**Introduction**

Completed100 XP

* 2 minutes

The term *governance* describes the general process of establishing rules and policies and ensuring that those rules and policies are enforced.

When running in the cloud, 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, like [PCI DSS](https://learn.microsoft.com/en-us/microsoft-365/compliance/offering-pci-dss).
* 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 Traders](https://www.tailwindtraders.com/) is a fictitious home improvement retailer. It operates retail hardware stores across the globe and online.



Tailwind Traders specializes in competitive pricing, fast shipping, and a large range of items. It's looking at cloud technologies to improve business operations and support growth into new markets. By moving to the cloud, the company plans to enhance its shopping experience to further differentiate itself from competitors.

**How will Tailwind Traders improve agility while maintaining control?**

Tailwind Traders is continuing its migration to the cloud. For its existing datacenter, development and test teams must submit support tickets to request access to virtual machines, storage, and networking components. It can take IT staff anywhere from two weeks to two months to purchase, provision, and configure these components.

By working in the cloud, you essentially have immediate access to compute, storage, and networking components. Many kinds of groups and users, including people from development, test, operations, and security teams, can potentially have direct access to cloud resources.

Going forward, Tailwind Traders could enforce similar processes that prevent teams from directly creating or configuring resources on Azure, similar to its existing approach where central IT provisions infrastructure. But the company knows that these restrictions reduce team agility and the ability to innovate. How can they enable innovation while still maintaining control?

In this module, you'll help the company explore ways it can enforce standards while still enabling teams to create and manage the cloud resources they need.

**Learning objectives**

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

* Make organizational decisions about your cloud environment by using the Cloud Adoption Framework for Azure.
* Define who can access cloud resources by using Azure role-based access control.
* Apply a resource lock to prevent accidental deletion of your Azure resources.
* Apply tags to your Azure resources to help describe their purpose.
* Control and audit how your resources are created by using Azure Policy.
* Enable governance at scale across multiple Azure subscriptions by using Azure Blueprints.

**Next unit: Control access to cloud resources by using Azure role-based access control**

**Control access to cloud resources by using Azure role-based access control**

Completed100 XP

* 4 minutes

When you have multiple IT and engineering teams, how can you control what access they have to the resources in your cloud environment? It's a good security practice to grant users only the rights they need to perform their job, and only to the relevant resources.

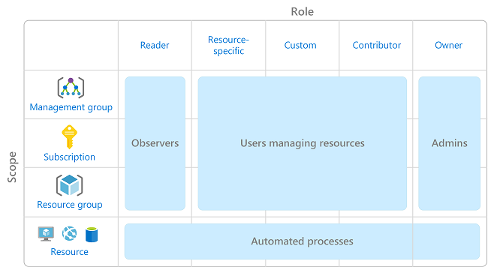
Instead of defining the detailed access requirements for each individual, and then updating access requirements when new resources are created, Azure enables you to control access through [Azure role-based access control](https://learn.microsoft.com/en-us/azure/role-based-access-control/overview) (Azure RBAC).

Azure provides built-in roles that describe common access rules for cloud resources. You can also define your own roles. Each role has an associated set of access permissions that relate to that role. When you assign individuals or groups to one or more roles, they receive all of the associated access permissions.

**How is role-based access control applied to resources?**

Role-based access control is applied to a *scope*, which is a resource or set of resources that this access applies to.

Here's a diagram that shows the relationship between roles and scopes.



Scopes include:

* A management group (a collection of multiple subscriptions).
* A single subscription.
* A resource group.
* A single resource.

*Observers*, *Users managing resources*, *Admins*, and *Automated processes* illustrate the kinds of users or accounts that would typically be assigned each of the various roles.

When you grant access at a parent scope, those permissions are inherited by all child scopes. For example:

* When you assign the [Owner](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#owner) role to a user at the management group scope, that user can manage everything in all subscriptions within the management group.
* When you assign the [Reader](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#reader) role to a group at the subscription scope, the members of that group can view every resource group and resource within the subscription.
* When you assign the [Contributor](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#contributor) role to an application at the resource group scope, the application can manage resources of all types within that resource group, but not other resource groups within the subscription.

**When should I use Azure RBAC?**

Use Azure RBAC when you need to:

* Allow one user to manage VMs 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.

These are just a few examples. You'll find the complete list of built-in roles at the end of this module.

**How is Azure RBAC enforced?**

Azure RBAC is enforced on any action that's initiated against an Azure resource that passes through Azure Resource Manager. Resource Manager is a management service that provides a way to organize and secure your cloud resources.

You typically access Resource Manager from the Azure portal, Azure Cloud Shell, Azure PowerShell, and the Azure CLI. Azure RBAC doesn't enforce access permissions at the application or data level. Application security must be handled by your application.

RBAC uses an *allow model*. When you're assigned a role, RBAC *allows* you to perform certain actions, such as read, write, or delete. If one role assignment grants you read permissions to a resource group and a different role assignment grants you write permissions to the same resource group, you have both read and write permissions on that resource group.

**Who does Azure RBAC apply to?**

You can apply Azure RBAC to an individual person or to a group. You can also apply Azure RBAC to other special identity types, such as service principals and managed identities. These identity types are used by applications and services to automate access to Azure resources.

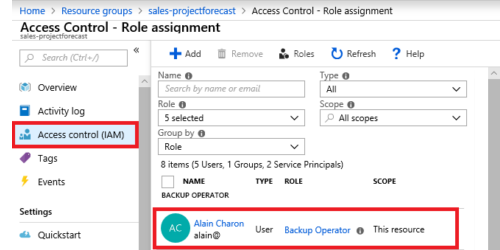
Tailwind Traders has the following teams with an interest in some part of their overall IT environment:

* IT Administrators This team has ultimate ownership of technology assets, both on-premises and in the cloud. The team requires full control of all resources.
* Backup and Disaster Recovery This team is responsible for managing the health of regular backups and invoking any data or system recoveries.
* Cost and Billing People in this team track and report on technology-related spend. They also manage the organization's internal budgets.
* Security Operations This team monitors and responds to any technology-related security incidents. The team requires ongoing access to log files and security alerts.

**How do I manage Azure RBAC permissions?**

You manage access permissions on the **Access control (IAM)** pane in the Azure portal. This pane shows who has access to what scope and what roles apply. You can also grant or remove access from this pane.

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



**Next unit: Prevent accidental changes by using resource locks**

**Prevent accidental changes by using resource locks**

Completed100 XP

* 3 minutes

A [resource lock](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources) prevents resources from being accidentally deleted or changed.

