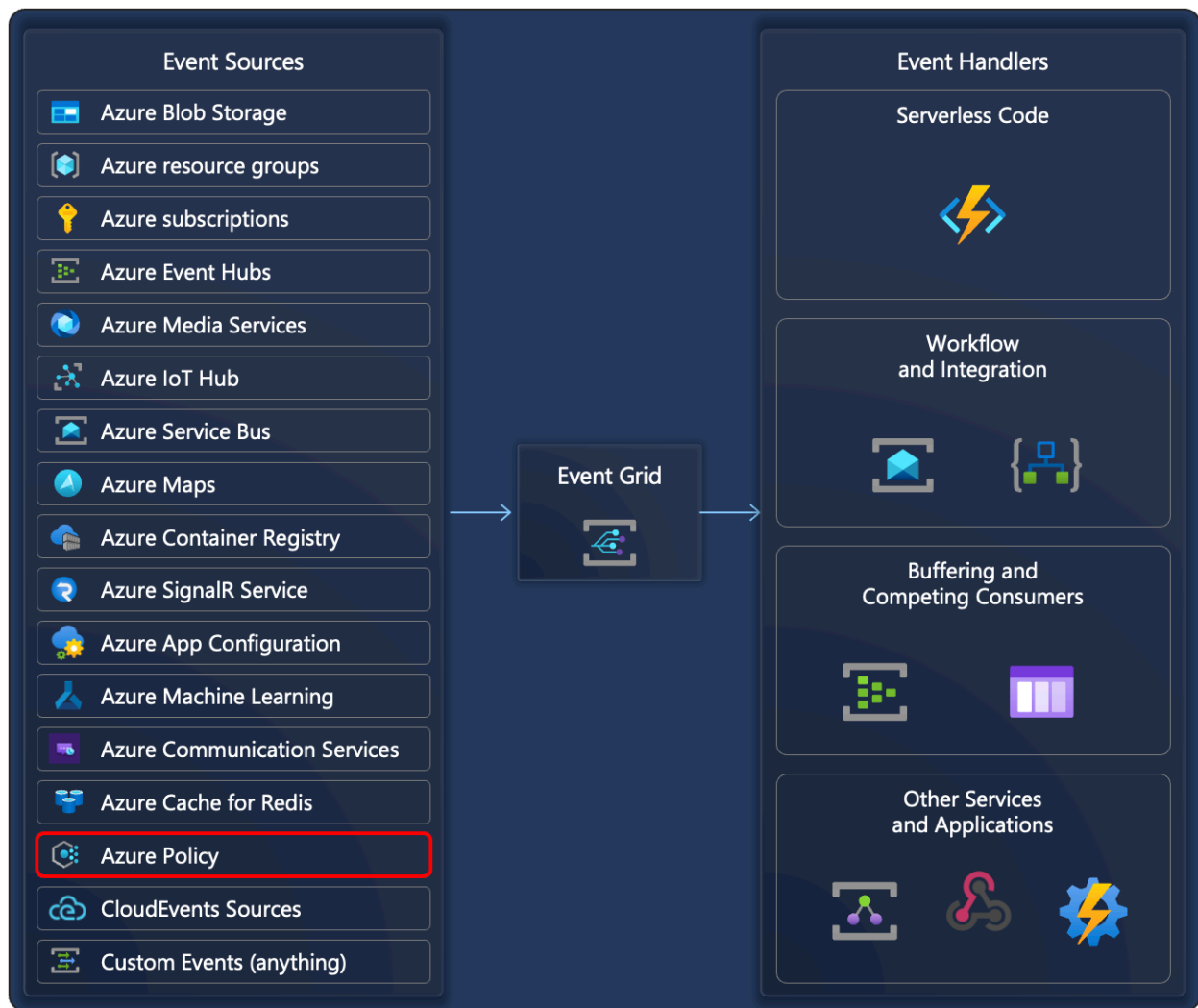
- b) **Application health check** - Assess the impact of the policy for the resources in Ring 5. Ensure that no unexpected side effects exist.
- 4. **Repeat for each ring (Non-production)** - Repeat step 3 for all nonproduction environment rings.
- 5. **Update assignment (Optional)** - If necessary, adjust the policy definition or assignment based on the evaluation of resources of the nonproduction environment and then reassign it to the resources in Ring 5 with the *enforcementMode* Enabled.
- 6. a) **Compliance check** - Reevaluate compliance after making changes (same as step 3a).

b) **Application health check** - Again, verify that the policy isn't causing issues (same as step 3b).
- 7. **Repeat for each ring (Non-production)** - Repeat step 6 for all nonproduction environment rings.
- 8. **Repeat for production rings** - After the policy is validated in a nonproduction environment, gradually deploy it to production environments, starting with a smaller subset (ring) and expanding the scope over time.

For more detailed steps on the safe deployment of Azure Policy assignments with different effects, see [Safe deployment of Azure Policy assignments](#).

Reacting to policy state changes

Azure Policy events allow applications to react to state changes. This integration is done without the need for complicated code or expensive and inefficient polling services. Events from Azure Policy (Event Source) are pushed through Microsoft Azure Event Grid to Event Handlers.



Azure Event Grid provides reliable delivery services to your applications through rich retry policies and dead-letter delivery. Event Grid takes care of the proper routing, filtering, and multicasting of the events to destinations through Event Grid subscriptions. For more information, see [Azure Event Grid](#).

An Event Handler is the place where the event is sent. Several services, such as Microsoft Azure Functions, Microsoft Azure Logic Apps, or your own custom HTTP listener, can be configured to handle events. You can also use any webhook for handling events.

For more information, see [Event handlers in Azure Event Grid](#), [Reacting to Azure Policy state change events](#), and the [Route policy state change events to Event Grid with Azure CLI](#) tutorial.

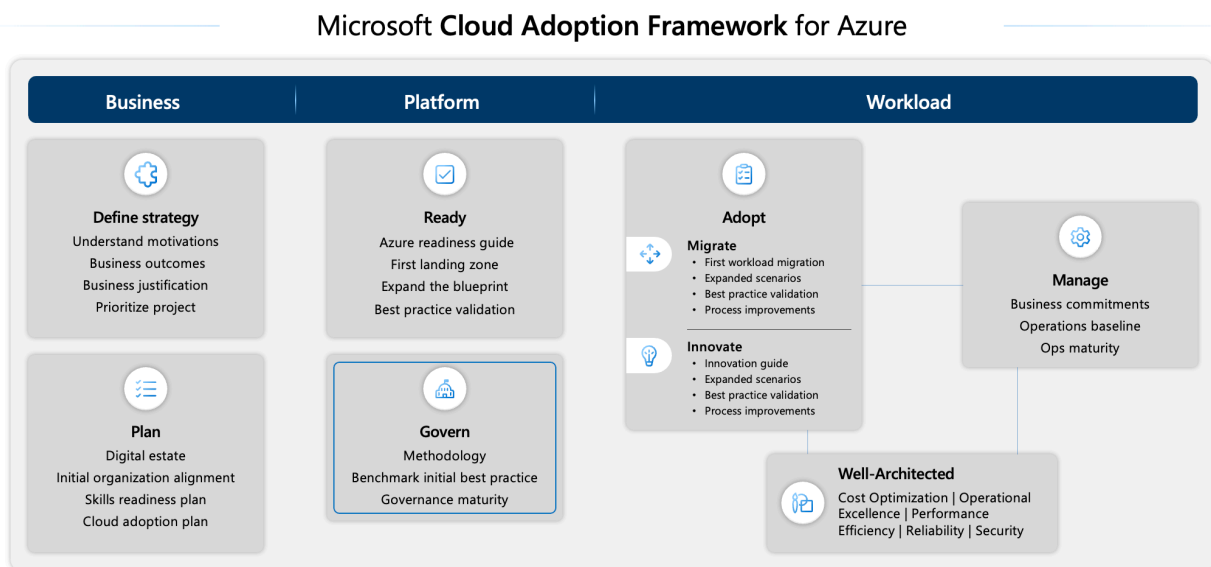
4. Cloud Adoption Framework for Azure

Cloud Adoption Framework for Azure

Completed

The Microsoft Cloud Adoption Framework for Azure offers comprehensive technical guidance for Microsoft Azure. This end-to-end framework helps cloud architects, IT experts, and business leaders reach their cloud adoption objectives. Cloud Adoption Framework includes best practices, documentation, and tools that Microsoft employees, partners, and customers contribute to formulate and implement effective business and technology strategies for the cloud. For more information, see [Microsoft Cloud Adoption Framework for Azure documentation - Cloud Adoption Framework | Microsoft Learn](#).

The following diagram provides high-level information about different methodologies that are included in Microsoft Cloud Adoption Framework for Azure for each phase of your cloud adoption life cycle. Within the framework, Microsoft Azure Policy plays a significant role in the governance framework to help govern your cloud environment and workloads.



Cloud governance refers to the management of cloud usage in your organization. The Cloud Adoption Framework - Govern methodology offers a systematic framework for setting up and improving cloud governance in Azure. This guidance applies to organizations across various industries and it addresses crucial areas, such as regulatory compliance, security, operations, cost management, data, resource management, and AI. It's essential for defining and maintaining efficient cloud use.

Comprehensive cloud governance oversees all aspects of cloud use, minimizes various risks (such as compliance, security, resource management, and data-related concerns), and optimizes cloud

operations throughout the organization. It ensures that cloud activities are consistent with the overall cloud strategy, facilitating the achievement of business objectives with reduced obstacles.

Steps for cloud governance

Cloud governance is a continuous process. It requires ongoing monitoring, evaluation, and adjustments to adapt to evolving technologies, risks, and compliance requirements. The Cloud Adoption Framework - Govern methodology divides cloud governance into five steps.



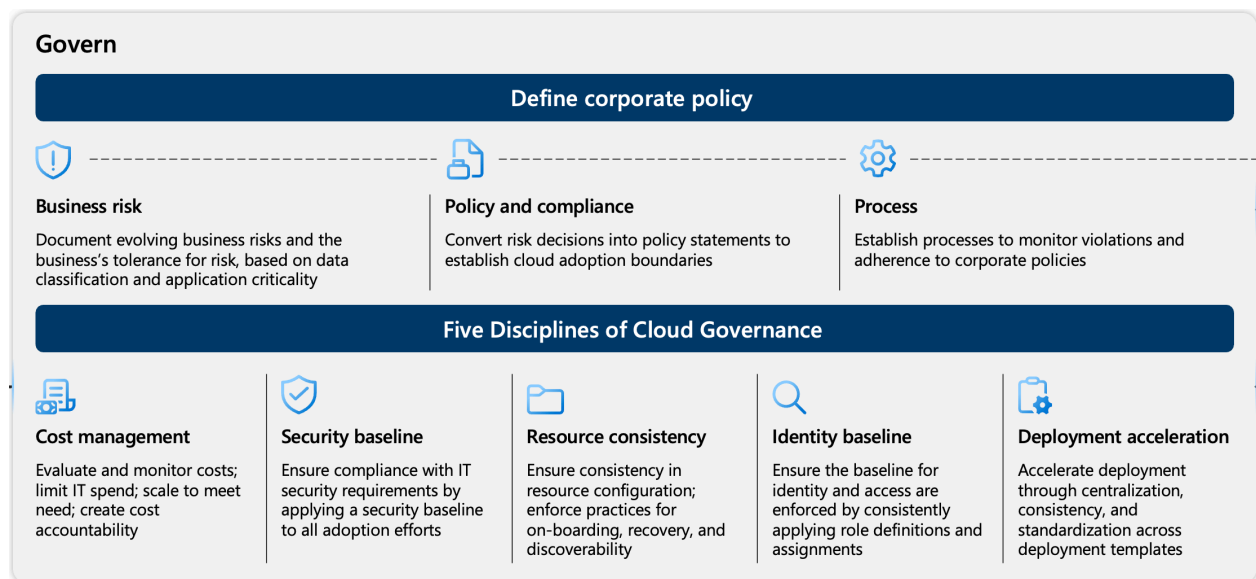
1. **Build a governance team** - Establish a dedicated cloud governance team that's responsible for defining, maintaining, and reporting on the progress of cloud governance policies.
2. **Assess cloud risks** - Conduct a thorough risk assessment that's unique to your organization, addressing all risk categories, including regulatory compliance, security, operations, costs, data management, resource management, and AI-related risks.
3. **Document cloud governance policies** - Clearly document cloud governance policies that dictate acceptable cloud usage and outline the rules and guidelines that mitigate identified risks.
4. **Enforce cloud governance policies** - Implement a systematic approach to ensure compliance with cloud governance policies. Use automated tools alongside manual oversight to enforce compliance. These tools help set guardrails, monitor configurations, and ensure adherence to policies.
5. **Monitor cloud governance** - Regularly monitor cloud usage and the governance teams to ensure ongoing compliance with the established cloud governance policies.

You should complete all five steps to establish cloud governance and regularly iterate on steps 2-5 to maintain cloud governance over time.

Considerations for defining a cloud governance policy

The key considerations when defining a corporate cloud governance policy are as follows:

- **Business risk** – You must document the evolving business risks and the business's tolerance for risk based on data classification and application criticality.
- **Policy and compliance** – You must convert risk decisions into policy statements to establish cloud adoption boundaries efficiently.
- **Process** – You must establish processes to monitor violations and adherence to corporate policies.



The five core disciplines of cloud governance are as follows:

- **Cost management** – Evaluates and monitors costs, including controlling IT expenditures to establish well-defined cost management. It also includes adjusting resources according to demand. It's crucial to exercise control over cloud expenditure to derive greater value from your investments.
- **Security baseline** – Ensures compliance with IT security requirements by applying a security baseline to all adoption efforts.
- **Resource consistency** – Ensures consistency in resource configuration and enforcing practices for onboarding, recovery, and discoverability.
- **Identity baseline** – Ensures that the baseline for identity and access is enforced by consistently applying role definitions and assignments.
- **Deployment acceleration** – Accelerates the deployment of policies through centralization, consistency, and standardization across deployment templates.

Adopting these measures simplifies the compliance process and also enhances the security and efficiency of your cloud environment.

Cloud governance with Azure Policy

Azure's primary governance tool is [Azure Policy](#). Azure Policy facilitates the governance of all resources, including current and forthcoming resources. It helps to enforce organizational standards and to assess compliance at scale by establishing guardrails across various resources.

Azure Policy allows for centralized management, allowing you to track compliance status and investigate changes leading to noncompliance. You can consolidate all compliance data into a singular repository, which simplifies auditing processes for effective cloud compliance and resource governance. The compliance dashboard offered by Azure Policy presents an aggregated view of the environment's overall state, with the ability to examine details at each resource and each policy level.

