Introduction

- 1 minute

In this module, you'll be introduced to some of the features and tools you can use to help with governance of your Azure environment. You'll also learn about tools you can use to help keep resources in compliance with corporate or regulatory requirements.

Learning objectives

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

- Describe the purpose of Azure Blueprints
- Describe the purpose of Azure Policy
- Describe the purpose of resource locks
- Describe the purpose of the Service Trust portal

Next unit: Describe the purpose of Azure Blueprints

Describe the purpose of Azure Blueprints

Completed100 XP

- 3 minutes

What happens when your cloud starts to grow beyond just one subscription or environment? How can you scale the configuration of features? How can you enforce settings and policies in new subscriptions?

Azure Blueprints lets you standardize cloud subscription or environment deployments. Instead of having to configure features like Azure Policy for each new subscription, with Azure Blueprints you can define repeatable settings and policies that are applied as new subscriptions are created. Need a new test/dev environment? Azure Blueprints lets you deploy a new Test/Dev environment with security and compliance settings already configured. In this way, development teams can rapidly build and deploy new environments with the knowledge that they're building within organizational requirements.

What are 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.

Azure Blueprints deploy a new environment based on all of the requirements, settings, and configurations of the associated artifacts. Artifacts can include things such as:

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

How do Azure Blueprints help monitor deployments?

Azure Blueprints are version-able, allowing you to create an initial configuration and then make updates later on and assign a new version to the update. With versioning, you can make small updates and keep track of which deployments used which configuration set.

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.

Next unit: Describe the purpose of Azure Policy

Describe the purpose of Azure Policy

Completed100 XP

- 3 minutes

How do you ensure that your resources stay compliant? Can you be alerted if a resource's configuration has changed?

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 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 Policies can be set at each level, enabling you to set policies on a specific resource, resource group, subscription, and so on. Additionally, Azure Policies are inherited, so if you set a policy at a high level, it will automatically be applied to all of the groupings that fall within the parent. For example, if you set an Azure Policy on a resource group, all resources created within that resource group will automatically receive the same policy.

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 size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment, including VMs that were created before the policy was created.

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

However, you still retain full control of your environment. If you have a specific resource that you don’t want Azure Policy to automatically fix, you can flag that resource as an exception – and the policy won’t automatically fix that resource.

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.

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 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.

Next unit: Describe the purpose of resource locks

Describe the purpose of resource locks

Completed100 XP

- 3 minutes

A resource lock 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. Resource locks prevent resources from being deleted or updated, depending on the type of lock. Resource locks can be applied to individual resources, resource groups, or even an entire subscription. Resource locks are inherited, meaning that if you place a resource lock on a resource group, all of the resources within the resource group will also have the resource lock applied.

Types of Resource Locks

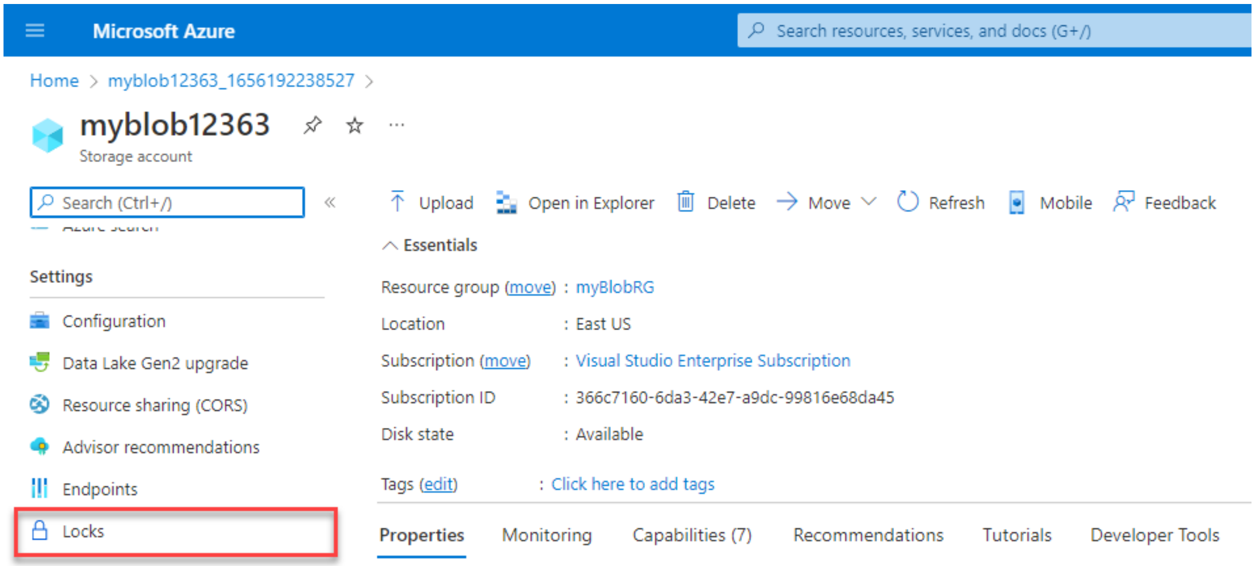
There are two types of resource locks, one that prevents users from deleting and one that prevents users from changing or deleting a resource.