Even with Azure role-based access control (Azure RBAC) policies in place, there's still a risk that people with the right level of access could delete critical cloud resources. Think of a resource lock as a warning system that reminds you that a resource should not be deleted or changed.

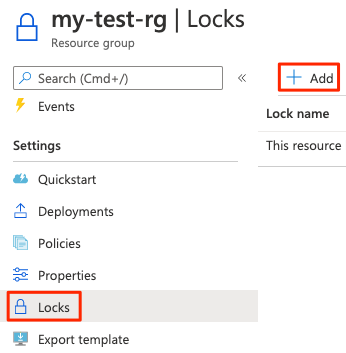
For example, at Tailwind Traders, an IT administrator was performing routine cleanup of unused resources in Azure. The admin accidentally deleted resources that appeared to be unused. But these resources were critical to an application that's used for seasonal promotions. How can resource locks help prevent this kind of incident from happening in the future?

**How do I manage resource locks?**

You can manage resource locks from the Azure portal, PowerShell, the Azure CLI, or from an Azure Resource Manager template.

To view, add, or delete locks in the Azure portal, go to the **Settings** section of any resource's **Locks** pane in the Azure portal.

Here's an example that shows how to add a resource lock from the Azure portal. You'll apply a similar resource lock in the next part.



**What levels of locking are available?**

You can apply locks to a subscription, a resource group, or an individual resource. You can set the lock level to **CanNotDelete** or **ReadOnly**.

* **CanNotDelete** means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock.
* **ReadOnly** means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the **Reader** role in Azure RBAC.

**How do I delete or change a locked resource?**

Although locking helps prevent accidental changes, you can still make changes by following a two-step process.

To modify a locked resource, you must first remove the lock. After you remove the lock, you can apply any action you have permissions to perform. This additional step allows the action to be taken, but it helps protect your administrators from doing something they might not have intended to do.

Resource locks apply regardless of RBAC permissions. Even if you're an owner of the resource, you must still remove the lock before you can perform the blocked activity.

**Combine resource locks with Azure Blueprints**

What if a cloud administrator accidentally deletes a resource lock? If the resource lock is removed, its associated resources can be changed or deleted.

To make the protection process more robust, you can combine resource locks with Azure Blueprints. Azure Blueprints enables you to define the set of standard Azure resources that your organization requires. For example, you can define a blueprint that specifies that a certain resource lock must exist. Azure Blueprints can automatically replace the resource lock if that lock is removed.

You'll learn more about Azure Blueprints later in this module.

**Next unit: Exercise - Protect a storage account from accidental deletion by using a resource lock**

**Exercise - Protect a storage account from accidental deletion by using a resource lock**

Completed100 XP

* 8 minutes

In this exercise, you see how resource locks help prevent accidental deletion of your Azure resources.

To do so, you create a resource group from the Azure portal. Think of a resource group as a container for related Azure resources. Then you add a lock to your resource group and verify that you can't delete the resource group.

You then add a storage account to your resource group and see how the lock from the parent resource group prevents the storage account from being deleted. A storage account is a container that groups a set of Azure Storage services together.

**Important**

You need your own [**Azure subscription**](https://azure.microsoft.com/free/) to complete the exercises in this module. If you don't have an Azure subscription, you can still read along.

**Create the resource group**

Here you create a resource group that's named **my-test-rg**.

1. Go to the [Azure portal](https://portal.azure.com/) and sign in.
2. At the top of the page, select **Resource groups**.
3. Select **+ New**. The **Create a resource group** page appears.
4. In the **Basics** tab, fill in the following fields.

**Setting**

**Value**

**Project details**

Subscription

*Your Azure subscription*

Resource group

**my-test-rg**

**Resource details**

Region

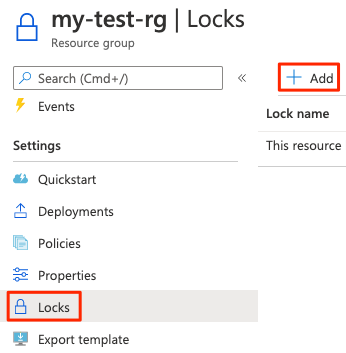
**(US) East US**

You can also select a region that's closer to you.

1. Select **Review + create**, and then select **Create**.

**Add a lock to the resource group**

Add a resource lock to the resource group. To do so:

1. From the Azure portal, select your resource group, **my-test-rg**.
2. Under **Settings**, select **Locks**, and then select **Add**.  
   
3. Fill in these fields.

**Setting**

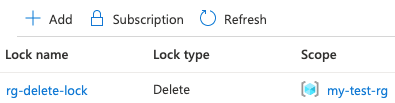
**Value**

Lock name

**rg-delete-lock**

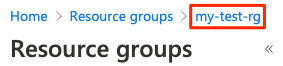
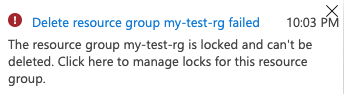
Lock type

**Delete**

1. Select **OK**. You see that the resource lock is applied to your resource group.  
   

**Verify that the resource group is protected from deletion**

Here, you verify protection by attempting to delete the resource group.

1. From the top of the page, select **my-test-rg** to go to your resource group's overview page.  
   
2. Select **Delete resource group**.  
   A screenshot of the Azure portal showing the Delete resource group button.
3. At the prompt, enter **my-test-rg**, and then select **OK**. You see a message that tells you that the resource group is locked and can't be deleted.  
   

**Protect a storage account from accidental deletion**

Here, you add a storage account to your resource group and see how the lock from the parent resource group prevents the storage account from being deleted. To do so:

1. From the Azure portal, at the top of the page, select **Home** to return to the start page.
2. Select **Storage accounts**. Then select **+ New**. The **Create storage account** page appears.
3. In the **Basics** tab, fill in the following fields.

**Note**

Replace **NNN** with a series of numbers. The numbers help to ensure that your storage account name is unique.

**Setting**

**Value**

**Project details**

Subscription

*Your Azure subscription*

Resource group

**my-test-rg**

**Instance details**

Storage account name

**mysaNNN**

Location

**(US) East US**

Performance

**Standard**

Account kind

**StorageV2 (general purpose v2)**

Replication

**Locally redundant storage (LRS)**

As before, you can also select a region that's closer to you.

1. Select **Review + create**, and then select **Create**. The deployment might take a few moments to complete.
2. Select **Go to resource**.
3. At the top of the page, select **Delete**.  
   A screenshot of the Azure portal showing the location of the Delete button.  
   You see a message that tells you the resource or its parent is locked and can't be deleted. Here's an example that shows the error message for a storage account that's named **mysa1234**.  
   A screenshot of the Azure portal showing the error message. The error message states that the storage account can't be deleted because the resource or its parent resource group is locked.

