# Azure Resource Locks

## (Preventing Accidental Deletion or Modification of Cloud Resources)

**Hsk**

**COURSE FACILITATOR: DR. OLOWOSULE OMOTOLA**

**DATE: OCTOBER 2022**

# Table of Content

# Objective of the StudyObjective of the Study

The general objective of this study is to give a detailed overview about one of the very important features of the Microsoft Azure Cloud Computing platform, Resource Locks.

- Heuristically demonstrate how it can be implemented to prevent accidental deletion or modification of an organisation's cloud resources.
- Examine its relationship to security and most of the best practices associated with it to ensure optimum use
- Analyse how it benefits security and organisations or businesses that use the Microsoft Azure services for their day-to-day business activities and transactions.

# Background of the Study

Day-to-day administration will sometimes require you to lock a subscription, resource group, or specific resource to prevent other users from accidentally deleting or modifying critical resources. When this need arises, you can set resource locks by setting lock levels to CanNotDelete or to ReadOnly you can ensure resources are not deleted or modified.

In the Azure portal, the locks are called Delete and ReadOnly respectively. The CanNotDelete lock allows authorised users to read and modify a resource, but not delete the resource. ReadOnly allows authorised users to read a resource but not delete or update the resource.

Applying the ReadOnly lock is similar to restricting all authorised users to the permissions granted by the reader role. When a lock is applied at a parent scope all resources within that scope inherit the lock. Resources added later, will also inherit the lock from the parent.

The most restrictive lock in any inheritance takes precedence. It's important to note that applying ReadOnly can sometimes lead to unexpected results because some operations that appear to be read operations are actually operations that require additional actions. For example, a read-only lock on a storage account will prevent all users from listing the keys. This is because the list keys operation is handled through a post request because the keys returned are available for right operations.

Similarly, placing a ReadOnly lock on an app service resource will prevent visual studio server Explorer from displaying files for the resource because displaying files requires right access. To create or delete management locks, you must have access to Microsoft.authorization/* or Microsoft.authorization/locks/*actions. Only the built-in owner and user access administrator roles are granted these actions.

Resource locks can be applied to subscriptions, resource groups, or individual resources as required. When you lock a subscription, all resources in that subscription (including ones added later) inherit the same lock. Once applied, these locks impact all users regardless of their roles. If it becomes necessary to delete or change a resource with a lock in place, then the lock will need to be removed before this can occur.

## Relationship to Security and Best Practices Recommendation

In cloud computing, security is a shared responsibility between the Cloud service providers and the cloud customers. While one is tasked with the responsibility of securing the infrastructure that the services run on, the other is tasked with the responsibility of protecting their data and deployed resources to ensure that its integrity and confidentiality is maintained.

While there’s no single remedy, there’s an array of operational practices that can begin to mitigate risks if applied in sustainable ways and the use of resource locks

is one of the ways of achieving this purpose. Like we stated earlier, the intention of using locks is to prevent accidents. A developer might be cleaning up a virtual machine but mistyped the name. An admin might be removing a storage account but forgot to switch subscriptions first. An engineer might be deleting a resource group that has a critical resource in it. Really it does happen.

While cloud infrastructure management is powerful because it enables organisations to rapidly deploy resources, it also makes it just as easy to delete those resources. Accidental resource deletion can have substantial bottom-line effects if product suites are offline and clients are unhappy. Therefore we recommend businesses and organisations and the general cloud customers that adopt cloud services especially that of Microsoft to ensure the construction of a list of all their infrastructure resources that would be unrecoverable.

Good examples of this might include Azure blob storage accounts, public IP addresses and databases. It's also a great opportunity to reflect on the backup and recovery solutions for layers of their infrastructure. Then prioritise this list of critical resources by focusing on the client impact if the resource was deleted. The most important resource is the one that has the potential to cause product downtime and the last one on the list may just affect a few internal developers. From your list, select the top 5-10 resources and jump into the Azure portal and add delete locks to these resources.

## Benefits of Azure Resource Locks

- **Azure Resource Locks as an extra layer of security for Cloud Resources and Environments:** Unauthorised access can cause many damages to an organisation's data, resources. It can affect the integrity and confidentiality of information. With Azure resource locks, cloud resources can be protected from unintentional, unauthorised changes.
- **It prevents deletion and modification of resources:** protects resources and data from accidental deletion or modification.

## Known Resource Locks Issues (Issues with implementing resource locks)

Azure Resource Locks are NOT as simple and as beneficial as they appear on the surface. Focusing on the Cannot Delete lock, there's a lot more going on under the hood that we don't see that also won't be apparent right away and could severely impact operation of resources at a future date.

- A cannot-delete lock on the resource group created by Azure Backup Service causes backups to fail.
- A cannot-delete lock on a resource group prevents Azure Machine Learning from auto scaling.
- The service supports a maximum of 18 restore points. When locked, the backup service cannot clean up restore points.

## The Implementation Process/Step by Step Procedure

The following simple steps illustrate how resource locking is implemented for various scopes or resource types.

Typically, the option to configure resource locking will appear in the **Settings** section for the selected resource. The option will appear as "**Resource locks**" when a subscription is selected and "**Locks**" when other resource types are selected.

To activate a resource lock, do the following:

• Select Locks in the **Settings** blade for the resource, resource group, or subscription that you want to lock.

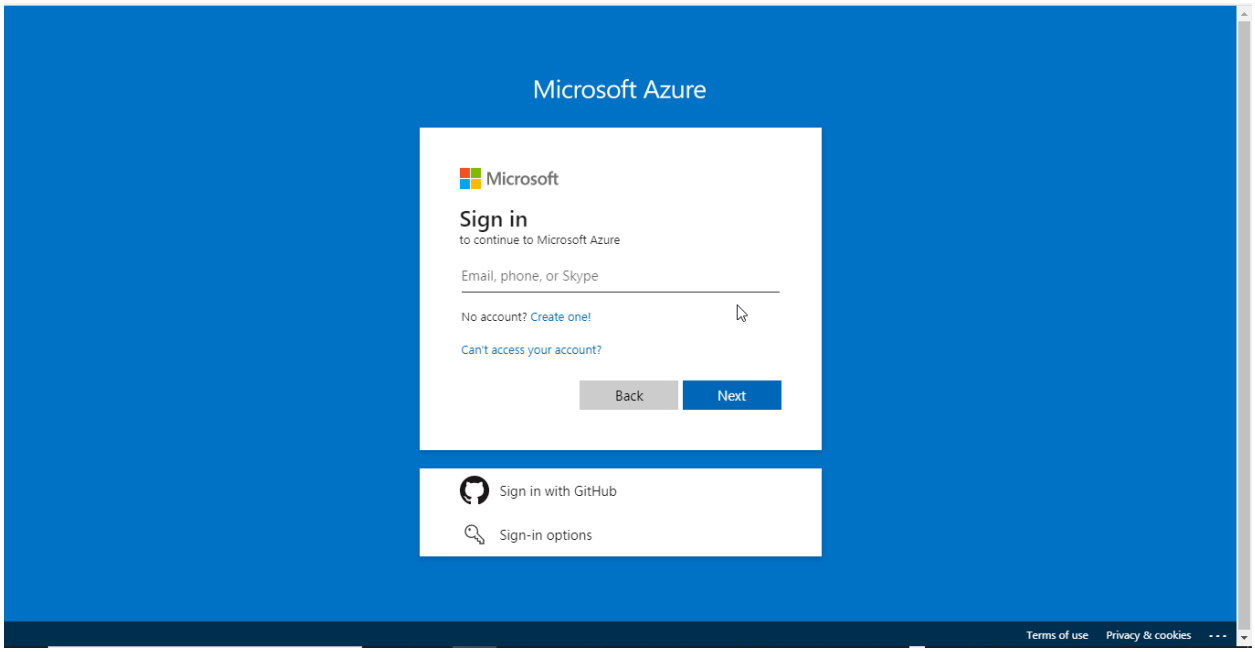
• Select **Add** to add a lock. Select the parent if you want to create a lock at that level. The currently selected resource inherits the parent's lock. You could, for example, lock the resource group to apply a lock to all of its resources.