- Delete means authorized users can still read and modify a resource, but they can't delete the resource.
- ReadOnly means authorized users can read a resource, but they can't delete or update the resource. Applying this lock is similar to restricting all authorized users to the permissions granted by the Reader role.

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 Settings pane in the Azure portal.



Resource group (move) : myBlobRG
Location : East US
Subscription (move) : Visual Studio Enterprise Subscription
Subscription ID : 366c7160-6da3-42e7-a9dc-99816e68da45
Disk state : Available
Tags (edit) : Click here to add tags

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. 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.

Next unit: Exercise - Configure a resource lock

Exercise - Configure a resource lock

Completed 100 XP

- 15 minutes

In this exercise, you'll create a resource and configure a resource lock. Storage accounts are one of the easiest types of resource locks to quickly see the impact, so you'll use a storage account for this exercise.

This exercise is a Bring your own subscription exercise, meaning you'll need to provide your own Azure subscription to complete the exercise. Don't worry though, the entire exercise can be completed for free with the 12 month free services when you sign up for an Azure account.

For help with signing up for an Azure account, see the Create an Azure account learning module.

Once you've created your free account, follow the steps below. If you don't have an Azure account, you can review the steps to see the process for adding a simple resource lock to a resource.

Task 1: Create a resource

In order to apply a resource lock, you have to have a resource created in Azure. The first task focuses on creating a resource that you can then lock in subsequent tasks.

1. Sign in to the Azure portal at https://portal.azure.com
2. Select Create a resource.
3. Under Categories, select Storage.
4. Unders Storage Account, select Create.
5. On the Basics tab of the Create storage account blade, fill in the following information. Leave the defaults for everything else.