Although you didn't create a lock specifically for the storage account, the lock you created for the parent resource group prevents you from deleting the resource. In other words, the storage account inherits the lock from the parent resource group.

**Delete the resource group and the storage account**

You no longer need your resource group or storage account. Here you remove both.

When you delete a resource group, you also delete its child resources, such as the storage account you previously created.

To delete the resource group, you first need to remove the resource lock.

1. From the Azure portal, select **Home** > **Resource groups** > **my-test-rg** to go to your resource group.
2. Under **Settings**, select **Locks**.
3. Locate **rg-delete-lock**, and select **Delete** on that same row.
4. Select **Overview**, and then select **Delete resource group**.
5. At the prompt, enter **my-test-rg**, and then select **OK**. The deletion operation might take a few moments to complete.
6. When the operation completes, select **Home** > **Resource groups**. You see that the **my-test-rg** resource group no longer exists in your account. Your storage account is also deleted.

Nice work. You can now apply resource locks to help prevent the accidental deletion of your Azure resources.

**Next unit: Organize your Azure resources by using tags**

**Organize your Azure resources by using tags**

Completed100 XP

* 3 minutes

As your cloud usage grows, it's increasingly important to stay organized. A good organization strategy helps you understand your cloud usage and can help you manage costs.

For example, as Tailwind Traders prototypes new ways to deploy its applications on Azure, it needs a way to mark its test environments so that it can easily identify and delete resources in these environments when they're no longer needed.

One way to organize related resources is to place them in their own subscriptions. You can also use resource groups to manage related resources. Resource *tags* are another way to organize resources. Tags provide extra information, or metadata, about your resources. This metadata is useful for:

* **Resource management** Tags enable you to locate and act on resources that are associated with specific workloads, environments, business units, and owners.
* **Cost management and optimization** Tags enable you to group resources so that you can report on costs, allocate internal cost centers, track budgets, and forecast estimated cost.
* **Operations management** Tags enable you to group resources according to how critical their availability is to your business. This grouping helps you formulate service-level agreements (SLAs). An SLA is an uptime or performance guarantee between you and your users.
* **Security** Tags enable you to classify data by its security level, such as *public* or *confidential*.
* **Governance and regulatory compliance** Tags enable you to identify resources that align with governance or regulatory compliance requirements, such as ISO 27001. Tags can also be part of your standards enforcement efforts. For example, you might require that all resources be tagged with an owner or department name.
* **Workload optimization and automation** Tags can help you visualize all of the resources that participate in complex deployments. For example, you might tag a resource with its associated workload or application name and use software such as Azure DevOps to perform automated tasks on those resources.

**How do I manage resource tags?**

You can add, modify, or delete resource tags through PowerShell, the Azure CLI, Azure Resource Manager templates, the REST API, or the Azure portal.

You can also manage tags by using Azure Policy. For example, you can apply tags to a resource group, but those tags aren't automatically applied to the resources within that resource group. You can use Azure Policy to ensure that a resource inherits the same tags as its parent resource group. You'll learn more about Azure Policy later in this module.

You can also use Azure Policy to enforce tagging rules and conventions. For example, you can 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.

**An example tagging structure**

A resource tag consists of a name and a value. You can assign one or more tags to each Azure resource.

After reviewing its business requirements, Tailwind Traders decides on the following tags.

**Name**

**Value**

**AppName**

The name of the application that the resource is part of.

**CostCenter**

The internal cost center code.

**Owner**

The name of the business owner who's responsible for the resource.

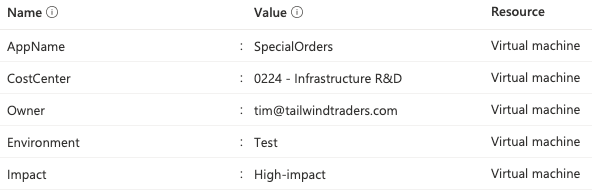
**Environment**

An environment name, such as "Prod," "Dev," or "Test."

**Impact**

How important the resource is to business operations, such as "Mission-critical," "High-impact," or "Low-impact."

Here's an example that shows these tags as they're applied to a virtual machine during provisioning.



The Tailwind Traders team can run queries, for example, from PowerShell or the Azure CLI, to list all resources that contain these tags.

Keep in mind that you don't need to enforce that a specific tag is present on all of 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.

**Next unit: Control and audit your resources by using Azure Policy**

# Control and audit your resources by using Azure Policy

Completed100 XP

* 5 minutes

In a previous exercise in this module, you identified your governance and business requirements. How do you ensure that your resources stay compliant? Can you be alerted if a resource's configuration has changed?

[Azure Policy](https://azure.microsoft.com/services/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 across all of your resource configurations so that those configurations stay compliant with corporate standards.

## How does Azure Policy define policies?

Azure Policy enables you to define both individual policies and groups of related policies, known as initiatives. Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies you've created. Azure Policy can also prevent noncompliant resources from being created.

Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that allows only a certain SKU (stock-keeping unit) size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment.

In some cases, Azure Policy can automatically remediate noncompliant resources and configurations to ensure the integrity of the state of the resources. For example, if all resources in a certain resource group should be tagged with **AppName** tag and a value of "SpecialOrders," Azure Policy will automatically reapply that tag if it was missing.

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

## Azure Policy in action

Implementing a policy in Azure Policy involves three tasks:

1. Create a policy definition.
2. Assign the definition to resources.
3. Review the evaluation results.

Let's examine each step in more detail.

### Task 1. Create a policy definition

A policy definition expresses what to evaluate and what action to take. For example, you could prevent VMs from being deployed in certain Azure regions. You also could audit your storage accounts to verify that they only accept connections from allowed networks.

Every policy definition has conditions under which it's enforced. A policy definition also has an accompanying effect that takes place when the conditions are met. Here are some example policy definitions:

* **Allowed virtual machine SKUs** This policy enables you to specify a set of VM SKUs that your organization can deploy.
* **Allowed locations** This policy enables you to restrict the locations that your organization can specify when it deploys resources. Its effect is used to enforce your geographic compliance requirements.
* **MFA should be enabled on accounts with write permissions on your subscription** This policy requires that multifactor authentication (MFA) be enabled for all subscription accounts with write privileges to prevent a breach of accounts or resources.
* **CORS should not allow every resource to access your web applications** Cross-origin resource sharing (CORS) is an HTTP feature that enables a web application running under one domain to access resources in another domain. For security reasons, modern web browsers restrict cross-site scripting by default. This policy allows only required domains to interact with your web app.
* **System updates should be installed on your machines** This policy enables Azure Security Center to recommend missing security system updates on your servers.