Azure Policy ensures consistent adherence to compliance standards and prevents misconfigurations. It also helps bring your resources to compliance through bulk remediation for existing resources and automatic remediation for new resources. Additionally, embedding Azure Policy directly into the Azure platform can significantly reduce the need for external approval processes, thus boosting developer productivity.

Some useful governance actions that you can enforce with Azure Policy include:

- Ensure that your team deploys Azure resources only to allowed regions.
- Enforce geo-replication rules to comply with your data residency requirements.
- Allow only certain virtual machine sizes for your cloud environment.
- Enforce the consistent application of taxonomic tags across resources.
- Recommend system updates on your servers.
- Allow multifactor authentication for all subscription accounts.
- Require resources to send diagnostic logs to an Azure Monitor Logs workspace.

Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies that you create. Azure Policy can also prevent noncompliant resources from being created. Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that only allows a certain size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment, including VMs that were created before the policy was created.

In some cases, Azure Policy can automatically remediate noncompliant resources and configurations to ensure the integrity of the state of the resources. For example, if all resources in a specific resource group need to have the *AppName* tag with a value of *SpecialOrders*, Azure Policy can automatically apply that tag if it's missing. However, you maintain full control over your environment. If there's a particular resource that you don't want Azure Policy to automatically update, you can mark that resource as an exception, and the policy won't automatically update it.

Azure Policy also integrates with Azure DevOps by applying any continuous integration and delivery pipeline policies that pertain to the predeployment and post-deployment phases of your applications.

The objective when designing an Azure Policy should be to strike a balance between control and stability in one area with speed and results in the other. The main reason to maintain balance is to ensure that the environment remains highly manageable so that you can implement necessary levels of control to ensure governance without hindering operational efficiency or productivity. In establishing and enforcing these controls, you must take care that your speed to achieve efficiency isn't affected. Hence, achieving this balance between control and stability with speed and results, often requires thoughtful compromise, and it's necessary to carefully evaluate the potential impact before introducing new policies.

For more information, see [Microsoft Cloud Adoption Framework for Azure - Cloud Adoption Framework | Microsoft Learn](#).

5. Check your knowledge

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/check>

Check your knowledge

Completed

6. Introduction

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/overview>

Introduction

Completed

Azure Policy is a service that allows you to create, assign, and manage governance policies that enforce rules and effects over Azure resources to ensure that they stay compliant with your IT governance standards. These policies enforce various rules and effects on resources, ensuring that they adhere to corporate standards and service-level agreements. These policies are described in JSON format and are known as policy definitions. Azure Policy is crucial for enforcing organizational standards and assessing compliance on a large scale.

Azure Policy initiatives are a collection of Azure Policy definitions that are grouped together toward a specific goal or purpose. By consolidating multiple Azure policies into a single item, Azure Policy initiatives allow centralized control and enforcement of configurations across Azure resources.

Industry-specific organizations in sectors like government, public sector, finance, and others accelerate digital transformation and achieve better business outcomes. They can achieve this objective by adopting specific, sovereignty-focused Azure Policy initiatives to address the complexity of compliance with national and regional regulatory requirements.

Customers can create Azure Policy initiatives that aid in customizing deployments to reduce the time needed to audit environments and help meet established regulatory compliance frameworks and government requirements. The initiatives help public sector partners and customers create cloud guardrails and enforce specific regulations effectively. They can layer policy initiatives to help form a complete solution for their specific needs, and they can use deployment automation to ensure consistency, best practices, and save time.

In this module, you learn how Azure Policy can help do the common tasks related to creating, assigning, and managing policies across your organization, such as:

- Assign a policy to enforce a condition for resources you create in the future
- Create and assign an initiative definition to track compliance for multiple resources
- Resolve a noncompliant or denied resource
- Implement a new policy across an organization

Important

Organizations are wholly responsible for ensuring their own compliance with all applicable laws and regulations. The information provided in this document does not constitute legal advice, and organizations should consult their legal advisors for any questions regarding regulatory compliance.

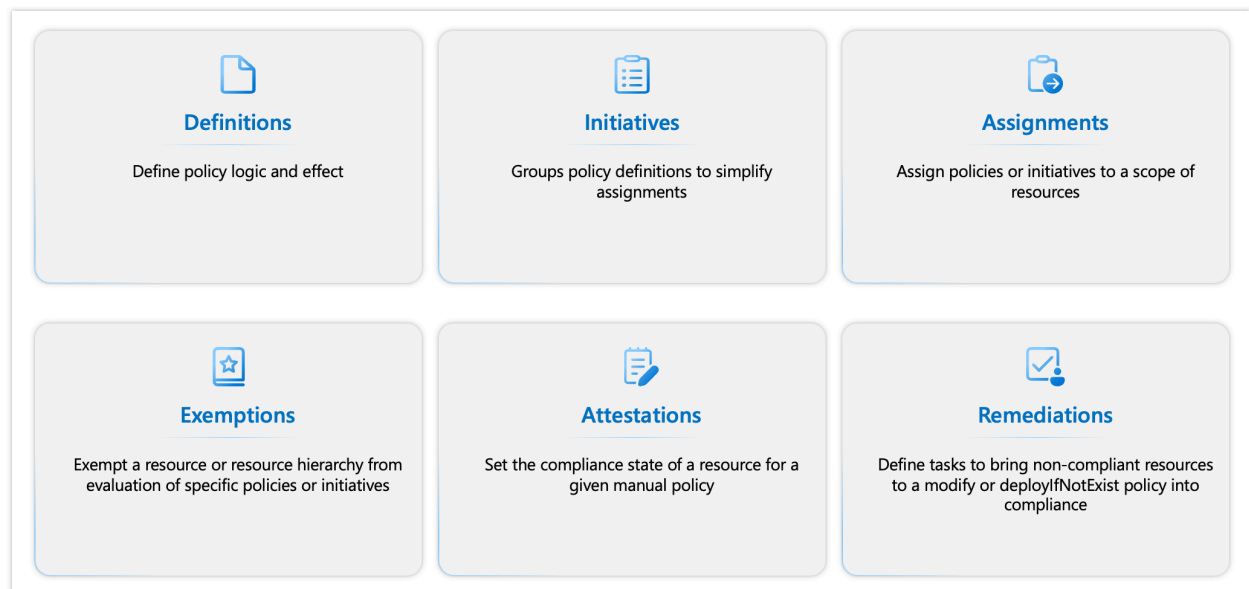
7. Azure Policy resources

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/azure-policy-resources>

Azure Policy resources

Completed

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. You explore these settings in the next unit, **Azure Policy definitions**. The primary concept to which these settings can be applied is scope.

Scope in Azure Policy is the same as the levels of hierarchy for governance in Azure. Four levels of management scope are under the root tenant: the management groups, subscriptions, resource groups, and resources. The definition might be saved in a management group or a subscription. The definition location determines the scope to which the initiative or policy can be assigned. An assignment also has several properties that set a scope. The use of these properties determines which resource for Azure Policy to evaluate and which resources count toward compliance

You can apply management settings at any of these levels of scope. The level that you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For more information, see [Scope in 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. The initiative definition contains all policy definitions to help track your compliance state for a larger goal, such as organizational compliance goals or compliance with regulatory frameworks. For example, multiple tagging policy definitions can be grouped into a single initiative, and rather than assigning each policy individually, you can apply the initiative to Azure resources. A JSON can be used to create a policy initiative definition. For more information, see [Azure Policy initiative definition structure](#).

For more information about Azure Policy built-ins and patterns, see [Azure Policy samples](#).

Built-in policy is a type of policy definition that's generated by Azure Resource Providers and is available by default. A group of such policy definitions is known as a **built-in initiative**. Azure built-in policy initiatives are a powerful tool set that enables centralized control across Azure resources and enforcement of specific configurations. These initiatives comprise a collection of policy definitions and support compliance with various regulatory frameworks, industry standards, and security best practices.

For a list of built-in policies and initiatives, see [Policies](#) and [Initiatives](#).

Custom policy is a type of policy definition that's written by a policy user when no built-in policy maps to your requirements. A group of such policy definitions is known as a **custom initiative**. Azure Policy custom initiatives help you to tailor a set of policies specifically to your organization's unique requirements, giving you control to enforce the standards and rules that best fit your environment. Microsoft for Sovereignty makes several custom policy initiatives and compliance mappings accessible through the [industry-policy-portfolio](#) repository on GitHub.

Microsoft for Sovereignty initiatives and compliance mappings, which expand on the Azure built-in initiatives, help you automate policy enforcement and foster a robust governance framework that reduces the risk of noncompliance. Further, the initiatives also strengthen data protection measures. Organizations can use the large suite of available regulatory compliance built-in initiatives while we continue to expand on other frameworks. These initiatives are available as Azure built-in and custom policy initiatives.

For more information, see [Microsoft for Sovereignty policy portfolio](#).

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.

Policy and initiative definitions are deployed to a definition location (management group or subscription). This location determines the scope to which the initiative or policy can be assigned. The location should be the resource container shared by all resources that you want to use the policy definition on. For more information, see [Definition location](#).

Policies and initiatives are assigned to a specific scope (management group, subscription, or resource group). While doing so, you can define several optional aspects, including the resource scope and policy definition.

- Optional *resource selectors* to allow gradual rollout based on resource location or type.
- Optional *overrides* to change the effect of a policy definition without modifying the underlying definition.
- *enforcementMode* can be disabled to support "what-if" scenarios without changing the definition, which is equivalent to changing the definition to an audit effect mode, but a way to

do it at assignment level. For example, if the policy has *Deny* effects, that denial isn't effective, but you can still view the result of the compliance evaluation of that policy.

- Optional *excluded scopes* to exclude inner containers or resources from the assignment scope.
- *Noncompliance messages* can be defined.
- *Parameters* can be assigned values.
- If you have a policy with the *deployIfNotExists* effect type, a *managed identity* can be assigned (system-assigned or user-assigned) to turn on remediation actions. An assignment has several properties that set a scope. The use of these properties determines which resource for Azure Policy to evaluate and which resources count toward compliance. These properties map to the following concepts:
 - **Inclusion** - For more information, see [Azure Policy assignment structure](#).
 - **Exclusion** - For more information, see [Azure Policy assignment structure excluded scopes](#).

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. Two exemption categories exist and are used to group exemptions:

- **Mitigated** - The exemption is granted because the policy intent is met through another method.
- **Waiver** - The exemption is granted because the noncompliance state of the resource is temporarily accepted.

For more information about policy exemption, see [Azure Policy exemption structure](#).

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.

For more information, see [Azure Policy attestation structure](#).

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.

For more information, see [Azure Policy remediation task structure](#).

8. Azure Policy definitions

<https://learn.microsoft.com/en-us/training/modules/sovereignty-policy-initiatives/azure-policy-definitions>

Azure Policy definitions

Completed

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.

Anatomy of a policy definition

You use JSON to create a policy definition that contains the elements shown in the following table.

Element	Description	Properties or values
<i>displayName</i> (string, max 128 characters)	Used to identify the policy definition.	
<i>description</i> (string, max 512 characters)	Provides context for when the definition is used.	
<i>policyType</i> (read-only string)	Indicates the origin of the policy definition. This property can't be set, but SDK returns three values which are visible in the portal.	• Built in: Provided and maintained by Microsoft. • Custom: Custom definitions created by the customer. • Static: Regulatory Compliance policy with Microsoft ownership.

Element	Description	Properties or values
<i>mode (string)</i>	Configured depending on the target of the policy: an Azure Resource Manager property or a Resource Provider property.	- Resource Manager Modes: - On All: Evaluates resource groups, subscriptions, and all resource types. - Indexed: Evaluates resource groups, subscriptions, and all resource types. - Resource Provider Modes (limited to Built in policies and fully supported): - Microsoft.Kubernetes.Data - Microsoft.KeyVault.Data - Microsoft.Network.Data - Resource Provider Modes (limited to Built in policies and in preview mode): - Microsoft.ManagedHSM.Data - Microsoft.DataFactory.Data
<i>version (string, optional)</i>	Built-in policy definitions can host multiple versions with the same definitionID. If no version number is specified, all experiences show the latest version of the definition.	
<i>metadata (object, optional, max 1,024 characters)</i>	Stores information about the policy definition.	Common properties (for Built-in policies): <i>version (string)</i>: Tracks details about the version of the contents of a policy definition. <i>category (string)</i>: Determines under which category in the Azure portal the policy definition is displayed. <i>preview (Boolean)</i>: True or false flag that indicates if the policy definition is in preview. <i>deprecated (Boolean)</i>: True or false flag that indicates if the policy definition is deprecated. <i>portalReview (string)</i>: Determines if parameters require review in the portal.
<i>parameters (object, optional)</i>	Help simplify your policy management by reducing the number of policy definitions. By including parameters in a policy definition, you can reuse that policy for different scenarios by using different values.	Properties: - name - type (String, Array, Object, Boolean, Integer, Float, DateTime) - metadata (description, displayName, strongType, assignPermissions) - defaultValue - allowedValues - schema

Element	Description	Properties or values
<i>policyRule</i> (object)	The effect of a policy is defined in the <i>policyRule</i>. The policy rule consists of the <i>if</i> and <i>then</i> blocks. • In the <i>if</i> block, you define one or more conditions that specify when the policy applies. • In the <i>then</i> block, you define the effect that happens when the <i>if</i> conditions result true.	

```
{
  "displayName": "Allowed locations",
  "description": "This policy enables you to restrict the locations your organization can use to deploy resources.",
  "policyType": "BuiltIn",
  "mode": "Indexed",
  "metadata": {
    "version": "1.0.0",
    "category": "General"
  },
  "parameters": {
    "listOfAllowedLocations": {
      "type": "Array",
      "metadata": {
        "description": "The list of locations that can be specified when deploying resources.",
        "strongType": "location",
        "displayName": "Allowed locations"
      }
    }
  },
  "policyRule": {
    "if": {
      "allOf": [
        {
          "field": "location",
          "notIn": "[parameters('listOfAllowedLocations')]"
        },
        {
          "field": "location",
          "notEquals": "global"
        },
        {
          "field": "type",
          "notEquals": "Microsoft.AzureActiveDirectory/b2cDirectories"
        }
      ]
    },
    "then": {
      "effect": "deny"
    }
  }
}
```

```
}
```

In the given example, *b2cDirectories* is excluded from the policy logic because its location field isn't a region (it can be "United States," "Europe," "Asia Pacific," or "Australia"). This logic can be enforced with a separate policy.