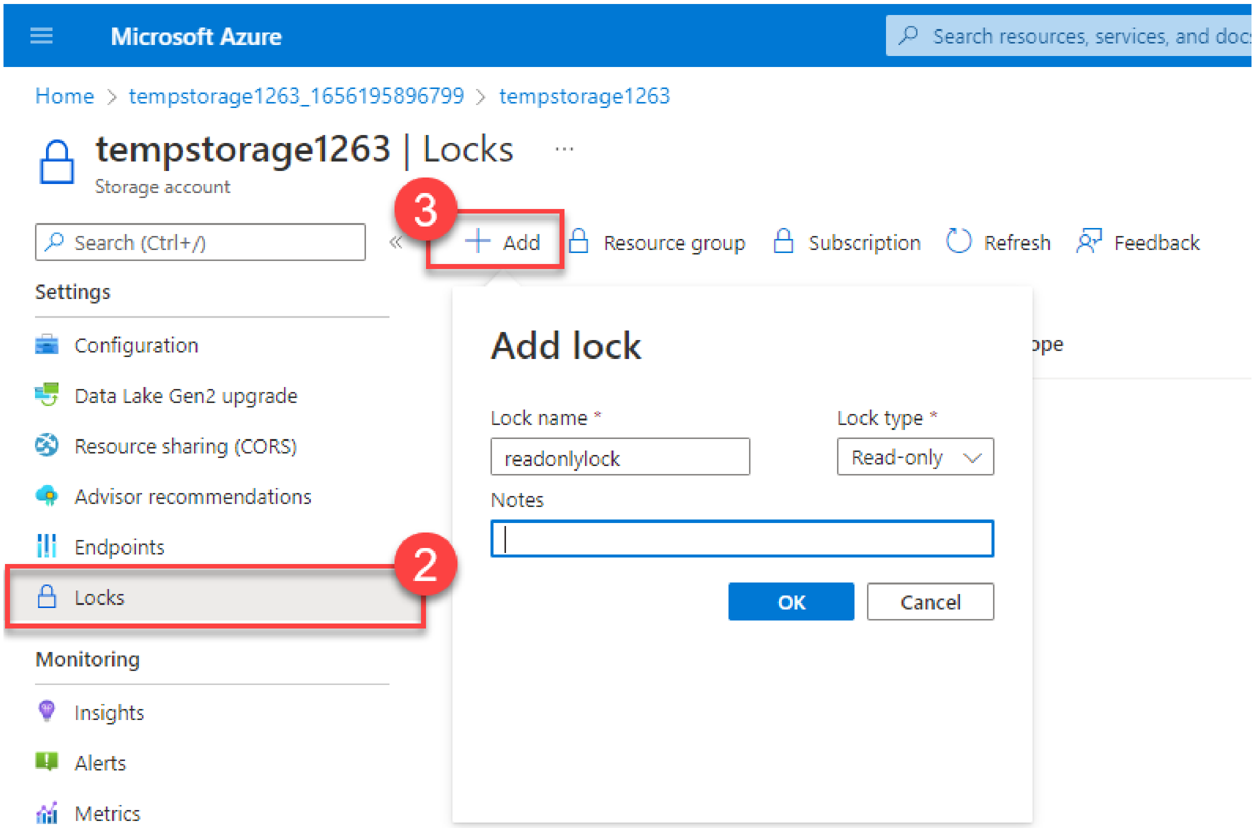
Setting	**Value**
Resource group	Create new
Storage account name	enter a unique storage account name
Location	default
Performance	Standard
Redundancy	Locally redundant storage (LRS)

6. Select Review + Create to review your storage account settings and allow Azure to validate the configuration.
7. Once validated, select Create. Wait for the notification that the account was successfully created.
8. Select Go to resource.

Task 2: Apply a read-only resource lock

In this task you apply a read-only resource lock to the storage account. What impact do you think that will have on the storage account?

1. Scroll down until you find the Settings section of the blade on the left of the screen.
2. Select Locks.
3. Select + Add.