### Task 2. Assign the definition to resources

To implement your policy definitions, you assign definitions to resources. A policy assignment is a policy definition that takes place within a specific scope. This scope could be a management group (a collection of multiple subscriptions), a single subscription, or a resource group.

Policy assignments are inherited by all child resources within that scope. If a policy is applied to a resource group, that policy is applied to all resources within that resource group. You can exclude a subscope from the policy assignment if there are specific child resources you need to be exempt from the policy assignment.

### Task 3. Review the evaluation results

When a condition is evaluated against your existing resources, each resource is marked as compliant or noncompliant. You can review the noncompliant policy results and take any action that's needed.

Policy evaluation happens about once per hour. If you make changes to your policy definition and create a policy assignment, that policy is evaluated over your resources within the hour.

## What are Azure Policy initiatives?

An Azure Policy initiative is a way of grouping related policies together. The initiative definition contains all of the policy definitions to help track your compliance state for a larger goal.

For example, Azure Policy includes an initiative named **Enable Monitoring in Azure Security Center**. Its goal is to monitor all of the available security recommendations for all Azure resource types in Azure Security Center.

Under this initiative, the following policy definitions are included:

* **Monitor unencrypted SQL Database in Security Center** This policy monitors for unencrypted SQL databases and servers.
* **Monitor OS vulnerabilities in Security Center** This policy monitors servers that don't satisfy the configured OS vulnerability baseline.
* **Monitor missing Endpoint Protection in Security Center** This policy monitors for servers that don't have an installed endpoint protection agent.

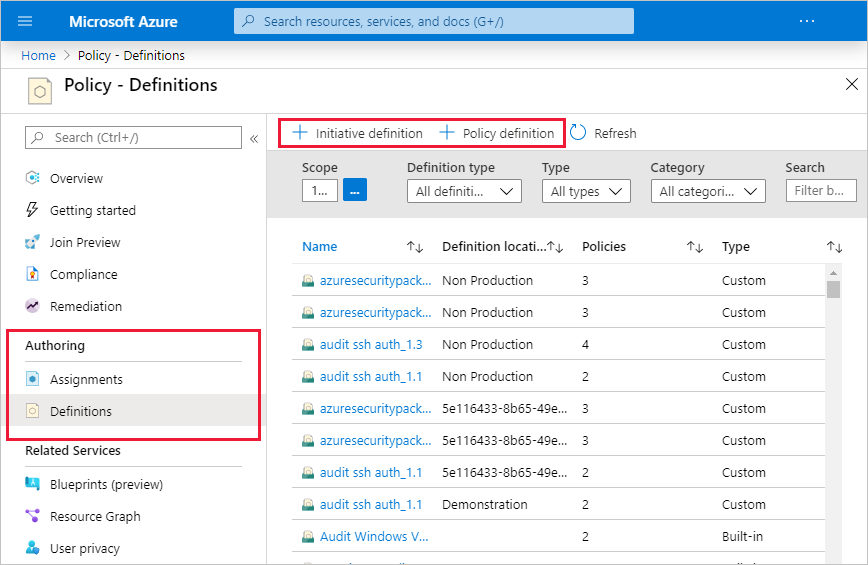
In fact, the **Enable Monitoring in Azure Security Center** initiative contains over 100 separate policy definitions.

Azure Policy also includes initiatives that support regulatory compliance standards, such as HIPAA and ISO 27001.

### How do I define an initiative?

You define initiatives by using the Azure portal or command-line tools. From the Azure portal, you can search the list of built-in initiatives that are built into Azure. You also can create your own custom policy definition.

The following image shows a few example Azure Policy initiatives in the Azure portal.



### How do I assign an initiative?

Like a policy assignment, an initiative assignment is an initiative definition that's assigned to a specific scope of a management group, a subscription, or a resource group.

Even if you have only a single policy, an initiative enables you to increase the number of policies over time. Because the associated initiative remains assigned, it's easier to add and remove policies without the need to change the policy assignment for your resources.

## Next unit: Exercise - Restrict deployments to a specific location by using Azure Policy

**Exercise - Restrict deployments to a specific location by using Azure Policy**

Completed100 XP

* 8 minutes

In this exercise, you create a policy in Azure Policy that restricts the deployment of Azure resources to a specific location. You verify the policy by attempting to create a storage account in a location that violates the policy.

Tailwind Traders wants to limit the location where resources can be deployed to the **East US** region. It has two reasons:

* **Improved cost tracking** To track costs, Tailwind Traders uses different subscriptions to track deployments to each of its regional locations. The policy will ensure that all resources are deployed to the **East US** region.
* **Adhere to data residency and security compliance** Tailwind Traders must adhere to a compliance rule that states where customer data can be stored. Here, customer data must be stored in the **East US** region.

Recall that you can assign a policy to a management group, a single subscription, or a resource group. Here, you assign the policy to a resource group so that policy doesn't affect any other resources in your Azure subscription.

**Important**

You need your own [**Azure subscription**](https://azure.microsoft.com/free/) to complete the exercises in this module. If you don't have an Azure subscription, you can still read along.

**Create the resource group**

Here you create a resource group that's named **my-test-rg**. This is the resource group to which you'll apply your location policy.

For learning purposes, you use the same resource group name that you used in the previous exercise. You can use the same name because you deleted the previous resource group.

1. Go to the [Azure portal](https://portal.azure.com/), and sign in.
2. Select **Create a resource**.
3. Enter **resource group** in the search box, and press Enter.
4. If you're taken to a search results pane, select **Resource group** from the results.
5. Select **Create**. Then, enter the following values for each setting.

**Setting**

**Value**

**Subscription**

*(Your Azure subscription)*

**Subscription > Resource group**

**my-test-rg**

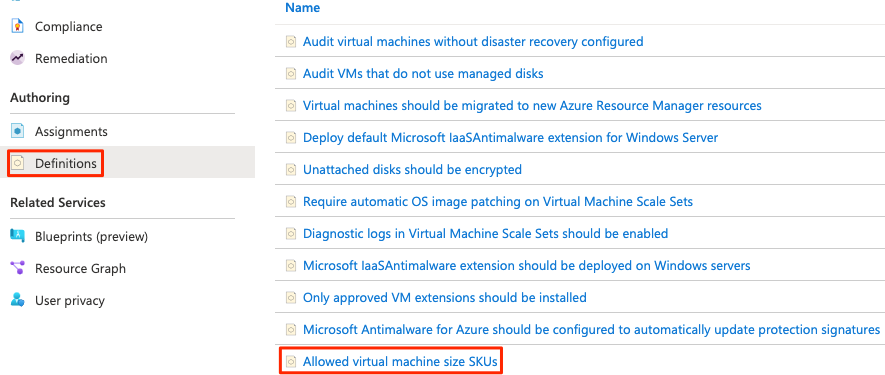
**Region**

**(US) East US**

1. Select **Review + create**, and then select **Create**.

**Explore predefined policies**

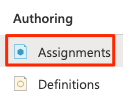
Before you configure your location policy, let's take a brief look at some predefined policies. As an example, you'll look at policies that relate to Azure Compute services.