• Give the lock a name and select a **Lock type** (also known as lock level). You can optionally include notes that describe the lock.

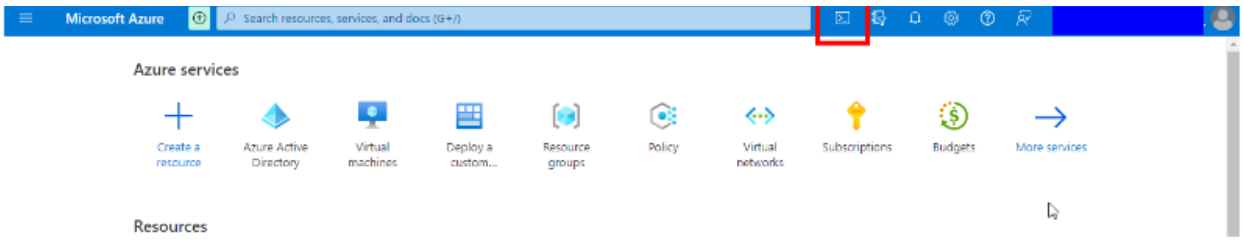
And that is it, the resource is securely protected with respect to the type of lock applied.

## Step by step procedures - Creating the resources to be locked

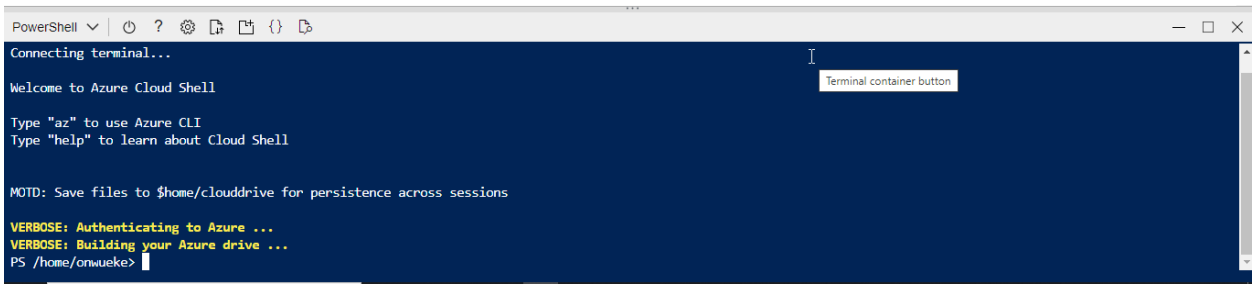
Sign in to the Azure portal



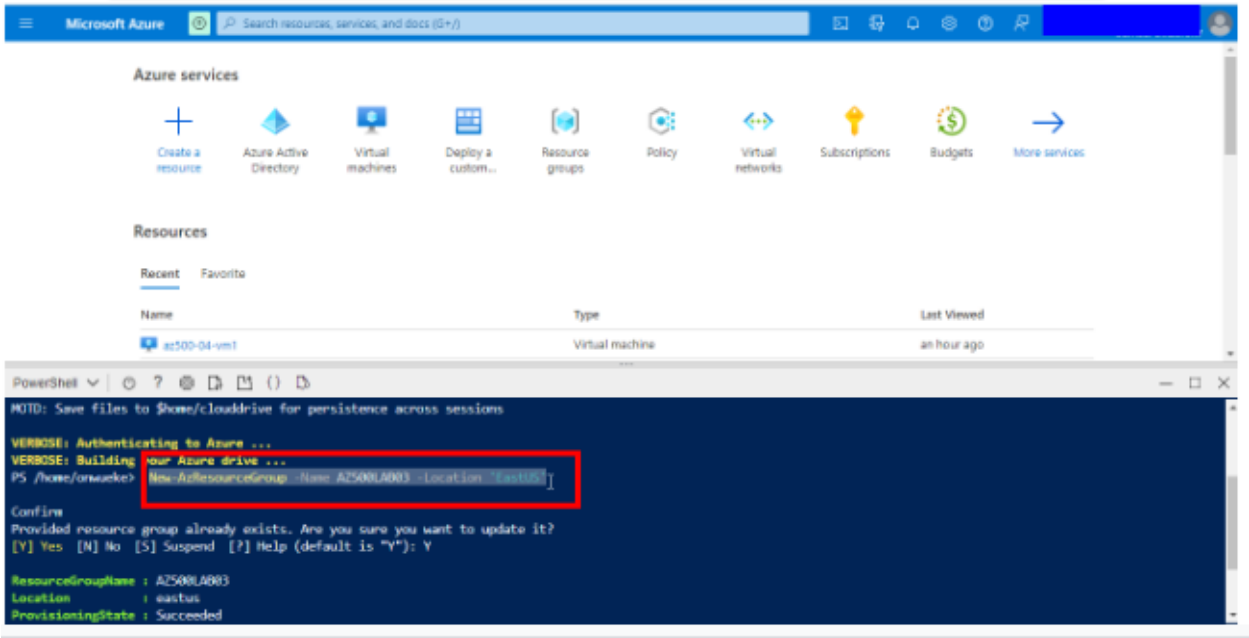

Once in the Azure portal, PowerShell is launched from the Azure Cloud shell, to be used in creating our resources