Lock name * readonlylock

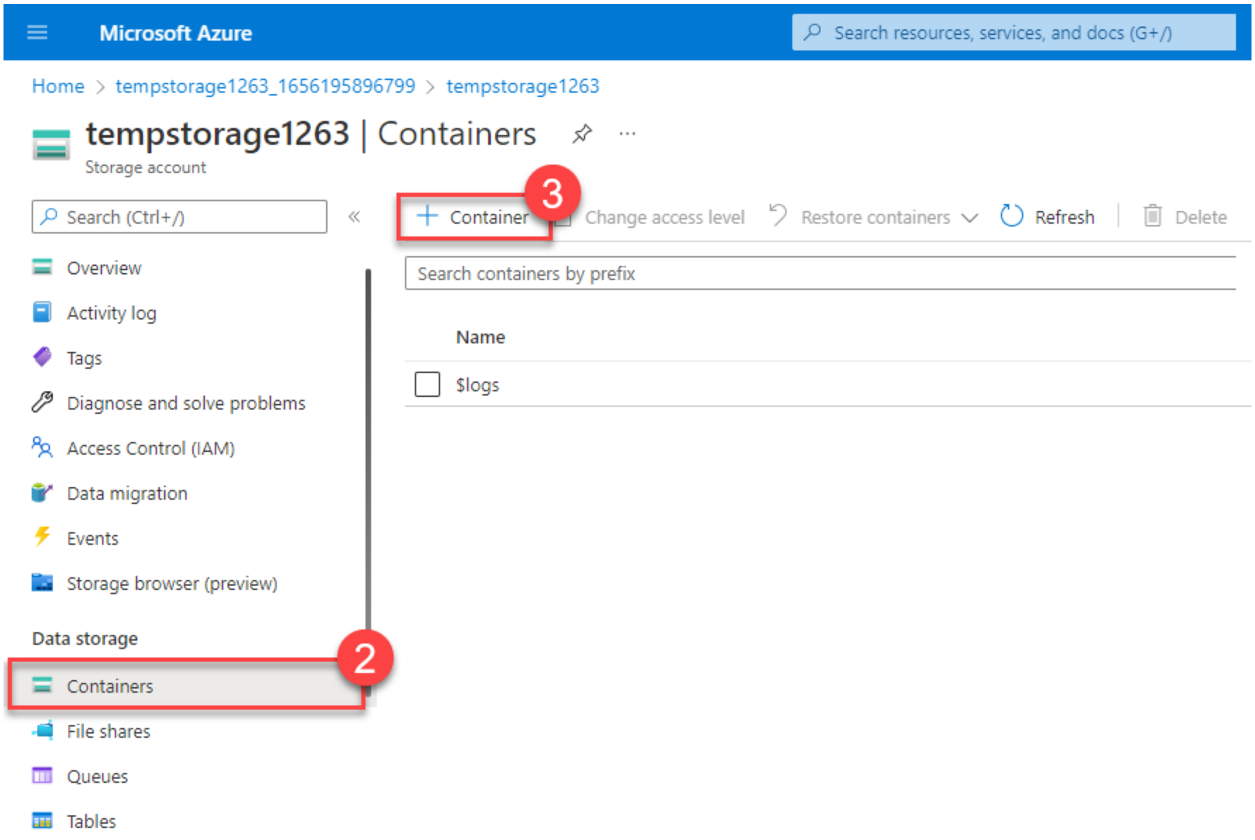
Lock type * Read-only

4. Enter a Lock name.
5. Verify the Lock type is set to Read-only.
6. Select OK.

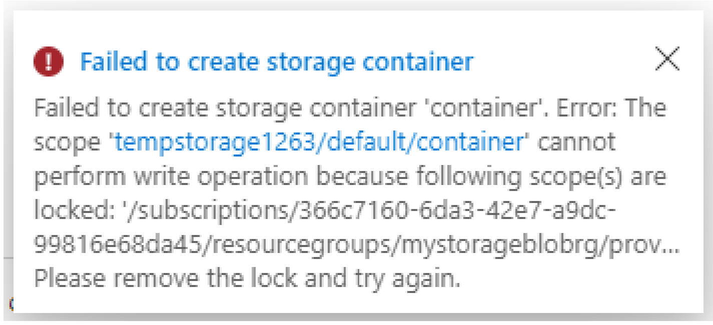
Task 3: Add a container to the storage account

In this task, you add a container to the storage account, this container is where you can store your blobs.

1. Scroll up until you find the Data storage section of the blade on the left of the screen.
2. Select Containers.
3. Select + Container.



