Introduction

3 minutes

The term governance describes the general process of establishing rules and policies. Governance ensures those rules and policies are enforced.

A good governance strategy helps you maintain control over the applications and resources that you manage in the cloud. Maintaining control over your environment ensures that you stay compliant with:

- Industry standards, such as information security management.
- Corporate or organizational standards, such as ensuring that network data is encrypted.

Governance is most beneficial when you have:

- Multiple engineering teams working in Azure.
- Multiple subscriptions to manage.
- Regulatory requirements that must be enforced.
- Standards that must be followed for all cloud resources.

Meet Tailwind Traders

Tailwind Trader is a fictitious home improvement retailer. It operates retail hardware stores across the globe and online. The Tailwind Traders CTO is aware of the opportunities offered by Azure, but also understands the need for strong governance. Without strong governance, the company may end up with a difficult to manage Azure environment and costs, which are hard to track and control. The CTO is interested in understanding how Azure manages and enforces governance standards.

Learning objectives

In this module, you'll learn how to:

- Design for governance.
- Design for management groups.

- Design for Azure subscriptions.
- Design for resource groups.
- Design for Azure policies.
- Design for resource tags.
- Design for Azure blueprints.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design Identity, Governance, and Monitoring

- Design Governance

Prerequisites

- Conceptual knowledge of governance policies, resource organization, and subscription management.
- Working experience with organizing resources, applying governance policies, and enforcing compliance requirements.

Next unit: Design for governance

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

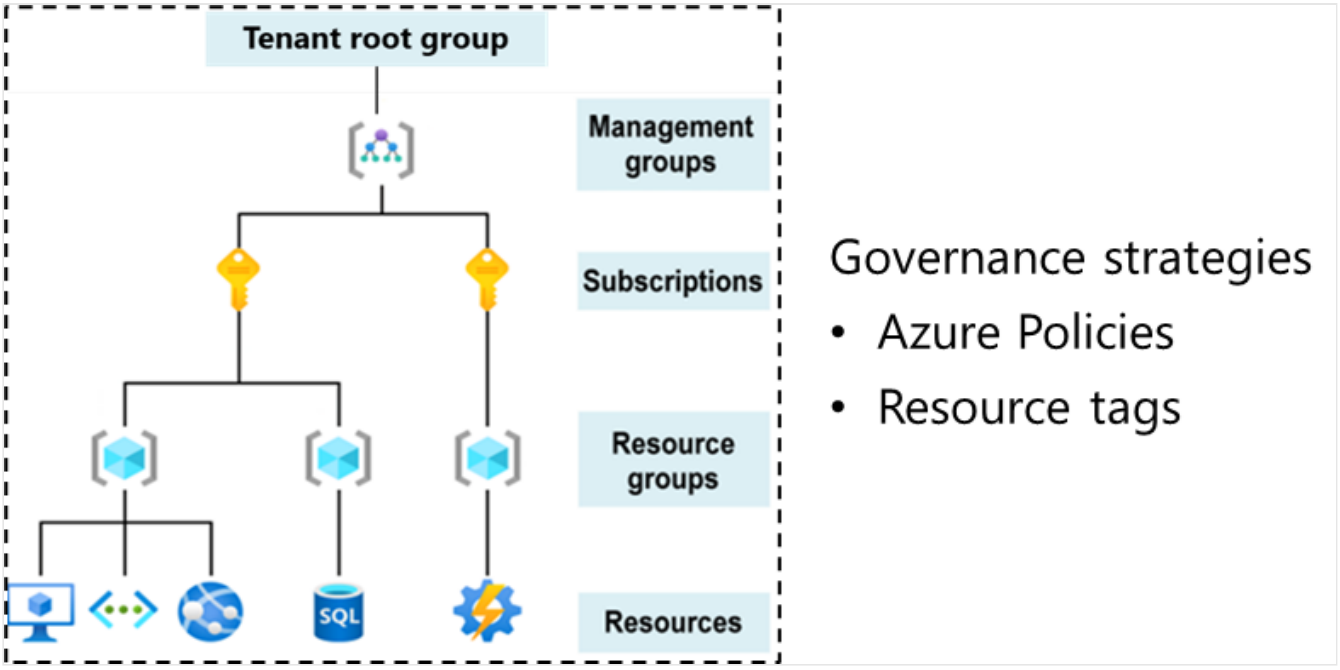
100 XP

Design for governance

3 minutes

Governance provides mechanisms and processes to maintain control over your applications and resources in Azure. Governance involves determining your requirements, planning your initiatives, and setting strategic priorities.

To effectively apply your governance strategies, you must first create a hierarchical structure. This structure lets you apply governance strategies exactly where they're needed. The governance strategies we'll cover in this module are Azure policy and resource tags.



A typical Azure hierarchy has four levels: management groups, subscriptions, resource groups, and resources. We'll examine each level in more detail, but here's an overview.

- **Management groups** help you manage access, policy, and compliance for multiple subscriptions.
- **Subscriptions** are logical containers that serve as units of management and scale. Subscriptions are also billing boundaries.
- **Resource groups** are logical containers into which Azure resources are deployed and managed.

- **Resources** are instances of services that you create. For example, virtual machines, storage, and SQL databases.

> **Note**
>
> The **tenant root group** contains all the management groups and subscriptions. This group allows global policies and Azure role assignments to be applied at the directory level.

Next unit: Design for management groups

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for management groups

3 minutes

Management groups are containers that help you manage access, policy, and compliance across **multiple subscriptions**. You can use management groups to:

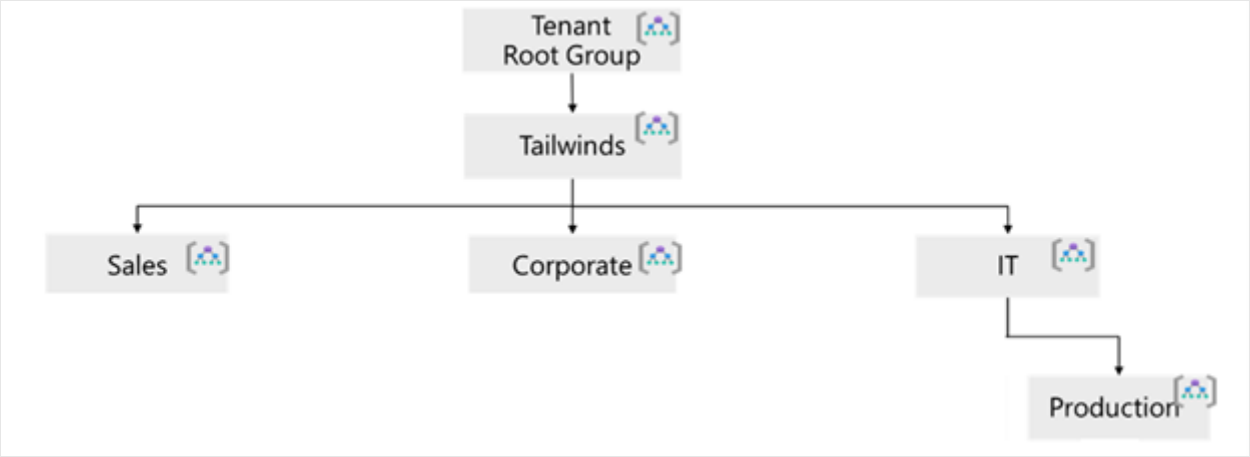
- Limit in which regions, across several subscriptions, virtual machines can be created.
- Provide user access to multiple subscriptions by creating one role assignment that will be inherited by other subscriptions.
- Monitor and audit, across subscriptions, role, and policy assignments.

Things to know about management groups

- Management groups can be used to aggregate policy and initiative assignments via Azure Policy.
- A management group tree can support up to six levels of depth. This limit doesn't include the tenant root level or the subscription level.
- Azure role-based access control authorization for management group operations isn't enabled by default.
- By default, all new subscriptions will be placed under the root management group.

Things to consider when creating management groups

Tailwind Traders has Sales, Corporate, and IT departments. The Sales department manages offices in the West and in the East. The Corporate main office includes Human Resources (HR) and Legal. The Information Technology (IT) department handles research, development, and production. There are currently two apps hosted in Azure. Here's a proposed management group hierarchy.



- **Design management groups with governance in mind.** For example, apply Azure policies at the management group level for all workloads that require the same security, compliance, connectivity, and feature settings.
- **Keep the management group hierarchy reasonably flat**. Ideally have no more than three or four levels. A hierarchy that is too flat doesn't provide flexibility and complexity for large organizations. A hierarchy with too many levels will be difficult to manage.
- **Consider a top-level management group.** This management group supports common platform policy and Azure role assignments across the whole organization. For example, the Tailwinds management group is a top-level management group for all organizational-wide policies.
- **Consider an organizational or departmental structure**. An organizational structure will be easy to understand. For example, the Sales, Corporate, and IT management groups.
- **Consider a geographical structure**. A geographical structure allows for compliance policies in different regions. For example, the West and East management groups in Sales.
- **Consider a production management group.** A production management group creates policies that apply to all corporate products. In our example, the Production management group provides product-specific policies for corporate apps.
- **Consider a sandbox management group.** A sandbox management group lets users experiment with Azure. The sandbox provides isolation from your development, test, and production environments. Users can experiment with resources that might not yet be allowed in production environments.
- **Consider isolating sensitive information in a separate management group.** In our example, the Corporate management group provides more standard and compliance policies for the main office.

100 XP

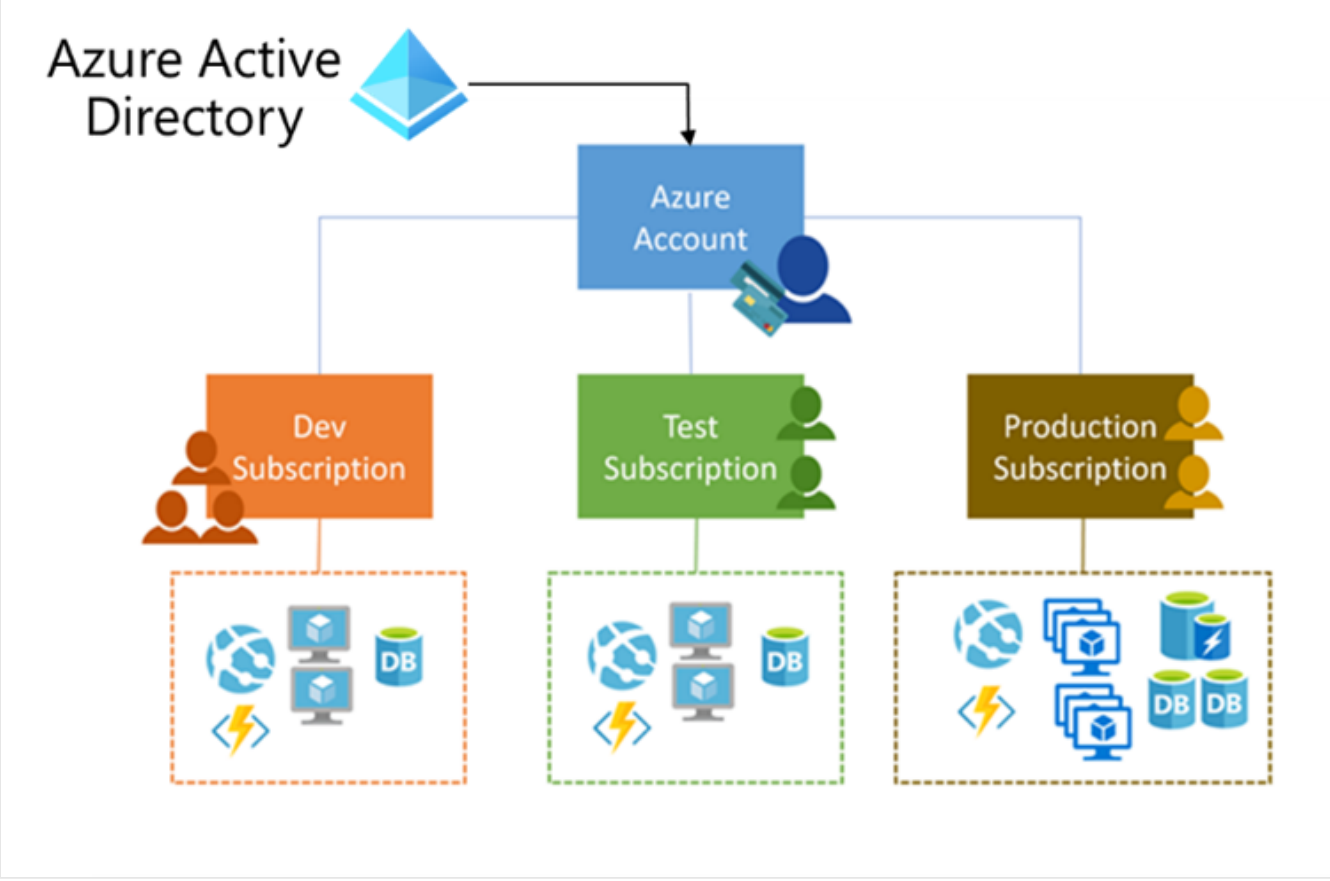
Design for subscriptions

3 minutes

Azure Subscriptions are logical containers that serve as units of management and scale and billing boundaries. Limits and quotas can be applied, and each organization can use subscriptions to manage costs and resources by group.

Things to know about subscriptions

Using Azure requires an Azure subscription. A subscription provides you with a logical container to provision and pay for Azure products and services. There are several types of subscriptions , such as Enterprise Agreement and Pay-as-You-Go.



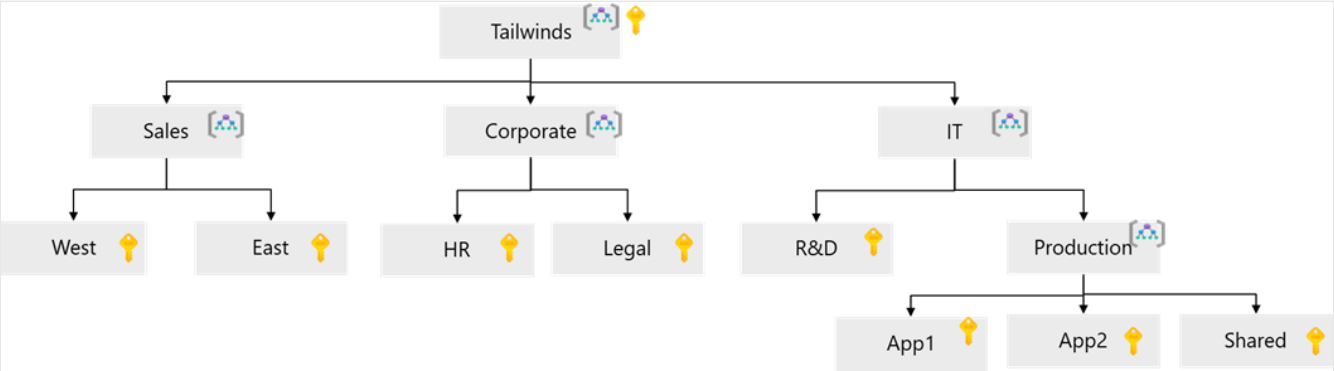
You can use subscriptions to:

- Organize specialized workloads that need to scale outside the existing subscription limits.
- Provide different billing environments such as development, test, and production.
- Achieve compliance by applying policies to a subscription.
- Manage and track costs for your organizational structure

- Manage and track costs for your organizational structure.

Things to consider when creating subscriptions

Tailwind Traders has established their management group structure. Now they need to determine where to assign subscriptions. Here's one possible solution.



- **Treat subscriptions as a democratized unit of management**. Align your subscriptions with business needs and priorities.
- **Group subscriptions together under management groups**. Grouping ensures that subscriptions with the same set of policies and Azure role assignments can inherit them from a management group. For example, both the West and East subscriptions will inherit policy from the Sales management group.
- **Consider a dedicated shared services subscription**. A shared services subscription ensures that all common network resources are billed together and isolated from other workloads. For example, Azure ExpressRoute and Virtual WAN.
- **Consider subscription scale limits.** Subscriptions serve as a scale unit for component workloads. For example, large, specialized workloads like high-performance computing, IoT, and SAP are all better suited to use separate subscriptions. Separate subscriptions will avoid resource limits (such as a limit of 50 Azure Data Factory integrations).
- **Consider administrative management.** Subscriptions provide a management boundary, which allows for a clear separation of concerns. Will each subscription need a separate administrator, or can you combine subscriptions? In our example, the Corporate management group could have a single subscription for both the HR and Legal departments.
- **Consider how you'll assign Azure policies?** Both management groups and subscriptions serve as a boundary for the assignment of Azure policies. For example, workloads such as Payment Card Industry (PCI) workloads typically require additional policies to achieve compliance. Instead of using a management group to group workloads that require PCI

compliance, you can achieve the same isolation with a subscription. These types of decisions ensure you don't have too many management groups with only a few subscriptions.

- **Consider network topologies.** Virtual networks can't be shared across subscriptions. Resources can connect across subscriptions with different technologies such as virtual network peering or Virtual Private Networks (VPNs). Consider which workloads must communicate with each other when you decide whether a new subscription is required.
- **Consider making subscription owners aware of their roles and responsibilities**. For example, conduct an access review using Azure AD Privileged Identity Management quarterly or twice a year. Access reviews ensure privileges don't proliferate as users move within the customer organization.

> **Note**
>
> One size doesn't fit all for subscriptions. What works for one business unit might not work for another.