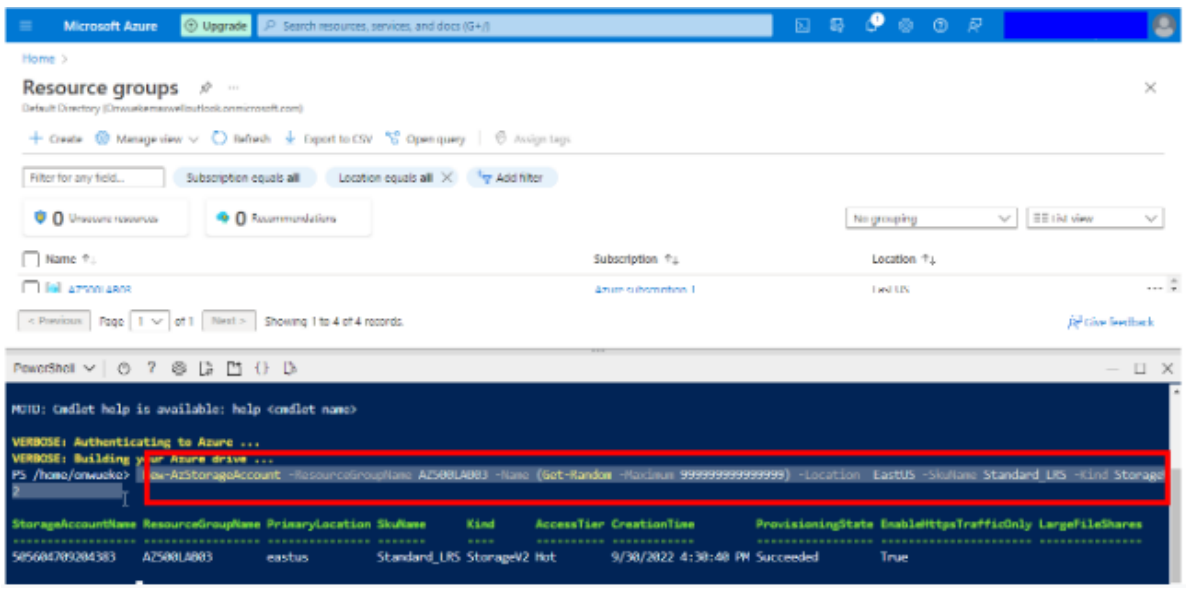

The PowerShell terminal launched and ready for codes



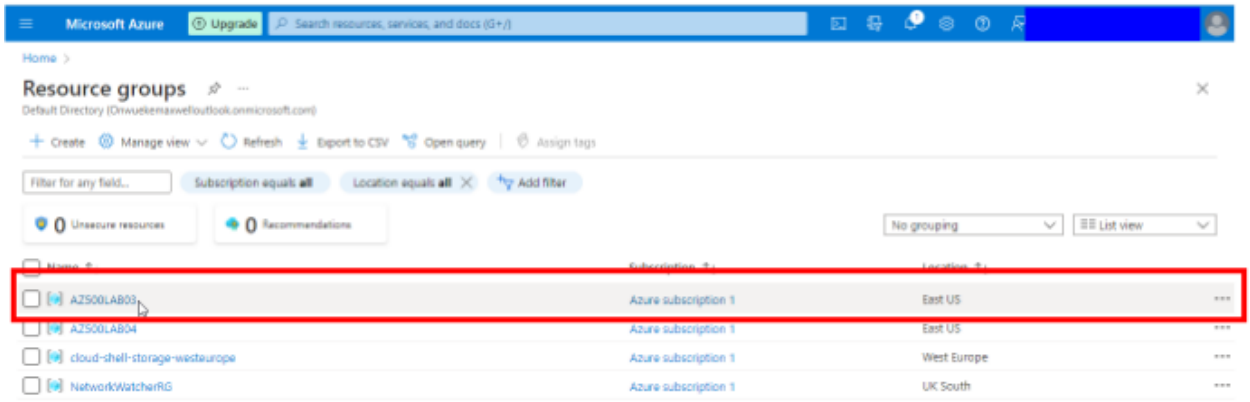
The following command was ran to create a resource group:
`New-AzResourceGroup -Name AZ500LAB03 -Location 'EastUS'`




The following command was ran to create a storage account and was added to the resource group created earlier:
`New-AzStorageAccount -ResourceGroupName AZ500LAB03 -Name (Get-Random -Maximum 999999999999999) -Location EastUS -SkuName Standard_LRS -Kind StorageV2`