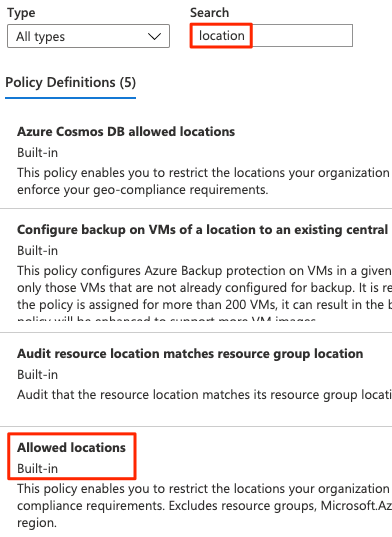
1. From the Azure portal, at the top of the page, select **Home** to return to the start page.
2. At the top of the page, enter **policy** in the search bar. Then, select **Policy** from the list of results to access Azure Policy.
3. Under **Authoring**, select **Definitions**.
4. From the **Category** dropdown list, select only **Compute**. Notice that the **Allowed virtual machine SKUs** definition enables you to specify a set of virtual machine SKUs that your organization can deploy. 

As an optional step, explore any other policies or categories that interest you.

**Configure the location policy**

Here you configure the allowed location policy by using Azure Policy. Then you assign that policy to your resource group. To do so:

1. From the **Policy** pane, under **Authoring**, select **Assignments**.  
     
   An assignment is a policy that has been assigned to take place within a specific scope. For example, a definition could be assigned to the subscription scope.
2. Select **Assign Policy**.  
   A screenshot of the Azure portal showing the Assign policy menu item.  
   You're taken to the **Assign policy** pane.
3. Under **Scope**, select the ellipsis.  
   From the dialog box that appears, set:
   1. **Subscription** field to your Azure subscription.
   2. **Resource Group** field to **my-test-rg**.
   3. Select **Select**.
4. Under **Policy definition**, select the ellipsis.
   1. In the search bar, enter *location*.
   2. Select the **Allowed locations** definition.
   3. Select **Select**.

  
This policy definition specifies the location into which all resources must be deployed. If a different location is chosen, deployment will fail.

1. Select **Next** to move to the **Parameters** tab.
2. From the **Allowed locations** dropdown list, select **East US**.
3. Select **Review + create**, and then select **Create**.  
   You see that the **Allowed locations** policy assignment is now listed on the **Policy | Assignments** pane. It enforces the policy on the **my-test-rg** resource group.  
   A screenshot of the Azure portal showing the Allowed locations policy assignment listed in the Policy Assignments pane.

**Verify the location policy**

Here you attempt to add a storage account to your resource group at a location that violates your location policy.

1. From the Azure portal, at the top of the page, select **Home** to return to the start page.
2. Select **Create a resource**.
3. Enter **storage account** in the search box, and press Enter.
4. If you're taken to a search results pane, select **Storage account** from the results.
5. Select **Create**. Then, enter the following values for each setting.

**Note**

Replace **NNN** with a series of numbers. Numbers help to ensure that your storage account name is unique.

**Setting**

**Value**

**Subscription**

*(Your Azure subscription)*

**Subscription > Resource group**

**my-test-rg**

**Storage account name**

**mysaNNN**

**Location**

**(Asia Pacific) Japan East**

**Performance**

**Standard**

**Account kind**

**StorageV2 (general purpose v2)**

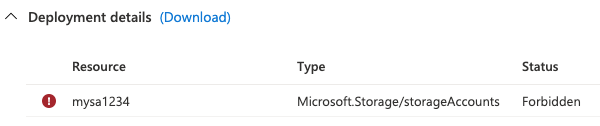
**Redundancy**

**Locally redundant storage (LRS)**

**Access tier (default)**

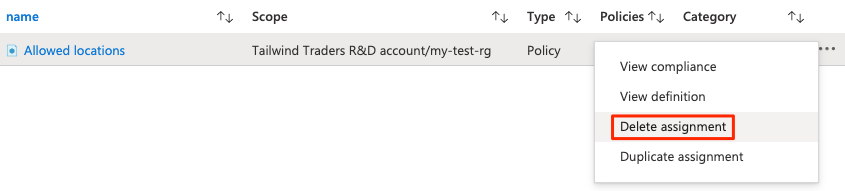
**Hot**

If you previously selected **Japan East** in your location policy, select a different region from the list.

1. Select **Review + create**, and then select **Create**.  
   You see a message that states that the deployment failed because of the policy violation. You also see the deployment details.  
   Here's an example that shows the deployment details for a storage account named **mysa1234**.  
   

**Delete the policy assignment**

You no longer need your policy assignment. Here you remove it from your subscription.

1. From the Azure portal, select **Home** > **Policy**.
2. Under **Authoring**, select **Assignments**.
3. On the **Allowed locations** row, select the ellipsis. Then, select **Delete assignment**. When prompted, select **Yes**.   
   You see that the **Allowed locations** policy assignment no longer exists.

As an optional step, you can try to create the storage account a second time to verify that the policy is no longer in effect.

**Delete the resource group**

You no longer need your resource group. Here you remove it from your subscription.

1. From the Azure portal, select **Home** > **Resource groups** > **my-test-rg** to go to your resource group.
2. Select **Overview**, and then select **Delete resource group**.
3. At the prompt, enter **my-test-rg**, and then select **OK**. The deletion operation might take a few moments to complete.
4. After the operation completes, select **Home** > **Resource groups**. You see that the **my-test-rg** resource group no longer exists in your account.

Great work! You've successfully applied a policy by using Azure Policy to restrict deployments of Azure resources to specific locations. You can now apply the policies that you need at the management group, subscription, or resource group level.

**Next unit: Govern multiple subscriptions by using Azure Blueprints**

**Govern multiple subscriptions by using Azure Blueprints**

Completed100 XP

* 4 minutes

So far, you've explored a number of Azure features that can help you implement your governance decisions, monitor the compliance of your cloud resources, and control access and protect critical resources from accidental deletion.

What happens when your cloud environment starts to grow beyond just one subscription? How can you scale the configuration of these features, knowing they need to be enforced for resources in new subscriptions?