Next unit: Design for resource groups

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for resource groups

3 minutes

Resource groups are logical containers into which Azure resources are deployed and managed. These resources can include web apps, databases, and storage accounts. You can use resource groups to:

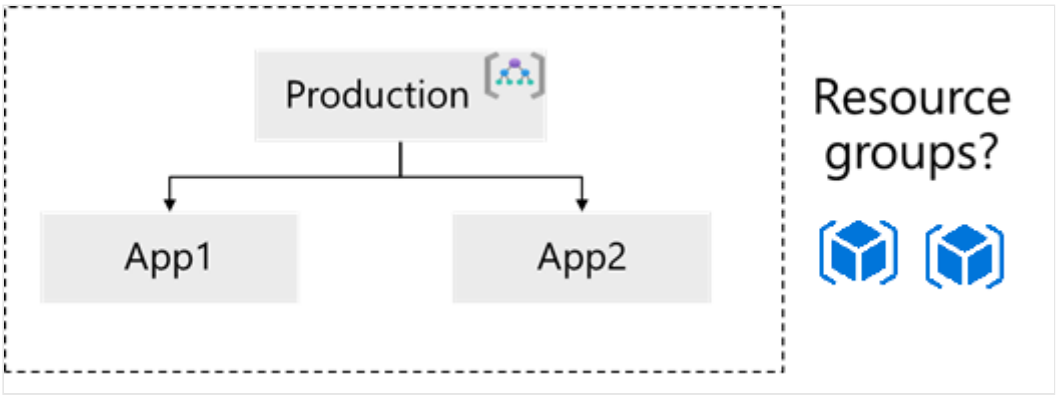
- Place resources of similar usage, type, or location in logical groups.
- Organize resources by life cycle so all the resources can be created or deleted at the same time.
- Apply role permissions to a group of resources or give a group access to administer a group of resources.
- Use resource locks to protect individual resources from deletion or change.

Things to know about resource groups

- Resource groups have their own location (region) assigned. This region is where the metadata is stored.
- If the resource group's region is temporarily unavailable, you can't update resources in the resource group because the metadata is unavailable. The resources in other regions will still function as expected, but you can't update them.
- Resources in the resource group can be in different regions.
- A resource can connect to resources in other resource groups. For example, you can have a web app that connects to a database in a different resource group.
- Resources can be moved between resource groups with a some exceptions.
- You can add a resource to or remove a resource from a resource group at any time.
- Resource groups can’t be nested.
- Each resource must be in one, and only one, resource group.
- Resource groups cannot be renamed.

Things to consider when creating resource groups

Tailwind Traders has two Azure-based apps (App1 and App2). Each app has a web service with SQL database, virtual machines, and storage. Tailwind Traders needs to decide how to organize their resource groups.



- **Consider group by type**. Group by type is most appropriate for on-demand services that aren't associated with an app. In our example, a resource group for the SQL databases (SQL-RG) and a separate resource group (WEB-RG) for the web services.

- **Consider group by app**. Group by app is appropriate when all the resources have the same policies and life cycle. This method could also be applied to test or prototype environments. For Tailwind Traders, App1 and App2 would have separate resource groups. Each group would have all the resources for that application.

- **Consider group by department, group by location (region), and group by billing (cost center)**. These grouping strategies aren't common but may be useful in your situation.
- **Consider a combination of organizational strategies.** Don't restrict your thinking to one strategy. Often a combination of different strategies is best.
- **Consider resource life cycle.** Do you want to deploy, update, and delete resources at the

same time? If so, you may want to place all those resources in one resource group.

- **Consider administration overhead**. How many resource groups would you like to manage? Do you have centralized or decentralized Azure administrators?
- **Consider resource access control**. At the resource group level, you can assign Azure policies, Azure roles, and resource locks. Resource locks prevent unexpected changes to critical resources.
- **Consider compliance requirements.** Do you need to ensure your resource group metadata is stored in a particular region?

Next unit: Design for resource tags

Continue >

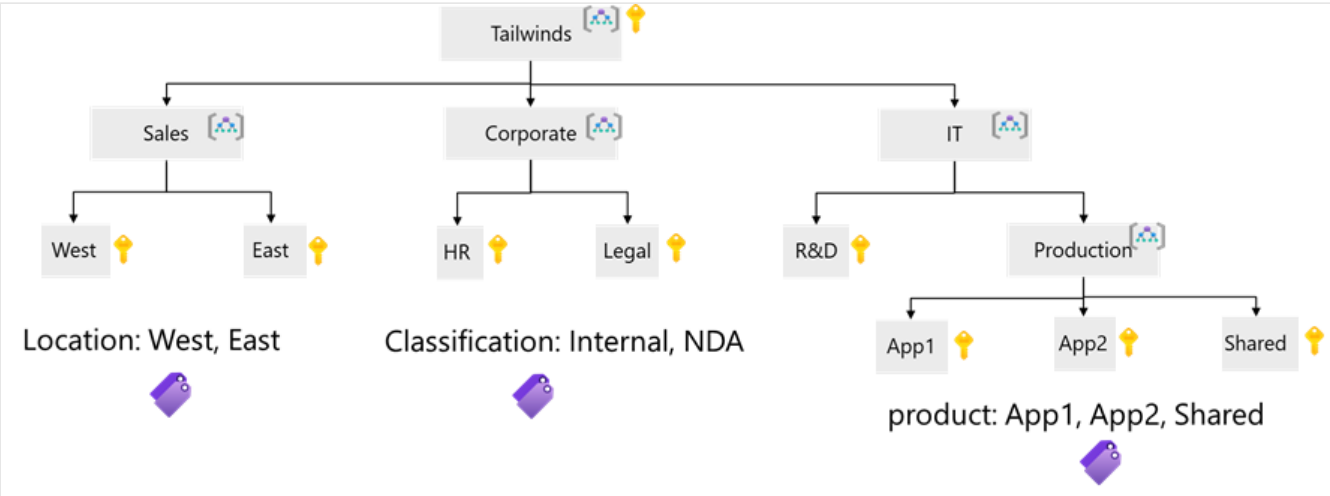
How are we doing? ☆ ☆ ☆ ☆ ☆

Things to know about resource tags

- A resource tag consists of a name-value pair. For example, env: production or env: test.
- You can assign one or more tags to each Azure resource, resource group, or subscription.
- You can add, modify, or delete resource tags. These actions can be done with PowerShell, the Azure CLI, Azure Resource Manager templates, the REST API, or the Azure portal.
- You can apply tags to a resource group. However, tags applied to the resource group aren't inherited by the resources.

Things to consider when creating resource tags

Tailwind Traders has created their organization hierarchy and now needs to determine which resource tags to apply.



- **Consider your organization's taxonomy.** Has your organization already defined terms for compliance or cost reporting? Aligning tags with accepted department nomenclature will make it easier to understand. For example, are office locations, confidentiality levels, and programs already defined?
- **Consider whether you need IT-aligned or business-aligned tagging.** Generally, you can choose IT-aligned tagging or business-aligned tagging. A combination of the two approaches can be effective. Many organizations are shifting from IT-aligned to

business-aligned tagging strategies.

Tagging scheme	Description	Example
IT-aligned tagging	Focuses on workload, application, function, or environment criteria. Reduces the complexity of monitoring assets. Simplifies making management decisions based on operational requirements.	Our printers are busy 80% of the time. We have five high-speed color printers and should buy more.
Business-aligned tagging	Focuses on accounting, business ownership, cost responsibility, or business criticality. Provides improved accounting for costs and value of IT assets to the overall business. Shifts the focus from an asset's operational cost to an asset's business value.	Our marketing department's promotional literature has increased sales revenue 10%. We should invest in more printing capabilities.

Consider the type of tagging that is required.

Resource tags generally fall into five categories functional, classification, accounting, partnership, and purpose.

Tag type	Examples	Description
Functional	app = catalogsearch1 tier = web webserver = apache env = prod, dev, staging	Categorize resources by their purpose within a workload, what environment they've been deployed to, or other functionality and operational details.
Classification	confidentiality = private SLA = 24hours	Classifies a resource by how it's used and what policies apply to it.
Accounting	department = finance program = business-initiative region =	Allows a resource to be associated with specific groups within an organization for billing purposes.

	northamerica	
Partnership	owner = jsmith contactalias = catsearchowners stakeholders = user1;user2;user3	Provides information about what people (outside of IT) are related or otherwise affected by the resource.
Purpose	businessprocess = support businessimpact = moderate revenueimpact = high	Aligns resources to business functions to better support investment decisions.

- **Consider starting with a few tags and then scaling out.** Your resource tagging approach can be simple or complex. Rather than identifying all the possible tags your organization needs, prototype using a few important or critical tags. Determine how effective the tagging is before adding more resource tags.
- **Consider using Azure policy to apply tags and enforce tagging rules and conventions**. Resource tagging is only effective if it's used consistently across an organization. You can use Azure policy to require that certain tags be added to new resources as they're provisioned. You can also define rules that reapply tags that have been removed.
- **Consider which resources require tagging.** Keep in mind that you don't need to enforce that a specific tag is present on all your resources. For example, you might decide that only mission-critical resources have the Impact tag. All non-tagged resources would then not be considered as mission critical.

> **Note**
>
> You'll need support from many different stakeholders to implement an effective resource tagging structure.

Next unit: Design for Azure policy

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure policy

3 minutes

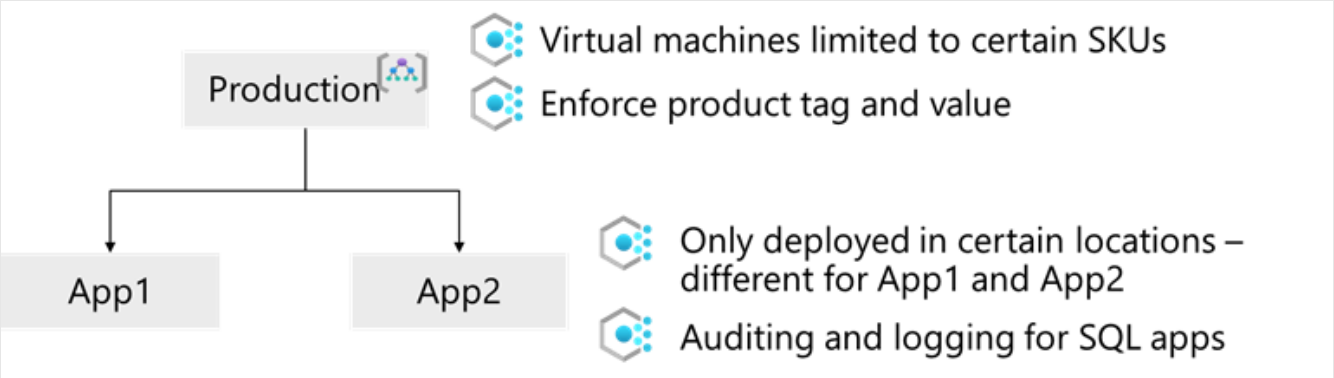
Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources. These policies enforce different rules over your resource configurations so that those configurations stay compliant with corporate standards.

Things to know about Azure policy

- Azure policy lets you define both individual policies and groups of related policies, called initiatives. Azure Policy comes with many built-in policy and initiative definitions.
- Azure policy lets you scope and enforce your policies at different levels in the organizational hierarchy.
- Azure policies are inherited down the hierarchy.
- Azure policy evaluates your resources and highlights resources that aren't compliant with the policies you've created.
- Azure policy can prevent noncompliant resources from being created.
- Azure Policy can automatically remediate noncompliant resources.
- Azure Policy evaluates all resources in Azure and Arc enabled resources (specific resource types hosted outside of Azure).
- Azure policy integrates with Azure DevOps by applying pre-deployment and post-deployment policies.

Things to consider when using Azure policy

Tailwind Traders is now ready to consider applying Azure policy to their apps. Some policies will be applied at the Production management group level. Other policies will be assigned at the app level.



- **Consider using the Azure policy compliance dashboard.** The Azure policy compliance dashboard provides an aggregated view to help evaluate the overall state of the environment. You can drill down to a per-resource, or per-policy level granularity. You can also use capabilities like bulk remediation for existing resources and automatic remediation for new resources, to resolve issues rapidly and effectively.
- **Consider when Azure policy evaluates resources**. Azure policy evaluates resources at specific times. It's important to understand when an evaluation is triggered. There may be a lag in identifying non-compliant resources. The following events or times will trigger an evaluation.
  - A resource has been created, deleted, or updated in scope with a policy assignment.
  - A policy or an initiative is newly assigned to a scope.
  - A policy or an initiative that's been assigned to a scope is updated.
  - The standard compliance evaluation cycle (happens once every 24 hours).
- **Consider what you'll do if a resource is non-compliant.** Organizations will vary in how they respond to non-compliant resources. Your strategy may be different depending on the resource. Here's some examples of what to do if a resource is non-compliant.
  - Deny a change to a resource.
  - Log changes to a resource.
  - Alter a resource before or after a change.
  - Deploy related compliant resources.
- **Consider when to automatically remediate non-compliant resources.** In some cases, Azure policy can automatically remediate noncompliant resources. Remediation is especially useful in resource tagging. Azure policy can tag resources and reapply tags that have been removed. For example, Azure policy can ensure all resources in a certain resource group should be tagged with the Location tag.

- **Consider how Azure policy is different from role-based access control (RBAC).** It's important not to confuse Azure Policy and Azure RBAC. Azure RBAC and Azure Policy should be used together to achieve full scope control.
    - You use Azure Policy to ensure that the resource state is compliant to your organization's business rules. Compliance doesn't depend on who made the change or who has permission to make changes. Azure Policy will evaluate the state of a resource, and act to ensure the resource stays compliant.
    - You use Azure RBAC to focus on user actions at different scopes. Azure RBAC manages who has access to Azure resources, what they can do with those resources, and what areas they can access. If actions need to be controlled, then use Azure RBAC. If an individual has access to complete an action, but the result is a non-compliant resource, Azure Policy still blocks the action.

Once you have determined your identity management solution, it's time to think about resource access. What resources should these identities be able to access? How will you enforce that access? How will you monitor and review the access?

A user's identity goes through several phases. Initially, the user will have no access. Access can then be granted through role-based access control and verified with Azure AD conditional access. Azure AD Identity Protection can be used to monitor the user's access. And then periodically Azure AD access reviews will confirm the access is still required.

Next unit: Design for role-based access control

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for role-based access control

3 minutes

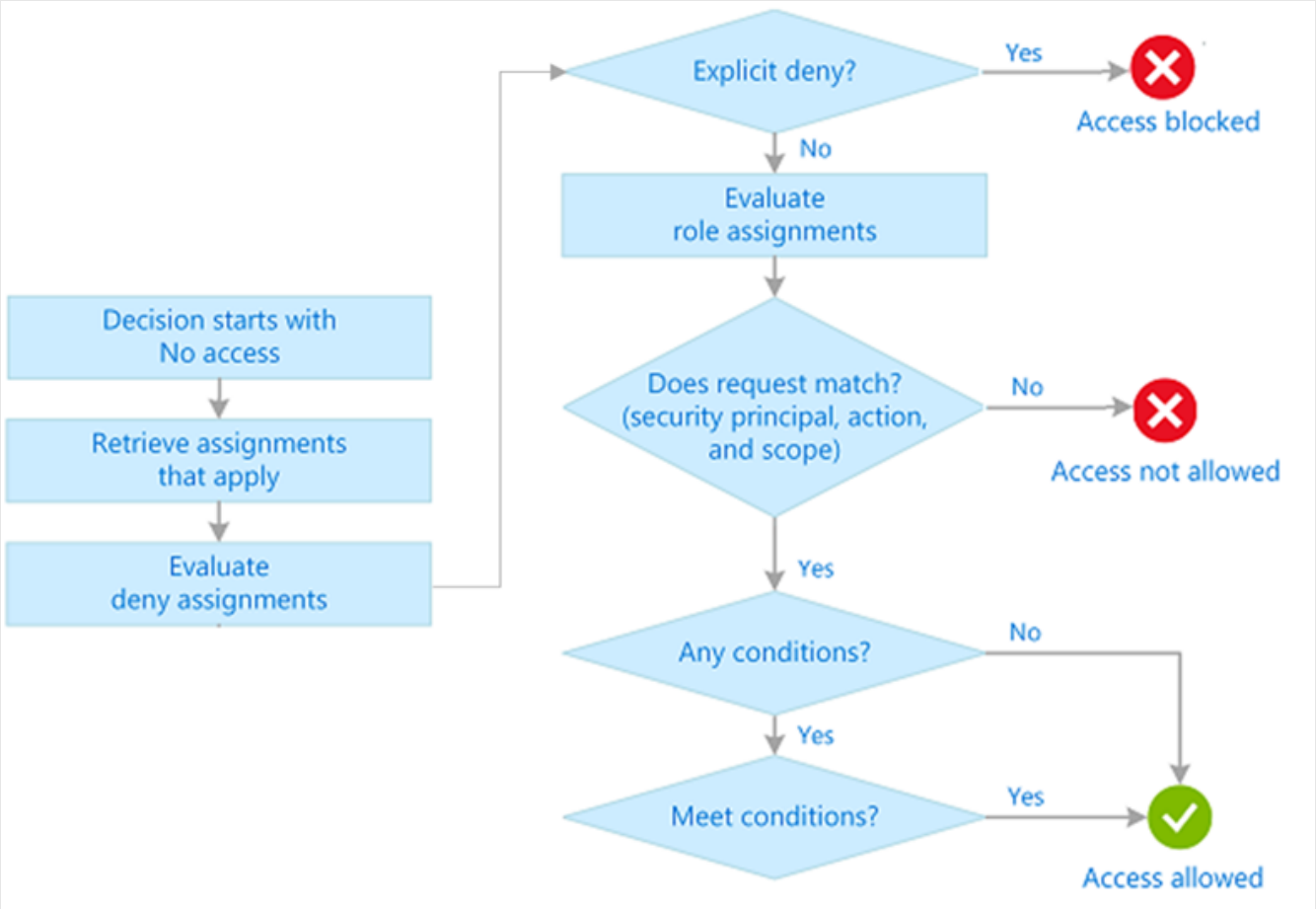
What is Azure role-based access control (RBAC)?

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 Tailwind Trader's development, engineering, and marketing teams. 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.

How does role-based access control work?

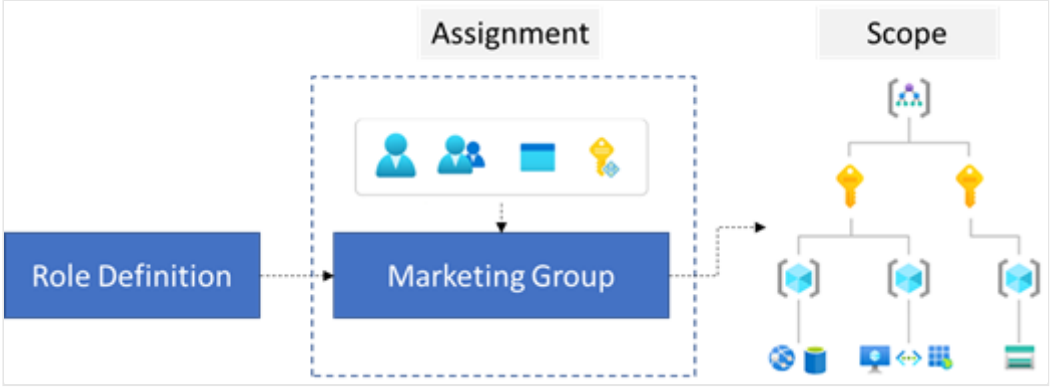
Azure RBAC evaluates each request for access. The evaluations will determine if access should be blocked, not allowed, or allowed.