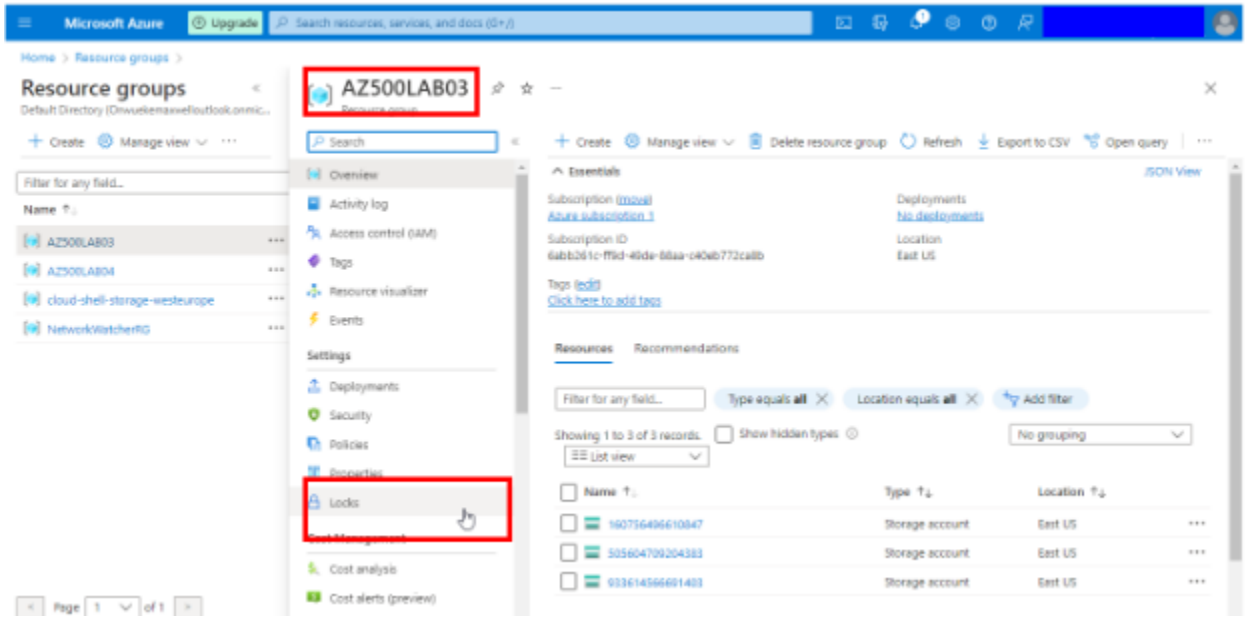


Resource group and storage resource created and ready to be locked

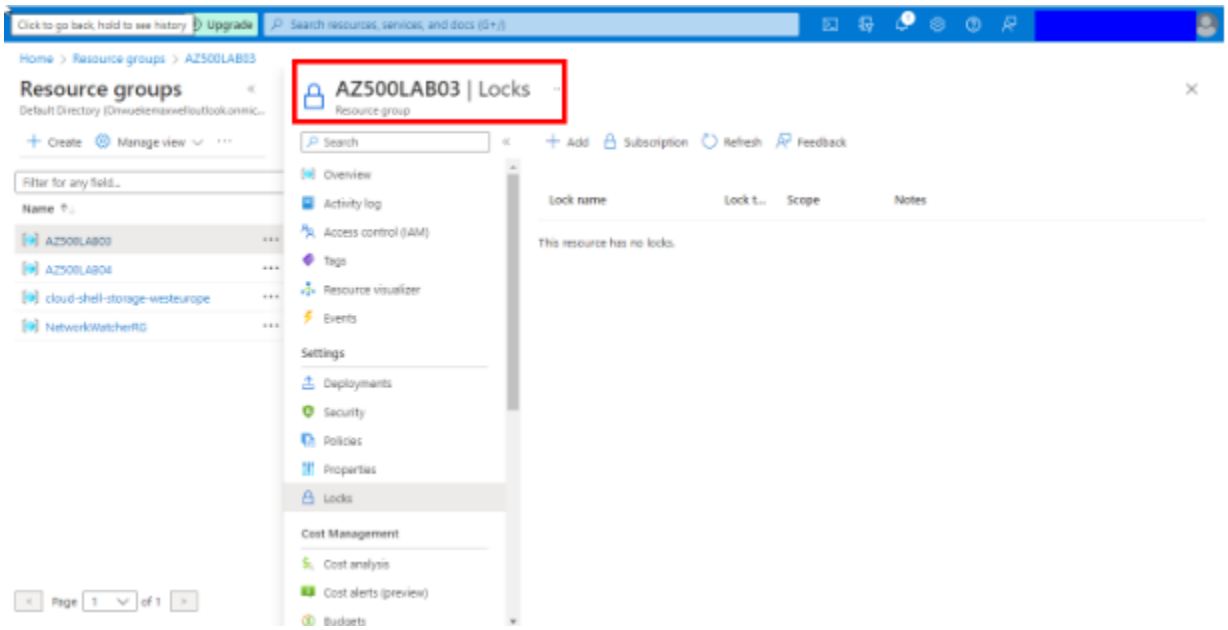



**Step by step procedure - Adding a Read only lock to the resources**

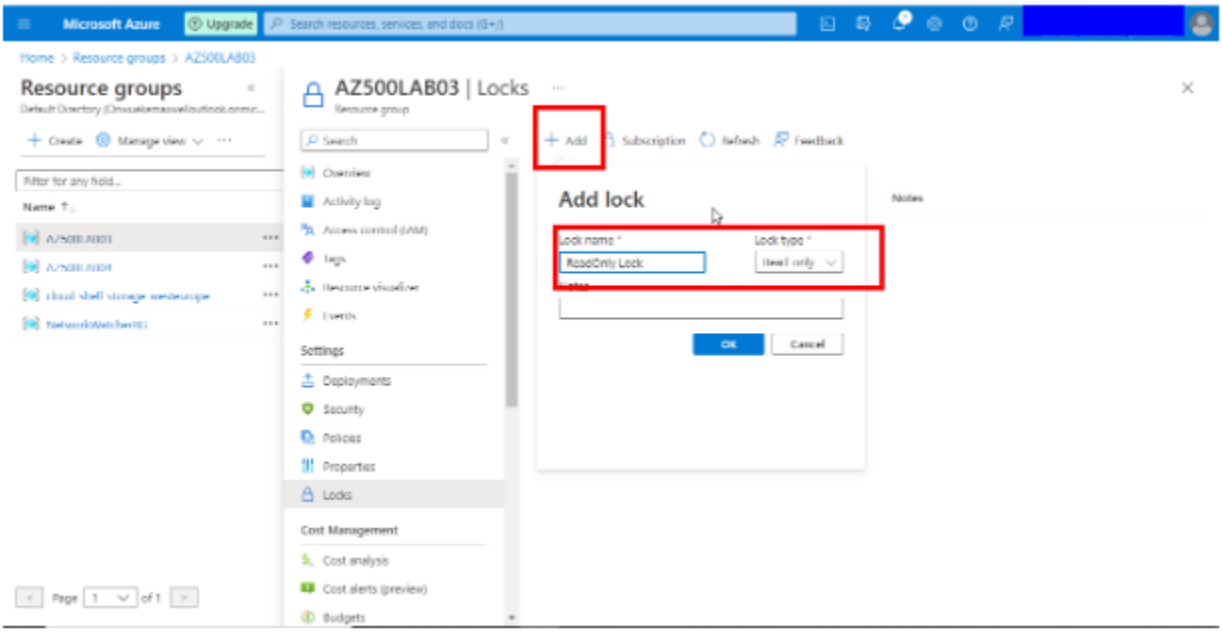
Resource group selected to reveal the Locks setting