4. Enter a container name and select Create.
5. You should receive an error message: Failed to create storage container.



Failed to create storage container

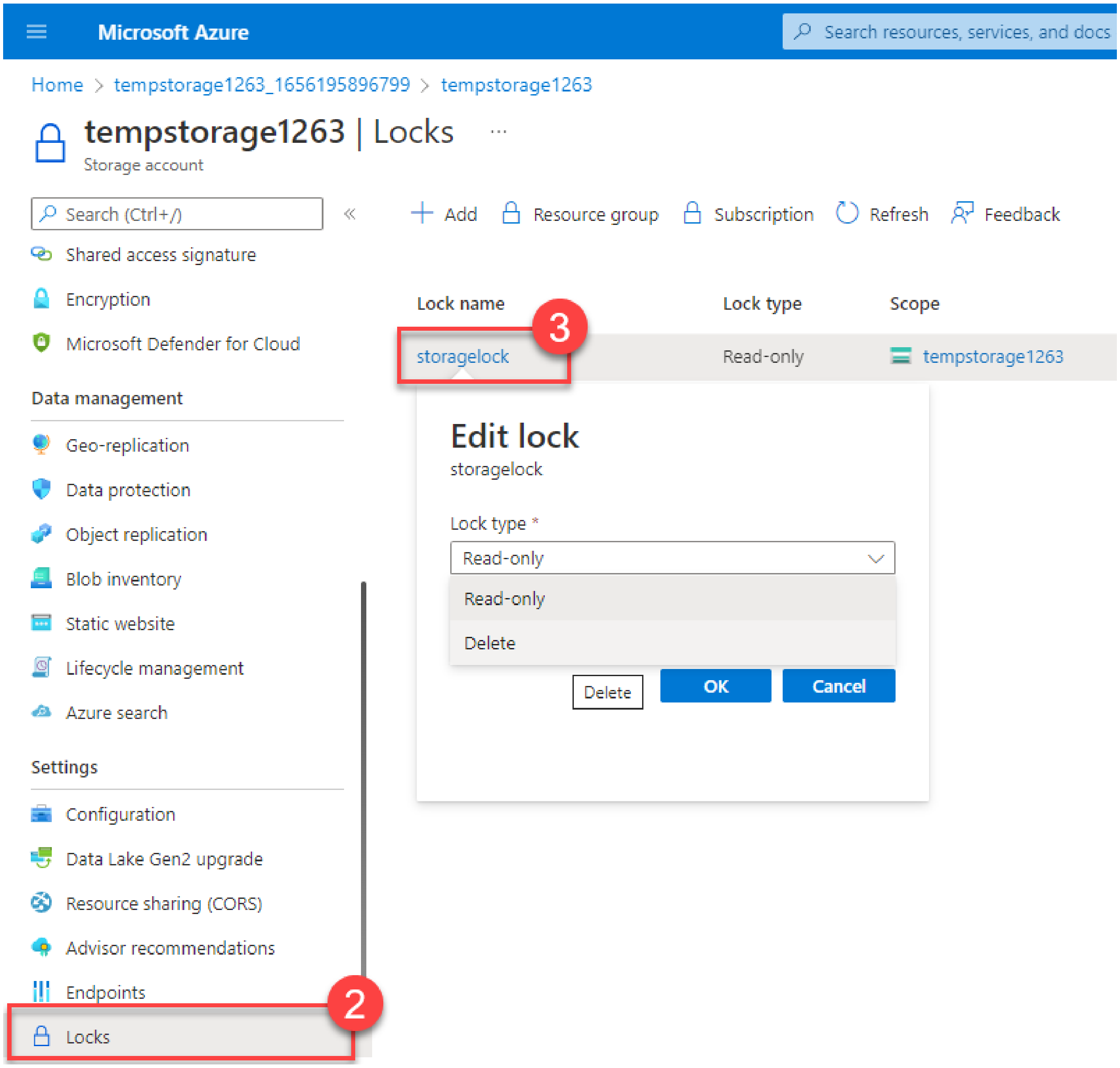
Failed to create storage container 'container'. Error: The scope 'tempstorage1263/default/container' cannot perform write operation because following scope(s) are locked: '/subscriptions/366c7160-6da3-42e7-a9dc-99816e68da45/resourcegroups/mystorageblobrg/prov... Please remove the lock and try again.

Note

The error message lets you know that you couldn't create a storage container because a lock is in place. The read-only lock prevents any create or update operations on the storage account, so you're unable to create a storage container.

Task 4: Modify the resource lock and create a storage container

1. Scroll down until you find the Settings section of the blade on the left of the screen.
2. Select Locks.
3. Select the read-only resource lock you created.
4. Change the Lock type to Delete and select OK.



Lock name	Lock type	Scope
storagelock	Read-only	tempstorage1263

Edit lock
storagelock

Lock type *
Read-only
- Read-only
- Delete

5. Scroll up until you find the Data storage section of the blade on the left of the screen.

6. Select Containers.
7. Select + Container.
8. Enter a container name and select Create.
9. Your storage container should appear in your list of containers.

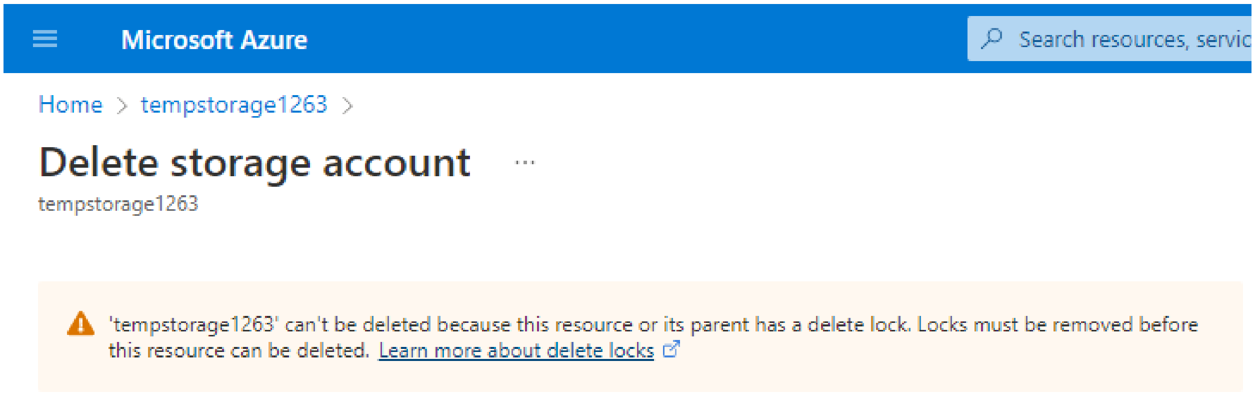
You can now understand how the read-only lock prevented you from adding a container to your storage account. Once the lock type was changed (you could have removed it instead), you were able to add a container.