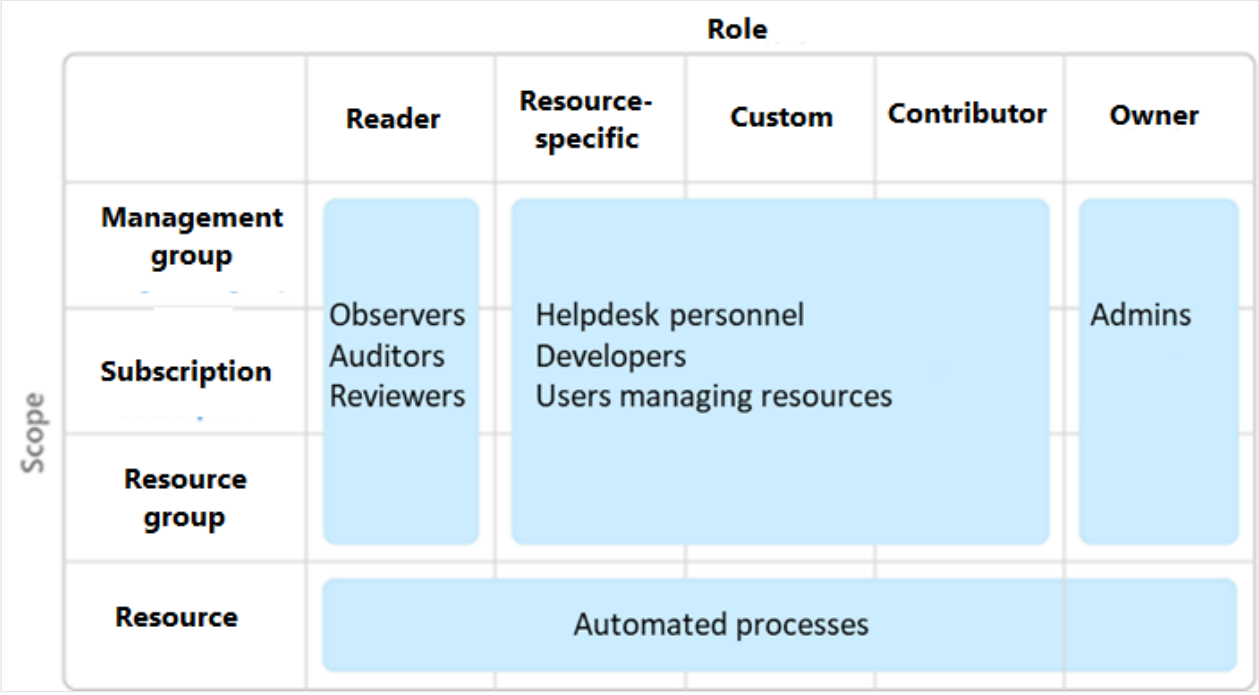
Things to consider when using role-based access control

- **Remember RBAC is an allow model**. An allow model means when you're assigned a role, Azure RBAC allows you to perform certain actions. For example, a role assignment could grant you read permissions to a resource group. To have write permissions the role would need to explicitly allow write access.

- **Assign at the highest scope level that meets the requirements.** Your first step is to accurately define the role definition and related permissions. Next assign roles to users, groups, and service principals. Lastly, scope the roles to management groups, subscriptions, resource groups, and resources. Be sure to assign at the highest scope level that meets the requirements.



- **Only grant users the access they need.** When planning your access control strategy, it's a best practice to grant users the least privilege to get their work done. This allows you to segregate duties within your team. By limiting roles and scopes, you limit what resources are at risk if the security principal is ever compromised. Creating a diagram like this, might help to explain Azure RBAC roles.



- **Assign roles to groups, not users**. To make role assignments more manageable, avoid assigning roles directly to users. Instead, assign roles to groups. Assigning roles to groups helps minimize the number of role assignments.

- **Know when to use Azure policies**. Azure policies focuses on resource properties. For example, during deployment, a policy can ensure users can only deploy certain virtual machines in a resource group. A combination of Azure policies and Azure RBAC can provide effective access control.

Area	Azure Policy	Role-based Access Control
Description	Ensure resources are compliant with a set of rules.	Authorization system to provide fine-grained access controls.
Focus	Focused on the properties of resources.	Focused on what resources the users can access.
Implementation	Specify a set of rules.	Assign roles and scopes.

Area	Azure Policy	Role-based Access Control
Default access	By default, rules are set to allow.	By default, all access is denied.

- **Know when to create a custom role.** Sometimes, the built-in roles don't grant the precise level of access you need. Custom roles allow you to define roles that meet the specific needs of your organization. Custom roles can be shared between subscriptions that trust the same Azure Active Directory.
- **Consider what happens if you have overlapping role assignments.** Azure RBAC is an additive model, so your effective permissions are the sum of your role assignments. Consider a user is granted the Contributor role at the subscription scope and the Reader role on a resource group. The sum of the Contributor permissions and the Reader permissions is effectively the Contributor role for the subscription. Therefore, in this case, the Reader role assignment has no impact.

Next unit: Design for Azure blueprints

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

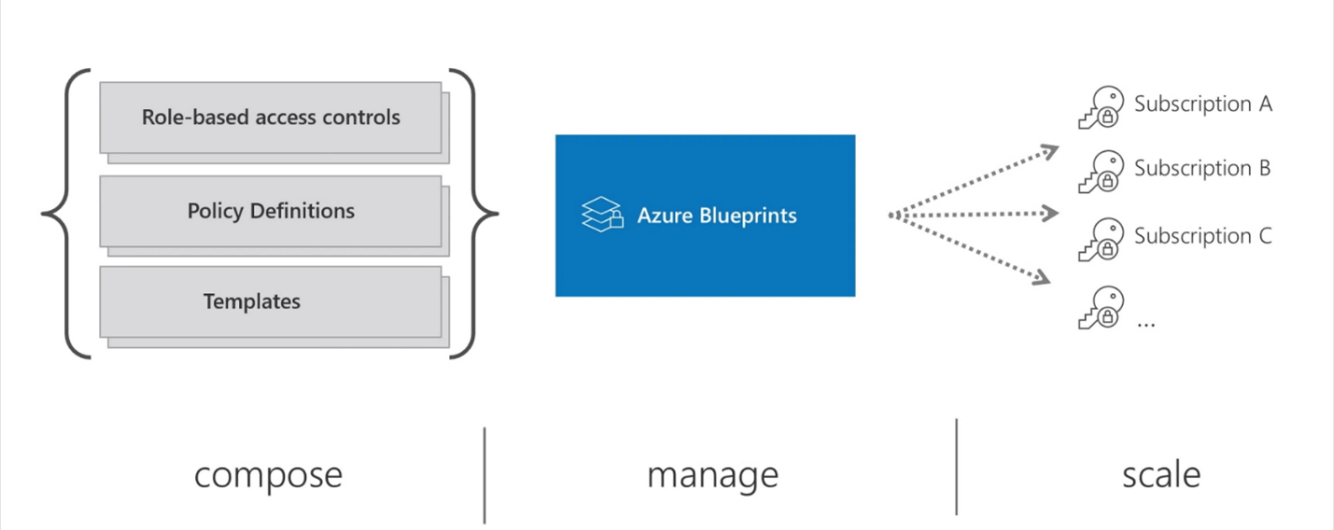
100 XP

Design for Azure blueprints

3 minutes

Azure Blueprints lets you define a repeatable set of governance tools and standard Azure resources that your organization requires. Azure Blueprints are used to scale governance practices throughout an organization.

A blueprint is a package related to the implementation of Azure cloud services, security, and design. A blueprint can be reused to maintain consistency and compliance.



With Azure Blueprints, the relationship between the blueprint definition (what should be deployed) and the blueprint assignment (what was deployed) is preserved. In other words, Azure creates a record that associates a resource with the blueprint that defines it. This connection helps you track and audit your deployments. Azure Blueprints orchestrates the deployment of various resource templates and other artifacts.

Resource	Hierarchy Options	Description
Resource Groups	Subscription	Create a new resource group for use by other artifacts within the blueprint. These placeholder resource groups enable you to organize resources exactly the way you want them structured and provides a scope limiter for included policy and role assignment artifacts and templates.

Resource	Hierarchy Options	Description
Azure Resource Manager Template	Subscription, Resource group	Templates, including nested and linked templates, are used to compose complex environments. Example environments: a SharePoint farm, Azure Automation State Configuration, or a Log Analytics workspace.
Policy Assignment	Subscription, Resource group	Allows assignment of a policy or initiative to the subscription the blueprint is assigned to. The policy or initiative must be within the scope of the blueprint definition location. If the policy or initiative has parameters, these parameters are assigned at creation of the blueprint or during blueprint assignment.
Role Assignment	Subscription, Resource group	Add an existing user or group to a built-in role to make sure the right people always have the right access to your resources. Role assignments can be defined for the entire subscription or nested to a specific resource group included in the blueprint.

How are Azure Blueprints different from Azure Policy

A policy is a default allow and explicit deny system focused on resource properties during deployment and for already existing resources. It supports cloud governance by validating those resources within a subscription adhere to requirements and standards.

A policy can be included as one of many artifacts in a blueprint definition. Including a policy in a blueprint enables the creation of the right pattern or design during assignment of the blueprint. The policy inclusion makes sure that only approved or expected changes can be made to the environment to protect ongoing compliance to the intent of the blueprint.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Knowledge check

3 minutes

Tailwind Traders is planning on making some significant changes to their governance solution. They have asked for your assistance with recommendations and questions. Here are the specific requirements.

- **Consistency across subscriptions.** It appears each subscription has different policies for the creation of virtual machines. The IT department would like to standardize the policies across the Azure subscriptions.
- **Ensure critical storage is highly available.** There are several critical applications that use storage. The IT department wants to ensure the storage is made highly available across regions.
- **Identify R&D costs.** The CTO wants to know how much a new project is costing. The costs are spread out across multiple departments.
- **ISO compliance.** Tailwind Traders wants to certify that it complies with the ISO 27001 standard. The standard will require resources groups, policy assignments, and templates.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. How can Tailwind Traders ensure policies are implemented across multiple subscriptions?

- ○ Add a resource tag that includes the required policy.
- ◉ Create a management group and place all the relevant subscriptions in the new management group.

  ✔ **Correct. A management group could include all the subscriptions. Then a policy could be scoped to the management group and applied to all the subscriptions.**
- ○ Add a resource group and place all the relevant subscriptions in it.

2. How can Tailwind Traders ensure applications use geo-redundancy to create highly available storage applications?

- ○ Add a resource tag to each storage account for geo-redundant

storage.

- ○ Add a geo-redundant resource group to contain all the storage accounts.
- ◉ Add an Azure policy that requires geo-redundant storage.

✔ **Correct. An Azure policy can enforce different rules over your resource configurations.**

3. How can Tailwind Traders report all the costs associated with a new product?

- ◉ Add a resource tag to identify which resources are used for the new product.

✔ **Correct. Resource tagging provides extra information, or metadata, about your resources. You could then run a cost report on all resources with that tag.**

- ○ Add a resource group and move all product assets into the group.
- ○ Add a spreadsheet and require each department to log their costs.

4. Which governance tool should Tailwind Traders use for the ISO 27001 requirements?

- ○ Management groups
- ◉ Azure blueprints.

✔ **Correct. Azure blueprints will deploy all the artifacts for ISO 27001 compliance.**

- ○ Resource groups.

Module complete:

Unlock achievement

How are we doing? ☆ ☆ ☆ ☆ ☆

Introduction

3 minutes

Meet Tailwind Traders

Tailwind Traders is a fictitious home improvement retailer. It operates retail hardware stores across the globe and online. The Tailwind Trader's CTO asks, "What is our identity solution?" At first, this seems like a simple question. But managing and protecting your corporate identities requires planning and careful design. In this module, we'll answer the questions:

- What identity providers does Azure offer?
- What identity protections are available?

Learning objectives

In this module, you'll learn how to:

- Design for identity and access management.
- Design for Azure Active Directory.
- Design for Azure AD B2B.
- Design for Azure AD B2C.
- Design for conditional access.
- Design for identity protection.
- Design for access reviews.
- Design for service principals for applications.
- Design for Azure Key Vault.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design authentication and authorization solutions.

- Recommend an identity solution.
- Recommend an access control solution for identities.
- Recommend an authorization solution.

Design Identities and Access for Applications

- Recommend a solution that securely stores passwords and secrets
- Recommend solutions to allow applications to access Azure resources
- Recommend a solution for integrating applications into Azure AD
- Recommend a user consent solution for applications

Prerequisites

- Working experience creating, assigning, and securing corporate identities.
- Conceptual knowledge of identity assignment solutions, role-based access control, and identity protection methods.

Next unit: Design for identity and access management

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for identity and access management

3 minutes

Azure Architects design Identity and access management (IAM) solutions. These solutions must work for all your users, apps, and devices. There are four basic guidelines for a strong IAM solution.

- **Unified identity management**. Manage all your identities and access to all your apps in a central location, whether they're in the cloud or on-premises, to improve visibility and control.
- **Seamless user experience**. Provide an easy, fast sign in experience to keep your users productive, reduce time managing passwords, and increase end-user productivity.
- **Secure adaptive access**. Protect access to resources and data using strong authentication and risk-based adaptive access policies without compromising user experience.
- **Simplified identity governance**. Control access to apps and data for all users and admins. Automated identity governance to ensure only authorized users have access.

Let's first focus on the identity solution. There are three basic choices.

If you need this	Use this
Provide identity and access management for employees in a cloud or hybrid environment.	Azure Active Directory (Azure AD)
Collaborate with guest users and external business partners like suppliers and vendors.	Azure AD Business to Business (B2B)

If you need this	Use this
Control how customers sign up, sign in, and manage their profiles when they use your applications.	Azure AD Business to Consumer (B2C)

Next unit: Design for Azure Active Directory

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

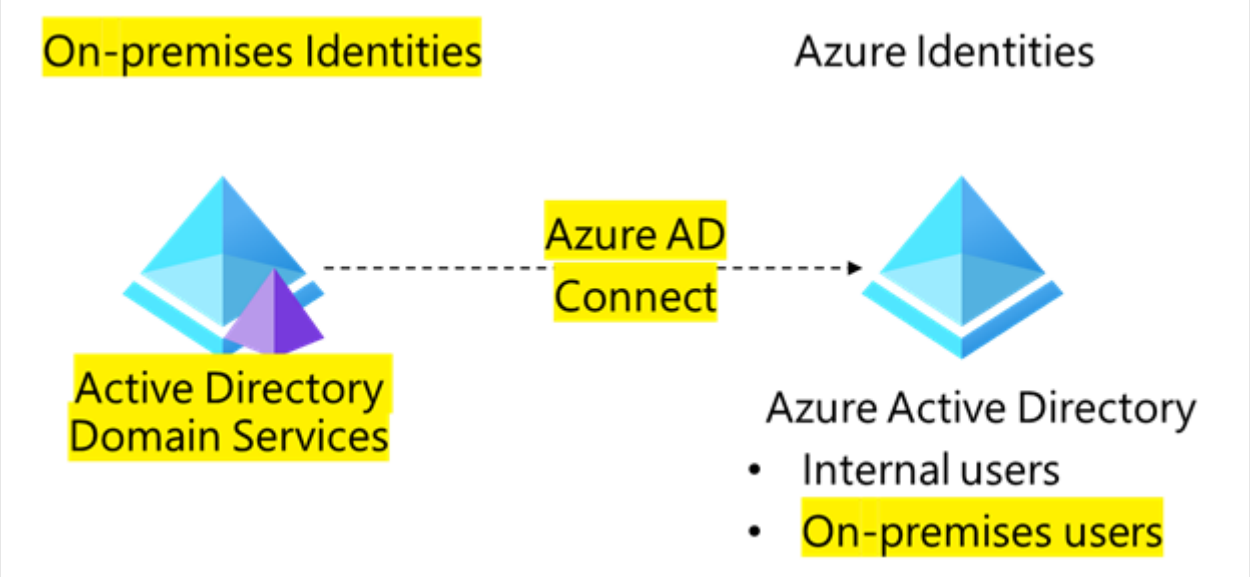
Design for Azure Active Directory

3 minutes

Let's begin with Azure Active Directory (Azure AD). Azure AD is the Azure solution for identity and access management. Azure AD is a multitenant, cloud-based directory, and identity management service. It combines core directory services, application access management, and identity protection into a single solution. Azure AD can be used in cloud or hybrid environments.

Cloud identity solution. You can use Azure AD as a cloud only solution for all your employee user accounts. Azure AD provides not only identity management but protection for those accounts. For example, role-based access control, conditional access, and access reviews. We'll cover those features, later in this module.

Hybrid identity solution. You can also use Azure AD in hybrid environments. Azure AD extends on-premises Active Directory to the cloud. With Azure AD Connect or Azure AD Connect cloud sync, you can bring on-premises identities into Azure AD. Once the on-premises accounts are in Azure AD they will get the benefits of easy management and identity protection.



Best practices with Azure ID identity management

- **Centralize identity management**. In a hybrid identity scenario, we recommend that you integrate your on-premises and cloud directories. Integration enables your IT team to

manage accounts from one location, whenever an account is created. Integration also helps your users be more productive by providing a common identity for accessing both cloud and on-premises resources.