The **Locks** setting selected and the **Locks** blade displayed

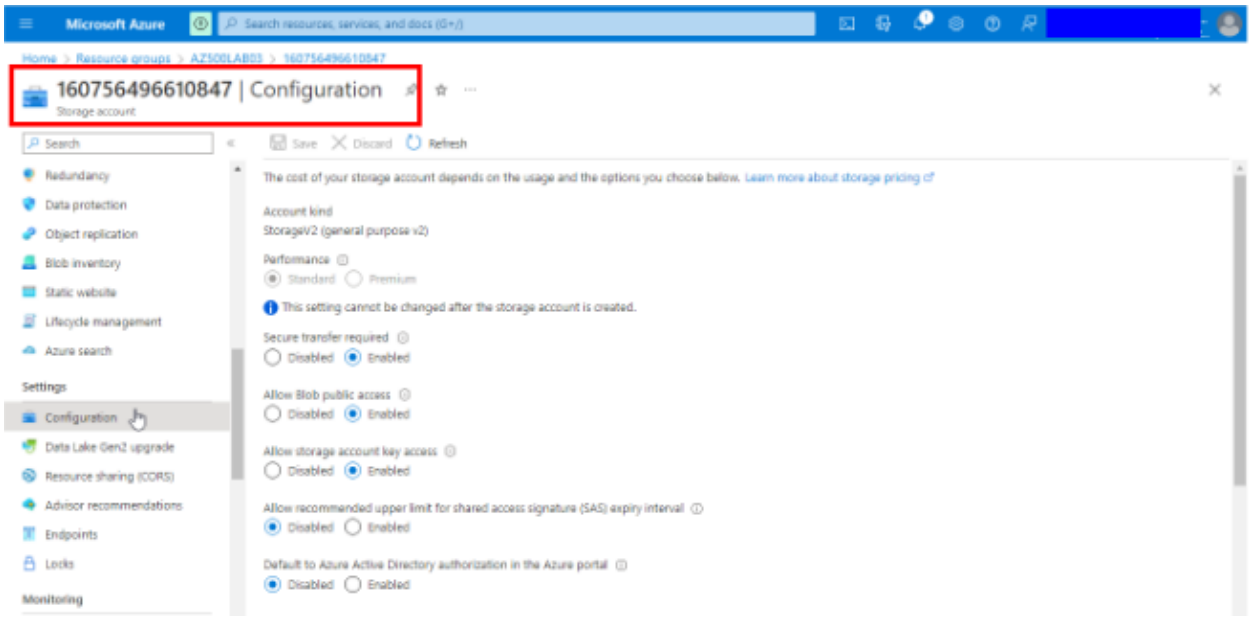



The **Add** locks option selected and **Read only** lock configured

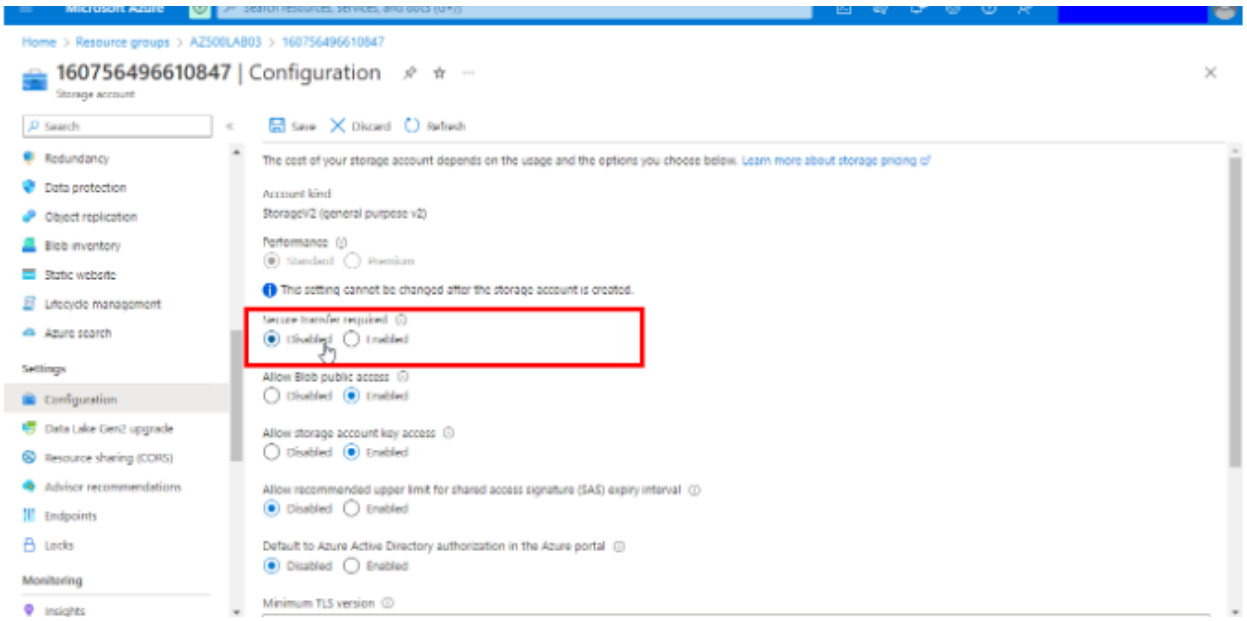


**Step by step procedure - Testing the Read only lock**

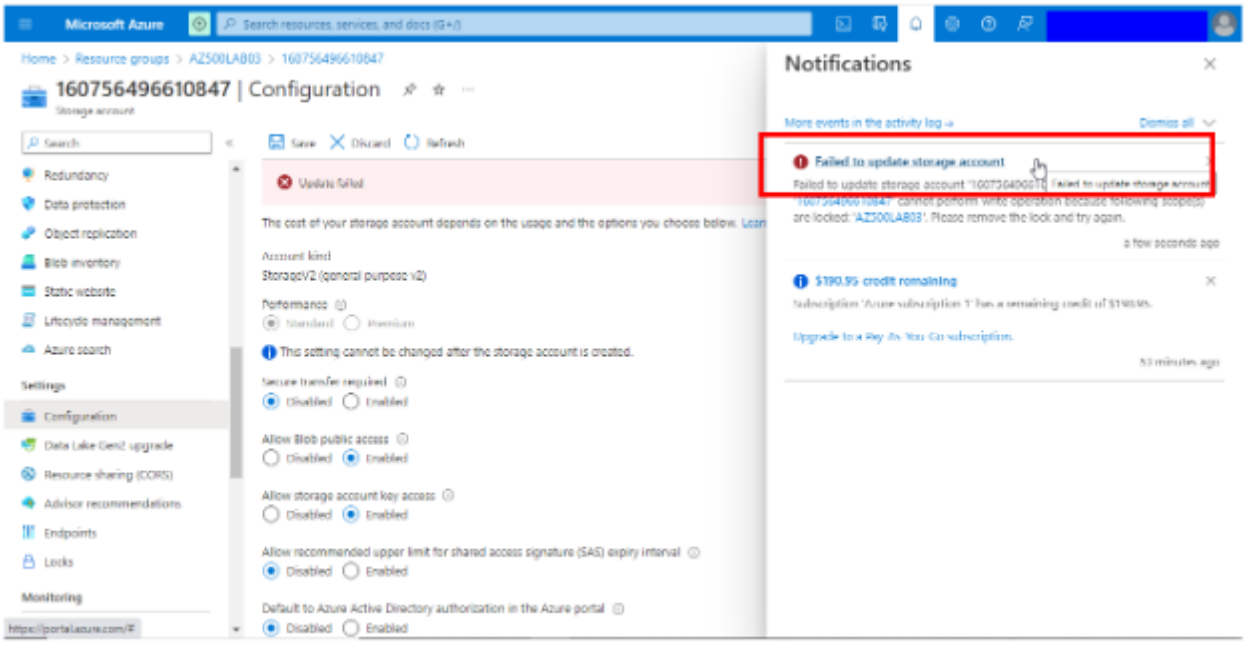
Here the Storage Resource contained in our locked Resource Group is selected to be modified. This will show that resource locks apply to child contents