Logical operators and conditions (*if* blocks)

The first part of the *policyRule* in an Azure Policy definition is the *if* block. This block defines the conditions for which the policy evaluates the resources. A policy definition can contain several conditional statements. Depending on your evaluation requirements, you might or might not need each statement to be true, and you might only need some of them to be true.

Logical operators supported in the *if* block

In the *if* condition, you can put different logical operators.

Operator	Type	Description
<i>not</i>	{condition or operator}	The <i>not</i> syntax inverts the result of the condition.
<i>allOf</i>	[[{condition or operator}, {condition or operator}]	The <i>allOf</i> syntax (like the logical <i>and</i> operation) requires all conditions to be true.
<i>anyOf</i>	[[{condition or operator}, {condition or operator}]	The <i>anyOf</i> syntax (like the logical <i>or</i> operation) requires one or more conditions to be true.

The following example shows use of the *allOf* logical operator:

```
{
  "if": {
    "allOf": [
      {
        "field": "location",
        "notIn": "[parameters('listOfAllowedLocations')]"
      },
      {
        "field": "location",
        "notEquals": "global"
      },
      {
        "field": "type",
        "notEquals": "Microsoft.AzureActiveDirectory/b2cDirectories"
      }
    ]
  },
  "then": {
    "effect": "deny"
  }
}
```

```
}  
}
```

Nested logical operations

Logical operations are optional and can be nested to create complex scenarios.

The following example shows a *not* operation nested in an *allOf* operation:

```
"if": {  
  "allOf": [  
    {  
      "not": {  
        "field": "tags",  
        "containsKey": "application"  
      }  
    },  
    {  
      "field": "type",  
      "equals": "Microsoft.Storage/storageAccounts"  
    }  
  ]  
},
```

Conditions

Properties like fields, values, or counts can be evaluated within a condition.

Fields	Conditions that evaluate whether the values of properties in the resource request payload meet certain criteria can be formed by using a field expression.	Name, fullName, kind, type, location, ID, identity.type, tags, tags['tagName'], property aliases
Value	Conditions that evaluate whether a value meets certain criteria can be formed by using a value expression.	
Count	Conditions that count how many members of an array meet certain criteria can be formed by using a count expression.	- Field count, value count - The current () function returns the value of the array member that's being evaluated

The condition in an Azure Policy assesses whether the evaluated values for the properties, such as Fields, Value, or Count, meets certain criteria. If the result of a function is an error, the policy results in a deny effect. This result can be avoided while testing by disabling *enforcementMode* in the assignment. For more information, see [Enforcement Mode](#).

Evaluation criteria	Value type
<i>equals</i>	stringValue
<i>notEquals</i>	stringValue

Evaluation criteria	Value type
<i>like</i>	stringValue
<i>notLike</i>	stringValue
<i>match</i>	stringValue
<i>notMatch</i>	stringValue
<i>matchInsensitively</i>	stringValue
<i>notMatchInsensitively</i>	stringValue
<i>contains</i>	stringValue
<i>notContains</i>	stringValue
<i>In</i>	["stringValue1", "stringValue2"]
<i>notIn</i>	["stringValue1", "stringValue2"]
<i>containsKey</i>	keyName
<i>notContainsKey</i>	keyName
<i>less</i>	dateValue
<i>less</i>	stringValue
<i>less</i>	intValue
<i>lessOrEquals</i>	dateValue
<i>lessOrEquals</i>	stringValue
<i>lessOrEquals</i>	intValue
<i>greater</i>	dateValue
<i>greater</i>	stringValue
<i>greater</i>	intValue
<i>greaterOrEquals</i>	dateValue
<i>greaterOrEquals</i>	stringValue
<i>greaterOrEquals</i>	intValue
<i>exists</i>	bool

The following JSON is an example of using conditions:

```
{
  "if": {
    "allOf": [
      {
        "value": "[resourceGroup().name]",
        "like": "*netrq"
```

```

    },
    {
      "field": "type",
      "notLike": "Network/*"
    }
  ]
},
"then": {
  "effect": "deny",
  "details": {
    "count": {
      "field": "Microsoft.Network/virtualNetworks/addressSpace.addressPrefixes[*]",
      "where": {
        "value": "[ipRangeContains('10.0.0.0/24', current('Microsoft.Network/virt
        "equals": "greater"
      }
    }
  }
}
}
}
}

```

Policy functions

Functions can be used to introduce extra logic into a policy rule. They're resolved in the policy rule of a policy definition and in the parameter values that are assigned to the policy definitions in an initiative.

The Resource Manager template functions are available to use in a policy rule except a [few policy functions and user-defined functions](#).

The `utcNow()` function is available to use in a policy rule but differs from use in an Azure Resource Manager template (ARM template). Unlike an ARM template, this function can be used outside `defaultValue`. It returns a string set to the current date and time in Universal ISO 8601 DateTime format `yyyy-MM-ddTHH:mm:ss.fffffffZ`.

The following table describes the functions that are only available in policy rules.

Function	Description
<code>addDays(dateTime, numberOfDaysToAdd)</code>	<code>dateTime</code> : [Required] string - String in the Universal ISO 8601 DateTime format 'yyyy-MM-ddTHH:mm:ss.FFFFFFFZ'. <code>numberOfDaysToAdd</code> : [Required] integer - Number of days to add.
<code>Field(fieldName)</code>	<code>fieldName</code> : [Required] string - Name of the field to retrieve. - Returns the value of that field from the resource evaluated by the <i>If</i> condition. <code>field</code> is primarily used with <code>auditIfNotExists</code> and <code>deployIfNotExists</code> to reference fields on the resource that are being evaluated.

Function	Description
<code>requestContext().apiVersion</code>	Returns the API version of the request that triggered policy evaluation. This value is the API version that was used in the PUT/PATCH request for evaluations on resource creation/update. The latest API version is always used during compliance evaluation on existing resources.
<code>policy()</code>	Returns the following information about the policy that's being evaluated. Properties can be accessed from the returned object. <pre>"assignmentId": "", "definitionId": "", "setDefinitionId": "", "definitionReferenceId": ""</pre>
<code>ipRangeContains(range, targetRange)</code>	<code>range</code> : [Required] string - String specifying a range of IP addresses to check if the <i>targetRange</i> is within range. <code>targetRange</code> : [Required] string - String specifying a range of IP addresses to validate as included within the <i>range</i>. Returns a <i>boolean</i> for whether the <i>range</i> IP address range contains the <i>targetRange</i> IP address range. Empty ranges or mixing between IP families isn't allowed and results in evaluation failure.
<code>current(indexName)</code>	Special function that can only be used inside count expressions.

Effect types (*then* blocks)

The second part of the *policyRule* in an Azure Policy definition is the *then* block. This block defines the effect that takes place when the policy rule is evaluated to match the condition resources. More than one effect can be valid for a given policy definition. Parameters are often used to specify allowed effect values (*allowedValues*) in such cases so that a single definition can be more versatile during assignment. Resource properties and logic in the policy rule can determine whether a certain effect is considered valid to the policy definition.

Effect	Description	Type
<i>disabled</i>	The <i>disabled</i> effect is a way to deactivate the policy. If a policy definition has <i>Disabled</i> as its effect, any assignments of that policy aren't active. This effect is checked first to determine if the policy rule should be evaluated. This flexibility makes it possible to deactivate a single assignment instead of deactivating all of that policy's assignments.	Synchronous evaluation
<i>append</i>	The <i>append</i> effect is used to add more fields to the requested resource during creation or update. It's mostly obsolete because <i>Modify</i> can also be used to add fields to the request.	Synchronous evaluation
<i>modify</i>	The <i>modify</i> effect is used to add, update, or remove properties or tags on a subscription or resource during creation or update. It allows	Synchronous evaluation

Effect	Description	Type
	Azure Policy to modify requests to Azure Resource Manager by altering fields to ensure compliance.	
<i>deny</i>	The <i>deny</i> effect is used to prevent a resource request that doesn't match defined standards through a policy definition and fails the request.	Synchronous evaluation
<i>denyAction</i>	The <i>denyAction</i> effect is used to block requests based on intended action to resources at scale. Currently, the only supported action is DELETE.	Synchronous evaluation
<i>audit</i>	The <i>audit</i> effect is used to create a warning event in the activity log when you're evaluating a noncompliant resource, but it doesn't stop the request.	Asynchronous evaluation
<i>auditIfNotExists</i>	The <i>auditIfNotExists</i> effect allows the auditing of resources that are related to the resource that matches the <i>if</i> condition but doesn't have the properties specified in the details of the <i>then</i> condition.	Asynchronous evaluation
<i>deployIfNotExists</i>	The <i>deployIfNotExists</i> policy definition runs a template deployment when the condition is met. It can trigger deployment of a related resource based on the compliance state of the currently evaluated resource.	Asynchronous evaluation
<i>manual</i>	The <i>manual</i> effect allows you to self-attest the compliance of resources or scopes. When a policy definition with <i>Manual</i> effect is assigned, you can set the compliance states of targeted resources or scopes through custom attestations.	Manual attestation

The following list provides general guidance around interchangeable effects:

- *audit*, *deny*, and either *modify* or *append* are often interchangeable.
- *auditIfNotExists* and *deployIfNotExists* are often interchangeable.
- *manual* isn't interchangeable.
- *disabled* is interchangeable with any effect.

Multiple policies can be assigned to a single resource at the same scope or at different scopes. Each policy mostly has a different effect defined. The condition and effect for each policy is independently evaluated. The net result of layering policy definitions is considered **cumulative most restrictive**.

Secure your Azure resources with Azure role-based access control (Azure RBAC)

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/1-introduction>

Introduction

Completed

For any organization using the cloud, securing your Azure resources such as virtual machines, websites, networks, and storage is a critical function. You want to ensure that your data and assets are protected, but still grant your employees and partners the access they need to perform their jobs. Azure RBAC is an authorization system in Azure that helps you manage who has access to Azure resources, what they can do with those resources, and where they have access.

Suppose you work for First Up Consultants, which is an engineering firm that specializes in circuit and electrical design. They've moved their workloads and assets to Azure to make collaboration easier across several offices and other companies. You work in the IT department at First Up Consultants. You're responsible for keeping the company's assets secure, but still allowing users to access the resources they need. You've heard that Azure RBAC can help you manage resources in Azure.

In this module, you'll learn how to use Azure RBAC to manage access to resources in Azure.

Learning objectives

In this module, you'll:

- Verify access to resources for yourself and others.
- Grant access to resources.
- View activity logs of Azure RBAC changes.

2. What is Azure RBAC?

What is Azure RBAC?

Completed

When it comes to identity and access, most organizations that are considering using the public cloud are concerned about two things:

1. Ensuring that when people leave the organization, they lose access to resources in the cloud.
2. Striking the right balance between autonomy and central governance. For example, giving project teams the ability to create and manage virtual machines in the cloud, while centrally controlling the networks those VMs use to communicate with other resources.

Microsoft Entra ID and Azure RBAC work together to make it simple to carry out these goals.

Azure subscriptions

First, remember that each 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.

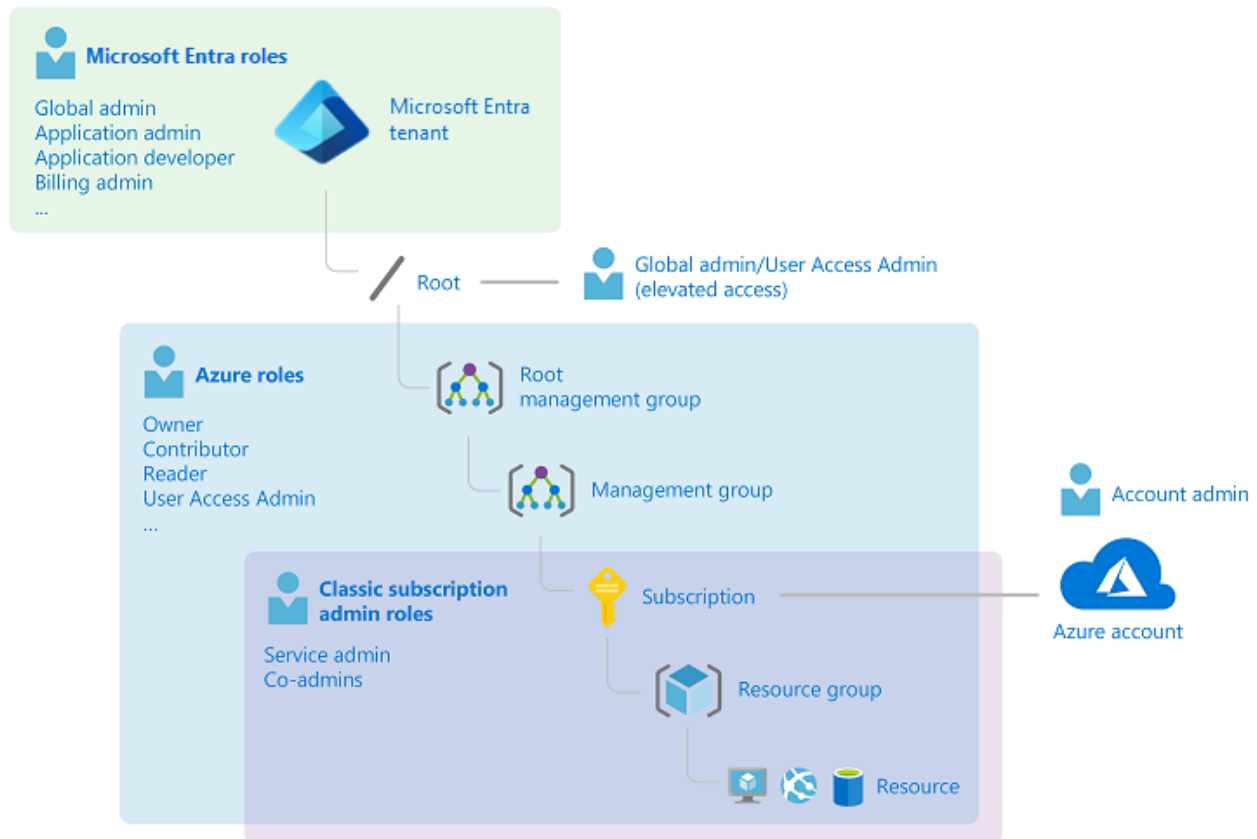
What's Azure RBAC?

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.

The following video describes Azure RBAC in detail:

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. For example, a user with access to a resource group can manage all the resources it contains like websites, virtual machines, and subnets. The Azure role that you assign dictates what resources the user, group, or application can manage within that scope.