Instead of having to configure features like Azure Policy for each new subscription, with [Azure Blueprints](https://azure.microsoft.com/services/blueprints) you can define a repeatable set of governance tools and standard Azure resources that your organization requires. In this way, development teams can rapidly build and deploy new environments with the knowledge that they're building within organizational compliance with a set of built-in components that speed the development and deployment phases.

Azure Blueprints orchestrates the deployment of various resource templates and other artifacts, such as:

* Role assignments
* Policy assignments
* Azure Resource Manager templates
* Resource groups

**Azure Blueprints in action**

When you form a cloud center of excellence team or a cloud custodian team, that team can use Azure Blueprints to scale their governance practices throughout the organization.

Implementing a blueprint in Azure Blueprints involves these three steps:

1. Create an Azure blueprint.
2. Assign the blueprint.
3. Track the blueprint assignments.

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.

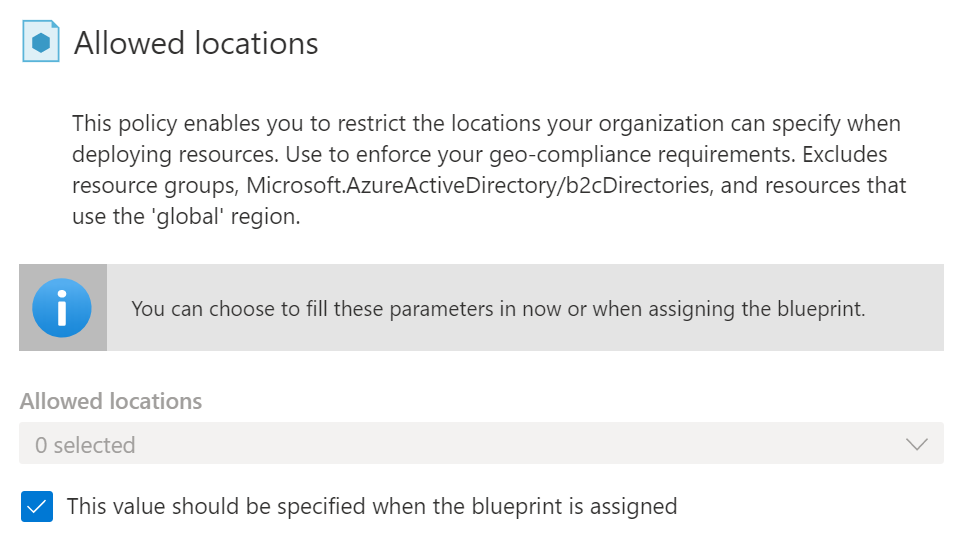
Blueprints are also versioned. Versioning enables you to track and comment on changes to your blueprint.

**What are blueprint artifacts?**

Each component in the blueprint definition is known as an *artifact*.

It is possible for artifacts to have no additional parameters (configurations). An example is the **Deploy threat detection on SQL servers** policy, which requires no additional configuration.

Artifacts can also contain one or more parameters that you can configure. The following screenshot shows the **Allowed locations** policy. This policy includes a parameter that specifies the allowed locations.



You can specify a parameter's value when you create the blueprint definition or when you assign the blueprint definition to a scope. In this way, you can maintain one standard blueprint but have the flexibility to specify the relevant configuration parameters at each scope where the definition is assigned.

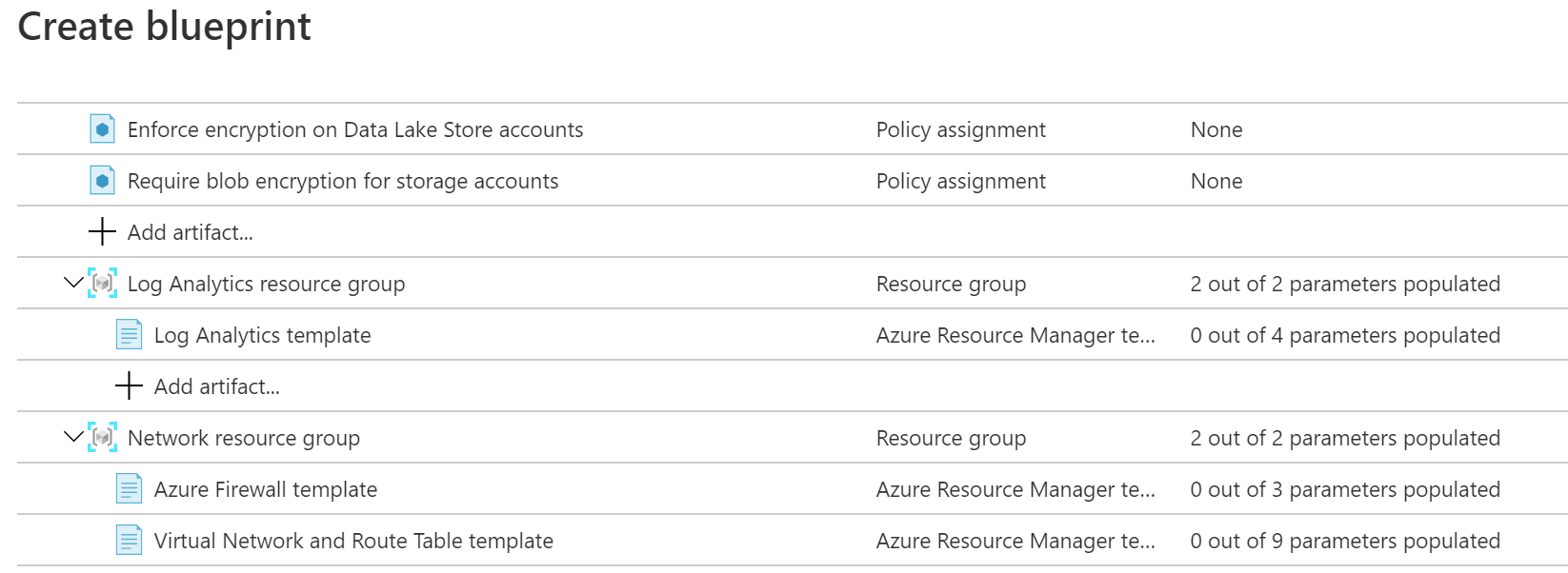
**How will Tailwind Traders use Azure Blueprints for ISO 27001 compliance?**

[ISO 27001](https://www.iso.org/isoiec-27001-information-security.html) is a standard that applies to the security of IT systems, published by the International Organization for Standardization. As part of its quality process, Tailwind Traders wants to certify that it complies with this standard. Azure Blueprints has several built-in blueprint definitions that relate to ISO 27001.