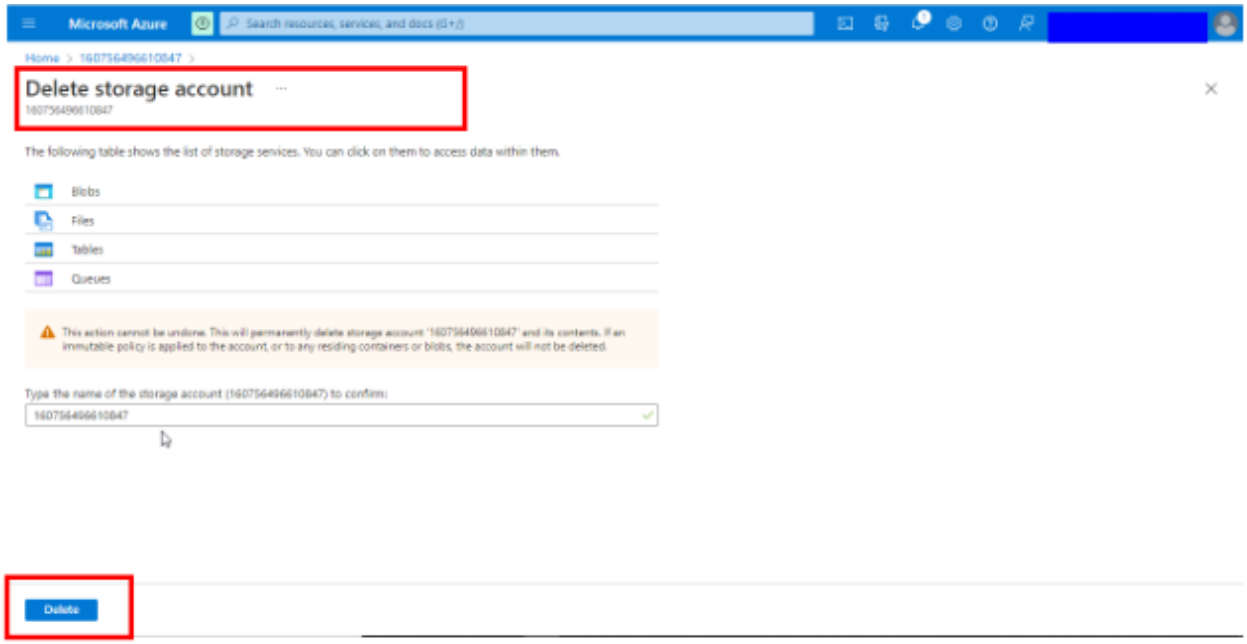

Here we try to change a configuration setting on the Storage Resource by setting the “Secure transfer required” option to Disabled



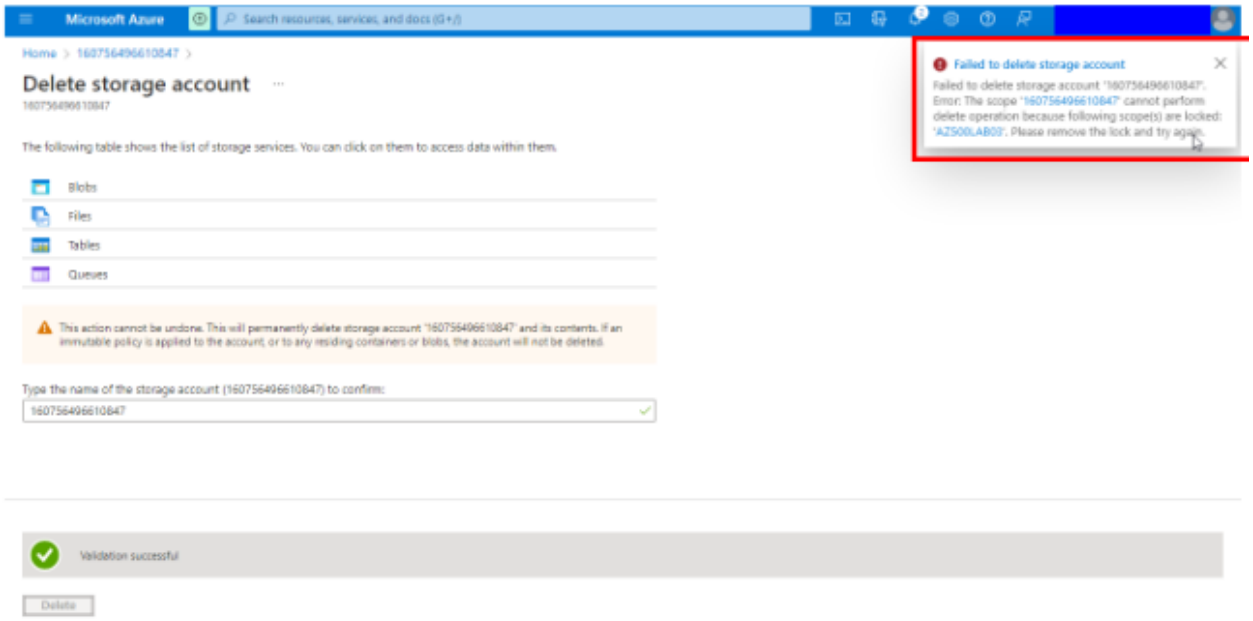
You can spot the notification “Failed to update storage account” because of the Read-only Lock




Here we try to delete the same storage account to show that a Read only lock will also prevent deletion

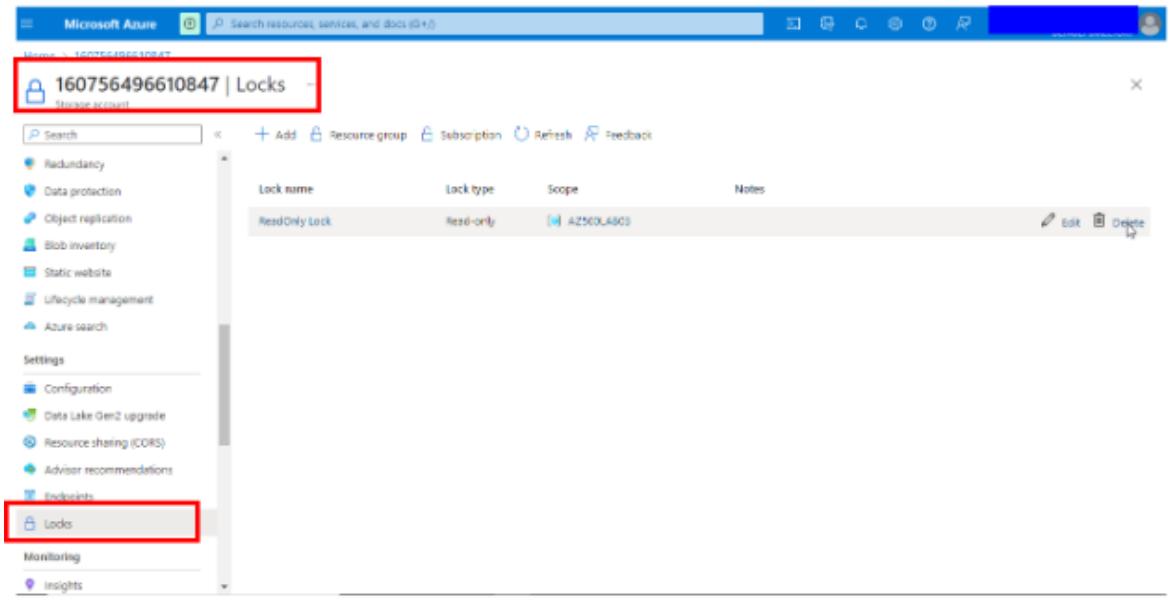


You can spot the notification “Failed to delete storage account” because of the Read-only Lock. You have now verified that a ReadOnly lock will stop accidental modification and deletion of a resource.