The following diagram depicts how the classic subscription administrator roles, Azure roles, and Microsoft Entra roles are related at a high level. Child scopes, such as service instances, inherit roles assigned at a higher scope, like an entire subscription.



In the preceding diagram, a subscription is associated with only one Microsoft Entra tenant. Also note that a resource group can have multiple resources, but it's associated with only one subscription. Although it's not obvious from the diagram, a resource can be bound to only one resource group.

What can I do with Azure RBAC?

Azure RBAC allows you to grant access to Azure resources that you control. Suppose you need to manage access to resources in Azure for the development, engineering, and marketing teams. You've started to receive access requests, and you need to quickly learn how access management works for Azure resources.

Here are some scenarios you can implement with Azure RBAC:

- Allow one user to manage virtual machines in a subscription and another user to manage virtual networks.
- Allow a database administrator group to manage SQL databases in a subscription.
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets.
- Allow an application to access all resources in a resource group.

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.

The following shows an example of the Access control (IAM) pane for a resource group. In this example, Alain has been assigned the Backup Operator role for this resource group.

The screenshot shows the Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (alain@contoso.com). The breadcrumb trail indicates the current location: Home > Resource groups > sales-projectforecast. The main pane is titled 'sales-projectforecast | Access control (IAM)'. On the left, a sidebar lists various tools, with 'Access control (IAM)' highlighted. The main content area shows the 'Role assignments' tab. It includes a search bar, filters (Type: All, Role: All, Scope: All scopes, Group by: Role), and a table of role assignments. The table has 25 items (15 Users, 1 Groups, 8 Service Principals, 1 Managed Identities). A red box highlights the first row, which shows the 'Backup Operator' role assigned to 'Alain' (alain@contoso.com) with the scope 'This resource'.

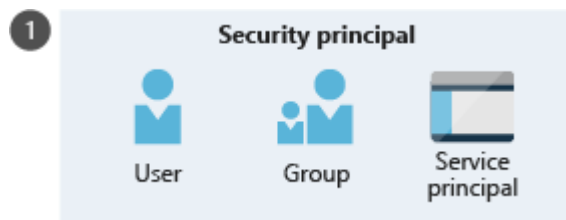
Name	Type	Role	Scope	Condition
Alain alain@contoso.com	User	Backup Operator	This resource	None
App2	App	Billing Reader	Subscription (Inherited)	None
Sales Admins	Group	Billing Reader	Subscription (Inherited)	None
user-assigned-identity	User-assigned Managed Identity	Billing Reader	Subscription (Inherited)	None

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*.

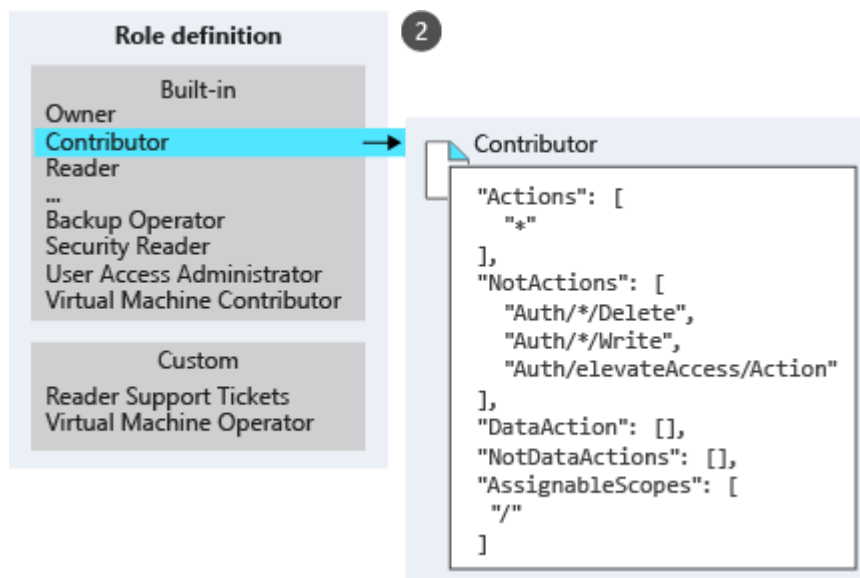
1. Security principal (who)

A *security principal* is just a fancy name for a user, group, or application to which you want to grant access.



2. Role definition (what)

A *role definition* 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.



Azure includes several built-in roles that you can use. The following lists four fundamental built-in roles:

- **Owner:** Has full access to all resources, including the right to delegate access to others.
- **Contributor:** Can create and manage all types of Azure resources, but can't grant access to others.
- **Reader:** Can view existing Azure resources.
- **User Access Administrator:** Lets you manage user access to Azure resources.

If the built-in roles don't meet the specific needs of your organization, you can create your own custom roles.

3. Scope (where)

Scope 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.

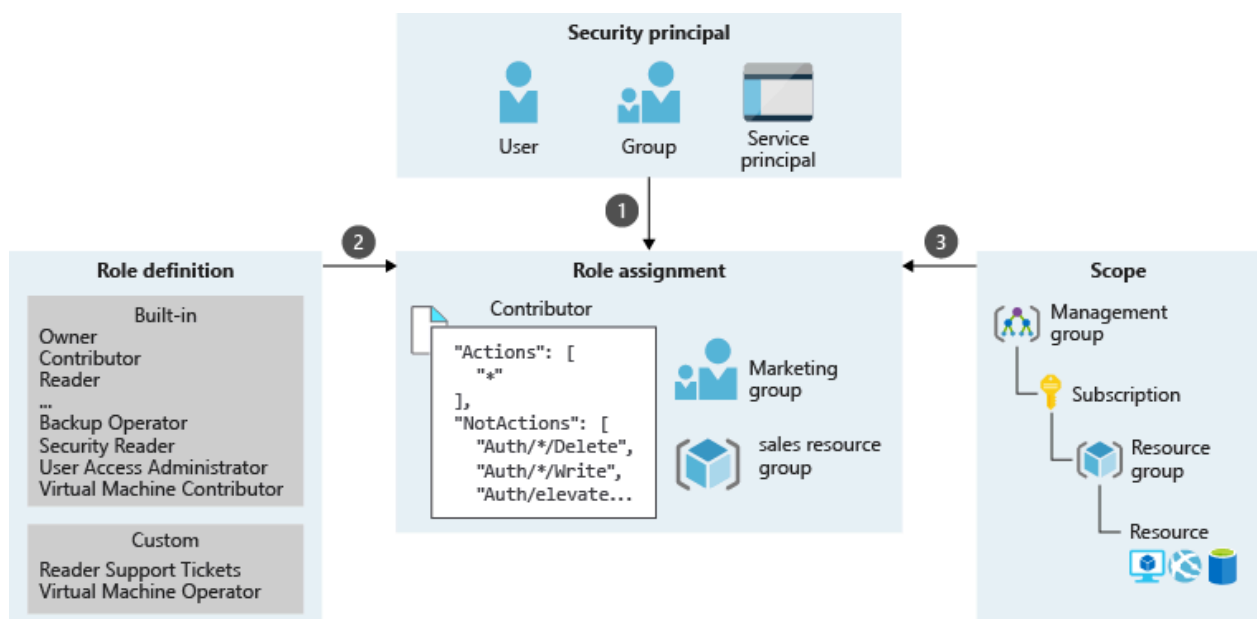
In Azure, you can specify a scope at multiple levels: management group, subscription, resource group, or resource. Scopes are structured in a parent-child relationship. When you grant access at a parent scope, the child scopes automatically inherit those permissions. For example, if a group is assigned the Contributor role at the subscription scope, it will inherit the role for all resource groups and resources within the subscription.



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.

The following example shows how the Marketing group has been assigned the Contributor role at the sales resource group scope.



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. For example, the [Contributor](#) role has both `Actions` and `NotActions`. The wildcard (*) in `Actions` indicates that it can perform all operations on the control plane. You'd then subtract the following operations in `NotActions` to compute the effective permissions:

- Delete roles and role assignments
- Create roles and role assignments
- Grant the caller User Access Administrator access at the tenant scope
- Create or update any blueprint artifacts
- Delete any blueprint artifacts

3. Knowledge check - What is Azure RBAC?

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/3-knowledge-check-rbac-overview>

Knowledge check - What is Azure RBAC?

Completed

4. Exercise - List access using Azure RBAC and the Azure portal

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/4-list-access>

Exercise - List access using Azure RBAC and the Azure portal

Completed

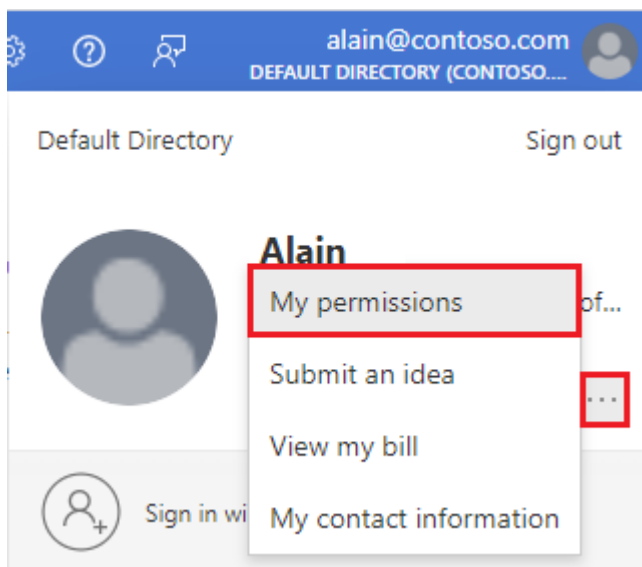
At First Up Consultants, you've been granted access to a resource group for the marketing team. You want to familiarize yourself with the Azure portal and see what roles are currently assigned.

You need an Azure subscription to complete the exercises. If you don't have an Azure subscription, create a [free account](#) and add a subscription before you begin. If you're a student, you can take advantage of the [Azure for students](#) offer.

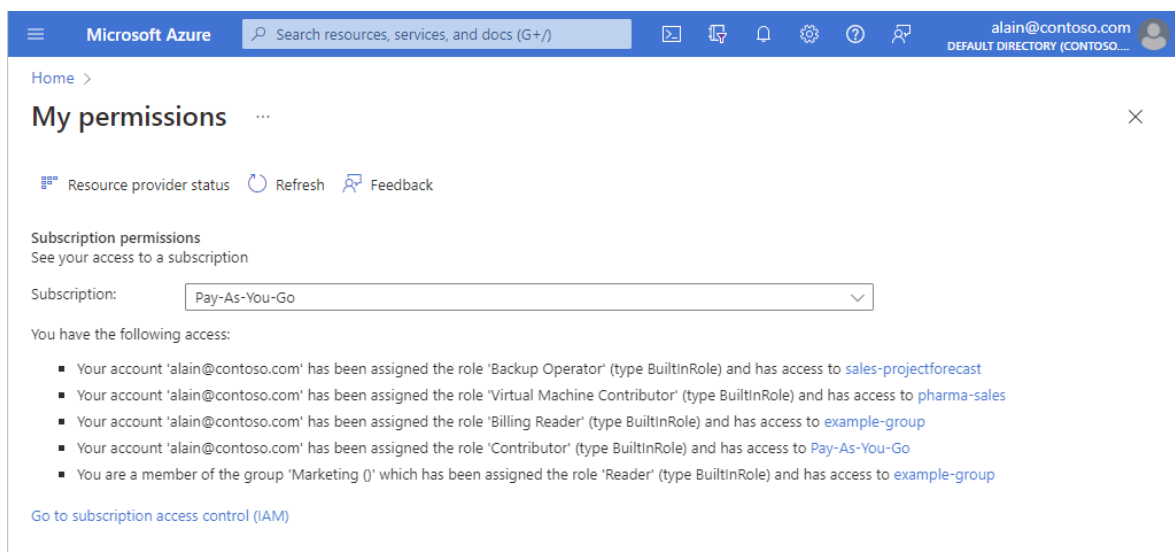
List role assignments for yourself

Follow these steps to see what roles are currently assigned to you.

1. Sign in to the [Azure portal](#).
2. On the **Profile** menu, select the ellipsis (...) to see more links.



3. Select **My permissions** to open the **My permissions** pane.

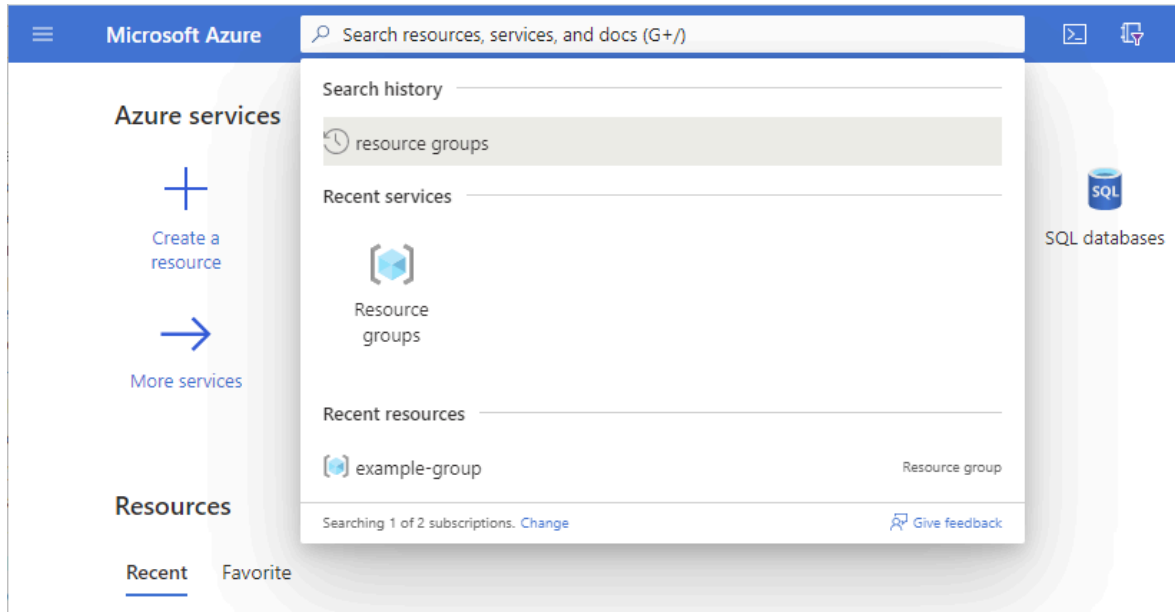


You'll find the roles that you've been assigned and the scope. Your list will look different.