- **Establish a single Azure AD instance**. Consistency and a single authoritative source will increase clarity and reduce security risks from human errors and configuration complexity. Designate a single Azure AD directory as the authoritative source for corporate and organizational accounts.
- **Don't synchronize accounts to Azure AD that have high privileges in your existing Active Directory instance.** By default, Azure AD Connect filters out these high privileged accounts. This configuration mitigates the risk of adversaries pivoting from cloud to on-premises assets (which could create a major incident).
- **Turn on password hash synchronization**. Password hash synchronization is a feature used to sync user password hashes from an on-premises Active Directory instance to a cloud-based Azure AD instance. This sync helps to protect against leaked credentials being replayed from previous sign-ins.
- **Enable single sign-on (SSO).** SSO reduces the need for multiple passwords. Multiple passwords increase the likelihood of users reusing passwords or using weak passwords. With SSO, users provide their primary work or school account for their domain-joined devices and company resources. Their application access can be automatically provisioned (or deprovisioned) based on their organization group memberships and their status as an employee.

> **Note**
>
> Organizations that don't integrate their on-premises identity with their cloud identity can have more overhead in managing accounts. This overhead increases the likelihood of mistakes and security breaches.

Next unit: Design for Azure Active Directory Business to Business

Continue

100 XP

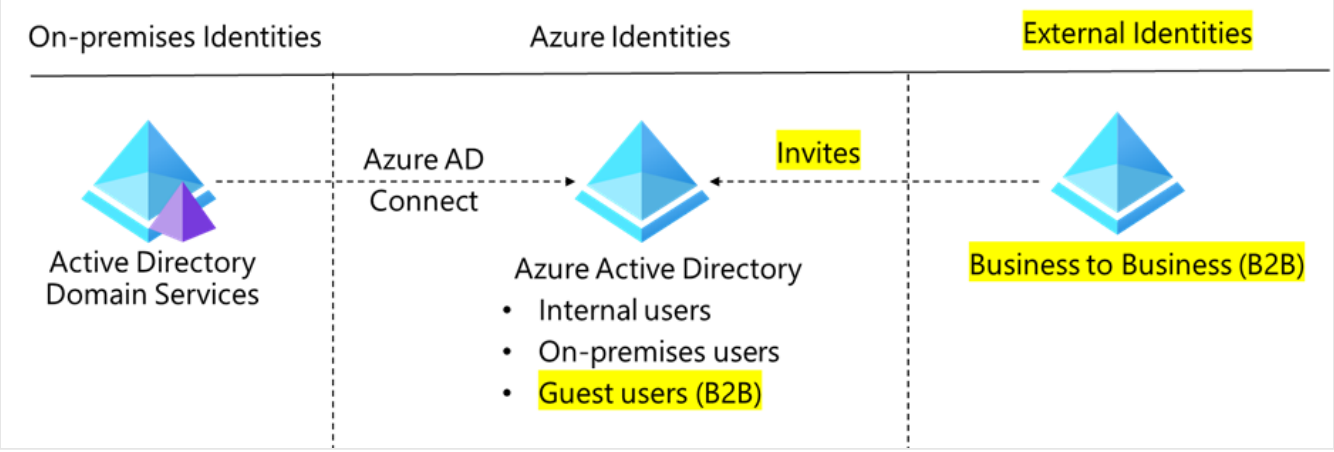
Design for Azure Active Directory Business to Business

3 minutes

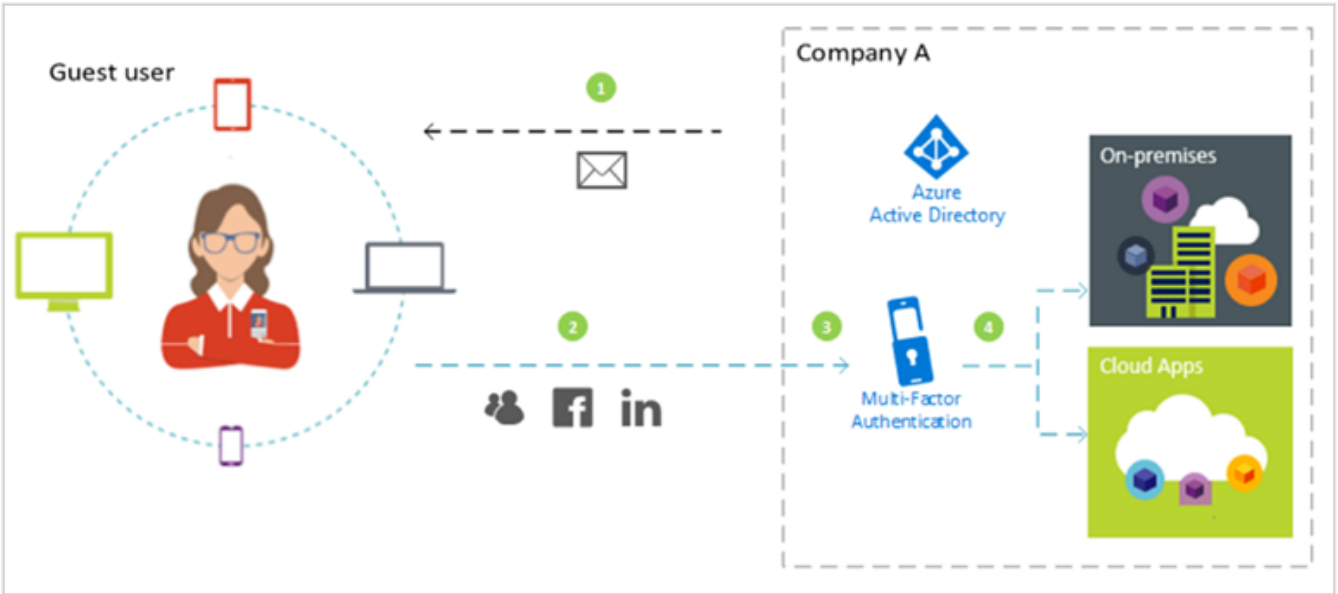
Every organization needs to work with external users. Azure AD Business to Business (B2B) is a feature of Azure AD that enables you to securely collaborate with external partners. Your partner users are invited as guest users. You remain in control of what they have access to, and for how long.

With Azure AD B2B, the partner uses their own identity management solution. Azure AD is not required. You don't need to manage external accounts or passwords. You don't need to sync accounts or manage account lifecycles. Guest users sign in to your apps and services with their own work, school, or social identities.

With Azure AD B2B, external users can use their identities to collaborate with your organization. Their identities are managed by the partner themselves, or by another external identity provider on their behalf.



The following steps show how Azure AD B2B lets you collaborate with external partner users. The numbers in the diagram are explained after the diagram.



1. You invite external users as guest users to your directory. For example, you fill in a form with your guest user's details and a custom invitation message.

2. Guest user receives an invitation via email. The first time the link is used, the user is asked for consent. The user must accept the permissions needed by Azure AD B2B before they can gain access.

3. If you've enabled multifactor authentication (MFA), the user provides these extra details for their account. When MFA is configured, the user must enter a verification code sent to their mobile device before they're granted access.

4. Your guest user is then forwarded to the access panel page. This page presents all the applications and services you've shared with them. These applications and services can be cloud-based, or on-premises.

Best practices for Azure AD B2B

- **Designate an application owner to manage guest users.** It's a good idea to delegate guest user access to application owners. Application owners are in the best position to decide who should be given access to a particular application.

- **Use conditional access policies to intelligently grant or deny access**. Conditional access policies use factors that aren't credential-based. For example, you can make it mandatory for users to be on specific device platforms, such as Android or Windows. Another example, you can block users if they don't meet the required location criteria.

- **Enable MFA**. You can use conditional access policies to require a MFA process, before they can access applications. This action ensures that everyone who uses the application must pass an additional authentication challenge before accessing it.

- **Integrate with identity providers**. Azure AD supports external identity providers like Facebook, Microsoft accounts, Google, or enterprise identity providers. You can set up federation with identity providers so your external users can sign in with their existing social or enterprise accounts instead of creating a new account just for your application.
- **Create a self-service sign-up user flow**. With a self-service sign-up user flow, you can create a sign-up experience for external users who want to access your apps. As part of the sign-up flow, you can provide options for different social or enterprise identity providers, and collect information about the user. You can also Customize the onboarding experience for B2B guest users.

Next unit: Design for Azure Active Directory Business to Customer

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Active Directory Business to Customer

3 minutes

Azure AD B2C is a type of Azure AD tenant that you use to manage customer identities and their access to your applications.

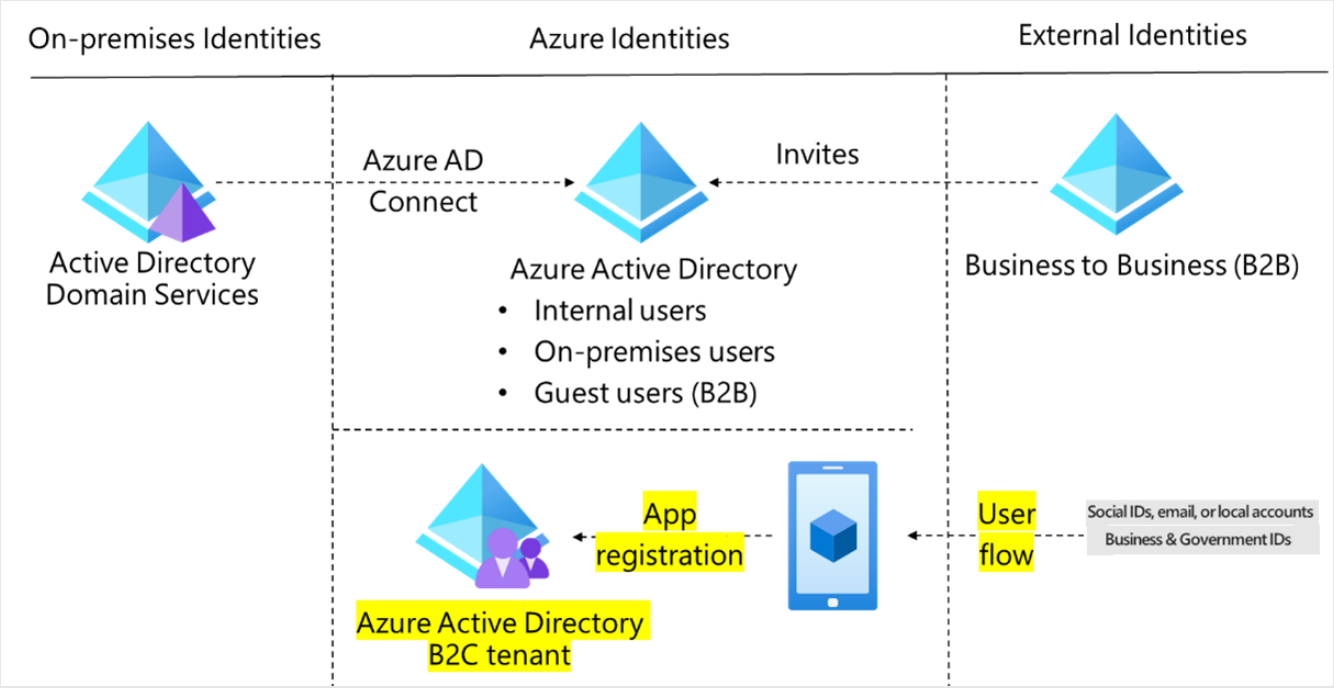
Use Azure AD B2C to:

- Securely authenticate your customers using their preferred identity providers.
- Capture sign in, preference, and conversion data for customers.
- Store custom attributes about customers to be leveraged by your applications
- Provide branded registration and sign in experiences.
- Separate the organization account from the customer account.

How does Azure B2C work?

Azure AD B2C requires an Azure AD tenant. This tenant isn't the same your organization's Azure AD tenant. You use an Azure AD tenant to represent an organization. Your Azure AD B2C tenant represents the identities that are used for customer applications.

With your Azure AD B2C tenant in place, you must register your app. You use user flows to manage things like sign-ins and sign ups. Your Azure AD B2C tenant lets you create multiple types of user flows.



Best practices for Business to Customer

- **Configure user journeys by using policies**. A user journey is the path that you want people to take in your application to achieve their goal. For example, a user might want to make a new account, or update their profile. Azure AD B2C comes with preconfigured policies called user flows. You can reuse the same user flows across different applications. Reusing user flows creates a consistent user journey across all applications.

- **Use identity providers to let users sign in using their social identities**. There's a long list of identity providers and more are being added. Social providers include Amazon, Azure AD, Facebook, LinkedIn, Twitter, and Microsoft accounts.

- **Customize your user interface**. You can customize the pages in your user flow. Write your own HTML and CSS or use built-in templates called page layout templates.

- **Integrate with external user stores**. Azure AD B2C provides a directory that can hold 100 custom attributes per user. However, you can also integrate with external systems. For example, use Azure AD B2C for authentication, but delegate to an external customer relationship management (CRM) or customer loyalty database as the source of truth for customer data.

- **Third-party identity verification and proofing**. Use Azure AD B2C to facilitate identity verification and proofing by collecting user data, then passing it to a third-party system to perform validation, trust scoring, and approval for user account creation.

> **Tip**

Take a few minutes to review the **WoodGrove Groceries tutorial.** WoodGrove Groceries is a live web application created by Microsoft to demonstrate several Azure AD B2C features.

With some basic knowledge on identity solutions, let's review our design choices.

Feature	Azure AD B2B	Azure AD B2C
Purpose	Collaborating with business partners from external organizations like suppliers, partners, vendors. Users appear as guest users in your directory. These users may or may not have managed IT.	Customers of your product. These users are managed in a separate Azure AD directory / tenant.
Users	Partner users acting on behalf of their company or employees of the company	Customers acting as themselves.
Profiles	Managed through access reviews, email verification, or access/deny lists.	Users manage their own profiles.
Discoverability	Partner users are discoverable and can find other users from their organization.	Customers are invisible to other users. Privacy and content are enforced.
Identity providers supported	External users can collaborate using work accounts, school accounts, any email address, SAML and WS-Fed based identity providers, Gmail, and Facebook.	Consumer users with local application accounts (any email address or user name), various supported social identities, and users with corporate and government-issued identities via SAML/WS-Fed based identity provider federation.
External user management	External users are managed in the same directory as employees but are typically annotated as guest users. Guest users can be managed the same way as employees, added to the same groups, and so on.	External users are managed in the Azure AD B2C directory. They're managed separately from the organization's employee and partner directory (if any).

Feature	Azure AD B2B	Azure AD B2C
Branding	Host/inviting organization's brand is used.	Fully customizable branding per application or organization.

Important

Take a few minutes to decide if Azure B2B or Azure B2C would be required by your organization. Write down a few thoughts on how these options would be used.

Next unit: Design for conditional access

Continue >

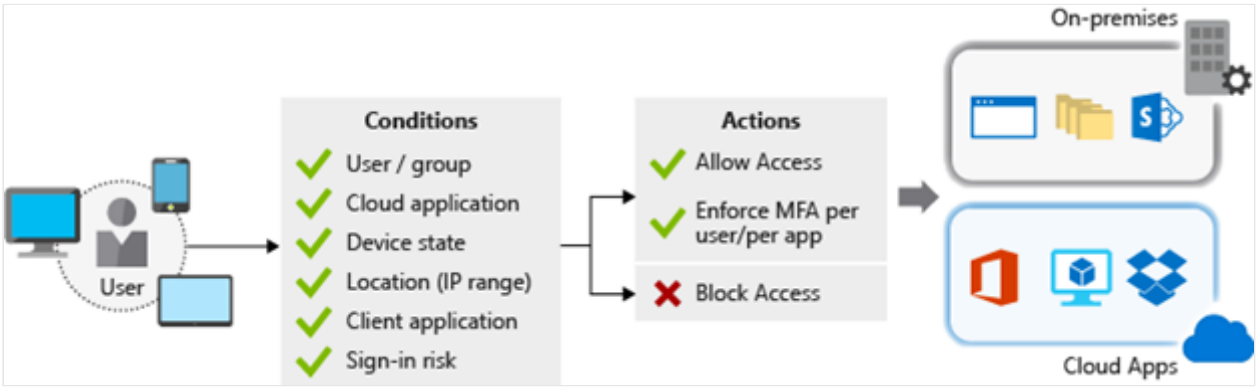
How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for conditional access

3 minutes

Conditional Access is a tool that Azure Active Directory uses to allow (or deny) access to resources. During sign-in, Conditional Access examines who the user is, where the user is, and from what device the user is requesting access. Based on these signals Conditional Access can allow access, enforce multifactor authentication, or deny access.



Here are some example conditional access situations.

- Require multifactor authentication (MFA). MFA can be used to provide a secondary authentication for accessing certain apps. MFA can be selectively applied to certain users, like administrators, or just users coming from external networks.
- Require access to services only through approved client applications. For example, only allow users to access Office 365 services from a mobile device if they use approved client apps, like the Outlook mobile app.
- Require users to access applications only from managed devices. A managed device is a device that meets your standards for security and compliance.
- Block access from untrusted sources, such as access from unknown or unexpected locations.

Things to consider when using conditional access

- **Use for enabling multifactor authentication for more granular control**. Conditional Access provides a more granular multifactor authentication experience for users. For example, a user might not be challenged for second authentication factor if they're at a

known location. However, they might be challenged for a second authentication factor if they're at an unexpected location.

- **Test by using report-only mode**. Report-only mode allows administrators to evaluate the impact of Conditional Access policies before enabling them in their environment. Report-only mode can help predict the number and names of users affected by common deployment initiatives. Use report-only mode to test blocking legacy authentication, requiring MFA, and implementing sign-in risk policies.

- **Exclude geographic areas from which you never expect a sign-in**. Azure Active Directory allows you to create named locations. Create a named location that includes all the geographic areas from which you would never expect a sign-in to occur. Then create a policy for all apps that blocks sign-in from that named location. Be sure to exempt your administrators from this policy.

- **Require managed devices**. The proliferation of supported devices to access your cloud resources helps to improve the productivity of your users. You probably don't want certain resources in your environment to be accessed by devices with an unknown protection level. For those resources, require that users can only access them using a managed device.

- **Require approved client applications.** Employees use their mobile devices for both personal and work tasks. In these scenarios, you must decide whether to manage the entire device or just the data on it. If managing only data and access, you can require only approved cloud apps. This can help to protect your corporate data.

- **Respond to potentially compromised accounts.** Three default policies can be enabled: require all users to register for MFA, require a password change for users who are high-risk, and require MFA for users with medium or high sign-in risk.

- **Block access.** Blocking access overrides all other assignments for a user and has the power to block your entire organization from signing on to your tenant. It can be used, for example, when you're migrating an app to Azure AD, but you aren't ready for anyone to sign-in yet. You can also block certain network locations from accessing your cloud apps or block apps using legacy authentication from accessing your tenant resources.