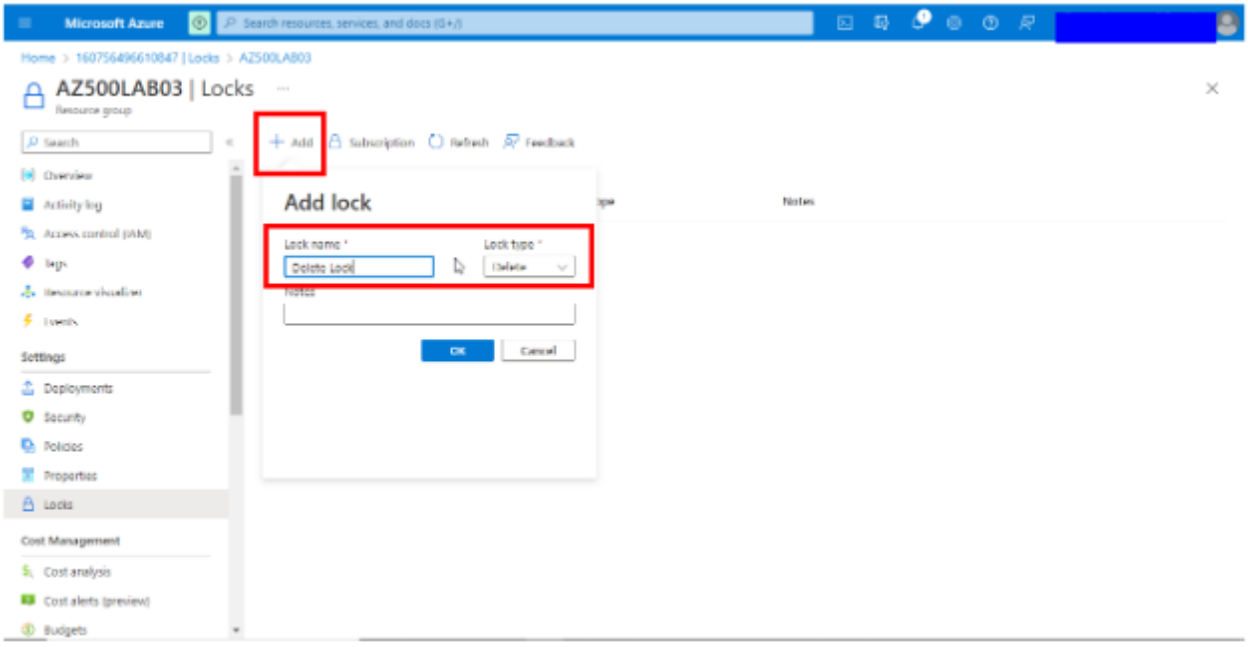



## Step by step procedure - Adding a Delete lock to resources

The Locks setting is selected for our Storage Account and ready to be locked.
*Note: At this point we have removed the Read only lock earlier added.*