List role assignments for a resource group

Follow these steps to see what roles are assigned at the resource group scope.

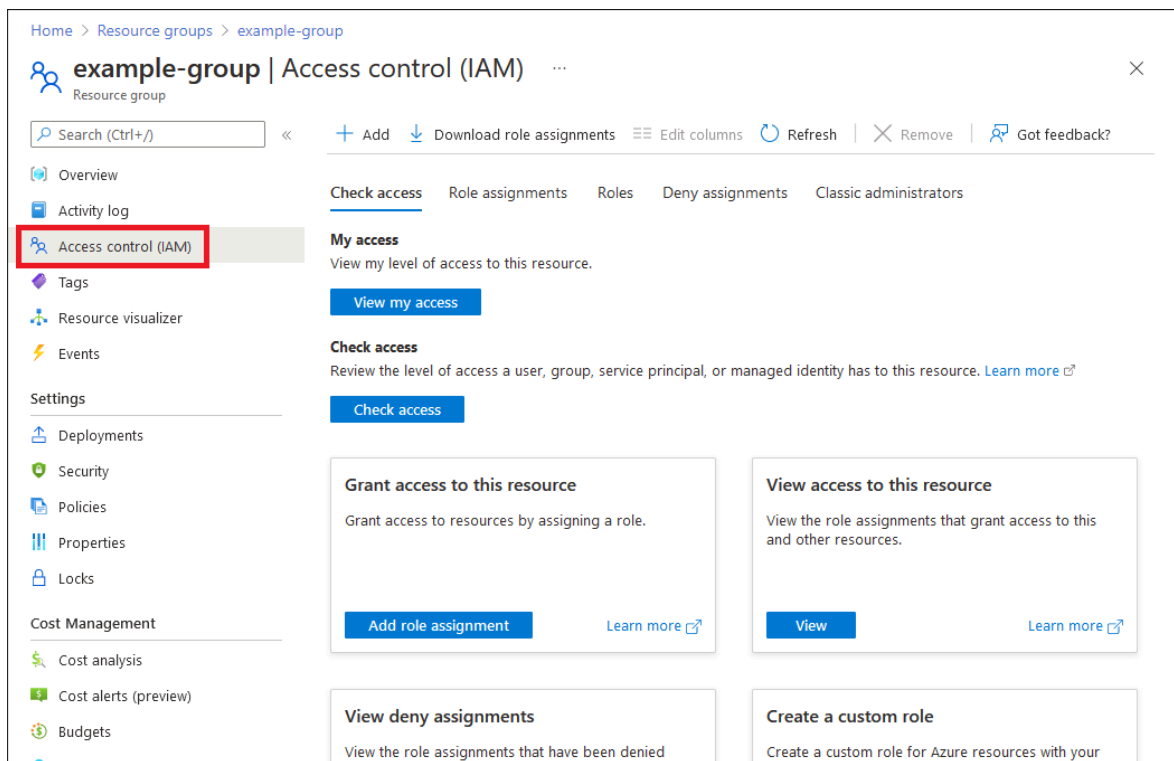
1. In the Search box at the top, search for and select **Resource groups**.



2. In the list of resource groups, select a resource group.

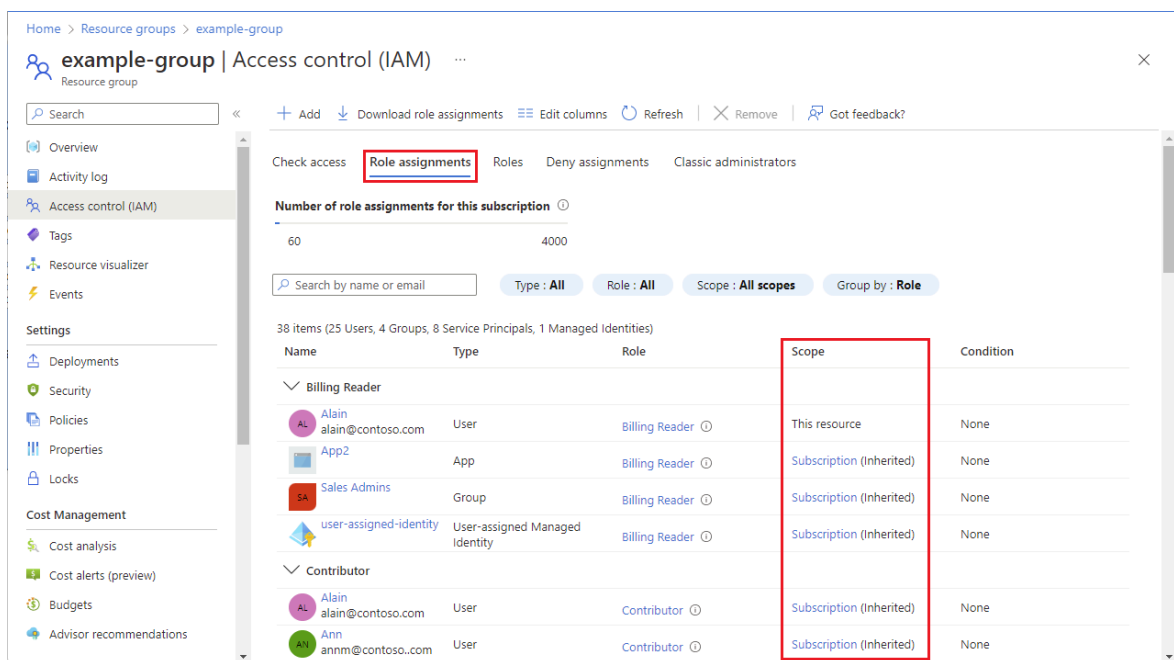
These steps use a resource group named **example-group**, but your resource group's name will be different.

3. On the left menu pane, select **Access control (IAM)**.



4. Select the **Role assignments** tab.

This tab shows who has access to the resource group. Notice that some roles are scoped to **This resource**, while others are **(Inherited)** from a parent scope.



List roles

As you learned in the previous unit, a role is a collection of permissions. Azure has more than 70 built-in roles that you can use in your role assignments. To list the roles:

- In the menu bar at the top of the pane, select the **Roles** tab to list of all the built-in and custom roles.

Select a role's **View** link in the **Details** column, then select the **Assignments** tab to display the number of users and groups assigned to that role.

[+ Add](#)
[Download role assignments](#)
[Edit columns](#)
[Refresh](#)
[Remove](#)
[Got feedback?](#)

[Check access](#)
[Role assignments](#)
[Roles](#)
[Deny assignments](#)
[Classic administrators](#)

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Type: **All**
Category: **All**

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Owner	Grants full access to manage all res...	BuiltInRole	General	View ...
Contributor	Grants full access to manage all res...	BuiltInRole	General	View ...
Reader	View all resources, but does not all...	BuiltInRole	General	View ...
Access Review Operat...	Lets you grant Access Review Syste...	BuiltInRole	None	View ...
AcrDelete	acr delete	BuiltInRole	Containers	View ...
AcrImageSigner	acr image signer	BuiltInRole	Containers	View ...
AcrPull	acr pull	BuiltInRole	Containers	View ...
AcrPush	acr push	BuiltInRole	Containers	View ...
AcrQuarantineReader	acr quarantine data reader	BuiltInRole	Containers	View ...
AcrQuarantineWriter	acr quarantine data writer	BuiltInRole	Containers	View ...
AgFood Platform Sens...	Provides contribute access to man...	BuiltInRole	None	View ...
AgFood Platform Servi...	Provides admin access to AgFood ...	BuiltInRole	AI + Machine Learning	View ...

In this unit, you learned how to list the role assignments for yourself in the Azure portal. You also learned how to list the role assignments for a resource group.

5. Exercise - Grant access using Azure RBAC and the Azure portal

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/5-grant-access>

Exercise - Grant access using Azure RBAC and the Azure portal

Completed

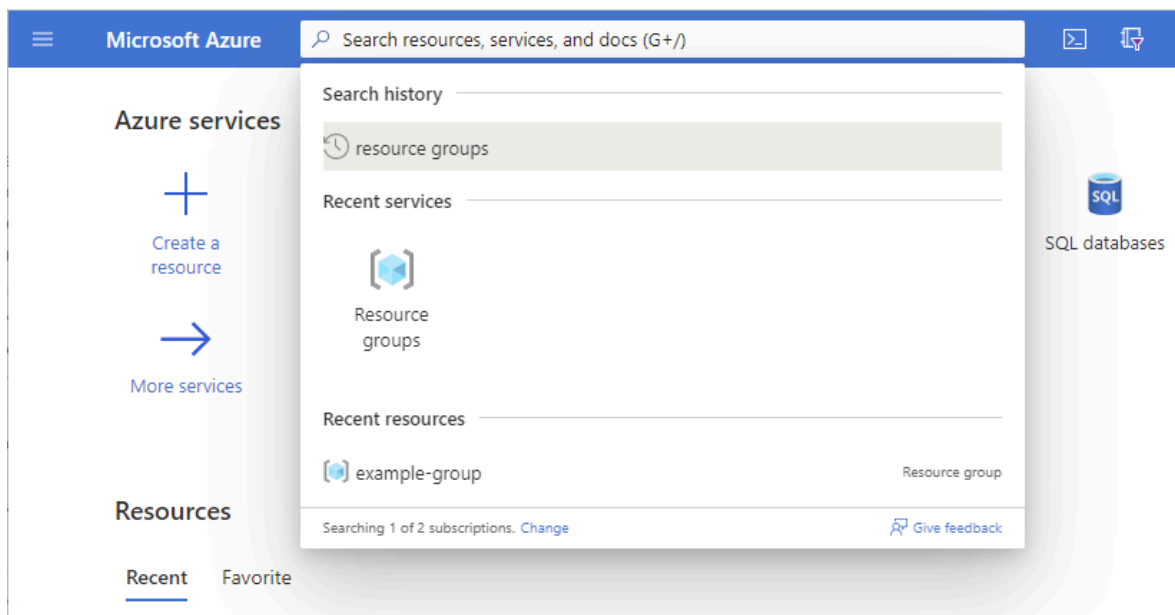
A coworker named Alain at First Up Consultants needs permission to create and manage virtual machines for a project on which they're working. Your manager has asked that you handle this

request. Using the best practice to grant users the least privileges to get their work done, you decide to assign Alain the Virtual Machine Contributor role for a resource group.

Grant access

Follow this procedure to assign the Virtual Machine Contributor role to a user at the resource group scope.

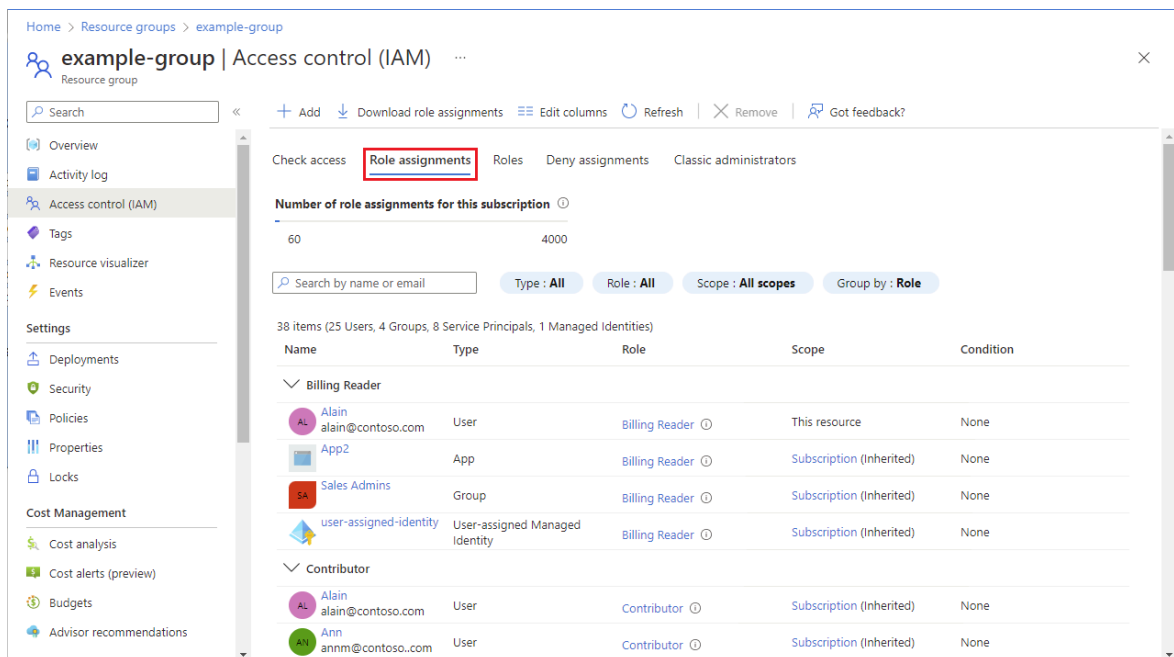
1. Sign in to the [Azure portal](#) as an administrator that has permissions to assign roles, such as [User Access Administrator](#) or [Owner](#).
2. In the Search box at the top, search for **Resource groups**.



3. In the list of resource groups, select a resource group.

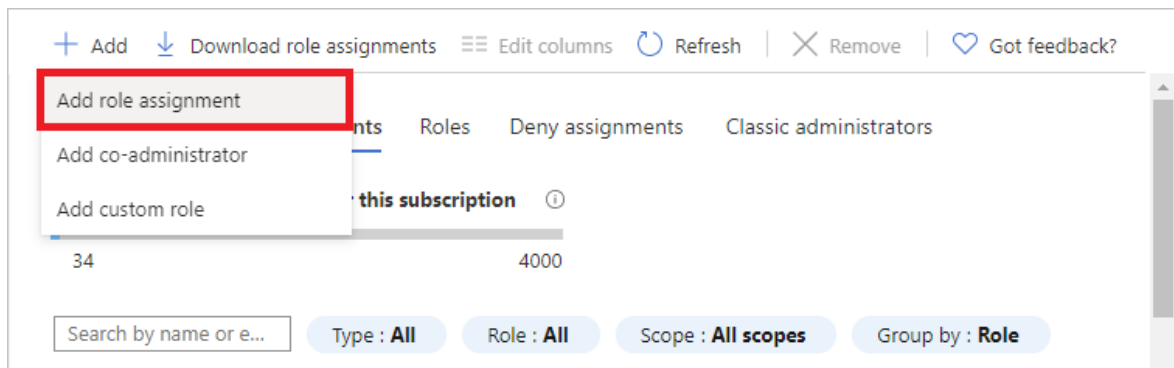
These steps use a resource group named **example-group**, but your resource group's name will be different.