- **Block legacy authentication protocols**. Attackers exploit weaknesses in older protocols every day, particularly for password spray attacks. Configure Conditional Access to block legacy protocols.

- **Use the What If** tool. The What If tool helps you plan and troubleshoot your Conditional Access policies. The What If tool enables you to test your proposed Conditional Access policies before you implement them.

Note

To use Conditional Access, you need an Azure AD Premium P1 or P2 license. If you have a Microsoft 365 Business Premium license, you also have access to Conditional Access features.

Next unit: Design for identity protection

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for identity protection

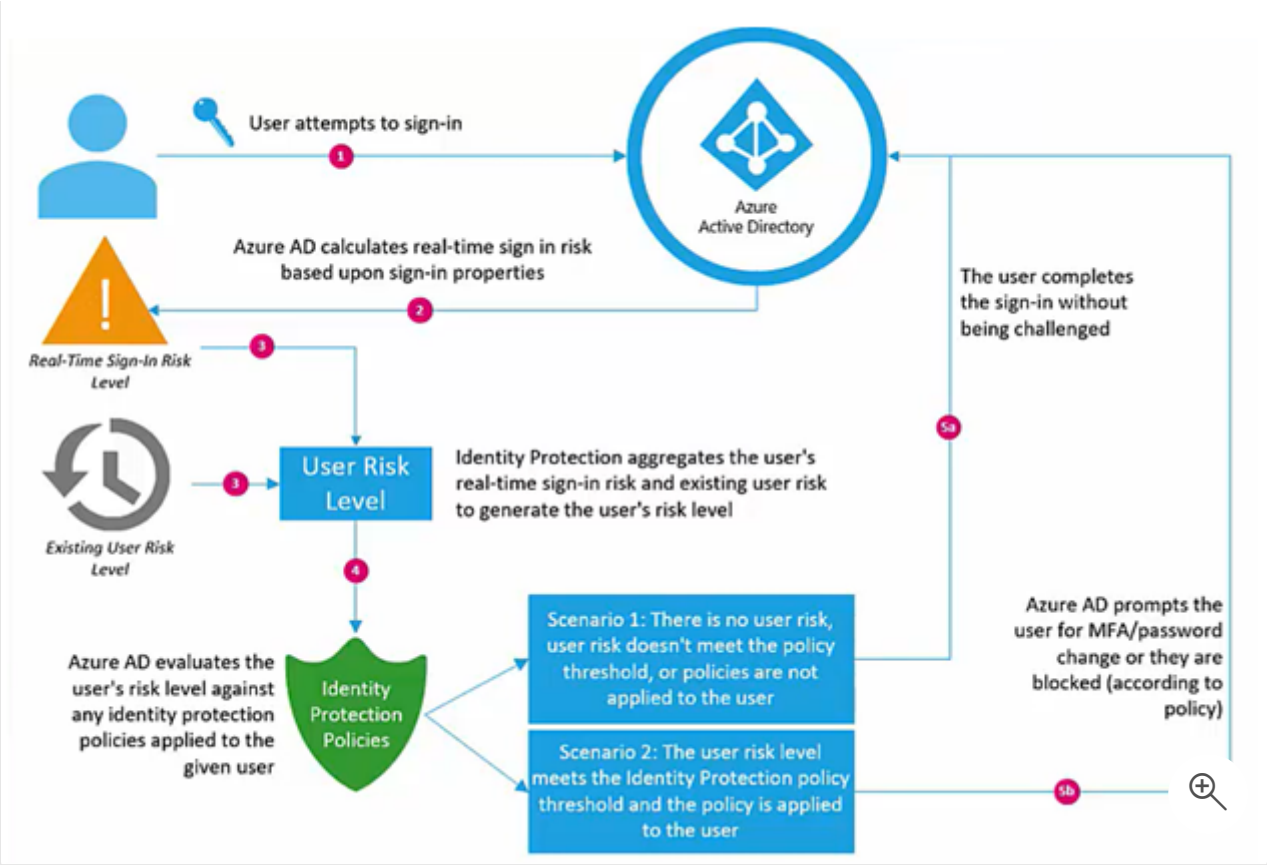
3 minutes

Identity Protection is a tool that allows organizations to accomplish three key tasks:

- Automate the detection and remediation of identity-based risks.
- Investigate risks using data in the portal.
- Export risk detection data to other tools.

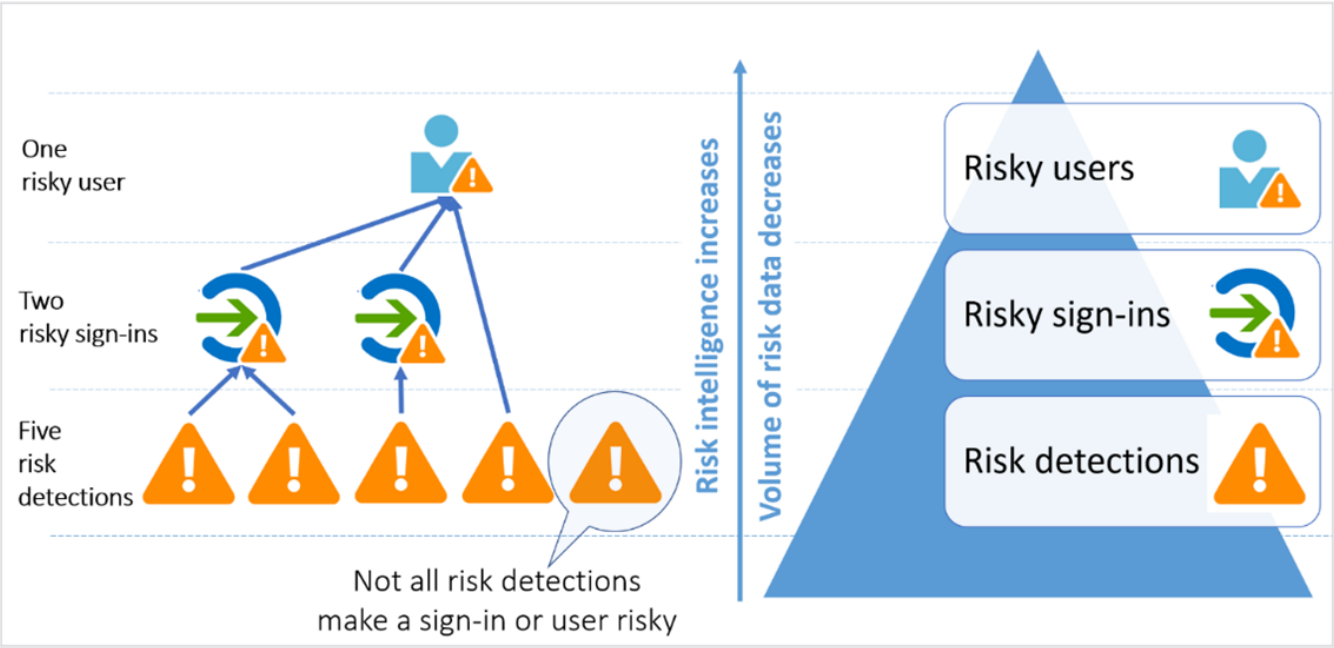
The signals generated by and fed to Identity Protection, can be exported to other tools. For example, Conditional Access can make decisions based on your organization's policies. You could feed information to a security information and event management (SIEM) tool for further investigation.

Here's an example where a user attempts to sign in to Azure Active Directory. Azure AD calculates real-time sign in risk based on sign in properties. Identity protection then aggregates the user's risk. If the risk level meets the Identity Protection policy threshold the user may be blocked or challenged by MFA. If the user risk level is acceptable, they're granted access.



Risk policies

Risk policy detections in Azure AD Identity Protection include any identified suspicious actions related to user accounts in the directory. There are two risk policies that are evaluated: user risk and sign in risk.



User risk policies

A user risk represents the probability that a given identity or account is compromised. For example, the user's valid credentials have been leaked. These risks are calculated offline using Microsoft's internal and external threat intelligence sources. Here are some user risks that can be identified.

- **Leaked credentials.** Microsoft checks for leaked credentials from the dark web, paste sites, or other sources. These leaked credentials are checked against Azure AD users' current valid credentials for valid matches.
- **Azure AD threat intelligence.** This risk detection type indicates user activity that is unusual for the given user or is consistent with known attack patterns.

> **Tip**
>
> Microsoft's recommendation is to set the user risk policy threshold to **High.**

Sign-in risk policies

A sign-in risk represents the probability that a given sign-in (authentication request) isn't authorized by the identity owner. Sign-in risk can be calculated in real-time or calculated offline. Here are some sign-in risks that can be identified.

- **Anonymous IP address**. This risk detection type indicates sign-ins from an anonymous IP address. For example, a Tor browser or anonymized VPNs.
- **Atypical travel**. This risk detection type identifies two sign-ins originating from geographically distant location. Given past behavior, at least one of the locations may also be atypical for the user.
- **Malware linked IP address**. This risk detection type indicates sign-ins from IP addresses infected with malware. This malware is known to actively communicate with a bot server.
- **Password spray**. This risk detection is triggered when a password spray attack has been performed. Password spray is one of the most popular attacks. Bad actors try to defeat lockout and detection by trying many users against one password.

> **Tip**
>
> Microsoft's recommendation is to set the sign-in risk policy to **Medium and above** and allow self-remediation options. Self-remediation options, like password change and multifactor authentication, will have less impact than blocking users.

100 XP

Design for access reviews

3 minutes

Over the time a user is employed by a company they may have several positions.

The Tailwind Traders CTO asks,

- As new employees join, how do we ensure they have the access they need to be productive?
- As users switch teams or leave the company, how do we make sure that their old access is removed?

Determine the purpose of the access review

You're considering using Azure Active Directory access reviews to address the CTO's concerns. An access review is a planned review of the access needs, rights, and history of user access. Access reviews mitigate risk by protecting, monitoring, and auditing access to critical assets.

Access reviews help ensure that the right people have the right access to the right resources. For example, access reviews could be used to review:

- User access to applications integrated with Azure AD for single sign-on (such as SaaS, line-of-business).
- Group memberships (synchronized to Azure AD, or created in Azure AD or Microsoft 365, including Microsoft Teams).
- Access Packages that group resources (groups, apps, and sites) into a single package to manage access.
- Azure AD roles and Azure Resource roles as defined in Privileged Identity Management (PIM).

Determine who will conduct the reviews

Access reviews are only as good as the person doing the reviewing. Selecting good reviewers is critical to your success. The creator of the access review decides who will conduct the review. This setting can't be changed once the review is started. Reviewers are represented by three personas:

- Resource Owners, who are the business owners of the resource.
- A set of individually selected delegates, as selected by the access reviews administrator.
- End users who will each self-attest to their need for continued access.

When creating an Access Review, administrators can choose one or more reviewers. All reviewers can start and carry out a review, choosing to grant users continued access to a resource or removing them.

Create an access review plan

Before implementing your access reviews, you should plan the types of reviews relevant to your organization. To do so, you'll need to make business decisions about what you want to review and the actions to take based on those reviews.

For example, here's an access review plan for Microsoft Dynamics.

Component	Value
Resources to review	Access to Microsoft Dynamics
Review frequency	Monthly
Who conducts the review	Dynamics business group program managers
Notification	Email 24 hours prior to review to alias Dynamics-PMs Include encouraging custom message to reviewers to secure their buy-in
Timeline	48 hours from notification

Component	Value
Automatic actions	Remove access from any account that has no interactive sign-in within 90 days by removing the user from the security group dynamics-access. Perform actions if not reviewed within timeline.
Manual actions	Reviewers may perform removals approval prior to automated action if desired.
Communications	Send internal (member) users who are removed an email explaining they're removed and how to regain access.

Next unit: Design service principals for applications

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design service principals for applications

3 minutes

Design managed identities

Azure managed identity is a feature of Azure Active Directory (Azure AD) that you can use free of charge. This feature automatically creates identities to allow apps to authenticate with Azure resources and services.

Tailwind is planning on moving applications from on-premises servers to Azure-hosted virtual machines (VMs). Now that you host the applications on VMs in Azure, you can use managed identities.

In this unit, you'll explore the managed identity feature. You'll learn how it works and what resources you can access in Azure.

What are managed identities in Azure?

A common challenge for developers is the management of secrets and credentials used to secure communication between different components making up a solution. Managed identities eliminate the need for developers to manage credentials. Managed identities provide an identity for applications to use when connecting to resources that support Azure Active Directory (Azure AD) authentication. Applications may use the managed identity to obtain Azure AD tokens. For example, an application may use a managed identity to access resources like Azure Key Vault where developers can store credentials in a secure manner or to access storage accounts.

A managed identity combines Azure AD authentication and Azure role-based access control (RBAC).

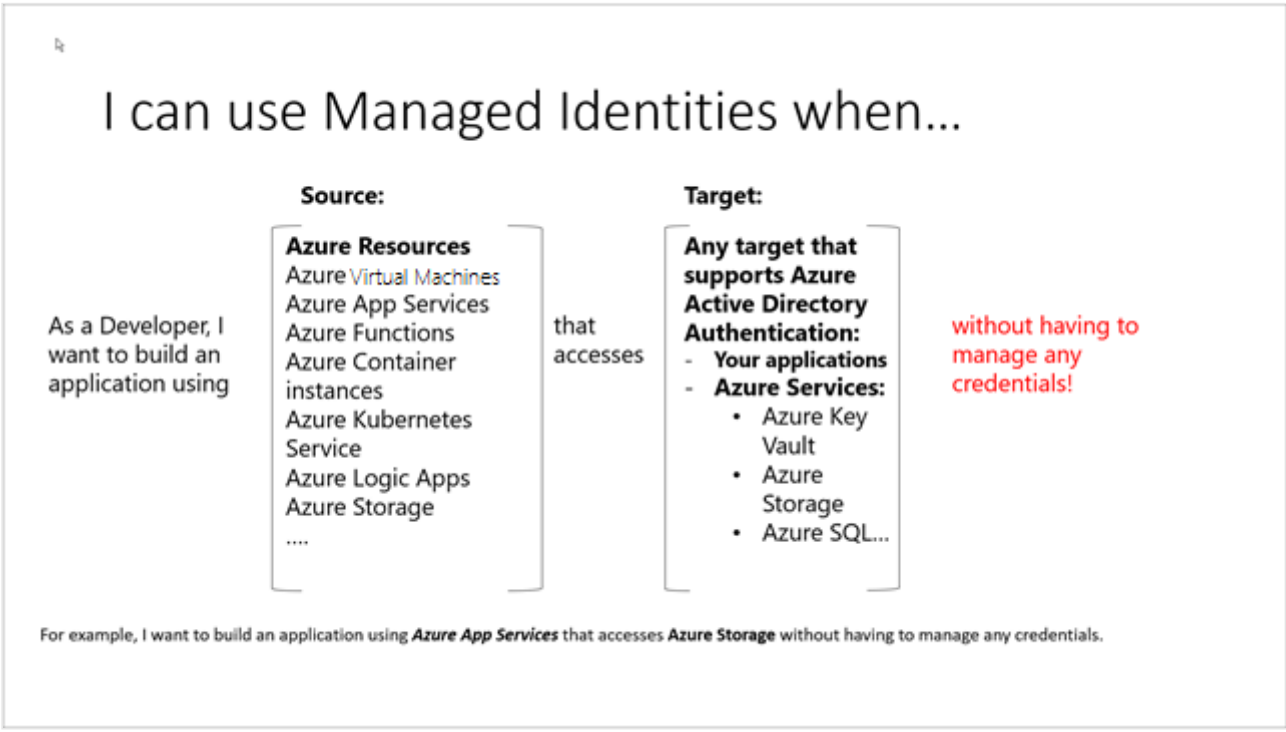
When you use managed identities, you don't need to rotate credentials or worry about expiring certifications. Azure handles credential rotation and expiration in the background. To configure an application to use a managed identity, you use the provided token to call the service.

When to use managed identities

When you work with managed identities, you should be familiar some common terms:

- Client ID: A unique ID that's linked to the Azure AD application and service principal that was created when you provisioned the identity.
- Object ID: The service principal object of the managed identity.
- Azure Instance Metadata Service: A REST API that's enabled when Azure Resource Manager provisions a VM. The endpoint is accessible only from within the VM.

Managed identities are available in all editions of Azure AD, including the Free edition included with an Azure subscription. Using it in App Service has no extra cost and requires no configuration, and it can be enabled or disabled on an app at any time.



Resources that support system assigned managed identities allow you to:

- Enable or disable managed identities at the resource level.
- Use RBAC roles to grant permissions.
- Review create, read, update, delete (CRUD) operations in Azure Activity logs.
- Review sign-in activity in Azure AD sign-in logs.

There are two types of managed identities:

- **System-assigned** Some Azure services allow you to enable a managed identity directly on a service instance. When you enable a system-assigned managed identity an identity is created in Azure AD that is tied to the lifecycle of that service instance. So when the resource is deleted, Azure automatically deletes the identity for you. By design, only that Azure resource can use this identity to request tokens from Azure AD.
- **User-assigned** You may also create a managed identity as a standalone Azure resource. You can create a user-assigned managed identity and assign it to one or more instances

of an Azure service. In the case of user-assigned managed identities, the identity is managed separately from the resources that use it.

When to use system assigned managed identity:

- Workloads that are contained within a single Azure resource
- Workloads for which you need independent identities.

When to use User-assigned managed identity:

- Workloads that run on multiple resources and which can share a single identity.
- Workloads that need pre-authorization to a secure resource as part of a provisioning flow.
- Workloads where resources are recycled frequently, but permissions should stay consistent.

You can add managed identities to virtual machines (VMs) in Azure. You decide to run your stock-tracking application inside a VM that has an assigned managed identity. This setup will allow the app to use an Azure key vault to authenticate without having to store a username and password in code.

Now that your company has migrated your VM from on-premises to Azure, you can remove the hard-coded authentication details from the application code. You want to use the more secure managed identity token for access to Azure resources.

Vault authentication with managed identities for Azure resources

Your application requires service passwords, connection strings, and other secret configuration values to do its job. Storing and handling secret values is risky, and every usage introduces the possibility of leakage. Azure Key Vault, in combination with managed identities for Azure resources, enables your Azure web app to access secret configuration values easily and securely without needing to store any secrets in your source control or configuration.