As an IT administrator, you decide to investigate the **ISO 27001: Shared Services Blueprint** definition. Here's an outline of your plan.

1. Define a management group that's named **PROD-MG**. Recall that a management group manages access, policies, and compliance across multiple Azure subscriptions. Every new Azure subscription is added to this management group when the subscription is created.
2. Create a blueprint definition that's based on the **ISO 27001: Shared Services Blueprint** template. Then publish the blueprint.
3. Assign the blueprint to your **PROD-MG** management group.

The following image shows artifacts that are created when you run an ISO 27001 blueprint from a template.



You see that the blueprint template contains policy assignments, Resource Manager templates, and resource groups. The blueprint deploys these artifacts to any existing subscriptions within the **PROD-MG** management group. The blueprint also deploys these artifacts to any new subscriptions as they're created and added to the management group.

**Next unit: Accelerate your cloud adoption journey by using the Cloud Adoption Framework for Azure**

# Accelerate your cloud adoption journey by using the Cloud Adoption Framework for Azure

Completed100 XP

* 3 minutes

The [Cloud Adoption Framework for Azure](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/) provides you with proven guidance to help with your cloud adoption journey. The Cloud Adoption Framework helps you create and implement the business and technology strategies needed to succeed in the cloud.

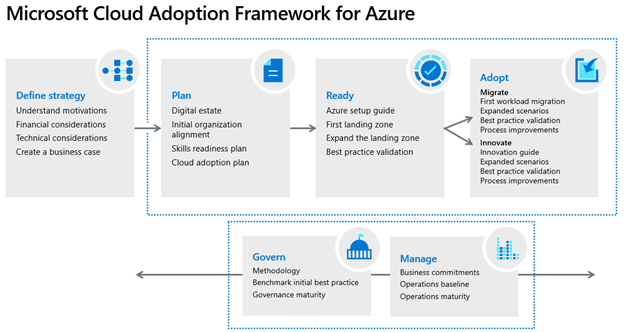
Tailwind Traders needs to control its cloud environment so that it complies with several industry standards, but it's not sure where to start. It has existing business requirements, and it understands how these requirements relate to its on-premises workloads. These requirements also must be met by any workloads it runs in the cloud.

You've been tasked with investigating what's available on Azure and to define and implement the governance strategy for Tailwind Traders. You decide to start with the Cloud Adoption Framework.

## What's in the Cloud Adoption Framework?

As mentioned in the video, Cloud Adoption Framework consists of tools, documentation, and proven practices. The Cloud Adoption Framework includes these stages:

1. Define your strategy.
2. Make a plan.
3. Ready your organization.
4. Adopt the cloud.
5. Govern and manage your cloud environments.



The govern stage focuses on cloud governance. You can refer back to the Cloud Adoption Framework for recommended guidance as you build your cloud governance strategy.

To help build your adoption strategy, the Cloud Adoption Framework breaks out each stage into further exercises and steps. Let's take a brief look at each stage.

### Define your strategy

Here, you answer why you're moving to the cloud and what you want to get out of cloud migration. Do you need to scale to meet demand or reach new markets? Will it reduce costs or increase business agility? When you define your cloud business strategy, you should understand [cloud economics](https://azure.microsoft.com/overview/cloud-economics) and consider business impact, turnaround time, global reach, performance, and more.

Here are the steps in this stage.



**Define and document your motivations**: Meeting with stakeholders and leadership can help you answer why you're moving to the cloud.



**Document business outcomes**: Meet with leadership from your finance, marketing, sales, and human resource groups to help you document your goals.



**Evaluate financial considerations**: Measure objectives and identify the return expected from a specific investment.



**Understand technical considerations**: Evaluate those technical considerations through the selection and completion of your first technical project.

### Make a plan

Here, you build a plan that maps your aspirational goals to specific actions. A good plan helps ensure that your efforts map to the desired business outcomes.

Here are the steps in this stage.

**Digital estate**: Create an inventory of the existing digital assets and workloads that you plan to migrate to the cloud.



**Initial organizational alignment**: Ensure that the right people are involved in your migration efforts, both from a technical standpoint as well as from a cloud governance standpoint.



**Skills readiness plan**: Build a plan that helps individuals build the skills they need to operate in the cloud.



**Cloud adoption plan**: Build a comprehensive plan that brings together the development, operations, and business teams toward a shared cloud adoption goal.



### Ready your organization

Here, you create a landing zone, or an environment in the cloud to begin hosting your workloads.

Here are the steps in this stage.



**Azure setup guide**: Review the Azure setup guide to become familiar with the tools and approaches you need to use to create a landing zone.



**Azure landing zone**: Begin to build out the Azure subscriptions that support each of the major areas of your business. A landing zone includes cloud infrastructure as well as governance, accounting, and security capabilities.



**Expand the landing zone**: Refine your landing zone to ensure that it meets your operations, governance, and security needs.



**Best practices**: Start with recommended and proven practices to help ensure that your cloud migration efforts are scalable and maintainable.

### Adopt the cloud

Here, you begin to migrate your applications to the cloud. Along the way, you might find ways to modernize your applications and build innovative solutions that use cloud services.

The Cloud Adoption Framework breaks this stage into two parts: migrate and innovate.

**Migrate**: Here are the steps in the migrate part of this stage.

**Migrate your first workload**: Use the Azure migration guide to deploy your first project to the cloud.



**Migration scenarios**: Use additional in-depth guides to explore more complex migration scenarios.



**Best practices**: Check in with the Azure cloud migration best practices checklist to verify that you're following recommended practices.



**Process improvements**: Identify ways to make the migration process scale while requiring less effort.



**Innovate**: Here are the steps in the innovate part of this stage.



**Business value consensus**: Verify that investments in new innovations add value to the business and meet customer needs.



**Azure innovation guide**: Use this guide to accelerate development and build a minimum viable product (MVP) for your idea.



**Best practices**: Verify that your progress maps to recommended practices before you move forward.



**Feedback loops**: Check in frequently with your customers to verify that you're building what they need.

### Govern and manage your cloud environments

Here, you begin to form your cloud governance and cloud management strategies. As the cloud estate changes over time, so do cloud governance processes and policies. You need to create resilient solutions that are constantly optimized.

**Govern**: Here are the steps in the govern part of this stage.

**Methodology**: Consider your end state solution. Then define a methodology that incrementally takes you from your first steps all the way to full cloud governance.



**Benchmark**: Use the [governance benchmark tool](https://cafbaseline.com/) to assess your current state and future state to establish a vision for applying the framework.