Task 5: Delete the storage account

You'll actually do this last task twice. Remember that there is a delete lock on the storage account, so you won't actually be able to delete the storage account yet.

1. Scroll up until you find Overview at the top of the blade on the left of the screen.
2. Select Overview.
3. Select Delete.

You should get a notification letting you know you can't delete the resource because it has a delete lock. In order to delete the storage account, you'll need to remove the delete lock.

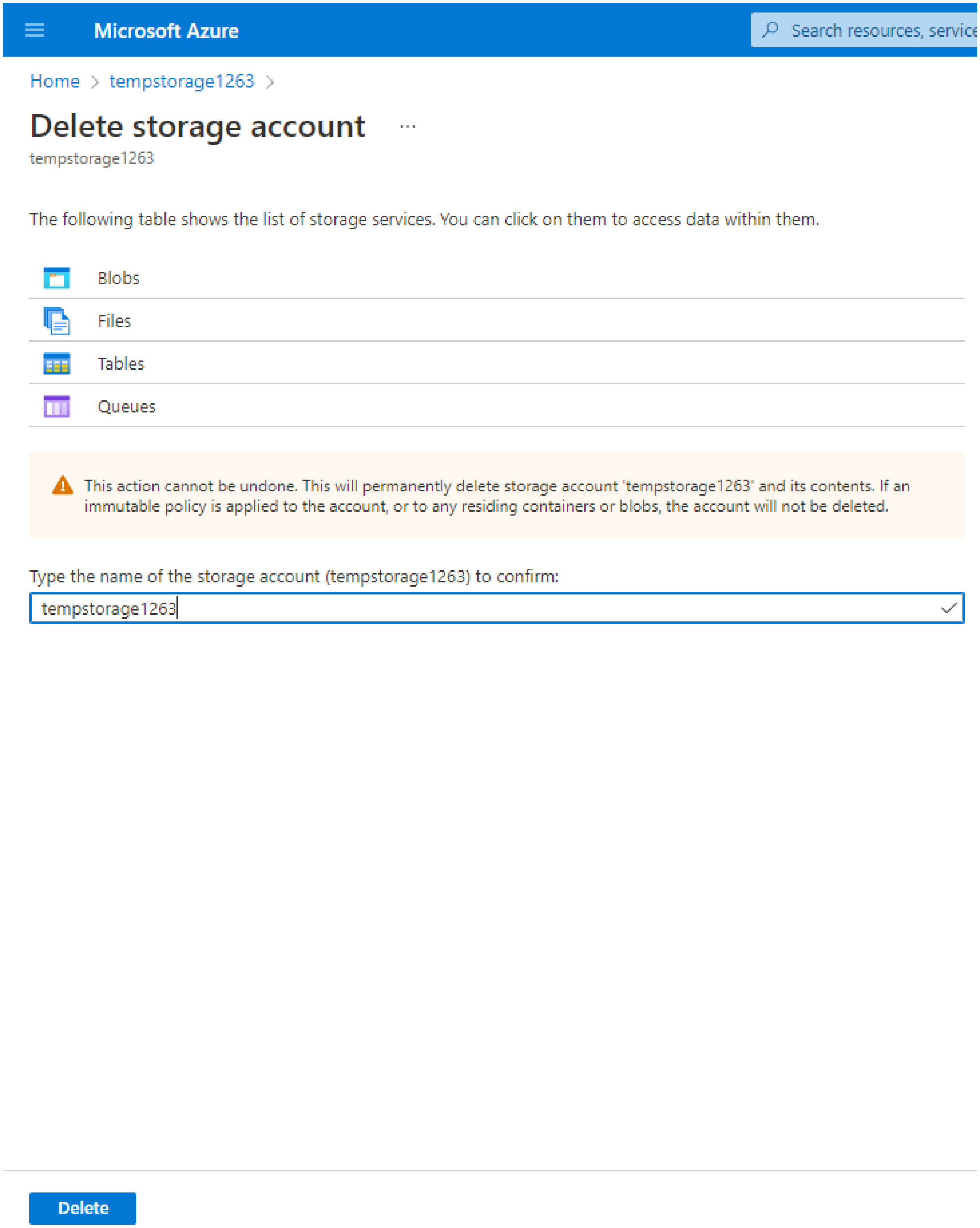


'tempstorage1263' can't be deleted because this resource or its parent has a delete lock. Locks must be removed before this resource can be deleted. Learn more about delete locks

Task 6: Remove the delete lock and delete the storage account

In the final task, you remove the resource lock and delete the storage account from your Azure account. This step is important. You want to make sure you don't have any idle resource just sitting in your account.

1. Select your storage account name in the breadcrumb at the top of the screen.
2. Scroll down until you find the Settings section of the blade on the left of the screen.
3. Select Locks.
4. Select Delete.
5. Select Home in the breadcrumb at the top of the screen.
6. Select Storage accounts
7. Select the storage account you used for this exercise.
8. Select Delete.
9. To prevent accidental deletion, Azure prompts you to enter the name of the storage account you want to delete. Enter the name of the storage account and select Delete.



Delete storage account

tempstorage1263

The following table shows the list of storage services. You can click on them to access data within them.

- Blobs
- Files
- Tables
- Queues

⚠ This action cannot be undone. This will permanently delete storage account 'tempstorage1263' and its contents. If an immutable policy is applied to the account, or to any residing containers or blobs, the account will not be deleted.

Type the name of the storage account (tempstorage1263) to confirm:

tempstorage1263 ✓

10. You should receive a message that the storage account was deleted. If you go Home > Storage accounts, you should see that the storage account you created for this exercise is gone.

Congratulations! You've completed configuring, updating, and removing a resource lock on an Azure resource.

Important

Make sure you complete Task 6, the removal of the storage account. You are solely responsible for the resources in your Azure account. Make sure you clean up your account after completing this exercise.