4. On the left menu pane, select **Access control (IAM)**.
5. Select the **Role assignments** tab to display the current list of role assignments at this scope.



6. Select **Add** > **Add role assignment**.

If you don't have permissions to assign roles, the **Add role assignment** option will be disabled.



The **Add role assignment** page opens.

7. On the **Role** tab, search for and select **Virtual Machine Contributor**.

Home > Resource groups > example-group | Access control (IAM) >

Add role assignment

Got feedback?

Role **Members** Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

virtual machine Type: All Category: All

Showing 10 of 372 roles

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Classic Virtual Machine Contributor	Lets you manage classic virtual machines, but not access to them, and not t...	BuiltInRole	Compute	View
Desktop Virtualization Power On Contributor	This role is in preview and subject to change. Provide permission to the Azur...	BuiltInRole	None	View
Desktop Virtualization Power On Off Contr...	This role is in preview and subject to change. Provide permission to the Azur...	BuiltInRole	None	View
Desktop Virtualization Virtual Machine Cont...	This role is in preview and subject to change. Provide permission to the Azur...	BuiltInRole	None	View
DevTest Labs User	Lets you connect, start, restart, and shutdown your virtual machines in your ...	BuiltInRole	Devops	View
Virtual Machine Administrator Login	View Virtual Machines in the portal and login as administrator	BuiltInRole	Compute	View
Virtual Machine Contributor	Lets you manage virtual machines, but not access to them, and not the virtu...	BuiltInRole	Compute	View
Virtual Machine Local User Login	View Virtual Machines in the portal and login as a local user configured on t...	BuiltInRole	None	View
Virtual Machine Operator	Can monitor and restart virtual machines.	CustomRole	None	View

Review + assign Previous Next

8. Select **Next**.

9. On the **Members** tab, select **Select members**.

10. Search for and select a user.

Home > Resource groups > example-group | Access control (IAM) >

Add role assignment

Got feedback?

Role **Members** Review + assign

Selected role
Virtual Machine Contributor

Assign access to
☒ User, group, or service principal
☐ Managed identity

Members
[+ Select members](#)

Name	Object ID	Type
No members selected		

Description
Optional

Review + assign Previous Next

Select members

Select

alain

AT Alain Team

Selected members:

AL Alain
alain@contoso.com [Remove](#)

Select Close

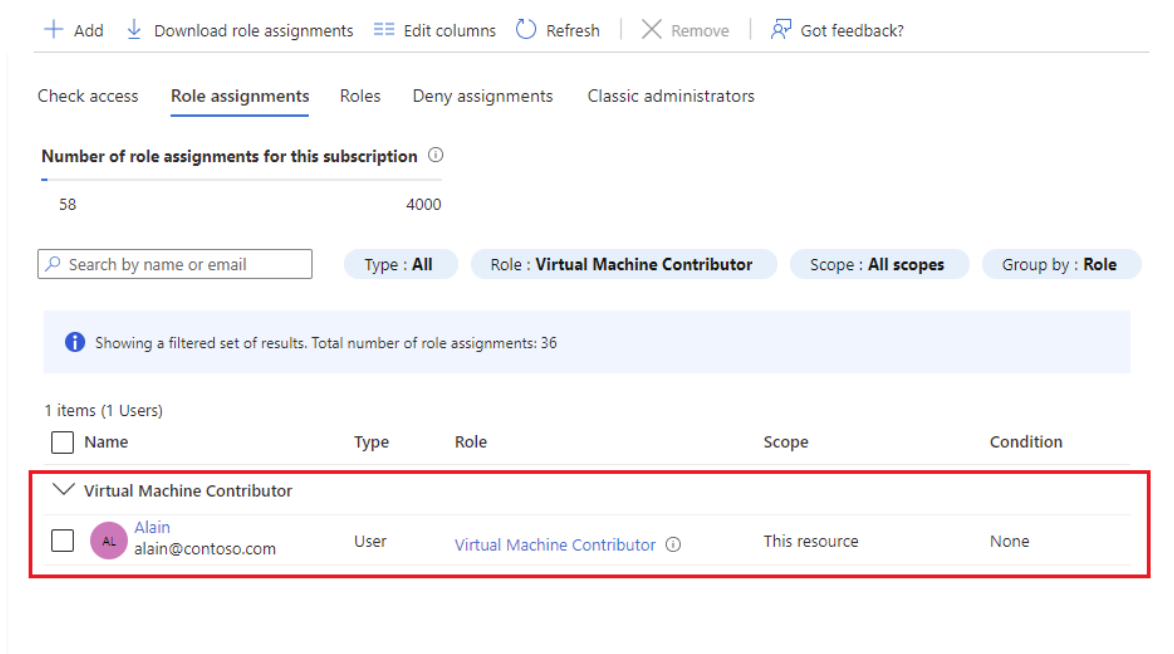
11. Select **Select** to add the user to the Members list.

12. Select **Next**.

13. On the **Review + assign** tab, review the role assignment settings.

14. Select **Review + assign** to assign the role.

After a few moments, the user is assigned the Virtual Machine Contributor role at the resource group scope. The user can now create and manage virtual machines just within this resource group.



Top navigation: + Add, Download role assignments, Edit columns, Refresh, Remove, Got feedback?

Check access | **Role assignments** | Roles | Deny assignments | Classic administrators

Number of role assignments for this subscription ⓘ

58 / 4000

Search by name or email | Type : All | Role : Virtual Machine Contributor | Scope : All scopes | Group by : Role

Showing a filtered set of results. Total number of role assignments: 36

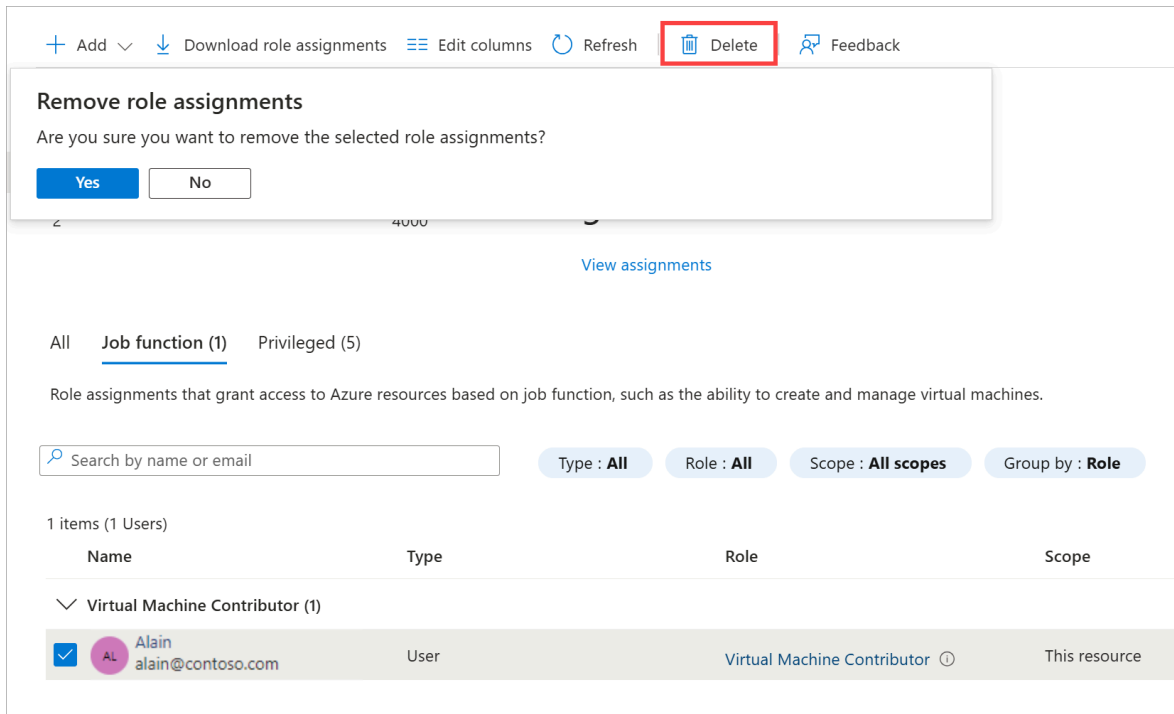
1 items (1 Users)

☐	Name	Type	Role	Scope	Condition
☒	Virtual Machine Contributor				
☐	 Alain alain@contoso.com	User	Virtual Machine Contributor ⓘ	This resource	None

Remove access

In Azure RBAC, you can remove a role assignment to remove access.

1. In the list of role assignments, select **View Assignments**.
2. Search for and check the box for the user with the Virtual Machine Contributor role.
3. Select **Delete**.



4. In the **Remove role assignments** message that appears, select **Yes**.

In this unit, you learned how to grant a user access to create and manage virtual machines in a resource group using the Azure portal.

6. Exercise - View activity logs for Azure RBAC changes

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/6-view-activity-logs>

Exercise - View activity logs for Azure RBAC changes

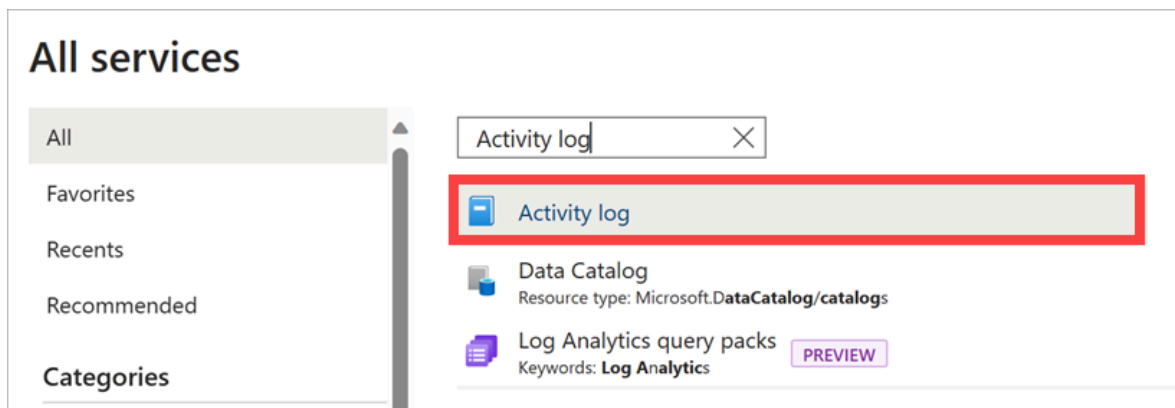
Completed

First Up Consultants reviews Azure RBAC changes quarterly for auditing and troubleshooting purposes. You know that changes get logged in the [Azure Activity Log](#). Your manager has asked if you can generate a report of the role assignment and custom role changes for the last month.

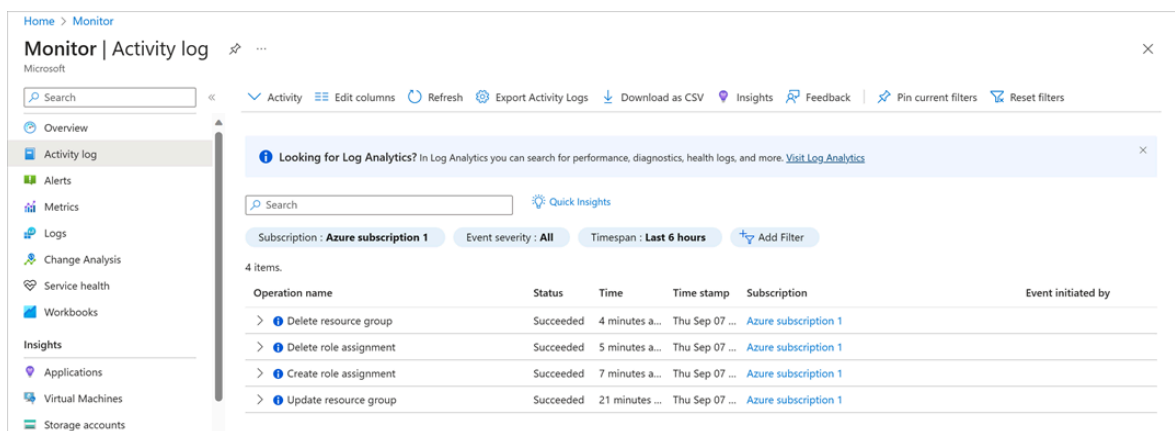
View activity logs

The easiest way to get started is to view the activity logs with the Azure portal.