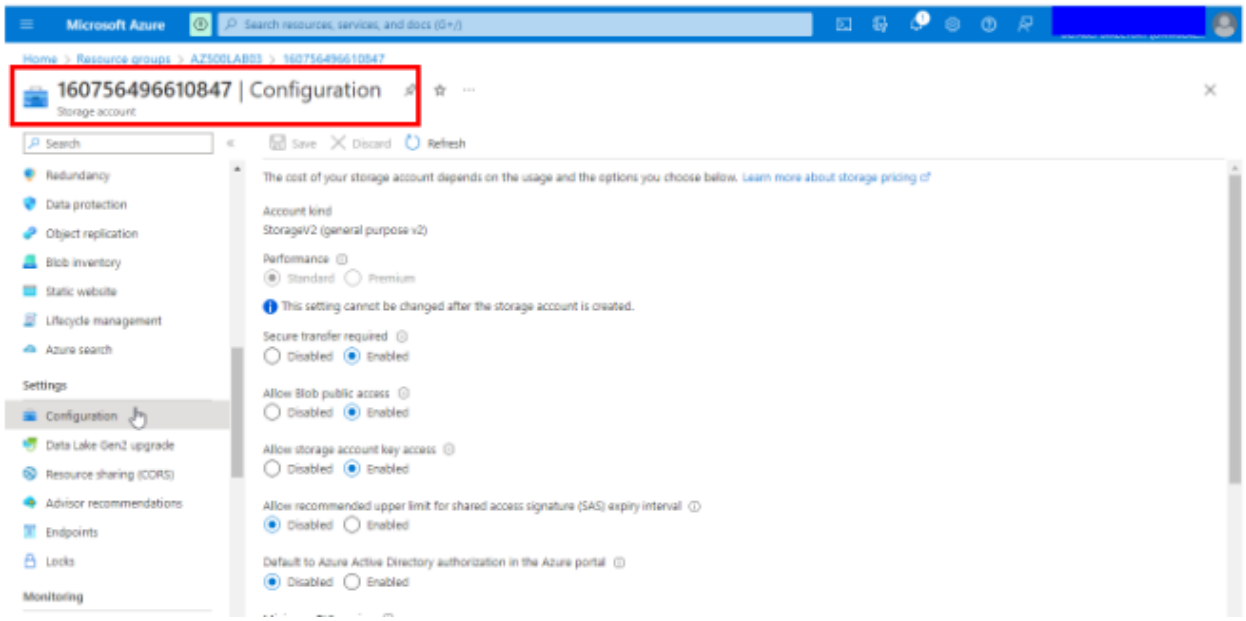


The Add locks option is selected and Delete lock configured

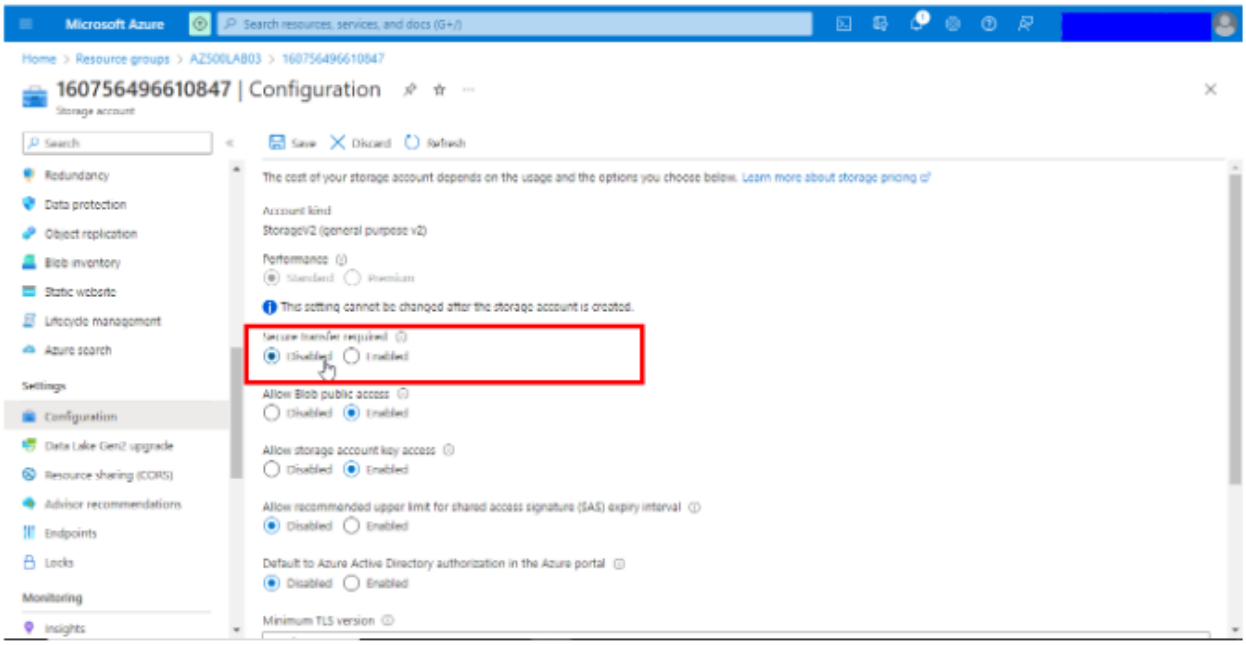



## Step by step procedure - Testing resource modification with the Delete lock applied

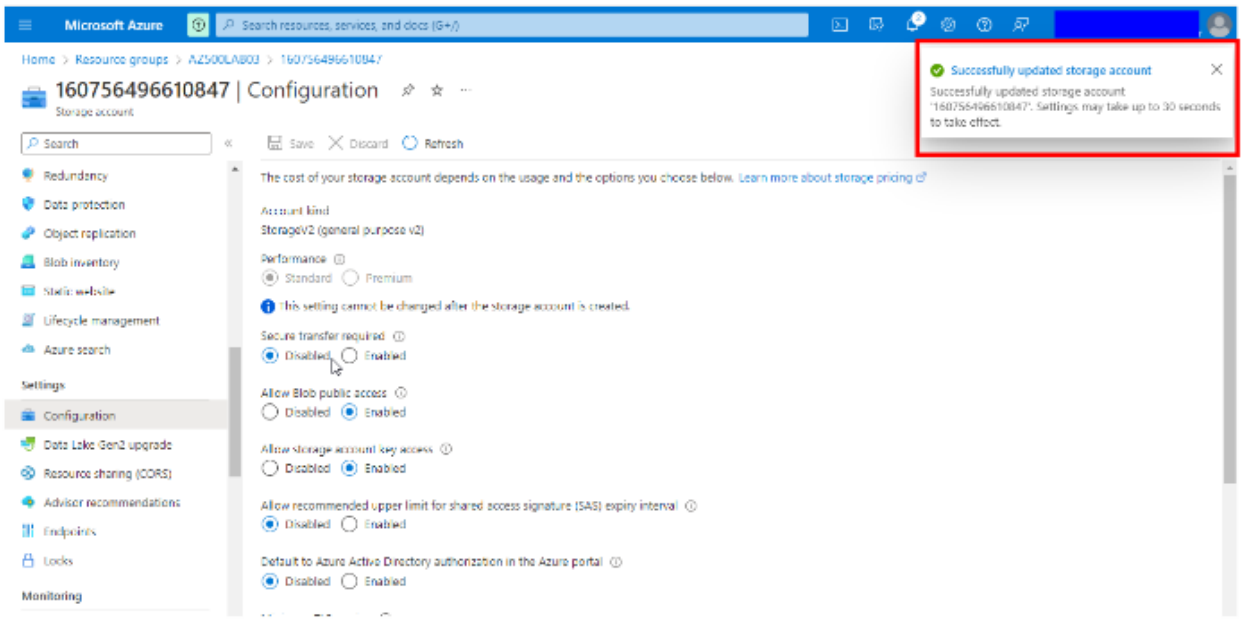
Here the storage resource is  selected to be modified.
*Note: This will show that with the Delete lock applied, modifications can still be made successfully.*



Here we try to change a configuration setting on the storage resource by setting the “Secure transfer required” option to Disabled

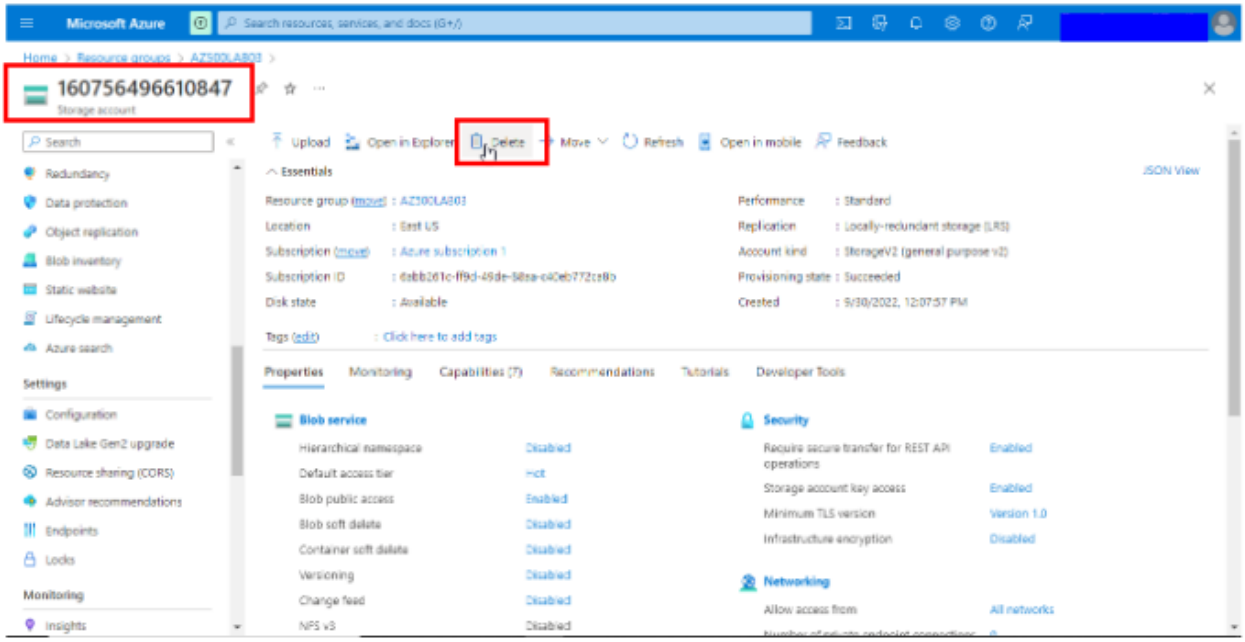



You can spot the notification “Successfully updated storage account” irrespective of the Delete Lock



## Step by step procedure - Testing resource deletion with the Delete lock applied

Here the storage resource is  selected and the Delete button clicked to effect resource deletion.




You can spot the message indicating that the resource cannot be deleted because the resource or its parent has a delete lock, effectively demonstrating the resource lock in effect.

## Challenges Encountered in the Course of the Project

- Some team members were initially not forthcoming with their individual task
- Individual schedules affected our timing
- We encountered general network issues during meetings
- Some team members were hardly available for meetings

## Learning experience

- Experienced excellent collaboration and teamwork even though the project was done virtually.
- We had the opportunity to meet and brainstorm with brilliant minds.
- Getting to have an in-depth knowledge about Resource Locks and how it works.
- Working with the Azure portal provided more familiarity and the workflows of the platform
- So many concepts and ideas to understand within a limited time frame.

# References:

**The Microsoft Learn Portal: https://learn.microsoft.com/en-us/**