Azure Key Vault uses Azure Active Directory (Azure AD) to authenticate users and apps that try to access a vault. To grant your web app access to the vault, you first need to register your app with Azure Active Directory. Registering creates an identity for the app. After the app has an identity, you can assign vault permissions to it.

Apps and users authenticate to Key Vault using an Azure AD authentication token. Getting a token from Azure AD requires a secret or certificate because anyone with a token could use the app identity to access all the secrets in the vault. To access resources that are secured by an Azure AD tenant, the entity that requires access must be represented by a security

principal. This requirement is true for both users (user principal) and applications (service principal). The security principal defines the access policy and permissions for the user/application in the Azure AD tenant. This enables core features such as authentication of the user/application during sign-in, and authorization during resource access.

Select application service principals

There are three types of service principal:

- Application - The type of service principal is the local representation, or application instance, of a global application object in a single tenant or directory. In this case, a service principal is a concrete instance created from the application object and inherits certain properties from that application object. A service principal is created in each tenant where the application is used and references the globally unique app object. The service principal object defines what the app can do in the specific tenant, who can access the app, and what resources the app can access. When an application is given permission to access resources in a tenant (upon registration or consent), a service principal object is created. When you register an application using the Azure portal, a service principal is created automatically. You can also create service principal objects in a tenant using Azure PowerShell, Azure CLI, Microsoft Graph, and other tools.
- Managed identity - This type of service principal is used to represent a managed identity. Managed identities eliminate the need for developers to manage credentials. Managed identities provide an identity for applications to use when connecting to resources that support Azure AD authentication. When a managed identity is enabled, a service principal representing that managed identity is created in your tenant. Service principals representing managed identities can be granted access and permissions, but cannot be updated or modified directly.
- Legacy - This type of service principal represents a legacy app, which is an app created before app registrations were introduced or an app created through legacy experiences. A legacy service principal can have credentials, service principal names, reply URLs, and other properties that an authorized user can edit, but does not have an associated app registration. The service principal can only be used in the tenant where it was created.

Relationship between application objects and service principals

The application object is the global representation of your application for use across all tenants, and the service principal is the local representation for use in a specific tenant. The application object serves as the template from which common and default properties are derived for use in creating corresponding service principal objects.

An application object has:

- A 1:1 relationship with the software application
- A 1:many relationship with its corresponding service principal object(s).

A service principal must be created in each tenant where the application is used, enabling it to establish an identity for sign-in and/or access to resources being secured by the tenant. A single-tenant application has only one service principal (in its home tenant), created and consented for use during application registration. A multi-tenant application also has a service principal created in each tenant where a user from that tenant has consented to its use.

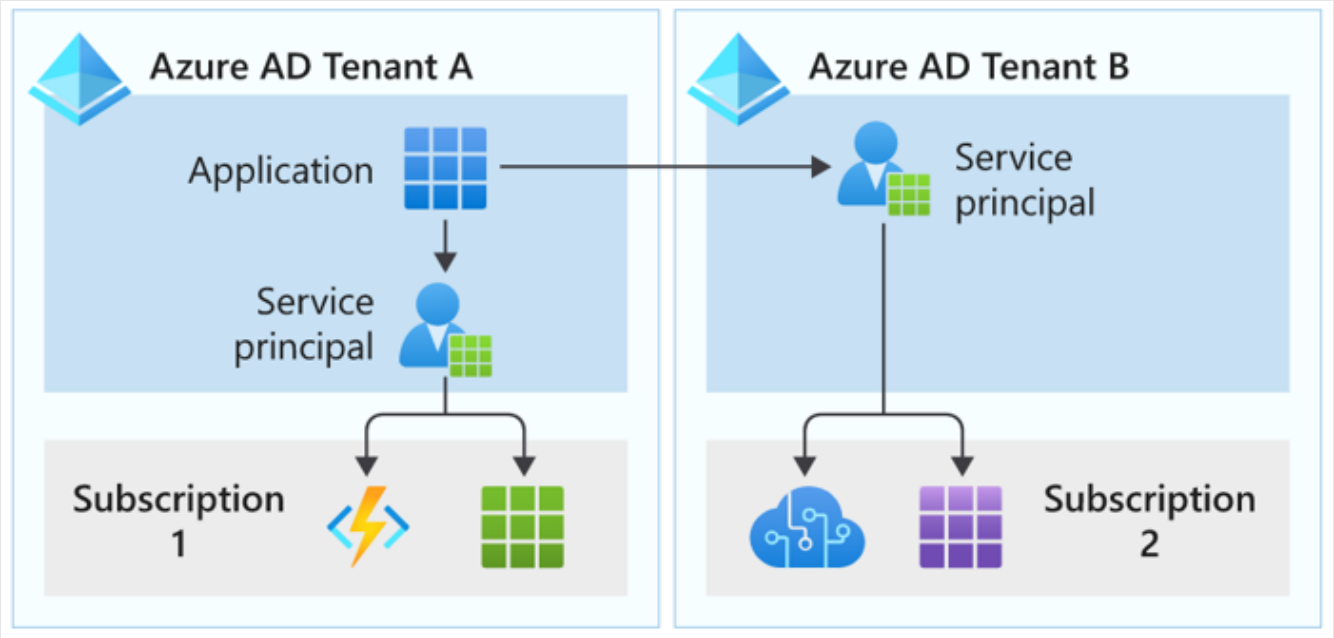
Applications represented in Azure AD

There are two representations of applications in Azure AD:

- Application objects - Although there are exceptions, application objects can be considered the definition of an application.
- Service principals - Can be considered an instance of an application. Service principals generally reference an application object, and one application object can be referenced by multiple service principals across directories.

Application objects describe the application to Azure AD and can be considered the definition of the application, allowing the service to know how to issue tokens to the application based on its settings. The application object will only exist in its home directory, even if it's a multi-tenant application supporting service principals in other directories

Service principals are what govern an application connecting to Azure AD and can be considered the instance of the application in your directory. For any given application, it can have at most one application object (which is registered in a "home" directory) and one or more service principal objects representing instances of the application in every directory in which it acts.



- Useful when Managed Identities cannot be used
- Often used to authenticate external applications to Azure resources

Design a user consent solution for applications

The Microsoft identity platform implements the OAuth 2.0 authorization protocol. This protocol is a method that a third-party app can use to access web-hosted resources on behalf of a user. The web-hosted resources can define a set of permissions that you can use to implement functionality in smaller chunks. Developers can leverage one of two types of permissions supported by the Microsoft identity platform depending on the app scenario.

Knowing the different types of permissions supported in Azure AD applications will enable you to design an access strategy that works for your organization. You'll also learn about the different consent framework models and how they are used to obtain permissions from users for use in custom apps.

Types of permissions

Microsoft identity platform supports two types of permissions: delegated permissions and application permissions.

- Delegated permissions are used by apps that have a signed-in user present. These permissions are provided to the application by the user so the app can perform actions on their behalf. This doesn't give permissions to the app, instead the user is simply allowing the app to act on their behalf using their permissions.
- Application permissions are used by apps that run without a signed-in user present.

Effective permissions

Effective permissions are the permissions that your app will have when making requests to the target resource. It's important to understand the difference between the delegated and application permissions that your app is granted and its effective permissions when making calls to the target resource.

For delegated permissions, the effective permissions of your app are the intersection of the delegated permissions the app has been granted and the privileges of the currently signed-in user. In other words, the app can never have more privileges than the signed-in user. Within organizations, the privileges of the signed-in user may be determined by policy or by membership in one or more administrator roles.

For example, assume your app has been granted the User.ReadWrite.All delegated permission. This permission enables your app to be used to read and update the profile of every user in an organization. If the signed-in user is a global administrator, your app can update the profile of every user in the organization. However, if the signed-in user isn't in an administrator role, your app can update only the profile of the signed-in user. It can't update the profiles of other users in the organization because the user that it has permission to act on behalf of does not have those privileges.

For application permissions, the effective permissions of your app will be the full level of privileges implied by the permission. For example, an app that has the User.ReadWrite.All application permission can update the profile of every user in the organization.

Best practices for requesting permissions

When building an app that uses Azure AD to provide sign-in and access tokens for secured endpoints, there are a few good practices you should follow.

- Only ask for the permissions required for implemented app functionality. Don't request user consent for permissions that you haven't yet implemented for your application.
- In addition, when requesting permissions for app functionality, you should request the least-privileged access. For example, if an app analyzes a user's email but takes no action on the mailbox, you shouldn't request the more permissive Mail.ReadWrite when Mail.Read will work.
- Apps should gracefully handle scenarios where the user doesn't grant consent to the app when permissions are requested. In the case where an app doesn't receive an access token with the required permissions, it should explain the situation to the user with options on how to remedy the issue.

Microsoft recommends restricting user consent to allow users to consent only for app from verified publishers, and only for permissions you select. For apps which do not meet this policy, the decision-making process will be centralized with your organization's security and identity administrator team.

After end-user consent is disabled or restricted, there are several important considerations to ensure your organization stays secure while still allowing business critical applications to be used. These steps are crucial to minimize impact on your organization's support team and IT administrators, while preventing the use of unmanaged accounts in third-party applications.

Next unit: Design for Azure Key Vault

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Key Vault

3 minutes

Storing and handling secrets, encryption keys, and certificates directly is risky, and every usage introduces the possibility of unintentional data exposure. Azure Key Vault provides a secure storage area for managing all your app secrets so you can properly encrypt your data in transit or while it's being stored.

Azure Key Vault helps solve the following problems:

- Secrets Management - Azure Key Vault can be used to Securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets
- Key Management - Azure Key Vault can also be used as a Key Management solution. Azure Key Vault makes it easy to create and control the encryption keys used to encrypt your data.
- Certificate Management - Azure Key Vault is also a service that lets you easily enroll, manage, and deploy public and private Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates for use with Azure and your internal connected resources.

Azure Key Vault has two service tiers: Standard, which encrypts with a software key, and a Premium tier, which includes hardware security module(HSM)-protected keys.

Why use Azure Key Vault?

- Separation of sensitive app information from other configuration and code, reducing the risk of accidental leaks.
- Restricted secret access with access policies tailored to the apps and individuals that need them.
- Centralized secret storage, allowing required changes to happen in only one place.
- Access logging and monitoring to help you understand how and when secrets are accessed.

Key Vault allows you to securely access sensitive information from within your applications:

- Keys, secrets, and certificates are protected without having to write the code yourself and you're easily able to use them from your applications.

- You allow customers to own and manage their own keys, secrets, and certificates so you can concentrate on providing the core software features. In this way, your applications will not own the responsibility or potential liability for your customers' tenant keys, secrets, and certificates.
- Your application can use keys for signing and encryption yet keeps the key management external from your application.
- You can manage credentials like passwords, access keys, and sas tokens by storing them in Key Vault as secrets.
- Manage certificates.

Design a solution using Keys and SAS tokens

A shared access signature (SAS) provides secure delegated access to resources in your storage account. With a SAS, you have granular control over how a client can access your data. For example:

- What resources the client may access.
- What permissions they have to those resources.
- How long the SAS is valid.

When to use a shared access signature

Use a SAS to give secure access to resources in your storage account to any client who does not otherwise have permissions to those resources.

A common scenario where a SAS is useful is a service where users read and write their own data to your storage account. In a scenario where a storage account stores user data, there are two typical design patterns:

- Clients upload and download data via a front-end proxy service, which performs authentication. This front-end proxy service allows the validation of business rules. But for large amounts of data, or high-volume transactions, creating a service that can scale to match demand may be expensive or difficult.
- A lightweight service authenticates the client as needed and then generates a SAS. Once the client application receives the SAS, it can access storage account resources directly. Access permissions are defined by the SAS and for the interval allowed by the SAS. The SAS mitigates the need for routing all data through the front-end proxy service.

Many real-world services may use a hybrid of these two approaches. For example, some data might be processed and validated via the front-end proxy. Other data is saved and/or read directly using SAS.

200 XP

Knowledge check

3 minutes

Tailwind Traders is planning on making some significant changes to their identity and access management solution. They have asked for your assistance on some recommendations and questions. Here are the specific requirements.

- **Device access to company applications.** The CTO has agreed to allow some level of device access. Employees at the company's retail stores will now be able to access certain company applications. This access, however, should be restricted to only approved devices.
- **Company reorganization.** A company-wide reorganization has affected many employees. These employees are now in new roles. The IT team needs to ensure users have the correct access based on their new jobs.
- **External developer accounts.** A new development project requires external software developers to access company data files. The IT team needs to create user accounts for approximately five developers.
- **User sign-in attempts.** A recent audit of user sign-ins attempts revealed anonymous IP addresses and unusual locations. The IT team wants to require multi-factor authentication for these attempted sign-ins.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. How can Tailwind Traders ensure that employees at the company's retail stores can access company applications only from approved tablet devices?

- ○ Single sign-on
- ◉ Conditional access

  ✔ That's correct. Conditional Access enables you to require users to access your applications only from approved, or managed, devices.
- ○ Multi-factor authentication

2. What should Tailwind Traders do to ensure employees have the correct permissions for their job role?

- ○ Create a conditional access policy
- ○ Review each user's role-based access control permissions
- ◉ Require an access review.

✔ **That's correct. An access review would give managers an opportunity to validate the employee's access.**

3. What should Tailwind Traders do to give access to the partner developers?

- ○ Use AD Connect to bring in the developer accounts
- ○ Ask the developers to sign in with a social identification. For example, Google, LinkedIn, or Facebook account.
- ◉ Invite the developers as guest users to their directory.

✔ **That's correct. In Business-to-Business scenarios guest user accounts are created. You can then apply the appropriate permissions.**

4. What solution would be best for the user sign-in attempts requirement?

- ○ Create a user risk policy
- ◉ Create a sign-in risk policy

✔ **That's correct. A sign-in risk policy can identify anonymous IP and atypical locations. Secondary multi-factor authentication can then be required.**

- ○ Require an access review.

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Introduction

3 minutes

Meet Tailwind Traders

Suppose you work for Tailwind Traders that is moving its systems to Azure, with a mixture of IaaS and PaaS services. In its previous environment, the organization had several instances where application performance was degraded, or systems became entirely unavailable. There was an extended delay to identify and resolve the issues. This situation affected customers' ability to access their accounts and led to poor user satisfaction.

The organization wants to design a monitoring strategy that performs full-stack monitoring across all solutions that it uses. There should also be insights and alerting into the collected data. The organization wants to quickly identify and minimize poor performance and system failures in the future. The practice of continuous monitoring must include analysis of platform metrics and logs to get visibility into the health and performance of services that are part of the architecture.

In this module, you'll explore the monitoring solutions available in Azure. You'll assess Azure Monitor and its features such as Application Insights and Azure Monitor Logs to analyze infrastructure and application performance and availability.

Learning objectives

In this module, you'll be able to:

- Design for Azure Monitor data sources
- Design for Log Analytics
- Design for Azure workbooks and Insights

- Design for Azure Workbooks and Insights
- Design for Azure Data Explorer
- Monitor resources for performance efficiency

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design Identity, Governance, and Monitoring Solutions

- Design a Solution for Logging and Monitoring.
  - Design a log routing solution
  - Recommend an appropriate level of logging
  - Recommend a monitoring tool(s) for a solution

Prerequisites

- Working experience with monitoring and logging cloud environments
- Conceptual knowledge of monitoring and logging

Next unit: Design for Azure Monitor data sources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Monitor data sources

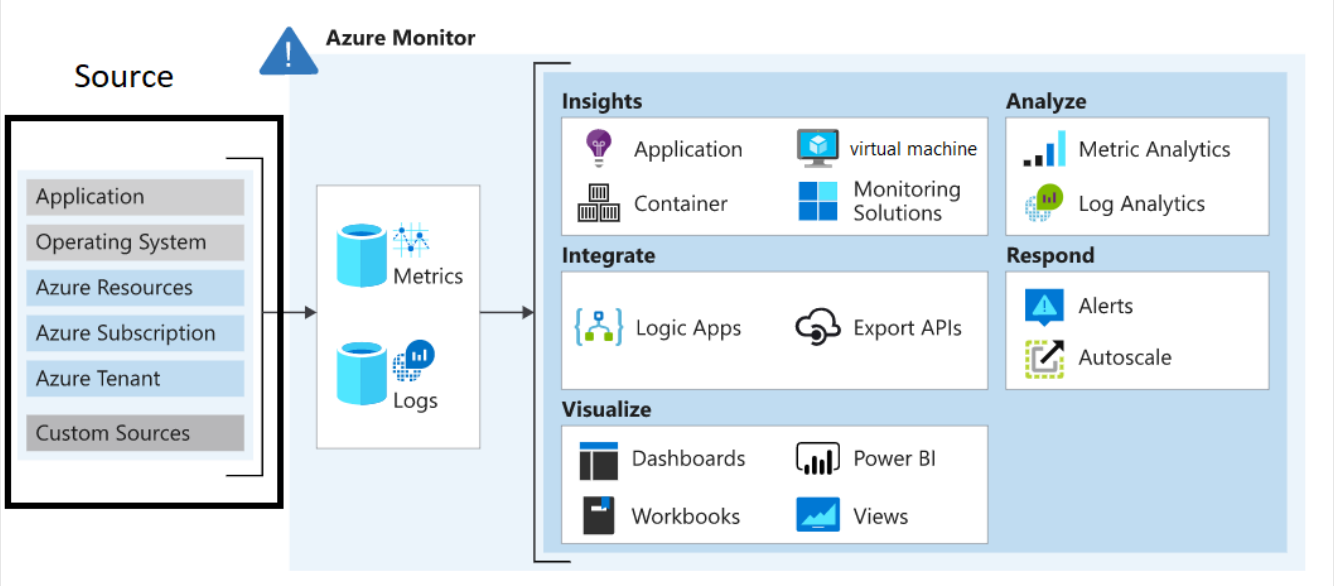
3 minutes

Azure Monitor is based on a common monitoring data platform that includes Logs and Metrics. Collecting data into this platform allows data from multiple resources to be analyzed together using a common set of tools in Azure Monitor. Monitoring data may also be sent to other locations to support certain scenarios, and some resources may write to other locations before they can be collected into Logs or Metrics. Azure Monitor Logs can store various data types each with their own structure. You can also perform complex analysis on logs data using log queries, which cannot be used for analysis of metrics data. Azure Monitor Metrics can support near real-time scenarios, making them useful for alerting and fast detection of issues.