1. Select **All services**, then search for **Activity log**.



2. Select **Activity log** to open the activity log.

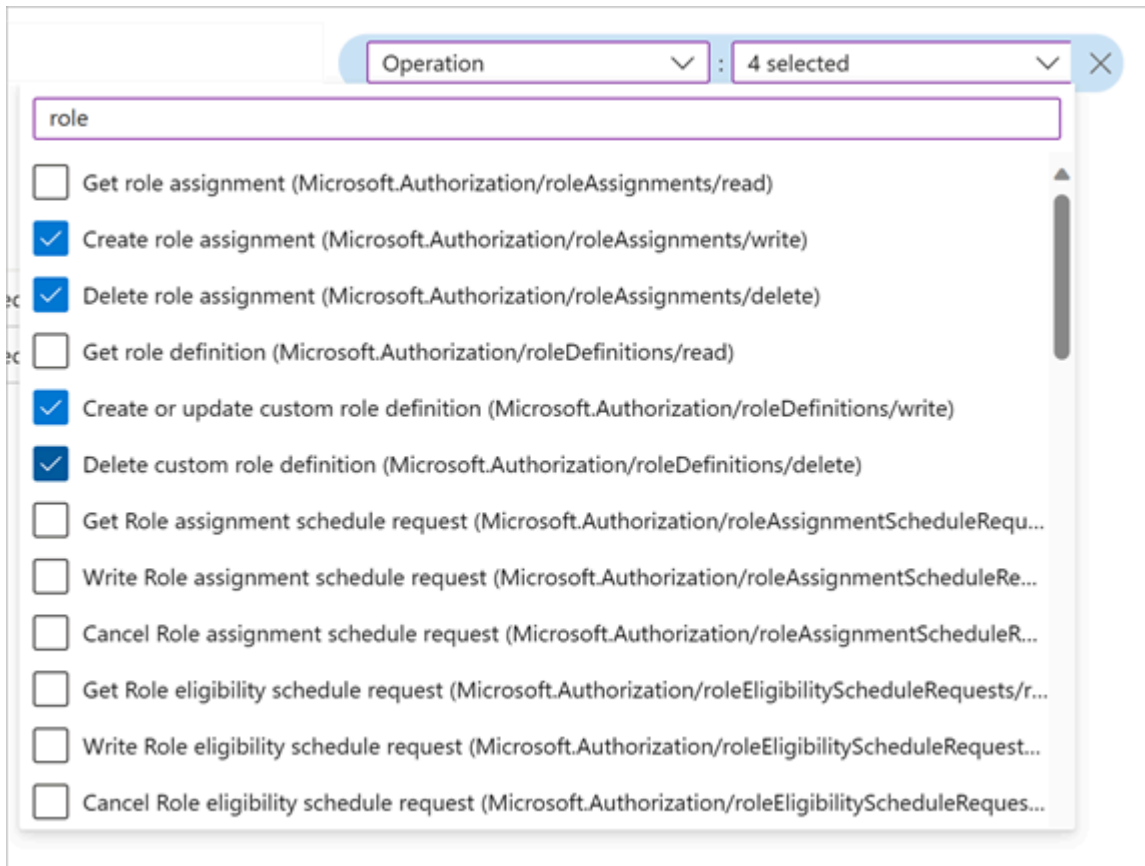


3. Set the **Timespan** filter to **Last month**.

4. Add an **Operation** filter and type **role** to filter the list.

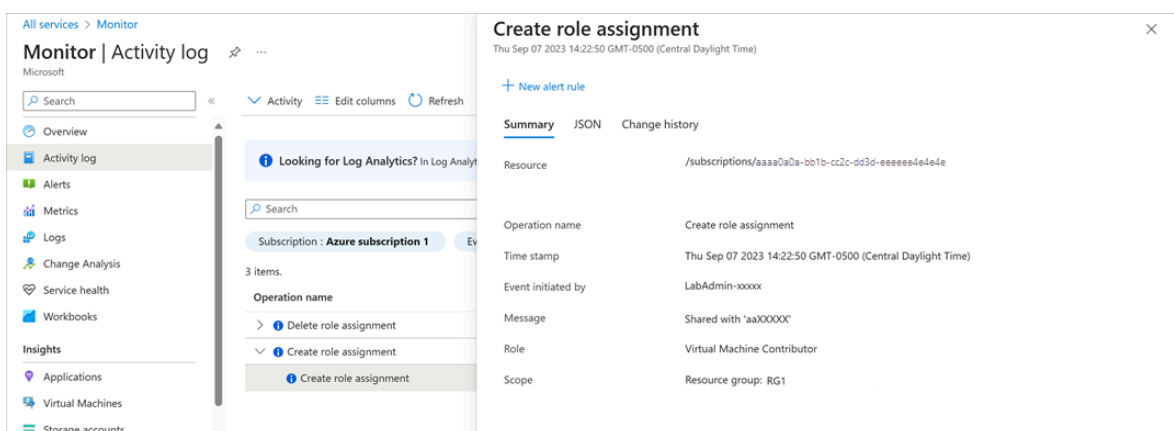
5. Select the following Azure RBAC operations:

- Create role assignment (roleAssignments)
- Delete role assignment (roleAssignments)
- Create or update custom role definition (roleDefinitions)
- Delete custom role definition (roleDefinitions)



After a moment, you'll get a list of all the role assignment and role definition operations for the last month. There's also a button at the top of the screen to download the activity log as a CSV file.

6. Select one of the operations to get the activity log details.



In this unit, you learned how to use Azure Activity Log to list Azure RBAC changes in the portal and generate a simple report.

7. Module assessment

Module assessment

Completed

8. Summary

<https://learn.microsoft.com/en-us/training/modules/secure-azure-resources-with-rbac/8-summary>

Summary

Completed

In this module, you learned about Azure RBAC, and how you can use it to secure your Azure resources. To grant access, you assign users a role at a particular scope. Using Azure RBAC, you can grant only the amount of access to users that they need to perform their jobs.

Azure RBAC has more than 200 built-in roles. However, if your organization needs specific permissions, you can create your own custom roles. Azure keeps track of your Azure RBAC changes in case you need to see what changes were made in the past.

Further reading

To continue learning about Azure RBAC, check out [What is Azure role-based access control \(Azure RBAC\)?](#).

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

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/allow-users-reset-their-password/1-introduction>

Introduction

Completed

Suppose you're an IT administrator for a large retail organization. Your organization starts using Microsoft Entra ID to allow employees to securely sign in and use software as a service (SaaS) apps. It also allows access to the organization's resources in Microsoft 365. You're overwhelmed with password-reset requests because you currently reset employees' passwords manually. To get these employees back to being productive quickly and reduce your workload, you decide to evaluate and set up self-service password reset in Microsoft Entra ID.

In this module, you learn how Azure supports this feature and how to set it up. Only paid subscriptions can leverage this, while free and pay-as-you-go can't.

By the end of this module, you'll be able to configure self-service password reset in Microsoft Entra ID.

Learning objectives

In this module, you:

- Decide whether to implement self-service password reset.
- Implement self-service password reset to meet your requirements.
- Configure self-service password reset to customize the experience.

Prerequisites

- Basic understanding of Microsoft Entra ID
- A working Microsoft Entra tenant with a subscription package that includes Microsoft Entra ID P1

- An account with at least the Authentication Policy Administrator role

2. What is self-service password reset in Microsoft Entra ID?

<https://learn.microsoft.com/en-us/training/modules/allow-users-reset-their-password/2-self-service-password-reset>

What is self-service password reset in Microsoft Entra ID?

Completed

3. Implement Microsoft Entra self-service password reset

<https://learn.microsoft.com/en-us/training/modules/allow-users-reset-their-password/3-implement-azure-ad-self-service-password-reset>

Implement Microsoft Entra self-service password reset

Completed

You've decided to implement self-service password reset (SSPR) in Microsoft Entra ID for your organization. You want to start using SSPR for a group of 20 users in the marketing department as a trial deployment. If everything works well, you'll enable SSPR for your whole organization.

In this unit, you'll learn how to enable SSPR in Microsoft Entra ID.

Prerequisites

Before you start to configure SSPR, you need a:

- **Microsoft Entra organization:** This organization must have at least a P1 or P2 trial license enabled.

- **Microsoft Entra account with Authentication Policy Administrator role:** You'll use this account to set up SSPR.
- **Non-administrative user account:** You'll use this account to test SSPR. It's important that this account isn't an administrator, because Microsoft Entra imposes extra requirements on administrative accounts for SSPR. This user, and all user accounts, must have a valid license to use SSPR.
- **Security group with which to test the configuration:** The non-administrative user account must be a member of this group. You'll use this security group to limit who you roll SSPR out to.

Scope of SSPR rollout

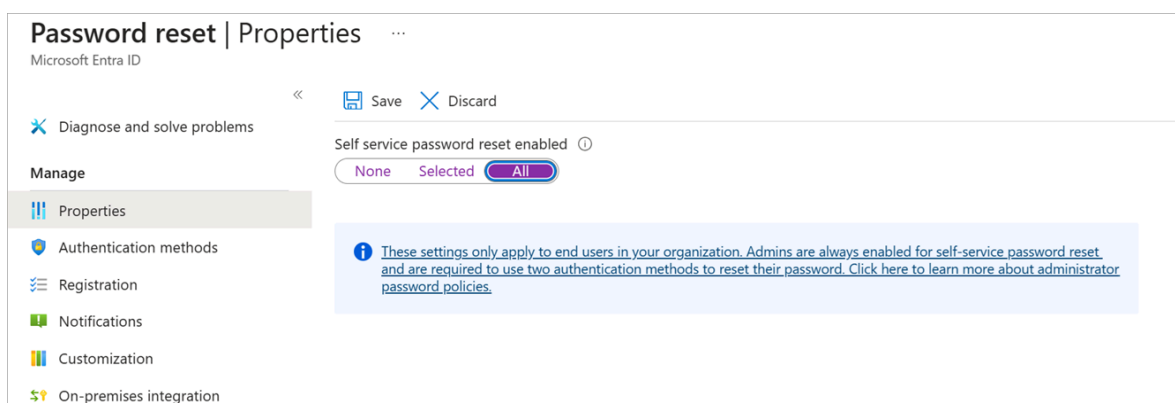
There are three settings for the **Self-service password reset enabled** property:

- **None:** No users in the Microsoft Entra organization can use SSPR. This value is the default.
- **Selected:** Only the members of the specified security group can use SSPR. You can use this option to enable SSPR for a targeted group of users who can test it and verify that it works as expected. When you're ready to roll it out broadly, set the property to **Enabled** so that all users have access to SSPR.
- **All:** All users in the Microsoft Entra organization can use SSPR.

Configure SSPR

Here are the high-level steps to configure SSPR:

1. Go to the [Azure portal](#), then to **Microsoft Entra ID > Manage > Password reset**.
2. **Properties:**
 - Enable SSPR.
 - You can enable it for all users in the Microsoft Entra organization or for selected users.
 - To enable for selected users, you must specify the security group. Members of this group can use SSPR.



3. **Authentication methods:**

- Choose whether to require one or two authentication methods.
- Choose the authentication methods that the users can use.

Password reset | Authentication methods ...

Microsoft Entra ID

<< Save Discard

Diagnose and solve problems

Manage

- Properties
- Authentication methods**
- Registration
- Notifications
- Customization
- On-premises integration
- Administrator Policy

Activity

- Audit logs
- Usage & insights

Troubleshooting + Support

- New support request

Authentication Methods for SSPR and Signin can now be managed in one converged policy. [Learn more](#)

Number of methods required to reset ⓘ

1 2

Methods available to users

- ☐ Mobile app notification
- ☐ Mobile app code
- ☒ Email
- ☒ Mobile phone (SMS only)
- ☐ Office phone ⓘ
- ☐ Security questions

These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password. [Click here to learn more about administrator password policies.](#)

4. **Registration:**

- Specify whether users are required to register for SSPR when they next sign in.
- Specify how often users are asked to reconfirm their authentication information.

Password reset | Registration ...

Microsoft Entra ID

<< Save Discard

Diagnose and solve problems

Manage

- Properties
- Authentication methods
- Registration**
- Notifications
- Customization
- On-premises integration

Require users to register when signing in? ⓘ

Yes No

Number of days before users are asked to re-confirm their authentication information * ⓘ

180

These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password. [Click here to learn more about administrator password policies.](#)

5. **Notifications:** Choose whether to notify users and administrators of password resets.

Password reset | Notifications ...

Microsoft Entra ID

<< Save Discard

✖ Diagnose and solve problems

Manage

- Properties
- Authentication methods
- Registration
- Notifications**
- Customization
- On-premises integration

Notify users on password resets? ⓘ

☒ Yes ☐ No

Notify all admins when other admins reset their password? ⓘ

☐ Yes ☒ No

6. **Customization:** Provide an email address or web page URL where your users can get help.

Password reset | Customization ...

Microsoft Entra ID

<< Save Discard

✖ Diagnose and solve problems

Manage

- Properties
- Authentication methods
- Registration
- Notifications
- Customization**
- On-premises integration

Customize helpdesk link ⓘ

☒ Yes ☐ No

Custom helpdesk email or URL ⓘ

✓

4. Exercise - Set up self-service password reset

<https://learn.microsoft.com/en-us/training/modules/allow-users-reset-their-password/4-exercise-set-up-self-service-password-reset>

Exercise - Set up self-service password reset

Completed

In this unit, you'll configure and test self-service password reset (SSPR) by using your email. You'll need to use your email to complete the password-reset process in this exercise.

Create a group

You want to roll out SSPR to a limited set of users first to make sure your SSPR configuration works as expected. Let's begin by creating a security group for the limited rollout.

1. In the Microsoft Entra organization you created, under **Manage**, select **Groups**.
2. Select **New Group**.
3. Enter the following values:

Setting	Value
Group type	Security
Group name	SSPRTesters
Group description	Members are testing the rollout of SSPR
Membership type	Assigned

4. Select **Create**.

New Group

 Got feedback?

Group type * ⓘ
Security

Group name * ⓘ
SSPRTesters

Group description ⓘ
Members are testing the rollout of SSPR

Azure AD roles can be assigned to the group ⓘ
☐ Yes ☒ No

Membership type * ⓘ
Assigned

Owners
No owners selected

Members
No members selected

Create

Create a user account

To test your configuration, create an account that's not associated with an administrator role. You'll also assign the account to the group you created.

1. In your Microsoft Entra organization, under **Manage**, select **Users**.
2. Select **+ New user**, select **Create new user** in the drop-down, and use the following values:

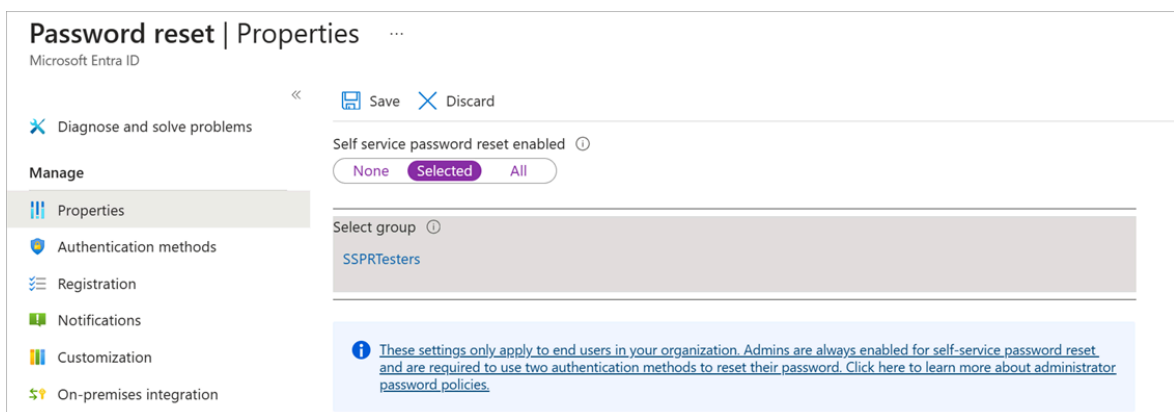
Setting	Value
User principal name	balas
Display name	Bala Sandhu
Password	Select the Copy icon next to the autogenerated password, then paste the password to a text editor like Notepad.

3. Select the **Assignments** tab.
4. Select **Add group**, check the box for the **SSPRTesters** group, and then the **Select** button.
5. Select **Review + create** and then select **Create**.

Enable SSPR

Now, you're ready to enable SSPR for the group.