**Initial governance foundation**: Create an MVP that captures the first steps of your governance plan.



**Improve the initial governance foundation**: Iteratively add governance controls that address tangible risks as you progress toward your end state solution.



**Manage**: Here are the steps in the manage part of this stage.



**Establish a management baseline**: Define your minimum commitment to operations management. A management baseline is the minimum set of tools and processes that should be applied to every asset in an environment.



**Define business commitments**: Document supported workloads to establish operational commitments with the business and agree on cloud management investments for each workload.



**Expand the management baseline**: Apply recommended best practices to iterate on your initial management baseline.



**Advanced operations and design principles**: For workloads that require a higher level of business commitment, perform a deeper architecture review to deliver on your resiliency and reliability commitments.

## Next unit: Create a subscription governance strategy

**Create a subscription governance strategy**

Completed100 XP

* 3 minutes

At the beginning of any cloud governance implementation, you identify a cloud organization structure that meets your business needs. This step often involves forming a *cloud center of excellence team* (also called a *cloud enablement team* or a *cloud custodian team*). This team is empowered to implement governance practices from a centralized location for the entire organization.

Teams often start their Azure governance strategy at the subscription level. There are three main aspects to consider when you create and manage subscriptions: billing, access control, and subscription limits.

Let's look at each of these aspects in more detail.

**Billing**

You can create one billing report per subscription. If you have multiple departments and need to do a "chargeback" of cloud costs, one possible solution is to organize subscriptions by department or by project.

Resource tags can also help. You'll explore tags later in this module. When you define how many subscriptions you need and what to name them, take into account your internal billing requirements.

**Access control**

A subscription is a deployment boundary for Azure resources. Every subscription is associated with an Azure Active Directory tenant. Each tenant provides administrators the ability to set granular access through defined roles by using Azure role-based access control.

When you design your subscription architecture, consider the deployment boundary factor. For example, do you need separate subscriptions for development and for production environments? With separate subscriptions, you can control access to each one separately and isolate their resources from one another.

**Subscription limits**

Subscriptions also have some resource limitations. For example, the maximum number of network Azure ExpressRoute circuits per subscription is 10. Those limits should be considered during your design phase. If you'll need to exceed those limits, you might need to add more subscriptions. If you hit a hard limit maximum, there's no flexibility to increase it.

Management groups are also available to assist with managing subscriptions. A management group manages access, policies, and compliance across multiple Azure subscriptions. You'll learn more about management groups later in this module.

**Next unit: Knowledge check**

**Knowledge check**

Completed200 XP

* 3 minutes

Consider the following scenario.

Tailwind Traders has created environments for development and testing for its e-commerce system.

Here's a diagram that shows the basic compute, database, and networking components found in each environment.

These environments provide a way for the team to build and test new application features. If you've gone through the [Plan and manage your Azure costs](https://learn.microsoft.com/en-us/training/modules/plan-manage-azure-costs/) module, then you've already seen this layout.

Although the development and test teams report to different departments, both environments exist under the same Azure subscription.

The IT manager wants to implement governance controls to help ensure that only authorized users can access these systems. Having these controls in place will also help them track and manage operating costs.

Choose the best response for each question. Then select **Check your answers**.

**Check your knowledge**

Top of Form

**1.**

How can Tailwind Traders allow some users to control the virtual machines in each environment but prevent them from modifying networking and other resources in the same resource group or Azure subscription?



Create a role assignment through Azure role-based access control (Azure RBAC).

**That's correct. Azure RBAC enables you to create roles that define access permissions. You might create one role that limits access only to virtual machines and a second role that provides administrators with access to everything.**



Create a policy in Azure Policy that audits resource usage.



Split the environment into separate resource groups.

**2.**

Which is the best way for Tailwind Traders to ensure that the team deploys only cost-effective virtual machine SKU sizes?



Create a policy in Azure Policy that specifies the allowed SKU sizes.

**That's correct. After you enable this policy, that policy is applied when you create new virtual machines or resize existing ones. Azure Policy also evaluates any current virtual machines in your environment.**



Periodically inspect the deployment manually to see which SKU sizes are used.



Create an Azure RBAC role that defines the allowed virtual machine SKU sizes.

**3.**

Which is likely the best way for Tailwind Traders to identify which billing department each Azure resource belongs to?



Track resource usage in a spreadsheet.



Split the deployment into separate Azure subscriptions, where each subscription belongs to its own billing department.



Apply a tag to each resource that includes the associated billing department.

**That's correct. Tags provide extra information, or metadata, about your resources. The team might create a tag that's named BillingDept whose value would be the name of the billing department. You can use Azure Policy to ensure that the proper tags are assigned when resources are provisioned.**

Bottom of Form

**Next unit: Summary**

**Summary**

Completed100 XP

* 2 minutes

You've been tasked with defining and implementing the governance strategy for Tailwind Traders.

Cloud governance requires good analysis and requirement gathering. Luckily, the Cloud Adoption Framework for Azure can help you define and implement your governance strategy. There are several services and features in Azure to support these efforts:

* Azure role-based access control (Azure RBAC) enables you to create roles that define access permissions.
* Resource locks prevent resources from being accidentally deleted or changed.
* Resource tags provide extra information, or metadata, about your resources.
* Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources.
* Azure Blueprints enables you to define a repeatable set of governance tools and standard Azure resources that your organization requires.

With these points in mind, you're ready to take the next step toward building a good cloud governance strategy.

**Learn more**

The [Control and organize Azure resources with Azure Resource Manager](https://learn.microsoft.com/en-us/training/modules/control-and-organize-with-azure-resource-manager/) module is a good next step. There you'll go deeper on how to use Azure Resource Manager to organize resources, enforce standards, and protect critical assets from deletion.

Here are additional resources to help you go further:

* Get started with the [Cloud Adoption Framework for Azure](https://learn.microsoft.com/en-us/training/modules/microsoft-cloud-adoption-framework-for-azure/).
* Learn more about [Azure subscription and service limits, quotas, and constraints](https://learn.microsoft.com/en-us/azure/azure-subscription-service-limits).
* Review the complete list of [Azure built-in roles](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles/) for Azure RBAC.
* To learn how Azure Policy can enforce tagging rules and conventions, see [Assign policies for tag compliance](https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/tag-policies/).
* For recommendations on how to implement your own tagging strategy, see [Resource naming and tagging decision guide](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/decision-guides/resource-tagging).
* Explore additional [Azure Policy samples](https://learn.microsoft.com/en-us/azure/governance/policy/samples).
* For a more advanced topic, see [Creating a custom policy definition](https://learn.microsoft.com/en-us/azure/governance/policy/tutorials/create-custom-policy-definition/). This tutorial gets you started.

**Module complete:**