This unit describes the different sources of monitoring data collected by Azure Monitor in addition to the monitoring data created by Azure resources.

Consider Tailwind Traders Azure environment. What sources of monitoring data might they want to collect?

Identify data sources and access method

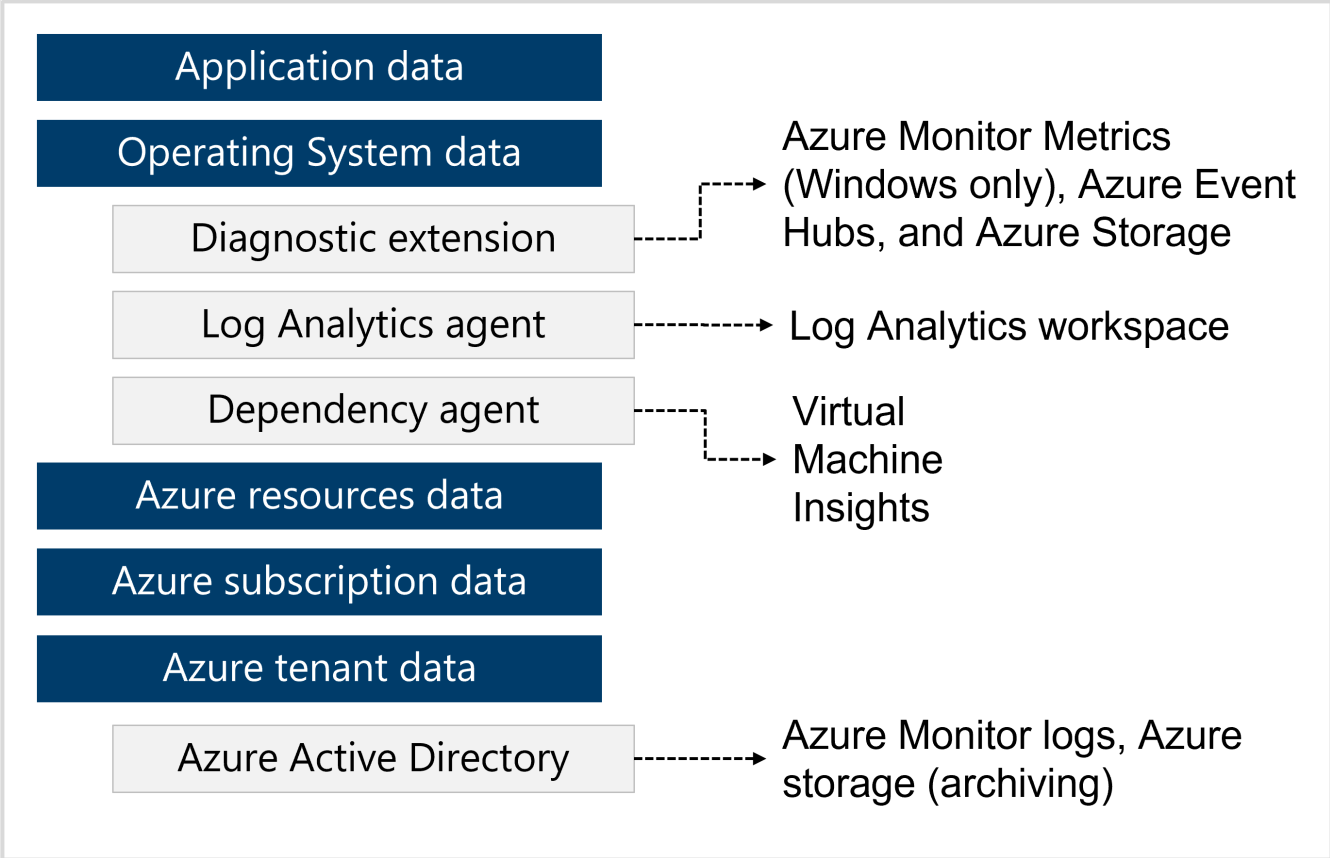


Sources of monitoring data from Azure applications can be organized into tiers, the highest tiers being your application itself and the lower tiers being components of the Azure platform. The method of accessing data from each tier varies. The application tiers are summarized in

the table below, and the sources of monitoring data in each tier are presented in the following sections. Visit Monitoring data locations in Azure for a description of each data location and how you can access its data.

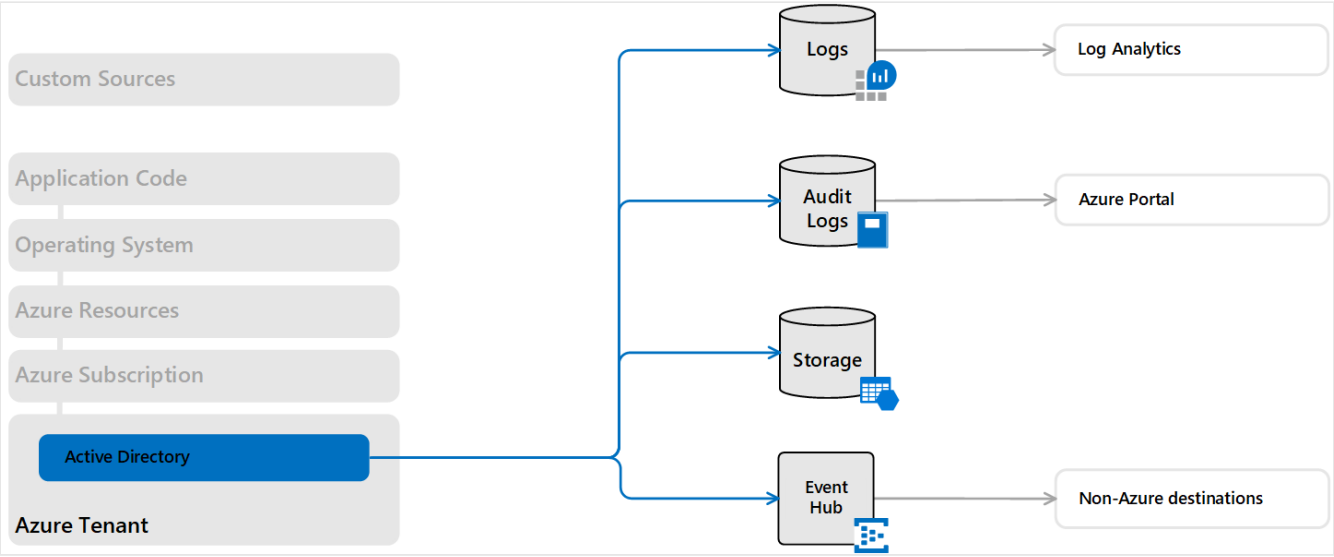
Azure Monitor collects data automatically from a range of components. For example:

- **Application data**: Data that relates to your custom application code.
- **Operating system data**: Data from the Windows or Linux virtual machines that host your application.
- **Azure resource data**: Data that relates to the operations of an Azure resource, such as a web app or a load balancer.
- **Azure subscription data**: Data that relates to your subscription. It includes data about Azure health and availability.
- **Azure tenant data**: Data about your Azure organization-level services, such as Azure Active Directory.



Azure tenant logging solutions

Telemetry related to your Azure tenant is collected from tenant-wide services such as Azure Active Directory.



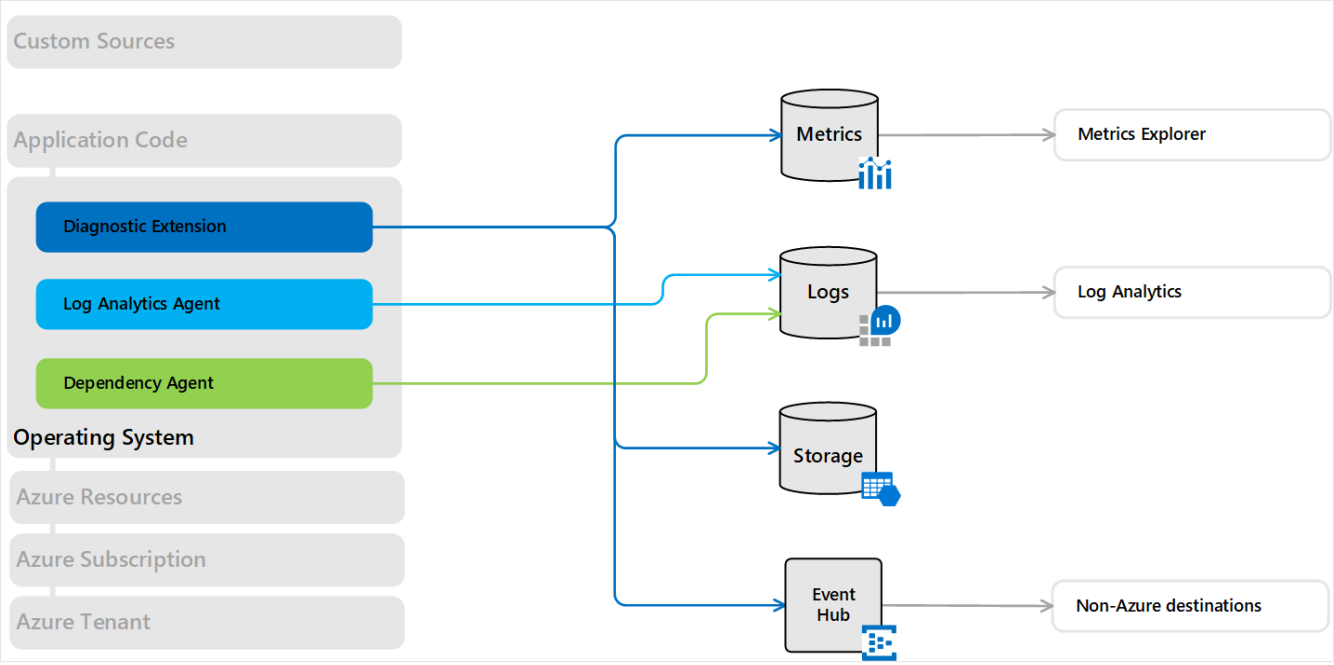
Azure Active Directory audit logs

Azure Active Directory reporting contains the history of sign-in activity and audit trail of changes made within a particular tenant.

Destination	Description	Reference
Azure Monitor Logs	Configure Azure AD logs to be collected in Azure Monitor to analyze them with other monitoring data.	Integrate Azure AD logs with Azure Monitor logs
Azure Storage	Export Azure AD logs to Azure Storage for archiving.	Tutorial: Archive Azure AD logs to an Azure storage account
Azure Event Hubs	Stream Azure AD logs to other locations using Event Hub.	Tutorial: Stream Azure Active Directory logs to an Azure Event Hub.
Azure Monitor partner integrations	Specialized integrations between Azure Monitor and other non-Microsoft monitoring platforms. Useful when you are already using one of the partners.	Extend Azure with solutions from partners

Operating system (guest) logging solutions

Compute resources in Azure, in other clouds, and on-premises have a guest operating system to monitor. With the installation of one or more agents, you can gather telemetry from the guest into Azure Monitor to analyze it with the same monitoring tools as the Azure services themselves.



Next unit: Design for Log Analytics

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Log Analytics

3 minutes

Azure Monitor stores log data in a Log Analytics workspace, which is an Azure resource and a container where data is collected, aggregated, and serves as an administrative boundary. While you can deploy one or more workspaces in your Azure subscription, there are several considerations you should understand in order to ensure your initial deployment is following our guidelines to provide you with a cost effective, manageable, and scalable deployment meeting your organization's needs.

Availability, Latency, and Cost	Immutable Storage
Premium blob storage	Legal hold policies
Hot, cool, and archive access tiers	Time-based retention policies

Data in a workspace is organized into tables, each of which stores different kinds of data and has its own unique set of properties based on the resource generating the data. Most data sources will write to their own tables in a Log Analytics workspace.

A Log Analytics workspace provides:

- A geographic location for data storage.
- Data isolation by granting different users access rights following one of our recommended design strategies.
- Scope for configuration of settings like pricing tier, retention, and data capping.

Workspaces are hosted on physical clusters. By default, the system is creating and managing these clusters. Customers that ingest more than 4TB/day are expected to create their own dedicated clusters for their workspaces - it enables them better control and higher ingestion rate.

Below is an overview of the design considerations, access control overview, and an understanding of the design implementations to consider for an IT organization like Tailwind traders.

Important considerations for an access control strategy

Identifying the number of workspaces, your need is influenced by one or more of the following requirements:

- You are a global company, and you need log data stored in specific regions for data sovereignty or compliance reasons.
- You are using Azure and you want to avoid outbound data transfer charges by having a workspace in the same region as the Azure resources it manages.
- You manage multiple departments or business groups. Each group should access their data but not the data of others. Also, there is no business requirement for a consolidated cross department or business group view.

IT organizations today are modeled following either a centralized, decentralized, or an in-between hybrid of both structures. As a result, the following workspace deployment models have been commonly used to map to one of these organizational structures:

- **Centralized**: All logs are stored in a central workspace and administered by a single team, with Azure Monitor providing differentiated access per-team. In this scenario, it is easy to manage, search across resources, and cross-correlate logs. The workspace can grow significantly depending on the amount of data collected from multiple resources in your subscription, with additional administrative overhead to maintain access control to different users. This model is known as "hub and spoke".
- **Decentralized**: Each team has their own workspace created in a resource group they own and manage, and log data is segregated per resource. In this scenario, the workspace can be kept secure and access control is consistent with resource access, but it's difficult to cross-correlate logs. Users who need a broad view of many resources cannot analyze the data in a meaningful way.
- **Hybrid**: Security audit compliance requirements further complicate this scenario because many organizations implement both deployment models in parallel. This commonly results in a complex, expensive, and hard-to-maintain configuration with gaps in logs coverage.

Centralized logging can help you uncover hidden issues that might be difficult to track down. With Log Analytics, you can query and aggregate data across logs. This cross-source correlation can help you identify issues or performance problems. Log Analytics receives monitoring data from your Azure resources and makes it available to consumers for analysis or visualization.

When using the Log Analytics agents to collect data, you need to understand the following in order to plan your agent deployment:

- To collect data from Windows agents, you can configure each agent to report to one or more workspaces, even while it is reporting to a System Center Operations Manager management group. The Windows agent can report up to four workspaces.
- The Linux agent does not support multi-homing and can only report to a single workspace.

If you are using System Center Operations Manager 2012 R2 or later:

- Each Operations Manager management group can be connected to only one workspace.
- Linux computers reporting to a management group must be configured to report directly to a Log Analytics workspace. If your Linux computers are already reporting directly to a workspace and you want to monitor them with Operations Manager, follow these steps to report to an Operations Manager management group.
- You can install the Log Analytics Windows agent on the Windows computer and have it report to both Operations Manager integrated with a workspace, and a different workspace.

Access control overview

With Azure role-based access control (Azure RBAC), you can grant users and groups only the amount of access they need to work with monitoring data in a workspace. This allows you to align with your IT organization operating model using a single workspace to store collected data enabled on all your resources. For example, you grant access to your team responsible for infrastructure services hosted on Azure virtual machines (VMs), and as a result they'll have access to only the logs generated by the VMs. This is following the new resource-context log model. The basis for this model is for every log record emitted by an Azure resource, it is automatically associated with this resource. Logs are forwarded to a central workspace that respects scoping and Azure RBAC based on the resources.

The data a user has access to is determined by a combination of factors that are listed in the following table. Each is described in the table below.

Factor	Description
Access mode	Method the user uses to access the workspace. Defines the scope of the data available and the access control mode that's applied.

Factor	Description
Access control mode	Setting on the workspace that defines whether permissions are applied at the workspace or resource level.
Permissions	Permissions applied to individual or groups of users for the workspace or resource. Defines what data the user will have access to.
Table level Azure RBAC	Optional granular permissions that apply to all users regardless of their access mode or access control mode. Defines which data types a user can access.

Recommend an access mode

The access mode refers to how a user accesses a Log Analytics workspace and defines the scope of data they can access.

Users have two options for accessing the data:

- **Workspace-context**: You can review all logs in the workspace you have permission to. Queries in this mode are scoped to all data in all tables in the workspace. This is the access mode used when logs are accessed with the workspace as the scope, such as when you select **Logs** from the **Azure Monitor** menu in the Azure portal.
- **Resource-context**: When you access the workspace for a particular resource, resource group, or subscription, such as when you select **Logs** from a resource menu in the Azure portal, you can view logs for only resources in all tables that you have access to. Queries in this mode are scoped to only data associated with that resource. This mode also enables granular Azure RBAC.

The following table summarizes and compares the access modes:

Issue	Workspace-context	Resource-context

Issue	Workspace-context	Resource-context
Who is each model intended for?	Central administration. Administrators who need to configure data collection and users who need access to a wide variety of resources. Also currently required for users who need to access logs for resources outside of Azure.	Application teams. Administrators of Azure resources being monitored.
What does a user require to review logs?	Permissions to the workspace. Explore **Workspace permissions** in Manage access using workspace permissions.	Read access to the resource. Explore **Resource permissions** in Manage access using Azure permissions. Permissions can be inherited (such as from the containing resource group) or directly assigned to the resource. Permission to the logs for the resource will be automatically assigned.
What is the scope of permissions?	Workspace. Users with access to the workspace can query all logs in the workspace from tables that they have permissions to. Explore Table access control	Azure resource. User can query logs for specific resources, resource groups, or subscription they have access to from any workspace but can't query logs for other resources.
How can user access logs?	Start logs from Azure Monitor and Log Analytics workspaces. Review the logs from Azure Monitor Workbooks.	Same as workspace-context, and you start logs from the Azure resource.