1. In your Microsoft Entra organization, under **Manage**, select **Password reset**.
2. On the **Properties** page, select **Selected**. Select the link under **Select Group**, select the box next to the **SSPRTesters** group, and then the **Select** button.
3. Select **Save**.



4. Under **Manage**, select the **Authentication methods**, **Registration**, and **Notifications** pages to review the default values. Ensure **Authentication methods** has **Email** selected.

5. Select **Customization**.
6. Select **Yes**, and then in the **Custom helpdesk email or URL** text box, enter **admin@organization-domain-name.onmicrosoft.com**. Replace "organization-domain-name" with the domain name of the Microsoft Entra organization you created. If you've forgotten the domain name, hover over your profile in the Azure portal.
7. Select **Save**.

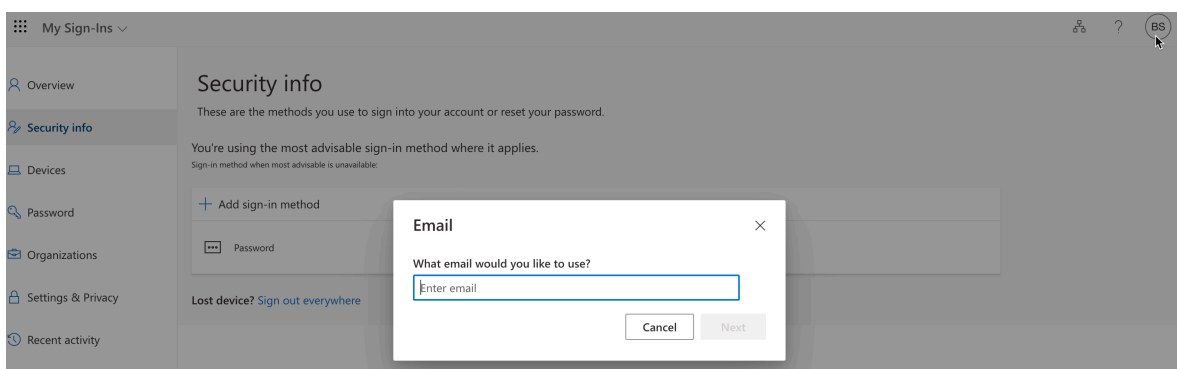
Register for SSPR

Now that the SSPR configuration is complete, register an email for the user you created.

Note

If you get a message that says "The administrator has not enabled this feature," use private/incognito mode in your web browser.

1. In a new browser window, go to <https://aka.ms/ssprsetup>.
2. Sign in with the user name **balas@organization-domain-name.onmicrosoft.com** and the password that you noted earlier. Remember to replace "organization-domain-name" with the domain name of the Microsoft Entra organization you created.
3. If you're asked to update your password, enter a new password of your choice. Make sure you note the new password.
4. Select the **Security info** tab, and then select **+ Add sign-in method**.
5. In the **Add a method** box, select **Email**.
6. Enter your email details.

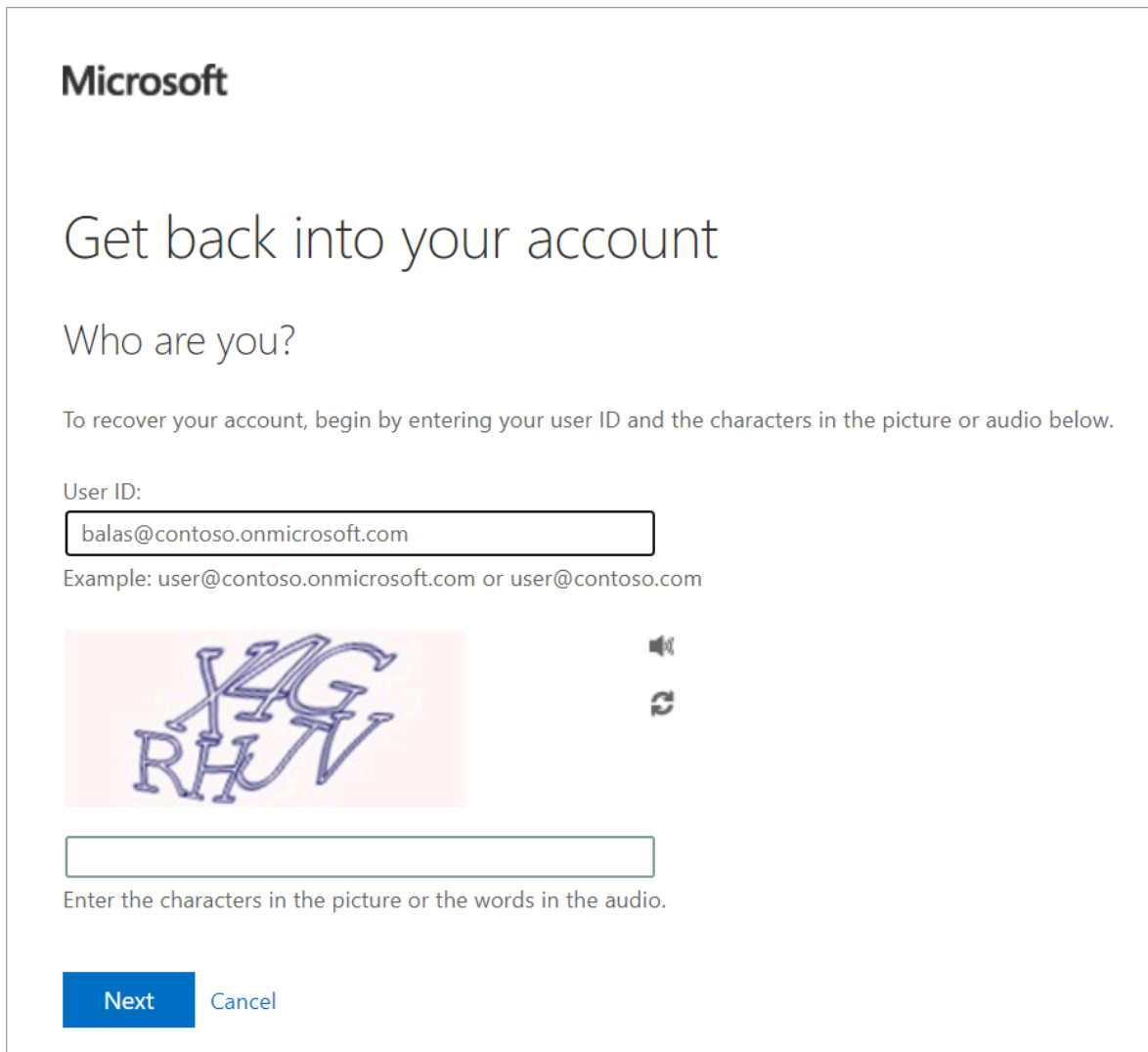


7. When you receive the code in your email, enter the code in the text box and select **Next**.

Test SSPR

Now, let's test whether the user can reset their password.

1. In a new browser window, go to <https://aka.ms/sspr>.
2. For **User ID**, type **balas@organization-domain-name.onmicrosoft.com**. Replace "organization-domain-name" with the domain you used for your Microsoft Entra organization.



Microsoft

Get back into your account

Who are you?

To recover your account, begin by entering your user ID and the characters in the picture or audio below.

User ID:

Example: user@contoso.onmicrosoft.com or user@contoso.com



Enter the characters in the picture or the words in the audio.

Next Cancel

3. Complete the CAPTCHA and select **Next**.
4. The **Email my alternate email** radio button is selected. Select **Email**.
5. When the email arrives, in the **Enter your verification code** text box, enter the code you were sent. Select **Next**.
6. Enter a new password, and then select **Finish**. Make sure you note the new password.
7. Close the browser window.

5. Exercise - Customize directory branding

Exercise - Customize directory branding

Completed

Suppose you've been asked to display your retail organization's branding on the Azure sign-in page to reassure users that they're passing credentials to a legitimate system. Here, you'll learn how to configure this custom branding.

To complete this exercise, you must have two image files:

- A page background image. This must be a PNG or JPG file, 1920 x 1080 pixels, and smaller than 300 KB.
- A company logo image. This must be a PNG or JPG file, 32 x 32 pixels, and smaller than 5 KB.

Customize Microsoft Entra organization branding

Let's use Microsoft Entra ID to set up the custom branding.

1. Sign in to the [Azure portal](#).
2. Go to your Microsoft Entra organization by selecting **Microsoft Entra ID**. If you're not in the right Microsoft Entra organization, go to your Azure profile, and select **Switch directory** to find your organization.
3. Under **Manage**, select **Company branding**. Then select the **Customize** button in the center of the screen.
4. Next to **Favicon**, select **Browse**. Select your logo image.
5. Next to **Background image**, select **Browse**. Select your page background image.
6. Select a **Page background color** or accept the default.

Customize default sign-in experience

Basics

Layout

Header

Footer

Sign-in form

Review

Customize the default sign-in experience to personalize the sign-in page for anyone signing-in to your tenant.

Background

Upload background image, favicon or choose background color.

Favicon ⓘ

Select file(s)

Browse

Image size: 32x32px
(resizable)
Max file size: 5KB
File Type: PNG (preferred),
JPG, or JPEG

Background image ⓘ

Select file(s)

Browse

ⓘ Image will appear darkened to improve contrast and legibility.

Image size: 1920x1080px
Max file size: 300KB
File Type: PNG, JPG, or JPEG

Page background color ⓘ

Not provided

Review + create

< Previous

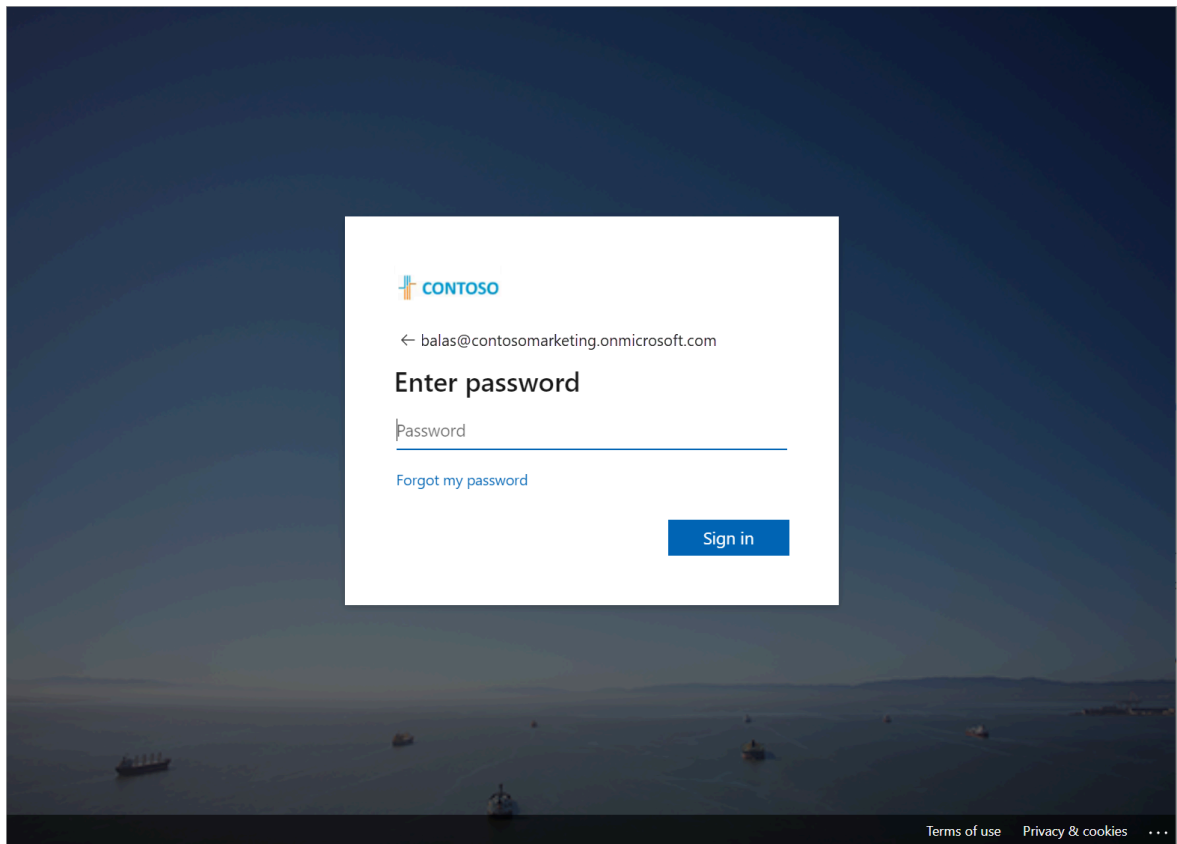
Next: Layout >

7. Select **Review + Create**, and then select **Create**.

Test the organization's branding

Now, let's use the account that we created in the last exercise to test the branding.

1. In a new browser window, go to <https://login.microsoft.com>.
2. Select the account for **Bala Sandhu**. Your custom branding is displayed.



3. Select **Forgot my password**.



Get back into your account

Who are you?

To recover your account, begin by entering your user ID and the characters in the picture or audio below.

User ID:

Example: user@contoso.onmicrosoft.com or user@contoso.com



Enter the characters in the picture or the words in the audio.

Next

Cancel

6. Summary

<https://learn.microsoft.com/en-us/training/modules/allow-users-reset-their-password/6-summary>

Summary

Completed

In this module, you've learned how you can use SSPR in Microsoft Entra ID to allow users to reset their forgotten or expired passwords. An administrator doesn't have to do the password reset. SSPR is secured by authentication methods of your choice. These methods can include a mobile authentication app, a code sent to you by an SMS text message, or security questions.

SSPR helps reduce the amount of work required from administrators. It also minimizes the productivity impact for users when they forget their password.

Clean up

Remember to clean up after you've finished.

- **Delete the user you created in Microsoft Entra ID:** Go to **Microsoft Entra ID > Manage > Users**. Check the box next to the user and select **Delete**. Select **OK**.
- **Delete the group you created in Microsoft Entra ID:** Go to **Microsoft Entra ID > Manage > Groups**. Check the box next to the group and select **Delete**. Select **OK**.
- **Turn off self-service password reset:** Go to **Microsoft Entra ID > Manage > Password reset**. Under **Self service password reset enabled**, select **None**. Select **Save**.

If you created a Premium trial Microsoft Entra tenant for this module, you can delete the tenant 30 days after the trial has expired.

Learn more

- [Tutorial: Enable users to unlock their account or reset passwords using Microsoft Entra self-service password reset](#)
- [How it works: Microsoft Entra self-service password reset](#)
- [Enable self-service password reset](#)