Next unit: Describe the purpose of the Service Trust portal

Describe the purpose of the Service Trust portal

Completed100 XP

- 3 minutes

The Microsoft Service Trust Portal is a portal that provides access to various content, tools, and other resources about Microsoft security, privacy, and compliance practices.

The Service Trust Portal contains details about Microsoft's implementation of controls and processes that protect our cloud services and the customer data therein. To access some of the resources on the Service Trust Portal, you must sign in as an authenticated user with your Microsoft cloud services account (Azure Active Directory organization account). You'll need to review and accept the Microsoft non-disclosure agreement for compliance materials.

Accessing the Service Trust Portal

You can access the Service Trust Portal at https://servicetrust.microsoft.com/.

The Service Trust Portal features and content are accessible from the main menu. The categories on the main menu are:

- **Service Trust Portal** provides a quick access hyperlink to return to the Service Trust Portal home page.
- **My Library** lets you save (or pin) documents to quickly access them on your My Library page. You can also set up to receive notifications when documents in your My Library are updated.
- **All Documents** is a single landing place for documents on the service trust portal. From **All Documents**, you can pin documents to have them show up in your **My Library**.

Note

Service Trust Portal reports and documents are available to download for at least 12 months after publishing or until a new version of document becomes available.

Next unit: Knowledge check

Summary

Completed100 XP

- 2 minutes

In this module, you learned about some of the features and tools you can use to help with governance of your Azure environment. You also learned about tools you can use to help keep resources in compliance with corporate or regulatory requirements.

Learning objectives

You should now be able to:

- Describe the purpose of Azure Blueprints
- Describe the purpose of Azure Policy
- Describe the purpose of resource locks
- Describe the purpose of the Service Trust portal

Additional resources

The following resources provide more information on topics in this module or related to this module.

- Intro to Azure blueprints is a Microsoft Learn module that covers Azure Blueprints in greater detail.
- Intro to Azure Policy is a Microsoft Learn module that introduces you further to Azure Policy.