Scale and ingestion volume rate limit

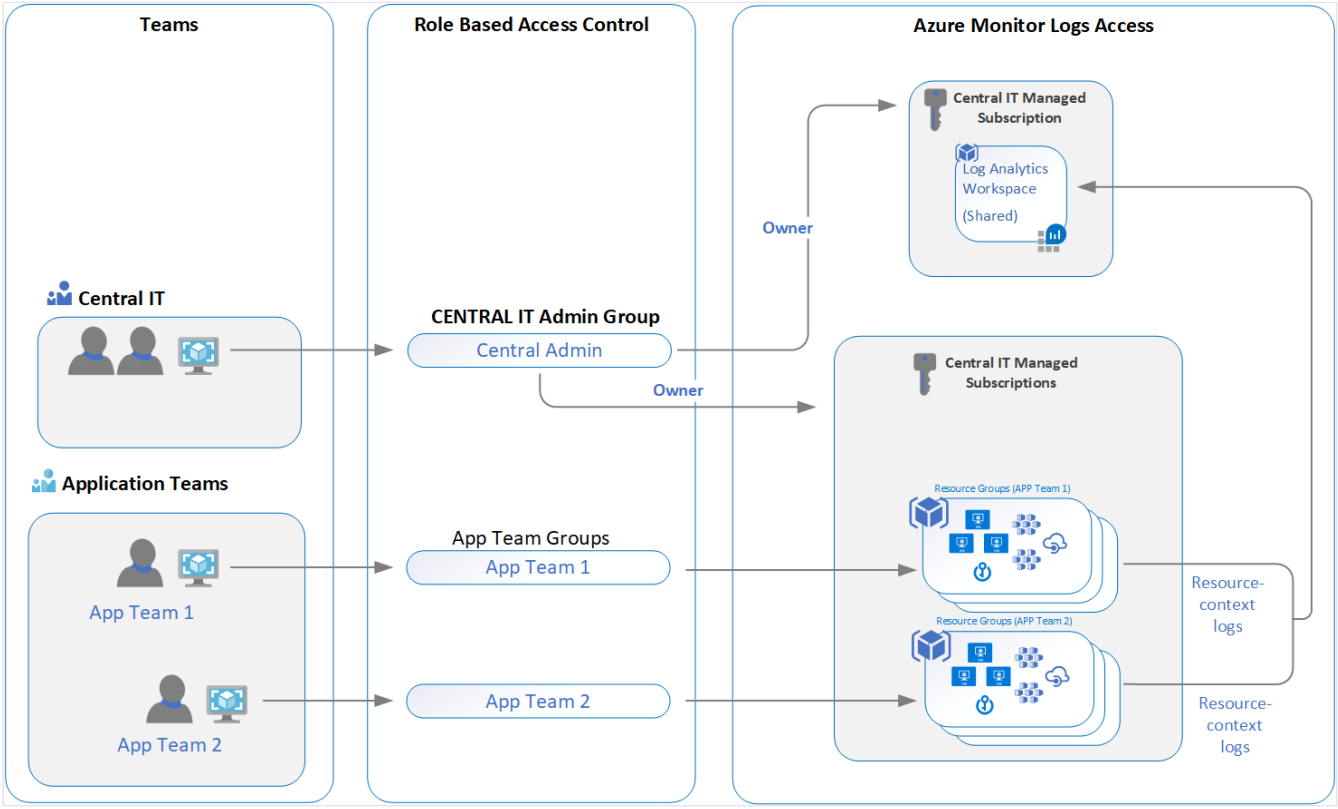
Azure Monitor is a high scale data service that serves thousands of customers sending petabytes of data each month at a growing pace. Workspaces are not limited in their storage space and can grow to petabytes of data. There is no need to split workspaces due to scale.

To protect and isolate Azure Monitor customers and its backend infrastructure, there is a default ingestion rate limit that is designed to protect from spikes and floods situations. The rate limit default is about **6 GB/minute** and is designed to enable normal ingestion. For more details on ingestion volume limit measurement, explore Azure Monitor service limits.

Customers that ingest less than 4TB/day will usually not meet these limits. Customers that ingest higher volumes or that have spikes as part of their normal operations shall consider moving to dedicated clusters where the ingestion rate limit could be raised.

When the ingestion rate limit is activated or get to 80% of the threshold, an event is added to the Operation table in your workspace. It is recommended to monitor it and create an alert. explore more details in data ingestion volume rate.

Recommendations



This scenario covers a single workspace design in your IT organization's subscription that is not constrained by data sovereignty or regulatory compliance or needs to map to the regions your resources are deployed within. It allows your organization's security and IT admin teams the ability to leverage the improved integration with Azure access management and more secure access control.

All resources, monitoring solutions, and Insights such as Application Insights and VM insights, supporting infrastructure and applications maintained by the different teams are configured to forward their collected log data to the IT organization's centralized shared workspace. Users on each team are granted access to logs for resources they have been given access to.

Once you have deployed your workspace architecture, you can enforce this on Azure resources with Azure Policy. It provides a way to define policies and ensure compliance with your Azure resources, so they send all their resource logs to a particular workspace. For example, with Azure virtual machines or virtual machine scale sets, you can use existing

policies that evaluate workspace compliance and report results or customize to remediate if non-compliant.

Next unit: Design for Azure Workbooks and Azure Insights

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Workbooks and Azure Insights

3 minutes

Design for Azure workbooks

Workbooks provide a flexible canvas for data analysis and the creation of rich visual reports within the Azure portal. They allow you to tap into multiple data sources from across Azure and combine them into unified interactive experiences. Authors of workbooks can transform this data to provide insights into the availability, performance, usage, and overall health of the underlying components. For instance, analyzing performance logs from virtual machines to identify high CPU or low memory instances and displaying the results as a grid in an interactive report.

Workbooks are currently compatible with the following data sources:

- Logs
- Metrics
- Azure Resource Graph
- Alerts
- Workload Health
- Azure Resource Health
- Azure Data Explorer

But the real power of workbooks is the ability to combine data from disparate sources within a single report. This allows for the creation of composite resource views or joins across resources enabling richer data and insights that would otherwise be impossible.

Customers use workbooks in several ways—exploring the usage of an app, going through a root cause analysis, and putting together an operational playbook, for example.

How would you leverage Azure workbooks for Tailwind Traders? What recommendations would you have based on their Azure environment and business needs?

Design for Azure Insights

The reputation of your organization depends on the performance, reliability, and security of its systems. It's critical to monitor your systems closely to identify any performance problems or attacks before they can affect users. For example, if your payment system is unable to process user transactions during a high-volume holiday sales period, your customers will likely lose confidence in your business. Designing insights as a part of your overall architecture will help identify performance issues.

Insights

Insights provide a customized monitoring experience for particular applications and services. They collect and analyze both logs and metrics. Here are just a few of the insights that are provided.

Insight	Description
Application Insights	Extensible Application Performance Management (APM) service to monitor your live web application on any platform.
Container insights	Monitors the performance of container workloads deployed to either Azure Container Instances or managed Kubernetes clusters hosted on Azure Kubernetes Service (AKS).
Cosmos DB insights	Provides information on the overall performance, failures, capacity, and operational health of all your Azure Cosmos DB resources in a unified interactive experience.
Networks insights	Provides comprehensive information on the health and metrics for all your network resource. The advanced search capability helps you identify resource dependencies, enabling scenarios like identifying resource that are hosting your website, by simply searching for your website name.
Resource Group insights	Triage and diagnose any problems your individual resources encounter, while offering context as to the health and performance of the resource group as a whole.

Insight	Description
Storage insights	Provides comprehensive monitoring of your Azure Storage accounts by delivering a unified report of your Azure Storage services performance, capacity, and availability.
VM insights	Monitors your Azure virtual machines (VM) and virtual machine scale sets at scale. It analyzes the performance and health of your Windows and Linux VMs, and monitors their processes and dependencies on other resources and external processes.
Key Vault insights	Provides comprehensive monitoring of your key vaults by delivering a unified report of your Key Vault requests, performance, failures, and latency.
Azure Cache for Redis insights	Provides a unified, interactive report of overall performance, failures, capacity, and operational health.

Use Application Insights to:

- Analyze and address issues and problems that affect your application's health and performance.
- Improve your application's development lifecycle.
- Measure your user experience and analyze users' behavior.

Application Insights is aimed at the development team, to help you understand how your app is performing and how it's being used. It monitors:

- **Request rates, response times, and failure rates** - Find out which pages are most popular, at what times of day, and where your users are. Determine which pages perform best. If your response times and failure rates go high when there are more requests, then perhaps you have a resourcing problem.
- **Dependency rates, response times, and failure rates** - Find out whether external services are slowing you down.
- **Exceptions** - Analyze the aggregated statistics, or pick specific instances and drill into the stack trace and related requests. Both server and browser exceptions are reported.
- **Page views and load performance** - reported by your users' browsers.

- **AJAX calls** from web pages - rates, response times, and failure rates.
- **User and session counts**.
- **Performance counters** from your Windows or Linux server machines, such as CPU, memory, and network usage.
- **Host diagnostics** from Docker or Azure.
- **Diagnostic trace logs** from your app - so that you can correlate trace events with requests.
- **Custom events and metrics** that you write yourself in the client or server code, to track business events such as items sold or games won.

Use Azure Monitor VM insights to:

- View the health and performance of your VMs
- Monitor your VMs at-scale across multiple subscriptions and resource groups.
- Want a topology view that shows the processes, and network connection details of your VMs and scale sets.
- Insights supports Azure virtual machines and scale sets
- Hybrid virtual machines connected with Azure Arc
- On-premises virtual machines
- Virtual machines hosted in another cloud environment

Use Azure Monitor container insights to:

- View the health and performance of your Kubernetes workloads at-scale across multiple subscriptions and resource groups.
- Want visibility into memory and processor performance metrics from controllers, nodes, and containers.
- Want view and store container logs for real time and historical analysis.
- Identify AKS containers that are running on the node and their average processor and memory utilization. This knowledge can help you identify resource bottlenecks.

- Identify processor and memory utilization of container groups and their containers hosted in Azure Container Instances.
- Identify where the container resides in a controller or a pod. This knowledge can help you view the controller's or pod's overall performance.
- Review the resource utilization of workloads running on the host that are unrelated to the standard processes that support the pod.
- Understand the behavior of the cluster under average and heaviest loads. This knowledge can help you identify capacity needs and determine the maximum load that the cluster can sustain.
- Configure alerts to proactively notify you or record it when CPU and memory utilization on nodes or containers exceed your thresholds, or when a health state change occurs in the cluster at the infrastructure or nodes health rollup.
- Integrate with Prometheus to view application and workload metrics it collects from nodes and Kubernetes using queries to create custom alerts, dashboards, and perform detailed analysis.
- Monitor container workloads deployed to AKS Engine on-premises and AKS Engine on Azure Stack.
- Monitor container workloads deployed to Azure Red Hat OpenShift.
- Monitor container workloads deployed to Azure Arc enabled Kubernetes.

Next unit: Design for Azure Data Explorer

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

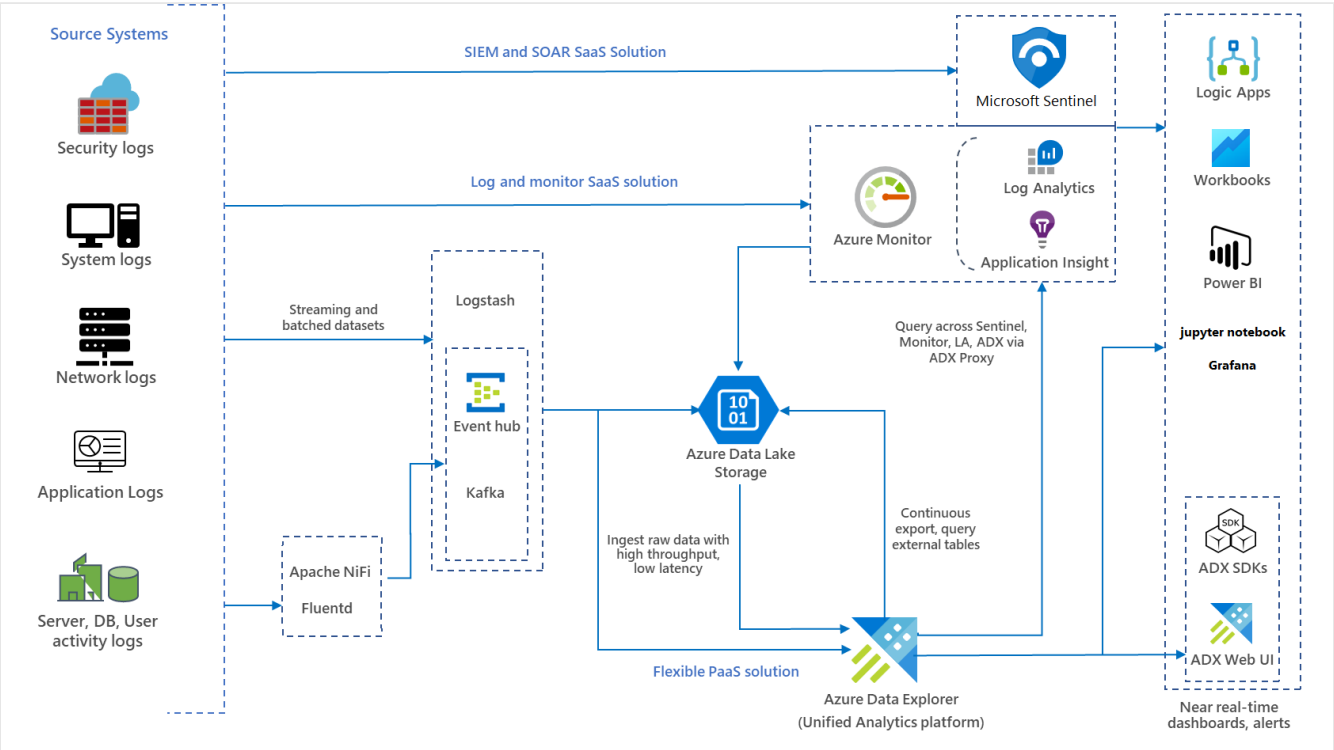
100 XP

Design for Azure Data Explorer

3 minutes

Azure Data Explorer is a fast and highly scalable data exploration service for log and telemetry data. It helps you handle the many data streams emitted by modern software, so you can collect, store, and analyze data. Azure Data Explorer is ideal for analyzing large volumes of diverse data from any data source, such as websites, applications, IoT devices, and more. This data is used for diagnostics, monitoring, reporting, machine learning, and additional analytics capabilities.

Below is an example of a hybrid end-to-end monitoring solution integrated with Azure Sentinel and Azure Monitor for ingesting streamed and batched logs from diverse sources, on-premises, or any cloud within an enterprise ecosystem. This could be a solution used in Tailwind Traders architecture to monitor various logs.



Combine features provided by Microsoft Sentinel and Azure Monitor with Azure Data Explorer to build a flexible and cost-optimized end-to-end monitoring solution. Below are some examples:

- Use Azure Monitor's native capabilities for IT asset monitoring, dashboarding, and alerting so you can ingest logs from VMs, services, and so on.

- Use Azure Data Explorer for full flexibility and control in all aspects for all types of logs in the following scenarios:
  - No out of the box features provided by Microsoft Sentinel and Azure Monitor SaaS solutions such as application trace logs.
  - Greater flexibility for building quick and easy near-real-time analytics dashboards, granular role-based access control, time series analysis, pattern recognition, anomaly detection and forecasting, and machine learning. Azure Data Explorer is also well integrated with ML services such as Databricks and Azure Machine Learning. This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.
  - Longer data retention is required in cost effective manner.
  - Centralized repository is required for different types of logs. Azure Data Explorer, as a unified big data analytics platform, allows you to build advanced analytics scenarios.

Next unit: Monitor resources for performance efficiency

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Monitor resources for performance efficiency

3 minutes

troubleshooting an application's performance requires monitoring and reliable investigation. Issues in performance can arise from database queries, connectivity between services, under-provision resources, or memory leaks in code.

Continuously monitoring services and checking the health state of current workloads is key in maintaining the overall performance of the workload. An overall monitoring strategy consider these factors:

- Scalability
- Resiliency of the infrastructure, application, and dependent services
- Application and infrastructure performance

What to consider when defining a monitoring strategy

Here is a list to consider ensuring you are monitoring your workloads with performance and scalability in mind:

- Enable and capture telemetry throughout your application to build and visualize end-to-end transaction flows for the application.
- Explore metrics from Azure services such as CPU and memory utilization, bandwidth information, current storage utilization, and more.
- Use resource and platform logs to get information about what events occur and under which conditions.
- For scalability, examine the metrics to determine how to provision resources dynamically and scale with demand.

- In the collected logs and metrics identify signs that make a system or its components suddenly become unavailable.
- Use log aggregation technology to gather information across all application components.
- Store logs and key metrics of critical components for statistical evaluation and predicting trends.
- Identify antipatterns in the code.

Follow these questions to assess the workload at a deeper level.

Assessment	Description
Are application logs and events correlated across all application components?	Correlate logs and events for subsequent interpretation. This will give you visibility into end-to-end transaction flows.
Are you collecting Azure Activity Logs within the log aggregation tool?	Collect platform metrics and logs to get visibility into the health and performance of services that are part of the architecture.
Are application and resource level logs aggregated in a single data sink, or is it possible to cross-query events at both levels?	Implement a unified solution to aggregate and query application and resource level logs, such as Azure Log Analytics.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Knowledge check

3 minutes

Tailwind Traders has several workloads being migrated to Azure. It is important you design logging and monitoring for the workloads based on the following requirements:

- **Host all logs in a single location.** The company has one team responsible for designing the logging and monitoring strategy for all workloads in Azure. This team needs a solution that is easy to manage, enables them to search across resources, and cross-correlate logs.
- **sign-in activity.** The security team requires a report of user sign-in activity.
- **Measure user experience and analyze users' behavior.** The reputation of the company depends on the performance, reliability, and security of its systems. It's critical to monitor your systems closely to identify any performance problems or attacks before they can affect users.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. Which Log Analytics Workspace deployment model best supports the company's needs to host all logs in a single location?

- (●) Centralized
  - ✔ **That's correct. The centralize model meets their needs.**
- ( ) Decentralized
- ( ) Hybrid

2. What solution should be used to log user sign-in activity?

- (●) Azure Active Directory Audit Logs
  - ✔ **That's correct. Azure Active Directory audit logs contain the history of sign-in activity. The logs also contain an audit trail of tenant changes.**
- ( ) VM Insights
- ( ) Azure Alerts

3. What monitoring tool should be used to measure user experience and analyze users' behavior for all external facing applications?

- (●) Azure Application Insights

Azure Application Insights

✔ That's correct. Application Insights can measure user experience and analyze users' behavior.

- ○ Azure Monitor container insights
- ○ Azure Activity log

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